OPTIMAL CONTROL POLICIES FOR THE RECOVERY OF SOCIO-PHYSICAL SYSTEMS

by

Hemant Gehlot

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Lyles School of Civil Engineering

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Satish V. Ukkusuri, Co-chair

Lyles School of Civil Engineering

Dr. Shreyas Sundaram, Co-chair

School of Electrical and Computer Engineering

Dr. Seungyoon Lee

College of Liberal Arts

Dr. Andrew Liu

School of Industrial Engineering

Approved by:

Dr. Dulcy M. Abraham

To my parents and my sister.

ACKNOWLEDGMENTS

I would like to first express my sincere gratitude to my advisors Dr. Satish V. Ukkusuri and Dr. Shreyas Sundaram. I am deeply grateful to Dr. Ukkusuri for his guidance, support, and patience during all phases of my PhD. I am also very grateful to Dr. Ukkusuri for giving me the opportunity and freedom to work on various interdisciplinary problems. I am also deeply grateful to Dr. Sundaram for keeping me motivated to solve theoretical problems and for providing me valuable guidance to prove the theoretical results in this dissertation. I would also like to express my sincere thanks to other members of my PhD advisory committee, Dr. Seungyoon Lee and Dr. Andrew Liu, for providing me valuable comments to improve the research in this dissertation.

I am also thankful to Dr. Harsha Honnappa, Dr. Kent Quanrud and Dr. Ray Decarlo with whom I have had many insightful discussions during my stay at Purdue.

I would also like to thank my former and current colleagues in the research groups of Dr. Ukkusuri and Dr. Sundaram for their support: Feng Zhu, Xianyuan Zhan, Arif M. Sadri, Xinwu Qian, Wenbo Zhang, Tho V. Le, Dheeraj Kumar, Fasil Sagir, Takahiro Yabe, Nishtha Mahajan, Zengxiang Lei, Lu Ling, Jiawei Xue, Sangung Park, Rajat Verma, Xiaowei Chen, Shagun Mittal, Eunhan Ka, Sara El Alaoui, Omar Hamim, Surendra R. Kancharla, Ashish Hota, Aritra Mitra, Amritha Prasad, Mustafa Abdallah, Lintao Ye, Tong Yao, Russell Shirey, Kananart Kuwaranancharoen, Lei Xin, Baike She, Yijing Xie and Jiajun Shen. I would also like to thank Aashish Jain, Sonali Srijan and Saumya Sangoi for their support during my PhD.

I am also deeply grateful to my parents and my sister for their constant support, without which it would not have been possible to complete this journey.

This dissertation is supported by National Science Foundation award CMMI 1638311. The author is grateful for the support. However, the findings reflect the opinions of the author alone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Disasters often cause significant damage to infrastructure systems and result in displacement of communities. Recovery after such events becomes complicated due to the presence of various interdependencies. For example, there are dependencies that are present between various types of infrastructure systems. Similarly, a displaced social group may decide to return after a disaster only when a sufficiently large number of groups in its social network return and certain infrastructure components in its neighborhood are repaired. In addition, there could be constraints on the availability of resources and manpower for recovery after disasters. The recovery decisions in such resource-constrained scenarios are often guided by practitioner knowledge, experience, and convention. Therefore, it is important to develop policies that can be efficiently implemented for optimal recovery of such systems.

In this dissertation, a mathematical abstraction of the above recovery problem is considered. Since the number of possible recovery strategies under such scenarios grows exponentially fast as the number of damaged infrastructure components increases, it is intractable to enumerate all possible solutions to find the optimal strategy. In addition, existing computational frameworks like integer programming and dynamic programming become computationally burdensome as the size of the problem increases. Therefore, this dissertation focuses on rigorously characterizing optimal and near-optimal policies for the recovery of socio-physical systems. First, we focus on the recovery of infrastructure components that face accelerated deterioration after disasters due to processes such as floods and corrosion. For the scenario when there is a single repair agency, it is proved that the optimal policy is to target the healthiest component at each time-step when the deterioration rate is larger than the repair rate whereas the optimal policy is to target the least healthy component at each time-step when the repair rate is sufficiently larger than deterioration rate. Next, various extensions of this problem involving heterogeneity in the deterioration and repair rates as well as weights across components, dependencies between components, and multiple repair agencies are considered. It is shown that the characterized policies depend on the relationship between the deterioration and repair rates for all these cases. After that, the problem of maximizing the number of displaced social groups that return, while addressing

the repair of infrastructure components, is considered. Optimal and near-optimal policies are characterized for various cases of this problem as well.

Finally, we provide greedy heuristics (motivated by the above optimal policies) and other computational methods to solve cases of the problem for which optimal policies are not currently known.

1. INTRODUCTION

1.1 Background

Most natural as well as man-made systems often face disruptions. One such example of disruptions would be a natural disaster such as a hurricane, flood, or earthquake that can cause huge loss of life, extensive damage to infrastructure and massive displacement of people [1]. Some disruptions such as fires and epidemics can be rapidly evolving, with the recent examples of Australian fires 2019-20 and COVID-19 pandemic reminding us the consequences of such events [2], [3]. Man-made events such as computer servers getting affected by a hacker [4], non-recurrent congestion in traffic networks due to unexpected events [5], etc., are also examples of disruptions. A common phenomenon in the aforementioned examples is that the states or health values of the components of a system change from their normal values due to disruptions. The objective of the recovery process after a disruption is to bring back the health values of the components to normal values. A normal state of a component is defined as the average state of the component on the days that do not face any significant incidents. Note that some of the aforementioned disruptions not only affect *physical* components such as infrastructure in disasters but also affect *social* components such as displacement of households after disasters. In such scenarios, there are several dependencies in the recovery of the components. For instance, the recovery of infrastructure after a disaster also affects the return patterns of displaced communities [6], illustrating the influence of physical components' recovery on the recovery of social components. Similarly, there can be dependencies between the recovery of social components, for e.g., the tendency of a displaced household to return back increases as more and more neighboring households return back after a disaster [7]. Similarly, there can be dependencies between the recovery of physical components, for e.g., before repairing sections of electric water pumps it might be necessary to first repair some sections of power network [8]. Therefore, we focus on the recovery of both the social and physical components and refer to the systems consisting of both the components as *socio-physical* systems.

In certain scenarios, it may not be possible to recover all the components to normal health values. For instance, infrastructure components face accelerated deterioration due to processes such as corrosion and flood waters after disasters. Due to these processes, some of the components may deteriorate to such a level that they require full replacement or their recovery becomes very expensive, which is undesirable [9]. Similarly, there is a limited amount of time before computer servers become fully compromised when attacked by a hacker or an infection [4], and if there is a limited availability of resources for protection, then some of the servers may get fully compromised. Likewise, in fires, the objective of fire-fighters is to ensure that the fire does not enter a state known as *flashover*, where there is little hope of saving the affected property or individuals [10]. In the case of multiple fires, the firefighters might have to decide the regions that they would target if there is a limit on the resources that are available for fire-fighting. In the aforementioned examples, if only one or a few components can be repaired at a time due to constraints on resources and/or manpower, some of the components may deteriorate extensively (or permanently fail) by the time they are started to be repaired. Note that such resource-constrained scenarios in disaster recovery are observed across the globe. For example, US had experienced shortage in its disaster response force during the hurricane season in 2018 [11]. Likewise, countries such as US, UK, India, etc., frequently face shortage in their fire-fighting personnel [12]–[14]. Similarly, in the recovery of social components also it may not be possible to bring the health values of all the components to the normal state. For instance, empirical studies have found that the return of people after disasters is dependent on factors such as the recovery of utilities, the influence from returned members in their social network and incentivization measures taken by recovery agencies for returning back [7], [15]. Since there can be a limit on the number of infrastructure components that can be recovered and the budget that is available for incentivizing the people to return back, it might not be possible to bring the health values of all the social components to normal values. Under these constraints, the objective of the recovery agencies would be to maximize the recovery of components.

Note that the main focus of this dissertation is disaster recovery, although the results that will be obtained could also be extended to other applications. We will consider an abstraction of the recovery problem. Figure 1.1 presents an example of the recovery problem consisting of socio-physical components after disasters. The solid filled orange boxes denote fully healthy physical components before a disaster in the left most upper corner of the illustration. The solid filled green boxes denote fully healthy social components. A solid edge starting from a social component i to social component j denotes the relationship that component i socially influences component j. A dashed edge starting from a physical component k to a social component l represents the relationship that social component l is dependent on physical component k. In the lower part of the left most column of the figure, the health values of the physical components A, B and C reduce from their full values (shown by the reduced solid fills in the boxes representing the components), representing the case that infrastructure components get damaged with varying extent after a disaster [16]. The social components are represented by green lined boxes in the figure representing displaced households after a disaster. We first focus on the recovery of physical components after a disaster (see the middle column of the figure) because the recovery of infrastructure components is an important factor that the displaced households take into consideration while deciding to return back [6], [17]. Suppose there is an agency that is responsible for repairing the physical components in an area. The agency follows a sequence of targeting the physical components such that if a component is targeted by the agency then it is fully repaired before the agency targets another component. If a physical component is not being targeted by the agency at a time then its health decreases due to the aforementioned deterioration processes until the health value reaches the state of permanent failure. The health of a physical component does not change any further once it reaches either full health or permanent failure. In the sequence that is followed in this example, the agency fully repairs components B and C but component A permanently fails by the time the agency starts to target it. The last column of the figure focuses on the recovery of social components. Due to budget constraints at most one social component can be incentivized to recover in this example (this represents the scenario where some households are incentivized to return back by the recovery agencies [18]). Social component D is first incentivized and therefore it gets recovered. After that, social component F also recovers due to social influence from component D and because one of the physical components (i.e., component C) on which it is dependent on, is recovered. However, social component E does not recover because the physical component A is not recovered (representing the case that households also take into account the recovery of infrastructure components such as utilities while making the decision to return).

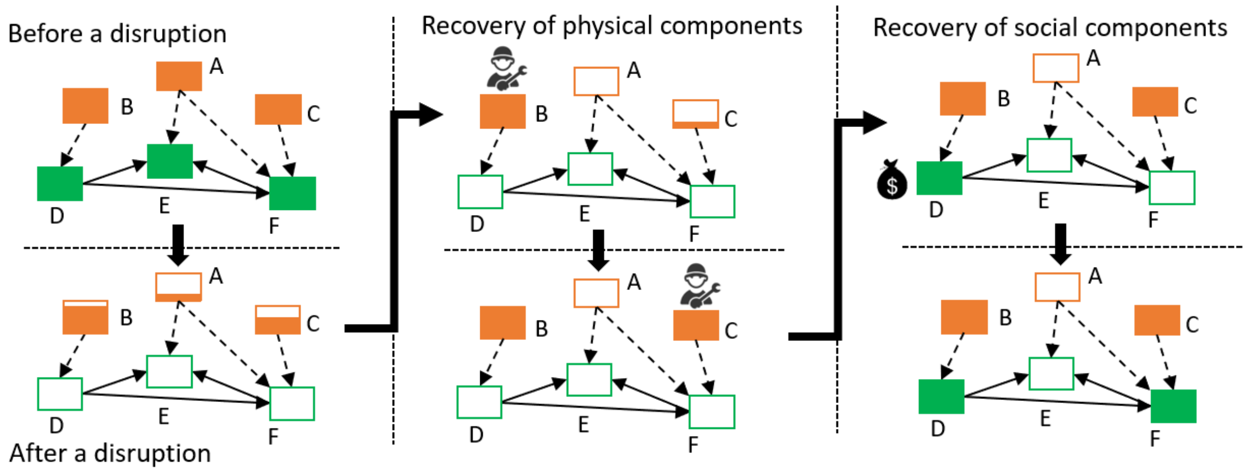

Figure 1.1. An illustration to show the recovery of socio-physical components after a disaster.

Consider the problem that only focuses on the recovery of physical components such that a single agency is responsible for the recovery and the agency targets to fully repair the components one by one (as in the second column of Figure 1.1). The total number of sequences in which the agency can target the components is equal to $N! = (N)(N-1)\cdots 1$. Note that the function $N!$ grows rapidly with N because $N!$ asymptotically grows faster than exponential functions like 2^N [19]. Therefore, enumerating all the sequences to find the sequence that maximizes the number of components that are repaired to full health is not an efficient method because the number of damaged components can be very large in catastrophic disruptions such as disasters. Therefore, we need to find efficient ways for solving the problem. Note that the recovery problem represents a *control* problem because of the following reasons. Consider the equation $x_{t+1} = f(x_t, u_t)$, where x_t is the state of the system at time-step t, u_t is the control action taken at time-step t and f is the function describing the system dynamics. Note that $u_t = \mu(x_t)$, where μ is a control policy that represents the mapping from states to actions. Consider a stage of the recovery process, say the recovery of physical components (in the second column of Figure 1.1). Then, x_t represents the vector denoting the health values of the physical components at time-step t and u_t denotes the physical component that is targeted by the agency at time-step t. The health value of the component that is targeted at time-step t increases and the health values of the remaining components decrease if they are not fully healthy or permanently failed. Thus, the health values of components at time-step $t+1$ (denoted by x_{t+1}) are a function of the health values of components at time-step t (denoted by x_t) and the control actions taken at time-step t (denoted by u_t). Therefore, characterizing optimal control policies can provide an efficient way to solve the recovery problem.

Thus, several important questions arise in the recovery of socio-physical systems. For instance, how to characterize efficient control policies that the recovery agencies should follow while targeting physical components, given that they have a choice of targeting many components but can only target one or a few components at a time? How should the recovery agencies choose social components for incentivizing to return back to their normal states when there are budget constraints on the incentives that are available and the recovery of social components is also dependent on the recovery of physical components? The answers to these questions would allow recovery agencies to make informed choices towards maximizing recovery after disruptions. The aim of this dissertation is to obtain the answers of such questions.

1.2 Current practice and literature review

In this section, first the current practices that are being followed in the recovery after disruptions like disaster recovery and fire fighting are reviewed. Then, we also review the existing literature to understand how such problems or related problems have been addressed so far.

1.2.1 Current practice

We first focus on the recovery of physical systems such as infrastructure after disasters. Infrastructure systems consist of services such as energy, water, sanitation, transportation and communications that are essential for social and economic activities [8]. Infrastructure recovery has been a key focus of disaster recovery due to their direct influence on the return patterns of displaced people, market and economy of the affected region, etc. Infrastructure loss can also affect emergency and social services in a disaster, just when they are needed the most. For instance, debris and flooded pathways can impede the movement of ambulances, search and rescue crew, etc. [8]. Therefore, the quality of basic services such as utilities, transportation, telecommunications, etc., matters greatly to a disaster affected region's pace of recovery [20].

Currently, National Disaster Recovery Framework (NDRF) of Federal Emergency Management Agency (FEMA) is the guide for effective recovery after all types of disasters and emergencies in the United States [21]. NDRF establishes a common platform and forum for how the whole community builds, sustains, and coordinates delivery of recovery capabilities. One of the main objectives of NDRF is to ensure coordination among various actors involved in this process such as federal, state and local governments, NGOs and private sector entities. The paper [22] suggested that formalized coordination and control are necessary to enhance coordination between organizations. Coordination of actors serves to unify strategy [23], reduces duplication of services [24] and reduces the waste of resources [25]. The paper [22] observed that the coordination efforts among government agencies were limited and the level of coordination among various organizations, which was measured in terms of mean network density, reduced significantly (about 68%) in the long run. The paper [26] also notes that the lack of coordination between local government and nonprofit organizations is a problematic characteristic of the disaster assistance framework. Therefore, we need to characterize optimal control policies that can be used to coordinate and maximize the recovery efforts after disasters. Also, while deciding the allocation of recovery projects among various agencies, cost is an important factor that is taken into account [27]. In some cases, if the recovery process needs to be sped up then time taken to repair is also considered along with the cost of repair. In those scenarios, $A + Bx$ rule is used to decide on how to allocate the recovery projects to various agencies, where A is the cost taken to repair, B is the time taken in repairing and x is a factor that governs the relative weight of these two factors. Therefore, we need to take into account any budget and time constraints that are there while ensuring that the recovery is maximized.

Note that the optimal decision making in such scenarios is highly dependent on the quality of infrastructure damage information that is available. Damage or disruption is generally measured in terms of the number of bridges that are closed, pipelines that are damaged, etc. [8]. There are various agencies or institutions that are responsible for estimating the overall damage after disasters. For instance, FEMA is responsible for estimating the overall damage after disasters in the United States. FEMA uses the estimation to evaluate the cost of damage and helps the state and local agencies when the costs overwhelms the resources of local and state authorities. FEMA takes inputs from various individuals and agencies at local and state level to do the estimation. There are also other agencies that conduct damage estimation, for e.g., Federal Highway Administration (FHWA) conducts damage estimation of roads as it is solely responsible for restoring public roads under the Emergency Relief (ER) program [27]. There are a number of techniques to assess damage ranging from highly accurate assessment methods like door-to-door and site assessments to efficient but not so accurate methods like self-reporting, fly-over surveys, windshield surveys and geospatial analysis [16]. The paper [28] also concludes that there are inaccuracies in damage estimates. For instance, underestimation can occur if some damage is not observed while inspection and overestimation can occur due to predisposition of unusually severe disasters. Due to lack of proper estimation of damage, the processes that govern the deterioration of an infrastructure may also not be exactly known. Therefore, there could be uncertainty in the way health value of an infrastructure component changes with time.

NDRF also points out that successful recovery upholds the values of timeliness in coordinating and efficiently conducting recovery activities and delivering assistance. It also minimizes delays and loss of opportunities. Lost opportunities can be in terms of permanent failure of some infrastructure components that require complete repair, which is considerably expensive and thus undesirable [29]. For instance, resurfacing of asphalt driveway costs about 2 USD/square feet, whereas digging of old driveway and complete replacement can cost in the range of 6-11 USD/square feet [30]. This difference between the minor repair and full replacement rates can create huge difference as the scale of damage after disasters is usually very high. For instance, U.S. Department of Transportation released about 29 million dollars in emergency funds to repair roads after Hurricane Sandy [31]. Therefore, appropriate timeliness in repairing infrastructure systems can significantly reduce the economic losses after disasters. After the Hurricane Sandy, Empire State News reported that "rapid remediation would reduce damage from the corrosive effects of water, pollutants and other factors" to which New York comptroller Thomas P. DiNaPoli responded that "the sooner we get contractors on the ground to assist residents and business owners, the faster New York will be back on its feet" [31].

NDRF also emphasizes the importance of social networks and other ties in disaster recovery. Individuals are better situated to manage their recovery once their basic needs such as shelter, food, and reunification with family and friends, are met [21]. Since the role of social influence is also important when the displaced people decide to return back (in addition to the importance of recovery of infrastructure), recovery agencies need to ensure that a sufficient number of people are incentivized to return back, so that they can subsequently influence people in their social network to return back. Therefore, recovery agencies provide some incentives to displaced people to return back [18]. However, there can be financial constraints and therefore the agencies may have to select the most influential people for incentivization.

For infrastructure recovery, U.S. Army Corps of Engineers (USACE) is assigned as the coordinating agency. After disasters, USACE coordinates with other agencies like FEMA, which is responsible for community planning and capacity building. In order to meet the goals of disaster infrastructure recovery, NDRF follows the public-private collaborative structures outlined in National Infrastructure Protection Plan (NIPP). NIPP first identifies and prioritizes different infrastructure systems across the country and then implements risk management activities across the identified systems. Prioritization can be done based on various factors such as public health and safety impacts, direct and indirect economic impacts and cascading impacts and interdependencies [32]. But there has been very less focus on the research of disaster recovery of different infrastructure systems in comparison to pre-disaster mitigation and preparedness. This is because infrastructure repair and restoration are largely thought of as emergency response problems where decision making is guided by practitioner knowledge, experience and convention [8], [33]. In addition, recovery process in disaster scenarios is complex due to accelerated deterioration processes that affect infrastructure systems [29] and also because of the limited resources and manpower that are available for recovery.

We now focus on the recovery of physical components after rapidly unfolding events like fires. National Fire Protection Association (NFPA) provides the information on appropriate resource allocation for firefighting agencies before the occurrence of fires [10]. The first step in the allocation of resources and personnel to different regions is based on the risk assessment of the regions that takes into account various factors such as potential property loss and life hazard to the population of the region. It has been found that in spite of these risk assessments, it is sometimes not possible to fully equip different agencies in order to tackle unusually large scale disasters due to constraints on budget and personnel [10], [34]. For instance, resources of state firefighters were not adequate during Australian Fires of 2019-20 and therefore they had to resort to crowdfunding for obtaining necessary supplies for firefighting [35]. In addition, the timeliness of response is very important in fires because the aim is to ensure that fires do not reach the state of flashover as mentioned before [10]. Furthermore, firefighting response also gets complicated like disaster recovery due to the presence of uncertainty coming from uncertain behavior of fire, inaccurate data, etc. [36], [37]. Like natural disasters, recovery agencies also need to incentivize some displaced people to return back after fires [38], so that they can influence other people to return back. Since there can be constraints on the budget that is available for incentivizing, we conclude that the challenges that are faced in recovery after fires is similar to the challenges faced during natural disasters' recovery.

1.2.2 Existing literature

We first focus on the recovery of physical components where there is a single repair agency that seeks to repair the components and can only target one component at a time. The health values of the physical components deteriorate if they are not being targeted by the agency until the health values reach the state of *permanent failure*. The objective of the agency is to target the components in the sequence that maximizes a performance criterion, e.g., maximizing the number of physical components that are brought back to full health (or *permanent repair* state). Note that our problem is motivated by the fact that after disruptions, components such as infrastructure, infected servers and fire-affected regions deteriorate rapidly in comparison to the deterioration faced during normal times

[9], [39] and therefore the health of a physical component does not significantly vary due to normal deterioration processes once the state of full health is reached. We first present the studies that focus on the recovery decisions in post-disaster scenarios. Some studies develop network flow-based models for physical infrastructure networks where maximizing flow is considered equivalent to maximizing recovery [40]. The paper [41] also develops a network flow based model that maximizes the resilience of interdependent infrastructure systems while minimizing the cost associated with restoration. Some studies in operations research have focused on finding an optimal sequence of targeting different roads in order to remove debris after disasters [42], some have focused on finding optimal schedules for repair of road networks [43] and some works have focused on developing decision-making systems for road-recovery considering human mobility [44]. However, these studies mainly focus on the recovery of individual elements belonging to a certain type of physical infrastructure (e.g., roads in a transportation network) rather than focusing on sequencing decisions for repairing portions of physical infrastructure with varying damage/health levels. Furthermore, these studies do not model deterioration of physical infrastructure over time if a component is not being repaired. Such deterioration is commonly observed in real-world post-disaster scenarios; for example, road infrastructure deteriorates over time in flooded areas after hurricanes [9].

The sequencing problem for targeting physical components falls into the general class of optimal control and scheduling of switched systems [45], [46] (or, more generally, hybrid systems [47], [48]). A switched system consists of multiple subsystems that are governed by different dynamical rules such that only one subsystem is active at each point of time. In our problem, each component corresponds to a subsystem, and the switching rule corresponds to which component the controller chooses to target for repair at each time-step. Since a repair agency chooses which component to target for repair at fixed intervals of time (e.g., on an hourly or daily basis in the case of natural disasters), our problem comes under the class of discrete-time switched systems. The main source of complexity in the optimal control and scheduling of discrete-time switched systems is the combinatorial number of feasible switching sequences [48]. In the past decade, there have been some advances in theoretical results and computational frameworks for solving switched systems. For example, the papers [45] and [46] characterize optimal/near-optimal control and scheduling policies for discrete-time switched linear systems with linear or quadratic cost/reward functions. However, there are no theoretical results or computational frameworks that efficiently solve the optimal control and scheduling problems for all types of switched systems and most results are formulation dependent [49]. Therefore, optimal control and scheduling of switched systems remains an area of active research.

At a high level, other switched system control problems such as scheduling of thermostatically controlled loads [50], [51] also have similarities to our problem. These studies characterize scheduling control policies so that the states (e.g., temperature) of the components (e.g., rooms) in the system always stay in a given interval. In these studies, the system becomes unstable (equivalent to the notion of permanent failure in our problem) if the state of a component violates any of the two interval thresholds. In contrast, our problem has one desirable threshold and one undesirable/failure threshold corresponding to the states of full health (or permanent repair) and extensive damage (or permanent failure), respectively. This difference leads us to characterize optimal policies of different types depending on the problem conditions. For example, we will show that non-jumping policies, where switching between different components is not allowed until a component is permanently repaired, turn out to be optimal under some conditions. In contrast, the aforementioned studies related to switched systems do not characterize non-jumping policies to be optimal; indeed, jumping is necessary to meet the objectives of those problems. Analogies to our problem can also be found in model predictive control problems where the objective of the problems is to ensure that the components' states lie within a set of desirable states and there is an associated penalty cost if the states reach an undesirable value [52], [53]. Similarly, the objective of our problem is to maximize the number of the components whose states can be brought back to the desirable threshold value (permanent repair) without ever reaching the undesirable threshold (permanent failure).

Our problem also has similarities to the problem of allocating resources (e.g., time slots) at a base station to many time-varying competing flows/queues [54]–[56]. However, these studies do not consider permanent failure of components or flows being serviced, and instead focus on either bounding the long-term state of the system, or maximizing long-term throughput or stability.

Problems of a similar flavor can also be found in optimal control of robotic systems [57]–[59] that persistently monitor changing environments. The objective of our problem is different from these studies as we focus on maximizing the number (or weight) of components whose states (i.e., health) reach a desirable threshold (i.e., permanent repair) whereas most of these studies minimize the average cumulative uncertainty across a set of locations over a given time horizon. Job scheduling problems with degrading processing times [60]–[62] as a function of job starting times also have analogies to our problem. A major difference between job scheduling and our problem is that in the former, a job is considered to be late if its *completion* time exceeds the corresponding due date, whereas in our problem, a component is considered to be failed if its health reaches the state of permanent failure before the repair agency *starts* to control it. Another important difference is that a job is completely processed even if its completion time exceeds the corresponding due date; in contrast, a component in our problem cannot be targeted if its health value reaches the state of permanent failure. Our problem also has similarities with scheduling analysis of real-time systems [63]; there, the analysis focuses on real-time tasks that become available for processing at different times; in contrast, all the components in our problem are available for control starting at the same time. Patient triage scheduling problems [64] also have some analogies to our problem; these problems only focus on non-jumping sequences and characterize optimal sequences assuming that if a task has less time left before expiration than another task, then the former task also takes less time to be completed than the latter task. However, this assumption does not hold for the problems that we consider in this report.

Machine repair problems (also known as machine interference problems) [65]–[67] also have high-level analogies to our problem. In these problems, machines operate for a period of time until they fail and are subsequently repaired by a set of servers. The objective of the servers is to decide the sequence for repairing failed machines in order to optimize a performance criterion. In some machine repair problems [66], the optimal policies are index based policies that assign the machine with the smallest index to a server at each time-step (an example of an index could be the product of cost per time and repair rate). This is analogous to our problem because we also characterize index based policies (e.g., targeting the healthiest component at each time-step). However, these studies do not consider permanent failure and permanent repair of machines because machines can switch between operating and non-operating states multiple times, and typically the objective of these studies is to optimize criteria such as average profit per unit time, whereas the objective of our problem is to maximize the number or weight of components that are permanently repaired. Similarly, the Gittins index has been frequently used in applications such as multi-armed bandit problems, stochastic job scheduling problems, etc., to compute the optimal policy [68], [69]; however, there are differences between the existing job scheduling studies and our work as described above.

If multiple repair agencies are considered in our recovery problem then that would require allocating the physical components to different agencies. Note that there are several works that have focused on optimal allocation of resources after disasters [70], [71]. However, most of the existing studies do not focus on characterizing optimal policies to allocate infrastructure components that deteriorate with time (if not being repaired).

We now focus on the recovery problem consisting of both the social and physical components. As mentioned before, factors such as lack of recovery of utility facilities and damage to road infrastructure, influence the decision to return back after disasters [17]. The paper [6] also acknowledged that residents' returns were impeded by storm debris on the roadways, lack of utility services, etc. Thus, the return of individuals and the recovery of physical infrastructure is interlinked. The role of social networks in the return behavior of people after disasters has also been documented in many studies [7], [72]. It is believed that the influence of social networks on migration patterns is cumulative, i.e., as more and more neighbors in the social network of an individual migrate, the tendency of an individual to migrate also increases [7], [73]. Since the objective of agencies that focus on the recovery of communities could be to maximize the recovery of social components like maximizing the return of displaced households after disasters, we focus on maximizing the spread of influence across a social network under the constraints on resources available for recovery.

There are many studies that have focused on the problem of *influence maximization*, which maximizes the spread of influence across a network with the constraint on the total number of nodes that are activated in the beginning (these nodes are referred as the *seed nodes*). Since the problem is NP-hard, it is unlikely to efficiently compute the optimal solution, and therefore the paper [74] focused on finding efficient algorithms that approximately solve the problem. One of the models for influence propagation that the paper [74] studied is the *linear threshold model*. In linear threshold model, each node in the social network has a threshold and an inactive node becomes active if the sum of influence from all of its active neighbors is at least equal to its threshold. The paper [74] considered randomized thresholds in the linear threshold model. It has been shown that exactly computing the spread of influence (i.e., the total number of nodes that become active) for a given set of seed nodes under the linear threshold model is #P-hard [75], implying that is unlikely to find an efficient algorithm to optimally solve that problem. Noting that it is possible to deterministically estimate the thresholds in real-world through surveys, the papers [76] and [77] define the *deterministic linear threshold model (DLTM)* where the thresholds of nodes are deterministic. It has been proved that the spread of influence for a given set of seed nodes can be computed in polynomial time under *DLTM* [77] and therefore we will focus on *DLTM* in this dissertation.

In the aforementioned studies, only the role of social influence is considered in the recovery of social components without considering the recovery of physical components. However, a household displaced after a disaster may not decide to return to its home unless some of the infrastructure components in its residential neighborhood are repaired and a sufficiently large number of people in its social network return back [15]. Therefore, we consider an additional set (or layer) of nodes (apart from the nodes in the social network) that are termed as *physical nodes* that cover (or map to) one or more social nodes. Since there are two types or layers of nodes, we term the combined network consisting of social and physical nodes as a *bilayer* network. A physical node exists in either one of the two states at a time, say *opened* and *closed*, where an opened (resp. closed) physical node represents a repaired (resp. damaged) infrastructure component in the context of disaster recovery. Therefore, a necessary condition for an inactive node to become active is that it should be covered by at least one of the opened physical nodes. Since infrastructure components deteriorate after disasters [9], it might not be possible to repair all the components in the available budget as some of the components may deteriorate extensively by the time they are targeted due to lack of resources and/or manpower [78]. Thus, we consider a constraint on the total number of physical nodes that can be opened.

Note that the paper [76] proved that the influence maximization problem under *DLTM* is NP-hard to approximate within any constant factor for the case when one or two active neighboring nodes are required to activate an inactive node but there exists an efficient algorithm that computes near-optimal solutions when only one active neighboring node is required to activate an inactive node [76]. Therefore, we will focus on the case where an inactive node in the social network requires at least one active neighboring node and at least one opened physical node covering it to become active.

1.3 Goals and objectives

From the previous sections, it can be concluded that there could be scenarios where there is a shortage of resources and manpower for the recovery of socio-physical components after a disruption. In certain scenarios, recovery gets further complicated due to deterioration processes that affect physical components. In such situations, it has been found that often the decision making is guided by practitioner experience and convention, which need not be optimal. Therefore, the focus of this dissertation is on the optimal recovery of socio-physical systems under the aforementioned constraints. We consider an abstraction of the recovery problem and mathematically define the problem. We argued before that finding the optimal solution through enumeration may not be efficient and therefore we need to find efficient ways for solving the problem. Since the recovery problem can be considered as a control problem, characterization of optimal control policies can provide an efficient way for solving the recovery problem.

From the review of existing studies, we conclude that most of the studies on disaster recovery focus on the recovery of a particular infrastructure network and do not consider deterioration. We also reviewed problems in the existing literature that have high level analogies to our recovery problem and listed the differences between our problem and the existing studies. From the literature review, we conclude that there are no existing studies that provide the characterization of optimal control policies for the recovery problem that we focus in this dissertation. Therefore, the goals of this dissertation can be divided into three parts:

1. To characterize optimal or near-optimal control policies for maximizing the recovery of socio-physical components after disruptions.
2. To develop computational frameworks and heuristics for solving the recovery problem.
3. To provide examples and guidelines for implementation of the above results.

Note that the major focus of this dissertation is towards the first goal, i.e., characterizing optimal or near-optimal policies for recovery. In order to meet the goals of this dissertation, we divide the work of this dissertation into several parts. As mentioned before, there exist several dependencies in the recovery of socio-physical systems. We will model different dependencies in a phased manner such that we will first analyze simpler instances of the problem and will then add more complexities to the problem. Therefore, we will first only focus on the recovery of physical components. In the recovery of the physical components, we will first consider the scenario where there are no dependencies and will later introduce dependencies between the physical components. Also, we will first focus on the case when there is a single agency that seeks to repair physical components and then focus on the case when there are multiple repair agencies. Finally, we will consider the case when there are dependencies between both the social and physical components. Note that approaches such as computational frameworks and heuristics compliment analytical approaches when the theoretical analysis becomes very complex. When all types of dependencies in the socio-physical systems are considered, the general problem may become intractable to solve and therefore computational frameworks and heuristics will be provided. Finally, real-world data will be used to the estimate the inputs of the considered problems and compare various solution approaches as mentioned above. Therefore, the objectives of this dissertation are as follows:

1. To characterize optimal or near-optimal control policies for the recovery of physical systems (when there is a single repair agency and without dependencies).

2. To characterize optimal or near-optimal control policies for the recovery of physical systems under dependencies.

3. To characterize optimal or near-optimal control policies for the recovery of physical systems when there are multiple repair agencies.

4. To characterize optimal or near-optimal control policies for the recovery of sociophysical systems.

5. To develop computational frameworks and heuristics to solve the general recovery problem.

6. To provide guidelines for initializing the inputs of the recovery problem through real-world data.

1.4 Research approaches

The research approaches that will be applied in this dissertation can be divided into the following two categories: 1) analytical approaches and 2) computational methods. In the following paragraphs, these approaches are explained in detail.

1.4.1 Analytical approaches

The first category of approaches involves performing mathematical analysis to characterize optimal or near-optimal policies. An advantage of analytical methods like characterizing optimal or near-optimal policies is that they provide efficient algorithms that have provable guarantees on the distance of the computed value from the optimal value. As mentioned before, the analysis will be divided into two parts where first the focus will be on maximizing the recovery of physical components and then the combined recovery of both the social and physical components will be considered. First, consider the problem where the objective is to maximize the recovery of physical components. Recall that there is a single agency that seeks to repair the physical components and the health values of the components deteriorate if they are not being targeted by the agency until the health values reach the state of permanent failure. We will first examine the properties of optimal policies for a *base* problem and then we will analyze various extensions of the base problem. In the base problem, the rates of repair and deterioration as well as the weights (denoting the relative importance of components) will be considered homogeneous across all the components.

For the base problem, we will mathematically analyze to check if the optimal policy is a function of the repair and deterioration rates. For example, it might happen that targeting the healthiest component at each time-step is optimal under certain conditions but targeting the least healthy component at each time-step is optimal in other conditions. For certain instances of the problem, it might also happen that it is unlikely to efficiently characterize optimal policies and for those cases we would try to prove that the problem is NP-hard [79]. Once the problem is shown to be NP-hard, we will target to characterize an approximation algorithm (or a near-optimal policy) that can be used to efficiently solve the problem such that the obtained value is guaranteed to be close to the optimal value. In some cases, it might even be NP-hard to approximate the problem within any constant factor from the optimal value and therefore we will target to prove the same for those cases.

In the base problem, we will assume that the weights and the rates of deterioration and repair are homogeneous across all the components. After analyzing the base problem, we will consider heterogeneity in weights as well as in deterioration and repair rates. Heterogeneity in weights represents the varying importance of different physical components, for e.g., weight of a physical component could represent the number of social components that are dependent on the physical component. Heterogeneity in repair rates represents the fact that various physical components need not be of the same size and therefore repair rates may vary across physical components. Also, the mechanisms governing the deterioration of various components can be different and thus heterogeneity in deterioration rates is justified.

Note that there can be several dependencies between various physical components such as infrastructure systems. For instance, in order to bring back the functionality of water systems (such as irrigation pumps), it might be necessary to first repair some sections of the power network [8]. We will extend the base problem to model such dependencies. Dependencies between the physical components could be modeled through edges between the components. Another extension of the base problem that we will consider is when there are multiple repair agencies and each agency charges a cost for repair such that the authority that allocates the components to the repair agencies has a budget [80], [81]. For this extension of the problem, we will analyze to check if the properties such as submodularity hold for the sets of components that are allocated to agencies under various conditions and characterize optimal or near-optimal policies based on that analysis [82].

We now focus on the problem that focuses on the recovery of both the social and physical components. We will represent the social components and the relationship between them through a directed graph, where the components are represented by the nodes of the graph and the relationships between the components are represented by the edges of the graph. As mentioned before, physical components are represented by an additional set of nodes called the *physical* nodes. Every physical node covers a set of social nodes so that all the social nodes that are covered by a physical node represent the social components that are dependent on the corresponding physical component. A social node can exist in any one of the two states at a time, *active* and *inactive*, depending on whether the corresponding social component is recovered or not, respectively. Similarly, a physical node has two states at a time, *open* and *closed*, depending on whether the corresponding physical component is recovered or not, respectively. We will consider the *progressive* cases [74] where an active social node does not become inactive again and once a physical node gets opened it does not closes again. We will derive the interactions between the social nodes from the *linear threshold model* [74], so that a necessary condition for an inactive social node to become active is that a sufficiently large number of its neighboring nodes should be active, representing the condition that a social component requires sufficient social influence to become active. Another necessary condition for an inactive social node to become active is that it should be covered by at least one of the opened physical nodes, representing the condition that the recovery of social components is also dependent on the recovery of physical components. We will consider discrete time-steps to represent the resolution of time at which an inactive social node decides to become active or not by checking the two aforementioned conditions. A set of physical nodes are opened at time-step zero and the states of all the physical nodes remain the same henceforth. As mentioned before, it might not be possible to recover all the physical components and therefore we consider a constraint on the total number of physical nodes that can be opened. A set of k social nodes is activated at time-step zero, which are referred as *seed* nodes. Since a social component such as a community or a household consists of multiple individuals, each social node has an associated weight. We will be focus on the problem whose objective is to determine the physical nodes that should be opened and the seed nodes that should be activated in order to maximize the total weight of the social nodes that get activated. Finally, we will extend this problem to the case when the health values of the physical nodes lie in the interval [0,1] and the set of physical nodes that get recovered is determined by the sequencing decisions that are followed in repairing them (i.e., considering deteriorating physical components as before).

1.4.2 Computational frameworks and heuristics

Although analytical methods like characterizing policies provide provable guarantees on the computed value from the optimal value, computational frameworks and heuristics compliment analytical approaches in solving optimization problems. For instance, the latter approaches are useful when the mathematical analysis becomes intractable. The following computational frameworks will be tested to model the general recovery problem:

1. *Integer programming (IP)* is an optimization program in which there are one or more variables that are constrained to be integers. IP is a relevant computational framework for the recovery problem because the recovery agencies need to decide one or more components that they target, i.e., the control actions in the recovery problem are discrete. However, there are no algorithms known to efficiently (i.e., in polynomial time) solve an arbitrary IP [83].

2. *Dynamic programming (DP)* is an optimization program that solves the problem by recursively dividing the problem into subproblems [84]. It very useful in solving control problems because a control problem has an associated system dynamics and DP divides the problem into subproblems at different time-steps. However, it may not be possible to efficiently compute the optimal solution using DP because the number of subproblems that need to solved can increase exponentially with the problem size [85].

We will explore the above approaches and select the approach(es) that is the best for solving the recovery problem in terms of efficiency and optimality. Note that although the above computational frameworks can be used to solve the problem for the cases where optimal policies are not characterized, all of these become computationally burdensome with the size of the problem. Therefore, we will provide heuristic policies (that are motivated from the optimal policies) for the cases of the problem where optimal policies are not currently known. We will evaluate the performance of these heuristic policies for small instances of our problem using one of the above computational frameworks.

We will also obtain real world data and quantify the impact of optimal sequencing in comparison to other approaches. For instance, the collected data could be used to estimate input values of the recovery problem such as initial health values, rates of repair and deterioration, etc. The estimation of inputs will also provide the guidelines for operationalising the characterized policies and computational frameworks of this dissertation.

The outline of this dissertation is as follows. In the next chapter, we will focus on the recovery of physical components when there is a single repair agency and there are no dependencies between the components. In Chapter 3, we will study the recovery of physical components under dependencies. In Chapter 4, we will focus on the the recovery of physical components when there are multiple repair agencies. In Chapter 5, we will focus on the recovery of both the social and physical components. In Chapter 6, we will present various computational frameworks and heuristics for solving the cases where the optimal policies are not known and present examples generated using real-world data. In the last chapter, we will conclude this work and provide potential future directions.

2. OPTIMAL CONTROL POLICIES FOR RECOVERY OF PHYSICAL COMPONENTS AFTER DISASTERS

2.1 Introduction

In this chapter, we only focus on the recovery of physical components as it is an important factor for the recovery of social components [6]. We consider a disaster scenario where multiple physical infrastructure components suffer damage. After the disaster, the health of these components continue to deteriorate over time, unless they are being repaired. Given this setting, we consider the problem of finding the optimal sequence to repair different infrastructure components in order to maximize the number of components that are eventually returned to full health. The main focus of this chapter is towards the case when the deterioration and repair rates are fixed with time but at the end of this chapter we also focus on the case when the rates vary with time. We start by formally defining the problem for the case when the rates do not vary with time.

2.2 Problem statement

There are $N(\geq 2)$ nodes indexed by the set $\mathcal{V} = \{1, 2, \ldots, N\}$, each representing a component (depending on the context, this could be a portion of physical infrastructure in a given area, an infected computer server, a fire-affected region, etc.). There is an entity (or controller) whose objective is to repair these components. We assume that time progresses in discrete time-steps, capturing the resolution at which the entity makes decisions about which node to repair. We index the time-steps with the variable $t \in \mathbb{N} = \{0, 1, 2, \ldots\}$. The *health* of each node $\mathrm{j} \in \mathcal{V}$ at time-step t is denoted by $v_t^{\mathrm{j}} \in [0, 1]$.[1] The initial health of each node j is denoted by $v_0^{\mathrm{j}} \in (0, 1)$. The aggregate state vector for the entire system at each time-step $t \in \mathbb{N}$ is given by $v_t = \{v_t^{\mathrm{j}}\}$, where $\mathrm{j} \in \{1, \ldots, N\}$. The weight of node $\mathrm{j} \in \{1, \ldots, N\}$ is denoted by $w^{\mathrm{j}} \in \mathbb{R}_{\geq 0}$, and represents its relative importance. For example, it can represent the number of households that are dependent on an infrastructure component, the number of files that are stored in a computer server, or the population of a fire-affected region.

[1]↑As mentioned earlier, the health values/states of the nodes are bounded by two thresholds: *permanent repair* and *permanent failure*. Therefore, by scaling the health values and repair/deterioration rates, we can take the range of health values to be the interval [0,1] without loss of generality.

Definition 2.2.1. *We say that node* j ***permanently fails*** *at time-step* t *if* $v_t^{\mathrm{j}} = 0$ *and* $v_{t-1}^{\mathrm{j}} > 0$. *We say that node* j *is* ***permanently repaired*** *at time-step* t *if* $v_t^{\mathrm{j}} = 1$ *and* $v_{t-1}^{\mathrm{j}} < 1$. *If a node permanently fails or is permanently repaired, then its health does not change thereafter.*

At each time-step t, the entity can target exactly one node to repair during that time-step.[2] Thus, the control action taken by the entity at time-step t is denoted by $u_t \in \mathcal{V}$. If node j is being repaired by the entity at time-step t and it has not permanently failed or repaired, its health increases by a quantity $\Delta_{inc}^{\mathrm{j}} \in [0, 1]$ (up to a maximum health of 1). If node j is not being repaired by the entity at time-step t and it has not permanently failed or repaired, its health decreases by a fixed quantity $\Delta_{dec}^{\mathrm{j}} \in [0, 1]$ (down to a minimum health of 0). Thus, $\{\Delta_{inc}^{\mathrm{j}}\}$ and $\{\Delta_{dec}^{\mathrm{j}}\}$ represent the vectors of the rates of repair and deterioration, respectively. For each node j, the dynamics of the control problem are given by

$$v_{t+1}^{\mathrm{j}} = \begin{cases} 1 & \text{if } v_t^{\mathrm{j}} = 1, \\ 0 & \text{if } v_t^{\mathrm{j}} = 0, \\ \min(1, v_t^{\mathrm{j}} + \Delta_{inc}^{\mathrm{j}}) & \text{if } u_t = \mathrm{j} \text{ and } v_t^{\mathrm{j}} \in (0, 1), \\ \max(0, v_t^{\mathrm{j}} - \Delta_{dec}^{\mathrm{j}}) & \text{if } u_t \neq \mathrm{j} \text{ and } v_t^{\mathrm{j}} \in (0, 1). \end{cases} \tag{2.1}$$

Definition 2.2.2. *For any given initial state values* $v_0 = \{v_0^{\mathrm{j}}\}$, *weights* $w = \{w^{\mathrm{j}}\}$, *and control sequence* $A = \{u_0, u_1, \ldots\}$, *let* $\mathcal{M}(v_0, A)$ *be the set of nodes that are permanently repaired through that sequence. That is,* $\mathcal{M}(v_0, A) = \{\mathrm{j} \in \mathcal{V} \mid \exists\, t \geq 0 \text{ s.t. } v_t^{\mathrm{j}} = 1\}$. *We define the* ***reward*** $J(v_0, w, A)$ *as the sum of the weights of the nodes in set* $\mathcal{M}(v_0, A)$. *Mathematically,* $J(v_0, w, A) = \sum_{\mathrm{j} \in \mathcal{M}(v_0, A)} w^{\mathrm{j}}$.

Based on the dynamics (2.1) and the reward definition given above, we study the following problem in this chapter.

Problem 1. *Given a set* $\mathcal{V}$ *of* N *nodes with initial health values* $v_0 = \{v_0^{\mathrm{j}}\}$, *weights* $w = \{w^{\mathrm{j}}\}$, *repair rates* $\{\Delta_{inc}^{\mathrm{j}}\}$, *and deterioration rates* $\{\Delta_{dec}^{\mathrm{j}}\}$, *find a control sequence* $A = \{u_0, u_1, \ldots\}$ *that maximizes the reward* $J(v_0, w, A)$.

[2] ↑We leave an investigation of the case where the entity can simultaneously target multiple nodes for future work.

Before presenting our analysis of the problem, we introduce the concept of a **jump**.

Definition 2.2.3. *The entity is said to have jumped at some time-step t if it starts targeting a different node before permanently repairing the node it targeted in the last time-step. That is, if $u_{t-1} = \mathrm{j}$, $v_t^{\mathrm{j}} < 1$ and $u_t \neq \mathrm{j}$ then the entity is said to have jumped at time-step t. A control sequence that does not contain any jumps is said to be a* ***non-jumping sequence****.*

We will generate sequences by providing feedback policies $\mu : [0,1]^N \to \mathcal{V}$, such that $u_t = \mu(v_t), \forall t$. Therefore, we will use the above terminology and definitions for policies as well. For example, a non-jumping policy is one that generates a non-jumping sequence.

We will split our analysis of the optimal control policy for Problem 1 into two parts: one for the case where $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$, and the other for the remaining cases.

2.3 Optimal control policies for $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}$, $\forall \mathrm{j} \in \{1, \ldots, N\}$

We first show that non-jumping policies are optimal when $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$. Subsequently, we show that when the repair rates are lower bounded by a positive real number, the optimal control sequence can be found via an algorithm whose computational time complexity is polynomial in the number of nodes (but exponential in a certain function of the repair and deterioration rates). We also explicitly characterize optimal and near-optimal policies under various conditions.

2.3.1 Optimality of non-jumping policies

First, we analyze properties of sequences containing at most one jump and later generalize to sequences containing an arbitrary number of jumps. We start with the following result.

Lemma 1. *Let there be $N (\geq 2)$ nodes, and suppose $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$. Consider the two control sequences A and B targeting N nodes as shown in Figures 2.1 and 2.2, respectively. Suppose sequence A permanently repairs all nodes and contains exactly one jump, where the entity partially repairs node i_1 before moving to node i_2 at time-step $\bar{t}_1^A$. Sequence A then permanently repairs nodes $\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k$, before returning to node i_1 and permanently repairing it. Sequence B is a non-jumping sequence that targets nodes in the* *order* $\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k, \mathrm{i}_1, \mathrm{i}_{k+1}, \ldots, \mathrm{i}_N$. *Let* t_j^A *(resp.* t_j^B*) be the number of time-steps taken to permanently repair node* i_j *in sequence* A *(resp. sequence* B*). Then, sequence* B *also permanently repairs all nodes, and furthermore, the following holds true:*

$$t_\mathrm{j}^A \geq t_\mathrm{j}^B + \left(2^{\mathrm{j}-2}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{2, \ldots, k\}, \tag{2.2}$$

$$t_1^A \geq t_1^B + (2^{k-1} - 2)\bar{t}_1^A, \tag{2.3}$$

$$t_\mathrm{j}^A \geq t_\mathrm{j}^B + \left(2^{\mathrm{j}-1} - 2^{\mathrm{j}-k}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{k+1, \ldots, N\}. \tag{2.4}$$

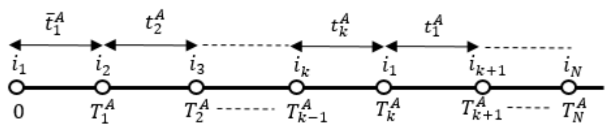

Figure 2.1. Sequence A with a single jump.

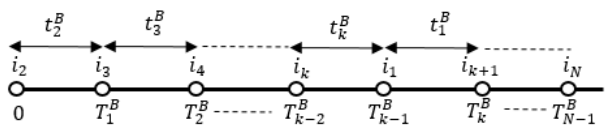

Figure 2.2. Non-jumping sequence B.

Proof. Let $T_1^A, T_2^A, \ldots, T_N^A$ be the time-steps at which sequence A starts targeting a new node, as shown in Fig. 2.1. Similarly, let $T_1^B, T_2^B, \ldots, T_{N-1}^B$ be the time-steps at which sequence B starts targeting a new node, as shown in Fig. 2.2.

We start by first proving condition (2.2), using mathematical induction on the index of nodes in the sequence. Consider $\mathrm{j} = 2$. At time-step T_1^A in sequence A, the health of node i_2 is given by

$$v_{T_1^A}^{\mathrm{i}_2} = v_0^{\mathrm{i}_2} - \Delta_{dec}^{\mathrm{i}_2} \bar{t}_1^A.$$

Let $\lceil x \rceil$ be the least integer greater than or equal to x. We now calculate t_2^A as

$$\begin{aligned} t_2^A = \left\lceil \frac{1 - v_{T_1^A}^{\mathrm{i}_2}}{\Delta_{inc}^{\mathrm{i}_2}} \right\rceil &= \left\lceil \frac{1 - v_0^{\mathrm{i}_2} + \Delta_{dec}^{\mathrm{i}_2} \bar{t}_1^A}{\Delta_{inc}^{\mathrm{i}_2}} \right\rceil \\ &\geq \left\lceil \frac{1 - v_0^{\mathrm{i}_2}}{\Delta_{inc}^{\mathrm{i}_2}} \right\rceil + \bar{t}_1^A = t_2^B + \bar{t}_1^A, \end{aligned}$$

which satisfies condition (2.2). Suppose that condition (2.2) holds for r nodes where $r < k$. If sequence A permanently repairs nodes $\mathrm{i}_2, \ldots, \mathrm{i}_r$, then so does sequence B (as each node is reached at an earlier time-step in B than in A, by the above inductive assumption). We now compute $v_{T_r^A}^{\mathrm{i}_{r+1}}$:

$$\begin{aligned} v_{T_r^A}^{\mathrm{i}_{r+1}} &= v_0^{\mathrm{i}_{r+1}} - \Delta_{dec}^{\mathrm{i}_{r+1}} T_r^A \\ &= v_0^{\mathrm{i}_{r+1}} - \Delta_{dec}^{\mathrm{i}_{r+1}} \left(\bar{t}_1^A + t_2^A + \ldots + t_r^A \right). \end{aligned}$$

Thus,

$$\begin{aligned} t_{r+1}^A &= \left\lceil \frac{1 - v_0^{\mathrm{i}_{r+1}} + \Delta_{dec}^{\mathrm{i}_{r+1}} \left(\bar{t}_1^A + t_2^A + \ldots + t_r^A \right)}{\Delta_{inc}^{\mathrm{i}_{r+1}}} \right\rceil \\ &\geq \left\lceil \frac{1 - v_0^{\mathrm{i}_{r+1}} + \Delta_{dec}^{\mathrm{i}_{r+1}} \left(t_2^B + \ldots + t_r^B \right)}{\Delta_{inc}^{\mathrm{i}_{r+1}}} + \frac{\Delta_{dec}^{\mathrm{i}_{r+1}} \left(\bar{t}_1^A + \bar{t}_1^A + 2\bar{t}_1^A + \ldots + 2^{r-2}\bar{t}_1^A \right)}{\Delta_{inc}^{\mathrm{i}_{r+1}}} \right\rceil \\ &\geq t_{r+1}^B + 2^{r-1}\bar{t}_1^A. \end{aligned}$$

So, we have shown condition (2.2) by induction. We now prove condition (2.3). Node i_1 is targeted again in sequence A at time-step T_k^A, at which point its health is

$$v_{T_k^A}^{\mathrm{i}_1} = v_0^{\mathrm{i}_1} + \bar{t}_1^A \Delta_{inc}^{\mathrm{i}_1} - \left(t_2^A + \ldots + t_k^A \right) \Delta_{dec}^{\mathrm{i}_1}. \tag{2.5}$$

Thus, the number of time-steps taken to permanently repair node i_1 in sequence A (the second time it is targeted in the sequence) is

$$t_1^A = \left\lceil \frac{1 - v_0^{\mathrm{i}_1} - \bar{t}_1^A \Delta_{inc}^{\mathrm{i}_1} + \left(t_2^A + \ldots + t_k^A \right) \Delta_{dec}^{\mathrm{i}_1}}{\Delta_{inc}^{\mathrm{i}_1}} \right\rceil. \tag{2.6}$$

Note that

$$t_2^A + \ldots + t_k^A \geq t_2^B + \ldots + t_k^B + \left(2^{k-1} - 1\right) \bar{t}_1^A, \tag{2.7}$$

by condition (2.2). Furthermore, in sequence B, the health of node i_1 at the time it is targeted is given by

$$v_{T_{k-1}^B}^{\mathrm{i}_1} = v_0^{\mathrm{i}_1} - \Delta_{dec}^{\mathrm{i}_1}(t_2^B + t_3^B + \cdots + t_k^B).$$

Comparing this to the health of node i_1 in sequence A at the time it is targeted (given by (2.5)), and using (2.7), we see that since i_1 is assumed to not have failed in sequence A, it will not have failed in sequence B as well. Thus, the number of time-steps required to permanently repair i_1 in sequence B is

$$t_1^B = \left\lceil \frac{1 - v_0^{\mathrm{i}_1} + \Delta_{dec}^{\mathrm{i}_1}(t_2^B + t_3^B + \cdots + t_k^B)}{\Delta_{inc}^{\mathrm{i}_1}} \right\rceil. \tag{2.8}$$

Thus, using (2.6), (2.7) and (2.8), we have

$$t_1^A \geq t_1^B + \left(2^{k-1} - 2\right) \bar{t}_1^A,$$

proving condition (2.3).

We now prove condition (2.4) via mathematical induction. Consider node i_{k+1}. At the time-step when this node is targeted in sequence A, its health is

$$\begin{aligned} v_{T_{k+1}^A}^{\mathrm{i}_{k+1}} &= v_0^{\mathrm{i}_{k+1}} - \Delta_{dec}^{\mathrm{i}_{k+1}} T_{k+1}^A \\ &= v_0^{\mathrm{i}_{k+1}} - \Delta_{dec}^{\mathrm{i}_{k+1}} \left(\bar{t}_1^A + t_2^A + \ldots + t_k^A + t_1^A\right). \end{aligned}$$

If node i_{k+1} has not failed at this point in sequence A, it has also not failed when it is reached in sequence B (as all nodes prior to i_{k+1} are permanently repaired faster in sequence B than in sequence A, as shown above). Thus, using (2.2) and (2.3),

$$\begin{aligned}
t_{k+1}^{A} = & \left\lceil \frac{1 - v_0^{\mathrm{i}_{k+1}} + \Delta_{dec}^{\mathrm{i}_{k+1}} \left(\bar{t}_1^A + t_2^A + \ldots + t_k^A + t_1^A \right)}{\Delta_{inc}^{\mathrm{i}_{k+1}}} \right\rceil \\
& \geq \left\lceil \frac{1 - v_0^{\mathrm{i}_{k+1}} + \Delta_{dec}^{\mathrm{i}_{k+1}} \left(t_2^B + \ldots + t_k^B + t_1^B \right)}{\Delta_{inc}^{\mathrm{i}_{k+1}}} \right. \\
& \left. + \frac{\Delta_{dec}^{\mathrm{i}_{k+1}} \left(\bar{t}_1^A + \bar{t}_1^A + 2\bar{t}_1^A + \ldots + 2^{k-2}\bar{t}_1^A + \left(2^{k-1} - 2\right) \bar{t}_1^A \right)}{\Delta_{inc}^{\mathrm{i}_{k+1}}} \right\rceil \\
& \geq t_{k+1}^B + \left(2^k - 2\right) \bar{t}_1^A,
\end{aligned}$$

which satisfies condition (2.4). Suppose condition (2.4) holds for $\mathrm{j} \in \{k+1, \ldots, r\}$, where $r < N$. Consider node i_{r+1}. Then, a similar inductive argument can be used to show that

$$t_{r+1}^A \geq t_{r+1}^B + \left(2^r - 2^{r+1-k}\right) \bar{t}_1^A.$$

This proves the third claim. □

The above result considered sequences containing exactly one jump. This leads us to the following key result pertaining to the optimal control policy when $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$.

Theorem 2.3.1. *Let there be $N (\geq 2)$ nodes, and suppose $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$. If there is a sequence A with one or more jumps that permanently repairs all the nodes of a set $\mathcal{Z} \subseteq \mathcal{V}$, then there exists a non-jumping sequence that permanently repairs all the nodes in set $\mathcal{Z}$. Thus, non-jumping sequences are optimal when $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$.*

Proof. We prove that given a sequence with an arbitrary number of jumps that permanently repairs a set $\mathcal{Z} \subseteq \mathcal{V}$, one can come up with a sequence that permanently repairs all the nodes in $\mathcal{Z}$, but has at least one fewer jump than the given sequence (and permanently repairs in less time than the given sequence). One can iteratively apply this result on the obtained sequences to eventually yield a non-jumping sequence that permanently repairs all the nodes in $\mathcal{Z}$ in less time as compared to the given sequence; thus, the reward obtained by the non-jumping sequence will be equal to the reward obtained by the given sequence consisting of an arbitrary number of jumps.

Consider the given sequence A that permanently repairs a set $\mathcal{Z}$ of nodes and suppose A contains one or more jumps. Remove all the nodes targeted by A that are not permanently repaired. This gives a new sequence B that only targets nodes in the set $\mathcal{Z}$. If B does not contain any jumps, then we are done. Otherwise, consider the *last* jump in B, and suppose it occurs at time-step T. Denote the portion of the sequence B from time-step $T-1$ onwards by C, and denote the portion of the sequence B from time-step 0 to time-step $T-2$ by $\overline{C}$. Now, note that sequence C contains exactly one jump. Thus, by Lemma 1, we can replace sequence C with another sequence D that contains no jumps and permanently repairs all nodes that are permanently repaired in C in less time. Create a new sequence $\overline{B}$ by concatenating the sequence $\overline{C}$ and the sequence D. Thus, $\overline{B}$ is a sequence with one fewer jump than B, and permanently repairs all the nodes in set $\mathcal{Z}$ and does so in less time. The first part of the result thus follows. The fact that non-jumping policies are optimal is then immediately obtained by considering A to be any optimal policy. □

2.3.2 Optimal sequencing when the repair rates are lower bounded by a positive real number

We now show that the optimal sequence can be found in polynomial time under certain conditions on the repair and deterioration rates. We start with the following result.

Lemma 2. *Let there be $N(\geq 2)$ nodes, and suppose $\Delta^{\text{j}}_{\text{dec}} \geq \Delta^{\text{j}}_{\text{inc}}, \forall \text{j} \in \{1, \ldots, N\}$. Define $n = \min_{\text{j}} \left\lfloor \frac{\Delta^{\text{j}}_{dec}}{\Delta^{\text{j}}_{\text{inc}}} \right\rfloor$, where $\left\lfloor \frac{\Delta^{\text{j}}_{dec}}{\Delta^{\text{j}}_{inc}} \right\rfloor$ is the largest integer less than or equal to $\frac{\Delta^{\text{j}}_{dec}}{\Delta^{\text{j}}_{inc}}$. Then, the number of nodes that can be permanently repaired by a non-jumping sequence is upper bounded by*

$$L = \min \left\{ N, \left\lfloor \log_{(1+n)} \left(\frac{n}{\min_{\text{j}} \{\Delta^{\text{j}}_{dec}\}} + 1 \right) + 1 \right\rfloor \right\}. \tag{2.9}$$

Proof. Theorem 2.3.1 showed that non-jumping sequences are optimal when $\Delta^{\text{j}}_{dec} \geq \Delta^{\text{j}}_{inc}, \forall \text{j}$. Next, note from the definition of n that for each time-step that a node j deteriorates (where its health decreases by Δ^{j}_{dec}), it will take at least n time-steps of repair to compensate for that deterioration. We can now bound the number of nodes that are permanently repaired by a non-jumping sequence as follows. The number of time-steps taken to permanently repair the first node is at least equal to 1. Then, the number of time-steps taken to permanently repair the second node in the sequence is at least equal to $1+n$ (for the second node in the sequence, it takes at least n time-steps to repair the health that is lost due to deterioration and it takes at least one additional time-step to repair the difference between the initial health and the permanent repair state). The number of time-steps taken to permanently repair the third node in the sequence is at least equal to $1+n(1+1+n)$, i.e., n times the number of time-steps spent on repairing the previous nodes in order to make up for the deterioration faced in those time-steps, and at least one additional time-step to permanently repair. By induction, it can be easily shown that the number of time-steps taken to permanently repair the $\mathrm{i_j}$th node in the sequence is at least equal to $(1+n)^{\mathrm{j}-1}$. Suppose there exists a non-jumping sequence that permanently repairs x nodes. Then, node i_x in that sequence should have positive health by the time the first $x-1$ nodes are permanently repaired. The largest time-step at which there is a node with positive health is upper bounded by $\max_{\mathrm{j}}\left\{\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}}\right\}$. Then, $(1+n)^0+(1+n)^1+\ldots+(1+n)^{x-2}=\frac{(1+n)^{x-1}-1}{n}<\max_{\mathrm{j}}\left\{\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}}\right\}$. Note that $\max_{\mathrm{j}}\left\{\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}}\right\}<\max_{\mathrm{j}}\left\{\frac{1}{\Delta_{dec}^{\mathrm{j}}}\right\}=\frac{1}{\min_{\mathrm{j}}\{\Delta_{dec}^{\mathrm{j}}\}}$ because $v_0^{\mathrm{j}}<1, \forall \mathrm{j}$. Thus, $x<\log_{(1+n)}\left(\frac{n}{\min_{\mathrm{j}}\{\Delta_{dec}^{\mathrm{j}}\}}+1\right)+1$. □

We now show that if L in (2.9) is upper bounded by a constant regardless of the value of N (which will happen when the ratio $\frac{n}{\min_{\mathrm{j}}\{\Delta_{dec}^{\mathrm{j}}\}}$ is upper bounded by a positive real number regardless of the value of N), the optimal sequencing policy can be computed in time that is polynomial in the number of nodes.

Theorem 2.3.2. *Let there be $N(\geq 2)$ nodes, and suppose $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j} \in \{1,\ldots,N\}$. Define $n=\min_{\mathrm{j}}\left\lfloor\frac{\Delta_{dec}^{\mathrm{j}}}{\Delta_{inc}^{\mathrm{j}}}\right\rfloor$. If $\frac{n}{\min_{\mathrm{j}}\{\Delta_{dec}^{\mathrm{j}}\}}$ is upper bounded by a positive real number regardless of the value of N, then the complexity of finding the optimal sequence is $O(LN^L)$, where L is given by (2.9).*

Proof. Note that computational time complexity typically refers to the time (or number of operations) asymptotically required by an algorithm as the input size increases. Thus, if $\frac{n}{\min_{\mathrm{j}}\{\Delta_{dec}^{\mathrm{j}}\}}$ is upper bounded by a positive real number regardless of number of nodes (N), then the quantity L in (2.9) is upper-bounded by a constant for all N. Under this condition, we enumerate all the non-jumping sequences of length L that need to be compared to find the optimal sequence. At the start of the first time-step, there are N choices of nodes that can be targeted; after permanently repairing the first node, there are $N-1$ choices of nodes that can be permanently repaired, and so on. Since the maximum number of nodes that can be permanently repaired is upper bounded by L, the number of sequences that need to be compared to find the optimal sequence is $O\left(N^L\right)$. Denote the set of non-jumping sequences of length L by $\mathcal{R}$. We compute the weighted number of nodes that are permanently repaired by the sequences in set $\mathcal{R}$ through simulation. Since a sequence can permanently repair at most L nodes, there would be $O(L)$ operations in the simulation while computing the weighted number of permanently repaired nodes. Thus, the complexity of computing the optimal sequence is $O\left(LN^L\right)$. Therefore, the complexity of finding the optimal sequence is polynomial in the number of nodes. $\square$

Remark 1. *Note that* $\frac{n}{\min_{\mathrm{j}}\{\Delta^{\mathrm{j}}_{dec}\}} = \frac{n}{\min_{\mathrm{j}}\{n_{\mathrm{j}}\Delta^{\mathrm{j}}_{inc}\}}$, *where* $n_{\mathrm{j}} \triangleq \frac{\Delta^{\mathrm{j}}_{dec}}{\Delta^{\mathrm{j}}_{inc}} \geq 1$. *By definition,* $n_{\mathrm{j}} \geq n, \forall \mathrm{j}$. *Thus,* $\frac{n}{\min_{\mathrm{j}}\{n_{\mathrm{j}}\Delta^{\mathrm{j}}_{inc}\}} \leq \frac{n}{\min_{\mathrm{j}}\{n\Delta^{\mathrm{j}}_{inc}\}} = \frac{1}{\min_{\mathrm{j}}\{\Delta^{\mathrm{j}}_{inc}\}}$. *Therefore, a sufficient condition for* $\frac{n}{\min_{\mathrm{j}}\{\Delta^{\mathrm{j}}_{dec}\}}$ *to be upper bounded by a positive real number is that* $\min_{\mathrm{j}}\{\Delta^{\mathrm{j}}_{inc}\}$ *be lower bounded by a positive real number (independent of N). Thus, the complexity of finding the optimal sequence is polynomial in the number of nodes if the repair rates are lower bounded by a positive real number regardless of the value of N, and for all* $\mathrm{j} \in \{1, \ldots, N\}$, $\Delta^{\mathrm{j}}_{\mathrm{dec}} \geq \Delta^{\mathrm{j}}_{\mathrm{inc}}$.

Example 1. *Suppose there are* $N(\geq 2)$ *nodes and for each node* j, $\Delta^{\mathrm{j}}_{\mathrm{dec}} = \Delta_{\mathrm{dec}} = 0.1$ *and* $\Delta^{\mathrm{j}}_{\mathrm{inc}} = \Delta_{\mathrm{inc}} = 0.1$. *Then,* $L = \min\{N, \lfloor \log_2(11) + 1 \rfloor\} = \min\{N, 4\}$. *Thus,* $L \leq 4$ *and therefore the complexity of finding the optimal sequence is* $O(4N^4) = O(N^4)$.

While Theorem 2.3.2 and Remark 1 establish that the optimal sequence can be found in polynomial-time (specifically, $O(LN^L)$) if the repair rates are bounded away from zero and $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}$ $\forall \mathrm{j} \in \{1, 2, \ldots, N\}$, the exponent L can be large if the repair rates are small. In the next section, we focus on instances of Problem 1 where the weights and the rates of repair and deterioration are homogeneous. For such instances, we show that the optimal policy can be explicitly characterized, regardless of the bound on the repair rates.

2.3.3 An optimal policy for homogeneous rates and weights

We now consider a special case of Problem 1 when the deterioration and repair rates as well as the weights are homogeneous across all the nodes, i.e., $\Delta_{dec}^{\mathrm{j}} = \Delta_{dec}, \forall \mathrm{j}$, $\Delta_{inc}^{\mathrm{j}} = \Delta_{inc}, \forall \mathrm{j}$, and $w^{\mathrm{j}} = \overline{w}, \forall \mathrm{j}$. Theorem 2.3.1 showed that non-jumping policies are optimal for general (heterogeneous) rates and weights, and when the rates of deterioration are larger than the rates of repair, and thus that result holds for homogeneous rates and weights as well. For the homogeneous case, we characterize the optimal non-jumping policy in the set of all non-jumping policies. The following lemma will be useful for a later result.

Lemma 3. *Let there be $N(\geq 2)$ nodes, and suppose for all $\mathrm{j} \in \{1, \dots, N\}$, $\Delta_{\mathrm{dec}}^{\mathrm{j}} = \Delta_{\mathrm{dec}}$ and $\Delta_{inc}^{\mathrm{j}} = \Delta_{inc}$. Consider a non-jumping sequence that permanently repairs all of the nodes. Under that sequence, suppose the order in which the nodes are targeted is $\mathrm{i}_1, \dots, \mathrm{i}_N$ and that t_{j} is the number of time-steps the entity takes to permanently repair node i_{j}. Define $Z_1 = v_0^{\mathrm{i}_1}$ and $Z_k = v_0^{\mathrm{i}_k} - \Delta_{\mathrm{dec}} \sum_{\mathrm{j}=2}^{k} \left\lceil \frac{1-Z_{\mathrm{j}-1}}{\Delta_{\mathrm{inc}}} \right\rceil$ for $k \in \{2, \dots, N\}$, where $\left\lceil \frac{1-Z_{\mathrm{j}-1}}{\Delta_{\mathrm{inc}}} \right\rceil$ is the smallest integer greater than or equal to $\frac{1-Z_{\mathrm{j}-1}}{\Delta_{\mathrm{inc}}}$. Then, the following holds true:*

$$\sum_{p=1}^{N-1} t_p = \sum_{\mathrm{j}=2}^{N} \left\lceil \frac{1 - Z_{\mathrm{j}-1}}{\Delta_{\mathrm{inc}}} \right\rceil . \tag{2.10}$$

The proof follows immediately from mathematical induction by noting that Z_{j} is the health of node i_{j} when it is reached in the sequence, and thus $t_{\mathrm{j}} = \left\lceil \frac{1-Z_{\mathrm{j}}}{\Delta_{\mathrm{inc}}} \right\rceil, \forall \mathrm{j} \in \{1, \dots, N\}$.

The next result presents necessary and sufficient conditions for a non-jumping sequence to permanently repair all nodes.

Corollary 1. *Let there be $N(\geq 2)$ nodes, and suppose for all $\mathrm{j} \in \{1, \dots, N\}$, $\Delta_{\mathrm{inc}}^{\mathrm{j}} = \Delta_{\mathrm{inc}}$ and $\Delta_{\mathrm{dec}}^{\mathrm{j}} = \Delta_{\mathrm{dec}}$. Consider a non-jumping sequence, where the order in which the nodes are targeted is $\mathrm{i}_1, \dots, \mathrm{i}_N$. Define $Z_1 = v_0^{\mathrm{i}_1}$ and $Z_k = v_0^{\mathrm{i}_k} - \Delta_{\mathrm{dec}} \sum_{\mathrm{j}=2}^{k} \left\lceil \frac{1-Z_{\mathrm{j}-1}}{\Delta_{\mathrm{inc}}} \right\rceil$ for $k \in \{2, \dots, N\}$. Then the following conditions are necessary and sufficient for all the nodes to eventually get permanently repaired:*

$$Z_k > 0 \quad \forall k \in \{1, \dots, N\}. \tag{2.11}$$

The proof follows trivially from the definition of Z_k, namely that Z_k is the health of node i_k at the time-step when all nodes before i_k in the sequence under consideration are permanently repaired and the entity starts repairing node i_k.

Based on Corollary 1, we now provide the optimal policy that permanently repairs the maximum number of nodes, under certain conditions on the initial health values, and rates of repair and deterioration.[3]

Theorem 2.3.3. *Let there be $N(\geq 2)$ nodes, and suppose for all $\mathrm{j} \in \{1, \ldots, N\}$, we have $\Delta^{\mathrm{j}}_{\mathrm{inc}} = \Delta_{inc}$, $\Delta^{\mathrm{j}}_{\mathrm{dec}} = \Delta_{dec}$, $w^{\mathrm{j}} = \overline{w}$ and $\Delta_{dec} \geq \Delta_{inc}$. Suppose $\Delta_{dec} = n\Delta_{\mathrm{inc}}$, where n is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v^{\mathrm{j}}_0 = m_{\mathrm{j}}\Delta_{\mathrm{inc}}$. Then, the non-jumping sequence that targets nodes in decreasing order of their initial health is optimal for Problem 1.*

Proof. Consider any optimal (non-jumping) sequence A, and let x be the number of nodes that are permanently repaired by that sequence. Denote this set of $x(\leq N)$ nodes as $\mathcal{Z}$. Let $\mathrm{i}_1, \ldots, \mathrm{i}_x$ be the order in which the sequence A permanently repairs the x nodes. The conditions $\Delta_{dec} = n\Delta_{\mathrm{inc}}$ and $1 - v^{\mathrm{j}}_0 = m_{\mathrm{j}}\Delta_{\mathrm{inc}}, \ \forall \mathrm{j} \in \{1, \ldots, N\}$ ensure that no node gets permanently repaired partway through a time-step. Thus, the necessary and sufficient conditions to permanently repair x nodes if a non-jumping sequence A targets the nodes in the order $\mathrm{i}_1, \ldots, \mathrm{i}_x$ are given by $Z_k > 0, \forall k \in \{1, \ldots, x\}$ from Corollary 1, where $Z_1 = v^{\mathrm{i}_1}_0$ and $Z_k = v^{\mathrm{i}_k}_0 - n\sum_{\mathrm{j}=2}^{k}(1 - Z_{\mathrm{j}-1})$ for $k \in \{2, \ldots, x\}$. Note that the ceiling functions in the definition of Z_k in Corollary 1 are dropped due to the conditions on the health values and the rates of repair and deterioration.

As $\Delta_{dec} = n\Delta_{\mathrm{inc}}$, we can expand these conditions as

$$v^{\mathrm{i}_k}_0 - n\sum_{\mathrm{j}=2}^{k}\Big((1 - v^{\mathrm{i}_{\mathrm{j}-1}}_0)(1+n)^{k-\mathrm{j}}\Big) > 0 \quad \forall k \in \{2, \ldots, x\}. \tag{2.12}$$

The conditions (2.12) can be alternatively written as

$$v^{\mathrm{i}_1}_0 n + v^{\mathrm{i}_2}_0 > n, \tag{2.13}$$

[3] Note that when weights are homogeneous across all the nodes, Problem 1 is equivalent to maximizing the number of nodes that are permanently repaired.

$$v_0^{\mathrm{i}_1} n(1+n) + v_0^{\mathrm{i}_2} n + v_0^{\mathrm{i}_3} > n(1+n) + n, \tag{2.14}$$

$$\vdots$$

$$v_0^{\mathrm{i}_1} n\,(1+n)^{x-2} + v_0^{\mathrm{i}_2} n(1+n)^{x-3} + \ldots + v_0^{\mathrm{i}_{x-1}} n + v_0^{\mathrm{i}_x} >$$
$$n\,(1+n)^{x-2} + n(1+n)^{x-3} + \ldots + n. \tag{2.15}$$

The right-hand side (RHS) of the above conditions do not depend on the sequence in which the nodes are permanently repaired. Consider the left-hand side (LHS) of the above conditions. In condition (2.13), the LHS would be the largest when node i_1 has the largest initial health (as coefficients corresponding to $v_0^{\mathrm{i}_1}$ and $v_0^{\mathrm{i}_2}$ are n and 1, respectively). In condition (2.14), the LHS would be the largest when node i_1 has the largest initial health and node i_2 has the second largest initial health (as coefficients corresponding to $v_0^{\mathrm{i}_1}, v_0^{\mathrm{i}_2}, v_0^{\mathrm{i}_3}$ are $n(1+n), n, 1$, respectively). Proceeding in this manner until the last condition (equation (2.15)), we see that the LHS would be largest when i_1 is the node with largest initial health, i_2 is the node with the second largest initial health and so on. Thus, if we define a non-jumping sequence B that targets the nodes of set $\mathcal{Z}$ in decreasing order of their initial health values, it would also permanently repair x nodes and hence will be optimal (since it permanently repairs the same number of nodes as the optimal sequence A). Consider another non-jumping sequence C that targets the top x nodes with the largest initial health values from the N nodes. Then, the sequence C would also permanently repair x nodes. This is because each node in sequence C has a higher initial health value (or at least equal) to the corresponding node in sequence B and thus sequence C satisfies the conditions (2.13)-(2.15). Thus, the policy of targeting the nodes in decreasing order of their initial health values would also permanently repair x nodes, and hence is optimal. □

Remark 2. *Theorem 2.3.1 shows that non-jumping policies are optimal when $\Delta_{\mathrm{dec}} \geq \Delta_{\mathrm{inc}}$. Furthermore, Theorem 2.3.3 shows that under certain conditions on the initial health values, repair/deterioration rates and weights, repairing the nodes in decreasing order of their initial health is optimal.* ***Equivalently, under the conditions given in these theorems, the***

optimal sequence is a feedback policy that targets the healthiest node at each time-step.

Example 2. *We now provide an example to illustrate the policy of targeting the healthiest node at each time-step. Consider $\Delta_{\text{dec}} = \Delta_{\text{inc}} = 0.2$, and three nodes having equal weights with initial health values $v_0^1 = 0.8, v_0^2 = 0.6$ and $v_0^2 = 0.4$. Figure 2.3 shows the progression of health values with time when the policy of targeting the healthiest node at each time-step is followed. Since node 1 has the largest initial health value, it is targeted in the first time-step and permanently repaired in that time-step. After permanently repairing node 1, entity starts targeting node 2 in the second time-step and permanently repairs it in three time-steps. Note that node 3 permanently fails after the first two time-steps and thus cannot be permanently repaired (we therefore do not show the health value of node 3 from the third time-step onwards in the figure).*

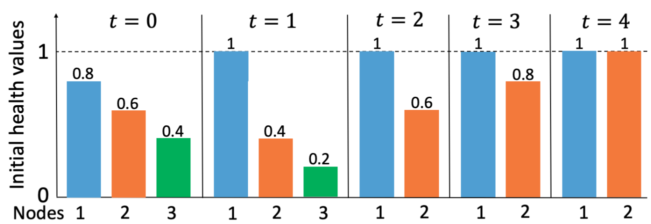

Figure 2.3. Progression of health values with time when the policy of targeting the healthiest node at each time-step is followed.

The above theorem relied on the initial health values and rates of repair/deterioration being such that each node requires an integer number of time-steps to be permanently repaired (allowing the ceiling functions in the characterization of the number of time-steps to be dropped). We now relax those conditions and present a policy that computes an approximately optimal solution. We start with the following result.

Lemma 4. *Let there be $N(\geq 2)$ nodes, and suppose for all $\mathrm{j} \in \{1, \ldots, N\}$, we have $\Delta^{\mathrm{j}}_{inc} = \Delta_{inc}$, $\Delta^{\mathrm{j}}_{dec} = \Delta_{dec}$, $w^{\mathrm{j}} = \overline{w}$ and $\Delta_{dec} \geq \Delta_{inc}$. Consider a non-jumping sequence A that permanently repairs $x(\leq N)$ nodes and let $\{\mathrm{i}_1, \ldots, \mathrm{i}_x\}$ be the order of nodes that are permanently repaired in A. Suppose there is a node h such that $v_0^h > v_0^{\mathrm{i}_1}$. Consider a* *non-jumping sequence B that first targets node h and then targets the remaining nodes of sequence A except node i_1. Then, sequence B permanently repairs at least $x-1$ nodes.*

Proof. Note that the number of time-steps taken to permanently repair node h in sequence B is less than or equal to the number of time-steps taken to permanently repair node i_1 in sequence A because $\left\lceil \frac{1-v_0^h}{\Delta_{inc}} \right\rceil \leq \left\lceil \frac{1-v_0^{\mathrm{i}_1}}{\Delta_{inc}} \right\rceil$ (as $v_0^h > v_0^{\mathrm{i}_1}$). Suppose h is not permanently repaired in sequence A. Then, x nodes would be permanently repaired in B because the nodes of the set $\{\mathrm{i}_2, \ldots, \mathrm{i}_x\}$ would be targeted at an earlier or equal time-step in sequence B than in sequence A. Now suppose h is permanently repaired in sequence A such that $h = \mathrm{i}_\mathrm{j}$ where $\mathrm{j} \in \{2, \ldots, x\}$. Then, $x-1$ nodes would be permanently repaired in B because the nodes of the set $\{\mathrm{i}_2, \ldots, \mathrm{i}_x\} \setminus \mathrm{i}_\mathrm{j}$ would be targeted at an earlier or equal time-step in sequence B than in sequence A. Thus, the result follows. $\square$

We now present an approximately optimal policy when $\Delta_{dec} \geq \Delta_{inc}$, without the additional conditions on the initial health values and rates imposed by Theorem 2.3.3.

Theorem 2.3.4. *Let there be $N (\geq 2)$ nodes, and suppose for all $\mathrm{j} \in \{1, \ldots, N\}$, we have $\Delta_{inc}^{\mathrm{j}} = \Delta_{inc}$, $\Delta_{dec}^{\mathrm{j}} = \Delta_{dec}$, $w^{\mathrm{j}} = \overline{w}$ and $\Delta_{dec} \geq \Delta_{inc}$. Then, the non-jumping sequence that targets nodes in decreasing order of their initial health permanently repairs at least half the nodes as that by an optimal sequence.*

Proof. Consider any optimal (non-jumping) sequence A, and let x be the number of nodes that are permanently repaired by A. Let T be the first time-step in A at which the healthiest node is not targeted. Denote the portion of sequence A from time-steps 0 through $T-1$ as B and the portion of A from time-step T onwards as C. We modify sequence C by Lemma 4 to form a sequence $\overline{C}$ such that the healthiest node is targeted at time-step T in $\overline{C}$. After that, we concatenate the portions B and $\overline{C}$ to form a sequence D. Then, sequence D permanently repairs at least $x-1$ nodes by Lemma 4.

We keep repeating the above procedure so that we finally obtain a sequence E where the healthiest node is targeted at each time-step. Note that sequence E would permanently repair at least $x/2$ nodes because 1) in each iteration of this procedure we move at least one node across the given sequence and the number of nodes that are permanently repaired in the modified sequence reduces at most by one and 2) in the last iteration of this procedure where there is only one node there is no decrease in the number of nodes that are permanently repaired because a node with positive health value is always permanently repaired in a non-jumping sequence. □

The following example shows that the factor of 1/2 indicated by the above result is tight.

Example 3. *Consider $\Delta_{\text{dec}} = 0.7, \Delta_{\text{inc}} = 0.6$, and two nodes having equal weights with initial health values $v_0^1 = 0.95$ and $v_0^2 = 0.6$. If the node with the largest initial health (i.e., node 1) is first targeted, then node 2 fails by the time the entity reaches it. However, if the node with the lowest initial health (i.e., node 2) is first targeted then it is possible to permanently repair both the nodes. Thus, when the conditions of Theorem 2.3.3 are not satisfied, the policy of targeting the nodes in decreasing order of health values might not be optimal; however, this policy permanently repairs at least half the nodes as that by the optimal policy as argued in Theorem 2.3.4.*

We now give an example to show that the policy that targets the healthiest node at each time-step may not be optimal when the deterioration and repair rates are not homogeneous.

Example 4. *Consider two nodes with equal weights, $v_0^1 = 0.9, v_0^2 = 0.4$, $\Delta_{dec}^1 = 0.6, \Delta_{dec}^2 = 0.6$, $\Delta_{inc}^1 = 0.1$, and $\Delta_{inc}^2 = 0.6$. If the policy of targeting the healthiest node at each time-step is followed then node 2 fails by the time the entity reaches it. However, if we follow the non-jumping sequence that first permanently repairs the least healthy node (i.e., node 2), then it is possible to permanently repair both of the nodes.*

We also give an example to show that the policy that targets the healthiest node at each time-step may not be optimal when weights are not homogeneous.

Example 5. *Consider two nodes such that $v_0^1 = 0.5$, $v_0^2 = 0.4$, $w^1 = 1$, $w^2 = 2$, and homogeneous rates $\Delta_{\text{dec}} = \Delta_{\text{inc}} = 0.1$. If the policy of targeting the healthiest node (i.e., node 1) at each time-step is followed then a reward of 1 is obtained; however, if node 2 is first targeted and permanently repaired, then a reward of 2 is obtained.*

Characterizing the optimal policy in the above cases is an avenue for future research. However, when the weights are heterogeneous (but the rates are homogeneous) we will next show that the policy that permanently repairs the largest number of nodes also returns an approximately optimal solution to Problem 1.

2.3.4 An approximately optimal policy for heterogeneous weights and homogeneous rates

We will start with the following general result, relating the optimal sequence for Problem 1 to the optimal sequence for permanently repairing the largest number of nodes (i.e., corresponding to the case where all weights are the same).

Theorem 2.3.5. *Let there be $N(\geq 2)$ nodes with initial health values $v_0 = \{v_0^{\mathrm{j}}\}$ and weights $w = \{w^{\mathrm{j}}\}$. Let $w_{min} = \min_{\mathrm{j}} w^{\mathrm{j}}$ and $w_{max} = \max_{\mathrm{j}} w^{\mathrm{j}}$. Let A be the optimal sequence for Problem 1, and let B be a control sequence that permanently repairs the largest number of nodes. Then, $\frac{J(v_0,w,A)}{J(v_0,w,B)} \leq \frac{w_{max}}{w_{min}}$, where $J(\cdot,\cdot,\cdot)$ is the reward function defined in Definition 2.2.2.*

Proof. As defined in Definition 2.2.2, let the set of nodes permanently repaired by the sequence B be denoted by $\mathcal{M}(v_0, B) \subseteq \mathcal{V}$, and suppose it contains x nodes. Then, the reward that is obtained by the policy B satisfies $J(v_0, w, B) = \sum_{\mathrm{j} \in \mathcal{M}(v_0,B)} w^{\mathrm{j}} \geq x w_{min}$. Let the number of nodes permanently repaired by the optimal sequence A be y. Then, the reward that is obtained by the optimal sequence satisfies $J(v_0, w, A) = \sum_{\mathrm{j} \in \mathcal{M}(v_0,A)} w^{\mathrm{j}} \leq y w_{max} \leq x w_{max}$ as $y \leq x$. Thus, the ratio of the reward obtained by the optimal sequence to the reward obtained by the sequence that permanently repairs the maximum number of nodes satisfies $\frac{J(v_0,w,A)}{J(v_0,w,B)} \leq \frac{x w_{max}}{x w_{min}} = \frac{w_{max}}{w_{min}}$. ☐

The above result holds regardless of the weights and rates. We now obtain the following result pertaining to instances of Problem 1 with homogeneous rates and heterogeneous weights; the proof follows directly from Theorems 2.3.3 and 2.3.5.

Corollary 2. *Let there be $N(\geq 2)$ nodes with initial health values $v_0 = \{v_0^{\mathrm{j}}\}$ and weights $w = \{w^{\mathrm{j}}\}$. Let $w_{min} = \min_{\mathrm{j}} w^{\mathrm{j}}$ and $w_{max} = \max_{\mathrm{j}} w^{\mathrm{j}}$. For all $\mathrm{j} \in \{1, \ldots, N\}$, suppose $\Delta_{\mathrm{inc}}^{\mathrm{j}} = \Delta_{\mathrm{inc}}$, $\Delta_{\mathrm{dec}}^{\mathrm{j}} = \Delta_{\mathrm{dec}}$ and $\Delta_{\mathrm{dec}} \geq \Delta_{\mathrm{inc}}$. Suppose $\Delta_{\mathrm{dec}} = n\Delta_{\mathrm{inc}}$, where n is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v_0^{\mathrm{i_j}} = m_{\mathrm{j}} \Delta_{\mathrm{inc}}$. Then, the policy that targets the healthiest node at each time-step provides a reward that is within a factor $\frac{w_{max}}{w_{min}}$ of the optimal reward.*

2.4 Optimal control policies for $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}$

We now turn our attention to the case where $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}$ for one or more $\mathrm{j} \in \{1, \ldots, N\}$. First, we define the concept of a **modified health value**.

Definition 2.4.1. *The modified health value of a node* j *at time t is the health value minus the rate of deterioration, i.e., $v^{\mathrm{j}}_t - \Delta^{\mathrm{j}}_{dec}$.*

Note that modified health value of a node is allowed to be negative, unlike the health value. We start with the following general result.

Lemma 5. *Let there be $N(\geq 2)$ nodes. Then, for $z \in \{1, 2, \ldots, N\}$, there exists a sequence that permanently repairs z nodes only if there exists a set $\mathcal{A}(z) = \{\mathrm{i}_1, \ldots, \mathrm{i}_z\} \subseteq \mathcal{V}$ such that*

$$v^{\mathrm{i_j}}_0 > (z - \mathrm{j})\Delta^{\mathrm{i_j}}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, z\}. \tag{2.16}$$

Proof. Suppose there exists a sequence that permanently repairs z nodes. At each time-step t, let $\mathcal{C}_t$ denote the set of nodes that have not been targeted at least once by the entity prior to t. Note that $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \ldots \supseteq \mathcal{C}_{z-1}$. At $t = 0$, $|\mathcal{C}_t| = N$ where $|\mathcal{C}_t|$ denotes the cardinality of set $\mathcal{C}_t$. At time $t = 1, |\mathcal{C}_t| = N - 1$ as there are $N - 1$ nodes that have not been targeted by the entity at least once. Each node k belonging to the set $\mathcal{C}_1$ should have initial health value larger than Δ^k_{dec} to survive until $t = 1$. At $t = 2, |\mathcal{C}_t| \geq N - 2$ as there are at least $N - 2$ nodes that have not been targeted by the entity at least once. Each node k belonging to the set $\mathcal{C}_2$ should have initial health value larger than $2\Delta^k_{dec}$ to survive until $t = 2$. Repeating this argument for the next $z - 3$ time-steps proves that there must be a permutation $(\mathrm{i}_1, \ldots, \mathrm{i}_z)$ of nodes that satisfies the condition (2.16) in order for z nodes to be permanently repaired. $\square$

Note that (2.16) represents necessary conditions that need to be satisfied by *any* sequence that permanently repairs all the nodes in the set $\mathcal{A}(z)$, regardless of the rates of repair and deterioration. We now provide the following result for the case when the rates of repair are significantly larger than the rates of deterioration.

Lemma 6. *Let there be $N(\geq 2)$ nodes. Let $z \leq N$ and suppose there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\} \subseteq \mathcal{V}$ that satisfies (2.16). Suppose $\Delta_{inc}^{\mathrm{i_j}} > (z-1)\Delta_{dec}^{\mathrm{i_j}}, \forall \mathrm{j} \in \{1, \ldots, z\}$ and $\Delta_{inc}^{\mathrm{i_j}} > \sum_{k\in\{1,\ldots,z\}\setminus \mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}$, $\forall \mathrm{j} \in \{1, \ldots, z\}$. Then, the sequence that targets the node with the least modified health in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ at each time-step will permanently repair all the nodes of the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$.*

Proof. Suppose there is a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ such that (2.16) holds. There are z possible cases depending upon which node in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ has the lowest initial modified health. The first case is when node i_z has the lowest initial modified health in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$, i.e., $v_0^{\mathrm{i}_z} - \Delta_{dec}^{\mathrm{i}_z} = \min_{\mathrm{j}\in\{1,\ldots,z\}}\{v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}}\}$. After the first time-step, if node i_z does not get permanently repaired, the health values of the nodes are

$$v_1^{\mathrm{i}_z} = v_0^{\mathrm{i}_z} + \Delta_{inc}^{\mathrm{i}_z} > (z-1)\Delta_{dec}^{\mathrm{i}_z}, \tag{2.17}$$

$$v_1^{\mathrm{i_j}} = v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}} > (z-1-\mathrm{j})\Delta_{dec}^{\mathrm{i_j}}, \tag{2.18}$$

$$\forall \mathrm{j} \in \{1, \ldots, z-1\},$$

where the inequality in (2.18) comes from (2.16). Thus, there exists a permutation $(\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_z)$ $= (\mathrm{i}_z, \mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{z-1})$ that satisfies the condition (2.16) at time $t = 1$. However, if node i_z gets permanently repaired after the completion of the first time-step, then $v_1^{\mathrm{i}_z} = 1$ and the health values of nodes $\{\mathrm{i}_1, \ldots, \mathrm{i}_{z-1}\}$ are given by (2.18). Thus, there exists a permutation $(\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_{z-1}) = (\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{z-1})$ such that $\{\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_{z-1}\}$ satisfies the condition (2.16) corresponding to the set $\mathcal{A}(z-1)$ at time $t = 1$ (i.e., $v_1^{\bar{\mathrm{i}}_\mathrm{j}} > (z-1-\mathrm{j})\Delta_{dec}^{\bar{\mathrm{i}}_\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, z-1\}$) along with $v_1^{\mathrm{i}_z} = 1$. We now consider the second case, when $v_0^{\mathrm{i}_{z-1}} - \Delta_{dec}^{\mathrm{i}_{z-1}} = \min_{\mathrm{j}\in\{1,\ldots,z\}}\{v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}}\}$, i.e., node i_{z-1} has the lowest initial modified health in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$. Then, after the completion of the first time-step, if node i_{z-1} does not get permanently repaired, the health values of the nodes are given by

$$v_1^{\mathrm{i}_{z-1}} = v_0^{\mathrm{i}_{z-1}} + \Delta_{inc}^{\mathrm{i}_{z-1}} > (z-1)\Delta_{dec}^{\mathrm{i}_{z-1}}, \tag{2.19}$$

$$v_1^{\mathrm{i}_z} = v_0^{\mathrm{i}_z} - \Delta_{dec}^{\mathrm{i}_z} > v_0^{\mathrm{i}_{z-1}} - \Delta_{dec}^{\mathrm{i}_{z-1}} > 0, \tag{2.20}$$

$$v_1^{\mathrm{i_j}} = v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}} > (z-1-\mathrm{j})\Delta_{dec}^{\mathrm{i_j}}, \tag{2.21}$$

$$\forall \mathrm{j} \in \{1, \ldots, z-2\}.$$

Note that the first inequality in condition (2.20) holds as $v_0^{\mathrm{i}_{z-1}} - \Delta_{dec}^{\mathrm{i}_{z-1}} = \min_{\mathrm{j}\in\{1,\ldots,z\}}\{v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}}\}$. The second inequality in condition (2.20) holds from (2.16). Thus, the nodes of the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ satisfy (2.16), but with the indices reordered. However, if node i_{z-1} gets permanently repaired after the completion of the first time-step, then $v_1^{\mathrm{i}_{z-1}} = 1$ and the health values of nodes $\{\mathrm{i}_z, \mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{z-2}\}$ are given by (2.20) and (2.21). Thus, $z-1$ nodes satisfy the condition (2.16) corresponding to the set $\mathcal{A}(z-1)$ along with $v_1^{\mathrm{i}_{z-1}} = 1$ after the completion of first time-step. The remaining $z-2$ cases similarly follow and are therefore omitted. Thus, at any time-step, if there are $x(\leq z)$ nodes that satisfy the condition (2.16) corresponding to the set $\mathcal{A}(x)$, and $z-x$ nodes that are permanently repaired, then there will be a permutation of $y(\leq x)$ nodes that satisfy the condition (2.16) corresponding to the set $\mathcal{A}(y)$ and $z-y$ nodes that are permanently repaired at the start of the next time-step. Therefore, no node that belongs to the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ would have health becoming zero at any time. Furthermore, if a node $\mathrm{i_j}$, where $\mathrm{j} \in \{1, \ldots, z\}$, is targeted by the entity at a time-step and it does not get permanently repaired then the average health of the nodes in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ increases by at least $\frac{\Delta_{inc}^{\mathrm{i_j}} - \sum_{k\in\{1,\ldots,z\}\setminus\mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}}{z}$. Note that $\frac{\Delta_{inc}^{\mathrm{i_j}} - \sum_{k\in\{1,\ldots,z\}\setminus\mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}}{z} > 0$ as $\Delta_{inc}^{\mathrm{i_j}} > \sum_{k\in\{1,\ldots,z\}\setminus\mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}, \quad \forall \mathrm{j} \in \{1, \ldots, z\}$. So, at each time-step, either the increase in average health of the nodes in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ is positive, or a node gets permanently repaired, or both. Therefore, all the nodes of the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\}$ would eventually be permanently repaired. □

Remark 3. *Note that the conditions on the deterioration and repair rates provided in Lemma 6 are a function of the particular set of z nodes satisfying (2.16); however, a stronger, but set independent, sufficient condition for the policy given in Lemma 6 to repair all z nodes would be $\Delta_{\mathrm{inc}}^{\mathrm{i_j}} > (N-1)\Delta_{\mathrm{dec}}^{\mathrm{i_j}}, \forall \mathrm{i_j} \in \mathcal{V}$ and $\Delta_{\mathrm{inc}}^{\mathrm{i_j}} > \sum_{\mathrm{i}_k\in\mathcal{V}\setminus\mathrm{i_j}} \Delta_{\mathrm{dec}}^{\mathrm{i}_k}, \quad \forall \mathrm{i_j} \in \mathcal{V}$ because for each set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\} \subseteq \mathcal{V}$ and for all $\mathrm{i_j} \in \mathcal{V}$, we have $(N-1)\Delta_{\mathrm{dec}}^{\mathrm{i_j}} > (z-1)\Delta_{\mathrm{dec}}^{\mathrm{i_j}}$ and $\sum_{\mathrm{i}_k\in\mathcal{V}\setminus\mathrm{i_j}} \Delta_{\mathrm{dec}}^{\mathrm{i}_k} > \sum_{\mathrm{i}_k\in\{\mathrm{i}_1,\ldots,\mathrm{i}_z\}\setminus\mathrm{i_j}} \Delta_{\mathrm{dec}}^{\mathrm{i}_k}$.*

We will use the above results to show that the optimal policy to solve Problem 1 is to target the node with the least modified health in a particular subset of nodes at each time-step, under certain conditions on the rates of repair and deterioration. This will then show that non-jumping policies are no longer necessarily optimal when $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}$ for one or more $\mathrm{j} \in \{1, \ldots, N\}$.

To derive this optimal policy, we start by presenting Algorithm 1, which generates a subset $\mathcal{Z}$ from the set of all nodes $\mathcal{V}$. Step 1 of the algorithm outputs a number x, which is the largest number such that there exists a set $\mathcal{A}(x)$ satisfying condition (2.16) in Lemma 5 (we will prove this below). Next, in Step 2, a specific subset $\mathcal{A}(x) \subseteq \mathcal{V}$ is created (termed $\mathcal{Z}$) such that $\mathcal{Z}$ is the set of x nodes with the largest sum of weights and that satisfies condition (2.16) in Lemma 5.

Algorithm 1 Generation of set $\mathcal{Z}$

Let there be $N(\geq 2)$ nodes.

1: **Computing the largest number x such that there exists a set $\mathcal{A}(x) = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{V}$ satisfying (2.16) in Lemma 5.** First, compute $\left\lceil \frac{v^{\mathrm{j}}_0}{\Delta^{\mathrm{j}}_{dec}} \right\rceil$ for each node $\mathrm{j} \in \mathcal{V}$. Then, set $x = 0$ and let $\mathcal{Y} = \mathcal{V}$ be the set of all N nodes. Repeat the following until the termination criterion is satisfied.

- If there is no node j in the set $\mathcal{Y}$ such that $\left\lceil \frac{v^{\mathrm{j}}_0}{\Delta^{\mathrm{j}}_{dec}} \right\rceil > x$, then terminate this step. Otherwise, let node $\mathrm{j} \in \mathcal{Y}$ be the node with the lowest value of $\left\lceil \frac{v^{\mathrm{j}}_0}{\Delta^{\mathrm{j}}_{dec}} \right\rceil$ that satisfies $\left\lceil \frac{v^{\mathrm{j}}_0}{\Delta^{\mathrm{j}}_{dec}} \right\rceil > x$ among all nodes in $\mathcal{Y}$. Remove node j from the set $\mathcal{Y}$ and set $x = x + 1$.

2: **Creating a set $\mathcal{Z}$ consisting of x nodes.** Let $\mathcal{R} = \mathcal{V}$ be the set of all N nodes, and let $\mathcal{Z} = \emptyset$. Among all nodes j in $\mathcal{R}$ whose initial health values are larger than $(x-1)\Delta^{\mathrm{j}}_{dec}$, remove the one whose weight is largest and add it to $\mathcal{Z}$. Next, among all nodes j in $\mathcal{R}$ whose initial health values are larger than $(x-2)\Delta^{\mathrm{j}}_{dec}$, remove the one whose weight is largest and add it to $\mathcal{Z}$. Continue in this way until x nodes have been added to $\mathcal{Z}$.

Remark 4. *Note that Algorithm 1 has polynomial time complexity. Specifically, Step 1 involves computing* $\left\lceil \frac{v^{\mathrm{j}}_0}{\Delta^{\mathrm{j}}_{dec}} \right\rceil$ *for each node* j*, which takes at most $O(N)$ operations, and then performing min operations over an $O(N)$ array at most N times. In Step 2, every iteration that involves choosing a node for set $\mathcal{Z}$ takes at most $O(N)$ operations (because it involves performing a max operation over an $O(N)$ array) and the maximum size of the set $\mathcal{Z}$ is N.*

We will now show that it is optimal to only target the nodes of set $\mathcal{Z}$ generated by Algorithm 1 in order to solve Problem 1. We first prove that Step 1 does indeed find the largest number x such that there exists a set $\mathcal{A}(x) \subseteq \mathcal{V}$ satisfying (2.16).

Lemma 7. *Let there be $N(\geq 2)$ nodes. The value of x that is computed in Step 1 of Algorithm 1 is the largest number such that there exists a set $\mathcal{A}(x) = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{V}$ satisfying (2.16) in Lemma 5.*

Proof. We prove this result through contradiction. Suppose the value of x that is computed in Step 1 of Algorithm 1 is not the largest number such that there exists a set $\mathcal{A}(x) = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{V}$ satisfying (2.16) in Lemma 5. Then, there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_y\} \subseteq \mathcal{V}$ of size $y(> x)$ satisfying

$$v_0^{\mathrm{i}_\mathrm{j}} > (y - \mathrm{j})\Delta_{dec}^{\mathrm{i}_\mathrm{j}}, \quad \forall \mathrm{j} \in \{1, \ldots, y\}. \tag{2.22}$$

Assume without loss of generality that these nodes are ordered such that

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil \geq \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil \geq \cdots \geq \left\lceil \frac{v_0^{\mathrm{i}_y}}{\Delta_{dec}^{\mathrm{i}_y}} \right\rceil. \tag{2.23}$$

Note that by (2.22), and the ordering given in (2.23), these quantities must satisfy

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > y - 1, \quad \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil > y - 2, \quad \ldots, \quad \left\lceil \frac{v_0^{\mathrm{i}_y}}{\Delta_{dec}^{\mathrm{i}_y}} \right\rceil > 0. \tag{2.24}$$

Now, under the above conditions, we compute the value of x in Step 1 of Algorithm 1. At the first iteration of Step 1, we have $\mathcal{Y} = \mathcal{V}$ and $x = 0$. By (2.24), there is at least one node $\mathrm{j} \in \mathcal{Y}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > 0$; for example, i_y satisfies this condition. Thus, Step 1 does not terminate at this iteration. Let $k_1 \in \mathcal{Y}$ be the node selected by Step 1, i.e., over all nodes $\mathrm{j} \in \mathcal{Y}$ that have $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > 0$, k_1 has the smallest such ratio. Note that $k_1 \notin \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{y-1}\}$ by the ordering in (2.23).

In the second iteration of Step 1, we have $\mathcal{Y} = \mathcal{V} \setminus \{k_1\}$ and $x = 1$. By (2.24), there is at least one node $\mathrm{j} \in \mathcal{Y}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > 1$; for example, i_{y-1} satisfies this condition. Thus, Step 1 does not terminate at this iteration. Let $k_2 \in \mathcal{Y}$ be the node selected by Step 1, i.e., over all nodes $\mathrm{j} \in \mathcal{Y}$ that have $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > 1$, k_2 has the smallest such ratio. Note that $k_2 \notin \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{y-2}\}$ by the ordering in (2.23).

Continuing in this way, in the r-th iteration of Step 1 (where $2 \leq r \leq y - 1$), we have $\mathcal{Y} = \mathcal{V} \setminus \{k_1, k_2, \ldots, k_{r-1}\}$ and $x = r - 1$. By (2.24), there is at least one node $\mathrm{j} \in \mathcal{Y}$ such that

$\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > r-1$; for example, i_{y-r+1} satisfies this condition. Thus, Step 1 does not terminate at the r-th iteration. Let $k_r \in \mathcal{Y}$ be the node selected by Step 1, i.e., over all nodes $\mathrm{j} \in \mathcal{Y}$ that have $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > r-1$, k_r has the smallest such ratio. Note that $k_r \notin \{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_{y-r}\}$ by the ordering in (2.23).

Finally, in the y-th iteration of Step 1, we have $\mathcal{Y} = \mathcal{V} \setminus \{k_1, k_2, \ldots, k_{y-1}\}$ and $x = y-1$. By (2.24), there is at least one node $\mathrm{j} \in \mathcal{Y}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > y-1$; for example, i_1 satisfies this condition. Let $k_y \in \mathcal{Y}$ be the node selected by Step 1, i.e., over all nodes $\mathrm{j} \in \mathcal{Y}$ that have $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > y-1$, k_y has the smallest such ratio. Thus, the variable x gets set to y at the end of this iteration. However, this leads to a contradiction because we assumed that $y > x$. Therefore, the value of x that is computed in Step 1 of Algorithm 1 is the largest number such that there exists a set $\mathcal{A}(x) \subseteq \mathcal{V}$ satisfying (2.16). □

We now come to the main result of this section.

Theorem 2.4.1. *Let there be $N(\geq 2)$ nodes and let $\mathcal{Z} = \{\mathrm{i}_1, \ldots, \mathrm{i}_{|\mathcal{Z}|}\}$ be the set that is formed by Algorithm 1, where $|\mathcal{Z}| = x$. Suppose $\Delta_{inc}^{\mathrm{i_j}} > (x-1)\Delta_{dec}^{\mathrm{i_j}}, \forall \mathrm{j} \in \{1, \ldots, x\}$ and $\Delta_{inc}^{\mathrm{i_j}} > \sum_{k \in \{1,\ldots,x\} \setminus \mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}, \forall \mathrm{j} \in \{1, \ldots, x\}$. Then, the optimal policy for Problem 1 is to target the node with the least modified health value in the set $\mathcal{Z}$ at each time-step. Under this policy, all nodes of the set $\mathcal{Z}$ are permanently repaired.*

Proof. Denote the policy that targets the node with the least modified health value in the set $\mathcal{Z}$ at each time-step as A. Then, by Lemma 6, all the nodes in set $\mathcal{Z}$ are permanently repaired by A as $\mathcal{Z}$ satisfies the condition (2.16) corresponding to the set $\mathcal{A}(x)$ (because of the way they are selected in Step 2 of Algorithm 1). Let B be a sequence other than the sequence A. Denote the reward obtained by sequences A and B as a and b, respectively. Denote the number of nodes that are permanently repaired by sequences A and B as x and y, respectively, and let $\mathcal{S}$ be the set of y nodes that are permanently repaired by sequence B. Then, $x \geq y$ by Lemma 5 and Lemma 7. We argue that $a \geq b$. Let $\mathrm{i_j}$ be the jth node that is added to the set $\mathcal{Z}$ by Step 2 of Algorithm 1. Denote the nodes of set $\mathcal{Z}$ by $\{\mathrm{i}_1, \ldots, \mathrm{i}_x\}$, and the nodes of set $\mathcal{S}$ as $\{\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_y\}$. In particular, the nodes $\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_y$ are ordered by performing a similar procedure as in Step 2 of Algorithm 1. That is, among all nodes j in $\mathcal{S}$ whose initial health values are larger than $(y-1)\Delta_{dec}^{\mathrm{j}}$, we denote the one with the largest weight as node

$\bar{\mathrm{i}}_1$. Next, among all nodes j (other than $\bar{\mathrm{i}}_\mathrm{i}$) in $\mathcal{S}$ whose initial health values are larger than $(y-2)\Delta^{\mathrm{j}}_{dec}$, we denote the one with the largest weight as $\bar{\mathrm{i}}_2$. We continue this until all the nodes $\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_y$ are defined. Note that there must exist at least one node whose initial health value satisfies the specified condition at each iteration, since the nodes in set $\mathcal{S}$ satisfy the condition (2.16) corresponding to the set $\mathcal{A}(y)$ in order for all to be permanently repaired.

We prove that there exists a one-to-one mapping between every element of set $\mathcal{S} = \{\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_y\}$ and an element of set $\mathcal{Z} = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\}$ such that each mapped node in $\mathcal{Z}$ has a weight that is at least as large as its paired node in $\mathcal{S}$, implying $a \geq b$ (note that it is possible to define such a mapping because $x \geq y$). Let the set of mapped nodes be denoted by $\mathcal{Z}^* = \{\mathrm{i}^*_1, \ldots, \mathrm{i}^*_y\} \subseteq \mathcal{Z}$. We create the set $\mathcal{Z}^*$ as follows. Node i^*_1 is the node with largest weight in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_{x-y+1}\}$, i^*_2 is the node with largest weight in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_{x-y+2}\} \setminus \mathrm{i}^*_1$, i^*_3 is the node with largest weight in the set $\{\mathrm{i}_1, \ldots, \mathrm{i}_{x-y+3}\} \setminus \{\mathrm{i}^*_1, \mathrm{i}^*_2\}$, and so on, until all y nodes of set $\mathcal{Z}^*$ have been defined. Next, for all $\mathrm{j} \in \{1, \ldots, y\}$, $\mathrm{i}^*_\mathrm{j} \in \mathcal{Z}^*$ is mapped to $\bar{\mathrm{i}}_\mathrm{j} \in \mathcal{S}$. We will argue that for all $\mathrm{j} \in \{1, \ldots, y\}, w^{\mathrm{i}^*_\mathrm{j}} \geq w^{\bar{\mathrm{i}}_\mathrm{j}}$.

First, note that i^*_1 is the node with largest weight among all nodes j in the set $\mathcal{V}$ whose initial health values are larger than $(y-1)\Delta^{\mathrm{j}}_{dec}$ due to the way the nodes $\{\mathrm{i}_1, \ldots, \mathrm{i}_{x-y+1}\}$ were chosen in Step 2 of Algorithm 1. Since $\bar{\mathrm{i}}_1$ is the node with largest weight among all nodes j in set $\mathcal{S}$ whose initial health values are larger than $(y-1)\Delta^{\mathrm{j}}_{dec}$ and $\mathcal{S} \subseteq \mathcal{V}$, $w^{\mathrm{i}^*_1} \geq w^{\bar{\mathrm{i}}_1}$ holds true. Next, note that the weight of i^*_2 satisfies the following: 1) it is at least as large as the second largest weight among all nodes j in set $\mathcal{V}$ whose initial health values are larger than $(y-1)\Delta^{\mathrm{j}}_{dec}$, and 2) it is at least as large as the largest weight among all nodes j in set $\mathcal{V}$ whose initial health values lie in the interval $((y-2)\Delta^{\mathrm{j}}_{dec}, (y-1)\Delta^{\mathrm{j}}_{dec}]$. Similarly, the weight of $\bar{\mathrm{i}}_2$ satisfies the following: 1) it is at least as large as the second largest weight among all nodes j in set $\mathcal{S}$ whose initial health values are larger than $(y-1)\Delta^{\mathrm{j}}_{dec}$ and 2) it is at least as large as the largest weight among all nodes j in set $\mathcal{S}$ whose initial health values lie in the interval $((y-2)\Delta^{\mathrm{j}}_{dec}, (y-1)\Delta^{\mathrm{j}}_{dec}]$. Therefore, $w^{\mathrm{i}^*_2} \geq w^{\bar{\mathrm{i}}_2}$ holds true because $\mathcal{S} \subseteq \mathcal{V}$. Continuing in this way we can show that for all $\mathrm{j} \in \{1, \ldots, y\}, w^{\mathrm{i}^*_\mathrm{j}} \geq w^{\bar{\mathrm{i}}_\mathrm{j}}$.

Thus, since the total weight of the nodes permanently repaired by sequence A is at least as large as the total weight of the nodes permanently repaired by any other sequence, we see that the sequence that targets the node with the least modified health in the set $\mathcal{Z}$ at each time-step is optimal for Problem 1. $\square$

We now provide an example to illustrate the generation of set $\mathcal{Z}$ and the policy of targeting the node with least modified health value in the set $\mathcal{Z}$ at each time-step.

Example 6. *Consider three nodes such that* $v_0^1 = 0.3, v_0^2 = 0.5, v_0^3 = 0.2,\ w^1 = 3, w^2 = 1, w^3 = 2,\ \Delta_{dec}^1 = 0.4, \Delta_{dec}^2 = 0.3, \Delta_{dec}^3 = 0.4,\ \Delta_{inc}^1 = 0.9,\ \Delta_{inc}^2 = 0.85$ *and* $\Delta_{inc}^2 = 0.95$. *The values of* $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil$ *for nodes 1, 2, and 3 are 1, 2, and 1, respectively. Applying Step 1 of Algorithm 1, the largest number* x *such that there exists a set* $\mathcal{A}(x) = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{V}$ *satisfying* (2.16) *is two. In Step 2 of the algorithm, node 2 is first selected for the set* $\mathcal{Z}$ *because it is the only node whose initial health value is larger than the corresponding deterioration rate. After this, node 1 is added to the set* $\mathcal{Z}$ *because it has the largest weight among the nodes 1 and 3, both of whose initial health values are positive. Therefore, set* $\mathcal{Z}$ *contains nodes 1 and 2. By Theorem 2.4.1, the optimal policy is to target the node with the least modified health value in the set* $\mathcal{Z}$ *at each time-step. At time-step 0, node 1 has the least modified health value in the set* $\mathcal{Z}$ *and thus it is targeted in the first time-step. Figure 2.4 shows the progression of health values of nodes when the optimal policy is followed (node 3 permanently fails after the first time-step and thus we do not show its health value from the second time-step onwards in the figure). The optimum reward in this example is thus given by* $w^1 + w^2 = 4$.

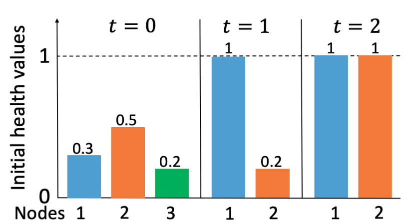

Figure 2.4. Progression of health values when the optimal policy is followed in Example 6.

We now consider a special case of Problem 1 when the weights are homogeneous across all the nodes, i.e., for all $\mathrm{j} \in \{1, \ldots, N\}$, $w^{\mathrm{j}} = \overline{w}$. We show that in this case, it is not required to generate the set $\mathcal{Z}$ through Algorithm 1 to optimally target the nodes.

Theorem 2.4.2. *Let there be $N(\geq 2)$ nodes such that for all* $\mathrm{j} \in \{1, \ldots, N\}$, $w^{\mathrm{j}} = \overline{w}$. *Suppose* $\Delta_{inc}^{\mathrm{j}} > (N-1)\Delta_{dec}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$ *and* $\Delta_{inc}^{\mathrm{j}} > \sum_{k \in \{1,\ldots,N\} \setminus \mathrm{j}} \Delta_{dec}^{k}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$. *Then, the policy that targets the node with the least modified health (and that has not permanently failed) at each time-step is optimal.*

Proof. Consider an optimal sequence A, and let $x(\leq N)$ be the number of nodes permanently repaired by that sequence. Denote the set of x nodes as $\mathcal{S}$. Then, set $\mathcal{S}$ satisfies the condition (2.16) corresponding to the set $\mathcal{A}(x)$ by Lemma 5. Based on the conditions on the repair and deterioration rates assumed in the theorem, the sequence B that targets the node with the least modified health at each time-step in $\mathcal{S}$ permanently repairs all of the nodes in $\mathcal{S}$ by Lemma 6.

Let $\mathcal{B}_0$ satisfy the condition (2.16) corresponding to the set $\mathcal{A}(x)$ at time-step 0 and denote the set of nodes that are in the permanent repair state at time-step 0 as $\overline{\mathcal{B}}_0$. Then, $\mathcal{B}_0$ is the set $\mathcal{S}$ and $\overline{\mathcal{B}}_0 = \emptyset$. Consider the policy in which the entity targets the node in $\mathcal{V}$ with the least modified health value (and that has not permanently failed) at each time-step. Then, in the first time-step, either the node with the least modified health value from the set $\mathcal{B}_0$ is targeted or a node outside the set $\mathcal{B}_0$ is targeted. If a node from the set $\mathcal{B}_0$ is targeted and at the end of first time-step no node gets permanently repaired, then $\mathcal{B}_0$ satisfies the condition (2.16) corresponding to the set $\mathcal{A}(x)$ and in that case we define the set $\mathcal{B}_1$ to be the same as set $\mathcal{B}_0$, and define $\overline{\mathcal{B}}_1 = \emptyset$. If a node from the set $\mathcal{B}_0$ is targeted and gets permanently repaired during that time-step, then the remaining $x-1$ nodes from the set $\mathcal{B}_0$ satisfy the condition (2.16) corresponding to the set $\mathcal{A}(x-1)$ (as argued in the proof of Lemma 6). In that case we define the set $\mathcal{B}_1$ to be the subset of $x-1$ nodes from $\mathcal{B}_0$ that are not permanently repaired, and define $\overline{\mathcal{B}}_1$ to be the node that lies in the set $\mathcal{B}_0 \setminus \mathcal{B}_1$. Consider the other case in which a node c not belonging to the set $\mathcal{B}_0$ is targeted in the first time-step. Then, if node c does not get permanently repaired, the health value of node c after the first time-step would be greater than $(x-1)\Delta_{dec}^{c}$ as $\Delta_{inc}^{c} > (N-1)\Delta_{dec}^{c} \geq (x-1)\Delta_{dec}^{c}$. Also, a set of $x-1$ nodes in the set $\mathcal{B}_0$ would satisfy the condition (2.16) corresponding to the set $\mathcal{A}(x-1)$:

$$v_1^{\mathrm{i_j}} = v_0^{\mathrm{i_j}} - \Delta_{dec}^{\mathrm{i_j}} > (x - \mathrm{j} - 1)\Delta_{dec}^{\mathrm{i_j}}, \quad \forall \mathrm{j} \in \{1, \ldots, x-1\}. \tag{2.25}$$

Thus, if node c does not get permanently repaired after the completion of the first time-step, then define $\mathcal{B}_1$ to be the set of nodes (consisting of node c and $x-1$ nodes from $\mathcal{B}_0$) that satisfy the condition (2.16) of the set $\mathcal{A}(x)$, and define $\overline{\mathcal{B}}_1 = \emptyset$. If node c gets permanently repaired after the completion of the first time-step, then $v_1^c = 1$ and the health values of a set of $x-1$ nodes in the set $\mathcal{B}_0$ satisfy (2.25). Thus, if node c gets permanently repaired after the end of the first time-step, then define $\mathcal{B}_1$ to be the set that satisfies the condition (2.16) corresponding to the set $\mathcal{A}(x-1)$, and define $\overline{\mathcal{B}}_1 = c$. We can repeat this argument for all the subsequent time-steps, noting that at the end of time-step t, depending on the sequence of nodes that are targeted by the entity, the initial health values of nodes, and deterioration and repair rates, there would always be a set $\mathcal{B}_t$ (of size x or less) that satisfies the condition (2.16) corresponding to the set $\mathcal{A}(|\mathcal{B}_t|)$ and there would be a set $\overline{\mathcal{B}}_t$ of size $x - |\mathcal{B}_t|$ consisting of permanently repaired nodes.

Denote the set of all nodes that have health values in the interval $(0, 1]$ at the beginning of time-step t by $\mathcal{C}_t$ (i.e., $\mathcal{C}_t$ consists of all the nodes except the nodes that are in the permanent failure state at the beginning of time-step t). Then, $\mathcal{C}_0 = \mathcal{V}$ and for all time-steps $t(\geq 0)$, $\mathcal{C}_{t+1} \subseteq \mathcal{C}_t$. Let a node $\mathrm{i_j} \in \mathcal{C}_t \setminus \overline{\mathcal{B}}_t$, where $\mathrm{j} \in \{1, \ldots, |\mathcal{C}_t \setminus \overline{\mathcal{B}}_t|\}$, be targeted by the entity at time-step t and assume that it does not get permanently repaired during time-step t. Recall that $\mathcal{C}_{t+1}$ consists of all the nodes except the nodes that are in permanent failure state at the beginning of time-step $t+1$. We now compute the difference in the average health values of the nodes in $\mathcal{C}_{t+1}$ and $\mathcal{C}_t$ as follows:

$$\frac{\sum_{\mathrm{i}_k \in \mathcal{C}_{t+1}} v_{t+1}^{\mathrm{i}_k}}{|\mathcal{C}_{t+1}|} - \frac{\sum_{\mathrm{i}_k \in \mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|} \geq \frac{\sum_{\mathrm{i}_k \in \mathcal{C}_{t+1}} v_{t+1}^{\mathrm{i}_k} - \sum_{\mathrm{i}_k \in \mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|}$$

$$= \frac{\sum_{\mathrm{i}_k \in \mathcal{C}_t} v_{t+1}^{\mathrm{i}_k} - \sum_{\mathrm{i}_k \in \mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|} = \frac{\sum_{\mathrm{i}_k \in \mathcal{C}_t \setminus \overline{\mathcal{B}}_t} \left(v_{t+1}^{\mathrm{i}_k} - v_t^{\mathrm{i}_k}\right)}{|\mathcal{C}_t|}$$

$$\geq \frac{\Delta_{inc}^{\mathrm{i_j}} - \sum_{k \in \{1, \ldots, |\mathcal{C}_t \setminus \overline{\mathcal{B}}_t|\} \setminus \mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}}{|\mathcal{C}_t|}.$$

The first inequality above is because $\mathcal{C}_{t+1} \subseteq \mathcal{C}_t$ for all time-steps t, the left equality is because the health value of a node belonging to the set $\mathcal{C}_t \setminus \mathcal{C}_{t+1}$ at the beginning of time-step $t+1$ is equal to zero (i.e., the permanent failure state), and the right equality is because all the nodes in the set $\overline{\mathcal{B}}_t$ are in permanent repair state for all time-steps greater than or equal to t. The last inequality is due to the fact that some of the nodes in the set $\mathcal{C}_t \setminus \overline{\mathcal{B}}_t$ that fail during time-step t may have had health values less than their corresponding deterioration rates, and thus the decrease in their health during that time-step will also be less than their deterioration rate. Note that $\frac{\Delta_{inc}^{\mathrm{i_j}} - \sum_{k \in \{1,\ldots,|\mathcal{C}_t \setminus \overline{\mathcal{B}}_t|\} \setminus \mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}}{|\mathcal{C}_t|}$ is lower bounded by a positive constant value equal to $\frac{\Delta_{inc}^{\mathrm{i_j}} - \sum_{k \in \{1,\ldots,N\} \setminus \mathrm{j}} \Delta_{dec}^{\mathrm{i}_k}}{N}$ because of the conditions on the repair and deterioration rates assumed in the theorem. Therefore, for each time-step $t(\geq 0)$, either the average health of the nodes in the set $\mathcal{C}_{t+1}$ is larger than the the average health of the nodes in the set $\mathcal{C}_t$, or a node from the set $\mathcal{C}_t \setminus \overline{\mathcal{B}}_t$ gets permanently repaired during time-step t, or both. Thus, x nodes would eventually get permanently repaired because $|\mathcal{C}_t| \geq x$ (as $\mathcal{B}_t \cup \overline{\mathcal{B}}_t \subseteq \mathcal{C}_t$), for all time-steps t. Therefore, if there is an optimal sequence A that permanently repairs $x(\leq N)$ nodes then the sequence that targets the node with the least modified health (and that has not permanently failed) at each time-step also permanently repairs x nodes. The result thus follows. □

Example 7. *We now provide an example to illustrate the policy of targeting the node with the lowest modified health value at each time-step. Consider the deterioration and repair rates to be homogeneous across the nodes,* $\Delta_{\text{dec}} = 0.1$*,* $\Delta_{\text{inc}} = 0.3$*, and three nodes having equal weights with initial health values* $v_0^1 = 0.8, v_0^2 = 0.6$ *and* $v_0^2 = 0.4$*. Figure 2.5 shows the progression of health values when the policy of targeting the least healthy node at each time-step is followed (note that when the deterioration rate is homogeneous across the nodes, targeting the least healthy node is equivalent to targeting the node with the least modified health value). Since node 3 is the least healthy node at the beginning of the first time-step it is targeted in the first time-step. At the beginning of second time-step, node 2 is the least healthy node and thus it is targeted in the second time-step, and so forth.*

It can be seen that optimal control sequences depend on the relationship between $\Delta_{dec}^{\mathrm{j}}$ and $\Delta_{inc}^{\mathrm{j}}$. When the rates and weights are homogeneous across all the nodes and $\Delta_{dec} \geq \Delta_{inc}$,

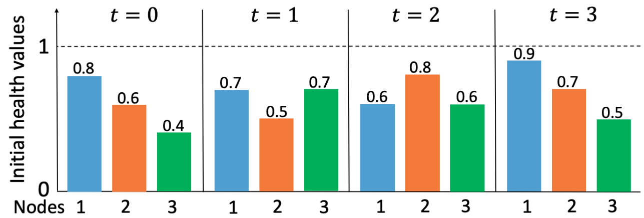

Figure 2.5. Progression of health values until $t = 3$ under the policy of targeting the least healthy node at each time-step (all nodes eventually get permanently repaired under this policy).

targeting the healthiest node at each time-step is the optimal feedback policy (under certain conditions on the initial health values) by Theorems 2.3.1 and 2.3.3, whereas targeting the least healthy node at each time-step is the optimal feedback policy when $\Delta_{\text{inc}} > (N-1)\Delta_{dec}$ by Theorem 2.4.2.

While we have identified the optimal policies for the above ranges of repair and deterioration rates, the characterization of the optimal policy when $\Delta_{dec} < \Delta_{\text{inc}} < (N-1)\Delta_{dec}$ remains open. We further investigate this case in Chapter 6. In the next section, we consider an extension of some of the above results when the deterioration and repair rates are time-varying.

2.5 Time-varying rates

In this section, we consider the case when the deterioration and repair rates are varying with time. Also, we focus on the case when the deterioration and repair rates are homogeneous across all nodes. Thus, if node j is being repaired by the entity at time-step t and it has not permanently failed or repaired, its health increases by a quantity $\Delta^t_{inc} \in [0, 1]$ (up to a maximum health of 1). If node j is not being repaired by the entity at time-step t and it has not permanently failed or repaired, its health decreases by a fixed quantity $\Delta^t_{dec} \in [0, 1]$

(down to a minimum health of 0). For each node j, the dynamics of the (controlled) recovery process are given by

$$v_{t+1}^{\mathrm{j}} = \begin{cases} 1 & \text{if } v_t^{\mathrm{j}} = 1, \\ 0 & \text{if } v_t^{\mathrm{j}} = 0, \\ \min(1, v_t^{\mathrm{j}} + \Delta_{inc}^t) & \text{if } u_t = \mathrm{j} \text{ and } v_t^{\mathrm{j}} \in (0,1), \\ \max(0, v_t^{\mathrm{j}} - \Delta_{\mathrm{dec}}^t) & \text{if } u_t \neq \mathrm{j} \text{ and } v_t^{\mathrm{j}} \in (0,1). \end{cases} \tag{2.26}$$

Lemma 8. *Let there be $N(\geq 2)$ nodes. Suppose the repair rate does not vary with time but deterioration rate can vary across time such that $\Delta_{\mathrm{dec}}^t = n_t \Delta_{\mathrm{inc}}, \forall t$, where n_t is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v_0^{\mathrm{j}} = m_{\mathrm{j}} \Delta_{\mathrm{inc}}$. Consider the two control sequences A and B targeting N nodes as shown in Figures 2.1 and 2.2, respectively. Suppose sequence A permanently repairs all nodes and contains exactly one jump, where the entity partially repairs node i_1 before moving to node i_2 at time-step $\bar{t}_1^A$. Sequence A then permanently repairs nodes $\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k$, before returning to node i_1 and permanently repairing it. Sequence B is a non-jumping sequence that targets nodes in the order $\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k, \mathrm{i}_1, \mathrm{i}_{k+1}, \ldots, \mathrm{i}_N$. Let t_{j}^A (resp. t_{j}^B) be the number of time-steps taken to permanently repair node i_{j} in sequence A (resp. sequence B). Then, sequence B also permanently repairs all nodes, and furthermore, the following holds true:*

$$t_{\mathrm{j}}^A \geq t_{\mathrm{j}}^B + \left(2^{\mathrm{j}-2}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{2, \ldots, k\}, \tag{2.27}$$

$$t_1^A \geq t_1^B + (2^{k-1} - 2)\bar{t}_1^A, \tag{2.28}$$

$$t_{\mathrm{j}}^A \geq t_{\mathrm{j}}^B + \left(2^{\mathrm{j}-1} - 2^{\mathrm{j}-k}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{k+1, \ldots, N\}. \tag{2.29}$$

Proof. Let $T_1^A, T_2^A, \ldots, T_N^A$ be the time-steps at which sequence A starts targeting a new node, as shown in Fig. 2.1. Similarly, let $T_1^B, T_2^B, \ldots, T_{N-1}^B$ be the time-steps at which sequence B starts targeting a new node, as shown in Fig. 2.2.

We start by first proving condition (2.27), using mathematical induction on the index of nodes in the sequence. Consider j = 2. At time-step T_1^A in sequence A, the health of node i_2 is given by

$$v_{T_1^A}^{\mathrm{i}_2} = v_0^{\mathrm{i}_2} - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{dec}^t.$$

We now calculate t_2^A as

$$\begin{aligned} t_2^A = \left\lceil \frac{1 - v_{T_1^A}^{\mathrm{i}_2}}{\Delta_{\mathrm{inc}}} \right\rceil &= \left\lceil \frac{1 - v_0^{\mathrm{i}_2} + \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{dec}^t}{\Delta_{\mathrm{inc}}} \right\rceil \\ &\geq \left\lceil \frac{1 - v_0^{\mathrm{i}_2}}{\Delta_{\mathrm{inc}}} \right\rceil + \bar{t}_1^A = t_2^B + \bar{t}_1^A, \end{aligned}$$

which satisfies condition (2.27).

Suppose that condition (2.27) holds for r nodes where $r < k$. If sequence A permanently repairs nodes $\mathrm{i}_2, \ldots, \mathrm{i}_r$, then so does sequence B (as each node is reached at an earlier time-step in sequence B than in sequence A, by the above inductive assumption). We now compute $v_{T_r^A}^{\mathrm{i}_{r+1}}$:

$$v_{T_r^A}^{\mathrm{i}_{r+1}} = v_0^{\mathrm{i}_{r+1}} - \sum_{t=0}^{T_r^A - 1} \Delta_{dec}^t,$$

where $T_r^A = \bar{t}_1^A + t_2^A + \ldots + t_r^A$. Note that

$$T_r^A = \bar{t}_1^A + t_2^A + \ldots + t_r^A \geq t_2^B + \ldots + t_r^B + \left(2^{r-1}\right) \bar{t}_1^A = T_{r-1}^B + \left(2^{r-1}\right) \bar{t}_1^A. \tag{2.30}$$

Thus,

$$\begin{aligned} t_{r+1}^A &= \left\lceil \frac{1 - v_0^{\mathrm{i}_{r+1}} + \sum_{t=0}^{T_r^A - 1} \Delta_{dec}^t}{\Delta_{\mathrm{inc}}} \right\rceil \\ &= \left\lceil \frac{1 - v_0^{\mathrm{i}_{r+1}} + \sum_{t=0}^{T_{r-1}^B - 1} \Delta_{dec}^t}{\Delta_{\mathrm{inc}}} + \frac{\sum_{t=T_{r-1}^B}^{T_r^A - 1} \Delta_{dec}^t}{\Delta_{\mathrm{inc}}} \right\rceil \\ &\geq \left\lceil \frac{1 - v_0^{\mathrm{i}_{r+1}} + \sum_{t=0}^{T_{r-1}^B - 1} \Delta_{dec}^t}{\Delta_{\mathrm{inc}}} \right\rceil + 2^{r-1}\bar{t}_1^A = t_{r+1}^B + 2^{r-1}\bar{t}_1^A. \end{aligned}$$

So, we have shown condition (2.27) by induction. We now prove condition (2.28). Node i_1 is targeted again in sequence A at time-step T_k^A, at which point its health is

$$v_{T_k^A}^{\mathrm{i}_1} = v_0^{\mathrm{i}_1} + \bar{t}_1^A \Delta_{\mathrm{inc}} - \sum_{t=\bar{t}_1^A}^{T_k^A - 1} \Delta_{\mathrm{dec}}^t.$$

Denote $n_t = \frac{\Delta_{\mathrm{dec}}^t}{\Delta_{\mathrm{inc}}}$. Then,

$$\begin{aligned} T_k^A &= \bar{t}_1^A + t_2^A + \ldots + t_k^A \\ &= \bar{t}_1^A + \left(t_2^B + \sum_{t=0}^{T_1^A - 1} n_t \right) + \left(t_3^B + \sum_{t=T_1^B}^{T_2^A - 1} n_t \right) + \ldots + \left(t_k^B + \sum_{t=T_{k-2}^B}^{T_{k-1}^A - 1} n_t \right) \\ &= \bar{t}_1^A + T_{k-1}^B + \sum_{t=0}^{T_1^A - 1} n_t + \sum_{t=T_1^B}^{T_2^A - 1} n_t + \ldots + \sum_{t=T_{k-2}^B}^{T_{k-1}^A - 1} n_t, \end{aligned} \tag{2.31}$$

because $t_k^A = t_k^B + \sum_{t=T_{k-2}^B}^{T_{k-1}^A - 1} n_t, \forall k \geq 3$ as no node gets permanently repaired in a partway time-step due to the assumptions made on the initial health values and deterioration and repair rates.

Let $\Delta_{dec}^{min} = \min_t \Delta_{\mathrm{dec}}^t, \forall t$. Suppose $T_{k-1}^B > \bar{t}_1^A$. Then,

$$\begin{aligned} \sum_{t=\bar{t}_1^A}^{T_k^A - 1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{T_{k-1}^B - 1} \Delta_{\mathrm{dec}}^t &= \sum_{t=\bar{t}_1^A}^{T_{k-1}^B - 1} \Delta_{\mathrm{dec}}^t + \sum_{t=T_{k-1}^B}^{T_k^A - 1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{\mathrm{dec}}^t - \sum_{t=\bar{t}_1^A}^{T_{k-1}^B - 1} \Delta_{\mathrm{dec}}^t \\ &= \sum_{t=T_{k-1}^B}^{T_k^A - 1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{\mathrm{dec}}^t \\ &\geq \Delta_{dec}^{min} \left(T_k^A - T_{k-1}^B \right) - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{\mathrm{dec}}^t \\ &= \Delta_{dec}^{min} \left(\bar{t}_1^A + \sum_{t=0}^{T_1^A - 1} n_t + \sum_{t=T_1^B}^{T_2^A - 1} n_t + \ldots + \sum_{t=T_{k-2}^B}^{T_{k-1}^A - 1} n_t \right) - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{\mathrm{dec}}^t \\ &\geq \Delta_{dec}^{min} \bar{t}_1^A + \sum_{t=0}^{T_1^A - 1} \Delta_{\mathrm{dec}}^t + \sum_{t=T_1^B}^{T_2^A - 1} \Delta_{\mathrm{dec}}^t + \ldots + \sum_{t=T_{k-2}^B}^{T_{k-1}^A - 1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{\bar{t}_1^A - 1} \Delta_{\mathrm{dec}}^t \\ &\geq \Delta_{dec}^{min} \bar{t}_1^A + \Delta_{dec}^{min} \left(2 + \ldots + 2^{k-2} \right) \bar{t}_1^A = \Delta_{dec}^{min} \left(2^{k-1} - 1 \right) \bar{t}_1^A, \end{aligned} \tag{2.32}$$

where the first inequality comes from the definition of Δ_{dec}^{min}, the next equality comes from (2.31), the inequality after that comes from the fact that for all t, $\Delta_{dec}^{min} n_t = \frac{\Delta_{dec}^{min}\Delta_{\mathrm{dec}}^t}{\Delta_{\mathrm{inc}}} \geq \Delta_{\mathrm{dec}}^t$, and the last inequality comes from the fact that for all $k \geq 2$, $T_k^A - T_{k-1}^B \geq 2^{k-1}\bar{t}_1^A$ (by claim (2.27)).

Suppose $T_{k-1}^B \leq \bar{t}_1^A$. Then,

$$\begin{aligned}
\sum_{t=\bar{t}_1^A}^{T_k^A-1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{T_{k-1}^B-1} \Delta_{\mathrm{dec}}^t &= \sum_{t=T_{k-1}^B}^{\bar{t}_1^A-1} \Delta_{\mathrm{dec}}^t + \sum_{t=\bar{t}_1^A}^{T_k^A-1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{T_{k-1}^B-1} \Delta_{\mathrm{dec}}^t - \sum_{t=T_{k-1}^B}^{\bar{t}_1^A-1} \Delta_{\mathrm{dec}}^t \\
&= \sum_{t=T_{k-1}^B}^{T_k^A-1} \Delta_{\mathrm{dec}}^t - \sum_{t=0}^{\bar{t}_1^A-1} \Delta_{\mathrm{dec}}^t \\
&\geq \Delta_{dec}^{min}\left(2^{k-1}-1\right)\bar{t}_1^A,
\end{aligned} \tag{2.33}$$

where the inequality comes from (2.32). Thus, the number of time-steps taken to permanently repair node i_1 in sequence A (the second time it is targeted in the sequence) is

$$\begin{aligned}
t_1^A &= \frac{1 - v_0^{\mathrm{i}_1} - \bar{t}_1^A\Delta_{\mathrm{inc}} + \sum_{t=\bar{t}_1^A}^{T_k^A-1} \Delta_{\mathrm{dec}}^t}{\Delta_{inc}} \\
&\geq \frac{1 - v_0^{\mathrm{i}_1} - \bar{t}_1^A\Delta_{\mathrm{inc}} + \Delta_{dec}^{min}\left(2^{k-1}-1\right)\bar{t}_1^A + \sum_{t=0}^{T_{k-1}^B-1} \Delta_{\mathrm{dec}}^t}{\Delta_{inc}} \\
&\geq \frac{1 - v_0^{\mathrm{i}_1} + \sum_{t=0}^{T_{k-1}^B-1} \Delta_{\mathrm{dec}}^t}{\Delta_{\mathrm{inc}}} + \left(2^{k-1}-2\right)\bar{t}_1^A = t_1^B + \left(2^{k-1}-2\right)\bar{t}_1^A,
\end{aligned}$$

where the first inequality comes from (2.32) and (2.33), and the last inequality comes from the fact that $\Delta_{dec}^{min} \geq \Delta_{inc}$. Thus, condition (2.28) holds true.

We now prove condition (2.29) via mathematical induction. Consider node i_{k+1}. At the time-step when this node is targeted in sequence A, its health is

$$v_{T_{k+1}^A}^{\mathrm{i}_{k+1}} = v_0^{\mathrm{i}_{k+1}} - \sum_{t=0}^{T_{k+1}^A-1} \Delta_{\mathrm{dec}}^t.$$

Note that

$$T_{k+1}^A = \bar{t}_1^A + t_2^A + \ldots + t_k^A + t_1^A \geq t_2^B + \ldots + t_k^B + t_1^B + \left(2^{k-1}\right)\bar{t}_1^A + \left(2^{k-1} - 2\right)\bar{t}_1^A = T_k^B + \left(2^k - 2\right)\bar{t}_1^A. \tag{2.34}$$

If node i_{k+1} has not failed at this point in sequence A, it has also not failed when it is reached in sequence B (as all nodes prior to i_{k+1} are permanently repaired faster in sequence B than in sequence A, as shown above). Thus, using (2.27) and (2.28),

$$\begin{aligned} t_{k+1}^A &= \left\lceil \frac{1 - v_0^{\mathrm{i}_{k+1}} + \sum_{t=0}^{T_{k+1}^A - 1} \Delta_{\text{dec}}^t}{\Delta_{\text{inc}}} \right\rceil \\ &= \left\lceil \frac{1 - v_0^{\mathrm{i}_{k+1}} + \sum_{t=0}^{T_k^B - 1} \Delta_{\text{dec}}^{\mathrm{i}_{k+1},t}}{\Delta_{\text{inc}}} + \frac{\sum_{t=T_k^B}^{T_{k+1}^A - 1} \Delta_{\text{dec}}^t}{\Delta_{\text{inc}}} \right\rceil \\ &\geq \left\lceil \frac{1 - v_0^{\mathrm{i}_{k+1}} + \sum_{t=0}^{T_k^B - 1} \Delta_{\text{dec}}^{\mathrm{i}_{k+1},t}}{\Delta_{\text{inc}}} \right\rceil + \left(2^k - 2\right)\bar{t}_1^A = t_{k+1}^B + \left(2^k - 2\right)\bar{t}_1^A. \end{aligned}$$

by (2.34), which satisfies condition (2.29). Suppose condition (2.29) holds for $\mathrm{j} \in \{k + 1, \ldots, r\}$, where $r < N$. Consider node i_{r+1}. Then, a similar inductive argument can be used to show that

$$t_{r+1}^A \geq t_{r+1}^B + \left(2^r - 2^{r+1-k}\right)\bar{t}_1^A.$$

This proves the third claim. □

Lemma 9. *Let there be $N(\geq 2)$ nodes. Suppose repair rates are homogeneous across all the nodes but deterioration rate can vary across time such that $\Delta_{\text{dec}}^t = n_t \Delta_{\text{inc}}, \forall t$, where n_t is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v_0^{\mathrm{j}} = m_{\mathrm{j}} \Delta_{\text{inc}}$. If there is a sequence A with one or more jumps that permanently repairs all the nodes of a set $\mathcal{Z} \subseteq \mathcal{V}$, then there exists a non-jumping sequence that permanently repairs all the nodes in set $\mathcal{Z}$ in less time. Then, non-jumping policies are optimal.*

The proof of the above result proceeds in the same way as the proof of Theorem 2.3.1.

Non-jumping sequences

Lemma 10. *Let there be $N(\geq 2)$ nodes. Suppose the repair rates are homogeneous across all the nodes but deterioration rate can vary across time such that $\Delta_{\mathrm{dec}}^{t} = n_t \Delta_{\mathrm{inc}}, \forall t$, where n_t is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v_0^{\mathrm{j}} = m_{\mathrm{j}} \Delta_{inc}$. Then, the non-jumping sequence that targets nodes in decreasing order of their initial health is optimal.*

Proof. Consider any optimal (non-jumping) sequence A, and let x be the number of nodes that are permanently repaired by that sequence. Denote this set of $x(\leq N)$ nodes as $\mathcal{Z}$. Let $\mathrm{i}_1, \ldots, \mathrm{i}_x$ be the order in which the sequence A permanently repairs the x nodes. Denote t_{j} to be the time taken to permanently repair node i_{j} in A. Note that the necessary and sufficient conditions to permanently repair x nodes if a non-jumping sequence A targets the nodes in the order $\mathrm{i}_1, \ldots, \mathrm{i}_x$ are as follows

$$v_0^{\mathrm{i}_1} > 0 \tag{2.35}$$

$$v_0^{\mathrm{i}_2} - \sum_{t=0}^{t_1 - 1} \Delta_{dec}^{t} > 0 \tag{2.36}$$

$$v_0^{\mathrm{i}_3} - \sum_{t=0}^{t_1 + t_2 - 1} \Delta_{dec}^{t} > 0 \tag{2.37}$$

$$\vdots$$

$$v_0^{\mathrm{i}_x} - \sum_{t=0}^{t_1 + \ldots + t_{x-1} - 1} \Delta_{dec}^{t} > 0. \tag{2.38}$$

We now argue that the left-hand side (LHS) of all the conditions (2.35)-(2.38) would be the largest when node i_1 has the largest initial health value, node i_2 has the second largest initial health value, node i_3 has the third largest initial health value and so on. Note that the LHS of (2.35) would be largest when node i_1 has the largest initial health value. We now focus on the remaining conditions. Note that for all $\mathrm{j} \in \{2, \ldots, x\}$, $v_0^{\mathrm{i}_{\mathrm{j}}} - \sum_{t=0}^{t_1 + \ldots + t_{\mathrm{j}-1} - 1} \Delta_{dec}^{t}$ would be largest when $c_{\mathrm{j}} = 1 - v_0^{\mathrm{i}_{\mathrm{j}}} + \sum_{t=0}^{t_1 + \ldots + t_{\mathrm{j}-1} - 1} \Delta_{dec}^{t}$ is the smallest. We now argue through induction on the index of nodes that for all $\mathrm{j} \in \{2, \ldots, x\}$, $c_{\mathrm{j}} = \Delta_{inc}\left(a_1^{\mathrm{j}} m_1 + \ldots + a_{\mathrm{j}-1}^{\mathrm{j}} m_{\mathrm{j}-1} + m_{\mathrm{j}}\right)$ where $a_1^{\mathrm{j}} \geq 1, \ldots, a_{\mathrm{j}-1}^{\mathrm{j}} \geq 1$ and c_{j} is the smallest when m_1 is the smallest, m_2 is the second smallest and so on (recall that for all $\mathrm{j} \in \{1, \ldots, N\}$, $m_{\mathrm{j}} = \frac{1-v_0^{\mathrm{j}}}{\Delta_{\mathrm{inc}}}$). Note that

$$c_2 = 1 - v_0^{\mathrm{i}_2} + \sum_{t=0}^{t_1-1} \Delta_{dec}^t = \Delta_{\mathrm{inc}}\left(m_2 + \sum_{t=0}^{m_1-1} n_t\right) = \Delta_{inc}\left(a_1^2 m_1 + m_2\right), \tag{2.39}$$

where $a_1^2 \geq 1$ because $n_t \geq 1, \forall t$. Thus, c_2 in (2.39) would be the smallest when m_1 takes the smallest possible value and m_2 takes the second smallest value (i.e., when node i_1 has the largest initial health value and node i_2 has the second largest initial health value) because $a_1^2 \geq 1$. We now assume that the inductive assumption holds for each node $\mathrm{i}_{\mathrm{j}-1}$ where $2 < \mathrm{j} \leq x$ and argue that it also holds for node i_{j}. Note that the time at which node $\mathrm{j} \in \{2, \ldots, x\}$ is reached in A is equal to $b_{\mathrm{j}} = m_1 + c_2 + \ldots + c_{\mathrm{j}-1} = m_1 + (a_1^2 m_1 + m_2) + \ldots + \left(a_1^{\mathrm{j}-1} m_1 + \ldots + a_{\mathrm{j}-2}^{\mathrm{j}-1} m_{\mathrm{j}-2} + m_{\mathrm{j}-1}\right)$ from the inductive assumption. Thus,

$$c_{\mathrm{j}} = 1 - v_0^{\mathrm{i}_{\mathrm{j}}} + \sum_{t=0}^{t_1+\ldots+t_{\mathrm{j}-1}-1} \Delta_{dec}^t = \Delta_{inc}\left(m_{\mathrm{j}} + \sum_{t=0}^{b_{\mathrm{j}}-1} n_t\right) = \Delta_{inc}\left(a_1^{\mathrm{j}} m_1 + \ldots + a_{\mathrm{j}-1}^{\mathrm{j}} m_{\mathrm{j}-1} + m_{\mathrm{j}}\right),$$

where the last equality comes from that fact that $a_k^l \geq 1, \forall k \leq \mathrm{j}-2, l \leq \mathrm{j}-1$ and $n_t \geq 1 \forall t$. From the inductive hypothesis, we know that for all $k \in \{2, \ldots, \mathrm{j}-1\}$, c_k is the smallest when m_1 has the smallest value, m_2 has the second smallest value and so on. From the inductive hypothesis and from the fact that $b_{\mathrm{j}}\Delta_{inc} = c_1 + \ldots + c_{\mathrm{j}-1}$, b_{j} is the smallest when m_1 has the smallest value, m_2 has the second smallest value and so on. Therefore, $a_1^{\mathrm{j}} m_1 + \ldots + a_{\mathrm{j}-1}^{\mathrm{j}} m_{\mathrm{j}-1}$ would be the smallest when m_1 has the smallest value, m_2 has the second smallest value and so on, among $\{m_1, \ldots, m_{\mathrm{j}-1}\}$. Since $a_1^{\mathrm{j}} \geq 1, \ldots, a_{\mathrm{j}-1}^{\mathrm{j}} \geq 1$, c_{j} would be the smallest when m_{j} has the largest value among $\{m_1, \ldots, m_{\mathrm{j}}\}$. Therefore, c_{j} would be the smallest when m_1 has the smallest value, m_2 has the second smallest value and so on, among $\{m_1, \ldots, m_{\mathrm{j}}\}$, proving the induction hypothesis. Thus, the non-jumping sequence B that targets the nodes of set $\mathcal{Z}$ in decreasing order of their initial health values also satisfies the conditions (2.35)-(2.38); and thus it would also permanently repair x nodes and hence will be optimal (since it permanently repairs the same number of nodes as the optimal sequence A). Consider another non-jumping sequence C that targets the top x nodes with the largest initial health values from the N

nodes. Then, the sequence C would also permanently repair x nodes. This is because each node in sequence C has a higher initial health value (or at least equal) to the corresponding node in sequence B and thus sequence C satisfies the conditions (2.35)-(2.38). Thus, the policy of targeting the nodes in decreasing order of their initial health values would also permanently repair x nodes, and hence is optimal. □

2.5.1 Optimal policies for $\Delta_{dec} < \Delta_{inc}$

Lemma 11. *Let there be $N(\geq 2)$ nodes. Then, the necessary condition for all the nodes to eventually get permanently repaired (regardless of the rates of repair and deterioration) is that there exists a permutation $(\mathrm{i}_1, \ldots, \mathrm{i}_N)$ such that*

$$v_0^{\mathrm{i}_N} > 0 \tag{2.40}$$

$$v_0^{\mathrm{i_j}} > \sum_{t=0}^{N-\mathrm{j}-1} \Delta_{\mathrm{dec}}^t, \quad \forall \mathrm{j} \in \{1, \ldots, N-1\}. \tag{2.41}$$

Proof. Suppose there exists a sequence that permanently repairs all the nodes. At each time-step t, use $\mathcal{C}^t$ to denote the set of nodes that have not been targeted at least once by the entity prior to t. Note that $\mathcal{C}^0 \supseteq \mathcal{C}^1 \supseteq \ldots \supseteq \mathcal{C}^{N-1}$. At $t = 0$, $|\mathcal{C}^t| = N$ where $|\mathcal{C}^t|$ denotes the cardinality of set $\mathcal{C}^t$. At time $t = 1, |\mathcal{C}^t| = N - 1$ as there are $N - 1$ nodes that have not been targeted by the entity at least once. Each node k belonging to the set $\mathcal{C}^1$ should have initial health value larger than Δ_{dec}^0 to survive until $t = 1$. At $t = 2, |\mathcal{C}^t| \geq N - 2$ as there are at least $N - 2$ nodes that have not been targeted by the entity at least once. Each node k belonging to the set $\mathcal{C}^2$ should have initial health value larger than $\Delta_{\mathrm{dec}}^0 + \Delta_{\mathrm{dec}}^1$ to survive until $t = 2$. Repeating this argument for the next $N - 4$ time-steps proves that there should be an permutation $(\mathrm{i}_1, \ldots, \mathrm{i}_N)$ that should satisfy the conditions (2.40)-(2.41) for all the nodes to eventually get permanently repaired. Note that (2.40)-(2.41) represent necessary conditions that need to be satisfied by *any* sequence that permanently repairs all the nodes, regardless of the rates of repair and deterioration. □

Lemma 12. *Let there be $N(\geq 2)$ nodes, $\Delta_{inc}^{min} = \min_t \Delta_{\mathrm{inc}}^t$ and $\Delta_{dec}^{max} = \max_t \Delta_{\mathrm{dec}}^t$. Suppose $\Delta_{inc}^{min} > (N-1)\Delta_{dec}^{max}$. Suppose there exists a permutation $(\mathrm{i}_1, \ldots, \mathrm{i}_N)$ such that* (2.40)-(2.41) *are satisfied. Then, the sequence that targets the least healthy node at each time-step will permanently repair all the nodes.*

Proof. Suppose the initial health values of the nodes satisfy the order $v_0^1 \geq \ldots \geq v_0^N$. If the sequence that targets the least healthy node at each time-step is followed, then after the completion of the first time-step if node N is not permanently repaired, then the health values of the nodes are given by

$$\begin{aligned}
v_1^N &= v_0^N + \Delta_{inc}^0 > (N-1)\Delta_{dec}^{max} \geq \sum_{t=1}^{N-1} \Delta_{dec}^t, \\
v_1^{\mathrm{j}} &= v_0^{\mathrm{j}} - \Delta_{dec}^0 > \sum_{t=1}^{N-\mathrm{j}-1} \Delta_{dec}^t \ \forall \mathrm{j} \in \{1, \ldots, N-2\}, \\
v_1^{N-1} &= v_0^{N-1} - \Delta_{dec}^0 > 0.
\end{aligned}$$

Note that the least health of nodes in the sequence after completion of the first time-step would be of either node $N-1$ or N. First, suppose $v_1^N < v_1^{N-1}$. Since node N has the least health, $v_1^{\mathrm{j}} > \sum_{t=1}^{N-1} \Delta_{dec}^t \ \forall \mathrm{j} \in \{1, \ldots, N-1\}$. Thus, conditions (2.40)-(2.41) are trivially satisfied with time shifted by one. Now, suppose $v_1^N > v_1^{N-1}$. Then, the N nodes satisfy the conditions (2.40)-(2.41) with time shifted by one and with the indices reordered to correspond to a new permutation of nodes. That is, the lowest health (i.e., that of node $N-1$) is positive, the second lowest node has health larger than Δ_{dec}^1, the third lowest node has health larger than $\Delta_{dec}^1 + \Delta_{dec}^2$, and so on. If node N is permanently repaired after the completion of the first time-step, then $v_1^N = 1$ and the health values of the remaining nodes after the end of the first time-step is given by (2.40)-(2.41). Thus, one of the nodes is permanently repaired and there exists a set of $N-1$ nodes that satisfy (2.40)-(2.41) with time shifted by one and N replaced by $N-1$. Thus, at any time-step, if there are $x(\leq N)$ nodes that satisfy the conditions (2.40)-(2.41) with time shifted away from zero, with N replaced by x, and $N-x$ nodes that are permanently repaired, then there will be a permutation of $y(\leq x)$ nodes that satisfies the conditions (2.40)-(2.41) with time shifted away from zero, with N replaced by y and $N-y$ nodes that are permanently repaired at the start of the next time-step. Therefore, no node's health would become zero at any time. Furthermore, if node j is targeted by the entity at a time-step and it does not get permanently repaired in less than a complete time-step then the average health of all the nodes increases by at least $\frac{\Delta_{inc}^{min}-(N-1)\Delta_{dec}^{max}}{N}$, $\quad$ j $\in \{1,\ldots,N\}$. Note that $\frac{\Delta_{inc}^{min}-(N-1)\Delta_{dec}^{max}}{N} > 0$ as $\Delta_{\text{inc}}^{min} > (N-1)\Delta_{\text{dec}}^{max}$. So, either the increase in average health at each time-step is positive or a node gets permanently repaired in a time-step or both. Therefore, all the nodes would eventually get permanently repaired. $\square$

Lemma 13. *Let there be $N(\geq 2)$ nodes, $\Delta_{\text{inc}}^{min} = \min_t \Delta_{\text{inc}}^{t}$ and $\Delta_{\text{dec}}^{max} = \max_t \Delta_{\text{dec}}^{t}$. Suppose $\Delta_{\text{inc}}^{min} > (N-1)\Delta_{\text{dec}}^{max}$. Then, the optimal policy is to target the least healthy node at each time-step.*

Proof. Consider an optimal sequence A, and let $x(\leq N)$ be the number of nodes permanently repaired by that sequence. Denote the set of x nodes as $\mathcal{S}$. By Lemma 11, there exists a permutation $(\mathrm{i}_1,\ldots,\mathrm{i}_x)$ of the nodes in the set $\mathcal{S}$ such that (2.40)-(2.41) are satisfied when N is replaced by x. Based on the conditions on the repair and deterioration rates assumed in the proposition, the sequence B that targets the node with the least modified health at each time-step in $\mathcal{S}$ permanently repairs all of the nodes in $\mathcal{S}$ by Lemma 12.

Let $\mathcal{B}_0$ be the set of nodes that satisfies (2.40)-(2.41) (with N replaced by x) at time-step 0 and denote the set of nodes that are in the permanent repair state at time-step 0 as $\overline{\mathcal{B}}_0$. Then, $\mathcal{B}_0$ is the set $\mathcal{S}$ and $\overline{\mathcal{B}}_0 = \emptyset$. Consider the policy in which the entity targets the node in $\mathcal{V}$ with the least health value (and that has not permanently failed) at each time-step. Then, in the first time-step, either the node with the least health value from the set $\mathcal{B}_0$ is targeted or a node outside the set $\mathcal{B}_0$ is targeted. If a node from the set $\mathcal{B}_0$ is targeted and at the end of first time-step no node gets permanently repaired, then all the nodes from the set $\mathcal{B}_0$ satisfy the conditions (2.40)-(2.41) with time shifted by one and with N replaced by x and in that case we define the set $\mathcal{B}_1$ to be the same as set $\mathcal{B}_0$, and define $\overline{\mathcal{B}}_1 = \emptyset$. If a node from the set $\mathcal{B}_0$ is targeted and gets permanently repaired during that time-step, then the remaining $x-1$ nodes from the set $\mathcal{B}_0$ satisfy the conditions (2.40)-(2.41) with time shifted by one and when N is replaced by $x-1$ (as argued in the proof of Lemma 12). In that case we define the set $\mathcal{B}_1$ to be the subset of $x-1$ nodes from $\mathcal{B}_0$ that are not permanently repaired, and define $\overline{\mathcal{B}}_1$ to be the node that lies in the set $\mathcal{B}_0 \setminus \mathcal{B}_1$. Consider the other case in which a node c not belonging to the set $\mathcal{B}_0$ is targeted in the first time-step. Then, if node c does not get permanently repaired, the health value of node c after the first time-step would be greater than $\sum_{t=1}^{x-1} \Delta_{dec}^{t}$ as $\Delta_{inc}^{min} > (N-1)\Delta_{dec}^{max} \geq \sum_{t=1}^{x-1} \Delta_{dec}^{t}$. Also, a set of $x-1$ nodes in the set $\mathcal{B}_0$ would satisfy the following due to conditions (2.40)-(2.41) with time shifted by one and with N replaced by x:

$$v_1^{\mathrm{i_j}} = v_0^{\mathrm{i_j}} - \Delta_{dec}^{t} > \sum_{t=1}^{x-\mathrm{j}-1} \Delta_{dec}^{t}, \quad \forall \mathrm{j} \in \{1, \ldots, x-1\}. \tag{2.42}$$

Thus, if node c does not get permanently repaired after the completion of the first time-step, then define $\mathcal{B}_1$ to be the set of nodes (consisting of node c and $x-1$ nodes from $\mathcal{B}_0$) that satisfies the conditions (2.40)-(2.41) with time shifted by one and with z replaced by x, and define $\overline{\mathcal{B}}_1 = \emptyset$. If node c gets permanently repaired after the completion of the first time-step, then $v_1^c = 1$ and the health values of a set of $x-1$ nodes in the set $\mathcal{B}_0$ satisfy (2.42). Thus, if node c gets permanently repaired after the end of the first time-step, then define $\mathcal{B}_1$ to be the set that consists of $x-1$ nodes that satisfy the conditions (2.40)-(2.41) with time shifted by one and with z replaced by $x-1$, and define $\overline{\mathcal{B}}_1 = c$. We can repeat this argument for all the subsequent time-steps, noting that at the end of time-step t, depending on the sequence of nodes that are targeted by the entity, the initial health values of nodes, and deterioration and repair rates, there would always be a set $\mathcal{B}_t$ (of size x or less) that would satisfy the conditions (2.40)-(2.41) with time shifted and with z replaced by $|\mathcal{B}_t|$ and there would be a set $\overline{\mathcal{B}}_t$ of size $x - |\mathcal{B}_t|$ consisting of permanently repaired nodes.

Denote the set of all nodes that have health values in the interval $(0, 1]$ at the beginning of time-step t by $\mathcal{C}_t$ (i.e., $\mathcal{C}_t$ consists of all the nodes except the nodes that are in the permanent failure state at the beginning of time-step t). Then, $\mathcal{C}_0 = \mathcal{V}$ and for all time-steps $t(\geq 0)$, $\mathcal{C}_{t+1} \subseteq \mathcal{C}_t$. Let a node $\mathrm{i_j} \in \mathcal{C}_t \setminus \overline{\mathcal{B}}_t$, where $\mathrm{j} \in \{1, \ldots, |\mathcal{C}_t \setminus \overline{\mathcal{B}}_t|\}$, be targeted by the entity at time-step t and assume that it does not get permanently repaired during time-step t. Recall that $\mathcal{C}_{t+1}$ consists of all the nodes except the nodes that are in permanent failure state at the beginning of time-step $t+1$. We now compute the difference in the average health values of the nodes in $\mathcal{C}_{t+1}$ and $\mathcal{C}_t$ as follows:

$$\begin{aligned}
&\frac{\sum_{\mathrm{i}_k\in\mathcal{C}_{t+1}} v_{t+1}^{\mathrm{i}_k}}{|\mathcal{C}_{t+1}|} - \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|} \\
&\qquad\qquad \geq \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_{t+1}} v_{t+1}^{\mathrm{i}_k}}{|\mathcal{C}_t|} - \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|} \\
&\qquad\qquad = \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_t} v_{t+1}^{\mathrm{i}_k}}{|\mathcal{C}_t|} - \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_t} v_t^{\mathrm{i}_k}}{|\mathcal{C}_t|} \\
&\qquad\qquad = \frac{\sum_{\mathrm{i}_k\in\mathcal{C}_t\setminus\overline{\mathcal{B}}_t} \left(v_{t+1}^{\mathrm{i}_k} - v_t^{\mathrm{i}_k}\right)}{|\mathcal{C}_t|} \\
&\qquad\qquad \geq \frac{\Delta_{\mathrm{inc}}^{min} - \sum_{k\in\{1,\ldots,|\mathcal{C}_t\setminus\overline{\mathcal{B}}_t|\}\setminus\mathrm{j}} \Delta_{\mathrm{dec}}^t}{|\mathcal{C}_t|}.
\end{aligned}$$

The first inequality above is because $\mathcal{C}_{t+1} \subseteq \mathcal{C}_t$ for all time-steps t, the first equality is because the health value of a node belonging to the set $\mathcal{C}_t \setminus \mathcal{C}_{t+1}$ at the beginning of time-step $t+1$ is equal to zero (i.e., the permanent failure state), and the next equality is because all the nodes in the set $\overline{\mathcal{B}}_t$ are in permanent repair state for all time-steps greater than or equal to t. The last inequality is due to the fact that some of the nodes in the set $\mathcal{C}_t \setminus \overline{\mathcal{B}}_t$ that fail during time-step t may have had health values less than deterioration rate, and thus the decrease in their health during that time-step will also be less than their deterioration rate. Note that $\frac{\Delta_{inc}^{min} - \sum_{k\in\{1,\ldots,|\mathcal{C}_t\setminus\overline{\mathcal{B}}_t|\}\setminus\mathrm{j}} \Delta_{dec}^t}{|\mathcal{C}_t|}$ is lower bounded by a positive constant value equal to $\frac{\Delta_{inc}^{min} - (N-1)\Delta_{dec}^{max}}{N} > 0$ because of the conditions on the repair and deterioration rates assumed in the proposition. Therefore, for each time-step $t(\geq 0)$, either the average health of the nodes in the set $\mathcal{C}_{t+1}$ is larger than the the average health of the nodes in the set $\mathcal{C}_t$, or a node from the set $\mathcal{C}_t \setminus \overline{\mathcal{B}}_t$ gets permanently repaired during time-step t, or both. Thus, x nodes would eventually get permanently repaired because $|\mathcal{C}_t| \geq x$ (as $\mathcal{B}_t \cup \overline{\mathcal{B}}_t \subseteq \mathcal{C}_t$), for all time-steps t. Therefore, if there is an optimal sequence A that permanently repairs $x(\leq N)$ nodes then the sequence that targets the least healthy node (and that has not permanently failed) at each time-step also permanently repairs x nodes. The result thus follows. □

Remark 5. *The above results for the time-varying deterioration and repair rates suggest that as long as the required relationships between the deterioration and repair rates are satisfied* *across all time-steps, it is still possible to characterize policies such as targeting the healthiest node at each time-step or the least healthy node at each time-step to be optimal for various cases of the problem. Therefore, such policies are* ***robust*** *to the variations of rates with time as long as the required relationships are satisfied.*

2.6 Conclusions

In this chapter, we characterized optimal sequencing policies of recovery actions in a post-disaster scenario, where multiple physical infrastructure components have been damaged, and an agency wishes to fully repair as many components as possible. We first considered the case when the deterioration and repair rates are fixed with time. For this case, we provided several characteristics of the optimal policy. We then showed that under certain conditions on the rates of repair and deterioration, we can explicitly characterize the optimal control policy as a function of the states. When the deterioration rate (when not being repaired) is larger than or equal to the repair rate, and the deterioration and repair rates as well as the weights are homogeneous across all the components, we showed that the policy that targets the component with the largest state value at each time-step is optimal under certain conditions. On the other hand, if the repair rates are sufficiently larger than the deterioration rates, the optimal control policy is to target the component whose state minus the deterioration rate is least in a particular subset of components at each time-step. Finally, we considered the case when the deterioration and repair rates vary with time but are homogeneous across components. We again provided optimal policies for this case as a function of the relationship between the repair rates and deterioration rates. These results illustrate that the above policies are robust to the variations of deterioration and repair rates with time, as long as some conditions between the rates are satisfied at each time-step.

2.7 Publications

The key contributions of this chapter are based on the following publications:

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V. Optimal Repair Policies for Systems Deteriorating After Disruptions. (Under review at IEEE Transactions on Automatic Control)

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V., 2019. Optimal sequencing policies for recovery of physical infrastructure after disasters, American Control Conference (ACC) 2019

3. POLICIES FOR THE RECOVERY OF PHYSICAL COMPONENTS WITH DEPENDENCIES AFTER DISASTERS

3.1 Introduction

In this chapter, we consider dependencies in the form of *precedence constraints* between the physical components. Precedence relations can exist between the physical components in various ways, for e.g., in order to bring back the functionality of water systems (such as irrigation pumps), it might be necessary to first repair some sections of the power network [8]. Precedence constraints can also model any predefined priorities that are associated between different components. For instance, if there are two fire affected regions with one region being densely populated and the other region being sparsely populated, then firefighters would prioritize targeting the former region [86]. We will show in this chapter that the policies that are characterized to be optimal without precedence constraints (i.e., in the previous chapter) need not be optimal when we have precedence constraints. In addition, the general problem with precedence constraints is NP-hard and thus we will provide near-optimal policies for various cases of this problem. Finally, there could be a constraint on the time that is available for repair due to the presence of various deadlines in the real-world. Therefore, we will also consider time constraint in this chapter. We start by formally defining the problem in the next section.

3.2 Problem statement

Note that the problem that we focus in this chapter is an extension of the problem of Section 2.2 except that for simplicity we consider homogeneous weights for all the physical nodes. Thus, we do not repeat the aspects of the problem that are the common with the previous chapter and start explaining how precedence and time constraints are considered in this chapter.

Let $T^* \in \mathbb{N} \cup \{\infty\}$ be the largest time-step that is available for repairing nodes (when $T^* = \infty$ we say that there is no time constraint). Also, there is a set of dependency constraints between different nodes, represented by a set $\mathcal{E} = \{(\mathrm{j}, k) | \mathrm{j}, k \in \mathcal{V}\}$ consisting of edges between various nodes. An edge $(\mathrm{j}, k) \in \mathcal{E}$ starting from node j and ending at node k

represents a precedence constraint that node j needs to be permanently repaired (i.e., to full health) before the entity can start targeting node k; j is an *in-neighbor* of k. The *in-neighbor set* of a node $k \in \mathcal{V}$ is the set $\mathcal{X}_k = \{\mathrm{j} \in \mathcal{V} | (\mathrm{j}, k) \in \mathcal{E}\}$. We assume that the precedence constraints form a directed acyclic graph (DAG) denoted by $G = \{\mathcal{V}, \mathcal{E}\}$; such graphs are a common form for representing general precedence constraints in other problems such as job scheduling [87].

We now define the *feasible control set* as follows.

Definition 3.2.1. *Let the health values of all the nodes at time-step t be given by $v_t = \{v_t^1, \ldots, v_t^N\}$. Then, the feasible control set at time-step t is the set $\mathcal{U}(v_t) \subseteq \mathcal{V}$ containing the nodes whose in-neighbors are all in the permanently repair state at time-step t, i.e., $\mathcal{U}(v_t) = \{k \in \mathcal{V} | v_t^{\mathrm{j}} = 1, \forall \mathrm{j} \in \mathcal{X}_k\}$.*

We now present the definition of a *control sequence that respects the precedence constraints* as follows.

Definition 3.2.2. *A control sequence $\bar{u}_{0:T^*} = \{\bar{u}_0, \bar{u}_1, \ldots, \bar{u}_{T^*}\}$ is said to respect the precedence constraints if for all $t \in [0, T^*]$, $\bar{u}_t \in \mathcal{U}(v_t)$.*

In this chapter, we assume that the weights of physical nodes are homogeneous. That is, $w^{\mathrm{j}} = w, \forall \mathrm{j}$. Then, maximizing the total weight of nodes that are permanently repaired is equivalent to maximizing the number of nodes that are permanently repaired. Thus, the reward function is defined as follows.

Definition 3.2.3. *Given a set of initial health values $\{v_0^{\mathrm{j}}\}$ and a control sequence $\bar{u}_{0:T^*}$ that respects the precedence constraints, the reward $J(v_0, \bar{u}_{0:T^*})$ is defined as the number of nodes that get permanently repaired through that sequence. More formally, $J(v_0, \bar{u}_{0:T^*}) = |\{\mathrm{j} \in \mathcal{V} \mid \exists\, t \text{ s.t. } 0 \leq t \leq T^* \text{ and } v_{t+1}^{\mathrm{j}} = 1\}|$.*

Based on the above definitions, the following problem is the focus of this chapter.

Problem 2 (Optimal Control Sequencing over a DAG).
Consider a directed acyclic graph $G = \{\mathcal{V}, \mathcal{E}\}$ consisting of $N (\geq 2)$ nodes with initial health values $\{v_0^{\mathrm{j}}\}$, along with repair and deterioration rates $\{\Delta_{inc}^{\mathrm{j}}\}$ and $\{\Delta_{dec}^{\mathrm{j}}\}$, respectively. Given

a time constraint $T^* \in \mathbb{N} \cup \{\infty\}$, find a control sequence $u^*_{0:T^*}$ that respects the precedence constraints and maximizes the reward $J(v_0, u^*_{0:T^*})$.

In this chapter, we will be obtaining sequences that are generated by state feedback policies, i.e., we will be providing a mapping $\mu : [0,1]^N \to \mathcal{V}$, such that $\bar{u}_t = \mu(v_t)$ for all $t \in [0, T^*]$. Thus, we will use the same terminology defined above (for sequences) for policies (e.g., a policy respecting the precedence constraints, a non-jumping policy, etc.). Given a policy μ that respects the precedence constraints, we denote the reward (from Definition 3.2.3) by $J_\mu(v_0)$, with the time-constraint being implicit.

We now define an α-optimal policy as follows.

Definition 3.2.4. *Let μ^* be an optimal policy for Problem 2. For $\alpha \in (0,1]$, a policy μ (that respects the precedence and time constraints) is said to be α-optimal if $J_\mu(v_0) \geq \alpha J_{\mu^*}(v_0), \forall v_0 \in [0,1]^N$.*

In the next section, we start the analysis of the problem for $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$.

3.3 Control sequences for $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$

In this section, we start by showing the optimality of non-jumping sequences when $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$.

3.3.1 Optimality of non-jumping sequences

To show that non-jumping policies are optimal when $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$, we first analyze the properties of sequences containing at most one jump and later generalize to sequences containing an arbitrary number of jumps. We start with the following result.

Lemma 14. *Let there be $N (\geq 2)$ nodes that have a set of precedence constraints given by a DAG $G = \{\mathcal{V}, \mathcal{E}\}$, $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $T^* \in \mathbb{N} \cup \{\infty\}$. Consider a control sequence A that respects the precedence and time constraints and targets N nodes as shown in Figure 3.1. Suppose sequence A permanently repairs all the nodes and contains exactly one jump, where the entity partially repairs node i_1 before moving to node i_2 at time-step*

$\bar{t}_1^A$. Sequence A then permanently repairs nodes $\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k$, before returning to node i_1 and permanently repairing it. Then, there exists a non-jumping control sequence B as shown in Figure 3.2 that targets nodes in the order $\{\mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_k, \mathrm{i}_1, \mathrm{i}_{k+1}, \ldots, \mathrm{i}_N\}$ (which respects the precedence and time constraints) and permanently repairs all the nodes. Furthermore, if t_{j}^A (resp. t_{j}^B) is the number of time-steps taken to permanently repair node i_{j} in sequence A (resp. sequence B), then the following holds true:

$$t_{\mathrm{j}}^A \geq t_{\mathrm{j}}^B + \left(2^{\mathrm{j}-2}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{2, \ldots, k\}, \tag{3.1}$$

$$t_1^A \geq t_1^B + (2^{k-1} - 2)\bar{t}_1^A, \tag{3.2}$$

$$t_{\mathrm{j}}^A \geq t_{\mathrm{j}}^B + \left(2^{\mathrm{j}-1} - 2^{\mathrm{j}-k}\right) \bar{t}_1^A \quad \forall \mathrm{j} \in \{k+1, \ldots, N\}. \tag{3.3}$$

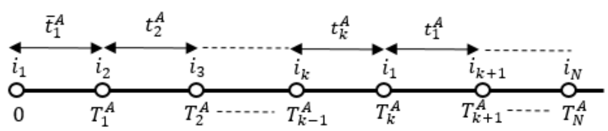

Figure 3.1. Sequence A with a single jump.

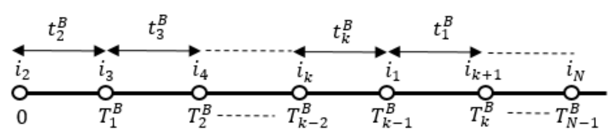

Figure 3.2. Non-jumping sequence B.

Proof. Since node i_1 is partially repaired by the entity before it targets the nodes $\mathrm{i}_2, \ldots, \mathrm{i}_k$ in sequence A, there cannot be an edge that starts from i_1 and ends at a node in the set $\{\mathrm{i}_2, \ldots, \mathrm{i}_k\}$. Thus, if sequence A follows the precedence constraints, the non-jumping sequence B also respects the precedence constraints. We can now leverage the analysis in Lemma 1, which proved that (3.1)-(3.3) hold and each node in sequence B is targeted at an earlier time-step than in sequence A. Therefore, sequence B permanently repairs all the nodes that are permanently repaired by sequence A while respecting the precedence and time constraints. Thus, the result follows. $\square$

The above result considered sequences containing exactly one jump. This leads us to the following key result pertaining to the optimal control policy when $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j}$.

Theorem 3.3.1. *Let there be $N(\geq 2)$ nodes that have a set of precedence constraints given by a DAG $G = \{\mathcal{V}, \mathcal{E}\}$, $\Delta^{\mathrm{j}}_{\mathrm{dec}} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $T^* \in \mathbb{N} \cup \{\infty\}$. Suppose there is a sequence A with one or more jumps that respects the precedence and time constraints, and permanently repairs $x(\leq N)$ nodes. Then, there exists a non-jumping sequence that respects the precedence and time constraints, and permanently repairs x nodes. Thus, non-jumping sequences are optimal when $\Delta^{\mathrm{j}}_{\mathrm{dec}} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$.*

The proof of the above result follows immediately by applying Lemma 14 on sequence A to obtain a sequence that has one less jump than A, and repeating this procedure on the obtained sequences to get a non-jumping sequence.

Since non-jumping sequences are optimal by the above result, we will only focus on non-jumping sequences when $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{\mathrm{inc}}, \forall \mathrm{j} \in \{1, \ldots, N\}$. We next present the NP-hardness result. This proof is inspired from the paper [88]; however, the job scheduling problem in the paper [88] does not consider deterioration of jobs and there are additional differences between the problem that is considered in the paper [88] and Problem 2 as mentioned in the review of job scheduling studies (in Chapter 1). We start by defining the NP-complete *Clique* problem [19] and an instance of a decision version of Problem 2 referred to as OR_d.

Problem 3 (*Clique*)**.** *Given an undirected graph $\overline{G} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ consisting of s vertices and q edges, and a positive integer $p(\leq s)$, does $\overline{G}$ have a complete subgraph of size p, i.e., a set of p vertices such that each pair of vertices in the set is connected by an edge in $\overline{\mathcal{E}}$?*

Problem 4 (OR_d)**.** *Given a directed acyclic graph $G = \{\mathcal{V}, \mathcal{E}\}$ consisting of $N(\geq 2)$ nodes with initial health values $\{v^{\mathrm{j}}_0\}$, along with repair and deterioration rates $\{\Delta^{\mathrm{j}}_{\mathrm{inc}}\}$ and $\{\Delta^{\mathrm{j}}_{\mathrm{dec}}\}$, respectively, $T^* = \infty$, and an integer z such that $0 \leq z \leq N$, is there a non-jumping control sequence $u^*_{0:T^*}$ that respects the precedence constraints and gives a reward $J(v_0, u^*_{0:T^*}) \geq N - z$?*

We now present the NP-hardness result.

Theorem 3.3.2. *Problem 2 is NP-hard.*

Proof. Given an instance of Clique we construct an instance of OR_d as follows. We construct a total of $N = s + q + 1$ nodes that are of three types as follows: a v-node is constructed corresponding to each vertex in $\overline{G}$, an e-node is constructed corresponding to each edge in $\overline{G}$, and a *root* node r is constructed. The parameters of these nodes are set as follows. For each v-node j, set $v_0^{\mathrm{j}} = \frac{2(2^{s+q}-1)+1}{2+2(2^{s+q}-1)+1}$ and $\Delta_{dec}^{\mathrm{j}} = \Delta_{inc}^{\mathrm{j}} = \frac{1}{2+2(2^{s+q}-1)+1}$. For each e-node j, set $v_0^{\mathrm{j}} = \frac{2(2^{\frac{p(p+1)}{2}}-1)+1}{2+2(2^{\frac{p(p+1)}{2}}-1)+1}$ and $\Delta_{dec}^{\mathrm{j}} = \Delta_{inc}^{\mathrm{j}} = \frac{1}{2+2(2^{\frac{p(p+1)}{2}}-1)+1}$. The root node r has the same parameters as a v-node. We construct directed edges such that there is an edge starting from a v-node and ending in an e-node if the vertex corresponding to the v-node lies at one of the ends of the edge (in graph $\overline{G}$) corresponding to the e-node. We also construct directed edges starting from the root node and ending in all the other nodes, so that a DAG is formed. Finally, set $z = q - \frac{p(p-1)}{2}$. Note that the constructed instance of OR_d is polynomial in the size of the given instance of Clique.

We will now prove that the answer to the given instance of Clique is *yes* if and only if the answer to the constructed instance of OR_d is *yes*. Suppose that the answer to the instance of Clique is *yes*. Then, we will show that it is possible to create a non-jumping sequence for the constructed instance of OR_d in which no more than $q - \frac{p(p-1)}{2}$ nodes permanently fail while respecting the precedence constraints. The first node in the created sequence is node r because of the precedence constraints. Note that node r is permanently repaired in the created sequence because the first node in any non-jumping sequence is always permanently repaired (when $T^* = \infty$). After permanently repairing node r, p v-nodes whose corresponding vertices form a complete graph in the Clique are targeted in the created sequence. Note that it takes two time-steps to permanently repair node r in the created sequence because $\frac{1-v_0^r}{\Delta_{dec}^r} = 2$. Also, the health value of each v-node j after two time-steps is equal to $v_2^{\mathrm{j}} = v_0^{\mathrm{j}} - 2\Delta_{dec}^{\mathrm{j}} = \frac{2(2^{s+q}-1)+1-2}{2+2(2^{s+q}-1)+1}$. Thus, the number of time-steps it takes to permanently repair the first v-node j in the created sequence is equal to $\frac{1-v_2^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 4$. Thus, the second v-node j starts getting targeted after six time-steps in the created sequence and at that time its health value is equal to $v_6^{\mathrm{j}} = v_0^{\mathrm{j}} - 6\Delta_{dec}^{\mathrm{j}} = \frac{2(2^{s+q}-1)+1-6}{2+2(2^{s+q}-1)+1}$. Thus, the number of time-steps it takes to permanently repair the second v-node j in the created sequence is equal to $\frac{1-v_6^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 8$. Proceeding in this way, it can be shown that the total number of time-steps taken to permanently repair node r and $p - 1$ v-nodes is equal to $2(2^p - 1)$ because the ith node in the created sequence (where $1 \le \mathrm{i} \le p$) takes 2^{i} time-steps to get permanently repaired. Note that $\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 2\left(2^{s+q} - 1\right) + 1$ for all v-nodes j, and represents the number of time-steps it takes for j to permanently fail if it is not targeted within the first $\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}}$ time-steps. Thus, for all v-nodes j, we define $\gamma_v = \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 2(2^{s+q} - 1) + 1$. Note that the time-step at which the pth v-node starts getting targeted in the created sequence is less than γ_v because $2(2^p - 1) \le 2^{s+1} - 2 < 2^{s+1} - 1 \le 2(2^{s+q} - 1) + 1$ (as $p \le s$ and $q \ge 0$). Thus, all the p v-nodes are permanently repaired in the created sequence. After permanently repairing p v-nodes that correspond to the solution of Clique, $\frac{p(p-1)}{2}$ e-nodes that correspond to the edges of the complete subgraph in the solution of Clique are targeted. Note that $\frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 2(2^{\frac{p(p+1)}{2}} - 1) + 1$ for all e-nodes j. Thus, for all e-nodes j, we define $\gamma_e = \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} = 2(2^{\frac{p(p+1)}{2}} - 1) + 1$. Note that there are $1 + p + \frac{p(p-1)}{2} - 1 = \frac{p(p+1)}{2}$ nodes that are targeted before the last e-node among the aforementioned e-nodes in the created sequence. Thus, the last e-node among the aforementioned e-nodes starts getting targeted in the created sequence at time-step $2(2^{\frac{p(p+1)}{2}} - 1)$ because the ith node in the created sequence (where $1 \le \mathrm{i} \le \frac{p(p+1)}{2}$) takes 2^{i} time-steps to get permanently repaired. Thus, all the aforementioned e-nodes get permanently repaired because $2(2^{\frac{p(p+1)}{2}} - 1) < \gamma_e$. Finally, the remaining $s - p$ v-nodes are targeted in the created sequence. Note that there are $1 + p + \frac{p(p-1)}{2} + s - p - 1 = s + \frac{p(p-1)}{2}$ nodes that are targeted before the last v-node in the created sequence. Thus, it takes $2(2^{s+\frac{p(p-1)}{2}} - 1)$ time-steps in order to start targeting the last v-node because the ith node in the created sequence (where $1 \le \mathrm{i} \le s + \frac{p(p-1)}{2}$) takes 2^{i} time-steps to get permanently repaired. Thus, it is possible to permanently repair all the v-nodes because $2(2^{s+\frac{p(p-1)}{2}} - 1) < \gamma_v$ (as $q \ge \frac{p(p-1)}{2}$). Thus, except the remaining $q - \frac{p(p-1)}{2}$ e-nodes, all the nodes are permanently repaired in the created sequence.

Now we show the opposite direction. Suppose the answer to the constructed instance of OR_d is *yes*. Then, there exists a non-jumping sequence in which at most $z = q - \frac{p(p-1)}{2}$ nodes permanently fail. Note that there are at most $N - 1 = s + q$ nodes that are permanently repaired before the last node that is targeted in the given sequence. Thus, the largest time-step at which the last targeted node in the given non-jumping sequence starts getting targeted is equal to $2(2^{s+q} - 1)$ because the ith node in the given sequence takes 2^{i} time-steps to get permanently repaired. Thus, all the v-nodes are permanently repaired in the given sequence because $2(2^{s+q}-1) < 2(2^{s+q}-1)+1 = \gamma_v$. Note that the first node that is targeted in the given sequence is node r because of the precedence constraints. Since the first node in the given sequence is permanently repaired and all v-nodes are also permanently repaired, the set of all the nodes that permanently fail in the given sequence is a subset of all the e-nodes. The number of e-nodes that are permanently repaired in the given sequence is at least equal to $\frac{p(p-1)}{2}$ because at most $q - \frac{p(p-1)}{2}$ nodes permanently fail in the given sequence and the total number of e-nodes is equal to q. Therefore, at least p v-nodes need to be permanently repaired before at least $\frac{p(p-1)}{2}$ e-nodes can be permanently repaired in the given sequence because of the following two reasons. First, a v-node is an in-neighbor of an e-node if the vertex corresponding to the v-node lies at one of the ends of the edge (in graph $\overline{G}$) corresponding to the e-node. Secondly, among all the undirected graphs that have $\frac{p(p-1)}{2}$ edges, a complete graph (containing p vertices) has the least number of vertices. Note that all the e-nodes that are permanently repaired in the given sequence start getting targeted before time-step $\gamma_e = 2(2^{\frac{p(p+1)}{2}} - 1) + 1$. Also, the maximum number of nodes (when the ith node takes 2^{i} time-steps to get permanently repaired) that can be targeted such that the last node starts getting targeted before time-step $2(2^{\frac{p(p+1)}{2}} - 1) + 1$ is equal to $\frac{p(p+1)}{2} + 1$. Thus, node r, p v-nodes and $\frac{p(p-1)}{2}$ e-nodes are targeted before time-step $2(2^{\frac{p(p+1)}{2}} - 1) + 1$ in the given sequence such that the vertices corresponding to the p v-nodes form a complete subgraph of size p in graph $\overline{G}$.

Since non-jumping sequences are optimal when the precedence constraints are given by a DAG, $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $T^* \in \mathbb{N} \cup \{\infty\}$ by Theorem 3.3.1, we conclude that Problem 2 is NP-hard. $\square$

Since Problem 2 is NP-hard, it is unlikely that the optimal solution can be efficiently computed for all instances of the problem [19]. Thus, we will characterize optimal or near-optimal policies for special instances of the problem.

3.3.2 Near-optimal policy

In this section, we characterize a near-optimal policy for the case when $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$. We will make the following assumption in this section.

Assumption 1. *For all* $\mathrm{j} \in \{1, \ldots, N\}$*, suppose* $\Delta^{\mathrm{j}}_{inc} = \Delta_{\mathrm{inc}}$*,* $\Delta^{\mathrm{j}}_{\mathrm{dec}} = \Delta_{dec}$*, and* $\Delta_{dec} \geq \Delta_{\mathrm{inc}}$*. Also, suppose there exists a positive integer* n *such that* $\Delta_{\mathrm{dec}} = n\Delta_{\mathrm{inc}}$*, and for each node* $\mathrm{j} \in \{1, \ldots, N\}$*, there exists a positive integer* m_{j} *such that* $1 - v^{\mathrm{j}}_0 = m_{\mathrm{j}}\Delta_{inc}$*.*

Note that the conditions $\Delta_{dec} = n\Delta_{inc}$ and $1 - v^{\mathrm{j}}_0 = m_{\mathrm{j}}\Delta_{inc}$, $\forall \mathrm{j} \in \{1, \ldots, N\}$, in the assumption ensure that no node gets permanently repaired partway through a time-step.

We now start with the following result.

Lemma 15. *Let there be* $N(\geq 2)$ *nodes with precedence constraints given by a DAG* $G = \{\mathcal{V}, \mathcal{E}\}$*. Suppose Assumption 1 holds and* $T^* \in \mathbb{N} \cup \{\infty\}$*. Consider a non-jumping sequence* A *that respects the precedence and time constraints, and permanently repairs* $x(\leq N)$ *nodes. Suppose there is a node* h *from the set of* N *nodes that has larger initial health value than the first node of sequence* A *and the precedence constraints allow node* h *to be permanently repaired before the first node of sequence* A*. Consider a non-jumping sequence* B *where node* h *is first targeted and then the remaining targeted nodes of sequence* A *are targeted after node* h*. Then, at least* $x - 1$ *nodes are permanently repaired in sequence* B *while respecting the precedence and time constraints.*

Proof. Let x be the number of nodes that are permanently repaired in sequence A. Denote the order of nodes in the sequence A as $\{\mathrm{i}_1, \ldots, \mathrm{i}_x\}$. The proof for the case when $x = 1$ is trivial because $x - 1 = 0$; we thus focus on the case when $x \geq 2$. Denote the ordering of the first $x - 1$ nodes in sequence B as $\{\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_{x-1}\}$, where $\bar{\mathrm{i}}_1 = h$. Note that the proof when $x = 2$ is also trivial because node $\bar{\mathrm{i}}_1 = h$ in the non-jumping sequence B is permanently repaired since the time taken to permanently repair node h in sequence B is less than the time taken to permanently repair the nodes in sequence A (because node h has larger initial health than node i_1). Therefore, the non-trivial cases are when $x \geq 3$.

We first argue that sequence B permanently repairs at least $x - 1$ nodes, regardless of the time it takes to repair the nodes. We prove the result by contradiction. Suppose sequence B permanently repairs less than $x - 1$ nodes. Let $\bar{\mathrm{i}}_k$ be the first node that permanently fails in sequence B, where $2 \leq k \leq x - 1$. We first argue that either $\bar{\mathrm{i}}_k = \mathrm{i}_{k-1}$ or $\bar{\mathrm{i}}_k = \mathrm{i}_k$. If node h does not appear in sequence A, then the first $x - 1$ nodes of sequence B are ordered as follows $\{h, \mathrm{i}_1, \ldots, \mathrm{i}_{x-2}\}$ and thus $\bar{\mathrm{i}}_k = \mathrm{i}_{k-1}$. On the other hand, if node h is permanently repaired in sequence A, let the position of node h in sequence A be z, where $z \in \{2, \ldots, x\}$. Then, $\bar{\mathrm{i}}_1 = h$, $\bar{\mathrm{i}}_\mathrm{j} = \mathrm{i}_{\mathrm{j}-1}, \quad \forall \mathrm{j} \in \{2, \ldots, z\}$ and $\bar{\mathrm{i}}_\mathrm{j} = \mathrm{i}_\mathrm{j}, \quad \forall \mathrm{j} \in \{z+1, \ldots, x-1\}$. Thus, if the first node that fails in sequence B (i.e., node $\bar{\mathrm{i}}_k$) is such that $k \leq z$, then $\bar{\mathrm{i}}_k = \mathrm{i}_{k-1}$; otherwise $\bar{\mathrm{i}}_k = \mathrm{i}_k$. We now consider the cases when $\bar{\mathrm{i}}_k = \mathrm{i}_{k-1}$ and $\bar{\mathrm{i}}_k = \mathrm{i}_k$ one by one.

Suppose $\bar{\mathrm{i}}_k = \mathrm{i}_{k-1}$. If node i_{k-1} fails in sequence B, then we show that node i_{k+1} must permanently fail in sequence A (where $2 \leq k \leq x-1$). Consider $k = 2$. Let $t_h = \frac{1-v_0^h}{\Delta_{inc}}$ be the number of time-steps taken to permanently repair node h from its initial health to the full health. If node i_1 permanently fails after permanently repairing node h in sequence B, then $v_0^{\mathrm{i}_1} \leq \Delta_{dec} t_h$. That is,

$$1 - v_0^{\mathrm{i}_1} \geq 1 - \Delta_{dec} t_h. \tag{3.4}$$

Denote $t_\mathrm{j} = \frac{1-v_0^{\mathrm{i}_\mathrm{j}}}{\Delta_{inc}}$ as the number of time-steps taken to permanently repair node i_j from its initial health to full health. Then,

$$\Delta_{dec} t_1 = \Delta_{dec} \left(\frac{1 - v_0^{\mathrm{i}_1}}{\Delta_{inc}} \right) = n \left(1 - v_0^{\mathrm{i}_1}\right) \geq \left(1 - v_0^{\mathrm{i}_1}\right), \tag{3.5}$$

as $n \geq 1$ and $v_0^{\mathrm{i}_1} < 1$. Therefore,

$$\Delta_{dec} t_1 \geq 1 - \Delta_{dec} t_h > 1 - \Delta_{dec} t_1, \tag{3.6}$$

where the first inequality from the left comes by conditions (3.4)-(3.5) and the second inequality comes from the fact that $t_h < t_1$ (as node h has larger initial health than node i_1). Note that in sequence A, it takes t_1 time-steps to repair node i_1, and $nt_1 + t_2$ time-steps to repair node i_2 (it takes nt_1 time-steps to repair the health that is lost by node i_2 due to deterioration and it takes t_2 time-steps to repair the difference in the initial health of i_2 and full health). By (3.6), the total reduction in the health of node i_3 after $(1+n)t_1 + t_2$ time-steps in sequence A satisfies

$$\begin{aligned}
\Delta_{dec} \left((1+n)t_1 + t_2\right) &= \Delta_{dec} \left(nt_1 + t_2\right) + \Delta_{dec} t_1 \\
&> \Delta_{dec} \left(nt_1 + t_2\right) + 1 - \Delta_{dec} t_1 \\
&> 1,
\end{aligned}$$

as $n \geq 1$ and $t_1, t_2 > 0$. Thus, node i_3 permanently fails in sequence A by the time it is reached, contradicting the fact that A permanently repairs $x(\geq 3)$ nodes.

Now consider the case when $k = 3$. If node i_2 permanently fails in sequence B, then $v_0^{\mathrm{i}_2} \leq \Delta_{dec}((1+n)t_h + t_1)$. Following the same arguments as before, we get

$$\Delta_{dec} t_2 > 1 - \Delta_{dec}\left((2+n)t_1\right). \tag{3.7}$$

As before, the total number of time-steps required to permanently repair nodes $\mathrm{i}_1, \mathrm{i}_2$, and i_3 in sequence A is equal to $(1+n)^2 t_1 + (1+n)t_2 + t_3$. Thus, the total reduction in the health of node i_4 by the time it is reached in sequence A satisfies

$$\Delta_{dec}\Big((1+n)^2 t_1 + (1+n)t_2 + t_3\Big) > 1,$$

by (3.7), $n \geq 1$ and $t_1, t_2, t_3 > 0$. Thus, i_4 permanently fails by the time it is reached in sequence A, leading to a contradiction.

We can repeat the above arguments to show that if node i_{k-1} permanently fails in sequence B, where $k > 3$, then

$$\Delta_{dec} t_{k-1} > 1 - \Delta_{dec}((2+n)(1+n)^{k-3} t_1 + (1+n)^{k-4} t_2 + \ldots + (1+n)^0 t_{k-2}). \tag{3.8}$$

Therefore, the total reduction in the health of node i_{k+1} after $(1+n)^{k-1} t_1 + (1+n)^{k-2} t_2 + \ldots + (1+n)^0 t_k$ time-steps in sequence A is

$$\begin{aligned} &\Delta_{dec}\Big((1+n)^{k-1} t_1 + \ldots + (1+n)^1 t_{k-1} + (1+n)^0 t_k\Big) = \\ &\Delta_{dec}\Big((1+n)^{k-1} t_1 + \ldots + n t_{k-1} + (1+n)^0 t_k\Big) + \Delta_{dec} t_{k-1}. \end{aligned} \tag{3.9}$$

Thus, for $k > 3$, we get

$$\Delta_{dec}\Big((1+n)^{k-1} t_1 + \ldots + (1+n)^1 t_{k-1} + (1+n)^0 t_k\Big) > 1$$

by (3.8), (3.9), $n \geq 1$ and $t_{\mathrm{j}} > 0\, \forall \mathrm{j}$. Since, the total reduction in the health value of node i_{k+1} before the entity starts targeting it is larger than one, it is not possible to permanently repair node i_{k+1} in sequence A, leading to a contradiction.

We now consider the case when $\bar{\mathrm{i}}_k = \mathrm{i}_k$ (along with $x \geq 3$). We prove that if node $\bar{\mathrm{i}}_k = \mathrm{i}_k$ permanently fails in sequence B, then it is not possible to permanently repair node $\bar{\mathrm{i}}_{k+1} = \mathrm{i}_{k+1}$ in sequence A. Recall that the case $\bar{\mathrm{i}}_k = \mathrm{i}_k$ happens when node h is permanently repaired in sequence A and $k > z$. Since $z \geq 2$, we have $k > 2$. Consider $k = 3$. Then, $z = 2$ (i.e., $\mathrm{i}_2 = h$) because $z < k$ and $z \geq 2$. If node $\bar{\mathrm{i}}_3 = \mathrm{i}_3$ permanently fails in sequence B, then

$$v_0^{\mathrm{i}_3} \leq \Delta_{dec}((1+n)t_h + t_1) < \Delta_{dec}\left((2+n)t_1\right),$$

or equivalently,

$$\Delta_{dec} t_3 > 1 - \Delta_{dec}\left((2+n)t_1\right). \tag{3.10}$$

Therefore, the total reduction in the health of node i_4 after $(1+n)^2 t_1 + (1+n)t_2 + t_3$ time-steps in A satisfies

$$\begin{aligned}
&\Delta_{dec}\left((1+n)^2 t_1 + (1+n)t_2 + t_3\right) \\
&= \Delta_{dec}\left((1+n)^2 t_1 + (1+n)t_2\right) + \Delta_{dec} t_3 > 1,
\end{aligned}$$

by (3.10), $n \geq 1$ and $t_1, t_2 > 0$. Thus, node i_4 permanently fails in sequence A, contradicting the fact that A permanently repairs x nodes.

Similarly, it can be shown that if node $\bar{\mathrm{i}}_k = \mathrm{i}_k$ permanently fails in sequence B, where $k > 3$, then

$$\begin{aligned}
\Delta_{dec} t_k > 1 - \Delta_{dec}((2+n)(1+n)^{k-3} t_1 + (1+n)^{k-4} t_2 + \ldots + (1+n)^{k-1-z} t_{z-1} + \\
(1+n)^{k-2-z} t_{z+1} + \ldots + (1+n)^0 t_{k-1}).
\end{aligned} \tag{3.11}$$

Therefore, the total reduction in the health of node i_{k+1} (where $k > 3$) after $(1+n)^{k-1}t_1 + (1+n)^{k-2}t_2 + \ldots + (1+n)^0 t_k$ time-steps in sequence A is

$$\begin{aligned}&\Delta_{dec}\left((1+n)^{k-1}t_1 + \ldots + (1+n)^1 t_{k-1} + (1+n)^0 t_k\right)\\&= \Delta_{dec}\left((1+n)^{k-1}t_1 + \ldots + (1+n)^1 t_{k-1}\right) + \Delta_{dec} t_k > 1,\end{aligned}$$

by (3.11), $n \geq 1$ and $t_j > 0\ \forall j$. Therefore, it would not be possible to permanently repair x nodes in sequence A, leading to a contradiction.

We will now show that sequence B permanently repairs $x-1$ nodes in less time than the time taken to permanently repair x nodes in sequence A. Suppose node h is not permanently repaired in sequence A. Consider $x = 3$. Then, the time taken to permanently repair the first two nodes, i.e., nodes h and i_1, in sequence B is equal to $(1+n)t_h + t_1$ and the time taken to permanently repair three nodes in sequence A is equal to $(1+n)^2 t_1 + (1+n)t_2 + t_3$. Note that $(1+n)t_h + t_1 < (2+n)t_1 < (1+n)^2 t_1 + (1+n)t_2 + t_3$ because $t_h < t_1$, $n \geq 1$ and $t_1, t_2, t_3 > 0$. Consider the case when $x > 3$. Then, the time taken to permanently repair $x-1$ nodes in sequence B is equal to

$$(1+n)^{x-2}t_h + (1+n)^{x-3}t_1 + \ldots + (1+n)^0 t_{x-2}, \tag{3.12}$$

and the time taken to permanently repair x nodes in sequence A is equal to

$$(1+n)^{x-1}t_1 + (1+n)^{x-2}t_2 + \ldots + (1+n)^0 t_x. \tag{3.13}$$

Then, $(1+n)^{x-2}t_h + (1+n)^{x-3}t_1 + (1+n)^{x-4}t_2 + \ldots + (1+n)^0 t_{x-2} < \left((2+n)(1+n)^{x-3}\right) t_1 + (1+n)^{x-4}t_2 + \ldots + (1+n)^0 t_{x-2} < (1+n)^{x-1}t_1 + (1+n)^{x-2}t_2 + \ldots + (1+n)^0 t_x$ as $t_h < t_1$, $n \geq 1$ and $t_j > 0, \forall j \in \{1, \ldots, x\}$.

Now consider the case when node h is permanently repaired in sequence A. When $z \geq x-1$ (and $x \geq 3$), the time taken to permanently repair $x-1$ nodes in sequence B and the time taken to permanently repair x nodes in sequence A are given by expressions (3.12) and (3.13), respectively, and therefore the proof proceeds in the same way as when node h is not permanently repaired in sequence A. Thus, the proof for $x = 3$ follows in exactly the same way as before because $z \geq 2$. We now focus on the case when $z \leq x-2$ and $x > 3$. Then, the total time taken to permanently repair $x-1$ nodes in sequence B is equal to

$$(1+n)^{x-2}t_h + (1+n)^{x-3}t_1 + \ldots + (1+n)^{x-1-z}t_{z-1} + \\ (1+n)^{x-2-z}t_{z+1} + \ldots + (1+n)^0 t_{x-1}, \quad (3.14)$$

and the time taken to permanently repair x nodes in sequence A is given by the expression (3.13). Note that the value of the expression (3.14) is less than the value of the expression (3.13) because $t_h < t_1$, $n \geq 1$ and $t_j > 0, \forall j \in \{1, \ldots, x\}$. Therefore, if sequence A permanently repairs x nodes by time-step T^*, then sequence B must permanently repair at least $x-1$ nodes by time-step T^*. Thus, the result follows. □

Using the above result, we now propose a near-optimal policy for a subclass of Problem 2.

Theorem 3.3.3. *Let there be $N(\geq 2)$ nodes with precedence constraints given by a DAG $G = \{\mathcal{V}, \mathcal{E}\}$. Suppose Assumption 1 holds and $T^* \in \mathbb{N} \cup \{\infty\}$. Then, the policy that targets the healthiest node at each time-step while respecting the precedence and time constraints is* 1/2*-optimal.*

Proof. Let A be the optimal (non-jumping) sequence. Then, we go to the first time-step $\overline{T}(\leq T^*)$ in sequence A at which the healthiest available node (i.e., the healthiest node that can be targeted at time-step $\overline{T}$ without violating the precedence constraints) is not targeted. Denote the portion of sequence A from time-step 0 to time-step $\overline{T}-1$ as $\overline{A}$ and the portion of sequence from time-step $\overline{T}$ onwards as $\underline{A}$. We modify the sequence $\underline{A}$ by Lemma 15 to generate sequence A^* such that the healthiest available node (while respecting the precedence and time constraints) is targeted at time-step $\overline{T}$ in A^*. Let y be the number of nodes that are permanently repaired by sequence $\underline{A}$. Then, at least $y-1$ nodes are permanently repaired in sequence A^* (while respecting the precedence and time constraints) by Lemma 15. Then, a sequence B is generated by concatenating $\overline{A}$ with A^*. After this, we go to the next time-step in sequence B at which the healthiest node (while respecting the precedence and time constraints) is not targeted and repeat this procedure. We iteratively repeat this procedure, so that we finally get a sequence in which the healthiest node (while respecting the precedence and time constraints) is targeted at each time-step. The number of nodes that are permanently repaired in the final sequence will be at least half of the number of nodes that are permanently repaired in sequence A because 1) at each iteration of this procedure we move one node across the given sequence and the number of nodes that are permanently repaired in the modified sequence reduces at most by one, and 2) in the last iteration of this procedure where there is only one node, there is no reduction in the number of permanently repaired nodes because if the last node in the given sequence can be permanently repaired then a healthier node that replaces it can also be permanently repaired. $\square$

We now show that the factor of $1/2$ in the above result is sharp, in that there exist problem instances where targeting the healthiest node at each time-step repairs only half as many nodes as an optimal policy.

Example 8. *Consider a graph consisting of three nodes as shown in Figure 3.3. Let the initial health values of the nodes 1, 2 and 3 be 0.6, 0.3 and 0.8, respectively. The deterioration and repair rates are homogeneous across all the nodes and both are equal to 0.1. Suppose $T^* = \infty$. The healthiest node that can be targeted in the first time-step is node 1 (as we need to permanently repair node 2 before targeting node 3). If node 1 is first permanently repaired then node 2 fails by the time the entity reaches it and we cannot permanently repair any more nodes. However, if node 2 is first permanently repaired then node 3 can also be permanently repaired. Note that although the policy that targets the healthiest node at each time-step (while respecting the precedence and time constraints) is not optimal in this example, it is indeed* $1/2$*-optimal as proved in Theorem 3.3.3.*

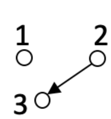

Figure 3.3. Graph for illustrating optimal policies in the presence of precedence constraints.

Remark 6. *In this chapter, we consider all the nodes to be equally important. However, one could also consider weights on the nodes to signify their relative importance, and the objective of Problem 2 can be changed to maximize the total weight of the nodes that are permanently repaired. Then, it is easy to argue that the policy characterized in Theorem 3.3.3 would be* $\frac{w_{min}}{2w_{max}}$*-optimal, where* w_{max} *and* w_{min} *are the maximum and minimum weights of nodes, respectively.*

Theorem 3.3.3 shows that the feedback policy that targets the healthiest node at each time-step while respecting the precedence and time constraints is 1/2-optimal. In the next section, we show that this policy is, in fact, optimal for special instances of the problem.

3.3.3 Optimal sequencing

The previous chapter showed that the policy that targets the healthiest node at each time-step is optimal when all the conditions of Theorem 3.3.3 are satisfied but when there are no precedence and time constraints. We now prove that the aforementioned policy is also optimal when there is a time constraint but there are no precedence constraints.

Theorem 3.3.4. *Let there be* $N(\geq 2)$ *nodes such that there are no precedence constraints (i.e.,* $\mathcal{E} = \emptyset$*). Suppose Assumption 1 holds and* $T^* \in \mathbb{N} \cup \{\infty\}$*. Then, the policy that targets the healthiest node at each time-step while respecting the time constraint is optimal.*

Proof. Consider any optimal (non-jumping) sequence $A = \{\mathrm{i}_1, \ldots, \mathrm{i}_x\}$ that permanently repairs $x(\leq N)$ nodes. Let $\overline{T}(\leq T^* + 1)$ be the total number of time-steps taken by the sequence A to permanently repair all x nodes.[1] Then,

$$\overline{T} = \frac{1 - Z_1}{\Delta_{inc}} + \frac{1 - Z_2}{\Delta_{inc}} + \ldots + \frac{1 - Z_x}{\Delta_{inc}}, \tag{3.15}$$

where Z_j is the health of node $\mathrm{i_j}$ when it is reached in the sequence, i.e., $Z_1 = v_0^{\mathrm{i}_1}$ and $Z_k = v_0^{\mathrm{i}_k} - n\sum_{\mathrm{j}=2}^{k}(1 - Z_{\mathrm{j}-1})$ for $k \in \{2, \ldots, x\}$. Alternatively, $Z_k = v_0^{\mathrm{i}_1} n\,(1+n)^{k-2} + v_0^{\mathrm{i}_2} n(1+n)^{k-3} + \ldots + v_0^{\mathrm{i}_{k-1}} n + v_0^{\mathrm{i}_k} - n\,(1+n)^{k-2} - n(1+n)^{k-3} - \ldots - n$ for all $k \in \{2, \ldots, x\}$.

[1]↑Note that the total number of time-steps is one larger than the largest time-step index T^* because the first time-step in a sequence is time-step 0.

Note that Z_1 is the largest when node i_1 has the largest initial health; Z_2 is the largest when node i_1 has the largest initial health and node i_2 has the second largest initial health (as the coefficients of $v_0^{\mathrm{i}_1}$ and $v_0^{\mathrm{i}_2}$ are n and 1, respectively); Z_3 is the largest when node i_1 has the largest initial health, node i_2 has the second largest initial health and node i_3 has the third largest initial health (as the coefficients of $v_0^{\mathrm{i}_1}$, $v_0^{\mathrm{i}_2}$ and $v_0^{\mathrm{i}_3}$ are $n(1+n)$, n and 1, respectively); and so on. Thus, the non-jumping sequence B that targets the nodes in decreasing order of their initial health is optimal when time is constrained because it permanently repairs x nodes in less time in comparison to the time taken by sequence A to permanently repair x nodes (as the value of $\overline{T}$ in (3.15) for sequence B is less than the corresponding value for sequence A). Therefore, the policy that targets the healthiest node at each time-step while respecting the time constraint is optimal by Theorem 3.3.1. □

We also show that the policy of targeting the healthiest node at each time-step (while respecting the precedence and time constraints) is optimal in certain classes of DAGs. We start with the following definition.

Definition 3.3.1. *Given a DAG, the nodes that do not have any incoming edges are said to belong to the "first level"; after removing all the nodes in the first level and all the outgoing edges of the nodes in the first level, the nodes that do not have any incoming edges are said to belong to the "second level", and so on. A DAG in which all the nodes in a level have incoming edges from all the nodes in the previous level is termed as a "complete series graph".*

Figure 3.4 shows a *complete series* graph that has three levels.

Proposition 3.3.1. *Let there be $N(\geq 2)$ nodes with precedence constraints given by a complete series graph $G = \{\mathcal{V}, \mathcal{E}\}$. Suppose Assumption 1 holds and $T^* \in \mathbb{N} \cup \{\infty\}$. Then, the policy that targets the healthiest node at each time-step while respecting the precedence and time constraints is optimal.*

Proof. Due to precedence constraints, we need to permanently repair all the nodes in the first level before targeting the other nodes in a *complete series* graph. Since the optimal policy is to target the healthiest node at each time-step (while respecting the time constraint) when there are no precedence constraints (by Theorem 3.3.4), we follow this policy for the nodes in the first level. After permanently repairing all the nodes in the first level, we need to permanently repair all the nodes in the second level before targeting other nodes. Thus, the policy of targeting the healthiest node at each time-step (while respecting the time constraint) in the second level is followed after permanently repairing all the nodes in the first level. The rest of the proof follows the above argument. □

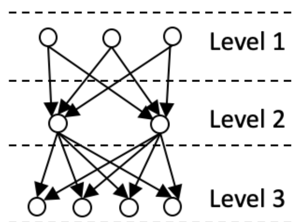

Figure 3.4. An example of a *complete series* graph.

So far, we assumed that $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \ldots, N\}$. In the next section, we analyze other cases.

3.4 Control sequences for $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$

We now consider the case when repair rates are sufficiently larger than the deterioration rates, i.e., $\Delta^{\mathrm{j}}_{inc} > (N-1)\Delta^{\mathrm{j}}_{dec}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $\Delta^{\mathrm{j}}_{inc} > \sum_{l \neq \mathrm{j}} \Delta^{l}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$. We will use the concept of a *modified health value* defined as follows.

Definition 3.4.1. *The modified health value of a node at a time-step is the health value of a node at that time-step minus its deterioration rate.*

3.4.1 Near-optimal policy

Consider a directed rooted tree where the total number of nodes is at most equal to $k(\geq 1)$. In the context of disaster recovery, such a tree can represent damaged infrastructure components (with dependencies) within a particular neighborhood. We will consider such trees for the representation of precedence constraints as follows.

Theorem 3.4.1. *Let there be $N(\geq 2)$ nodes with precedence constraints represented by a graph $G = \{\mathcal{V}, \mathcal{E}\}$, which is a set of disjoint rooted trees that each contain at most $k(\geq 1)$ nodes. Suppose $\Delta^{\mathrm{j}}_{\mathrm{inc}} > (N-1)\Delta^{\mathrm{j}}_{\mathrm{dec}}, \forall \mathrm{j} \in \{1, \ldots, N\}$, $\Delta^{\mathrm{j}}_{inc} > \sum_{l \neq \mathrm{j}} \Delta^{l}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$ and $T^* = \infty$. Then, the policy that targets the node with the least modified health value at each time-step while respecting the precedence constraints is $\frac{1}{k}$-optimal.*

Proof. Let $\mathcal{R}$ be the set of root nodes of the trees. Let A denote the sequence that targets the node with the least modified health value in the set $\mathcal{R}$ at each time-step, and let B denote the optimal sequence. Let the number of nodes that are permanently repaired by sequences A and B be y and x, respectively. We will show that $y \geq \frac{x}{k}$. We prove this by contradiction. Suppose $\frac{x}{k} > y$. Denote the set of root nodes that are permanently repaired in sequence B by $\mathcal{F} \subseteq \mathcal{R}$, and let $z \triangleq |\mathcal{F}|$. Note that $z \geq \left\lceil \frac{x}{k} \right\rceil$ because the root node in a tree should be permanently repaired before targeting other nodes in the tree. Then, $z \geq y+1$ because $\frac{x}{k} > y$ implies $\left\lceil \frac{x}{k} \right\rceil \geq y+1$. We now construct a sequence C by modifying sequence B such that at every time-step of sequence B where a node not belonging to the set $\mathcal{F}$ is targeted, we replace that node with a node from the set $\mathcal{F}$. Then, sequence C also permanently repairs all the nodes in the set $\mathcal{F}$ because the nodes of set $\mathcal{F}$ are targeted more frequently in sequence C than in sequence B. Since the optimal policy is to target the node with the least modified health value at each time-step when there are no precedence and time constraints (by Theorem 2.4.2), we obtain a contradiction because sequence C permanently repairs $z \geq y+1$ nodes whereas sequence A (which is the optimal sequence to permanently repair the set of root nodes) permanently repairs y nodes. Therefore, the assumption that $\frac{x}{k} > y$ is not true.

Let the sequence that targets the node with the least modified health value (in the set $\mathcal{V}$) at each time-step while respecting the precedence constraints be denoted by D. We now argue that sequence D permanently repairs at least as many nodes as sequence A. Denote the set of root nodes that are permanently repaired in sequence A by $\mathcal{Z} \subseteq \mathcal{R}$ and the number of nodes in the set $\mathcal{Z}$ by y. Then, there exists an ordering of nodes in the set $\mathcal{Z}$, denoted by $\{\mathrm{i}_1, \mathrm{i}_2, \ldots, \mathrm{i}_y\}$, such that the initial health values of those nodes satisfy

$$v_0^{\mathrm{i_j}} > (y - \mathrm{j})\Delta^{\mathrm{i_j}}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, y\} \tag{3.16}$$

because of the following argument. Let $\mathcal{A}_t \subseteq \mathcal{Z}$ denote the set of nodes that have not been targeted at least once by the entity prior to time t in sequence A. Note that $\mathcal{Z} = \mathcal{A}_0 \supseteq \mathcal{A}_1 \supseteq \ldots \supseteq \mathcal{A}_{y-1}$. At time $t = 0$, $|\mathcal{A}_t| = y$, where $|\mathcal{A}_t|$ denotes the cardinality of set $\mathcal{A}_t$. At $t = 1$, $|\mathcal{A}_t| = y - 1$ as there are $y - 1$ nodes of the set $\mathcal{Z}$ that have not been targeted at least once by the entity in sequence A and thus each node $k \in \mathcal{A}_1$ should have initial health value larger than Δ^k_{dec} to survive until $t = 1$. At $t = 2$, $|\mathcal{A}_t| \geq y - 2$ as there are at least $y - 2$ nodes of the set $\mathcal{Z}$ that have not been targeted at least once by the entity in sequence A and thus each node $k \in \mathcal{A}_2$ should have initial health value larger than $2\Delta^k_{\text{dec}}$. Repeating this argument for the next $y - 3$ time-steps, we can argue that there must be a permutation $\{\mathrm{i}_1, \ldots, \mathrm{i}_y\}$ of nodes that satisfies the conditions (3.16) in order for y nodes to be permanently repaired in sequence A.

After this, it can be argued that sequence D permanently repairs at least y nodes by the same way as in Theorem 2.4.2. Let the number of nodes that are permanently repaired by sequence D be $\overline{y}$. Then, $\overline{y} \geq y$ and therefore $\frac{x}{\overline{y}} \leq \frac{x}{y} \leq k$. Thus, the policy that targets the node with the least modified health value at each time-step while respecting the precedence constraints is $\frac{1}{k}$-optimal. $\square$

We now show that the factor of $\frac{1}{k}$ in Theorem 3.4.1 is sharp, in that there exist problem instances where targeting the node with the least modified health value at each time-step repairs only $\frac{1}{k}$ times the nodes as an optimal policy.

Example 9. *Consider the graph in Figure 3.3, which is a set of disjoint rooted trees that each contain at most two nodes. Let the initial health values of the nodes 1, 2 and 3 be 0.01, 0.02 and 0.8, respectively, and $T^* = \infty$. The deterioration and repair rates are homogeneous across all the nodes and are equal to 0.1 and 0.25, respectively. The least healthy node that can be targeted in the first time-step is node 1 (since deterioration rates are homogeneous across all the nodes, targeting the node with the least modified health value is equivalent to targeting the node with least health value). However, if node 1 is targeted in the first time-step then node 2 fails by the time the entity reaches it and we cannot target any more nodes. However, if node 2 is first permanently repaired then node 3 can also be permanently repaired. Note that although the policy that targets the least healthy node at each time-step while respecting* *the precedence constraints is not optimal in this example, it is indeed $\frac{1}{k}$-optimal as proved in Theorem 3.4.1.*

We now also provide an example to show that the policy that is characterized in Theorem 3.4.1 need not be $\frac{1}{k}$-optimal when $T^* \in \mathbb{N}$.

Example 10. *Consider a graph consisting of two nodes without any precedence constraints (i.e., the graph is a union of disjoint trees that each contain one node). Let the initial health values of the nodes 1 and 2 be 0.01 and 0.11, respectively. The deterioration and repair rates are homogeneous across all the nodes and are equal to 0.1 and 0.11, respectively, and $T^* = 10$. If the policy of targeting the node with the least health value at each time-step is followed then no node gets permanently repaired while respecting the time constraint. However, if the non-jumping policy that first targets node 2 is followed, then node 2 can be permanently repaired. Thus, the policy that is characterized in Theorem 3.4.1 is not a 1-optimal policy (or optimal policy) in this example.*

Therefore, characterizing near-optimal policies for this case under a time constraint is an avenue for future research.

3.4.2 Optimal sequencing

In the last chapter, we proved that the policy that targets the node with the least modified health value at each time-step is optimal when there are no precedence and time constraints. Examples 9 and 10 showed that this need not be true when there is a precedence or time constraint. However, the policy of targeting the node with the least modified health value at each time-step is optimal for special cases such as when the precedence constraints are given by a *complete series* graph (defined in the previous section).

Proposition 3.4.1. *Let there be $N(\geq 2)$ nodes with precedence constraints given by a complete series graph $G = \{\mathcal{V}, \mathcal{E}\}$. Suppose $\Delta^{\mathrm{j}}_{inc} > (N-1)\Delta^{\mathrm{j}}_{dec}, \forall \mathrm{j} \in \{1, \ldots, N\}$, $\Delta^{\mathrm{j}}_{inc} > \sum_{l \neq \mathrm{j}} \Delta^{l}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$ and $T^* = \infty$. The optimal policy is to target the node with the least modified health at each time-step while respecting the precedence constraints.*

This result can be proved similarly as Proposition 3.3.1 by using the fact that the policy of targeting the node with the least modified health at each step is optimal when there are no precedence and time constraints.

3.5 Conclusions

In this chapter, we studied the problem of finding (near-)optimal control policies for targeting different components whose states are disrupted, when there exist precedence constraints between the components and a constraint on the time that is available for repairing components. We proved that the general problem is NP-hard, and therefore we characterized near-optimal control policies for special instances of the problem. We showed that when the deterioration rates are larger than or equal to the repair rates and the precedence constraints are given by a DAG, it is optimal to continue repairing a component until its state reaches the fully recovered state before switching to repair any other component. Under the aforementioned assumptions and when the deterioration and the repair rates are homogeneous across all the components, we proved that the control policy that targets the healthiest component at each time-step while respecting the precedence and time constraints fully repairs at least half the number of components that would be fully repaired by an optimal policy. Finally, we proved that when the repair rates are sufficiently larger than the deterioration rates, the precedence constraints are given by a set of disjoint trees that each contain at most k nodes, and there is no time constraint, the policy that targets the component with the least value of health minus the deterioration rate at each time-step while respecting the precedence constraints fully repairs at least $1/k$ times the number of components that would be fully repaired by an optimal policy.

3.6 Publications

The key contributions of this chapter are based on the following publications:

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V. Control Policies for Recovery of Interdependent Systems After Disruptions. (Under review at IEEE Transactions on Control of Network Systems)

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V., 2019. Approximation algorithms for the recovery of infrastructure after disasters under precedence constraints, IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS) 2019

4. POLICIES FOR MULTI-AGENCY RECOVERY OF PHYSICAL COMPONENTS AFTER DISASTERS

4.1 Introduction

In this chapter, we focus on the case when there are multiple agencies available for repair after disasters [80]. Specifically, there is an authority whose task is to allocate the components to the agencies within a given budget, so that the total number of components that are fully repaired by the agencies is maximized. This problem represents the scenarios where the government and other emergency management agencies have a limited budget in order to pay smaller agencies or teams to repair the infrastructure components [81], [89]. A naive solution to this problem would be to allocate the largest possible set of components within the budget constraint to the lowest cost agency; however, this agency may not be able to repair all the allocated components. Thus, we need to come up with strategic ways of allocating to the components to different repair agencies, which is the main contribution of this chapter. We start by formally describing the problem in the next section.

4.2 Problem statement

Consider a scenario where there are $N(\geq 2)$ nodes belonging to the set $\mathcal{V}$. For simplicity, we do not consider precedence constraints between the nodes in this chapter. Also, note that we do not repeat the definitions of terms like health values, permanent failure and permanent repair for nodes that are the same as in Chapter 2. There are $Q(\geq 1)$ repair agencies that are referred to as *entities*. The set of all entities is represented by the set $\mathcal{T}$. We assume that time progresses in discrete steps representing the resolution at which control actions are taken. Note that there are two types of control actions. The first type of control action is the *allocation* of nodes to entities. There is an *authority* that allocates the nodes to entities at time-step 0. Note that there is an *allocation constraint* on the set of nodes that each entity can be allocated. Mathematically, each entity $h \in \mathcal{T}$ can only be allocated a subset from the set of nodes $\mathcal{Z}_h \subseteq \mathcal{V}$ such that $\mathcal{Z}_1 \cup \ldots \cup \mathcal{Z}_Q = \mathcal{V}$. In the real-world, this constraint could represent scenarios such as a utility company not being equipped to repair damaged roads, or a repair agency not having the resources to target infrastructure components outside its region of focus, and so on. Let $\mathcal{U}_h \subseteq \mathcal{Z}_h$ be the set of nodes that is allocated to entity h (note that for each pair of entities $\mathrm{i} \in \mathcal{T}$ and $\mathrm{j} \in \mathcal{T}$, $\mathcal{U}_\mathrm{i} \cap \mathcal{U}_\mathrm{j} = \emptyset$). Then, the vector $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_Q\}$ is referred to as an allocation of nodes. The second type of control action refers to the sequences that are followed by the entities to repair the allocated nodes. Let $u_t^h \in \mathcal{U}_h$ be the node that is targeted by entity h at time-step t. Then, $u_{0:\infty}^h = \{u_0^h, u_1^h, \ldots\}$ represents the repair sequence that is followed by entity h.

If node $\mathrm{j} \in \mathcal{U}_h$ is targeted by entity h at a given time-step then its health value increases by an amount $\Delta_{inc}^{\mathrm{j},h}$ (given that node j is not in permanent failure or permanent repair state at that time-step). If node $\mathrm{j} \in \mathcal{V}$ is not targeted by any entity at a given time-step then its health value decreases by an amount $\Delta_{dec}^{\mathrm{j}}$ (given that node j is not in permanent failure or permanent repair state at that time-step). Therefore, for each node $\mathrm{j} \in \mathcal{V}$, the dynamics of its health value are given by

$$v_{t+1}^{\mathrm{j}} = \begin{cases} 1 & \text{if } v_t^{\mathrm{j}} = 1, \\ 0 & \text{if } v_t^{\mathrm{j}} = 0, \\ \min(1, v_t^{\mathrm{j}} + \Delta_{inc}^{\mathrm{j},h}) & \text{if } \mathrm{j} \in \mathcal{U}_h, u_t^h = \mathrm{j}, \text{and } v_t^{\mathrm{j}} \in (0,1), \\ \max(0, v_t^{\mathrm{j}} - \Delta_{dec}^{\mathrm{j}}) & \text{if } u_t^h \neq \mathrm{j}, \forall h \in \mathcal{T}, \text{and } v_t^{\mathrm{j}} \in (0,1). \end{cases} \tag{4.1}$$

Each entity $h \in \mathcal{T}$ charges a cost $c_h^{\mathrm{i}} \in \mathbb{R}_{\geq 0}$ if it is allocated $\mathrm{i}(\geq 1)$ nodes to the authority. Thus, for all $h \in \mathcal{T}, \mathrm{i} \geq 1$, we have $c_h^{\mathrm{i}+1} \geq c_h^{\mathrm{i}}$ because an entity would require additional resources to repair a larger set as compared to a smaller set. Also, we make the following assumption regarding the costs.

Assumption 2. *For all $h \in \mathcal{T}, \mathrm{i} \geq 1$, we have $c_h^{\mathrm{i}+1} - c_h^{\mathrm{i}} \geq c_h^{\mathrm{i}+2} - c_h^{\mathrm{i}+1}$.*

In the real-world, the above assumption represents the scenario where the increase in the repair costs due to the addition of a node to a set either stays the same or reduces as the size of the set increases, which is analogous to the concept of *economies of scale* where the average costs for an agency or an enterprise start falling as output increases [90]. Note that there is a budget $\beta \in \mathbb{R}_{\geq 0}$ on the total cost that can be paid by the authority to entities for repairing the nodes. That is, for each allocation $\mathcal{U}$, $\sum_{h \in \mathcal{T}} c_h^{|\mathcal{U}_h|} \leq \beta$.

Remark 7. *Note that although from the above definition of costs it may seem that an entity charges a cost corresponding to the number of nodes that are allocated to it even if the entity may not permanently repair some of the nodes that are allocated to it. However, the policies that we provide in this chapter ensure that each entity permanently repairs all the nodes that are allocated to it.*[1]

We now provide the definition of reward function as follows.

Definition 4.2.1. *Given a set of initial health values $v_0 = \{v_0^1, v_0^2, \ldots, v_0^N\}$, an allocation $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_Q\}$ that respects the allocation constraint $\{\mathcal{Z}_h\}$ and budget constraint β, and repair sequences $\bar{u}_{0:\infty} = \{\bar{u}_{0:\infty}^1, \bar{u}_{0:\infty}^2, \ldots, \bar{u}_{0:\infty}^Q\}$ for each entity, the reward $J(v_0, \mathcal{U}, \bar{u}_{0:\infty})$ is defined as the total number of nodes that get permanently repaired through the repair sequences after the allocation. More formally, $J(v_0, \mathcal{U}, \bar{u}_{0:\infty}) = |\{\mathrm{j} \in \mathcal{V} \mid \exists\, t \geq 0 \text{ s.t. } v_t^{\mathrm{j}} = 1\}|$.*

Based on the above definitions, we now define the problem that we focus in this chapter.

Problem 5. *Given a set $\mathcal{V}$ of N nodes with initial health values $v_0 = \{v_0^{\mathrm{j}}\}$, a set $\mathcal{T}$ of Q entities with target constraint $\{\mathcal{Z}_h\}$, costs $\{c_h^{\mathrm{i}}\}$ satisfying Assumption 2, repair rates $\{\Delta_{\mathrm{inc}}^{\mathrm{j},h}\}$, deterioration rates $\{\Delta_{dec}^{\mathrm{j}}\}$, and budget β, find an allocation $\mathcal{U}$ (that respects the allocation and budget constraints) and sequences $\bar{u}_{0:\infty}$ that maximize the reward $J(v_0, \mathcal{U}, \bar{u}_{0:\infty})$.*

We now provide the concept of a jump for the case when there are multiple entities.

Definition 4.2.2. *If an entity h starts targeting a node before permanently repairing the node it targeted in the last time-step, then the entity is considered to have jumped at that time-step. Mathematically, if $u_{t-1}^h = \mathrm{j}$, $v_t^{\mathrm{j}} < 1$ and $u_t^h \neq \mathrm{j}$, then a jump is said to have been made by entity h at time-step t. A control sequence that does not contain any jumps is called a* ***non-jumping sequence****.*

We refer to the allocation $\mathcal{U}$ along with the repair sequences $\bar{u}_{0:\infty}$ as a policy. We now define an α-policy as follows.

Definition 4.2.3. *Let $\{\mathcal{U}^*, \bar{u}_{0:\infty}^*\}$ be the optimal policy (for allocation and subsequent repair sequencing) of Problem 5. Then, for any $\alpha \in (0, 1]$, a policy $\{\mathcal{U}, \bar{u}_{0:\infty}\}$ (that respects the*

[1]↑This need not hold for the results in Section 4.4.3 but that does not affect the optimality of the solution as we assume that the budget is sufficiently large in that section.

allocation and budget constraints) is said to be α-policy if for all $v_0 \in [0,1]^N$, we have $J(v_0, \mathcal{U}, \bar{u}_{0:\infty}) \geq \alpha J(v_0, \mathcal{U}^*, \bar{u}^*_{0:\infty})$.

We first show that Problem 5 in general is NP-hard and later characterize optimal and near-optimal policies for various cases of the problem.

4.3 NP-hardness

We first define the decision version of *X3C* problem [91].

Problem 6 (*X3C*)**.** *Given a set $\mathcal{X}$ where $|\mathcal{X}| = 3q$ and collection $\mathcal{C}$ of 3-element subsets of $\mathcal{X}$, does there exist a subcollection $\overline{\mathcal{C}} \subseteq \mathcal{C}$ such that each element of the set $\mathcal{X}$ lies in exactly one of the 3-element subset of $\overline{\mathcal{C}}$.*

We now define the decision version of Problem 5 which we refer to as P.

Problem 7 (P)**.** *Given a set $\mathcal{V}$ of N nodes with initial health values $v_0 = \{v_0^{\mathrm{j}}\}$, a set $\mathcal{T}$ of Q entities with allocation constraint $\{\mathcal{Z}_h\}$ and costs $\{\mathrm{c}_h^{\mathrm{i}}\}$, repair rates $\{\Delta_{\mathrm{inc}}^{\mathrm{j},h}\}$, deterioration rates $\{\Delta_{\mathrm{dec}}^{\mathrm{j}}\}$, budget β, a mapping between the nodes and a positive number z entities, does there exist an allocation $\mathcal{U}$ (that respects the allocation and budget constraints) and sequences $\bar{u}_{0:\infty}$ such that the reward $J(v_0, \mathcal{U}, \bar{u}_{0:\infty}) \geq z$.*

We now provide the main result of this section.

Theorem 4.3.1. *Problem 5 is NP-hard.*

Proof. Given an instance of *X3C* we construct an instance of P as follows. For each element in $\mathcal{X}$, construct a node j such that $v_0^{\mathrm{j}} = 0.4$. Also, for each 3-element subset in $\mathcal{C}$ construct an entity h such that the nodes corresponding to elements in the 3-element subset belong to the set $\mathcal{Z}_h$. Let the deterioration and repair rates be homogeneous across all nodes and entities and be equal to 0.1 and 0.3, respectively. Also, let the costs charged for allocating one, two and three nodes be equal to 1 unit, 1.5 units and 1.75 units, respectively, for each entity and budget $\beta = 1.75q$. Finally, let $z = 3q$.

We first prove that if the answer to the given instance of *X3C* is *yes*, then the answer to the constructed instance of P is also *yes*. Suppose the answer to the given instance of

X3C is *yes*. Then, there exists a subcollection $\overline{\mathcal{C}} \subseteq \mathcal{C}$ such that each element of the set $\mathcal{X}$ lies in exactly one of the 3-element subset of $\overline{\mathcal{C}}$. Consider the allocation $\mathcal{U}$ where for all entities, the nodes that lie in the set $\mathcal{Z}_h$ are allocated to h (this allocation satisfies the budget constraint because the sum of the costs that are charged by all the entities is equal to $1.75|\overline{\mathcal{C}}| = 1.75q$). After that, each entity follows the policy of targeting the least healthy node at each time-step in its allocated set. Then, the total number of nodes that would be permanently repaired is equal to $3q$ because each entity is allocated a set $\{\mathrm{i}_1, \mathrm{i}_2, \mathrm{i}_3\}$ where $v_0^{\mathrm{i}_\mathrm{j}} > (3-\mathrm{j})\Delta_{dec}^{\mathrm{i}_\mathrm{j}}, \forall \mathrm{j} \in \{1, 2, 3\}$, and the collection $\overline{\mathcal{C}}$ contains all the elements of set $\mathcal{X}$. Thus, the answer to the constructed instance of P is yes.

Now we prove that if the answer to the constructed instance of P is *yes*, then the answer to the given instance of *X3C* is also *yes*. Suppose the answer to the constructed instance of P is yes. Then, there exists a policy (of allocation and repair sequences) that permanently repair $3q$ nodes. Note that the given allocation would be such that each entity is allocated three nodes; otherwise, the allocation would not satisfy the budget constraint (because $1.75q$ is the minimum total cost that is charged across any allocation that allocates $3q$ nodes within the allocation and budget constraints, and this value is only obtained when all the nodes are allocated to exactly q entities). Thus, consider the 3-element subsets that correspond to the entities that are allocated nodes in the given allocation and refer to the union of these 3-element subsets as $\overline{\mathcal{C}}$. Then, each element of the set $\mathcal{X}$ lies in exactly one of the 3-element subset of $\overline{\mathcal{C}}$. Thus, the answer to the given instance of *X3C* is yes and therefore the result follows. □

Since Problem 5 is NP-hard, we will now focus on various special cases of the problem and characterize optimal and near-optimal policies for them.

4.4 Policies for $\Delta_{inc}^{\mathrm{j},h} > \Delta_{dec}^{\mathrm{j}}, \forall \mathrm{j} \in \mathcal{V}, h \in \mathcal{T}$

In this section, we assume that the repair rates are sufficiently larger than the deterioration rates. That is, we make the following assumption.

Assumption 3. *For all* $\mathrm{j} \in \{1, \ldots, N\}$ *and* $h \in \{1, \ldots, Q\}$, *we assume* $\Delta_{inc}^{\mathrm{j},h} > (N-1)\Delta_{dec}^{\mathrm{j}}$ *and* $\Delta_{inc}^{\mathrm{j},h} > \sum_{k \in \{1,\ldots,N\} \setminus \mathrm{j}} \Delta_{dec}^{k}$.

We now present the definition of *modified health value.*

Definition 4.4.1. *The modified health value of a node at a time-step is its health value minus its deterioration rate.*

We start by reviewing some useful results pertaining to repair of nodes by a single entity (from Chapter 2).

4.4.1 Finding the largest subset of nodes that can be repaired by a single agency

We will use the following result (Lemma 5 of Chapter 2).

Lemma 16. *Suppose there are $\overline{N}(\geq 2)$ nodes represented by the set $\overline{\mathcal{V}} \subseteq \mathcal{V}$, and Assumption 3 holds. Then, there exists a sequence that allows a given entity to permanently repair $z(\leq \overline{N})$ nodes only if there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_z\} \subseteq \overline{\mathcal{V}}$ such that*

$$v_0^{\mathrm{i_j}} > (z - \mathrm{j})\Delta_{dec}^{\mathrm{i_j}}, \quad \forall \mathrm{j} \in \{1, \ldots, z\}. \tag{4.2}$$

Based on Lemma 16, we now present Algorithm 2 that takes a given set $\overline{\mathcal{V}} \subseteq \mathcal{V}$, and returns the largest subset $\mathcal{Y} \subseteq \overline{\mathcal{V}}$ that can be permanently repaired by a given entity h.

Algorithm 2 Finding the largest set $\mathcal{Y}$ that can be permanently repaired by a given entity

---

Suppose Assumption 3 holds and a set $\overline{\mathcal{V}} \subseteq \mathcal{V}$ is given. Compute $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil$ for each node $\mathrm{j} \in \overline{\mathcal{V}}$. Set $\mathcal{Y} = \emptyset$ and $z = 0$, and repeat the following until the termination criterion is satisfied.

- Stop, if there is no node $\mathrm{j} \in \overline{\mathcal{V}}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > z$. Otherwise, let $\mathrm{j} \in \overline{\mathcal{V}}$ be the node with the lowest value of $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil$ that satisfies $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{dec}^{\mathrm{j}}} \right\rceil > z$ among all the nodes in $\overline{\mathcal{V}}$. Remove node j from the set $\overline{\mathcal{V}}$ and add it to the set $\mathcal{Y}$, and set $z = z + 1$.

Remark 8. *Note that Algorithm 2 has polynomial-time complexity as argued in Chapter 2.*

We now present the following result (Lemma 7 of Chapter 2).

Lemma 17. *Suppose Assumption 3 holds and a set $\overline{\mathcal{V}} \subseteq \mathcal{V}$ is given. Let $\mathcal{Y}$ be the set obtained from Algorithm 2 and let $z = |\mathcal{Y}|$. Then, z is the largest number such that there exists a set $\mathcal{Y} = \{\mathrm{i}_1, \ldots, \mathrm{i}_z\} \subseteq \overline{\mathcal{V}}$ satisfying (4.2).*

We will also use the following results (from Lemma 6 and Theorem 2.4.2).

Lemma 18. *Suppose a set $\mathcal{U}_h \in \mathcal{V}$ containing $\overline{N}(\leq N)$ nodes is allocated to an entity h and Assumption 3 holds. Suppose there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_{\overline{N}}\} \subseteq \mathcal{U}_h$ such that*

$$v_0^{\mathrm{i_j}} > (\overline{N} - \mathrm{j})\Delta_{dec}^{\mathrm{i_j}}, \quad \forall \mathrm{j} \in \{1, \ldots, \overline{N}\}. \tag{4.3}$$

Then, the sequencing policy that targets the node with the least modified health value at each time-step permanently repairs all the nodes of set $\mathcal{U}_h$.

Lemma 19. *Suppose a set $\mathcal{U}_h \in \mathcal{V}$ of nodes is allocated to an entity h and Assumption 3 holds. In order to maximize the number of nodes that are permanently repaired by h, an optimal sequencing policy is that entity h should target the node with the least modified health value at each time-step in the set $\mathcal{U}_h$.*

Based on these results, we now turn our attention to the problem we are focusing in this chapter, namely allocating nodes to *multiple* entities.

4.4.2 Allocating nodes to multiple entities when each entity can target all the nodes

In this section, we make the following assumption.

Assumption 4. *For each $h \in \mathcal{T}$, we have $\mathcal{Z}_h = \mathcal{V}$. Also, for each pair of entities $k \in \mathcal{T}$ and $l \in \mathcal{T}$, if $c_k^{\mathrm{i}} \leq c_l^{\mathrm{i}}$ for any $\mathrm{i} \geq 1$, then $c_k^{\mathrm{j}+1} - c_k^{\mathrm{j}} \leq c_l^{\mathrm{j}+1} - c_l^{\mathrm{j}}, \forall \mathrm{j} \geq 1$ and $c_k^1 \leq c_l^1$.*

The first part of the above assumption ensures that each entity can target all the nodes. The second part of the assumption ensures that for each pair of entities if one of them has lower or equal cost than the other entity for a given number of nodes that are allocated, then the former entity always has lower or equal difference in the costs that are charged for any given consecutive numbers of nodes as compared to the latter entity. This assumption implies that there is a single ordering of entities based on the costs they charge and that remains the same regardless of the number of nodes that are allocated (as $c_k^{\mathrm{i}} = c_k^1 + (c_k^2 - c_k^1) + \ldots + (c_k^{\mathrm{i}} - c_k^{\mathrm{i}-1}) \leq c_l^1 + (c_l^2 - c_l^1) + \ldots + (c_l^{\mathrm{i}} - c_l^{\mathrm{i}-1}) = c_l^{\mathrm{i}}, \forall \mathrm{i} \geq 1)$. Therefore, we denote to the *lowest cost entity* as the one that charges the least cost among all entities regardless of the number of nodes that are allocated (and similarly we can define *second lowest cost entity* and so on).

We now start by presenting a property of the optimal allocation policy.

Lemma 20. *Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities, $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and Assumption 3 and 4 holds. Then, it is optimal to allocate the largest number of nodes to the lowest cost entity, the second largest number of nodes to the second lowest cost entity, and so on.*

Proof. Denote an optimal policy as $A = \{C, D\}$ where C represents the allocation of nodes and D represents the subsequent sequencing policy followed by the entities for repairing the nodes. By Lemma 19, we can assume without loss of generality that D is the sequencing policy where each entity targets the node with the least modified health in its allocated set at each time-step. Let y_{j} be the number of nodes that are allocated to entity $\mathrm{j} \in \mathcal{T}$ in C. Suppose there exists a pair of entities $\{k, l\} \subseteq \mathcal{T}$ and $\mathrm{i} \geq 1$ such that $c_k^{\mathrm{i}} < c_l^{\mathrm{i}}$ and $y_k < y_l$. Let $\overline{C}$ be an allocation where entity k is allocated the set of nodes that is allocated to entity l in C, entity l is allocated the set of nodes that is allocated to k in C and all the remaining entities are allocated the same set of nodes as in C (and the sequencing policy given by Lemma 19 is followed after the allocation). Then, the allocation $\overline{C}$ would satisfy the budget constraint because $c_k^{\mathrm{i}} \leq c_l^{\mathrm{i}}, \forall \mathrm{i}$, $y_k < y_l$ and allocation C satisfies the budget constraint. Also, the number of nodes that would be permanently repaired by policy $B = \{\overline{C}, D\}$ would be the same as that by policy A because of Assumption 3 and Lemma 19. Therefore, one can iteratively apply the aforementioned argument to obtain a policy where the largest number of nodes are allocated to the lowest cost entity, the second largest number of nodes are allocated to the second lowest cost entity, and so on. Thus, the result follows. $\square$

We now present Algorithm 3 for allocating nodes to entities. The main idea of Algorithm 3 is to allocate the nodes by going through the entities in the increasing order of their costs using Algorithm 2.

The following is the main result of this section.

Algorithm 3 Allocation of nodes to entities

Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities and Assumption 3 holds. Set $\mathcal{B} = \mathcal{V}$, $\mathcal{C} = \mathcal{T}$ and $\gamma = \beta$. Repeat the following until the termination criterion is satisfied.

1. Stop if $\mathcal{B} = \emptyset$, $\mathcal{C} = \emptyset$ or $\gamma < \underline{c}$, where $\underline{c} = \min\{c_h^1, h \in \mathcal{C}\}$ is the lowest cost for allocating a single node among the entities in set $\mathcal{C}$.

2. Otherwise, let s be the entity that has the lowest cost in the set $\mathcal{C}$. Remove s from set $\mathcal{C}$. Denote the set that is obtained from Algorithm 2 when $\overline{\mathcal{V}} = \mathcal{B}$ as $\mathcal{Y}$. Let x be the largest number of nodes that can be allocated to entity s within budget γ. If $x \geq |\mathcal{Y}|$, then allocate set $\mathcal{Y}$ to entity s, remove set $\mathcal{Y}$ from set $\mathcal{B}$ and let $\gamma = \gamma - c_s^{|\mathcal{Y}|}$. Otherwise, allocate an arbitrary set $\overline{\mathcal{Y}} \subseteq \mathcal{Y}$ to entity s, with $|\overline{\mathcal{Y}}| = x$, remove set $\overline{\mathcal{Y}}$ from set $\mathcal{B}$ and let $\gamma = \gamma - c_s^x$.

Theorem 4.4.1. *Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities, $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and Assumptions 2, 3, and 4 holds. Then, the allocation provided by Algorithm 3, along with the sequencing policy where each entity targets the node with the least modified health value in its allocated set at each time-step, is optimal for Problem 5.*

Proof. Let $A = \{C, D\}$ be an optimal policy where the allocation of nodes is represented by C and the sequencing policy by D. By Lemma 19, we can assume without loss of generality that D is the sequencing policy where each entity targets the node with the least modified health in its allocated set at each time-step. Without loss of generality, we assume that each entity permanently repairs all nodes allocated to it in policy A (otherwise, the allocated sets can be reduced in size without affecting the optimality of policy A). Suppose that y nodes are permanently repaired by following sequencing policy D after allocation C such that each entity $\mathrm{j} \in \mathcal{T}$ permanently repairs $y_{\mathrm{j}}(\geq 0)$ nodes. Then, $y_1 + \ldots + y_Q = y$. Without loss of generality, we assume that $\{1, \ldots, Q\}$ represents the order of entities in the non-decreasing order of their costs (i.e., for all pairs $\{k, l\} \subseteq \mathcal{T}, \mathrm{i} \geq 1$, $c_k^{\mathrm{i}} \leq c_l^{\mathrm{i}}$ if $k \leq l$).

Let $\overline{y}$ be the total number of nodes that are permanently repaired when Algorithm 3 along with sequencing policy D is followed, where each entity $\mathrm{j} \in \mathcal{T}$ permanently repairs $\overline{y}_{\mathrm{j}}(\geq 0)$ nodes. We now argue that $\sum_{l=1}^{Q} \overline{y}_l \geq \sum_{l=1}^{Q} y_l$. Consider the case when $Q = 1$. We prove the result through contradiction. Suppose $\overline{y}_1 < y_1$. Note that there exists a set of nodes $\{\mathrm{i}_1^1, \ldots, \mathrm{i}_{y_1}^1\}$ that satisfies $v_0^{\mathrm{i}_k^1} > (y_1 - k)\Delta_{dec}^{\mathrm{i}_k^1}, \quad \forall k \in \{1, \ldots, y_1\}$ by Lemma 16 since y_1 nodes are permanently repaired by entity 1 after allocation C (along with sequencing policy D). Without loss of generality, we can assume that for all $k \in \{1, \ldots, y_1\}$, node i_k^1 is the node with the lowest value of $\left\lceil \frac{v_0^{\mathrm{i}_k^1}}{\Delta_{dec}^{\mathrm{i}_k^1}} \right\rceil$ such that $\left\lceil \frac{v_0^{\mathrm{i}_k^1}}{\Delta_{dec}^{\mathrm{i}_k^1}} \right\rceil > y_1 - k$ in the set $\mathcal{V} \setminus \{\mathrm{i}_{k+1}^1, \ldots, \mathrm{i}_{y_1}^1\}$. Then, under the aforementioned conditions, Algorithm 3 would allocate at least y_1 nodes to entity 1 as policy A satisfies the budget constraint. However, this leads to a contradiction because Algorithm 3 should allocate $\overline{y}_1(< y_1)$ nodes. Thus, the assumption that $\overline{y}_1 < y_1$ is false.

Now consider the case when $Q = 2$. We again prove by contradiction. Suppose $\overline{y}_1 + \overline{y}_2 < y_1 + y_2$. Note that $y_1 \geq y_2$ by Lemma 20. Note that for each entity $\mathrm{j} \in \{1, 2\}$, there exists a set $\{\mathrm{i}_1^{\mathrm{j}}, \ldots, \mathrm{i}_{y_{\mathrm{j}}}^{\mathrm{j}}\}$ that satisfies $v_0^{\mathrm{i}_k^{\mathrm{j}}} > (y_{\mathrm{j}} - k)\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}, \quad \forall k \in \{1, \ldots, y_{\mathrm{j}}\}$ by Lemma 16 since y_{j} nodes are repaired by entity j after allocation C (along with sequencing policy

D). Without loss of generality, we can assume that for all $\mathrm{j} \in \{1,2\}, k \in \{1,\ldots,y_{\mathrm{j}}\}$, node $\mathrm{i}_k^{\mathrm{j}}$ is the node with the lowest value of $\left\lceil \frac{v_0^{\mathrm{i}_k^{\mathrm{j}}}}{\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}} \right\rceil$ such that $\left\lceil \frac{v_0^{\mathrm{i}_k^{\mathrm{j}}}}{\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}} \right\rceil > y_{\mathrm{j}} - k$ in the set $\mathcal{V} \setminus \{\mathrm{i}_{k+1}^1,\ldots,\mathrm{i}_{y_1}^1\}$ (resp. $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1,\mathrm{i}_{k+1}^2,\ldots,\mathrm{i}_{y_2}^2\}$) if $\mathrm{j}=1$ (resp. $\mathrm{j}=2$); otherwise, we could swap $\mathrm{i}_k^{\mathrm{j}}$ with the node l that has the lowest value of $\left\lceil \frac{v_0^l}{\Delta_{dec}^l} \right\rceil$ such that $\left\lceil \frac{v_0^l}{\Delta_{dec}^l} \right\rceil > y_{\mathrm{j}} - k$ in the set $\mathcal{V} \setminus \{\mathrm{i}_{k+1}^1,\ldots,\mathrm{i}_{y_1}^1\}$ (resp. $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1,\mathrm{i}_{k+1}^2,\ldots,\mathrm{i}_{y_2}^2\}$) if $\mathrm{j}=1$ (resp. $\mathrm{j}=2$) without affecting the optimality of policy A. Then, under the aforementioned conditions, we argue that Algorithm 3 would allocate a total of y_1+y_2 nodes to the two entities as follows. Suppose there is no node l in the set $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1\}$ such that $\left\lceil \frac{v_0^l}{\Delta_{dec}^l} \right\rceil > y_1$. Then, Algorithm 2 would allocate y_1 nodes to entity 1 (i.e., the nodes $\{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1\}$) and at least y_2 nodes to entity 2 (i.e., at least the nodes $\{\mathrm{i}_1^2,\ldots,\mathrm{i}_{y_2}^2\}$) while satisfying the budget constraint. Suppose there is a node l_1 in the set $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1\}$ such that $\left\lceil \frac{v_0^{l_1}}{\Delta_{dec}^{l_1}} \right\rceil > y_1$ but there is no node l_2 in the set $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1\} \cup l_1$ such that $\left\lceil \frac{v_0^{l_2}}{\Delta_{dec}^{l_2}} \right\rceil > y_1+1$. Then, Algorithm 2 would allocate y_1+1 nodes to entity 1 (i.e., the nodes $\{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1\} \cup l_1$) and at least y_2-1 nodes to entity 2 (i.e., at least the nodes $\{\mathrm{i}_2^2,\ldots,\mathrm{i}_{y_2}^2\}$) while satisfying the budget constraint. That is because $c_{y_1+1}^1 + c_{y_2-1}^2 = c_{y_1}^1 + (c_{y_1+1}^1 - c_{y_1}^1) + c_{y_2}^2 - (c_{y_2}^2 - c_{y_2-1}^2) \leq c_{y_1}^1 + c_{y_2}^2$ as $c_{y_1+1}^1 - c_{y_1}^1 \leq c_{y_1}^1 - c_{y_1-1}^1 \leq c_{y_1}^2 - c_{y_1-1}^2 \leq c_{y_2}^2 - c_{y_2-1}^2$ (where the first inequality on the left comes from Assumption 2, the second inequality from the left comes from Assumption 4 and the last inequality comes from Assumption 2 as $y_1 \geq y_2$). Proceeding in this way for the remaining cases, we can argue that Algorithm 3 would allocate at least a total of y_1+y_2 nodes to the two entities. However, this leads to a contradiction because Algorithm 3 should allocate $\overline{y}_1+\overline{y}_2 (< y_1+y_2)$ nodes. Therefore, $\overline{y}_1+\overline{y}_2 < y_1+y_2$ does not hold.

Consider the case when $Q=3$. We again prove by contradiction. Suppose $\overline{y}_1+\overline{y}_2+\overline{y}_3 < y_1+y_2+y_3$. Note that $y_1 \geq y_2 \geq y_3$ by Lemma 20. Note that for each entity $\mathrm{j} \in \{1,2,3\}$, there exists a set $\{\mathrm{i}_1^{\mathrm{j}},\ldots,\mathrm{i}_{y_{\mathrm{j}}}^{\mathrm{j}}\}$ that satisfies $v_0^{\mathrm{i}_k^{\mathrm{j}}} > (y_{\mathrm{j}}-k)\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}, \quad \forall k \in \{1,\ldots,y_{\mathrm{j}}\}$ by Lemma 16 since y_{j} nodes are repaired by entity j after allocation C (along with sequencing policy D). Without loss of generality, we can assume that for all $\mathrm{j} \in \{1,2,3\}, k \in \{1,\ldots,y_{\mathrm{j}}\}$, node $\mathrm{i}_k^{\mathrm{j}}$ is the node with the lowest value of $\left\lceil \frac{v_0^{\mathrm{i}_k^{\mathrm{j}}}}{\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}} \right\rceil$ such that $\left\lceil \frac{v_0^{\mathrm{i}_k^{\mathrm{j}}}}{\Delta_{dec}^{\mathrm{i}_k^{\mathrm{j}}}} \right\rceil > y_{\mathrm{j}}-k$ in the set $\mathcal{V}\setminus\{\mathrm{i}_{k+1}^1,\ldots,\mathrm{i}_{y_1}^1\}$ (resp. $\mathcal{V} \setminus \{\mathrm{i}_1^1,\ldots,\mathrm{i}_{y_1}^1,\ldots,\mathrm{i}_1^{\mathrm{j}-1},\ldots,\mathrm{i}_{y_{\mathrm{j}-1}}^{\mathrm{j}-1},\mathrm{i}_{k+1}^{\mathrm{j}},\ldots,\mathrm{i}_{y_{\mathrm{j}}}^{\mathrm{j}}\}$) if $\mathrm{j}=1$ (resp. $\mathrm{j} \geq 2$); otherwise, we could swap $\mathrm{i}_k^{\mathrm{j}}$ with the node l that has the lowest value of $\left\lceil \frac{v_0^l}{\Delta_{dec}^l} \right\rceil$ such that $\left\lceil \frac{v_0^l}{\Delta_{dec}^l} \right\rceil > y_{\mathrm{j}} - k$

in the set $\mathcal{V} \setminus \{\mathrm{i}^1_{k+1}, \ldots, \mathrm{i}^1_{y_1}\}$ (resp. $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}, \ldots, \mathrm{i}^{\mathrm{j}-1}_1, \ldots, \mathrm{i}^{\mathrm{j}-1}_{y_{\mathrm{j}-1}}, \mathrm{i}^{\mathrm{j}}_{k+1}, \ldots, \mathrm{i}^{\mathrm{j}}_{y_{\mathrm{j}}}\}$) if $\mathrm{j} = 1$ (resp. $\mathrm{j} \geq 2$) without affecting the optimality of policy A. Then, under the aforementioned conditions, we argue that Algorithm 3 would allocate at least a total of $y_1 + y_2 + y_3$ nodes to the three entities as follows. Suppose there is no node l_1 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\}$ such that $\left\lceil \frac{v_0^{l_1}}{\Delta^{l_1}_{dec}} \right\rceil > y_1$ and there is no node l_2 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}, \mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}\}$ such that $\left\lceil \frac{v_0^{l_2}}{\Delta^{l_2}_{dec}} \right\rceil > y_2$. Then, Algorithm 2 would allocate y_1 nodes to entity 1 (i.e., the nodes $\{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\}$), y_2 nodes to entity 2 (i.e., the nodes $\{\mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}\}$) and at least y_3 nodes to entity 3 (i.e., at least the nodes $\{\mathrm{i}^3_1, \ldots, \mathrm{i}^3_{y_3}\}$) while satisfying the budget constraint. Suppose there is a node l_1 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\}$ such that $\left\lceil \frac{v_0^{l_1}}{\Delta^{l_1}_{dec}} \right\rceil > y_1$, there is no node l_2 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\} \cup l_1$ such that $\left\lceil \frac{v_0^{l_2}}{\Delta^{l_2}_{dec}} \right\rceil > y_1 + 1$ and there is no node l_3 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}, l_1, \mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}\}$ such that $\left\lceil \frac{v_0^{l_3}}{\Delta^{l_3}_{dec}} \right\rceil > y_2$. Then, Algorithm 2 would allocate $y_1 + 1$ nodes to entity 1 (i.e., the nodes $\{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\} \cup l_1$), y_2 nodes to entity 2 (i.e., the nodes $\{\mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}\}$) and at least $y_3 - 1$ nodes to entity 3 (i.e., at least the nodes $\{\mathrm{i}^3_2, \ldots, \mathrm{i}^3_{y_3}\}$) while satisfying the budget constraint. That is because $c^1_{y_1+1} + c^2_{y_2} + c^3_{y_3-1} = c^1_{y_1} + (c^1_{y_1+1} - c^1_{y_1}) + c^2_{y_2} + c^3_{y_3} - (c^3_{y_3} - c^3_{y_3-1}) \leq c^1_{y_1} + c^2_{y_2} + c^3_{y_3}$ as $c^1_{y_1+1} - c^1_{y_1} \leq c^1_{y_1} - c^1_{y_1-1} \leq c^3_{y_1} - c^3_{y_1-1} \leq c^3_{y_3} - c^3_{y_3-1}$ (where the first inequality on the left comes from Assumption 2, the second inequality from the left comes from Assumption 4 and the last inequality comes from Assumption 2 as $y_1 \geq y_3$). Suppose there is a node l_1 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\}$ such that $\left\lceil \frac{v_0^{l_1}}{\Delta^{l_1}_{dec}} \right\rceil > y_1$, there is no node l_2 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\} \cup l_1$ such that $\left\lceil \frac{v_0^{l_2}}{\Delta^{l_2}_{dec}} \right\rceil > y_1 + 1$, there is a node l_3 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}, l_1, \mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}\}$ such that $\left\lceil \frac{v_0^{l_3}}{\Delta^{l_3}_{dec}} \right\rceil > y_2$ but no node l_4 in the set $\mathcal{V} \setminus \{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}, l_1, \mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}, l_2\}$ such that $\left\lceil \frac{v_0^{l_4}}{\Delta^{l_4}_{dec}} \right\rceil > y_2 + 1$. Then, Algorithm 2 would allocate $y_1 + 1$ nodes to entity 1 (i.e., the nodes $\{\mathrm{i}^1_1, \ldots, \mathrm{i}^1_{y_1}\} \cup l_1$), $y_2 + 1$ nodes to entity 2 (i.e., the nodes $\{\mathrm{i}^2_1, \ldots, \mathrm{i}^2_{y_2}, l_2\}$) and at least $y_3 - 2$ nodes to entity 3 (i.e., at least the nodes $\{\mathrm{i}^3_3, \ldots, \mathrm{i}^3_{y_3}\}$) while satisfying the budget constraint. That is because $c^1_{y_1+1} + c^2_{y_2+1} + c^3_{y_3-2} = c^1_{y_1} + (c^1_{y_1+1} - c^1_{y_1}) + c^2_{y_2} + (c^2_{y_2+1} - c^2_{y_2}) + c^3_{y_3} - (c^3_{y_3} - c^3_{y_3-2}) \leq c^1_{y_1} + c^2_{y_2} + c^3_{y_3}$ as $c^1_{y_1+1} - c^1_{y_1} \leq c^1_{y_1} - c^1_{y_1-1} \leq c^3_{y_1} - c^3_{y_1-1} \leq c^3_{y_3} - c^3_{y_3-1}$, $c^2_{y_2+1} - c^2_{y_2} \leq c^2_{y_2-1} - c^2_{y_2-2} \leq c^3_{y_2-1} - c^3_{y_2-2} \leq c^3_{y_3-1} - c^3_{y_3-2}$ and $c^3_{y_3} - c^3_{y_3-2} \leq c^3_{y_3} - c^3_{y_3-1} + c^3_{y_3-1} - c^3_{y_3-2}$. Proceeding in this way for the remaining cases, we can argue that Algorithm 2 would allocate at least a total of $y_1 + y_2 + y_3$ nodes to the three entities. However, this leads to a contradiction because Algorithm 3 should allocate $\overline{y}_1+\overline{y}_2+\overline{y}_3(< y_1+y_2+y_3)$ nodes. Therefore, the assumption $\overline{y}_1+\overline{y}_2+\overline{y}_3 < y_1+y_2+y_3$ is not true.

Proceeding in the above way we can argue that $\sum_{l=1}^{Q}\overline{y}_l \geq \sum_{l=1}^{Q} y_l$ for all values of Q. Thus, the number of nodes that are permanently repaired by Algorithm 3 along with sequencing policy D is not less than that by allocation C along with sequencing policy D. Thus, the result follows. $\square$

Remark 9. *Note that Algorithm 3 has polynomial-time complexity because it involves a loop that executes Algorithm 2 at most Q times.*

We now give an example to illustrate Algorithm 3.

Example 11. *Consider four nodes a, b, c and d such that $v_0^a = 0.05, v_0^b = 0.15, v_0^c = 0.06$ and $v_0^d = 0.07$. Suppose there are two entities e and f such that for all $\mathrm{i} \geq 1$, $c_{\mathrm{e}}^{\mathrm{i}} = \mathrm{i}c_{\mathrm{e}}^1$ and $c_f^{\mathrm{i}} = \mathrm{i}c_f^1$, where $c_{\mathrm{e}}^1 = 6$ units and $c_f^1 = 8$ units. Also, $\mathcal{Z}_{\mathrm{e}} = \mathcal{Z}_f = \mathcal{V}$. The total budget is $\beta = 19$ units. Also, suppose $\Delta_{inc}^{a,\mathrm{e}} = \Delta_{inc}^{a,f} = \Delta_{inc}^{b,\mathrm{e}} = \Delta_{inc}^{b,f} = \Delta_{inc}^{c,\mathrm{e}} = \Delta_{inc}^{c,f} = \Delta_{inc}^{d,\mathrm{e}} = \Delta_{inc}^{d,f} = 0.4$ and $\Delta_{\mathrm{dec}}^a = \Delta_{\mathrm{dec}}^b = \Delta_{\mathrm{dec}}^c = \Delta_{\mathrm{dec}}^d = 0.1$. If Algorithm 3 is followed for the allocation then entity e (which is the least costly among the two entities) is first allocated the nodes a and b as node a is the node $\mathrm{j} \in \{a, b, c, d\}$ with the lowest value of $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{\mathrm{dec}}^{\mathrm{j}}} \right\rceil$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{\mathrm{dec}}^{\mathrm{j}}} \right\rceil > 0$, b is the only node $\mathrm{j} \in \{b, c, d\}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{\mathrm{dec}}^{\mathrm{j}}} \right\rceil > 1$ and there is no node $\mathrm{j} \in \{c, d\}$ such that $\left\lceil \frac{v_0^{\mathrm{j}}}{\Delta_{\mathrm{dec}}^{\mathrm{j}}} \right\rceil > 2$. Note that $\gamma = 19 - 6 - 6 = 7$ after allocating the nodes to entity e. Thus, it is not possible to allocate any nodes to entity f as $\gamma = 7 < 8 = c_f^1$. After the allocation of nodes a and b to entity e, the sequence of targeting the node with the least modified health value at each time-step is followed by entity e, which permanently repairs both nodes.*

In this section, we focused on the case when each entity can target all the nodes and thus we will be focusing on the case when it may not be possible for each entity to target all the nodes and a condition on the budget constraint holds.

4.4.3 Allocating nodes to multiple entities when the budget is sufficiently large

In this section, we focus on the case when the *budget is sufficiently large.* The condition on the budget being sufficiently large means that for all instances of the problem, the optimal solution is the same as the optimal solution of the problem if there was no budget constraint (i.e., $\beta = \infty$). Thus, we make this assumption in this chapter.

Assumption 5. *β is sufficiently large.*

We start by defining the *valuation* of a set.

Definition 4.4.2. *Let $v_h(\mathcal{S})$ be the maximum number of nodes that can be permanently repaired by entity h from the set $\mathcal{S}$ (with the parameters corresponding to entity h and set $\mathcal{S}$ being implicit). Note that the set $\mathcal{S}$ may contain nodes that do not belong to the set $\mathcal{Z}_h$. Then, we define $v_h(\mathcal{S})$ to be the valuation of set $\mathcal{S}$ for entity h.*

Remark 10. *For our proposes, we allow $v_h(\mathcal{S})$ to be also defined for a set $\mathcal{S}$ that contains nodes that do not belong to the set $\mathcal{Z}_h$ as mentioned above (this allows defining the valuations of each entity h over all possible subsets of set $\mathcal{V}$, which is useful in providing Algorithm 5 as will be seen later). Let $\overline{\mathcal{S}} = \mathcal{S} \cap \mathcal{Z}_h$ and $\mathcal{S}^* = \mathcal{S} \setminus \overline{\mathcal{S}}$. Then, $v_h(\mathcal{S}) = v_h\left(\overline{\mathcal{S}}\right)$ as the nodes of the set $\mathcal{S}^*$ cannot be targeted by entity h.*

For our purposes, we now provide the following definitions.

Definition 4.4.3. *A set function f is said to be monotone if $f(\mathcal{S} \cup \{x\}) \geq f(\mathcal{S}), \forall \mathcal{S}, x$.*

Definition 4.4.4 (Submodular function [92])**.** *A set function f is said to be submodular if $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) \geq v_h(\mathcal{S} \cup \{y\} \cup \{x\}) - v_h(\mathcal{S} \cup \{y\}), \forall \mathcal{S}, \forall x, y \notin \mathcal{S}$.*

We first argue that the function v_h is monotone.

Lemma 21. *The function v_h is monotone for all h.*

Proof. Let y be the maximum number of nodes that are permanently repaired when $\mathcal{S}$ is allocated to entity h and D be the optimal repair sequence when $\mathcal{S}$ is allocated to entity h. Then, the maximum number of nodes that would be permanently repaired when the set $\mathcal{S} \cup \{x\}$ is allocated to entity h is at least y because entity h can follow sequence D in that case and it will permanently repair y nodes. Thus, the result follows. □

We now present the following result.

Lemma 22. $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) \leq 1, \forall \mathcal{S}, x, h.$

Proof. Note that $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) = 0, \forall \mathcal{S}, x, h$ holds trivially when $x \in \mathcal{S}$ because $\mathcal{S} \cup \{x\} = \mathcal{S}$. Therefore, we focus on the case when $x \notin \mathcal{S}$. Note that if $x \notin \mathcal{Z}_h$, then $v_h(\mathcal{S} \cup \{x\}) = v_h(\mathcal{S})$ and thus the result follows. Therefore, we focus on the case when $x \in \mathcal{Z}_h$. Let $\mathcal{S}^* \subseteq \mathcal{S}$ be the largest set of nodes that is permanently repaired by entity h when it is allocated the set $\mathcal{S}$, i.e., $v_h(\mathcal{S}) = |\mathcal{S}^*|$. Let $\overline{\mathcal{S}}^* \subseteq \mathcal{S} \cup \{x\}$ be the largest set of nodes that is permanently repaired by entity h when it is allocated the set $\mathcal{S} \cup \{x\}$. We now prove the result by contradiction. Suppose the optimal sequence for the set $\mathcal{S} \cup \{x\}$ permanently repairs $v_h(\mathcal{S}) + 2 = |\mathcal{S}^*| + 2$ nodes, i.e., $v_h(\mathcal{S} \cup x) - v_h(\mathcal{S}) = 2$. Let D be the repair sequence that permanently repairs the largest number of nodes when entity h is allocated the set $\mathcal{S} \cup \{x\}$. Note that $\overline{\mathcal{S}}^* \setminus x \subseteq \mathcal{S}$. Consider a sequence E that is the same as sequence D except that an arbitrary node from the set $\overline{\mathcal{S}}^* \setminus x$ is targeted at those time-steps in E where node x is targeted in sequence D. Thus, if entity h follows sequence E on the set $\overline{\mathcal{S}}^* \setminus x$, then it would permanently repair at least $|\overline{\mathcal{S}}^*| - 1$ nodes as the nodes in the set $\overline{\mathcal{S}}^* \setminus x$ would be more frequently targeted in sequence E than that in sequence D. However, the number of nodes that are permanently repaired by sequence E on the set $\overline{\mathcal{S}}^* \setminus x \subseteq \mathcal{S}$ cannot exceed $|\mathcal{S}^*|$ because the latter is the largest number of nodes that can be permanently repaired when entity h is allocated the set $\mathcal{S}$. Thus, $|\overline{\mathcal{S}}^*| - 1 \leq |\mathcal{S}^*|$, which leads to a contradiction because $v_h(\mathcal{S} \cup x) - v_h(\mathcal{S}) = |\overline{\mathcal{S}}^*| - |\mathcal{S}^*| = 2$. Thus, the result follows. $\square$

Lemma 23. *If $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) = 1$, then node x is permanently repaired by entity h when it is allocated the set $\mathcal{S} \cup \{x\}$.*

Proof. Note that $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) = 1$ is only possible if $x \in \mathcal{Z}_h$ and thus we assume $x \in \mathcal{Z}_h$. We prove this result by contradiction. Suppose node x is not permanently repaired by entity h when it is allocated the set $\mathcal{S} \cup \{x\}$. Let $\overline{\mathcal{S}}^*$ be the largest set that is permanently repaired by entity h when it is allocated the set $\mathcal{S} \cup \{x\}$. Then, $\overline{\mathcal{S}}^* \subseteq \mathcal{S}$. Let $\mathcal{S}^*$ be the largest set that is permanently repaired by entity h when it is allocated the set $\mathcal{S}$. Then, $|\overline{\mathcal{S}}^*| \leq |\mathcal{S}^*|$ from the definition of the set $\mathcal{S}^*$, leading to a contradiction because $v_h(\mathcal{S} \cup x) - v_h(\mathcal{S}) = |\overline{\mathcal{S}}^*| - |\mathcal{S}^*| = 1$. Thus, the result follows. $\square$

Remark 11. *Note that Lemmas 21, 22 and 23 hold regardless of any condition on the deterioration and repair rates.*

Lemma 24. *Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities, and Assumptions 3 and 5 hold. Then, function v_h is submodular for all h.*

Proof. In order to prove that v_h is a submodular function, it is sufficient to prove that for all $\mathcal{S}$ and for all $x, y \notin \mathcal{S}$, if $v_h(\mathcal{S} \cup \{x\}) - v_h(\mathcal{S}) = 0$ then $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) - v_h(\mathcal{S} \cup \{y\}) = 0$ (because of the definition of submodularity and Lemmas 21 and 22). Note that if $x \notin \mathcal{Z}_h$, then $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = v_h(\mathcal{S} \cup \{y\})$ and thus the result follows. If $y \notin \mathcal{Z}_h$, then $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = v_h(\mathcal{S} \cup \{x\}) = v_h(\mathcal{S}) = v_h(\mathcal{S} \cup \{y\})$, and thus the result follows. Therefore, we focus on the case when $x \in \mathcal{Z}_h$ and $y \in \mathcal{Z}_h$.

Suppose $v_h(\mathcal{S} \cup \{x\}) = v_h(\mathcal{S}) = t$. Note that $t \leq v_h(\mathcal{S} \cup \{y\} \cup \{x\}) \leq t + 1$ by Lemmas 21 and 22. Also, if $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = t$ then $v_h(\mathcal{S} \cup \{y\}) = t$ because of Lemma 21 and since $v_h(\mathcal{S}) = t$; thus, the submodularity property of function v_h trivially holds. Therefore, we focus on the case when $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = t+1$ (as $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) \leq t+1$ by Lemma 22).

Suppose $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = t+1$. Note that $t \leq v_h(\mathcal{S} \cup \{y\}) \leq t+1$. If $v_h(\mathcal{S} \cup \{y\}) = t$ then the function v_h is not submodular. Thus, we prove the result by contradiction. Suppose $v_h(\mathcal{S} \cup \{y\}) = t$. We first consider the case when the set $\mathcal{S}$ contains one node; say $\mathcal{S} = \{\mathrm{i}_1\}$. Since the initial health value of each node is positive (i.e., $\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 0$), any sequence would permanently repair node i_1 and thus we have $v_h(\mathcal{S}) = 1$. Also, recall that $v_h(\mathcal{S} \cup \{x\}) = t$ and thus $v_h(\mathcal{S} \cup \{x\}) = 1$ (as $t = 1$). Therefore, $0 < \left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil \leq 1$ and $0 < \left\lceil \frac{v_0^{x}}{\Delta_{dec}^{x}} \right\rceil \leq 1$ (if this is not the case, then $v_h(\mathcal{S} \cup \{x\}) = 2$ by Lemma 18). Similarly, $0 < \left\lceil \frac{v_0^{y}}{\Delta_{dec}^{y}} \right\rceil \leq 1$ as $v_h(\mathcal{S} \cup \{y\}) = t = 1$. Under the above conditions, we obtain $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = 1$. However, this leads to contradiction because $v_h(\mathcal{S} \cup \{y\} \cup \{x\}) = t + 1 = 2$. Thus, our assumption that $v_h(\mathcal{S} \cup \{y\}) = t$ does not hold.

We now focus on the case when $\mathcal{S} = \{\mathrm{i}_1, \mathrm{i}_2\}$. Note that the case $t = 1$ proceeds in a similar manner as when $\mathcal{S} = \{\mathrm{i}_1\}$. Thus, we focus on the case when $t = 2$. If $t = 2$, then $v_h(\mathcal{S}) = 2$. Thus, without loss of generality, we have

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 1, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil > 0, \tag{4.4}$$

by Lemma 16. Also, recall that $v_h(\mathcal{S} \cup x) = t = 2$. Suppose $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 1$. Then, either

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 1, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 1, \tag{4.5}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 2, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 2, \tag{4.6}$$

by (4.4) (if this is not the case, then $v_h(\mathcal{S} \cup \{x\}) = 3$ by Lemma 18). Similarly, if $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 2$, then

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 2,\ 0 < \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil < 3, \tag{4.7}$$

by (4.4). Note that $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil$ cannot be larger than two (because in that case we would have $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil > 2, \left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 1, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil > 0$, and thus $v_h(\mathcal{S} \cup \{x\}) = 3$, which leads to a contradiction). Also, we have analogous conditions for the initial health values of the set $\mathcal{S} \cup \{y\}$ when $v_h(\mathcal{S} \cup \{y\}) = t = 2$. Now, we focus on the value of $v_h(\mathcal{S} \cup \{x\} \cup \{y\})$ under the above conditions. If $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = \left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 1$ or $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = \left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 2$, then we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 2$ by (4.5)-(4.6) and (4.7), respectively; however, this leads to a contradiction because $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = t + 1 = 3$. Suppose $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 1$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 2$, then (4.7) holds and we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 2$ (we can argue similarly for the case when $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 2$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 1$); this again leads to a contradiction. Thus, our assumption $v_h(\mathcal{S} \cup \{y\}) = t$ does not hold.

We now focus on the case when $\mathcal{S} = \{\mathrm{i}_1, \mathrm{i}_2, \mathrm{i}_3\}$. Note that the case when $t = 1$ proceeds in a similar manner as when $\mathcal{S} = \{\mathrm{i}_1\}$ and the case when $t = 2$ proceeds in a similar manner as that when $\mathcal{S} = \{\mathrm{i}_1, \mathrm{i}_2\}$ and $t = 2$. Thus, we focus on the case when $t = 3$. If $t = 3$, then $v_h(\mathcal{S}) = 3$. Thus, without loss of generality, we have

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 2, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil > 1, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil > 0. \tag{4.8}$$

Also, recall that $v_h(\mathcal{S} \cup \{x\}) = t = 3$. Suppose $\left\lceil \frac{v_0^{x}}{\Delta_{dec}^{x}} \right\rceil = 1$. Then, either

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 2, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil > 1, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 1, \tag{4.9}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil > 2, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 2, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 2, \tag{4.10}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 3, 1 < \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil < 4. \tag{4.11}$$

If $\left\lceil \frac{v_0^{x}}{\Delta_{dec}^{x}} \right\rceil = 2$, then either

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil \geq 3, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 2, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 1, \tag{4.12}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 1, \tag{4.13}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, 1 < \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil < 4, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 2, \tag{4.14}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 3. \tag{4.15}$$

If $\left\lceil \frac{v_0^{x}}{\Delta_{dec}^{x}} \right\rceil = 3$, then either

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, 1 < \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil < 4, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 1, \tag{4.16}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, 1 < \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil < 4, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 2, \tag{4.17}$$

or,

$$\left\lceil \frac{v_0^{\mathrm{i}_1}}{\Delta_{dec}^{\mathrm{i}_1}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_2}}{\Delta_{dec}^{\mathrm{i}_2}} \right\rceil = 3, \left\lceil \frac{v_0^{\mathrm{i}_3}}{\Delta_{dec}^{\mathrm{i}_3}} \right\rceil = 3. \tag{4.18}$$

Also, we have analogous conditions for the initial health values of the set $\mathcal{S} \cup \{y\}$ when $v_h(\mathcal{S} \cup \{y\}) = t = 3$. Now, we focus on the value of $v_h(\mathcal{S} \cup \{x\} \cup \{y\})$ under the above conditions. If $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = \left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 1$, $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = \left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 2$, or $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = \left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 3$, then we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 3$ by (4.9)-(4.11), (4.12)-(4.15) and (4.16)-(4.18), respectively (this leads to a contradiction because $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = t + 1 = 4$). Suppose $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 1$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 2$, then (4.12)-(4.15) hold and thus we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 3$ (we can similarly argue when $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 2$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 1$); however, this again leads to a contradiction. Suppose $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 1$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 3$, then (4.16)-(4.18) hold and thus we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 3$, which again leads to contradiction (we can similarly argue when $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 3$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 1$). Suppose $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 2$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 3$, then (4.16)-(4.18) hold and thus we obtain $v_h(\mathcal{S} \cup \{x\} \cup \{y\}) = 3$, which again leads to contradiction (we can similarly argue when $\left\lceil \frac{v_0^x}{\Delta_{dec}^x} \right\rceil = 3$ and $\left\lceil \frac{v_0^y}{\Delta_{dec}^y} \right\rceil = 2$). Therefore, our assumption that $v_h(\mathcal{S} \cup \{y\}) = t = 2$ does not hold.

We can proceed in the same way when $|\mathcal{S}| > 3$. Thus, the result follows. ☐

We now provide an algorithm to compute valuation of a given set for an entity.

Algorithm 4 Computing $v_h(\mathcal{S})$

Suppose Assumption 3 holds, and a set $\mathcal{S} \subseteq \mathcal{V}$ and an entity h are given.

1. Construct a set $\overline{\mathcal{S}}$ that contains the nodes of the set $\mathcal{S} \cap \mathcal{Z}_h$.
2. Run Algorithm 2 on the set $\overline{\mathcal{S}}$ and let z be the output of that algorithm.
3. Set $v_h(\mathcal{S}) = z$.

We now present Algorithm 5 for allocating nodes to entities that is motivated from an algorithm from [82]. The idea behind the algorithm is to iteratively assign nodes to entities that value them the most. The input of the problem requires that the valuations by entities are submodular and thus the contribution of our work is to prove that (as in Lemma 24).

Algorithm 5 Allocation of nodes to entities

Suppose Assumption 3 holds.

1. Let $\mathcal{X}_1 = \emptyset, \mathcal{X}_2 = \emptyset, \ldots, \mathcal{X}_Q = \emptyset$.
2. For $y = 1, \ldots, N$, repeat the following:
   (a) Let h be the entity with the largest value of $v_h\left(\mathcal{X}_h \cup \{y\}\right) - v_h\left(\mathcal{X}_h\right)$.
   (b) Set $\mathcal{X}_h = \mathcal{X}_h \cup \{y\}$.
3. For each $h \in \mathcal{T}$, let $\mathcal{U}_h = \mathcal{X}_h \cap \mathcal{Z}_h$. Then, $\{\mathcal{U}_1, \ldots, \mathcal{U}_Q\}$ denote the sets of allocated nodes.

Remark 12. *Note that Algorithm 5 has polynomial-time complexity as the valuations can be polynomially computed using Algorithm 4.*

We now present the main result of this section.

Theorem 4.4.2. *Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities, and Assumptions 3 and 5 hold. Then, the allocation algorithm provided in Algorithm 5 along with the sequencing policy where each entity targets the node with the least modified health value in its allocated set at each time-step is 1/2-optimal.*

The proof of the above result comes from Lemma 24 and [82].

We now focus on another instance of our problem where $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j},h}_{\mathrm{inc}}, \forall \mathrm{j} \in \mathcal{V}, h \in \mathcal{T}$.

4.5 Policies for $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j},h}_{\mathrm{inc}}, \forall \mathrm{j} \in \mathcal{V}, h \in \mathcal{T}$

In this section, we analyze the case when the deterioration rates are larger than the repair rates. We will use the following assumption in this section.

Assumption 6. *Suppose for all $\mathrm{j} \in \{1, \ldots, N\}$ and $h \in \{1, \ldots, Q\}$, $\Delta^{\mathrm{j},h}_{\mathrm{inc}} = \Delta^{h}_{\mathrm{inc}}$, $\Delta^{\mathrm{j}}_{\mathrm{dec}} = \Delta_{dec}$, $\Delta_{dec} \geq \Delta^{h}_{\mathrm{inc}}$ and $c^{\mathrm{i}}_h = c^{\mathrm{i}}$. Also, for each entity $h \in \{1, \ldots, Q\}$, suppose there exists a positive integer n_h such that $\Delta_{dec} = n_h \Delta^{h}_{\mathrm{inc}}$. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$ and entity $h \in \{1, \ldots, Q\}$, suppose there exists a positive integer m^{h}_{j} such that $1 - v^{\mathrm{j}}_0 = m^{h}_{\mathrm{j}} \Delta^{h}_{\mathrm{inc}}$.*

The above assumption ensures that no node gets permanently repaired partway through a time-step.

We will use the following result for repair of nodes by a single entity from Chapter 2.

Lemma 25. *Suppose a set $\mathcal{U}_h$ is allocated to an entity h and Assumption 6 holds. In order to maximize the number of nodes that are permanently repaired by h, an optimal sequencing policy is that entity h should target the healthiest node in $\mathcal{U}_h$ at each time-step.*[2]

We now start with the following result.

Lemma 26. *Let there be $N(\geq 2)$ nodes, $Q(\leq N)$ entities, $\beta = \infty$, and suppose Assumption 6 holds. Consider a policy $A = \{C, D\}$ where C is the allocation policy and D is a non-jumping sequencing policy where each entity targets a set of nodes in decreasing order of their initial health. Suppose k is the healthiest node at $t = 0$ that is not targeted by an entity at $t = 0$ in D such that there is an entity, say entity a, where $\mathcal{Z}_a = k$ and entity a targets a node with less health value than node k at $t = 0$ in D. Suppose entity a targets nodes $\{\mathrm{i}_1, \ldots, \mathrm{i}_f\}$ in sequencing policy D and permanently repairs all of them. Let x be the number of nodes that are permanently repaired in policy A. Consider another policy $B = \{C, E\}$ where allocation $\overline{C}$ is the same as C except that k is allocated to entity a, and entity a targets node k at $t = 0$ and follows the sequence $\{\mathrm{i}_1, \ldots, \mathrm{i}_{f-1}\}$ afterwards in E; all the other entities target the remaining nodes in the same order as that in D. Then, at least $x - 1$ nodes are permanently repaired in B.*

Proof. Note that the allocation $\overline{C}$ satisfies the budget constraint since $\beta = \infty$. We first focus on entity a. Note that $k \notin \{\mathrm{i}_1, \ldots, \mathrm{i}_f\}$ because $v_0^k > v_0^{\mathrm{i}_1}$ and a targets the nodes $\{\mathrm{i}_1, \ldots, \mathrm{i}_f\}$ in the decreasing order of their initial health in sequencing policy D. Also, the nodes $k, \mathrm{i}_1, \ldots, \mathrm{i}_{f-1}$ are targeted at an earlier time in sequencing policy E by entity a as compared to the nodes $\mathrm{i}_1, \ldots, \mathrm{i}_f$ in D since $v_0^k > \max\{v_0^{\mathrm{i}_1}, \ldots, v_0^{\mathrm{i}_f}\}$ and $v_0^{\mathrm{i}_1} \geq \ldots \geq v_0^{\mathrm{i}_f}$ (as each entity targets a set of the nodes in the decreasing order of their initial health). Thus, entity a permanently repairs the nodes $k, \mathrm{i}_1, \ldots, \mathrm{i}_{f-1}$ in E.

[2]↑ Equivalently, the optimal sequence is the non-jumping sequence that targets the nodes in decreasing order of their initial health values in the set $\mathcal{U}_h$.

We now compare the number of nodes that are permanently repaired by the entities apart from entity a in the sequencing policies D and E. Note that either node k is not permanently repaired in D or node k is permanently repaired by an entity $b(\neq a)$ in D (note that entities a and b cannot be the same because entity a does not target node k in D as mentioned before). Consider the case when no entity permanently repairs node k in D. Then, all the entities other than entity a would permanently repair the same number of nodes in E as that in D. We now consider the case when node k is permanently repaired by entity b in D. Suppose entity b permanently repairs g nodes in D. Denote the order that is followed by entity b for targeting the nodes in D as $\{\bar{\mathrm{i}}_1, \bar{\mathrm{i}}_2, \ldots, \bar{\mathrm{i}}_g\}$, where $\bar{\mathrm{i}}_{\mathrm{j}} = k$ such that $1 \leq \mathrm{j} \leq g$. Since entity b follows the same order for targeting the nodes in E as that followed in D, at least $g-1$ nodes (i.e., the nodes $\{\bar{\mathrm{i}}_1, \ldots, \bar{\mathrm{i}}_{\mathrm{j}-1}, \bar{\mathrm{i}}_{\mathrm{j}+1}, \ldots, \bar{\mathrm{i}}_g\}$) would be permanently repaired by b in E as these nodes would start to get targeted at an earlier or same time-step in comparison to D. Note that all the remaining entities (i.e., entities apart from a and b) would permanently repair the same number of nodes in both D and E. Thus, the number of nodes that are permanently repaired in policy $B = \{\overline{C}, E\}$ is at least equal to $x - 1$. $\square$

We now present the main result of this section, namely an *online* policy where the nodes are sequentially allocated to entities and the entities permanently repair their currently allocated nodes before they are allocated new nodes.

Theorem 4.5.1. *Suppose there are $N(\geq 2)$ nodes, $Q(\leq N)$ entities, $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and Assumption 6 holds. Consider the online policy where at each time-step the healthiest node that is currently not being targeted is allocated to an entity that is currently not repairing any node given that it is possible to do that allocation, until there are no more nodes to allocate or the budget runs out. Then, the aforementioned policy is 1/2-optimal for Problem 5.*

Proof. Let $A = \{C, D\}$ be an optimal policy where the allocation of nodes is given by C and the subsequent sequencing policy by D. Without loss of generality, we can assume that D is the policy where each entity targets the healthiest node in its allocated set at each time-step (or, the non-jumping sequence of targeting the allocated nodes in the decreasing order of their initial health) by Lemma 25. We now argue that the online policy would permanently repair at least half the number of nodes as that by policy A because of the following argument.

Let $\mathcal{X}_t \subseteq \mathcal{T}$ contain all the entities that are unassigned at time-step t (i.e., those entities that are not currently busy in targeting previously allocated nodes at time-step t) in sequencing policy D. Let $\mathcal{Y}_t \subseteq \mathcal{V}$ be the set of all the nodes that have health values in the interval $(0, 1)$ at time-step t and have not been targeted before time-step t in sequencing policy D. Let l_t be equal to the number of nodes in $\mathcal{Y}_t$ that have at least one entity in the set $\mathcal{X}_t$ that can target them. We now go to the first time-step T in sequencing policy D at which $l_T \geq 1$. Suppose k is the healthiest node in the set $\mathcal{Y}_T$ such that k is not targeted by an entity in $\mathcal{X}_T$ but there is an entity (say entity a) in the set $\mathcal{X}_t$ such that $\mathcal{Z}_a = k$. Suppose entity a permanently repairs f nodes from time-step T onwards in D. Let E be a sequencing policy which is the same as D from time-step 0 to $T-1$ but the portion of E from time-step T onwards is such that node k is targeted by entity a at time-step T, entity a targets the first $f-1$ nodes that it targeted in D after targeting node k and the remaining entities target the nodes in the same order as in D. Then, at least $x-1$ nodes would be permanently repaired in sequencing policy E by Lemma 26.

We iteratively repeat the above procedure so that at each time-step t the healthiest node that has not been targeted before is allocated to an entity that is available at time-step t (given that the allocation is possible). Note that since the number of nodes that are permanently repaired in the given sequencing policy either decreases or remains the same in each iteration of this procedure, the total cost of the permanently repaired nodes does not increase during this procedure as the costs are homogeneous across all the entities by Assumption 6. Thus, the final allocation satisfies the budget constraint as allocation C is a feasible allocation. Therefore, the aforementioned online policy is 1/2-optimal because 1) at each iteration of this iterative procedure, we move at least one node across the given sequencing policy and the number of nodes that are permanently repaired reduces by at most one, and 2) in the last iteration of this procedure when there is only one node, there is no decrease in the number of nodes that are permanently repaired because if the last node in the given sequencing policy can be permanently repaired then a healthier node can also be permanently repaired as once a node starts to get targeted in a non-jumping sequencing policy it is always permanently repaired. □

We now provide an example to illustrate the online policy.

Example 12. *Consider four nodes a, b, c and d such that $v_0^a = 0.9, v_0^b = 0.8, v_0^c = 0.6$ and $v_0^d = 0.5$. Suppose there are two entities* e *and f such that $c_{\mathrm{e}}^1 = c_f^1 = c^1 = 6$ units and $\mathcal{Z}_{\mathrm{e}} = \mathcal{Z}_f = \{a, b, c, d\}$. The total budget is $\beta = 23$ units. Also, suppose $\Delta_{inc}^{\mathrm{e}} = \Delta_{inc}^f = 0.1$ and $\Delta_{\mathrm{dec}} = 0.2$. Then, the nodes a and b are allocated to the entities* e *and f, respectively, (note that a and b could have also been allocated to f and* e*, respectively) at time-step $t = 0$ when the online policy is followed. Note that the budget that is available after the allocation of nodes at $t = 0$ is equal to $\gamma = 23 - 6 - 6 = 11$. Then, the time after time-step 0 at which the healthiest node gets permanently repaired is $t = 1$ when node a gets permanently repaired, and entity* e *becomes available and thus can be allocated another node. Thus, node c is allocated to entity* e *and the budget that is available after this allocation is $\gamma = 11 - 6 = 5$. Since $\gamma = 5 < 6 = c^1$, it is not possible to allocate node d to any entity. Thus, nodes a and c are allocated to* e *and node b is allocated to f. Figure 4.1 shows the progression of health values and budget at time-steps 0, 1 and 7.*

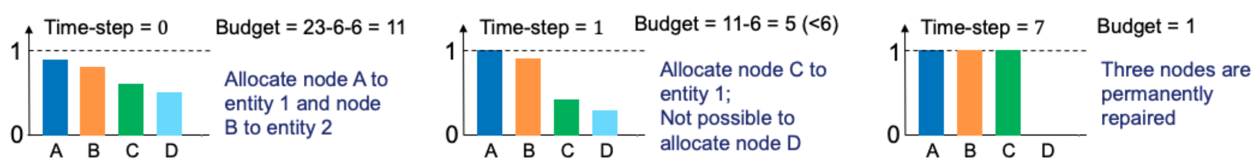

Figure 4.1. Progression of health values and budget at time-steps 0, 1 and 7 for Example 12.

Note that although we assumed that the costs are homogeneous across the entities in Assumption 6, the problem of finding the optimal policy under this assumption is non-trivial as shown in the following examples.

Example 13. *Consider three nodes a, b and c such that $v_0^a = 0.9, v_0^b = 0.8$ and $v_0^c = 0.2$. Suppose there are two entities d and* e *such that $c^1 = 6$ units and $\mathcal{Z}_d = \mathcal{Z}_{\mathrm{e}} = \{a, b, c\}$. The* *total budget is $\beta = 25$ units. Also, suppose $\Delta^{d}_{\mathrm{inc}} = \Delta^{\mathrm{e}}_{\mathrm{inc}} = 0.1$ and $\Delta_{\mathrm{dec}} = 0.2$. If the nodes are allocated based on the online policy then the nodes a and b are allocated to entities d and* e*, respectively, (note that a and b could have also been allocated to* e *and d, respectively) at $t = 0$. Note that by the time node a (i.e., the healthiest node at $t = 0$) is permanently repaired, node c permanently fails and thus is not allocated to any entity. Therefore, the nodes a and b are permanently repaired from the online policy. However, if the nodes a and b are allocated to entity d, and node c is allocated to entity* e*, then the aforementioned allocation satisfies the budget constraint and it is possible to repair all the nodes by following the sequence where each entity targets the healthiest node in the allocated set at each time-step. Thus, the online policy, in general, is not optimal. Note that although the aforementioned policy is not optimal in this example, it is indeed 1/2-optimal as proved in Theorem 4.5.1.*

Note that the policy of giving the largest subset of nodes that can be repaired to one entity, the second largest subset of nodes that can be repaired to the second entity and so on, is optimal in the above example but it may not be optimal under Assumption 6 as shown next (note that this policy was indeed optimal in the previous section).

Example 14. *Consider four nodes a, b, c and d such that $v^a_0 = 0.9, v^b_0 = 0.8$, $v^c_0 = 0.4$ and $v^d_0 = 0.3$. Suppose there are two entities* e *and f such that $c^1 = 6$ units and $\mathcal{Z}_{\mathrm{e}} = \mathcal{Z}_f = \{a, b, c, d\}$. The total budget is $\beta = 25$ units. Also, suppose $\Delta^{\mathrm{e}}_{\mathrm{inc}} = \Delta^{f}_{\mathrm{inc}} = \Delta_{\mathrm{dec}} = 0.1$. If an entity (say entity* e*) is to be allocated the largest set of nodes that it can permanently repair, then* e *would be allocated nodes a and b. Then, entity f would be allocated node c as it can only repair one node out of the remaining nodes c and d. Thus, node d would not be allocated to any entity in the aforementioned allocation policy and thus only nodes a, b and c are permanently repaired. However, if the proposed online policy is used then nodes a and b are allocated to the two entities (in any order) at $t = 0$. After permanently repairing node a (resp. b) it is possible to allocate node c (resp. d) to the entity that becomes available. Thus, all the nodes are permanently repaired by the online policy in this example.*

Although the above examples show that finding the optimal policy in general is non-trivial, the proposed online policy is optimal when there is a single entity as discussed next.

Proposition 4.5.1. *Suppose there are $N(\geq 2)$ nodes, $Q = 1$, $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, and Assumption 6 holds. Consider the online policy where at each time-step the healthiest node that has not been targeted before is allocated to the entity if the entity is currently not repairing any node, until there are no more nodes to allocate or the budget runs out. Then, the aforementioned policy is optimal for Problem 5.*

Proof. The proof of this result starts similarly as the proof of Theorem 4.5.1 by considering an optimal policy $A = \{C, D\}$ where the nodes are targeted in the decreasing order of their initial health value in sequencing policy D. Note that since there is only one entity, there will be no time-step T in D where the healthiest node is not targeted by the entity and thus there will be no reduction in the length of D due to the iterative procedure as in Theorem 4.5.1. Since the allocation C satisfies the budget constraint, the result follows. $\square$

We now give an example to show that the online policy is not 1/2-optimal when the deterioration and repair rates are heterogeneous across the nodes.

Example 15. *Consider five nodes a, b, c, d and e such that $v_0^a = 0.8, v_0^b = 0.8, v_0^c = 0.6, v_0^d = 0.6$ and $v_0^{\mathrm{e}} = 0.6$. Suppose there are two entities f and g such that $c^{\mathrm{i}} = \mathrm{i}c^1$, $c^1 = 1$ unit, and $\mathcal{Z}_f = \mathcal{Z}_g = \{a, b, c, d, \mathrm{e}\}$. The total budget is $\beta = 6$ units. The deterioration rates are given by $\Delta_{\text{dec}}^{a,f} = \Delta_{\text{dec}}^{a,g} = \Delta_{\text{dec}}^{b,f} = \Delta_{\text{dec}}^{b,g} = 0.05, \Delta_{\text{dec}}^{c,f} = \Delta_{\text{dec}}^{c,g} = \Delta_{\text{dec}}^{d,f} = \Delta_{\text{dec}}^{d,g} = 0.2$ and $\Delta_{\text{dec}}^{\mathrm{e},f} = \Delta_{\text{dec}}^{\mathrm{e},g} = 0.6$. The repair rates are given by $\Delta_{\text{inc}}^{a,f} = \Delta_{\text{inc}}^{a,g} = \Delta_{\text{inc}}^{b,f} = \Delta_{\text{inc}}^{b,g} = 0.05, \Delta_{\text{inc}}^{c,f} = \Delta_{\text{inc}}^{c,g} = \Delta_{\text{inc}}^{d,f} = \Delta_{\text{inc}}^{d,g} = 0.2$ and $\Delta_{\text{inc}}^{\mathrm{e},f} = \Delta_{\text{inc}}^{\mathrm{e},g} = 0.4$. If the online policy is followed then nodes a and b are allocated to entities f and g (in any order) at $t = 0$ but the remaining nodes permanently fail by the time they are reached (as shown in Table 4.1); so only nodes a and b are allocated and are permanently repaired. However, consider the allocation where the nodes a, c and e are allocated to entity f and nodes b and d are allocated to entity g (note that this allocation satisfies the budget constraint). After the allocation, entity f permanently repairs all the three allocated nodes when it first targets node e, then node c and finally node a (as shown in Table 4.2). Also, entity g permanently repairs the remaining two nodes by first targeting node d and then node b (as shown in Table 4.3). Since all the five nodes are permanently repaired after the aforementioned allocation in comparison to two nodes that* *are permanently repaired by the online policy, the latter policy is not 1/2-optimal when the deterioration and repair rates are heterogeneous across the nodes.*

Table 4.1. Health progression in Example 15 when the online policy is followed.

Time-step (t)	v_t^a	v_t^b	v_t^c	v_t^d	v_t^{e}
0	0.8	0.8	0.6	0.6	0.6
4	1	1	0	0	0

Table 4.2. Health progression in Example 15 when nodes a, c and e are targeted by entity f.

Time-step (t)	v_t^a	v_t^c	v_t^{e}
0	0.8	0.6	0.6
1	0.75	0.4	1
4	0.6	1	1
12	1	1	1

Table 4.3. Health progression in Example 15 when nodes b and d are targeted by entity g.

Time-step (t)	v_t^b	v_t^d
0	0.8	0.6
2	0.7	1
8	1	1

We now provide an example to show that the online policy is not 1/2-optimal when the costs are heterogeneous across the entities.

Example 16. *Consider five nodes a, b, c, d and* e *such that $v_0^a = v_0^b = v_0^c = v_0^d = v_0^{\mathrm{e}} = 0.95$. Suppose there are two entities f and g such that $c_f^{\mathrm{i}} = \mathrm{i}c_f^1$, $c_g^{\mathrm{i}} = \mathrm{i}c_g^1$, $c_f^1 = 1$, $c_g^1 = 5$, and $\mathcal{Z}_f = \mathcal{Z}_g = \{a, b, c, d, \mathrm{e}\}$. Also, the total budget is $\beta = 6$ units. Suppose $\Delta_{\mathrm{inc}}^f = \Delta_{\mathrm{inc}}^g = \Delta_{\mathrm{dec}} = 0.1$. Then, two nodes get allocated to the entities at $t = 0$ in the online policy and the remaining budget after the allocation of these nodes is $\gamma = 6 - 1 - 5 = 0$. Thus, it is not possible to allocate any more nodes to the entities and therefore two nodes are permanently repaired by* *the online policy. However, if all the nodes are allocated to entity f (note that this allocation satisfies the budget constraint), then it permanently repairs all the nodes. Since five nodes are permanently repaired by the aforementioned policy in comparison to two nodes by the online policy, the latter policy is not 1/2-optimal when the costs are heterogeneous across the entities.*

4.6 Conclusions

In this chapter, we characterized policies for allocation and repair sequencing for recovering damaged components after disasters when there are multiple entities available for repair. We characterized the optimal policy for allocation and repair sequencing when the repair rates are sufficiently larger than the deterioration rates and it is possible for each entity to target any node. For the case when the repair rates are sufficiently larger than the deterioration rates and the budget is sufficiently large, we provided a policy that permanently repairs at least half the number of components as that by an optimal policy. For the case when the deterioration rates are larger than or equal to the repair rates, the rates are homogeneous across the components, and the costs charged by the entities for repair are equal, we characterized a policy for allocation and repair sequencing that permanently repairs at least half the number of components as that by an optimal policy.

4.7 Publications

The key contributions of this chapter are based on the following publication:

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V., 2021. Policies for Multi-Agency Recovery of Physical Infrastructure After Disasters, American Control Conference (ACC) 2021

5. POLICIES FOR RECOVERY OF SOCIO-PHYSICAL COMPONENTS AFTER DISASTERS

5.1 Introduction

In the previous chapters, we only focused on the recovery of physical components. Therefore, we now consider the recovery of both the social and physical components. The problem that we focus in this chapter arises in scenarios where a displaced social group decides to return back after a disaster only after a sufficiently large number of groups in its social network return back and some infrastructure components in its neighborhood are repaired. For simplicity, we will mainly focus on the problem where the number of physical components that can be repaired is given and later we consider extensions of the results for the case where the recovery of physical components is a function of their initial health values, deterioration and repair rates, and the repair sequencing decisions that are followed. We start by formally defining the problem for the former case in the next section.

5.2 Problem statement

We refer to the social components and physical components as *social nodes* and *physical nodes*, respectively. In the context of disaster recovery, a social node could represent a social group such as a community or a household, and a physical node could represent an infrastructure component such as roads in an area or a portion of the power network. The set of social nodes is represented by $\mathcal{W}$, where $|\mathcal{W}| = M$. The weight of a social node $\mathrm{j} \in \mathcal{W}$ is denoted by $w_{\mathrm{j}} \in \mathbb{R}_{>0}$; for example, this weight could represent the number of individuals in the corresponding social group. Every social node exists in one of two possible states at each point in time, *active* or *inactive*. Active (resp. inactive) state of a social node represents the scenario where the corresponding social group has returned (resp. is displaced) after a disaster. Since individuals belonging to the same social group (such as a household or tight-knit community) are interconnected, and decisions (such as whether to return after a disaster) are often made collectively by the individuals in the group, we will use a single state to represent the states of all the individuals within a group [7], [93]. The set of physical nodes is represented by $\mathcal{V}$, where $|\mathcal{V}| = N$. Every physical node exists in one of two possible states at each point in time, *opened* or *closed*. Open state of a physical node corresponds to the fully repaired state and the close state of a physical node represents the damaged state of a physical node. Note that this representation of states for the physical nodes is different from the one that is considered in the previous chapters and we also focus on the problem where the physical nodes are modeled in the same way as before towards the end of this chapter.

The relationships between the different social nodes are represented by a directed graph $G = \{\mathcal{W}, \mathcal{E}\}$. An edge $(\mathrm{i}, \mathrm{j}) \in \mathcal{E}$ represents a directed edge starting from social node i and ending in social node j; social node i is an *incoming* neighbor of social node j. There exists a mapping between the social and physical nodes such that each physical node *covers* one or more social nodes. For all $l \in \mathcal{V}$, the (non-empty) set of social nodes that are covered by physical node l is denoted by $\mathcal{Q}_l \subseteq \mathcal{W}$. Also, each social node is covered by at least one of the physical nodes, i.e., $\cup_{l \in \mathcal{V}} \mathcal{Q}_l = \mathcal{W}$. We assume that there are no edges present between the physical nodes.[1]

We assume that time progresses in discrete time-steps capturing the resolution at which decisions are made by the social nodes to become active or not [74]. We index time-steps by $t \in \mathbb{M} = \{0, 1, 2, \ldots\}$. The total number of social nodes that can be activated at time-step 0 due to the constraints on the budget is given by $K_s (\leq M)$. The social nodes that are activated at time-step 0 are referred as the *seed* nodes. In the context of disaster recovery, seed nodes could represent the households that are provided various incentives and aids such as transition assistance, accelerated tax returns, disaster housing assistance, etc., by government agencies that help them to return back [94], [95]. We consider the *progressive* case where an active social node does not switch back to the inactive state [74]. There is a constraint $K_p (\leq N)$ on the total number of physical nodes that are opened. The decision to open a physical node or not is taken at time-step 0. For physical nodes also, we consider the progressive case, i.e., once a physical node opens at time-step 0, it remains opened for all subsequent time-steps. In the context of disaster recovery, that assumption is justified because infrastructure components face accelerated deterioration after disasters [9], so it can

[1]↑We keep the analysis involving dependencies between physical nodes as a future avenue for research.

be assumed that normal deterioration processes do not significantly change the health of a component once it is repaired [78].

The interactions between the different social nodes are derived from the *DLTM* [77] (recall that existing studies such as [77] do not focus on influence maximization under *DLTM* where the spread of influence in the social network is also a function of the recovery of nodes in another network and thus the work presented in this chapter studies influence maximization while considering physical nodes along with social nodes). Specifically, denote the number of incoming neighbors (in graph G) of social node j by η_{j}. Each social node $\mathrm{j} \in \mathcal{W}$ has a threshold $\theta_{\mathrm{j}} \in \mathbb{Z}_{>0}$ such that $1 \leq \theta_{\mathrm{j}} \leq \eta_{\mathrm{j}}$. Let $\eta_{\mathrm{j},t}$ be the number of active incoming neighbors (in graph G) of social node j at time-step t. An inactive social node j at time-step t becomes active at time-step $t+1$ if the number of active incoming neighbors of node j at time-step t is at least equal to θ_{j} (i.e., $\eta_{\mathrm{j},t} \geq \theta_{\mathrm{j}}$), and in our setting, at least one of the physical nodes that covers social node j is in the open state at time-step t. Note that the states of social nodes are guaranteed to reach a steady state after at most M time-steps, because at least one social node gets activated in each time-step until the states stop changing. Therefore, we refer to the total weight of the social nodes that are activated by the end of time-step M as the total weight of the social nodes that *eventually* get activated. Under the assumptions and dynamics described above, we focus on the following problem.

Problem 8. *Given a social network $G = \{\mathcal{W}, \mathcal{E}\}$ of $M(\geq 1)$ social nodes with node weights $\{w_{\mathrm{j}}\}$ and node thresholds $\{\theta_{\mathrm{j}}\}$, and a set $\mathcal{V}$ of $N(\geq 1)$ physical nodes where the covering of social nodes by physical nodes is given by $\{\mathcal{Q}_l\}$, determine $K_s(\leq M)$ social nodes that should be selected as the seed nodes and $K_p(\leq N)$ physical nodes that should be opened in order to maximize the total weight of the social nodes that eventually get activated.*

We first argue that Problem 8 is NP-hard to approximate within any constant factor (in general). After that, we will look at various special cases of this problem and characterize optimal/approximation algorithms to solve them.

5.3 Inapproximability

We first define an α-optimal algorithm [19].

Definition 5.3.1 (α-optimality)**.** *Let C be the optimal value of a maximization problem and $\overline{C}$ be the value computed by an algorithm A. Then, for $\alpha \in (0, 1]$, A is α-optimal if $\overline{C} \geq \alpha C$ for all the instances of the maximization problem.*

We now present the following inapproximability result.

Proposition 5.3.1. *Problem 8 is NP-hard to approximate within a factor $M^{1-\epsilon}$ for any $\epsilon \in (0, 1)$.*[2]

Proof. Consider the instances of Problem 8 where $K_p = N$ (i.e., all the physical nodes can be opened), $w_{\mathrm{j}} = w, \forall \mathrm{j} \in \mathcal{W}$ (i.e., the weights of all the social nodes are the same) and for all $\mathrm{j} \in \mathcal{W}, \theta_{\mathrm{j}} \leq 2$ (i.e., each inactive social node requires one or two active incoming neighboring nodes to get activated). Since all the physical nodes can be opened, such instances of Problem 8 are equivalent to the instances of the influence maximization problem under *DLTM*, which are NP-hard to approximate within a factor of $M^{1-\epsilon}$ for any $\epsilon \in (0, 1)$, where M is the number of social nodes (see Theorem 5 of [76]). Thus, the result follows. $\square$

Although the influence maximization problem under *DLTM* is NP-hard to approximate within any constant factor when each inactive social node requires one or two active incoming neighboring nodes to get activated (i.e., for all $\mathrm{j} \in \mathcal{W}, \theta_{\mathrm{j}} \leq 2$), the paper [76] showed that the problem has a constant factor approximation algorithm when each inactive social node requires only one active incoming neighboring node to become active (i.e., for all $\mathrm{j} \in \mathcal{W}, \theta_{\mathrm{j}} = 1$). Therefore, we will analyze Problem 8 under the following assumption in this chapter.

Assumption 7. *For all* $\mathrm{j} \in \mathcal{W}, \theta_{\mathrm{j}} = 1$.

With regard to the physical nodes, we will consider the scenario where the total number of physical and social nodes are equal and each physical node covers exactly one social node (i.e., there is a bijective mapping between the physical and social nodes). In the context of disaster recovery, this represents the case when a social group such as a household makes the decision to return or not after a disaster depending on whether an infrastructure component

[2]This means that there cannot be a ρ approximation algorithm for Problem 8 such that $\rho \leq M^{1-\epsilon}$ for any $\epsilon \in (0, 1)$, unless P = NP.

such as power connection at its home has been restored or not; thus, there is a one-to-one mapping between the household and the power connection at its home. Therefore, we make the following assumption (along with Assumption 7) in this chapter.

Assumption 8. *$N = M$ and for all $l \in \mathcal{V}, |\mathcal{Q}_l| = 1$, with $\cup_{l\in\mathcal{V}} Q_l = \mathcal{W}$.*

The following result shows that Problem 8 remains challenging *even* under Assumptions 7 and 8.

Proposition 5.3.2. *Problem 8 under Assumptions 7 and 8 is NP-hard.*

The above result follows by noting that by choosing $K_p = N$ in the above problem, the physical nodes are removed from consideration, and we get back to the problem that is proved to be NP-hard by [74] (see Theorem 2.4 of [74]).

Since Problem 8 under Assumptions 7 and 8 is NP-hard, it is not possible to compute the optimal solution in polynomial-time, unless P = NP. Therefore, we characterize approximation algorithms for special cases of the problem in the next section.

5.4 Approximation algorithms

We first provide the definition of *reachability* in a network.

Definition 5.4.1 (Reachability)**.** *A social node $\mathrm{j} \in \mathcal{W}$ is said to be reachable from a set $\mathcal{A} \subseteq \mathcal{W}$, if there exists a path in the graph $G = \{\mathcal{W}, \mathcal{E}\}$ starting from a node $\mathrm{i} \in \mathcal{A}$ and ending in node j.*

Note that in the above definition it is assumed that each node $\mathrm{j} \in \mathcal{W}$ is reachable from itself. We now define $\sigma(\mathcal{A})$ and $\sigma_w(\mathcal{A})$.

Definition 5.4.2. *Let $\mathcal{A} \subseteq \mathcal{W}$ be a set of social nodes and $\mathcal{B}$ be the set of all social nodes that are reachable from the set $\mathcal{A}$. Then we define $\sigma(\mathcal{A}) \triangleq |\mathcal{B}|$ and $\sigma_w(\mathcal{A}) \triangleq \sum_{\mathrm{j}\in\mathcal{B}} w_{\mathrm{j}}$.*

Note that $\sigma_w(\mathcal{A})$ can be exactly computed in polynomial-time (e.g., using Depth First Search (DFS) or Breadth First Search (BFS) [19]). We will now present some useful properties of $\sigma_w(\mathcal{A})$. We first present the following definitions.

Definition 5.4.3. *A set function f is said to be monotone if $f(\mathcal{A} \cup \{\mathrm{j}\}) \geq f(\mathcal{A}), \forall \mathrm{j}, \mathcal{A}$.*

Definition 5.4.4 (Submodular function [96])**.** *A set function f is said to be submodular if $f(\mathcal{A} \cup \{\mathrm{j}\}) - f(\mathcal{A}) \geq f(\mathcal{B} \cup \{\mathrm{j}\}) - f(\mathcal{B}), \forall \mathrm{j}, \mathcal{A}, \mathcal{B}$, when $\mathcal{A} \subseteq \mathcal{B}$ and $\mathrm{j} \notin \mathcal{B}$.*

We have the following result.

Lemma 27. *The function σ_w is monotone and submodular.*

Proof. The proof of monotonicity comes directly from the fact that for any $\mathcal{A} \subseteq \mathcal{W}$ and $\mathrm{j} \in \mathcal{W}$, the total weight of the social nodes that are reachable from the set $\mathcal{A} \cup \{\mathrm{j}\}$ is at least equal to the total weight of the social nodes that are reachable from the set $\mathcal{A}$.

We now prove submodularity. Consider a social node $\mathrm{j} \in \mathcal{W} \setminus \mathcal{B}$ and two sets of social nodes $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{W}$. Let $\mathcal{C}$ be the set of social nodes that are reachable from social node j but not reachable from any node of the set $\mathcal{A}$. Let $\mathcal{D}$ be the set of social nodes that are reachable from social node j but not reachable from any node of the set $\mathcal{B}$. Then, $\mathcal{D} \subseteq \mathcal{C}$ because $\mathcal{A} \subseteq \mathcal{B}$. Then, $\sigma_w(\mathcal{A} \cup \{\mathrm{j}\}) - \sigma_w(\mathcal{A}) = \sum_{\mathrm{i} \in \mathcal{C}} w_{\mathrm{i}} \geq \sum_{\mathrm{i} \in \mathcal{D}} w_{\mathrm{i}} = \sigma_w(\mathcal{B} \cup \{\mathrm{j}\}) - \sigma_w(\mathcal{B})$. Thus, the result follows. ☐

We now present a greedy algorithm (see Algorithm 6) that we will use in the subsequent analysis. Note that Algorithm 6 has polynomial-time complexity because Step 1 involves

Algorithm 6 Greedy selection of social nodes

Suppose Assumptions 7 and 8 hold. Set iteration $\mathrm{i} = 0$ and $\mathcal{A}_0 = \emptyset$.

1: For $\mathrm{i} = 1$ to K_s, do the following.
   - Let $\mathrm{j} \in \mathcal{W} \setminus \mathcal{A}_{\mathrm{i}-1}$ be a social node such that $\mathrm{j} \in \arg\max_{c \in \mathcal{W} \setminus \mathcal{A}_{\mathrm{i}-1}} \sigma_w(\mathcal{A}_{\mathrm{i}-1} \cup \{c\}) - \sigma_w(\mathcal{A}_{\mathrm{i}-1})$ (breaking ties between social nodes by choosing the social node with largest weight). Define $\mathcal{A}_{\mathrm{i}} = \mathcal{A}_{\mathrm{i}-1} \cup \{\mathrm{j}\}$.

2: Output $\mathcal{A}_{K_s}$ and $\sigma_w\left(\mathcal{A}_{K_s}\right)$.

K_s iterations where each iteration involves performing a max operation over an $O(M)$ array and σ_w function can be computed in polynomial-time as argued earlier. We will use the following result in our analysis later (this result is inspired from Theorem 3 of [76]; [76] did not consider weighted social nodes, therefore we state the following result for the sake of completeness).

Lemma 28. *Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $N(\geq 1)$ social nodes and a set $\mathcal{V}$ of N physical nodes. Then, Algorithm 6 is $\frac{\mathrm{e}-1}{\mathrm{e}}$-optimal for Problem 8 when $K_p = N$, where e is the base of the natural logarithm.*

The proof of this result comes from the fact that when $K_p = N$, all physical nodes can be opened, and that σ_w is a monotone submodular function [96].

We now present an approximation algorithm (see Algorithm 7) for Problem 8 under Assumptions 7 and 8, for general $K_p \leq N$.

Algorithm 7 Selection of seed nodes and opening of physical nodes

1: Run Algorithm 6 to obtain the set $\mathcal{A}_{K_s}$.

2: Suppose Assumptions 7 and 8 hold (and $K_p \leq N$). Consider the following cases.

1. If $K_s > K_p$, then we select K_p social nodes with the largest weights among all the nodes in $\mathcal{W}$ as the seed nodes and open the corresponding physical nodes of those seed nodes.
2. If $K_p \geq \sigma(\mathcal{A}_{K_s}) \geq K_s$, then we first select the nodes in the set $\mathcal{A}_{K_s}$ as the seed nodes. After this, we open the physical nodes corresponding to all the social nodes that are reachable from the set $\mathcal{A}_{K_s}$.
3. If $K_s \leq K_p < \sigma(\mathcal{A}_{K_s})$, we first select the set $\mathcal{A}_{K_s}$ as the seed set and open their corresponding physical nodes. We color all the social nodes white, except the seed nodes which are colored black. After this, we simulate the following process to open $K_p - K_s$ additional physical nodes. At every time-step in the simulation, we select the node j with the largest weight among all the white nodes that have at least one black incoming neighbor, color node j black and open the physical node of node j, until K_p physical nodes are open.

Note that Algorithm 7 has polynomial-time complexity because of the following arguments. The first step of the algorithm has polynomial-time complexity because Algorithm 6 is a polynomial-time algorithm. We now focus on the complexity of Step 2 of Algorithm 7. The complexity of case 1 is polynomial-time as $O(K_p)$ operations are required; note that $K_p = O(M)$ as $K_p \leq N = M$. Also, note that $\sigma(\mathcal{A}_{K_s})$ can be computed in polynomial-time as argued earlier. Therefore, case 2 has polynomial-time complexity because all the reachable nodes from $\mathcal{A}_{K_s}$ can be identified in polynomial-time by DFS [19]. Case 3 has polynomial-time complexity because after setting the seed set, the simulation takes $O(K_p - K_s)$ operations. We now prove that Algorithm 7 is an approximation algorithm.

Theorem 5.4.1. *Suppose Assumptions 7 and 8 hold. Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $N(\geq 1)$ social nodes and a set $\mathcal{V}$ of N physical nodes. Let the largest and the smallest weights of the social nodes be w^{max} and w^{min}, respectively. Then, Algorithm 7 is $\left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{w^{max}}{w^{min}}\}\right)^{-1}$-optimal.*

Proof. Suppose $K_s > K_p$. Then, Algorithm 7 gives the optimal solution because the maximum number of social nodes that can eventually be activated is equal to K_p, and K_p social nodes with the largest weights in the set $\mathcal{W}$ are activated by the algorithm.

We now consider the case when $K_s \leq K_p$. Note that $\sigma(\mathcal{A}_{K_s}) \geq K_s$ because each social node is reachable from itself. Thus, there are two subcases when $K_s \leq K_p$: (i) $K_p \geq \sigma\left(\mathcal{A}_{K_s}\right)$ and (ii) $K_p < \sigma\left(\mathcal{A}_{K_s}\right)$. Suppose $K_p \geq \sigma\left(\mathcal{A}_{K_s}\right) \geq K_s$. Then, all the reachable nodes from the seed set $\mathcal{A}_{K_s}$ can eventually be activated. Denote $\mathcal{A}^* \in \arg\max_{|\mathcal{A}| \leq K_s} \sigma_w(\mathcal{A})$ as the optimal seed set when $K_p = N$, and Assumptions 7 and 8 hold. Let C be the optimal value of Problem 8 under Assumptions 7 and 8. Then,

$$\frac{C}{\sigma_w(\mathcal{A}_{K_s})} \leq \frac{\sigma_w(\mathcal{A}^*)}{\sigma_w(\mathcal{A}_{K_s})} \leq \frac{\mathrm{e}}{\mathrm{e}-1},$$

where the left most inequality comes from the fact that $C \leq \sigma_w(\mathcal{A}^*)$ because $\sigma_w(\mathcal{A}^*)$ is the optimal value when Assumptions 7 and 8 hold but there is no constraint on the total number of physical nodes that can be opened, and the last inequality comes from Lemma 28.

Suppose $K_s \leq K_p < \sigma\left(\mathcal{A}_{K_s}\right)$. Note that the number of social nodes that eventually get activated by Algorithm 7 is equal to the number of black colored nodes, which is equal to K_p. Let x be the number of social nodes that eventually get activated by the optimal solution. Denote the optimal value as C and let the total weight of the social nodes that are eventually activated by Algorithm 7 be D. Then, $C \leq x w^{max}$ and $D \geq K_p w^{min}$ from the definitions of w^{max} and w^{min}, respectively. Thus, $\frac{C}{D} \leq \frac{x w^{max}}{K_p w^{min}} \leq \frac{K_p w^{max}}{K_p w^{min}} = \frac{w^{max}}{w^{min}}$ as $x \leq K_p$.

Combining the bounds from each of the above three cases we see that Algorithm 7 is $\left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{w^{max}}{w^{min}}\}\right)^{-1}$-optimal. □

We now provide an example to illustrate Algorithm 7.

Example 17. *Consider a social network as shown in Figure 5.1 with the corresponding weights shown in the parentheses. Suppose that Assumptions 7 and 8 hold. Thus, there*

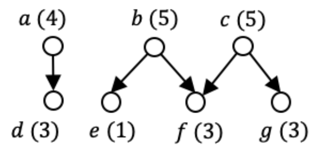

Figure 5.1. Graph for illustrating Algorithm 7.

is a physical node corresponding to each social node but physical nodes are not shown in the figure for simplicity. Suppose $K_s = 2$ *and* $K_p = 4$*. Then, nodes* c *and* a *are selected as the seed nodes when* $\mathcal{A}_{K_s}$ *is determined by Algorithm 6. Thus,* $\sigma(\mathcal{A}_{K_s}) = 5$*. Note that* $\sigma(\mathcal{A}_{K_s}) = 5 > 4 = K_p$*. Thus,* $K_s \leq K_p < \sigma(\mathcal{A}_{K_s})$ *holds and the nodes* $\{a, c, d, f\}$ *are activated when Algorithm 7 is applied with the total weight of the activated nodes being equal to 15 (note that any pair of nodes from the set* $\{d, f, g\}$ *could have been activated after the selection of seed nodes). Note that it is optimal to select nodes* b *and* c *as the seed nodes and open the physical nodes corresponding to* b, c, f *and* g *with the total weight of the activated nodes being equal to 16. Note that although Algorithm 7 is not optimal in this example, it is indeed* $\left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{w^{max}}{w^{min}}\}\right)^{-1} = \left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, 5\}\right)^{-1} = \frac{1}{5}$*-optimal as proved in Theorem 5.4.1.*

We now focus on another instance of Problem 8 under Assumptions 7 and 8. Consider the following assumption.

Assumption 9. *Suppose the graph* $G = \{\mathcal{W}, \mathcal{E}\}$ *is a directed bipartite graph consisting of two sets of social nodes* $\mathcal{I}$ *and* $\mathcal{J}$ *such that* $\mathcal{I} \cup \mathcal{J} = \mathcal{W}$ *and* $\mathcal{I} \cap \mathcal{J} = \emptyset$*. Also, suppose that every edge of the set* $\mathcal{E}$ *starts from a node of the set* $\mathcal{I}$ *and ends in a node of the set* $\mathcal{J}$ *such that there is at least one edge that starts from every node of the set* $\mathcal{I}$ *and there is at least one edge that ends in every node of the set* $\mathcal{J}$*. Suppose the weights of all the nodes of the set* $\mathcal{I}$ *lie in the interval* $[\underline{w}_{\mathcal{I}}, \overline{w}_{\mathcal{I}}]$ *and the weights of all the nodes of the set* $\mathcal{J}$ *lie in the interval* $[\underline{w}_{\mathcal{J}}, \overline{w}_{\mathcal{J}}]$ *such that* $\underline{w}_{\mathcal{I}} \geq \overline{w}_{\mathcal{J}}$*.*

In the context of disaster recovery, Problem 8 under Assumptions 7, 8 and 9 represents the scenario where small communities (represented by the set $\mathcal{J}$) are socially influenced by larger communities (represented by the set $\mathcal{I}$).

Note that Problem 8 under Assumptions 7, 8 and 9 is NP-hard because by setting $K_p = N$, it is possible to open all physical nodes and we get back to the problem that has been proved to be NP-hard in Theorem 2.4 of [74]. We now present the following result.

Theorem 5.4.2. *Suppose Assumptions 7, 8 and 9 hold, i.e., there is a bipartite directed graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $N(\geq 1)$ social nodes such that $\mathcal{I} \cup \mathcal{J} = \mathcal{W}$ and $\mathcal{I} \cap \mathcal{J} = \emptyset$, along with a set $\mathcal{V}$ of N physical nodes. Then, Algorithm 7 is $\left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}}\}\right)^{-1}$-optimal.*

Proof. Note that the proofs for the cases when $K_s > K_p$ and $K_p \geq \sigma\left(\mathcal{A}_{K_s}\right) \geq K_s$ follow in the same way as in the proof of Theorem 5.4.1. Thus, we focus on the case when $K_s \leq K_p < \sigma\left(\mathcal{A}_{K_s}\right)$. Suppose $K_s \leq |\mathcal{I}|$. Then, $\mathcal{A}_{K_s} \subseteq \mathcal{I}$ because the weight of each node $\mathrm{i} \in \mathcal{I}$ is larger than or equal to the weight of each node $\mathrm{j} \in \mathcal{J}$ (as $\underline{w}_{\mathcal{I}} \geq \overline{w}_{\mathcal{J}}$), and for all nodes $\mathrm{j} \in \mathcal{J}$, there is no node $\bar{\mathrm{j}} \in \mathcal{W}$ such that j is an incoming neighbor of $\bar{\mathrm{j}}$. Denote the optimal value as C and let the total weight of the social nodes that are eventually activated by Algorithm 7 be $\overline{C}$. Recall that Algorithm 7 eventually activates K_p social nodes when $K_s \leq K_p < \sigma\left(\mathcal{A}_{K_s}\right)$. Thus, $C \leq K_s\overline{w}_{\mathcal{I}} + (K_p - K_s)\overline{w}_{\mathcal{J}}$ and $\overline{C} \geq K_s\underline{w}_{\mathcal{I}} + (K_p - K_s)\underline{w}_{\mathcal{J}}$ because it is not possible to activate more than K_s nodes in the set $\mathcal{I}$ as no node in the set $\mathcal{I}$ has any incoming neighboring node and $\mathcal{A}_{K_s} \subseteq \mathcal{I}$. Note that

$$\begin{aligned}
\overline{C}\,\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}} &\geq K_s\underline{w}_{\mathcal{I}}\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}} + (K_p - K_s)\underline{w}_{\mathcal{J}}\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}} \\
&\geq K_s\underline{w}_{\mathcal{I}}\overline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}} + (K_p - K_s)\underline{w}_{\mathcal{J}}\underline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}} \\
&\geq C\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}},
\end{aligned}$$

because $\overline{w}_{\mathcal{J}} \geq \underline{w}_{\mathcal{J}}$ and $\overline{w}_{\mathcal{I}} \geq \underline{w}_{\mathcal{I}}$. Therefore, $\frac{C}{\overline{C}} \leq \frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}}$.

Suppose $K_s > |\mathcal{I}|$. Then, $\mathcal{I} \subset \mathcal{A}_{K_s}$ because of the same argument as given for $\mathcal{A}_{K_s} \subseteq \mathcal{I}$ when $K_s \leq |\mathcal{I}|$. Thus, $\mathcal{A}_{K_s}$ contains the top K_s social nodes with largest weight in the set $\mathcal{W}$ because in each iteration $\mathrm{i} > |\mathcal{I}|$ of Algorithm 6, $\sigma_w(\mathcal{A}_{\mathrm{i}-1} \cup \mathrm{j}) - \sigma_w(\mathcal{A}_{\mathrm{i}-1}) = \sum_{k \in \mathcal{W}} w_k - \sum_{k \in \mathcal{W}} w_k = 0, \forall \mathrm{j} \in \mathcal{W} \setminus \mathcal{A}_{\mathrm{i}-1}$ (as for all $\mathrm{i} > |\mathcal{I}|$, $\mathcal{I} \subseteq \mathcal{A}_{\mathrm{i}-1}$ and each node $\mathrm{j} \in \mathcal{W} \setminus \mathcal{A}_{\mathrm{i}-1}$ has an incoming neighbor in the set $\mathcal{I}$) and ties are resolved in Algorithm 6 by choosing the social node with largest weight. Therefore, Algorithm 7 gives the optimal solution because after opening the physical nodes corresponding to the seed nodes in Algorithm 7, physical nodes corresponding to the top $K_p - K_s$ social nodes with largest weight in the set $\mathcal{W} \setminus \mathcal{A}_{K_s}$ are opened (since $\mathcal{W} \setminus \mathcal{A}_{K_s} \subset \mathcal{J}$).

Thus, Algorithm 7 is $\left(\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}}\}\right)^{-1}$-optimal. □

Remark 13. *Note that the parameter α corresponding to Algorithm 7 in Theorem 5.4.2 is greater than or equal to the one characterized in Theorem 5.4.1 for Problem 8 under Assumptions 7, 8 and 9 because the largest and the smallest weights among all the social nodes are $\overline{w}_{\mathcal{I}}$ and $\underline{w}_{\mathcal{J}}$, respectively, and $\frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}} \leq \frac{\overline{w}_{\mathcal{I}}}{\underline{w}_{\mathcal{J}}}$ (as $\underline{w}_{\mathcal{I}} \geq \overline{w}_{\mathcal{J}}$).*

Suppose $\underline{w}_{\mathcal{I}} = \overline{w}_{\mathcal{I}}$ and $\underline{w}_{\mathcal{J}} = \overline{w}_{\mathcal{J}}$. Then, the following result holds by Theorem 5.4.2 because $\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}}\} = \max\{\frac{\mathrm{e}}{\mathrm{e}-1}, 1\} = \frac{\mathrm{e}}{\mathrm{e}-1}$.

Corollary 3. *Suppose $\underline{w}_{\mathcal{I}} = \overline{w}_{\mathcal{I}}$, $\underline{w}_{\mathcal{J}} = \overline{w}_{\mathcal{J}}$ and Assumptions 7, 8 and 9 hold, i.e., there is a bipartite directed graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $N(\geq 1)$ social nodes such that $\mathcal{I} \cup \mathcal{J} = \mathcal{W}$ and $\mathcal{I} \cap \mathcal{J} = \emptyset$, along with a set $\mathcal{V}$ of N physical nodes. Then, Algorithm 7 is $\frac{\mathrm{e}-1}{\mathrm{e}}$-optimal.*

5.4.1 Evaluation of Algorithm 7

We now evaluate the performance of the approximation algorithm that we characterized (Algorithm 7) with a *brute-force method*. In the brute-force method, all the possible combinations of K_s seed nodes and K_p open physical nodes are enumerated to find an optimal solution. Consider the social network as shown in Figure 5.1 that satisfies Assumptions 7, 8 and 9 such that $M = N = 7$, $|\mathcal{I}| = 3$, $\overline{w}_{\mathcal{I}} = 5$, $\underline{w}_{\mathcal{I}} = 4$, $\overline{w}_{\mathcal{J}} = 3$ and $\underline{w}_{\mathcal{J}} = 1$. Suppose $K_s = 2$ and $K_p = 4$. Consider the first row of Table 5.1. Then, the second column of that row shows the computation time (in seconds) for the brute-force method, the third column shows the computation time (in seconds) for Algorithm 7 and the fourth column shows the ratio of the optimal value (computed by the brute-force method) with respect to the value computed by Algorithm 7. In the subsequent rows, we increase M, K_s and K_p such that additional social nodes (and thus physical nodes) are added in the set $\mathcal{J}$ such that Assumptions 7, 8 and 9 hold. We can see that the computation time for the brute-force method increases rapidly with the size of the problem but the increase in the computation time for Algorithm 7 is much slower with the problem size. Also, note that $\max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{\overline{w}_{\mathcal{I}}\,\overline{w}_{\mathcal{J}}}{\underline{w}_{\mathcal{I}}\,\underline{w}_{\mathcal{J}}}\} = \max\{\frac{\mathrm{e}}{\mathrm{e}-1}, \frac{15}{4}\} = 3.75$ for all the considered instances in this example. Thus, the ratio of the value computed by

Algorithm 7 to the optimal value would not be less than or equal to $\frac{1}{3.75} = 0.27$ by Theorem 5.4.2. Therefore, brute-force method is not an efficient method for solving the problem, illustrating the benefit of our approximation algorithms.

Table 5.1. Results when the problem parameters are varied.

Parameters (M, K_s, K_p)	Brute-force time (s)	Algorithm 7 time (s)	Ratio of approx. to optimal value
$7, 2, 4$	0.01	0.004	0.93
$9, 3, 5$	0.07	0.005	1.00
$11, 4, 6$	0.65	0.005	0.87
$13, 5, 7$	9.85	0.005	0.83
$15, 6, 8$	164.14	0.005	0.80

All the instances of Problem 8 that we have considered until now are NP-hard (and thus it is not possible to efficiently compute the optimal solution). We present a special instance of Problem 8 under Assumptions 7 and 8 in the next section that can be optimally solved in polynomial-time.

5.5 Optimal algorithm when the social network is a disjoint union of out-trees

We start by defining an *out-tree* as follows.

Definition 5.5.1. *An out-tree is a directed rooted tree with all the edges pointed away from the root node.*

We make the following assumption (along with Assumptions 7 and 8) in this section.

Assumption 10. $G = \{\mathcal{W}, \mathcal{E}\}$ *is a set of disjoint out-trees (i.e., a forest of out-trees).*

Note that there are several studies that have observed the presence of tree-type hierarchical structures in social networks [97]–[101]. For instance, a social network in the form of a single out-tree could represent multiple social groups that have a hierarchy of power or influence between them, with the social group corresponding to the root node being the most influential [98]. In addition, directed star graphs[3] such as out-stars are frequently

[3]↑A star graph is a tree that has a single node with more than one neighbors.

used in social network analysis to represent bi-level hierarchies between social groups [99], [100]. Also, the paper [101] collected social network information of households in a village and found that the social network forms an out-star during crisis periods (such as disasters) where the root node denotes an influential household of the village.

We first discuss how the presence of physical nodes and weighted social nodes make our problem more challenging and interesting than in the case without physical nodes, even under Assumption 10. Consider the instance of Problem 8 under Assumptions 7, 8 and 10 when $K_p = N$, i.e., it is possible to open all physical nodes. Then, the optimal solution for this case can be easily computed as follows: select the K_s out-trees that have the largest total weight of social nodes, set the root nodes of the selected out-trees as the seed nodes and open all the N physical nodes. Similarly, consider another instance of Problem 8 under Assumptions 7, 8 and 10 when $w_{\mathrm{j}} = w, \forall \mathrm{j} \in \mathcal{W}$, i.e., the weights of all the social nodes are the same (for general $K_p \leq N$). Then, the optimal solution can be easily computed by repeating the following procedure until K_s seed nodes are selected or K_p physical nodes are open: select an out-tree with the largest number of nodes among the set of out-trees that have not been selected before, set the root node j of the out-tree as the seed node, and open the physical nodes corresponding to an arbitrary out-tree of size l that is rooted at node j, where l is the minimum of the size of the selected out-tree and the remaining number of physical nodes that can be opened. However, the optimal strategy is not at all obvious when the social nodes have heterogeneous weights and there is a constraint on the number of physical nodes that can be opened as argued in the next example.

Example 18. *Consider a social network as shown in Figure 5.2 with the corresponding weights shown in parentheses. Suppose that Assumptions 7, 8 and 10 hold such that $M = N = 6$. There is a physical node corresponding to each social node (physical nodes are not shown in the figure for simplicity). Suppose $K_s = 2$ and $K_p = 3$. Since $K_p = 3 < 6 = M$ it is not possible to open all the physical nodes and thus the above algorithm that allowed all physical nodes to open cannot be used to find the optimal solution. Also, if the algorithm that assumed homogeneous weights for the social nodes is used, then node a would be set as a seed node and physical nodes corresponding to an out-tree that is rooted at node a but has three social nodes would be opened. However, the aforementioned solution would not be* *optimal because the optimal solution is to select nodes c and f as the seed nodes and open the physical nodes corresponding to the social nodes c, d and f; the total weight of the activated nodes in the optimal solution is 22.*

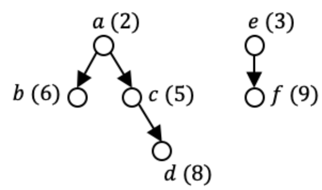

Figure 5.2. Graph for illustrating Example 18.

Note that there may even be social networks where multiple seed nodes are required in one out-tree in the optimal solution (i.e., non-contiguous portions of the out-tree need to be activated), which is not the case in situations without physical nodes. Thus, we will present an algorithm (Algorithm 8) in the following discussion that is optimal for Problem 8 under Assumptions 7, 8 and 10 (even when the weights are heterogeneous across the social nodes and it may not be possible to open all the physical nodes).

In the first step of Algorithm 8, G is modified to another forest $\overline{G}$ by running Algorithm 9 on each out-tree of G. In Algorithm 9, dummy nodes with zero weights are added so that each node in the modified network has at most two outgoing neighbors; the condition that each node has at most two outgoing nodes ensures that Algorithm 8 has polynomial-time complexity (we discuss this later in Remark 14). Figure 5.3 shows a forest G containing a single out-tree on the left-hand side (LHS) where node a has more than two outgoing neighbors; on the right-hand side (RHS) is an out-tree $\overline{G}$ that is generated by adding dummy nodes f and g so that each node has at most two outgoing neighbors. In the second step of Algorithm 8, $\overline{G}$ is modified to an out-tree $\underline{G}$ through the addition of dummy nodes. Finally, Algorithm 10 is run on $\underline{G}$ to obtain an optimal solution in the last step of Algorithm 8. Note that Algorithm 10 is a Dynamic Programming algorithm that first computes the optimal values for the out-trees rooted at the outgoing neighbors of each node v in $\underline{G}$ before computing the optimal value for the out-tree rooted at v; we provide more details on the parameters that are computed in Algorithm 10 later. Note that while choosing a solution in the last step of Algorithm 8, a dummy node is not set as a seed node. Also, physical nodes are not mapped to the dummy nodes in the aforementioned steps and thus the only necessary condition for an inactive dummy node to become active is that at least one of its incoming neighboring nodes should be active.

Algorithm 8 Optimal algorithm when the social network is a forest of out-trees

Consider a social network G that is a forest of out-trees.

1: Run Algorithm 9 on each of the out-trees of forest G to obtain a modified forest $\overline{G}$.

2: If there is a single out-tree in $\overline{G}$, then set $\underline{G} = \overline{G}$ and proceed to the next step. Otherwise, construct an out-tree $\underline{G}$ as follows. Construct a dummy node i with zero weight. Let $\mathcal{L}$ be the set containing all the root nodes of the out-trees in $\overline{G}$. Set node j = i and let x be the number of out-trees in $\overline{G}$. Then, repeat the following until the termination criterion is reached.

- Stop if $x < 2$. If $x = 2$, construct edges starting from node j and ending in all the nodes in the set $\mathcal{L}$. Otherwise, construct an edge starting from node j and ending in an arbitrary node m in the set $\mathcal{L}$. Then, remove node m from the set $\mathcal{L}$. Also, construct a dummy node l with zero weight and an edge starting from node j and ending in node l. Set j $= l$ and $x = x - 1$.

3: Run Algorithm 10 on $\underline{G}$ to find the seed nodes and the physical nodes to open.

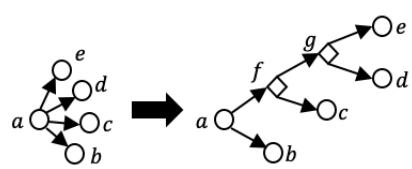

Figure 5.3. Converting an out-tree (left) to another out-tree (right) where each node has at most two outgoing neighbors (the diamonds represent dummy nodes).

We now present the conditions that are used in Algorithm 10. Consider an out-tree $\underline{G} = \{\underline{\mathcal{W}}, \underline{\mathcal{E}}\}$, where each node has at most two outgoing neighbors. Denote the root node of $\underline{\mathcal{W}}$ as r. Let $f_v(k, l)$ be the optimal value of the total weight of eventually activated social nodes for the out-tree rooted at node $v \in \underline{\mathcal{W}}$ when there are at most k seed nodes and at most l open physical nodes in the out-tree rooted at node v. Let $\overline{f}_v(k, l)$ (resp. $\underline{f}_v(k, l)$) be the optimal value of the total weight of eventually activated social nodes for the out-tree rooted at node $v \in \underline{\mathcal{W}}$ when there are at most k seed nodes and at most l open physical

Algorithm 9 Addition of dummy nodes to an out-tree

Consider an out-tree $G^* = \{\mathcal{W}^*, \mathcal{E}^*\}$.

1: Repeat the following until the termination criterion is reached.

- Stop if there is no node in $\mathcal{W}^*$ that has more than two outgoing neighbors. Otherwise, arbitrarily select a node i $\in \mathcal{W}^*$ that has more than two outgoing neighbors. Let $\mathcal{L}$ be the set containing all the outgoing neighbors of i. Remove all the edges between node i and the nodes in $\mathcal{L}$, and set node j = i and $x = |\mathcal{L}|$. Then, repeat the following until the termination criterion is reached.
  - Stop if $x < 2$. If $x = 2$, construct edges starting from node j and ending in all the nodes in the set $\mathcal{L}$. Otherwise, construct an edge starting from node j and ending in an arbitrary node m in the set $\mathcal{L}$. Then, remove node m from the set $\mathcal{L}$. Also, construct a dummy node l with zero weight and an edge starting from node j and ending in node l. Then, set j = l and $x = x - 1$.

nodes in the out-tree rooted at v such that node v is active (resp. inactive). Note that if v is a non-dummy node, then $\overline{f}_v(k,l)$ is only defined when $l \geq 1$; if v is a dummy node, then $\overline{f}_v(k,l)$ is only defined when $v \neq r$,[4] $k \leq K_s - 1$ and $l \leq K_p - 1$ (because the predecessor node of a dummy node needs to be active in order for the dummy node to be active). Let $\overline{f}^a_v(k,l)$ be the optimal value of the total weight of eventually activated social nodes for the out-tree rooted at node $v \in \underline{\mathcal{W}}$ when there are at most k seed nodes and at most l open physical nodes in the out-tree rooted at v such that node v is active but is not a seed node. Note that if v is a non-dummy node, then $\overline{f}^a_v(k,l)$ is only defined when $k \leq K_s - 1$ and $1 \leq l \leq K_p - 1$; if v is a dummy node, then $\overline{f}^a_v(k,l)$ is only defined when $v \neq r$, $k \leq K_s - 1$ and $l \leq K_p - 1$ (as the predecessor node of v should also be active). Let $\overline{f}^b_v(k,l)$ be the optimal value of the total weight of eventually activated social nodes for the out-tree rooted at node $v \in \underline{\mathcal{W}}$ when there are at most k seed nodes and at most l open physical nodes in the out-tree rooted at v such that node v is active along with being a seed node; note that $\overline{f}^b_v(k,l)$ is only defined when $k \geq 1$ and $l \geq 1$, and only for non-dummy nodes because a dummy node cannot be set as a seed node. Let $\mathcal{F}_v(k,l)$ be a set of seed nodes and $\mathcal{H}_v(k,l)$ be a set of physical nodes that are opened in the out-tree rooted at v to obtain $f_v(k,l)$. Also, let $\overline{\mathcal{F}}_v(k,l)$ (resp. $\underline{\mathcal{F}}_v(k,l)$) be a set of seed nodes and $\overline{\mathcal{H}}_v(k,l)$ (resp. $\underline{\mathcal{H}}_v(k,l)$) be a set of physical nodes that are opened in the out-tree rooted at v to obtain $\overline{f}_v(k,l)$ (resp. $\underline{f}_v(k,l)$). Finally, let $\overline{\mathcal{F}}^a_v(k,l)$ (resp. $\overline{\mathcal{F}}^b_v(k,l)$) be a set of seed nodes and $\overline{\mathcal{H}}^a_v(k,l)$ (resp. $\overline{\mathcal{H}}^b_v(k,l)$) be a set of physical nodes that are opened in the out-tree rooted at v to obtain $\overline{f}^a_v(k,l)$ (resp. $\overline{f}^b_v(k,l)$).

Note that for each node $v \in \underline{\mathcal{W}}$, if $\overline{f}_v(k,l)$ is defined and $\overline{f}_v(k,l) \geq \underline{f}_v(k,l)$, then

$$f_v(k,l) = \overline{f}_v(k,l), \mathcal{F}_v(k,l) = \overline{\mathcal{F}}_v(k,l), \mathcal{H}_v(k,l) = \overline{\mathcal{H}}_v(k,l); \tag{5.1}$$

otherwise,

$$f_v(k,l) = \underline{f}_v(k,l), \mathcal{F}_v(k,l) = \underline{\mathcal{F}}_v(k,l), \mathcal{H}_v(k,l) = \underline{\mathcal{H}}_v(k,l). \tag{5.2}$$

[4]↑Recall that the root node can be a dummy node because we start by constructing a dummy node in Step 2 of Algorithm 8 if $\overline{G}$ contains more than one out-tree.

Also, if only $\overline{f}^a_v(k,l)$ is defined or if both $\overline{f}^a_v(k,l)$ and $\overline{f}^b_v(k,l)$ are defined but $\overline{f}^a_v(k,l) \geq \overline{f}^b_v(k,l)$, then

$$\overline{f}_v(k,l) = \overline{f}^a_v(k,l), \overline{\mathcal{F}}_v(k,l) = \overline{\mathcal{F}}^a_v(k,l), \overline{\mathcal{H}}_v(k,l) = \overline{\mathcal{H}}^a_v(k,l); \tag{5.3}$$

otherwise,

$$\overline{f}_v(k,l) = \overline{f}^b_v(k,l), \overline{\mathcal{F}}_v(k,l) = \overline{\mathcal{F}}^b_v(k,l), \overline{\mathcal{H}}_v(k,l) = \overline{\mathcal{H}}^b_v(k,l). \tag{5.4}$$

For each node $v \in \underline{\mathcal{W}}$, if $k = 0$ or $l = 0$,

$$\underline{f}_v(k,l) = 0, \underline{\mathcal{F}}_v(k,l) = \emptyset, \underline{\mathcal{H}}_v(k,l) = \emptyset. \tag{5.5}$$

Let $\underline{\mathcal{W}}_{\mathrm{e}} \subseteq \underline{\mathcal{W}}$ be the set of leaf nodes[5] and $\underline{\mathcal{W}}_d \subseteq \underline{\mathcal{W}}$ be the set of dummy nodes in $\underline{G}$. Note that $\underline{\mathcal{W}}_d \cap \underline{\mathcal{W}}_{\mathrm{e}} = \emptyset$ from the construction of $\underline{G}$. Then, for each $v \in \underline{\mathcal{W}}_{\mathrm{e}}$, if $k \geq 1, l \geq 1$,

$$\underline{f}_v(k,l) = 0, \underline{\mathcal{F}}_v(k,l) = \emptyset, \underline{\mathcal{H}}_v(k,l) = \emptyset, \tag{5.6}$$

$$\overline{f}^a_v(k,l) = \overline{f}^b_v(k,l) = w_v, \overline{\mathcal{F}}^a_v(k,l) = \emptyset, \overline{\mathcal{F}}^b_v(k,l) = v, \overline{\mathcal{H}}^a_v(k,l) = \overline{\mathcal{H}}^b_v(k,l) = \overline{v}, \tag{5.7}$$

where $\overline{v}$ is the physical node corresponding to v. Note that for each $v \in \underline{\mathcal{W}}_{\mathrm{e}}$, if $k = 0, l \geq 1$,

$$\overline{f}^a_v(k,l) = w_v, \overline{\mathcal{F}}^a_v(k,l) = \emptyset, \overline{\mathcal{H}}^a_v(k,l) = \overline{v}. \tag{5.8}$$

For each $v \in \underline{\mathcal{W}}_d$, when $k \geq 0, l = 0$,

$$\overline{f}^a_v(k,l) = 0, \overline{\mathcal{F}}^a_v(k,l) = \emptyset, \overline{\mathcal{H}}^a_v(k,l) = \emptyset. \tag{5.9}$$

Let $\mathcal{L}_v = \{u_1, u_2\}$ be the set of outgoing neighbors of $v \in \underline{\mathcal{W}}$ and $\underline{\mathcal{W}}_{\mathrm{i}} = \underline{\mathcal{W}} \setminus \underline{\mathcal{W}}_{\mathrm{e}}$ be the set of internal nodes in $\underline{G}$. For each $v \in \underline{\mathcal{W}}_{\mathrm{i}}$, $\mathrm{j} \in \{1,2\}$, $0 \leq k_{\mathrm{j}} \leq K_s$ and $0 \leq l_{\mathrm{j}} \leq K_p$, let

[5]↑A leaf node is a node that does not have an outgoing neighboring node.

$g_{u_j}(k_j, l_j) = \max\{\underline{f}_{u_j}(k_j, l_j), \overline{f}^b_{u_j}(k_j, l_j)\}$ if $\overline{f}^b_{u_j}(k_j, l_j)$ is defined, otherwise $g_{u_j}(k_j, l_j) = \underline{f}_{u_j}(k_j, l_j)$. Then, for each $v \in \underline{\mathcal{W}}_i$, if $k \geq 1$, $l \geq 1$,

$$\underline{f}_v(k, l) = \max_{k_1+k_2 \leq k; l_1+l_2 \leq l} \textstyle\sum_{j=1}^2 g_{u_j}(k_j, l_j), \tag{5.10}$$

$$\{k_1^*, l_1^*, k_2^*, l_2^*\} \in \operatorname*{arg\,max}_{k_1+k_2 \leq k; l_1+l_2 \leq l} \textstyle\sum_{j=1}^2 g_{u_j}(k_j, l_j), \tag{5.11}$$

$$\underline{\mathcal{F}}_v(k, l) = \cup_{j=1}^2 \underline{\mathcal{I}}_{u_j}(k_j^*, l_j^*), \underline{\mathcal{H}}_v(k, l) = \cup_{j=1}^2 \underline{\mathcal{J}}_{u_j}(k_j^*, l_j^*), \tag{5.12}$$

where for all $j \in \{1, 2\}$, $\underline{\mathcal{I}}_{u_j}(k_j^*, l_j^*) = \overline{\mathcal{F}}^b_{u_j}(k_j^*, l_j^*)$, and $\underline{\mathcal{J}}_{u_j}(k_j^*, l_j^*) = \overline{\mathcal{H}}^b_{u_j}(k_j^*, l_j^*)$ if $\overline{f}^b_{u_j}(k_j^*, l_j^*)$ is defined and $\overline{f}^b_{u_j}(k_j^*, l_j^*) \geq \underline{f}_{u_j}(k_j^*, l_j^*)$, otherwise $\underline{\mathcal{I}}_{u_j}(k_j^*, l_j^*) = \underline{\mathcal{F}}_{u_j}(k_j^*, l_j^*)$ and $\underline{\mathcal{J}}_{u_j}(k_j^*, l_j^*) = \underline{\mathcal{H}}_{u_j}(k_j^*, l_j^*)$.

Let $\underline{\mathcal{W}}_n = \underline{\mathcal{W}} \setminus \underline{\mathcal{W}}_d$ be the set of non-dummy nodes in $\underline{G}$. For each $v \in \underline{\mathcal{W}}_n \cap \underline{\mathcal{W}}_i$, when $k \geq 0$, $l \geq 1$,

$$\overline{f}^a_v(k, l) = w_v + \max_{k_1+k_2 \leq k, l_1+l_2 \leq l-1} \textstyle\sum_{j=1}^2 f_{u_j}(k_j, l_j), \tag{5.13}$$

$$\{k_1^*, l_1^*, k_2^*, l_2^*\} \in \operatorname*{arg\,max}_{k_1+k_2 \leq k, l_1+l_2 \leq l-1} \textstyle\sum_{j=1}^2 f_{u_j}(k_j, l_j), \tag{5.14}$$

$$\overline{\mathcal{F}}^a_v(k, l) = \cup_{j=1}^2 \mathcal{F}_{u_j}(k_j^*, l_j^*), \overline{\mathcal{H}}^a_v(k, l) = \{\overline{v}\} \cup_{j=1}^2 \mathcal{H}_{u_j}(k_j^*, l_j^*). \tag{5.15}$$

For each $v \in \underline{\mathcal{W}}_d \cap \underline{\mathcal{W}}_i$, when $k \geq 0$, $l \geq 1$, $\overline{f}^a_v(k, l)$ is computed in the same way as when $v \in \underline{\mathcal{W}}_n \cap \underline{\mathcal{W}}_i$ except we set $l = l + 1$ in the RHS of (5.13)-(5.14) and ensure $\overline{\mathcal{H}}^a_v(k, l) = \cup_{j=1}^2 \mathcal{H}_{u_j}(k_j^*, l_j^*)$.

Also, for each $v \in \underline{\mathcal{W}}_n \cap \underline{\mathcal{W}}_i$, when $k \geq 1$, $l \geq 1$,

$$\overline{f}^b_v(k, l) = w_v + \max_{k_1+k_2 \leq k-1, l_1+l_2 \leq l-1} \textstyle\sum_{j=1}^2 f_{u_j}(k_j, l_j), \tag{5.16}$$

$$\{k_1^*, l_1^*, k_2^*, l_2^*\} \in \operatorname*{arg\,max}_{k_1+k_2 \leq k-1, l_1+l_2 \leq l-1} \textstyle\sum_{j=1}^2 f_{u_j}(k_j, l_j), \tag{5.17}$$

$$\overline{\mathcal{F}}_v^b(k,l) = \{v\} \cup_{\mathrm{j}=1}^{2} \mathcal{F}_{u_\mathrm{j}}(k_\mathrm{j}^*, l_\mathrm{j}^*), \tag{5.18}$$

$$\overline{\mathcal{H}}_v^b(k,l) = \{\overline{v}\} \cup_{\mathrm{j}=1}^{2} \mathcal{H}_{u_\mathrm{j}}(k_\mathrm{j}^*, l_\mathrm{j}^*). \tag{5.19}$$

For each $v \in \underline{\mathcal{W}}_\mathrm{i}$ when $|\mathcal{L}_v| = 1$, the above analysis holds by setting $k_2 = 0$ and $l_2 = 0$ in (5.10)-(5.19).

Algorithm 10 Dynamic Programming Algorithm

Consider an out-tree $\underline{G} = \{\underline{\mathcal{W}}, \underline{\mathcal{E}}\}$, where each node has at most two outgoing neighbors.

1: For each node $v \in \underline{\mathcal{W}}_\mathrm{e}$, initialize the values of $f_v(k,l)$, $\mathcal{F}_v(k,l)$, $\mathcal{H}_v(k,l), \forall k \geq 0, l \geq 0$ by (5.1)-(5.8).

2: Repeat the following until the termination criterion is reached:

- Stop if there is no node $v \in \underline{\mathcal{W}}$ for which $f_v(k,l)$ has not been computed for all $k \geq 0$ and $l \geq 0$. Otherwise, arbitrarily select a node v such that for all $\underline{k} \geq 0$ and $\underline{l} \geq 0$, $f_{\underline{v}}(\underline{k}, \underline{l})$ has been computed for each outgoing neighbor $\underline{v}$ of v. Then, compute $f_v(k,l)$, $\mathcal{F}_v(k,l)$, $\mathcal{H}_v(k,l)$ for all k and l using (5.1)-(5.5), (5.9)-(5.19).

3: Let r be the root node of $\underline{\mathcal{W}}$. Then, set the seed nodes as the nodes in the set $\mathcal{F}_r(K_s, K_p)$ and open the physical nodes in the set $\mathcal{H}_r(K_s, K_p)$.

Note that Algorithms 9 and 10 are inspired from the paper [102]; however, the paper [102] focuses on influence maximization problem under *DLTM* when the social network is a directed rooted tree with all the edges pointed towards the root node and thus does not consider the presence of physical nodes and heterogeneous weights for social nodes. Therefore, the presence of physical nodes and weighted social nodes make our problem more challenging and interesting as mentioned before.

We now present the main result of this section.

Theorem 5.5.1. *Suppose Assumptions 7, 8 and 10 hold, i.e., there is a forest $G = \{\mathcal{W}, \mathcal{E}\}$ of out-trees with $N (\geq 1)$ social nodes, along with a set $\mathcal{V}$ of N physical nodes. Then, Algorithm 8 is optimal.*

Proof. Since the Steps 1 and 2 of Algorithm 8 generate an out-tree $\underline{G}$ from G by adding dummy nodes, we first show that Algorithm 10 is optimal for graph $\underline{G}$ by arguing that (5.1)-(5.19) hold. Note that (5.1)-(5.4) trivially hold, so we focus on the other conditions. In condition (5.5), $\underline{f}_v(k,l) = 0$ because node v is inactive from the definition of $\underline{f}_v(k,l)$ and thus it is not possible to activate other nodes in the out-tree of v since either $k = 0$ (i.e., there is no seed node in the out-tree of v) or $l = 0$ (i.e., no physical node is opened in the out-tree of v). Also, $\underline{\mathcal{F}}_v(k,l) = \emptyset$ and $\underline{\mathcal{H}}_v(k,l) = \emptyset$ ensure that $\underline{f}_v(k,l) = 0$, and thus (5.5) holds. Note that (5.6) (resp. (5.7)) holds trivially since v is an inactive (resp. active) leaf node; also note that while computing $\overline{f}^a_v(k,l)$ node v is not set as a seed node but is set as a seed node while computing $\overline{f}^b_v(k,l)$ because of the definitions of $\overline{f}^a_v(k,l)$ and $\overline{f}^b_v(k,l)$. The condition (5.8) follows in the same way as (5.7).

We now focus on the conditions for non-leaf (i.e., interior) nodes. The condition (5.9) follows from the fact that no non-dummy node can be activated in the out-tree rooted at v since $l = 0$. Note that since node v is not active in $\underline{f}_v(k,l)$, functions $\overline{f}^a_{u_1}(k_1,l_1)$ and $\overline{f}^a_{u_2}(k_2,l_2)$ that assume node v (which is the predecessor of u_1 and u_2) is active are not considered in the definitions of $g_{u_1}(k_1,l_1)$ and $g_{u_2}(k_2,l_2)$ in (5.10)-(5.11) and thus (5.10)-(5.12) hold. The remaining conditions (i.e., (5.13)-(5.19)) require that v is an active non-dummy node (thus physical node $\overline{v}$ is opened and then at most $l-1$ physical nodes can be opened corresponding to the remaining social nodes in the out-tree rooted at v). Note that node v is not selected as a seed node in (5.15) but is selected as a seed node in (5.18) due to the definitions of $\overline{f}^a_v(k,l)$ and $\overline{f}^b_v(k,l)$. Next, the computation of $\overline{f}^a_v(k,l)$ when $v \in \underline{\mathcal{W}}_d \cap \underline{\mathcal{W}}_{\mathrm{i}}$, $k \geq 0$ and $l \geq 1$, is similar to that when $v \in \underline{\mathcal{W}}_n \cap \underline{\mathcal{W}}_{\mathrm{i}}$, $k \geq 0$ and $l \geq 1$ with the difference that there is no physical node corresponding to $v \in \underline{\mathcal{W}}_d$ and we thus set $l = l + 1$ in the RHS of (5.13)-(5.14) and ensure $\overline{\mathcal{H}}^a_v(k,l) = \cup_{\mathrm{j}=1}^{2} \mathcal{H}_{u_{\mathrm{j}}}(k^*_{\mathrm{j}}, l^*_{\mathrm{j}})$. Note that when node v has only one outgoing neighbor u_1, then the conditions for the case when there are two outgoing neighbors would hold by not allocating any seed nodes and opening any physical nodes corresponding to the social nodes in the out-tree of u_2. Thus, (5.1)-(5.19) hold and therefore Algorithm 10 computes an optimal solution for $\underline{G}$ from the definitions of $\mathcal{F}_r(K_s, K_p)$ and $\mathcal{H}_r(K_s, K_p)$. Note that an optimal solution for $\underline{G}$ is also optimal for G because the weight of each dummy node is zero, a dummy node is not set as a seed node in $\underline{G}$ and there are no physical nodes corresponding to the dummy nodes. Thus, the result follows. □

Remark 14. *Note that Algorithm 8 is a polynomial-time algorithm because of the following. First, Algorithm 9 has polynomial-time complexity because the outer loop in Step 1 of Algorithm 9 is executed at most $O(M)$ times as there are at most M nodes and the inner loop* *in the Step 1 of Algorithm 9 is also executed at most $O(M)$ times as the maximum number of outgoing neighbors of a node is $M-1$. Thus, Step 1 of Algorithm 8 has polynomial-time complexity as there are at most M out-trees in G. Step 2 of Algorithm 8 also has polynomial-time complexity as the loop in that step is executed $O(M)$ times (as there are at most M root nodes). Finally, Step 3 of Algorithm 8 has polynomial-time complexity because of the following. Let $\underline{M}$ be the number of nodes in $\underline{G}$; note that $\underline{M} = O(M)$ from Steps 1 and 2 of Algorithm 8. Then, the combined complexity of Steps 1 and 2 in Algorithm 10 is $O(\underline{M}K_s^2K_p^2)$ because there are $O(\underline{M}K_sK_p)$ parameters that need to be computed since $k \in \{0, K_s\}, l \in \{0, K_p\}$, and in each computation of parameters such as $\underline{f}_v(k,l)$ and $\overline{f}_v(k,l)$, there are at most $O(K_sK_p)$ comparisons that need to be made;*[6] *note that $K_s = O(M)$ and $K_p = O(M)$ because $K_s \leq M$ and $K_p \leq M$. Finally, Step 3 of Algorithm 10 takes $O(\underline{M})$ operations.*

We now revisit Example 18 to illustrate Algorithm 8.

Example 19. *Consider the instance of Problem 8 focused in Example 18. When Algorithm 9 is applied to that example, graph $\overline{G}$ would be the same as graph G in the first step of the algorithm as each node in G has at most two outgoing neighbors. In the second step, a dummy node r would be added as the root node as shown in Figure 5.4 to form an out-tree $\underline{G}$. Finally, Algorithm 10 is run on $\underline{G}$ to obtain the optimal solution (where nodes c and f are selected as the seed nodes and the physical nodes corresponding to the social nodes c, d and f are opened).*

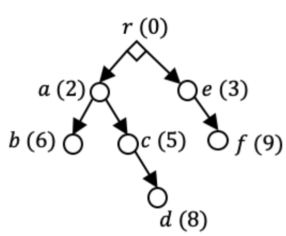

Figure 5.4. Graph obtained by adding a dummy node to the graph of Figure 5.2.

[6]↑Note that $O(K_sK_p)$ comparisons are required as each node $v \in \underline{\mathcal{W}}$ has at most two outgoing neighbors and that is why dummy nodes are added in Algorithm 8 to ensure that each node $v \in \underline{\mathcal{W}}$ has at most two outgoing neighbors.

We assumed so far that the number of physical nodes that can be repaired is given (which is equal to K_p). Thus, we now consider the case when the recovery of physical nodes is a function of their initial health values, deterioration and repair rates, and the repair sequencing decisions that are followed.

5.6 Alternate optimal algorithm when the social network is an out-star

Note that *out-star* is a special type of out-tree where the root node has direct edges with all the other nodes. Thus, the algorithm that is provided in last section would also be optimal when the social network is given by an out-star. In this section, we argue that a slight modification to Algorithm 7 could be used to obtain the optimal solution for out-stars.[7] Consider the modified algorithm as follows.

Algorithm 11 Selection of seed nodes and opening of physical nodes

---

1: Run Algorithm 6 to obtain the set $\mathcal{A}_{K_s}$.

2: Suppose Assumptions 7 and 8 hold (and $K_p \leq M$). Consider the following cases.

1. If $K_s \geq K_p$, then we select K_p social nodes with the largest weights among all the nodes in $\mathcal{W}$ as the seed nodes and open the corresponding physical nodes of those seed nodes.
2. If $K_s < K_p$ and $K_p \geq \sigma\left(\mathcal{A}_{K_s}\right)$, then we first select the nodes in the set $\mathcal{A}_{K_s}$ as the seed nodes. After this, we open the physical nodes corresponding to all the social nodes that are reachable from the set $\mathcal{A}_{K_s}$.
3. If $K_s < K_p$ and $K_p < \sigma\left(\mathcal{A}_{K_s}\right)$, we first select the set $\mathcal{A}_{K_s}$ as the seed set and open their corresponding physical nodes. We color all the social nodes white, except the seed nodes which are colored black. After this, we simulate the following process to open $K_p - K_s$ additional physical nodes. At every time-step in the simulation, we select the node j with the largest weight among all the white nodes that have at least one black incoming neighbor, color node j black and open the physical node of node j, until K_p physical nodes are open.

Theorem 5.6.1. *Suppose Assumptions 7, 8 hold, and, $G = \{\mathcal{W}, \mathcal{E}\}$ is an out-star with $N(\geq 1)$ social nodes, along with a set $\mathcal{V}$ of N physical nodes. Then, Algorithm 11 is optimal.*

[7]↑Note that there are minor differences in terms of the strict versus non-strict inequalities that are considered in the modified algorithm and Algorithm 7. For e.g., $K_s < K_p$ case is considered as $K_s \leq K_p$ here.

Proof. We consider three cases as follows. When $K_s \geq K_p$, it is optimal to select K_p social nodes with the largest weights among all the nodes in $\mathcal{W}$ as the seed nodes and open the corresponding physical nodes of those seed nodes.

We now consider the case when $K_s < K_p$. First, consider the case when $K_p \geq \sigma(\mathcal{A}_{K_s})$. Note that the set $\mathcal{A}_{K_s}$ would contain the root node because the root node has outgoing edges to all the other nodes of the network. Thus, $\sigma(\mathcal{A}_{K_s}) = M$ and therefore $K_p = M$ (as $K_p \geq \sigma(\mathcal{A}_{K_s})$). Therefore, all the nodes can be activated in this case by making the root node as the seed node and opening all the physical nodes (thus the optimal solution is obtained). We now consider the case when $K_p < \sigma(\mathcal{A}_{K_s})$ (and $K_s < K_p$). We first argue that the root node is active in the optimal solution. We prove this by contradiction. Suppose the root node is not active in the optimal solution. Then, all the active nodes are seed nodes as those nodes do not have any incoming neighboring nodes apart from the root node. Let x be the number of active nodes in an optimal solution. Note that $x \leq K_p$ because it is not possible to open more than K_p physical nodes. Suppose $x < K_p$. Then, the root node can be set as a seed node by removing a seed node from one of the activated nodes in the optimal solution and opening the physical node corresponding to the seed node (it is possible to open an additional physical node as $x < K_p$). Then, the total weight of the active nodes increases by the weight of the seed node by the aforementioned modification, leading to a contradiction that the former solution is optimal. Thus, the root node will always be active in an optimal solution. Now consider the case when $x = K_p$. Thus, $K_s < x$ as $K_s < K_p$. This leads to a contradiction because all the active nodes are the seed nodes as mentioned before but it is not possible to have more than K_s seed nodes. Thus, the root node will always be active in an optimal solution. Therefore, if root node is set as the seed node (and its physical node is opened) and the physical nodes corresponding to the $K_p - 1$ largest weight social nodes are opened then we would get an optimal solution because we are maximizing the total weight of the activated nodes except the seed node (note that seed node is always present in an optimal solution so we only need to maximize the total weight of the remaining active nodes).[8] □

[8]↑Note that this algorithm would not work for out-trees in general because activating the root node need not be optimal in some scenarios.

5.7 Repair sequencing of physical nodes

In this section, we focus on the case when the recovery of physical nodes is given by the sequencing dynamics as provided in the previous chapters. Thus, there exist a set of N physical nodes with initial health values in the interval $(0, 1)$. For simplicity, we assume that the deterioration and repair rates of the physical nodes are homogeneous across all nodes and are given by Δ_{dec} and Δ_{inc}, respectively. Suppose a sequence is followed to repair physical nodes and due to this sequence a set $\mathcal{Y} \subseteq \mathcal{V}$ of physical nodes get permanently repaired. Then, the set $\mathcal{Y}$ is equivalent to the "open" physical nodes of previous sections. That is, a social node can only be in active state if its corresponding physical node belongs to the set $\mathcal{Y}$ (since Assumption 8 holds). Note that unlike in the previous sections where the number of physical nodes that can be opened was given (and thus any set of K_p physical nodes could be opened), the set of physical nodes that can be opened in this section is a function of the repair sequencing decisions that are followed and various other parameters of the problem. Thus, the problem that we focus in this section brings more challenges in characterizing optimal or near-optimal policies. We now present the results for special cases of this problem depending on the relationship between the deterioration and repair rates.

5.7.1 Policies for $\Delta^{\mathrm{j}}_{dec} \geq \Delta^{\mathrm{j}}_{\mathrm{inc}}, \forall \mathrm{j} \in \{1, \ldots, N\}$

We first focus on the case when the deterioration rate is larger than or equal to the repair rate. We first argue that non-jumping repair sequences are optimal for physical nodes.

Theorem 5.7.1. *Suppose Assumptions 7 and 8 hold. Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $M(\geq 1)$ social nodes and a set $\mathcal{V}$ of M physical nodes. Suppose $\Delta^{\mathrm{j}}_{\mathrm{dec}} \geq \Delta^{\mathrm{j}}_{\mathrm{inc}}, \forall \mathrm{j} \in \{1, \ldots, N\}$. Then, non-jumping repair sequences are optimal for physical nodes.*

Proof. The proof comes from the fact that for any jumping sequence that permanently repairs x physical nodes of a set $\mathcal{Z}$, there exists a non-jumping policy that permanently repairs all the x physical nodes of the set $\mathcal{Z}$ (by Theorem 2.3.1 of Chapter 2). □

Since non-jumping sequences are optimal when $\Delta_{dec} \geq \Delta_{inc}$, we now focus on providing policies where repair sequences for physical nodes are always non-jumping.

Non-jumping sequences

We make the following assumption in this section.

Assumption 11. *Suppose $\Delta_{\text{dec}} = n\Delta_{\text{inc}}$, where n is a positive integer. Also, for each node $\mathrm{j} \in \{1, \ldots, N\}$, suppose there exists a positive integer m_{j} such that $1 - v_0^{\mathrm{j}} = m_{\mathrm{j}}\Delta_{\text{inc}}$.*

Let $\mathcal{A}$ be a set of social nodes. Denote $\mathcal{F}(\mathcal{A})$ to be the set of physical nodes that cover the set $\mathcal{A}$ (i.e., $\cup_{l \in \mathcal{F}(\mathcal{A})} \mathcal{Q}_l = \mathcal{A}$) and let $\mathcal{S}(\mathcal{A})$ be the set of all the nodes that are reachable from the set $\mathcal{A}$.

We now present the following result.

Lemma 29. *Suppose Assumptions 7, 8 and 11 hold. Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $M(\geq 1)$ social nodes and a set $\mathcal{V}$ of M physical nodes. Let $\mathcal{A}_{K_s}$ be the set that is obtained by Algorithm 6. Let x be the number of physical nodes that are permanently repaired when the policy of targeting the healthiest node at each time-step is followed in the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$. Suppose $x = \sigma(\mathcal{A}_{K_s})$. Then, consider the policy that targets the healthiest node in the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ at each time-step and the nodes in the set $\mathcal{A}_{K_s}$ being selected as the seed nodes. Then, the aforementioned algorithm is $\frac{\mathrm{e}-1}{\mathrm{e}}$-optimal.*

The proof of the above result is similar to the proof of Theorem 5.4.1 (for the case when $K_p \geq \sigma(\mathcal{A}_{K_s}) \geq K_s$).

Remark 15. *Note that the above result would hold for any sequence that permanently repairs all the physical nodes corresponding to the set $\mathcal{S}(\mathcal{A}_{K_s})$. Since we know that under the given conditions on the repair and deterioration rates, the policy that targets the healthiest node at each time-step is optimal (by Theorem 2.3.3), we use this policy in the above result.*

We now present a result for the case when the social network is given by a forest of out-trees.

Lemma 30. *Suppose Assumptions 7, 8 and 11 hold. Let $G = \{\mathcal{W}, \mathcal{E}\}$ be a forest of out-trees with $M(\geq 1)$ social nodes and a set $\mathcal{V}$ of M physical nodes. Denote the set of physical nodes corresponding to the activated social nodes in the solution computed by Algorithm 8 as $\mathcal{B}$. Let x be the number of physical nodes that are permanently repaired when the policy* *of targeting the healthiest node at each time-step is followed in the set $\mathcal{B}$. Suppose $x = |\mathcal{B}|$. Then, consider the policy that targets the healthiest node in the set $\mathcal{B}$ at each time-step and selects the seed nodes from Algorithm 8. Then, the aforementioned algorithm is optimal.*

The proof of the above result is similar to that of Lemma 29.

5.7.2 Policies for $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j} \in \{1, \dots, N\}$

In this section, we focus on the case when the repair rate is larger than the deterioration rate.

Lemma 31. *Suppose Assumptions 7 and 8 hold. Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $M(\geq 1)$ social nodes and a set $\mathcal{V}$ of M physical nodes. Suppose $\Delta^{\mathrm{i_j}}_{\mathrm{inc}} > (M-1)\Delta^{\mathrm{i_j}}_{dec}, \forall \mathrm{j} \in \{1, \dots, M\}$ and $\Delta^{\mathrm{i_j}}_{\mathrm{inc}} > \sum_{k \in \{1,\dots,M\} \setminus \mathrm{j}} \Delta^{\mathrm{i}_k}_{dec}, \forall \mathrm{j} \in \{1, \dots, M\}$. Let x be the largest number such that there exists a set $\{\mathrm{i}_1, \dots, \mathrm{i}_x\} \subseteq \mathcal{V}$ satisfying $v^{\mathrm{i_j}}_0 > (x - \mathrm{j})\Delta^{\mathrm{i_j}}_{dec}, \quad \forall \mathrm{j} \in \{1, \dots, x\}$. Suppose $K_s > x$. We generate a set $\mathcal{Z}$ of x physical nodes with weights equal to the weights of the corresponding social nodes using Step 2 of Algorithm 1 and follow the policy that targets the node with the least modified health value at each time-step in $\mathcal{Z}$. Finally, we set the social nodes corresponding to the set $\mathcal{Z}$ as the seed set. Then, the aforementioned algorithm is optimal.*

The proof of the above result follows directly from the facts that the policy that targets the least healthy node of the set $\mathcal{Z}$ at each time-step is the optimal policy (and thus it is not possible to permanently repair more than x physical nodes in the set $\mathcal{V}$) and also the aforementioned policy permanently repairs all the physical nodes in the set $\mathcal{Z}$ by Theorem 2.4.1. After this, the proof follows in the same way as Theorem 5.4.1 when $K_s > K_p$.

We now present a result where we do not make the assumption of $K_s > x$ as in the above result.

Lemma 32. *Suppose Assumptions 7 and 8 hold. Let there be a graph $G = \{\mathcal{W}, \mathcal{E}\}$ with $M(\geq 1)$ social nodes and a set $\mathcal{W}$ of M physical nodes. Suppose $\Delta^{\mathrm{i_j}}_{\mathrm{inc}} > (M-1)\Delta^{\mathrm{i_j}}_{dec}, \forall \mathrm{j} \in \{1, \dots, M\}$ and $\Delta^{\mathrm{i_j}}_{\mathrm{inc}} > \sum_{k \in \{1,\dots,M\} \setminus \mathrm{j}} \Delta^{\mathrm{i}_k}_{dec}, \forall \mathrm{j} \in \{1, \dots, M\}$. Let x be the largest number such that there exists a set $\{\mathrm{i}_1, \dots, \mathrm{i}_x\} \subseteq \mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ satisfying $v^{\mathrm{i_j}}_0 > (x - \mathrm{j})\Delta^{\mathrm{i_j}}_{dec}, \quad \forall \mathrm{j} \in \{1, \dots, x\}$.*

Suppose $x = \sigma(\mathcal{A}_{K_s})$. Then, the policy that targets the node with the least modified health value in the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ at each time-step is followed. After this, we select seed nodes as the nodes in the set $\mathcal{A}_{K_s}$. Then, the aforementioned algorithm is $\frac{\mathrm{e}-1}{\mathrm{e}}$-optimal.

The proof of the above result follows directly from the facts that the policy that targets the least healthy node of the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ at each time-step permanently repairs all the physical nodes in the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ by Theorem 2.4.1. After this, the proof follows in the same way as Theorem 5.4.1 when $K_p \geq \sigma(\mathcal{A}_{K_s}) \geq K_s$.

Lemma 33. *Suppose Assumptions 7 and 8 hold. Let $G = \{\mathcal{W}, \mathcal{E}\}$ be a forest of out-trees with $M(\geq 1)$ social nodes and a set $\mathcal{V}$ of M physical nodes. Suppose $\Delta^{\mathrm{i_j}}_{inc} > (M-1)\Delta^{\mathrm{i_j}}_{dec}, \forall \mathrm{j} \in \{1, \ldots, M\}$ and $\Delta^{\mathrm{i_j}}_{inc} > \sum_{k \in \{1,\ldots,M\}\setminus \mathrm{j}} \Delta^{\mathrm{i}_k}_{dec}, \forall \mathrm{j} \in \{1, \ldots, M\}$. Denote the set of physical nodes corresponding to the activated social nodes in the solution computed by Algorithm 8 as $\mathcal{B}$. Let x be the largest number such that there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{B}$ satisfying $v^{\mathrm{i_j}}_0 > (x - \mathrm{j})\Delta^{\mathrm{i_j}}_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, x\}$. Suppose $x = |\mathcal{B}|$. Then, the policy that targets the node with the least modified health value in the set $\mathcal{B}$ at each time-step and selects the seed nodes as that obtained from Algorithm 8 is optimal.*

The proof of this result is similar to that of Lemma 32.

Note that the results that we provide for the case when there is sequencing of physical nodes as compared to the previous case (when the number of physical nodes that could be opened was given) are more restricted in terms of the assumptions that we need to make and the cases of the problem for which the results are provided. Thus, we will also provide computational frameworks and heuristics for obtaining reasonable solutions for the general problem.

5.8 Conclusions

In this chapter, we consider a scenario where each physical component covers one or more social components, and a necessary condition to activate a social component is that it should be covered by at least one of the opened physical components. There is a constraint on the total number of physical components that can be opened (in addition to the number of social components that can be chosen as seed components). This problem has applications in contexts such as disaster recovery where a displaced social group may decide to return to its home only if some infrastructure components in its residential neighborhood have been repaired and a sufficiently large number of groups in its social network have returned back. The general problem is NP-hard to approximate within any constant factor and therefore we provided optimal and approximation algorithms for special instances of the problem.

5.9 Publications

The key contributions of this chapter are based on the following publication:

- Gehlot, H., Sundaram, S. and Ukkusuri, S.V. Algorithms for Influence Maximization in Socio-Physical Networks (Under review at IEEE Transactions on Control of Network Systems)

6. COMPUTATIONAL METHODS AND GUIDELINES FOR REAL-WORLD APPLICATION

6.1 Introduction

In this chapter, we will first present various computational methods for solving general instances of the recovery problems that we discussed in the last chapters for which the optimal or near-optimal policies are not currently known. We will first present Mixed Integer Linear Programming (MILP) formulations for various versions of the problem that were discussed in the previous chapters and then provide heuristic methods to efficiently solve these problems. In the second half of this chapter, we will provide guidelines and examples that would help in the real-world implementation of the policies that are characterized in the previous chapters.

6.2 Computational methods

In this section, we present integer programming formulations and heuristics for the recovery problems that are presented in the last chapters. Note that the problems that were focused in the last chapters could be divided into the following two broad classes. The first class of the problems is when we only focused on the recovery of physical components where the objective was to maximize the total number or weight of the physical components that are permanently repaired. We considered various instances of this class in Chapters 2, 3 and 4 such as when there are dependencies in the form of precedence constraints, multiple repair agencies, etc. The second class of problems is where the objective is to maximize the recovery of social components while considering the recovery of physical components. We start by presenting an integer programming formulation for the first class of problems in the next section.

6.2.1 MILP for recovery of physical components

We now present the Mixed Integer Linear Programming formulation for maximizing the recovery of physical components under various constraints.

$$\max \sum_{\mathrm{j}=1}^{N} w^{\mathrm{j}} r_{T+1}^{\mathrm{j}} \tag{6.1}$$

$$\text{s.t.} \quad \sum_{h \in \mathcal{T}} y^{\mathrm{j},h} \leq 1, \;\; \forall \mathrm{j} \in \{1, \ldots, N\}; \tag{6.2}$$

$$y^{\mathrm{j},h} = 0, \;\; \forall \mathrm{j} \notin \mathcal{Z}_h, h \in \mathcal{T}; \tag{6.3}$$

$$\sum_{h \in \mathcal{T}} \sum_{k=1}^{N} (c_h^k - c_h^{k-1}) s^{k,h} \leq \beta; \tag{6.4}$$

$$y^{\mathrm{j},h} \geq x_t^{\mathrm{j},h}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.5}$$

$$\sum_{\mathrm{j}=1}^{N} x_t^{\mathrm{j},h} \leq 1, \;\; \forall t \in \{0, \ldots, T\}, h \in \mathcal{T}; \tag{6.6}$$

$$v_{t+1}^{\mathrm{j}} - v_t^{\mathrm{j}} = \sum_{h \in \mathcal{T}} \Delta_{\mathrm{inc}}^{\mathrm{j},h} x_t^{\mathrm{j},h} - \Delta_{dec}^{\mathrm{j}} \left(1 - \sum_{h \in \mathcal{T}} x_t^{\mathrm{j},h} \right) + \Delta_{dec}^{\mathrm{j}} r_t^{\mathrm{j}}, \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.7}$$

$$x_t^{\mathrm{j},h} \leq z_t^{\mathrm{j}} \overline{M}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.8}$$

$$x_t^{\mathrm{j},h} \leq \left(1 - r_t^{\mathrm{j}}\right) \overline{M}, \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.9}$$

$$-v_t^{\mathrm{j}} < \left(1 - z_t^{\mathrm{j}}\right) \overline{M}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.10}$$

$$-v_t^{\mathrm{j}} \geq -z_t^{\mathrm{j}}\overline{M}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.11}$$

$$v_t^{\mathrm{j}} - 1 < r_t^{\mathrm{j}}\overline{M}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.12}$$

$$v_t^{\mathrm{j}} - 1 \geq -\left(1 - r_t^{\mathrm{j}}\right)\overline{M}, \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.13}$$

$$\sum_{l=1}^{N} y^{l,h} - 1 < s^{\mathrm{j},h}\overline{M}, \;\; \forall \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.14}$$

$$\sum_{l=1}^{N} y^{l,h} - 1 \geq -\left(1 - s^{\mathrm{j},h}\right)\overline{M}, \forall \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.15}$$

$$r_t^{\mathrm{i}} \geq x_t^{\mathrm{j},h}, \; \forall t \in \{0, \ldots, T\}, (\mathrm{i}, \mathrm{j}) \in \mathcal{E}, h \in \mathcal{T}; \tag{6.16}$$

$$r_t^{\mathrm{j}}, z_t^{\mathrm{j}} \in \{0, 1\}, \;\; \forall t \in \{0, \ldots, T+1\}, \mathrm{j} \in \{1, \ldots, N\}; \tag{6.17}$$

$$x_t^{\mathrm{j},h} \in \{0, 1\}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.18}$$

$$y^{\mathrm{j},h}, s^{\mathrm{j},h} \in \{0, 1\}, \;\; \forall \mathrm{j} \in \{1, \ldots, N\}, h \in \mathcal{T}; \tag{6.19}$$

$$v_t^{\mathrm{j}} \in \mathbb{R}, \;\; \forall t \in \{0, \ldots, T\}, \mathrm{j} \in \{1, \ldots, N\}. \tag{6.20}$$

Variable r_t^{j} in (6.1) is equal to one if the health value of component j is greater than or equal to one at the beginning of time-step t, otherwise it is equal to zero.[1] Thus, the objective function is to maximize the total weight of the components that are permanently repaired within a sufficiently large time-step T. Note that $T = T^*$ if there is a finite time

[1]↑Note that we do not constrain the health values to lie in the interval [0,1] for the ease of formulating the MILP although this does not affect the optimality of the solution.

constraint T^* (i.e., $T^* \in \mathbb{N}$), otherwise T can be taken to be any sufficiently large positive integer. Condition (6.2) ensures that each component is allocated to at most one agency. Condition (6.3) ensures that for each agency h, if component j does not lie in the set $\mathcal{Z}_h$, then $y^{\mathrm{j},h} = 0$ (i.e., component j cannot be allocated to agency h). Note that $s^{k,h}$ is equal to one if agency h is allocated at least k components, otherwise it is equal to zero. Thus, condition (6.4) ensures that the total cost that is paid to the agencies does not exceed the budget β. Condition (6.5) ensures that if a component j is not allocated to agency h, then h cannot target component j at any time-step. Variable $x_t^{\mathrm{j},h}$ in (6.6) is equal to one if agency h targets component j at time-step t, otherwise it is equal to zero. Thus, condition (6.6) ensures that each agency targets at most one component at any time. Condition (6.7) ensures that if component j has health value in $(0, 1)$ at the beginning of time-step t and it is targeted by agency h at time-step t then its health value increases by $\Delta_{inc}^{\mathrm{j},h}$ at the beginning of time-step $t + 1$, otherwise it decreases by $\Delta_{dec}^{\mathrm{j}}$ at the beginning of time-step $t + 1$. Let z_t^{j} be equal to zero if the health value of component j is less than or equal to 0 at the beginning of time-step t, otherwise it is equal to one, and $\overline{M}$ be a sufficiently large positive number. Then, the constraints (6.8) and (6.9) ensure that if the health value of component j does not lie in the interval $(0, 1)$ at the beginning of time-step t, then no agency targets component j at time-step t. Conditions (6.10) and (6.11) are definitional constraints for $\{z_t^{\mathrm{j}}\}$, conditions (6.12) and (6.13) are definitional constraints for $\{r_t^{\mathrm{j}}\}$, and conditions (6.14) and (6.15) are definitional constraints for $\{s^{\mathrm{j},h}\}$. Condition (6.16) ensures that if there is a precedence constraint starting from component i and ending in component j, then component i should be permanently repaired at the beginning of time-step t for component j to be targeted at time-step t. Conditions (6.17)-(6.20) ensure that $\{r_t^{\mathrm{j}}\}, \{z_t^{\mathrm{j}}\}$, $\{x_t^{\mathrm{j},h}\}$, $\{y^{\mathrm{j},h}\}$ and $\{s^{\mathrm{j},h}\}$ are binary variables, and $\{v_t^{\mathrm{j}}\}$ are real numbers.

Figure 6.1 shows the variation of average computational time with the number of components; the computational time is averaged over 50 instances of Problem 1 where the deterioration and repair rates and the weights are considered to be homogeneous across the components, there is a single repair agency, there are no dependencies between the components, there are no time and budget constraints, the initial health values and the deterioration and repair rates are generated uniformly random from the interval (0,1), and T is set to 30.

We used IBM CPLEX to solve the MILP [103]. It can be seen that the computational time increases rapidly with the number of components.

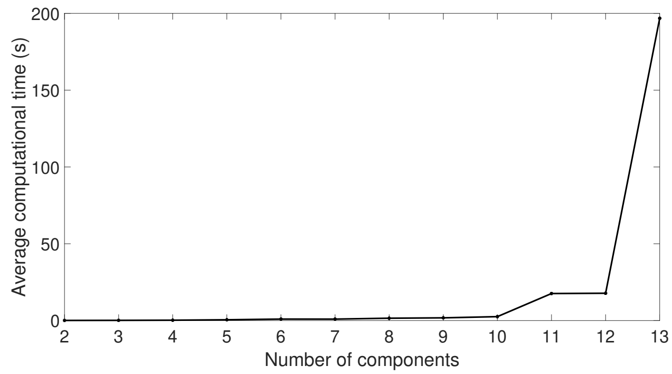

Figure 6.1. Variation of average computational time of MILP with the number of components.

Although the above MILP can be used to find the optimal solutions for the general cases of the recovery of physical components, it becomes computationally burdensome as the number of components increase, illustrating the benefit of explicitly characterizing optimal policies whenever possible. In the next section, we will provide some ways to create greedy policies (motivated by the form of our optimal policies for the cases considered in the previous chapters) for cases where we currently do not have optimal policies.

6.2.2 Heuristics for recovery of physical components

In this section, we provide ways of creating heuristics for general instances of the recovery of physical components. For simplicity, we focus on the case when there is a single repair agency, there are no dependencies between the physical components and there are no time and budget constraints (i.e., Problem 1). But a similar methodology could be used to design heuristics for other variants this problem.

Before presenting the heuristics, we present another computational framework in the form of a Dynamic Program (DP) that can also be used to solve Problem 1.

Dynamic programming formulation

We first start by introducing a Dynamic Programming (DP) formulation for Problem 1 (i.e., when there is a single repair agency, there are no dependencies between the physical components and there are no time and budget constraints). Note that Problem 1 (with the reward definition given in Definition 2.2.2) falls into the class of deterministic total reward problems under an infinite horizon [104]. Let $v \in [0,1]^N$ be the state vector (representing the health values of components) for the entire system at a given time. Denote $J^*(v) = \max_U J(v,w,U)$ (where $J(v,w,U)$ is defined in Definition 2.2.2). Then, the following condition is satisfied by Theorem 7.1.3 of [104],

$$J^*(v) = \max_{u \in \mathcal{V}} \{g(v,u) + J^*(v)\}, \quad \forall v \in \mathcal{S}, \tag{6.21}$$

where v is the state (health value) vector that is obtained via (2.1), $\mathcal{S}$ is the set of all the possible state vectors and $g(v,u)$ is equal to the weight of component u if the state vector at the end of time-step $t-1$ is v, component u is targeted at time-step t and u gets permanently repaired by the end of time-step t; otherwise it is equal to zero.

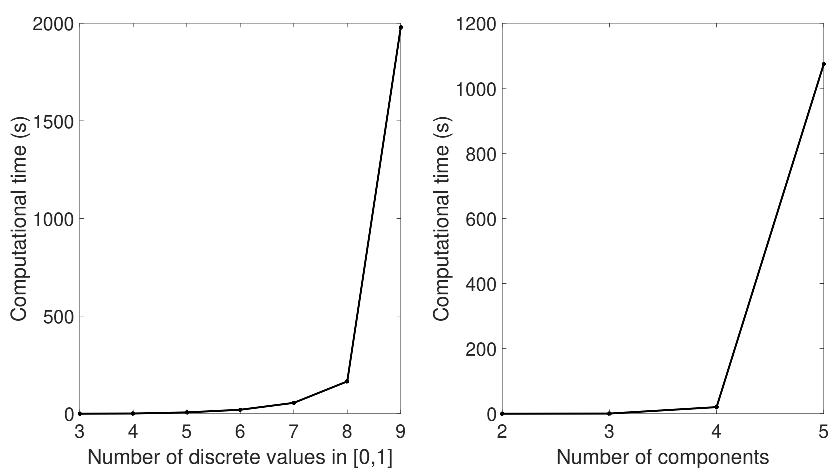

Figure 6.2. Variation of computational time for Value Iteration with the number of discrete values d in the interval [0,1] (left) and with the number of components, where the health value of each components is discretized into $d = 6$ values (right).

We use Value Iteration (VI) to find the value function $J^*(v)$; this method recursively generates a sequence $\{J_1(v), J_2(v), \ldots\}$ of values given the values $J_0(v), \forall v \in \mathcal{S}$ as follows:

$$J_{k+1}(v) = \max_{u \in \mathcal{V}} \{g(v,u) + J_k(v)\}, \forall v \in \mathcal{S}. \tag{6.22}$$

Note that for all $v \in \mathcal{S}$, $J_k(v)$ is guaranteed to converge to $J^*(v)$ by VI when we set $J_0(v) = 0, \forall v \in \mathcal{S}$ (by Theorem 7.2.12 of [104]). Also, note that the implementation of this approach requires discretization of health values in the interval [0,1] and appropriate rounding of the deterioration and repair rates to define the set $\mathcal{S}$. Let $d(\geq 3)$ be the number of discrete values in the interval [0,1] after the discretization. Then, all the deterioration and repair rates are rounded to the nearest multiples of $\frac{1}{d-1}$. Note that $|\mathcal{S}| = d^N$. The left plot in Figure 6.2 shows the variation of computational time with d when we consider the scenario where the deterioration and repair rates as well as the weights are homogeneous across components, $N = 4$, and $\Delta_{dec} = \Delta_{inc} = \frac{1}{d-1}$. It can be seen that the computational time increases rapidly with the number of discrete values. The variation of computational time with the number of components (i.e., N) is shown in the right plot of Figure 6.2 where we consider homogeneous rates and weights, $\Delta_{dec} = \Delta_{inc} = 0.2$ and $d = 6$. It can be seen that the computational time increases rapidly with the number of components.

We now compare the performance of various greedy heuristics (whose details are provided next) with the above computational frameworks (for Problem 1). Specifically, let A be the policy that targets the component with the largest product of health value, weight and repair rate at each time-step. Let B be the policy that targets the component with the least modified health value in the set $\mathcal{Z}$ (as in Theorem 2.4.1) at each time-step. Recall that policy A is optimal when the rates and the weights are homogeneous across components and certain conditions hold (by Theorem 2.3.3) and policy B is optimal when the repair rates are sufficiently larger than the deterioration rates (by Theorem 2.4.1). In each of the following section, we consider 50 scenarios where the initial health values and the weights are generated uniformly randomly from the interval $(0, 1)$ and report the results that are averaged over these scenarios.

Problem 1 under $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}$, and heterogeneous weights and rates

We first consider the case when $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j}$. While we characterized optimal and near-optimal policies for this case when the rates and the weights are homogeneous across components (in Theorems 2.3.3 and 2.3.4), we do not currently have the optimal policies when these parameters are heterogeneous across components. Thus, we evaluate the performance of greedy policies A and B in comparison to the DP and MILP frameworks for this case. Suppose there are five components $\{1, 2, 3, 4, 5\}$ with the deterioration rates $\{0.7, 0.8, 0.48, 0.24, 0.55\}$ and repair rates $\{0.6, 0.5, 0.45, 0.2, 0.54\}$. We consider four subsets of these components to evaluate the performance of policies A and B: the set of two components $\{1, 2\}$, the set of three components $\{1, 2, 3\}$, the set of four components $\{1, 2, 3, 4\}$ and a set of all five components. The first four rows of Table 6.1 present the average reward values that are obtained from the MILP, the DP when $d = 4$, the DP when $d = 5$, and policies A and B. The last four rows of the table present the average computational time (in seconds) taken by the various approaches. It can be seen that the reward values that are obtained from the DP become closer to that obtained from the MILP as d increases. However, DP takes significantly larger computational time than the MILP when d and N are increased (because the total number of states in the DP is $|\mathcal{S}| = d^N$). Also, the reward value that is obtained from policy A is reasonably close to that obtained from the MILP and the computational time taken by this policy is significantly less than that by the MILP and the DP. Thus, policy A could be used as a heuristic for this case. Note that even though policy A seems to perform reasonably well on average for the considered scenarios, it can perform arbitrarily poorly for particular instances of the problem.[2] Thus, characterizing optimal policies for this case remains open. Since solving DP is very computationally burdensome, we provide more extensive tests between policies A and B, and MILP for upto 10 components. We consider ten components $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

[2]↑Suppose there are $N(\geq 6)$ components such that for all $\mathrm{j} \in \{1, \ldots, N\}$, $\Delta_{dec}^{\mathrm{j}} = \Delta_{dec} = \frac{1}{N}$, and $\Delta_{inc}^{\mathrm{j}} = \Delta_{inc} = \frac{1}{N}$. Out of the N components, let there be a set $\mathcal{B}$ with $\lfloor \log_2(N+1) + 1 \rfloor$ components that have initial health values equal to $1 - \Delta_{inc} = 1 - \frac{1}{N}$ and weight equal to 1, and a set $\mathcal{C}$ having the remaining components with initial health values equal to $\Delta_{inc} = \frac{1}{N}$ and weight equal to $N - 2$. Then, policy A would obtain a reward at most equal to $\lfloor \log_2(N+1) + 1 \rfloor$ (from Lemma 2) by targeting the components in the set $\mathcal{B}$ whereas it is optimal to target a component in the set $\mathcal{C}$ to obtain the reward equal to $N - 2$, implying that this policy performs increasingly poorly as the number of components (N) increases.

with the deterioration rates $\{0.7, 0.8, 0.48, 0.24, 0.55, 0.38, 0.64, 0.41, 0.92, 0.78\}$ and repair rates $\{0.6, 0.5, 0.45, 0.2, 0.54, 0.29, 0.55, 0.26, 0.72, 0.68\}$. Then, we consider five sets to evaluate policies A and B: set of six components $\{1, 2, 3, 4, 5, 6\}$, set of seven components $\{1, 2, 3, 4, 5, 6, 7\}$, ..., and set of all ten components. Table 6.2 present the average reward values and average computational time (in seconds) for policies A and B, and MILP. We again can see that policy A performs better than policy B and the computational time for MILP increases significantly with the number of components but that is not the case for the two policies.

Table 6.1. Comparison of greedy policies, MILP, and DP with the number of components varying in the interval [2,5] (the first four rows present the average reward values and the last four rows present the average computational time in seconds).

N	MILP	DP, $d=4$	DP, $d=5$	A	B
2	0.67	0.35	0.43	0.60	0.59
3	0.93	0.48	0.64	0.81	0.80
4	1.20	0.64	0.82	1.05	0.97
5	1.27	0.53	0.76	1.09	0.87
2	0.0536	0.0028	0.0030	0.0003	0.0009
3	0.0964	0.0315	0.1355	0.0003	0.0010
4	0.1642	0.0863	7.692	0.0003	0.0012
5	0.2443	19.2688	276.0743	0.0003	0.0012

Problem 1 under $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc} < (N-1)\Delta^{\mathrm{j}}_{dec}$, and heterogeneous weights and rates

We now focus on the case when the repair rates are moderately larger than the deterioration rates, for which we have not characterized the optimal policy. Thus, we evaluate the performance of various greedy policies for this case. Suppose there are five components $\{1, 2, 3, 4, 5\}$ with the deterioration rates $\{0.3, 0.4, 0.7, 0.75, 0.5\}$ and repair rates $\{0.4, 0.5, 0.8, 0.9, 0.6\}$. We consider four subsets of these components to evaluate the performance of policies A and B: the set of two components $\{1, 2\}$, the set of three components $\{1, 2, 3\}$, the set of four components $\{1, 2, 3, 4\}$ and a set of all five components. Table 6.3 shows the performance (in terms of average reward values) and average computational time

Table 6.2. Comparison of greedy policies, MILP, and DP with the number of components varying in the interval [6,10] (the first five rows present the average reward values and the last five rows present the average computational time in seconds).

N	MILP	A	B
6	1.42	1.22	0.90
7	1.50	1.33	1.02
8	1.52	1.29	0.91
9	1.55	1.35	0.96
10	1.66	1.44	1.14
6	0.390	0.0003	0.0012
7	0.471	0.0003	0.0012
8	0.655	0.0003	0.0012
9	0.668	0.0003	0.0012
10	0.804	0.0003	0.0014

of the MILP, the DP with two values of d, and policies A and B. It can be seen that the reward values obtained from policy B are closest to that obtained from the MILP. Also, the performance of the DP improves with increasing d but the computational time increases rapidly with both d and N. Finally, the MILP provides the largest reward among all the approaches but its computational time increases rapidly with the number of components as seen in Section 6.2.1. Thus, policy B could be used as a heuristic to obtain approximate solutions for this case. Since solving DP is very computationally burdensome, we provide more extensive tests between policies A and B, and MILP for upto 10 components. We consider ten components with the deterioration rates $0.3, 0.4, 0.7, 0.75, 0.5, 0.4, 0.2, 0.8, 0.6, 0.1$ and repair rates $0.4, 0.5, 0.8, 0.9, 0.6, 0.7, 0.3, 0.9, 0.65, 0.4$. Then, we consider five subsets: set of first six components, set of first seven components, ..., and set of all ten components. Table 6.4 present the average reward values and average computational time (in seconds) for the policies A and B, and MILP. We again can see that policy B performs better than policy A and the computational time for MILP increases significantly with the number of components but that is not the case for the two policies.

Thus, policies A and B could be used as heuristics for the cases of Problem 1 where the optimal policies are not currently known.

Table 6.3. Comparison of greedy policies, MILP, and DP with number of components varying in the interval $[2, 5]$ (the first four rows present the average reward values and the last four rows present the average computational time in seconds).

N	MILP	DP, $d = 4$	DP, $d = 5$	A	B
2	0.89	0.41	0.83	0.70	0.89
3	1.25	0.58	1.01	1.00	1.21
4	1.37	0.75	1.10	1.05	1.34
5	1.56	0.97	1.21	1.09	1.50
2	0.0899	0.0020	0.0028	0.0002	0.0009
3	0.1345	0.0407	0.2321	0.0003	0.0013
4	0.1421	1.0451	10.3602	0.0003	0.0014
5	0.1873	20.8674	354.6683	0.0003	0.0014

Table 6.4. Comparison of greedy policies, MILP, and DP with the number of components varying in the interval [6,10] (the first five rows present the average reward values and the last five rows present the average computational time in seconds).

N	MILP	A	B
6	1.71	1.25	1.57
7	1.95	1.38	1.69
8	1.91	1.43	1.68
9	1.99	1.42	1.81
10	2.38	1.64	2.09
6	0.303	0.0003	0.0014
7	0.441	0.0003	0.0014
8	0.496	0.0003	0.0014
9	0.623	0.0004	0.0015
10	1.012	0.0004	0.0015

Heuristics for Problems 2 and 5

The above discussion of heuristics focused on Problem 1 but a similar methodology could be used to design heuristics for other extensions of Problem 1 (i.e., Problems 2 and 5). For instance, the following heuristics (that are motivated from the policies that are characterized for Problem 2) could be used for cases of Problem 2 where the optimal or near-optimal policies are not currently known:

1. Policy that targets the component with the largest product of health value and repair rate at each time-step while respecting the precedence constraints.

2. Policy that targets the component with the least modified health value at each time-step while respecting the precedence constraints.

For Problem 5, the following heuristics (that are motivated from the policies that are characterized for Problem 5) could be used for cases of Problem 5 where the optimal or near-optimal policies are not currently known:

1. Algorithm 3 that allocates components to agencies in the increasing order of their costs using an algorithm that is a modification of Algorithm 2 such that allocation constraints are considered.

2. A modification of Algorithm 5 that stops if the budget runs out (in addition to the existing stopping criteria).

3. *Online* policy where at each time-step the healthiest component that is currently not being targeted is allocated to an agency that is currently not repairing any component given that it is possible to do that allocation, until there are no more components to allocate or the budget runs out.

Until now, we only focused on the recovery of physical components and thus we will now focus on the problem of maximizing the return of displaced social groups while considering the recovery of physical components.

6.2.3 MILP for recovery of socio-physical systems

In this section, we present the integer programming formulation for the recovery problem where the objective is to maximize the total weight of the social groups that return while considering the recovery of physical components (i.e., the problems focused in Chapter 5). Note that the problem where the set of physical components that are repaired is a function of the sequencing decisions that are followed as well as the parameters corresponding to the physical components is more complex than the problem where the number of physical components that can be repaired is given (i.e., Problem 8). Thus, we provide the MILP for the former problem as follows:

$$\max \sum_{\mathrm{j}\in\mathcal{W}} w_{\mathrm{j}} y_N^{\mathrm{j}} \tag{6.23}$$

$$\text{s.t. Conditions (6.2)-(6.20) hold;} \tag{6.24}$$

$$\sum_{\mathrm{j}\in\mathcal{W}} y_0^{\mathrm{j}} = K_s; \tag{6.25}$$

$$y_\tau^{\mathrm{j}} + \sum_{\mathrm{i}\in\mathcal{N}_{\mathrm{j}}} y_\tau^{\mathrm{i}} \geq y_{\tau+1}^{\mathrm{j}}, \ \ \forall\tau \in \{0,\ldots,N-1\}, \mathrm{j}\in\mathcal{W}; \tag{6.26}$$

$$r_{T+1}^{s_{\mathrm{j}}} \geq y_\tau^{\mathrm{j}}, \ \ \forall\tau \in \{0,\ldots,N\}, \mathrm{j}\in\mathcal{W}; \tag{6.27}$$

$$y_{\tau+1}^{\mathrm{j}} \geq y_\tau^{\mathrm{j}}, \ \ \forall\tau \in \{0,\ldots,N-1\}, \mathrm{j}\in\mathcal{W}; \tag{6.28}$$

$$y_\tau^{\mathrm{j}} \in \{0,1\} \ \ \forall\tau \in \{0,\ldots,N\}, \mathrm{j}\in\mathcal{W}. \tag{6.29}$$

Let $y_\tau^{\mathrm{j}} \in \{0,1\}$ be equal to one if social component j is active at the beginning of time-step τ, otherwise it is equal to zero. Then, the objective function in (6.23) is equal to the total weight of social components that are active at the beginning of time-step N (i.e., the total weight of eventually activated social components). Condition (6.24) represents the constraints related to the recovery of physical components. Constraint (6.25) denotes the condition that at the beginning of time-step 0 there are K_s active social components (i.e., there are K_s seed components). Let $\mathcal{N}_{\mathrm{j}}$ be the set of incoming neighbors of social component $\mathrm{j}\in\mathcal{W}$. Constraint (6.26) represents the condition that for a social component j to be active at the beginning of time-step $\tau+1$, a necessary condition is that at least one of its incoming neighboring components or component j itself is active at the beginning of time-step τ (note that we are focusing on simple directed graphs that do not have self loops). Constraint (6.27) represents the condition that if a social component j is active at the beginning of time-step τ, then its corresponding physical component should be permanently repaired at the beginning of time-step $T+1$ (where T is a sufficiently large time-step corresponding to recovery of physical components as in Section 6.2.1). Condition (6.28) ensures that once a social component j becomes active it does not switch back to inactive state. Condition (6.29) ensures that $\{y_\tau^{\mathrm{j}}\}$ are binary variables.

Note that the above formulation is considering the most general form of recovery of physical components (i.e, there could be dependencies between physical components, multiple repair agencies, budget and time constraints, etc.) by considering the constraints (6.2)-(6.20) from Section 6.2.1. Also, the MILP for Problem 8, (i.e., when the number of physical components that can be repaired is given to be equal to K_p) can be obtained by replacing the conditions (6.24) and (6.27) with $\sum_{\mathrm{j}\in\mathcal{W}} y_\tau^{\mathrm{j}} \leq K_p, \forall \tau \in \{0,\ldots,N-1\}$ in the above formulation.

We now provide some heuristics for solving the combined recovery of socio-physical systems because solving computational frameworks like MILPs can be computationally burdensome as the size of problem increases.

Heuristics for socio-physical systems

We provide heuristics for the case when the recovery of physical components is such that there is a single repair agency, no dependencies between physical components, and no time and budget constraints (however, it is possible to extend these heuristics by considering the heuristics for Problems 2 and 5 as provided before).

Let $\mathcal{A}_{K_s}$ be the set of social components that is obtained from Algorithm 6. Denote the set that is reachable from the set $\mathcal{A}_{K_s}$ as $\mathcal{S}(\mathcal{A}_{K_s})$ and the set of physical components of set $\mathcal{S}(\mathcal{A}_{K_s})$ as $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$. The heuristics are as follows:

1. Suppose $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j}$ and the social network is a general directed graph. Let A be the policy that targets the physical component with the largest value of health times the repair rate at each time-step in the set $\mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ and selects the seed components as the set $\mathcal{A}_{K_s}$.

2. Suppose $\Delta_{inc}^{\mathrm{j}} > (N-1)\Delta_{dec}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $\Delta_{inc}^{\mathrm{j}} > \sum_{k \in \{1,\ldots,N\} \setminus \mathrm{j}} \Delta_{dec}^{k}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$ and the social network is a general directed graph. Let x be the largest number such that there exists a set $\{\mathrm{i}_1, \ldots, \mathrm{i}_x\} \subseteq \mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ satisfying $v_0^{\mathrm{i_j}} > (x - \mathrm{j})\Delta_{dec}, \quad \forall \mathrm{j} \in \{1, \ldots, x\}$. Consider a set $\mathcal{Z} \subseteq \mathcal{F}(\mathcal{S}(\mathcal{A}_{K_s}))$ of size x that is constructed using Step 2 of Algorithm 1 by using the weights of social components. Let A be the policy that targets the physical component with the least value of health minus the deterioration rate in set $\mathcal{Z}$ at each time-step and selects the seed components as the set $\mathcal{A}_{K_s}$.

3. Suppose $\Delta_{dec}^{\mathrm{j}} < \Delta_{inc}^{\mathrm{j}} < (N-1)\Delta_{dec}^{\mathrm{j}}, \forall \mathrm{j}$ and the social network is a general directed graph. Then, we follow the same policy as that when $\Delta_{inc}^{\mathrm{j}} > (N-1)\Delta_{dec}^{\mathrm{j}}, \forall \mathrm{j} \in \{1, \ldots, N\}$ and $\Delta_{inc}^{\mathrm{j}} > \sum_{k \in \{1,\ldots,N\} \setminus \mathrm{j}} \Delta_{dec}^{k}, \quad \forall \mathrm{j} \in \{1, \ldots, N\}$ (that is because we found that the heuristic that targets the physical component with the least modified health value in the set $\mathcal{Z}$ at each time-step performs better on average among the tested heuristics in Section 6.2.2).

We also tested the above heuristics on the following example. Consider a social network as shown in 6.3 that contains seven components with the weights of the components shown in the parentheses. There are seven physical components (with a one-to-one mapping with the social components) that are not shown in the figure for simplicity. The initial health of a physical component corresponding to social component j is represented by v_0^{j}. In this example, $\{v_0^a, v_0^b, v_0^c, v_0^d, v_0^{\mathrm{e}}, v_0^f, v_0^g\} = \{0.3, 0.2, 0.9, 0.4, 0.5, 0.7, 0.8\}$. We consider two cases. The first case is when the deterioration rates are larger than or equal to the repair rates. Suppose $\{\Delta_{dec}^a, \Delta_{dec}^b, \Delta_{dec}^c, \Delta_{dec}^d, \Delta_{dec}^{\mathrm{e}}, \Delta_{dec}^f, \Delta_{dec}^g\} = \{0.7, 0.8, 0.4, 0.5, 0.9, 0.65, 0.95\}$ and $\{\Delta_{inc}^a, \Delta_{inc}^b, \Delta_{inc}^c, \Delta_{inc}^d, \Delta_{inc}^{\mathrm{e}}, \Delta_{inc}^f, \Delta_{inc}^g\} = \{0.6, 0.7, 0.3, 0.5, 0.8, 0.6, 0.94\}$, where $\Delta_{dec}^{\mathrm{j}}$ and $\Delta_{inc}^{\mathrm{j}}$ are the deterioration rate and repair rate, respectively, for the physical component corresponding to social component j. For this case, we use the heuristic corresponding to $\Delta_{dec}^{\mathrm{j}} \geq \Delta_{inc}^{\mathrm{j}}, \forall \mathrm{j}$ as described above. Table 6.5 shows the reward values and computational times (in seconds) that are computed with varying values of K_s. It can be seen that the heuristic algorithm performs as good as the MILP when the K_s is less but the performance of heuristic algorithm decreases as K_s increases (however, the ratio of the value that is computed by the heuristic to the value that is computed by MILP does not become less than 1/3). Also, the average computational time for using heuristic is about 0.01 seconds and that for MILP is about 0.4 seconds (note that this difference would rapidly increase as the number of social components increase and thus it is not feasible to use MILP for solving large problems). The second case that we consider is when the repair rates are larger than the deterioration rates. Suppose $\{\Delta^a_{dec}, \Delta^b_{dec}, \Delta^c_{dec}, \Delta^d_{dec}, \Delta^e_{dec}, \Delta^f_{dec}, \Delta^g_{dec}\} = \{0.06, 0.09, 0.08, 0.07, 0.085, 0.05, 0.065\}$ and $\{\Delta^a_{inc}, \Delta^b_{inc}, \Delta^c_{inc}, \Delta^d_{inc}, \Delta^e_{inc}, \Delta^f_{inc}, \Delta^g_{inc}\} = \{0.9, 0.91, 0.99, 0.95, 0.92, 0.93, 0.94\}$. For this case, we use the heuristic corresponding to $\Delta^j_{dec} < \Delta^j_{inc}, \forall j$ as described above. Table 6.6 shows the reward values and computational times (in seconds) that are computed with varying values of K_s. It can be seen that the difference in the reward values that are computed by the heuristic and MILP reduces as K_s increases. Also, the average computational time for using heuristic is about 0.025 seconds and that for MILP is about 0.3 seconds.

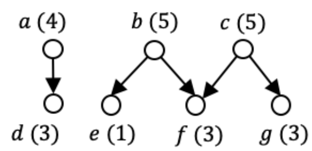

Figure 6.3. A social network for testing heuristics.

Table 6.5. Reward values computed by MILP and heuristic algorithm for different values of K_s when $\Delta^j_{dec} \geq \Delta^j_{inc}, \forall j$.

K_s	MILP	Heuristic algorithm
1	8	8
2	10	8
3	24	8
4	24	8
5	24	8
6	24	8
7	24	8

In the next section, we would present some guidelines and examples from real-world data that would help in the implementation of the results that are presented in this dissertation.

Table 6.6. Reward values computed by MILP and heuristic algorithm for different values of K_s when $\Delta^{\mathrm{j}}_{dec} < \Delta^{\mathrm{j}}_{inc}, \forall \mathrm{j}$.

K_s	MILP	Heuristic algorithm
1	11	5
2	18	12
3	24	24
4	24	24
5	24	24
6	24	24
7	24	24

6.3 Guidelines and examples for implementation of policies

The objective of this section would be to provide guidelines on how to use these models in the real-world. That is, we would create some scenarios and provide guidelines for repair agencies on how to make decisions in those cases.

6.3.1 Recovery of physical components

We first discuss the aspects of the recovery process that only concern the recovery of physical components.

Physical components

In the real-world, there could be some infrastructure components that do not deteriorate further once they are damaged after a disruption. For example, if a road is covered with debris, then it may not face significant damage with time. However, if a road is flooded with water then it would experience deterioration [105]. Thus, the repair agencies can first categorize the components into deteriorating and non-deteriorating components during the inspection. Then, the repair agencies can first focus on the deteriorating components in order to maximize the components that reach permanent repair and after that repair the remaining components.

Also, if the health values of some infrastructure components are highly interdependent, then they should be considered as a single component (rather than as separate components) by assigning the weights of the components accordingly. For instance, some components of the power network may require to be aggregated in order to be relevant for our model because all of them could become functional as soon as one of them works; this could also be the case for transportation networks because as soon as a blocked downstream road becomes functional, all the upstream roads may become functional.

Initial health values

Usually there is a gap between the time infrastructure components get damaged after a disaster and the time when the repair of components starts. For example, the inspection agencies start evaluating damaged infrastructure after a couple of days of the disaster occurrence [106]. In many cases, if it is deemed that repair is necessary after the inspection, then emergency repairs are provided quickly [106]. Thus, the initial health values of the components could be considered as the health values at the starting of the repair period.[3] Also, while evaluating infrastructure components, typically the inspection agencies categorize the damage into several categories such as no damage, light damage, medium damage, heavy damage and complete damage [106]–[108]. In that scenario, the states corresponding to complete damage and no damage can be represented by the health values equal to 0 and 1, respectively, and the remaining damage states can be assigned the health values equal to 0.25, 0.5 and 0.75 depending on the degree of damage (given that no further information is available on the quantification of these damage types). Note that concepts such as *permanent repair* are mentioned in repair manuals, e.g., emergency relief manual of Federal Highway Administration (FHWA) describes *permanent repair* state to be the state that corresponds to the pre-disaster health value of an infrastructure [109], which is analogous to our interpretation of these concepts.

Finally, in some cases such as in flooded regions, it may not be possible to assess the damage of submerged infrastructure such as roads, pipelines, etc. In those cases, flooding depths or storm surge could be used as an indirect measurement for evaluating the damage

[3]↑Note that our models and results are also valid if the repair period starts after a longer duration say weeks or months after the disasters and thus the initial health values should be computed by the damage values at the start of the repair period.

values (or health values). For example, it has been found that higher storm surge is associated with larger damage to bridges [106]. Thus, these types of indirect measures could be used to estimate the initial health values of infrastructure components.

Deterioration rates

As mentioned before, we specifically focus on the repair of infrastructure components that deteriorate with time. Thus, the repair agencies should use the relevant deterioration models for the infrastructure components that they target. For example, if a component represents a set of flooded roads and rutting is used as a measure of the health of a road segment, then rutting models could be used to determine deterioration rates [105]. Note that for some measures like International Roughness Index (IRI), the variation of deterioration with time may not be fixed and in those our models can be considered as approximate methods for computing the deterioration rates (since we assumed that the deterioration and repair rates are fixed with time).

Note that in some repair manuals, e.g., Federal Highway Administration (FHWA), assessment of deterioration processes is usually not stated in damage inspections [109]. Thus, our work explicitly incorporates those processes for increasing the resilience of infrastructure after disasters. There are some studies such as [9] that try to quantify the impact of duration of submergence on various parameters that represent the strength of pavements (note that these parameters are different from rutting and roughness indices). They find that in some types of pavements there may not be significant reduction in strength of pavements with time but in some cases there is significant reduction with time. Thus, even though there may not be significant impact of duration in some cases it is better to quantify the deterioration rate regardless of its magnitude. Note that there are studies such as [110] that acknowledge deterioration processes accelerate after disasters due to processes such as floods. Finally, deterioration processes can also affect other systems like power generation systems, e.g., by corroding the pipelines that are used to transport water through oxidizing and corroding boiler tubes [111], [112].

Repair rates

The repair rate for an agency is the rate at which it recovers physical components. Depending on the type of infrastructure component that is being targeted, repair rate is usually known to the repair agencies and thus estimating this parameter is not difficult in the real-world.

Time-steps

Time-steps denote how frequently a recovery agency decides to repair different components. If a recovery agency decides to daily repair physical components, then a time-step denotes a day but if the decisions are made on a weekly basis then a time-step would be a week. Thus, our problem is time agnostic and the time resolution is dependent on the decision-making frequency of repair agencies.

Weights of physical components

Weights of infrastructure components represent their relative importance. For example, large infrastructure components could be assigned larger weights in comparison to small infrastructure components. Apart from the size of infrastructure components, another way to represent the weights of these components could be to consider their functional importance. For example, average traffic flow on a road could represent the weight of that road, number of houses connected to a gas pipe could represent the importance of that pipe, etc. Weights could also be more precise in terms of the benefit that they offer after repair, e.g., the reduction in overall system travel time a road or a bridge offers after repair could used as a weight. It is possible to estimate such parameters by a black box or a simulator that quantifies the benefit of repairing different bridges/roads.

Precedence constraints

Precedence constraints could represent dependencies as well as sequencing priorities among infrastructure components. For instance, precedence constraints could be used to represent the relationship between transportation and gas networks in an area because some sections of damaged transportation network may required to be first repaired before some sections of the gas pipeline network become functional [113]. Also, repair of inter-states such as I-10, which is a part of the national defence network, would have higher priority over the repair of local roads [106] and thus precedence constraints could also be used for such priorities.

The paper [114] mentions that although there could be several interdependencies between infrastructure components such as coal-fired electric generation network depending on rail supply system for its coal supply, rail supply system depending on electricity for its operation, etc., but in some cases one of these dependencies could be *loose*. For example, it is generally believed that there would be a local supply of coal in the power network that could last for a few months and thus the dependence of electric generation on rail system is loose. Therefore, focusing on unidirectional dependencies in form of precedence constraints makes sense in many cases from the real-world point of view.

Allocation of components to agencies

The emergency relief manual of FHWA and other reports provide low cost and less time as some of the main factors that emergency authorities take into account while deciding to bid the components to different agencies [109], [115]. These studies refer to this rule as A+Bx bidding where A is the cost that is charged, B is the time its takes for repair and x is a factor that balances the two criteria. Although the problem that we focus in Chapter 4 focuses on maximizing the number of components that are permanently repaired within the budget constraint, a formulation that minimizes cost could also be formulated.

Sequencing decisions

These decisions correspond to the different sequences that are followed in practice for the repair of infrastructure components. In practice, for scenarios such as repair of roads, first the debris and water are cleared, then some tests are carried out to inspect the damage and finally the repair starts. In some scenarios, it is required to wait for a significant amount of time for the water to clear off before the recovery process can start but that need not be a constraint as pumps can be used to remove water [116]. Note that the initial health values in our model, correspond to the health values of the components at the time the repair process starts and therefore these values should be estimated at the appropriate time. Usually, information on the sequencing decisions that are followed in the real-world containing information on the constraints on resources and manpower, if any, is not fully available in online resources such as media reports, research papers, etc., but partial information is sometimes available online. For example, Virginia Department of Transportation (VDOT) worked on repairing two roads that were damaged by flash floods of July 2019 concurrently [117].

Example of bridge damage after Hurricane Ike

We collected data from bridge damage after Hurricane Ike in the Houston/Galveston Region [107]. The paper [107] focused on 53 bridges that were damaged due to a number of reasons such as storm surge, wave loading, scour, etc. We focused on the bridges that were damaged by scour[4] since scour represents a deterioration process that occurs over a reasonably long period of time in comparison to other factors such as impact damage. There were 24 bridges that faced scour as one of the factors for their damage. Out of these bridges, we did not focus on 3 bridges that were classified as either service or county roads because county roads are managed by county department and service roads are usually not classified as roads because they bridge one road to another and have different standards from the usual roads. Out of the 21 bridges that we focused, 15 of them faced heavy damage, 3 of them faced medium damage, and the remaining 3 bridges faced light damage. We did not have further information on the damage information for these bridges and therefore we evenly divided the interval [0,1] to represent heavy, medium and light damage with the initial health values equal to 0.25, 0.5 and 0.75, respectively.

We first consider the case when there is a single agency. Since there is no information that is available on the deterioration and repair rates, we assumed the case of homogeneous rates and focused on the five values of $[\Delta_{dec}, \Delta_{inc}]$ as follows: $[0.25, 0.25]$, $[0.195, 0.375]$, $[0.14, 0.5]$,

[4]↑Scour is removal of sediments like sand, gravel around bridge abutments and piers caused by swiftly moving water.

$[0.085, 0.625], [0.03, 0.75]$; note that the left most choice of $[\Delta_{dec}, \Delta_{inc}] = [0.25, 0.25]$ comes from the fact that the initial health values take the values $0.25, 0.5$ and 0.75 and thus we want to have the rates such that no component gets permanently repaired partway through a time-step when the policy of targeting the healthiest component at each time-step is followed; the right most rates satisfy the condition $\Delta_{inc} > (N-1)\Delta_{dec}$ and the intermediate values are uniformly spaced between the extreme values. Also, let A be the policy that targets the healthiest component at each time-step, B be the policy that targets the least healthy component at each time-step and C be the policy that targets the least healthy component in the set $\mathcal{Z}$ at each time-step (recall that set $\mathcal{Z}$ is generated from Algorithm 1). Figure 6.4 presents the number of components that are permanently repaired by policies A, B and C (with colors red, blue and green, respectively) and the reward obtained by MILP (in black) for different values of the deterioration and repair rates. Note that policy A is the optimal policy when the deterioration rate is larger than or equal to the repair rate and it can be seen that this policy permanently repairs the largest number of components among all the policies (and equal to the optimal value) when $\Delta_{dec} = \Delta_{inc} = 0.25$. Also, policies B and C are optimal when the repair rate is sufficiently larger than the deterioration rate, which is satisfied for the case when $\Delta_{dec} = 0.03$ and $\Delta_{inc} = 0.75$ (as $0.75 > (20)(0.03)$); it can be observed that policies B and C permanently repair the same number of components (and equal to the optimal value). For the remaining three intermediate cases, i.e., when $\Delta_{dec} < \Delta_{inc} \leq (N-1)\Delta_{dec}$, policy A permanently repairs the largest number of components when $\Delta_{dec} = 0.195$ and $\Delta_{inc} = 0.375$ whereas policy C permanently repairs the largest number of components for the remaining two cases. Also, we observe that policy C performs better than or equal to policy B for all the cases that we consider here.

We now consider the case when the weights for the infrastructure components (i.e., bridges in this example) are heterogeneous. We used the Annual Average Daily Traffic (AADT) on different types of roads to estimate the weights for the components. Note that the paper [107] focuses mainly on the bridges in rural areas, thus we used the maximum AADT values for different types of roads from the report [118] to compute the weights for the bridges. Note that the maximum AADT values for the local, connector and arterial roads are 400, 1855 and 8500, respectively. Thus, we assign the weights equal to 1, 4.6375, and

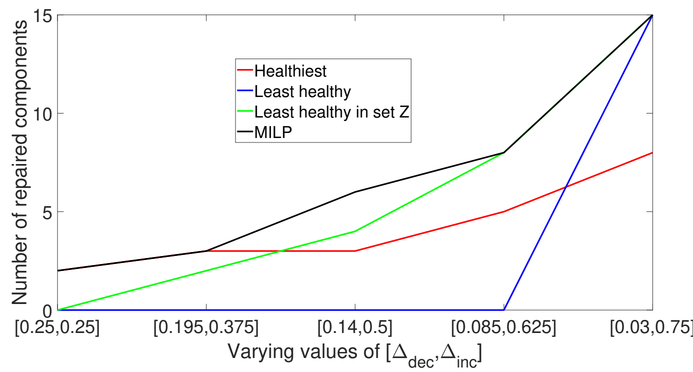

Figure 6.4. Number of components that are permanently repaired by different policies.

21.25 for local, connector and arterial roads, respectively. Let A be the policy that targets the healthiest component at each time-step, B be the policy that targets the component with the largest value of health times the weight at each time-step and C be the policy that targets the least healthy component in the set $\mathcal{Z}$ at each time-step (set $\mathcal{Z}$ is generated from Algorithm 1). Figure 6.5 presents the total weight of the components that are permanently repaired by policies A, B and C (with colors red, blue and green, respectively) and the reward obtained by MILP (in black) for different values of the deterioration and repair rates. Note that unlike the case when the weights are homogeneous, we have not characterized the optimal policy when the weights are heterogeneous and the deterioration rate is larger than or equal to the repair rate. It can be seen that policy B gives the largest value of reward when $\Delta_{dec} = \Delta_{inc} = 0.25$ (in fact it gives the optimal value in this example). Also, policy B performs better than policy A for all the considered cases. However, policy C performs the best when $\Delta_{dec} = 0.03$ and $\Delta_{inc} = 0.75$ because we have characterized it to be optimal when the repair rate is sufficiently larger than the deterioration rate. Finally, neither of the three policies completely dominate the intermediate three cases when $\Delta_{dec} < \Delta_{inc} \leq (N-1)\Delta_{dec}$ but the the largest values obtained among them perform reasonably well in comparison to that obtained by MILP. Note that although we provide the reward obtained through MILP for this example, some of the instances that we tested took quite a long time to be computed. For example, when $\Delta_{dec} = 0.085$ and $\Delta_{inc} = 0.625$ and the component weights are heterogeneous, it took about 15 minutes to obtain the value from MILP which is much larger than the few fractions of the seconds that it takes to obtain the solutions by any of these policies.

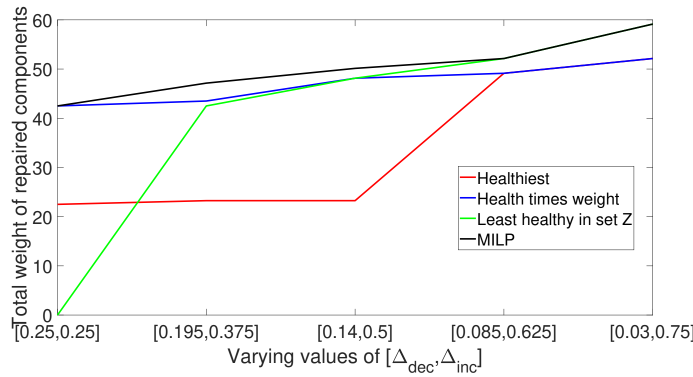

Figure 6.5. Total weight of the components that are permanently repaired by different policies.

Example with multiple repair agencies

In this section, we consider the case when there are multiple repair agencies in the example that was presented in the last section. That is, we focus on 21 physical components corresponding to the bridges that are damaged after Hurricane Ike. We focus on the case when the weights, and deterioration and repair rates are homogeneous across all components and the repair rates are homogeneous across all repair agencies. We did not have the information regarding the costs and the corresponding bids that were made for the repair of these bridges. Thus, in order to get an estimate of the bids that are charged to repair such infrastructure components, we referred to the paper [119] that provides the bids that were placed for repairing I-10 Twin Spans after Hurricane Katrina. The prices for the three bids that were placed are equal to USD 31, USD 38 and USD 93. In the example that we consider in this section, we focus on the case when there are two repair agencies that charge 31 and 38 units for repair. We evaluate the performance of following two policies in this section. Let A be the the online policy where at each time-step the healthiest component that is currently not being targeted is allocated to an agency that is currently not repairing any component given that it is possible to do that allocation, until there are no more components to allocate or the budget runs out. Let B be Algorithm 3 (where sets of components are allocated to the agencies in the increasing order of their costs).

We first focus on the case when the budget that is available with the central authority is very large to ensure that the budget constraint is not relevant in determining the optimal solution (e.g., suppose $\beta = 38*21 = 798$). Figure 6.6 presents the total number of components that are permanently repaired by policies A and B, and MILP. It can be seen that the values that are obtained by MILP in this figure are larger than or equal to that obtained in Figure 6.4 (i.e., when there is a single agency). Also, policy A performs better than policy B when we are towards the left side in Figure 6.6 (i.e., closer to the case when $\Delta_{dec} \geq \Delta_{inc}$) and this trend changes as we move towards the right side in the figure.

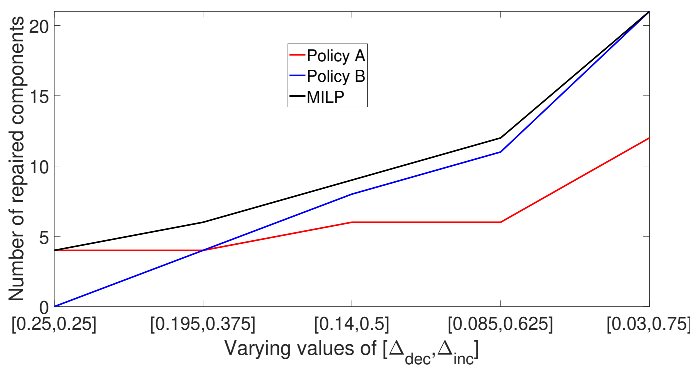

Figure 6.6. Number of components that are permanently repaired by different policies when the budget is sufficiently large.

We now consider the case when the budget $\beta = 200$ units. Figure 6.7 presents the total number of components that are permanently by policies A and B, and MILP, for this case. We can see that number of components that are permanently repaired by all the three approaches reduce in comparison to the values in Figure 6.7 due to the presence of a smaller budget. Also, an interesting observation is that policy A performs better than policy B for all the considered cases, a phenomenon that is not observed in the case when the budget is sufficiently large (in that case policy B performs better than or equal to policy A in most of the cases).

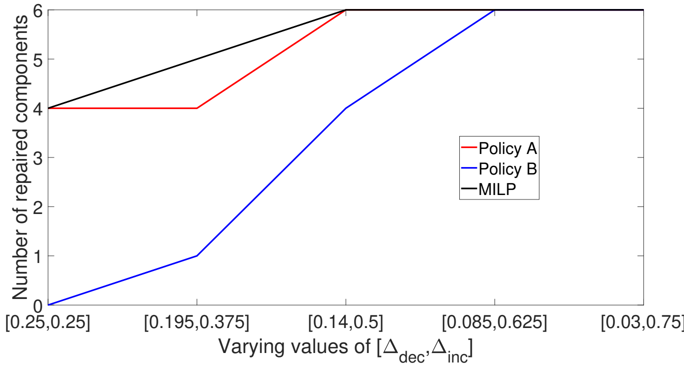

Figure 6.7. Number of components that are permanently repaired by different policies when $\beta = 200$.

Example with precedence constraints

We now consider an example where there are precedence constraints between the physical components. The paper [114] provides an example of infrastructure dependencies from California. Figure 6.8 provides a graph depicting dependencies between a set of infrastructure components. Component a represents power generation, b represents natural gas production, c is for steam generation, d is heavy oil production, e represents transmission of refined crude oil like gasoline, f represents transportation infrastructure that depends on oil products such as vehicle fuels, and g represents electric water pumps. Note that we slightly modified the dependencies that are provided in [114] as follows: in some cases the dependency of power generation on gas supply could be loose and thus we do not consider an edge starting from component b and ending in component a; we combined the effect of reduced transmission of oil products like gasoline on different transportation networks like air transportation and road transportation to a single component for transportation; and removed some of the infrastructure components such as agriculture and banking for which it is hard to associate a deterioration process.

Suppose that the components in Figure 6.8 have equal weights. We consider a scenario after wildfires in California, where 80% of the infrastructure components are severely damaged, 15% of them are mildly damaged and the remaining 5% faced little or no damage [120]. Thus, we consider 50 instances where the initial health values of the components

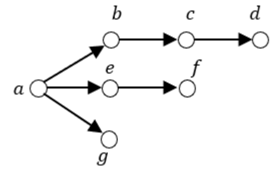

Figure 6.8. A graph depicting precedence constraints.

are randomly generated from the following distribution: a value is generated from the set $\{0.1, 0.2, 0.3, 0.4\}$ with probability 0.8, a value is generated from the set $\{0.5, 0.6, 0.7, 0.8\}$ with probability 0.15 and a value equal to 0.9 is selected with probability 0.05. Let A be the policy that targets the healthiest component at each time-step while respecting the precedence constraints and B be the policy that targets the least healthy component at each time-step while respecting the precedence constraints (we have argued before that these policies could be used as heuristics for the cases where the optimal or near-optimal policies are not currently known). Also, we consider five sets of $[\Delta_{dec}, \Delta_{inc}]$ as follows: $[0.1, 0.1], [0.0875, 0.175], [0.075, 0.25], [0.0625, 0.325], [0.05, 0.4]$. Then, Figure 6.9 shows the average number of components that are permanently repaired over the considered instances for policies A and B (shown by lines red and blue, respectively) and the average reward obtained by MILP (shown by black line) for different values of the rates. It can be seen that policy A permanently repairs more components than that by policy B when $\Delta_{dec} = \Delta_{inc} = 0.1$ (recall that we characterized policy A to be near-optimal in this case). Similarly, policy B permanently repairs more components than that by policy A when $\Delta_{dec} = 0.05$ and $\Delta_{inc} = 0.4$ (recall that we characterized policy B to be near-optimal in this case). Note that the maximum difference between the policies is also about one component so it is not that large for our case where there are a total of 7 components. Also, the best reward values from the two policies perform reasonably close to the rewards obtained by MILP. Finally, for a given value of the deterioration and repair rates, the total time taken by the two policies is around 0.03 seconds whereas for MILP it is about 4-15 seconds. Thus, the above policies could be used as efficient heuristics for computing approximate solutions.

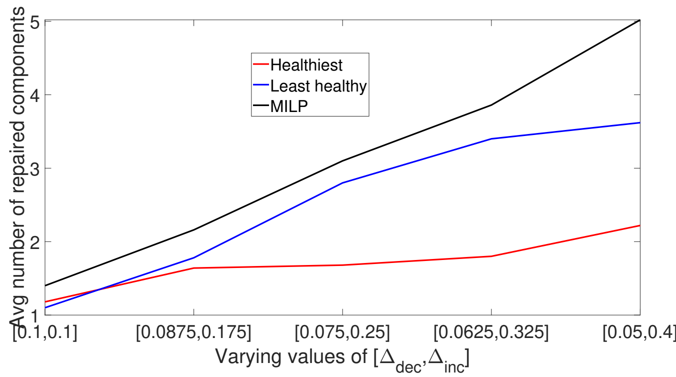

Figure 6.9. Average number of components that are permanently repaired by different policies.

6.3.2 Recovery of socio-physical systems

We now discuss the aspects of the recovery process that arise with the consideration of social components.

Social network

Social network can represent interactions between any set of social groups that influence each other on the matters related to decision making such as whether to return back after a disaster or not. An example of a social network under crisis situations such as disasters is provided by the paper [101], where the social network is represented by an out-star.

Incentivization

There are some incentives and benefits that are provided by government agencies like IRS in USA in terms of accelerated tax refunds to people who have been affected by disasters as a helping hand in rebuilding after such events [95]. These agencies also offer deductions for taxes for damaged properties like homes, car, personal belongings, etc [121]. Note that the aforementioned tax benefits are only available when people live in "federally declared disaster area". FEMA also provides housing assistance for those who have lost home due to presidentially declared disaster in addition to providing assistance to several other necessary expenses [122]. Some agencies like USAID also provide return and transition assistance in form of transportation to return home, help for rebuilding home, etc. [94]. In some cases, it could happen that the funding is limited and in that case the agencies would have to strategically provide the incentives.

Examples for social recovery

We consider a social network consisting of 39 households from a village in Kerela, India [101]. In the crisis period, the social network is represented by an out-star where one influential household influences the remaining households in the village. Note that it is possible to efficiently compute the optimal solution for an out-star using either Algorithm 8 or Algorithm 11 (given that a necessary condition for an inactive social group to become active is that at least one of its incoming neighbors should be active and there is a one-to-one mapping between the social groups and infrastructure components). We set the weight of a household to be equal to the number of members in it. The paper [101] did not provide the sizes corresponding to the households but gave a distribution of sizes. Thus, we consider a uniform distribution where 51 percent of households have size in the interval [1,4], 46 percent of the households have size in the interval [5,8] and 3 percent of the households have size in the interval [9,12]. Note that there was no information on how to estimate the constraints on number the seed groups and the number of infrastructure components that could be repaired. Thus, we consider 10000 instances where the weights of the social components are randomly generated from the aforementioned distribution, and K_s and K_p are uniformly and independently generated from the interval $[1, N]$. We found that the average reward for all the instances is 119. Then, we divided all the instances for the cases when $K_s \geq K_p$ and $K_s < K_p$ (there are about 5000 instances for both the cases). We found that the average reward for the case when $K_s \geq K_p$ is equal to 92 and for the case when $K_p > K_s$ is equal to 147. An intuitive reason for the average reward being larger when $K_p > K_s$ as compared to when $K_p \leq K_s$ is that physical components corresponding to all the activated social components need to be open but a few seed components can influence many social components.

We consider another social network consisting of 100 households from a village in Tamil Nadu, India [123]. In the crisis period, this social network forms a general directed graph (more details of this graph can be found in [123]). Unlike the previous example, there was no information available on the distribution of the sizes of the households. Therefore, we used the mean household size of Tamil Nadu to assign the weights for the households. Note that the mean household size of Tamil Nadu is 3.5 and thus we consider a uniform distribution in the interval [1,6] for generating the weights. As before, we consider 10000 instances where the weights of the social components are randomly generated from the aforementioned distribution, and K_s and K_p are uniformly and independently generated from the interval $[1, N]$. We found that the average reward for all the instances is 194. Also, we found that the average reward for the case when $K_s \geq K_p$ is equal to 168, which is less than the average reward of 220 when $K_p > K_s$ as seen in the example before. Note that we do not solve MILP to find the optimal solution as that can be computationally burdensome due to large size of the problem (as there are 100 components).

6.4 Conclusions

In this chapter, we first presented computational frameworks for various types of recovery problems. We find that these computational frameworks become computationally burdensome as the problem size increases. Thus, we provide heuristic policies (motivated by the form of our optimal policies for the cases that we consider in the previous chapters) for the cases where optimal or near-optimal policies are not currently known. Towards the end, we consider various case studies that are motivated from the real-world data and analyze the effect of using different policies. We also provide guidelines and challenges that are associated with the implementation of these results to various scenarios.

7. CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Introduction

In this chapter, we will first summarize the main conclusions from the previous chapters. After this, we will present the main contributions of this work. Finally, we will discuss the open problems that remain and provide directions to solve such problems.

7.2 Main conclusions

In the first chapter, we introduced the recovery problems that we focus in this dissertation and provided an overview of the existing practices and studies in the broad field of disaster recovery. We found that there are no existing studies and reports that provide guidelines for recovering deteriorating infrastructure components that fail irreversibly if they are not repaired in time. In addition, there are no existing works that focus on characterizing recovery policies for systems that contain interdependent socio-physical components. It was argued that there could be constraints on the availability of resources and manpower for recovery after disasters. The recovery decisions in such resource-constrained scenarios are often guided by practitioner knowledge, experience, and convention. Therefore, it is important to develop policies that can be efficiently implemented for optimal recovery of such systems. It was argued that the number of possible recovery strategies for such problems grows exponentially fast as the number of damaged components increases and thus it is intractable to enumerate all possible solutions to find the optimal strategy. Therefore, we need to find efficient ways of solving such problems. Thus, we aimed to address these issues in the other chapters.

Chapters 2-4 focused on characterizing optimal and near-optimal policies for the recovery of infrastructure components. In Chapter 2, we considered the case when there is a single repair agency and it seeks to repair deteriorating components. We showed that when the deterioration rate is larger than or equal to the repair rate and these rates are homogeneous across all the components, it is optimal to target the healthiest component at each time-step under certain conditions. When the repair rates are sufficiently larger than the deterioration rates, we showed that is optimal to target the component with the least value of health minus the deterioration rate at each time-step in a particular set at each time-step. Thus, we see that these policies depend on the relationship between the deterioration and repair rates. We also showed that many of these results can also be extended for the case when the deterioration and repair rates vary with time as long as the relative magnitudes of the deterioration and repair rates satisfy a particular set of conditions at each time-step.

In Chapter 3, we focused on the case when there are dependencies in the form of precedence constraints between infrastructure components and there is a constraint on the time that is available for repair. We provided examples to show that the policies that are optimal for the case when there are no dependencies need not be optimal when we have precedence and time constraints. Also, we proved that the general problem with precedence and time constraints is NP-hard. Thus, we provided (near-)optimal policies for the case when there are precedence constraints. Specifically, we proved that the policy that targets the healthiest component at each time-step while respecting the precedence constraints permanently repairs at least half the components as the optimal policy when the precedence constraints are given by a directed acyclic graph. Also, we showed that the policy of targeting the component with the least value of health minus the deterioration rate at each time-step while respecting the precedence constraints is also a near-optimal policy when the precedence constraints are given by directed trees and there is no time constraint. Therefore, the policies that are characterized to be optimal without considering the precedence and time constraints (as in Chapter 2) could be used as near-optimal policies for the case when there are such constraints (given that these policies respect these constraints).

Chapter 4 focused on the case when there are multiple agencies for repairing infrastructure components, there are allocation constraints between agencies and components, each agency charges a cost for repairing components and there is a central authority whose task is to allocate the components to various agencies under a given budget. We characterized optimal and near-optimal policies for different cases of the multiple agency recovery problem. For the case when the deterioration and repair rates are homogeneous across components, costs charged by agencies are equal and deterioration rate is larger than or equal to the repair rate, we provided a policy that permanently repairs at least half the components as the optimal policy. For the case when the repair rates are sufficiently larger than the deterioration rates and each agency can target any component, we first provided a policy for allocation of components where the largest sets of components that can be repaired by a single agency is first allocated to lowest cost agency and this process is repeated for the remaining set of components and agencies where the sets of components are allocated to the agencies in the increasing order of their costs. The aforementioned allocation along with the sequencing policy where the component with the least value of health minus deterioration rate is targeted at each time-step is optimal. Finally, for the case when the repair rates are sufficiently larger than the deterioration rates and the budget is sufficiently large, we provided a near-optimal policy for allocation and sequencing of components. An important point is that it is possible to separate the allocation and sequencing policies in this case because we know the optimal repair sequences for a single repair agency case from Chapter 2 and the policies that we provided in Chapter 4 ensure that all the components that are allocated to the repair agencies are permanently repaired.

Chapter 5 focuses on characterizing policies for the combined recovery of both the social and physical components. Specifically, we focus on the problem where the objective is to maximize the return of displaced social groups such that a displaced social group returns if at least one of its neighboring social group returns back and the physical component corresponding to the displaced social group is also repaired. We first considered a simplified version of this problem where the physical components can exist in one of the two states at any time (i.e., damaged or repaired) and the maximum number of physical components that can be repaired is given. We first prove that the general problem is NP-hard and therefore we characterize optimal and near-optimal algorithms (or policies) for various cases of the problem. Then, we provide a near-optimal algorithm for the case when the social network is a general directed graph. After that, we provide a near-optimal algorithm for the case when the social network is a bipartite graph such that this algorithm offers a better approximation guarantee as compared to the algorithm corresponding to the general directed graph. Then, we focus on the problem where the social network is a disjoint union of out-trees and we characterize an optimal algorithm for that case. In the end, we focus on the problem where the initial health values of the physical components lie in the interval [0,1] and the repair of physical components is governed by the sequencing decisions as in the previous chapters.

For this problem, we characterized optimal and near-optimal algorithms for special cases of the problem.

In Chapter 6, we provided computational frameworks and guidelines for the application of the results that are presented in the previous chapters. We first provided computational methods like Mixed Integer Linear Program (MILP) and heuristics for two broad class of problems that are considered in this dissertation. The first class of problems is the one that maximizes the total weight of the physical components that are permanently repaired considering all the aspects that have been discussed in the Chapters 2-4 such as heterogeneity in the weights and rates across the components, dependencies between the components, and multiple repair agencies. We find that the existing computational frameworks like MILP and Dynamic Programming become computationally burdensome as the size of the problem increases. Thus, we also provided heuristics that are motivated from the policies characterized in the previous chapters for the cases where the optimal policies are not currently known. The second class of problems is the one whose objective is to maximize the total weight of the social groups that return while considering the repair of infrastructure components. We again provided MILP and heuristics for this class of problems. After that, we provided some guidelines and examples that are motivated from the real-world data to help in the implementation of the policies and heuristics that are provided in this dissertation. For instance, we gave examples on how to initialize parameters like weights of physical and social components, initial health values of physical components, social networks, etc., in the guidelines and the case studies that we provided.

We now summarize the main contributions of this dissertation in the next section.

7.3 Summary of main contributions

The key contributions of this dissertation are as follows.

- We provide optimal policies for maximizing the repair of deteriorating components that fail irreversibly if they are not targeted in time. These policies depend on the relationship between the deterioration and repair rates of these components.

- We characterize near-optimal policies for maximizing the recovery of deteriorating components under precedence and time constraints depending on the relationship between the deterioration and repair rates.

- We provide policies for optimal allocation and repair of deteriorating components when there are multiple repair agencies under budget and allocation constraints.

- We characterize policies for recovery of socio-physical systems when there is a one-to-one mapping between the social and physical components and a social component becomes active (i.e., the corresponding social group returns back) if it has at least one active neighboring social component and its corresponding physical component is repaired.

- We provide computational methods in terms of mixed integer linear programming formulations and heuristics for the cases of the problems where the optimal or near-optimal policies are not currently known. We also provide guidelines using examples that are motivated from the real-world data for implementation of the policies that are characterized in this dissertation.

- The contributions of this dissertation to the real-world scenarios is that we prove that certain policies like non-jumping sequences that are actually used in the real-world are in fact optimal under certain conditions. Similarly, the robustness results towards the end of Chapter 2 allow using such policies for the cases where these rates are not exactly known but there is some sense of the relationships that are satisfied (e.g., the deterioration rate is faster than the repair rate).

In this dissertation, we have tackled many cases that arise in the recovery after disasters but there are still many extensions that are possible and thus we will focus on such extensions in the next section.

7.4 Directions for future work

There are several future extensions that are possible from this dissertation. The list of these extensions along with the guidance on how to tackle these issues are provided as follows.

- Although we provided heuristics for the case when the rates for physical components satisfy $\Delta_{dec} < \Delta_{inc} < (N-1)\Delta_{dec}$, the characterization of the optimal policy for this case remains open. We saw that sometimes the policy that targets the healthiest component at each time-step performs better than the policy that targets the least healthy component at each time-step and sometimes not (and there are examples to show that neither of these policies are optimal for this region). Thus, a policy that is a combination of the above two policies might be optimal. It may also be possible that this case of the problem is NP-hard and thus it may not be possible characterize the optimal policy for this case. Therefore, arguing that this case of the problem is NP-hard would also be an interesting direction.
- In Chapter 2, we considered time-varying deterioration and repair rates towards the end of the chapter but there are still complex cases for which the optimal policies are not known. For example, the case where the deterioration rates are larger than or equal to the repair rates in a particular time-step and in the next time-step the repair rates are larger than the deterioration rates, remains open. For these cases, a policy that is a combination of policies such as targeting the healthiest component at each time-step or targeting the least healthy component at each time-step might be a good candidate for an optimal policy.
- Real world settings involve a transit time when switching between physical components and thus considering that would be another useful extension in the sequencing decisions. We think that the non-jumping policies such as targeting the healthiest component at each time-step might still be optimal if there are additional costs in terms of transit times in traveling between different com-

ponents but policies that could have jumps such as targeting the least healthy component at each time-step need not be optimal under those scenarios. Therefore, characterizing optimal policies for such cases remains open.

- In the real-world, there may be uncertainty in the estimation of parameters such as health values, deterioration rates, repair rates, etc. Thus, characterizing policies for such cases remains open. Note that if there is stochasticity in deterioration and repair rates such that repair rates are sufficiently larger than the deterioration rates for all time-steps then it is possible to use the results on time-varying rates in Chapter 2 to argue that the policy of targeting the least healthy component at each time-step would be optimal. However, more complex variations of stochasticity in such parameters would be an avenue for future work.

- We considered time constraint for the repair of deteriorating components in Chapter 3 but we did not provide sequencing policies for the case when the repair rates are sufficiently larger than the deterioration rates. Thus, characterizing policies for that case remains open for future. Note that we provided an example to show that the policy that targets the component with the least modified health value at each time-step while respecting the time constraint is not optimal. Thus, a policy that is a modification of the aforementioned policy and considers time-constraint might be a good candidate as an optimal or near-optimal policy. Also, time could be a metric in the objective function rather than as a constraint. That is the objective of the problem could be to minimize the total repair time so that at least a given number of components are permanently repaired. We believe that sequencing policies for the case when the deterioration rates are larger than the repair rates may still hold for the aforementioned problem but further analysis might be required for characterizing policies for the remaining cases.

- In Chapter 4, we assumed that each component is allocated to at most one repair agency. However, it may be possible that one component may be targeted by more than one repair agency at a time (with the objective being to maximize the total number of components that are permanently repaired by all agencies). Thus, allowing multiple repair agencies to target a component at a time is an open problem. Note that if multiple agencies are allowed to repair a component at a time, then the way the costs are charged for repair would have to be different from the existing method as it would not make sense for the authority to pay an amount to the agencies that is larger than the amount it would have paid if only one of the repair agencies would have repaired that component.

- Another extension for the case of multiple repair agencies would be when all the repair agencies are selfish and maximize their own utilities rather than following the allocations that are provided by a central authority. Solving such scenarios might involve using game-theoretic tools so that proper equilibrium strategies can be identified. In the multiple repair agencies case, there could also be heterogeneity in the information that is available to various repair agencies regarding the health values of various components. For example, an agency might have full information on the initial health values of a set of components that lie in its geographic proximity but may have partial information on the health values of the remaining components. In those situations, exploring distributed strategies for information collection and sharing could be very efficient. Thus, characterizing policies under various information mechanisms would also be an interesting extension.

- The objective of the authority in Chapter 4 was to allocate components so that maximum number of components are permanently repaired. However, there could be other objectives such as minimizing total costs for repair or minimizing a weighted sum of the costs and time taken to repair (as in the A+Bx rule of [109] that is discussed in Chapter 4) while ensuring that at least a given number of components are permanently repaired. Note that if the objective is only to minimize the total costs for repair then it might be a very hard problem because under the concave cost structure as given in Assumption 2, the problem might become analogous to minimizing a concave function which in general is NP-hard [124]. However, it might still be possible to find optimal or near-optimal policies for special instances of that problem.

- In Chapters 3 and 4, we considered the extensions of Chapter 2 to take into account dependencies and multiple agencies cases separately. However, considering dependencies between the components when there are multiple repair agencies remains open. We believe that it should be feasible to characterize policies for this case by thinking along the same lines as in Chapters 3 and 4.

- In Chapter 5, we characterized algorithms for cases of Problem 8 under the Assumptions 7 and 8, and therefore characterizing approximation algorithms under more general conditions would be of interest. Recall that the influence maximization problem under *DLTM* is NP-hard to approximate within any constant factor when each inactive social component requires one or two active incoming neighboring components to get activated. Thus, it would be suggested to focus on characterizing optimal or near-optimal algorithms for special cases of the problem when Assumption 8 is removed but Assumption 7 is kept.

- In Chapter 5, we characterized algorithms for a limited number of cases when the recovery of physical components is a function of the sequencing decisions that are followed and the input parameters of the problem (i.e., when there are dynamics for the recovery of physical components). Thus, characterizing optimal or near-optimal algorithms for the remaining cases of this problem remains open.

- In Chapter 5, we considered the social ties between the social components to be static with time. However, these ties could change with time. For instance, with the passage of time certain social groups may loose the interest to return back and thus may require larger social influence to return. That is, an inactive social component could require more than one active incoming neighbors to become active with the passage of time. Thus, modeling such dynamic social connections would be an interesting extension. In addition, there may be additional conditions that are required in the social influence apart from requiring at least one neighboring component to return. For instance, certain social groups may only decide to return when one of their social groups with whom they share a very old connection return. Similarly, social ties representing family connections might have different importance as compared to other types of ties. One way to model such heterogeneous ties could be to consider edges with heterogeneous weights between the social components. This problem could be modeled as the influence maximization problem under the deterministic linear threshold model when there weighted edges and physical components. Since this problem is a generalization of the problem that is considered in this dissertation, it will pose new challenges. Thus, heuristic policies that are motivated from the algorithms that are characterized in this dissertation could also be helpful for efficiently obtaining approximate solutions.

- Considering dependencies between the physical components in Chapter 5 would also be an important extension (such as considering precedence constraints between physical components as in Chapter 3). Also, we focused on the recovery of social components while considering the recovery of physical components in Chapter 5 but there could also be a feedback effect of the social recovery on the recovery of physical components because the return of displaced communities exerts pressure on various agencies to speed up the repair of various infrastructure systems [125]. Note that for the problems considered in Chapter 5, the recovery of physical components is dependent on the recovery of social components since the repair of physical components considers the objective of maximizing the social recovery. However, the repair of physical components is a one shot event in Chapter 5 since it happens at time-step 0 and does not change later due to the recovery of social components. Thus, incorporating the repair of physical components as a feedback effect of the recovery of social components would be an interesting extension.

- In certain recovery scenarios, it may be adequate to obtain partially repaired components. For instance, some repair agencies may consider physical components to be reasonably repaired when their health value reaches 90% of the pre-disaster levels. Similarly, some displaced social groups might decide to return when physical components have been partially repaired in their neighborhood. Such settings involving heterogeneity in the repair thresholds across physical components would be interesting extensions for the problems that were considered in this dissertation.

- Characterizing policies or algorithms with improved near-optimality ratios, specially where the existing ratios are a function of the bounds of the weights of components (as in Theorem 5.4.1) or the number of components in a tree (as in Theorem 3.4.1), would provide significant contributions to the policies that are characterized in this dissertation.

- In this dissertation, we assumed that the initial states of the physical and social components are given. However, estimating these values is not easy and thus using proper techniques to do that is important for the recovery of such systems. In the event that there is no direct way to estimate the states of various components, indirect methods like social media tools could be used. For example, one could try to use the tweets from Twitter to estimate the damage values of an infrastructure. Note that developing methods for estimation of states of various components is an active area of research and thus improving the existing methods for state estimation is an avenue for future research.

- In Chapter 6, we provided some examples that are motivated from real-world data to test various policies and computational methods. However, we did not compare these results with the recovery decisions that were followed in the real-world. For instance, if someone wishes to compare the sequencing decisions that are followed in the real-world for the recovery physical components with the policies that are characterized in this dissertation, then one would require complete data of the sequencing decisions that are followed in a particular setting, i.e., it should include the sequencing decisions that are followed by each repair team that is involved in the recovery process. Also, there should be complete information related to the parameters like initial health values, deterioration rates, and repair rates. Note that if there is no significant deterioration of components in a particular scenario then that data is not relevant for this analysis. Similarly, if there are no constraints on the repair agencies that are available for recovery in a particular scenario, then that data is also not relevant for this analysis because the policies that are provided in this dissertation are only meaningful for the cases where there are constraints on manpower or resources. Usually, the complete information on sequencing decisions that satisfies the above conditions is not readily available in the media reports, research papers and other works that can be accessed online. Therefore, we did not put this analysis in this dissertation. However, this kind of data might be possible to collect by contacting recovery practitioners. Similarly, in order to make such comparisons for the socio-physical systems, we would need complete data on social networks for the groups that were displaced after disasters, damage information of the infrastructure systems on which these groups depend upon, and the time at which different groups decided to return back and various infrastructure components got repaired.

REFERENCES

[1] E. S. Blake, T. B. Kimberlain, R. J. Berg, J. P. Cangialosi, and J. L. Beven Ii, "Tropical cyclone report: Hurricane Sandy," *National Hurricane Center*, vol. 12, pp. 1–10, 2013.

[2] R. H. Nolan, M. M. Boer, L. Collins, V. Resco de Dios, H. Clarke, M. Jenkins, B. Kenny, and R. A. Bradstock, "Causes and consequences of eastern australia's 2019-20 season of mega-fires," *Global change biology*, 2020.

[3] M. Polyakova, G. Kocks, V. Udalova, and A. Finkelstein, "Initial economic damage from the covid-19 pandemic in the united states is more widespread across ages and geographies than initial mortality impacts," *Proceedings of the National Academy of Sciences*, vol. 117, no. 45, pp. 27 934–27 939, 2020.

[4] D. J. Leversage and E. J. Byres, "Estimating a system's mean time-to-compromise," *IEEE Security & Privacy*, vol. 6, no. 1, pp. 52–60, 2008.

[5] H. Gehlot and S. V. Ukkusuri, "User equilibrium with a policy-based link transmission model for stochastic time-dependent traffic networks," *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 1013–1043, 2019.

[6] E. Fussell, "The long-term recovery of new orleans' population after hurricane katrina," *American Behavioral Scientist*, vol. 59, no. 10, pp. 1231–1245, 2015.

[7] W. Li, C. A. Airriess, A. C.-C. Chen, K. J. Leong, and V. Keith, "Katrina and migration: Evacuation and return by african americans and vietnamese americans in an eastern new orleans suburb," *The Professional Geographer*, vol. 62, no. 1, pp. 103–118, 2010.

[8] S. Chang, "Socioeconomic impacts of infrastructure disruptions," *Oxford Research Encyclopedia of Natural Hazard Science*, 2016.

[9] K. Gaspard, M. Martinez, Z. Zhang, and Z. Wu, "Impact of Hurricane Katrina on roadways in the New Orleans area," *Technical Assistance Rep. No. 07-2TA*, 2007.

[10] M. F. C. Association *et al.*, *Fire service deployment: Assessing community vulnerability.* NFPA, 2009.

[11] *U.s. disaster-response force stretched thin as hurricane season starts | reuters*, https://www.reuters.com/article/us-usa-disaster-fema-insight-idINKBN1J91C8, (Accessed on 04/08/2021).

[12] *Why is the u.s. facing a federal firefighter shortage? - pacific standard*, https://psmag.com/news/why-is-the-us-facing-a-federal-firefighter-shortage, (Accessed on 04/08/2021).

[13] *The uk has a shortage of 5,000 firefighters - here's how to apply | burnley express*, https://www.burnleyexpress.net/read-this/uk-has-shortage-5000-firefighters-heres-how-apply-3022477, (Accessed on 04/08/2021).

[14] *Delhi fire department reeling under lack of staff, equipment - mail today news*, https://www.indiatoday.in/mail-today/story/delhi-fire-department-reeling-under-lack-of-staff-equipment-1455640-2019-02-14, (Accessed on 04/08/2021).

[15] R. Green, L. K. Bates, and A. Smyth, "Impediments to recovery in new orleans' upper and lower ninth ward: One year after hurricane katrina," *Disasters*, vol. 31, no. 4, pp. 311–335, 2007.

[16] FEMA, "Damage Assessment Operation Manual," 2016. [Online]. Available: https://www.fema.gov/media-library/assets/documents/109040.

[17] L. K. Siebeneck and T. J. Cova, "Spatial and temporal variation in evacuee risk perception throughout the evacuation and return-entry process," *Risk Analysis*, vol. 32, no. 9, pp. 1468–1480, 2012.

[18] C. M. Charoenkalunyuta, A. Tuladhar, and J. Zevenbergen, "Community resilience in disaster prone areas based on land rights/ownerships," in *32nd Asian conference on remote sensing 2011*, 2011, pp. 556–571.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[20] A. Liu, M. Fellowes, and M. Mabanta, *Special edition of the Katrina index: A one year review of key indicators of recovery in post-storm New Orleans*. Brookings Institution Washington, DC, 2006.

[21] FEMA, "National Disaster Recovery Framework," 2011. [Online]. Available: https://www.fema.gov/pdf/recoveryframework/ndrf.pdf.

[22] A. Opdyke, F. Lepropre, A. Javernick-Will, and M. Koschmann, "Inter-organizational resource coordination in post-disaster infrastructure recovery," *Construction Management and Economics*, vol. 35, no. 8-9, pp. 514–530, 2017.

[23] M. Jahre and L.-M. Jensen, "Coordination in humanitarian logistics through clusters," *International Journal of Physical Distribution & Logistics Management*, vol. 40, no. 8/9, pp. 657–674, 2010.

[24] I. M. Nolte, E. C. Martin, and S. Boenigk, "Cross-sectoral coordination of disaster relief," *Public Management Review*, vol. 14, no. 6, pp. 707–730, 2012.

[25] G. A. Okhuysen and B. A. Bechky, "10 coordination in organizations: An integrative perspective," *Academy of Management annals*, vol. 3, no. 1, pp. 463–502, 2009.

[26] IoM, *Healthy, Resilient, and Sustainable Communities After Disasters: Strategies, Opportunities, and Planning for Recovery*. National Academies Press, 2015.

[27] *Emergency relief manual may 31, 2013*, https://www.fhwa.dot.gov/reports/erm/er.pdf, (Accessed on 03/09/2021).

[28] M. W. Downton and R. A. Pielke, "How accurate are disaster loss data? the case of us flood damage," *Natural Hazards*, vol. 35, no. 2, pp. 211–228, 2005.

[29] R. Zimmerman, "Planning restoration of vital infrastructure services following hurricane sandy: Lessons learned for energy and transportation," *Journal of Extreme Events*, vol. 1, no. 01, p. 1 450 004, 2014.

[30] HomeAdvisor, "How much does it cost to repair asphalt paving," 2019. [Online]. Available: https://www.homeadvisor.com/cost/outdoor-living/repair-asphalt-paving/.

[31] R. Lahood, "Hurricane sandy's effects on atlantic seaboard roads, pavement widespread," 2012. [Online]. Available: www.pothole.info.

[32] H. S. R. T. Force, "Hurricane sandy rebuilding strategy," *US Department of Housing and Urban Development, Washington DC*, 2013.

[33] M. G. Karlaftis, K. L. Kepaptsoglou, and S. Lambropoulos, "Fund allocation for transportation network recovery following natural disasters," *Journal of Urban Planning and Development*, vol. 133, no. 1, pp. 82–89, 2007.

[34] CRFPD, "Standards of Response Coverage," 2017.

[35] M. McGowan and H. Davidson, *Volunteer firefighters in australia warned not to crowdfund for equipment*, Dec. 2019. [Online]. Available: www.theguardian.com.

[36] M. P. Thompson and D. E. Calkin, "Uncertainty and risk in wildland fire management: A review," *Journal of environmental management*, vol. 92, no. 8, pp. 1895–1909, 2011.

[37] D. L. Martell, "A review of operational research studies in forest fire management," *Canadian Journal of Forest Research*, vol. 12, no. 2, pp. 119–140, 1982.

[38] S. Mannakkara and S. Wilkinson, “Build back better principles for economic recovery: Case study of the victorian bushfires,” *Journal of business continuity & emergency planning*, vol. 6, no. 2, pp. 164–173, 2013.

[39] L. C. Smeby Jr, *Fire and emergency services administration: Management and leadership practices*. Jones & Bartlett Publishers, 2013.

[40] S. G. Nurre, B. Cavdaroglu, J. E. Mitchell, T. C. Sharkey, and W. A. Wallace, “Restoring infrastructure systems: An integrated network design and scheduling (inds) problem,” *European Journal of Operational Research*, vol. 223, no. 3, pp. 794–806, 2012.

[41] Y. Almoghathawi, K. Barker, and L. A. Albert, “Resilience-driven restoration model for interdependent infrastructure networks,” *Reliability Engineering & System Safety*, vol. 185, pp. 12–23, 2019.

[42] M. Çelik, Ö. Ergun, and P. Keskinocak, “The post-disaster debris clearance problem under incomplete information,” *Operations Research*, vol. 63, no. 1, pp. 65–85, 2015.

[43] S.-S. Liu, M. F. A. Arifin, W. T. Chen, and Y.-H. Huang, “Emergency repair scheduling model for road network integrating rescheduling feature,” *Applied Sciences*, vol. 11, no. 4, p. 1447, 2021.

[44] S.-h. Joo, Y. Ogawa, and Y. Sekimoto, “Decision-making system for road-recovery considering human mobility by applying deep q-network,” in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 4075–4084.

[45] W. Zhang, J. Hu, and A. Abate, “On the value functions of the discrete-time switched LQR problem,” *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2669–2674, 2009.

[46] D. Gorges, M. Izak, and S. Liu, “Optimal control and scheduling of switched systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 135–140, 2010.

[47] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, “Dynamic programming for constrained optimal control of discrete-time linear hybrid systems,” *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

[48] A. Bemporad and N. Giorgetti, “Logic-based solution methods for optimal control of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 963–976, 2006.

[49] F. Zhu and P. J. Antsaklis, “Optimal control of hybrid switched systems: A brief survey,” *Discrete Event Dynamic Systems*, vol. 25, no. 3, pp. 345–364, 2015.

[50] T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas, "Green scheduling of control systems for peak demand reduction," in *50th IEEE Conference on Decision and Control, and European Control Conference (CDC-ECC)*, 2011, pp. 5131–5136.

[51] P. Nilsson and N. Ozay, "On a class of maximal invariance inducing control strategies for large collections of switched systems," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, ACM, 2017, pp. 187–196.

[52] P. O. Scokaert and J. B. Rawlings, "Feasibility issues in linear model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1649–1659, 1999.

[53] P. Mhaskar, N. H. El-Farra, and P. D. Christofides, "Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control," *Systems & Control Letters*, vol. 55, no. 8, pp. 650–659, 2006.

[54] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 6, pp. 1333–1344, 2007.

[55] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, 2005.

[56] R. Buche and H. J. Kushner, "Control of mobile communication systems with time-varying channels via stability methods," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1954–1962, 2004.

[57] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

[58] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2012.

[59] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1659–1664, 2014.

[60] T. E. Cheng, Q. Ding, and B. M. Lin, "A concise survey of scheduling with time-dependent processing times," *European Journal of Operational Research*, vol. 152, no. 1, pp. 1–13, 2004.

[61] C.-M. Wei, J.-B. Wang, and P. Ji, “Single-machine scheduling with time-and-resource-dependent processing times,” *Applied Mathematical Modelling*, vol. 36, no. 2, pp. 792–798, 2012.

[62] C. Ng, S. Li, T. E. Cheng, and J. Yuan, “Preemptive scheduling with simple linear deterioration on a single machine,” *Theoretical Computer Science*, vol. 411, no. 40-42, pp. 3578–3586, 2010.

[63] F. Zhang and A. Burns, “Schedulability analysis for real-time systems with EDF scheduling,” *IEEE Transactions on Computers*, no. 9, pp. 1250–1258, 2009.

[64] N. T. Argon, S. Ziya, and R. Righter, “Scheduling impatient jobs in a clearing system with insights on patient triage in mass casualty incidents,” *Probability in the Engineering and Informational Sciences*, vol. 22, no. 3, pp. 301–332, 2008.

[65] L. Haque and M. J. Armstrong, “A survey of the machine interference problem,” *European Journal of Operational Research*, vol. 179, no. 2, pp. 469–482, 2007.

[66] S. M. Iravani and B. Kolfal, “When does the cmu rule apply to finite-population queueing systems?” *Operations Research Letters*, vol. 33, no. 3, pp. 301–304, 2005.

[67] R. Righter, “Optimal policies for scheduling repairs and allocating heterogeneous servers,” *Journal of applied probability*, pp. 536–547, 1996.

[68] S. Aalto, U. Ayesta, and R. Righter, “On the gittins index in the m/g/1 queue,” *Queueing Systems*, vol. 63, no. 1-4, p. 437, 2009.

[69] J. C. Gittins, “Bandit processes and dynamic allocation indices,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 41, no. 2, pp. 148–164, 1979.

[70] W. Orabi, A. B. Senouci, K. El-Rayes, and H. Al-Derham, “Optimizing resource utilization during the recovery of civil infrastructure systems,” *Journal of management in engineering*, vol. 26, no. 4, pp. 237–246, 2010.

[71] J. Sun and Z. Zhang, “A post-disaster resource allocation framework for improving resilience of interdependent infrastructure networks,” *Transportation Research Part D: Transport and Environment*, vol. 85, p. 102 455, 2020.

[72] W. N. Adger, T. P. Hughes, C. Folke, S. R. Carpenter, and J. Rockström, “Social-ecological resilience to coastal disasters,” *Science*, vol. 309, no. 5737, pp. 1036–1039, 2005.

[73] D. S. Massey, J. Arango, G. Hugo, A. Kouaouci, A. Pellegrino, and J. E. Taylor, “Theories of international migration: A review and appraisal,” *Population and development review*, pp. 431–466, 1993.

[74] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2003, pp. 137–146.

[75] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in *2010 IEEE international conference on data mining*, IEEE, 2010, pp. 88–97.

[76] Z. Lu, W. Zhang, W. Wu, B. Fu, and D. Du, “Approximation and inapproximation for the influence maximization problem in social networks under deterministic linear threshold model,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, IEEE, 2011, pp. 160–165.

[77] Z. Lu, W. Zhang, W. Wu, J. Kim, and B. Fu, “The complexity of influence maximization problem in the deterministic linear threshold model,” *Journal of combinatorial optimization*, vol. 24, no. 3, pp. 374–378, 2012.

[78] H. Gehlot, S. Sundaram, and S. V. Ukkusuri, “Optimal sequencing policies for recovery of physical infrastructure after disasters,” in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 3605–3610.

[79] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge university press, 2011.

[80] M. Janssen, J. Lee, N. Bharosa, and A. Cresswell, “Advances in multi-agency disaster management: Key elements in disaster research,” *Information Systems Frontiers*, vol. 12, no. 1, pp. 1–7, 2010.

[81] M. C. Comerio, “Estimating downtime in loss modeling,” *Earthquake Spectra*, vol. 22, no. 2, pp. 349–365, 2006.

[82] B. Lehmann, D. Lehmann, and N. Nisan, “Combinatorial auctions with decreasing marginal utilities,” *Games and Economic Behavior*, vol. 55, no. 2, pp. 270–296, 2006.

[83] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of computer computations*, Springer, 1972, pp. 85–103.

[84] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, 2. Athena scientific Belmont, MA, 1995, vol. 1.

[85] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[86] P. Sandin, “Firefighting ethics: Principlism for burning issues,” *Ethical Perspectives*, vol. 16, no. 2, pp. 225–251, 2009.

[87] V. S. Gordon, C. N. Potts, V. A. Strusevich, and J. D. Whitehead, "Single machine scheduling models with deterioration and learning: Handling precedence constraints via priority generation," *Journal of Scheduling*, vol. 11, no. 5, p. 357, 2008.

[88] M. Garey and D. S. Johnson, "Scheduling tasks with nonuniform deadlines on two processors," *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 461–467, 1976.

[89] D. R. Godschalk, A. Rose, E. Mittler, K. Porter, and C. T. West, "Estimating the value of foresight: Aggregate analysis of natural hazard mitigation benefits and costs," *Journal of Environmental Planning and Management*, vol. 52, no. 6, pp. 739–756, 2009.

[90] G. J. Stigler, "The economies of scale," *The Journal of Law and Economics*, vol. 1, pp. 54–71, 1958.

[91] M. R. Garey and D. S. Johnson, *Computers and intractability*. freeman San Francisco, 1979, vol. 174.

[92] H. Narayanan, *Submodular functions and electrical networks*. Elsevier, 1997, vol. 54.

[93] O. Patterson, F. Weil, and K. Patel, "The role of community in disaster response: Conceptual models," *Population Research and Policy Review*, vol. 29, no. 2, pp. 127–141, 2010.

[94] *What it takes to rebuild a village after a disaster: Stories from internally displaced children and families of hurricane katrina and their lessons for our nation*, https://www.childrensdefense.org/wp-content/uploads/2018/08/rebuild-village-hurricane-katrina-rita-children.pdf, (Accessed on 11/16/2020).

[95] *Tax credits for those affected by natural disasters - turbotax tax tips & videos*, https://turbotax.intuit.com/tax-tips/tax-relief/tax-credits-for-those-affected-by-natural-disasters/L7Xcggyfp, (Accessed on 11/03/2020).

[96] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[97] F. Gilbert, P. Simonetto, F. Zaidi, F. Jourdan, and R. Bourqui, "Communities and hierarchical structures in dynamic social networks: Analysis and visualization," *Social Network Analysis and Mining*, vol. 1, no. 2, pp. 83–95, 2011.

[98] J. L. Martin, "Structures of power in naturally occurring communities," *Social networks*, vol. 20, no. 3, pp. 197–225, 1998.

[99] P. Wang, G. Robins, P. Pattison, and E. Lazega, “Exponential random graph models for multilevel networks,” *Social Networks*, vol. 35, no. 1, pp. 96–115, 2013.

[100] M. Shumate and E. T. Palazzolo, “Exponential random graph (p*) models as a method for social network analysis in communication research,” *Communication Methods and Measures*, vol. 4, no. 4, pp. 341–371, 2010.

[101] V. Deepa, A. Lakshmanan, and V. Sreeja, “Centrality and reciprocity in directed social networks-a case study,” *Malaya Journal of Matematik (MJM)*, no. 1, 2019, pp. 479–484, 2019.

[102] A. Wang, W. Wu, and L. Cui, “On bharathi–kempe–salek conjecture for influence maximization on arborescence,” *Journal of Combinatorial Optimization*, vol. 31, no. 4, pp. 1678–1684, 2016.

[103] R. Lima and E. Seminar, “IBM ILOG CPLEX-what is inside of the box?” In *Proc. 2010 EWO Seminar*, 2010, pp. 1–72.

[104] M. L. Puterman, “Markov decision processes,” *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[105] M. Sultana, G. Chai, S. Chowdhury, T. Martin, Y. Anissimov, and A. Rahman, “Rutting and roughness of flood-affected pavements: Literature review and deterioration models,” *Journal of Infrastructure Systems*, vol. 24, no. 2, p. 04 018 006, 2018.

[106] J. Padgett, R. DesRoches, B. Nielson, M. Yashinsky, O.-S. Kwon, N. Burdette, and E. Tavera, “Bridge damage and repair costs from Hurricane Katrina,” *Journal of Bridge Engineering*, vol. 13, no. 1, pp. 6–14, 2008.

[107] M. Stearns and J. E. Padgett, “Impact of 2008 Hurricane Ike on bridge infrastructure in the Houston/Galveston region,” *Journal of Performance of Constructed Facilities*, vol. 26, no. 4, pp. 441–452, 2012.

[108] FEMA, *Damage assessment operations manual*, 2016.

[109] FHWA, *Emergency Relief Manual*. Congressional Research Service, 2013.

[110] M. Sultana, G. Chai, T. Martin, and S. Chowdhury, “A review of the structural performance of flooded pavements,” in *26th ARRB Conference–Research Driving Efficiency, Sydney, New South Wales*, 2014.

[111] *How corrosion affects power plant performance - polygon group*, https://www.polygongroup.com/en-US/blog/how-corrosion-affects-power-plant-performance/, (Accessed on 11/07/2020).

[112] *Water chemistry: Power plant life and death*, https://www.powermag.com/water-chemistry-power-plant-life-and-death/, (Accessed on 11/07/2020).

[113] Y. Lin, B. Chen, J. Wang, and Z. Bie, "A combined repair crew dispatch problem for resilient electric and natural gas system considering reconfiguration and dg islanding," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2755–2767, 2019.

[114] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE control systems magazine*, vol. 21, no. 6, pp. 11–25, 2001.

[115] *Standard*, https://www.dot.ny.gov/main/business-center/engineering/specifications/busi-e-standards-usc/usc-repository/2019_1_specs_usc.pdf, (Accessed on 10/07/2020).

[116] *All highway 37 lanes reopen after flood closure – marin independent journal*, https://www.marinij.com/2019/03/02/all-highway-37-lanes-reopen-after-flood-closure/, (Accessed on 10/08/2020).

[117] *Vdot aiming to reopen flood-damaged mclean roads this fall | tysons reporter*, https://www.tysonsreporter.com/2019/09/05/vdot-aiming-to-reopen-flood-damaged-mclean-roads-this-fall/, (Accessed on 10/08/2020).

[118] U. DOT, "Highway functional classification concepts, criteria and procedures," FHWA–PL–13–026, US Department of Transportation, Federal Highway ..., Tech. Rep., 2013.

[119] J. Ryan, G. Schexnayder, and E. Scheuermann, "Five miles in forty five days rebuilding the i 10 twin spans," in *2006 Annual Conference & Exposition*, 2006, pp. 11–645.

[120] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," *arXiv preprint arXiv:1805.00713*, 2018.

[121] *Tax relief for hurricane, wildfire, flood and other natural disaster victims | kiplinger*, https://www.kiplinger.com/taxes/601293/tax-relief-for-hurricane-wildfire-flood-and-other-natural-disaster-victims, (Accessed on 11/03/2020).

[122] *Individual disaster assistance | fema.gov*, https://www.fema.gov/individual-disaster-assistance, (Accessed on 11/03/2020).

[123] S. Arumugam, D. Haughton, B. Vasanthi, and C. Zhang, "Reciprocity in social networks-a case study in tamil nadu, india," 2014.

[124] H. P. Benson, "Concave minimization: Theory, applications and algorithms," in *Handbook of global optimization*, Springer, 1995, pp. 43–148.

[125] C. A. Airriess, W. Li, K. J. Leong, A. C.-C. Chen, and V. M. Keith, "Church-based social capital, networks and geographical scale: Katrina evacuation, relocation, and recovery in a new orleans vietnamese american community," *Geoforum*, vol. 39, no. 3, pp. 1333–1346, 2008.