OPERATIONS ANALYTICS AND OPTIMIZATION FOR UNSTRUCTURED SYSTEMS: CYBER COLLABORATIVE ALGORITHMS AND PROTOCOLS FOR AGRICULTURAL SYSTEMS

by

Puwadol Dusadeerungsikul

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Industrial Engineering

West Lafayette, Indiana

May 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Shimon Y. Nof, Chair

School of Industrial Engineering

Dr. Seokcheon Lee

School of Industrial Engineering

Dr. David J. Cappelleri

School of Mechanical Engineering

Dr. Hong Wan

Edward P. Fitts Department of Industrial and Systems Engineering, NC State University

Approved by:

Dr. Abhijit Deshmukh

To Mom and Dad

ACKNOWLEDGMENTS

Through this dissertation, I received advice and support from friends, colleagues, and mentors. All of them have made a lasting impact, and without them, I could not accomplish my goals.

First and foremost, I would like to express my deepest gratitude to my advisor, mentor, co-author, and friend, Professor Shimon Y. Nof, who taught me how to become a good researcher and how to be positive in any situation in life. His support, guidance, encouragement, and patience are valuable and push me to my best capabilities. I also would like to thank the members of my advisory committee, Professor Seokcheon Lee, Professor David J. Cappelleri, and Professor Hong Wan, for their valuable time and inputs on my research. I also would like to thank Professor Avital Bechar for all the supports and suggestions.

Special thanks to all my colleagues and friends at the PRISM Center, especially to Dr. Guo, Dr. Win, Ashwin, Billy, and Maitreya. It is my pleasure to be part of the team with talented researchers.

Last but not least, my gratitude goes to my family for their unconditional love and support to my mother, Jiamjai, my father, Prakart, and my brother, Dhiti, for being a source of power this hard time.

TABLE OF CONTENTS

LIST OF FIGURES 11
LIST OF TABLES 13
LIST OF ABBREVIATIONS 14
LIST OF SYMBOLS 16
ABSTRACT 20
CHAPTER 1. INTRODUCTION 22
1.1 Research Motivation 22
1.2 General Assumptions 24
1.2.1 Assumption 1: System scope assumption 24
1.2.2 Assumption 2: Modeling assumption 25
1.2.3 Assumption 3: Agents assumption 26
1.2.4 Assumption 4: Terminologies assumption 26
1.3 Research Problem and Research Questions 27
1.3.1 Research problem 27
1.3.2 Research questions 27
Research question 1 (RQ_1) 27
Research question 2 (RQ_2) 28
Research question 3 (RQ_3) 28
Research question 4 (RQ_4) 28
1.4 Dissertation Structure 28
CHAPTER 2. LITERATURE REVIEW 29
2.1 Agriculture Robotics System (ARS) 29
2.1.1 Harvesting Robot 29
2.1.2 Data collection and crops inspection robots 30
2.1.3 Environmental Control System 32
2.1.4 Spraying Robot 32
2.1.5 Yield Improvement Robot 34
2.2 Cyber-Augmented System in Agriculture 35
2.2.1 Cyber-Physical System (CPS) in Agriculture 35

2.2.2 Internet of Things and Internet of Services (IoT/IoS) in agriculture 36
2.3 Collaborative Control Theory (CCT) 37
2.3.1 Design principle of CCT 37
2.3.2 Task Administration Protocol (TAP) 39
2.3.3 Precision Collaboration (PCol) 39
2.4 Optimization in Agriculture 43
2.4.1 Traveling Salesman Problem for Robotic Monitoring and Inspection 43
2.4.2 Object Search by Robot.. 44
2.4.3 Optimal Cost Balancing in Agriculture.. 45
2.4.4 Multi-agent System Optimization 47
2.5 Summary of Research Gaps 48
CHAPTER 3. COLLABORATIVE CONTROL PROTOCOL FOR CYBER-PHYSICAL SYSTEM (CCP-CPS) 51
3.1 Collaborative Control Protocol for Cyber-Physical System (CCP-CPS) design 51
3.1.1 CPS framework for PA 51
Wireless sensor network (WSN) deployment 53
Cloud platform 54
Transmission mode 54
Agricultural robots and other actuators 55
3.1.2 CCP-CPS design 56
3.1.3 Collaborative control theory in CCP-CPS 59
Collaborative architecture for CCP-CPS.. 59
Collaboration requirement plan (CRP) 60
Error prevention and conflict resolution (EPCR).. 61
3.1.4 Monitoring alternatives 63
CPS Scheme 64
Non-CPS Scheme.. 65
3.2 Experiments and results.. 66
3.2.1 Experiment 3.1 – Total operation time.. 66
Results and analysis 66
3.2.2 Experiment 3.2 – Conflicts and error tolerance 66

Results and analysis 67
3.2.3 Experiment 3.3 - Emergency response 68
Results and analysis 68
3.2.4 Conclusion and discussion 69
CHAPTER 4. COLLABORATIVE CONTROL PROTOCOL FOR EARLY DETECTION OF STRESS IN PLANTS (CCP-ED) 70
4.1 Task description 70
4.2 CCP-ED 71
4.2.1 Collaboration Requirement Planning (CRP) 72
CRP-I: Plan Generation 72
CRP-II: Plan Execution & Revision 73
4.2.2 Error Prevention and Conflict Resolution (EPCR) 74
4.2.3 Elements in CCP-ED 75
4.2.4 CCP-ED design 76
Collaborative control protocol for early detection of stress in plants (CCP-ED) steps 76
A routing algorithm for agriculture robotics algorithm 78
Adaptive search algorithm 81
4.3 Experiments 83
4.3.1 Experimental setting 83
4.3.2 Performance metrics 83
Number of Infected Plants Found (NIPF) 84
Overall Robotic Effectiveness - ORE 84
4.3.3 Experimental Results 85
Analysis of ORE 87
4.3.4 Conclusion and Discussion 88
CHAPTER 5. OPTIMAL INSPECTION PROFIT FOR PRECISION AGRICULTURE 90
5.1 Type of system errors in AS 90
5.2 Monitoring Profit 92
5.2.1 Overage Cost 93
5.2.2 Underage Cost 93
5.3 Optimal Balancing 94

5.3.1 Dynamic Adaptive Search Algorithm ... 96
5.3.2 Propagation Probability Map Model ... 96
5.4 Experiments, Results, and Analysis ... 98
5.4.1 Experimental Setting ... 99
5.4.2 Search policies alternatives ... 99
Dynamic Adaptive Search ... 99
Static Adaptive Search ... 100
Always Search ... 101
None Search ... 101
5.4.3 Computational Experiment and Results ... 101
System Performance Analysis ... 101
True Positive Ratio ... 102
True Negative Ratio ... 102
Experimental Results ... 103
Cost Analysis ... 104
Monitoring Profit ... 104
Experimental Results ... 104
Sensitivity Analysis ... 106
Experimental Results ... 106
5.5 Conclusion and Discussions ... 108
CHAPTER 6. MULTI-AGENT SYSTEM OPTIMIZATION IN GREENHOUSE FOR TREATING PLANTS ... 109
6.1 Cyber Collaborative for Plant Treatment (C2T) ... 109
6.1.1 Cyber Collaborative Greenhouse Operation Types ... 110
6.1.2 Type of Plant Treatment Tasks ... 110
6.1.3 Task Operational Cost ... 111
6.1.4 Task Operational Time ... 112
6.2 Cyber Collaborative Greenhouse System Architecture ... 112
6.3 CRP-H Protocol design ... 113
6.3.1 CRP-I Optimization ... 114
6.3.2 A mathematical model for CRP-I ... 115

Mathematical model 115
6.3.3 CRP-II Harmonization 116
6.3.4 A mathematical model for CRP-II 116
Mathematical model 116
6.3.5 Collaborative Robots Scheduling (CRS) Algorithm 118
6.4 Human Role and Human-in-the-loop Design for C2T 120
6.5 Experiments 121
6.5.1 General Experimental Setting 121
Mode of operation 121
Time 121
Capability 121
Weight 121
Cost 122
6.5.2 Modeling Procedure 122
6.5.3 Experiment 6.1: Performance Analysis in Planned Operations 122
System Description 122
6.5.4 Experiment 6.2: Performance Analysis with Unplanned Operations 123
System Description 123
6.5.5 Experiment 6.3: Performance Analysis with HITL design 123
System Description 123
6.6 Results 124
6.6.1 Experiment 6.1 Results 124
Cost-saving 125
Time-saving 125
6.6.2 Experiment 6.2 Results 126
6.6.3 Experiment 6.3 Results 128
6.7 Conclusion and Discussion 130
CHAPTER 7. CONCLUSION AND DISCUSSION 132
7.1 Summary and Original Contributions 132
7.2 Limitation and Future Research Directions 139
REFERENCES 141

VITA 154
PUBLICATIONS 156

LIST OF FIGURES

Figure 1.1. System scope assumption 25

Figure 2.1. (Left) Fruit mass prediction; (Right) Number of leaves prediction (Finkelshtain, Bechar, Yovel, & Kósa, 2017) 31

Figure 2.2. Disease and stress monitoring by the robot (Schor et al., 2016; Schor et al., 2017).. 31

Figure 2.3. An autonomous sprayer (Dar, Edan, & Bechar, 2011) 33

Figure 2.4. Machine visualization for mapping in the greenhouse (Dar et al., 2011) 33

Figure 2.5. (Left) Computing optimal trajectory tree pruning; (Right) End-effector for pruning (Bechar, Nof, & Wachs, 2014) 34

Figure 2.6. Precision Collaboration Support Framework (Nof, 2015) 41

Figure 2.7. Object targeting and monitoring 41

Figure 2.8. Mapping, localizing, and human detection tasks 42

Figure 2.9. Collaboration support features and their precision requirements (Bechar et al., 2012) 42

Figure 3.1. CPS framework for monitoring plants conditions in a greenhouse (Guo et al., 2018) 52

Figure 3.2. CPS Framework for greenhouse monitoring system 52

Figure 3.3. Computational structure for CCP-CPS agricultural robot 56

Figure 3.4. CCP-CPS relationship diagram 58

Figure 3.5. CCP-CPS workflow chart 58

Figure 3.6. Collaborative architecture for CCP-CPS 59

Figure 3.7. CRP architecture for agricultural CCP-CPS 60

Figure 3.8. Robot conflict propagation and the resolution process (Guo et al., 2017) 62

Figure 3.9. CPS Scheme 64

Figure 3.10. Non-CPS Scheme 65

Figure 3.11. Experiment 3.2 results 67

Figure 4.1. Agricultural robotic system operation; North and West propagation directions assumed as given by experts for this crop season 71

Figure 4.2. Sub-tour elimination 79

Figure 4.3. Adaptive search when propagation directions are known from previous research 82

Figure 4.4. NIPF under each protocol design ... 86
Figure 4.5. ORE under each protocol design ... 86
Figure 4.6. ORE and its components ... 88
Figure 5.1. Type of errors in a monitoring system ... 91
Figure 5.2. Disease propagation ... 97
Figure 5.3. Dynamic Search Algorithm ... 100
Figure 6.1. Cyber Collaborative Greenhouse system architecture ... 113
Figure 6.2. CRP-H protocol ... 114
Figure 6.3. Schedule S' ... 119
Figure 6.4. Schedule S ... 119
Figure 6.5. Experiment 6.1 results ... 125
Figure 6.6. Experiment 6.1: Cost and time-saving at different number of tasks ... 126
Figure 6.7. Experiment 6.2 results ... 127
Figure 6.8. Experiment 6.3 results ... 129
Figure 6.9. Cost differences between CRP-H and baseline when percentage of UR increases . 129

LIST OF TABLES

Table 1.1. Main role and responsibility of agents 26
Table 2.1. Challenge in Precision Agriculture with CCT principles and methods 38
Table 2.2. Summary of the research gaps corresponding with the RQs and methodologies 49
Table 2.3. Recent research on agricultural robotics for crop monitoring (sample) 50
Table 3.1. Different sensors monitoring different parameters 53
Table 3.2. Communication technologies applied in CCP-CPS 55
Table 3.3. Experiment 3.1 results 66
Table 3.4. Experiment 3.3 results 68
Table 4.1. Major potential errors and conflicts examples in ARS planning and control 75
Table 4.2. Alternative protocols design 83
Table 4.3. ANOVA analysis 87
Table 5.1. Summary of parameter settings 99
Table 5.2. Performance analysis 103
Table 5.3. Cost analysis 105
Table 5.4. Sensitivity analysis 1 107
Table 5.5. Sensitivity analysis 2 107
Table 5.6. Sensitivity analysis 3 107
Table 5.7. Sensitivity analysis 4 108
Table 6.1. Task type in C2T 111
Table 6.2. Result of Experiment 6.1 124
Table 6.3. Result of Experiment 6.2 127
Table 6.4. Result of Experiment 6.2 128
Table 7.1. Relationship between research questions, contributions, and dissertation structure . 138

LIST OF ABBREVIATIONS

		Page Where First Defined
AD	Association and Dissociation	37
ARS	Agriculture Robotic System	22
AS	Adaptive Search Algorithm	14
BM	Best Matching	37
C	Conflicts	61
C2T	Cyber Collaborative for Plant Treatment	109
CCP-CPS	Collaborative Control Protocol for Cyber-Physical System	51
CCP-ED	Collaborative Control Protocol for Early Detection Stress in Plants	70
CCT	Collaborative Control Theory	23
C&E	Conflicts and Errors	23
CFT	Collaborative Fault Tolerance	37
CPS	Cyber-Physical System	23
CRM	Collaboration Requirement Metrix	72
CRP	Collaboration Requirement Planning	37
CRP-H	Collaboration Requirement Planning protocol for HUB-CI	109
CRS	Collaborative Robots Scheduling	118
CVC	Collaborative Visualization and Comprehension	37
D-AS	Dynamic Adaptive Search Algorithm	90
E	Error	74
ELOCC	Emergent Lines of Collaboration and Command	37
EPCR	Error Prevention and Conflict Resolution	37
EWP	e-Work Parallelism	37
FCFS	First Come First Serve	123
FN	False Negative	91
FP	False Positive	91
HITL	Human in the Loop	112

HUB-CI	HUB for Collaborative Intelligence	109
IoT/IoS	Internet of Things and Internet of Services	23
KISS	Keep It Simple, System	37
MDR-CPS	Monitoring, Detecting, and Responding – Cyber-Physical System	51
MRTA	Multi-Robot Task Allocation	47
NVM	Newsvendor Model	45
PA	Precision Agriculture	22
PCol	Precision Collaboration	23
R Layer	Resource layer	51
S-AS	Static Adaptive Search Algorithm	100
TAP	Task Administration Protocol	39
T Layer	Task layer	51
TN	True Negative	91
TP	True Positive	91
TSP	Traveling Salesman Problem	43
UR	Unplanned Request	110
WSN	Wireless Sensor Network	51
WSPT	Weighted Shortest Processing Time first	118

LIST OF SYMBOLS

CHAPTER 3

Indices		Page Where First Defined
m	Task (m = 1, 2, 3, …, $\mathcal{M}$)	51
n	Resource (n = 1, 2, 3, …, $\mathcal{N}$)	51
t	Time (t = 1, 2, 3, …, $\mathcal{T}$)	61

Parameters

$ARS_{Group}(t)$	ARS at multiple agents at time t	61
$Resource_n$	Resource n	51
$Task_m$	Task m	51
$Time_t$	Time t	51

CHAPTER 4

Indices

i, j	Location (i = 1, 2, 3, …, $\mathcal{I}$ and j = 1, 2, 3, …, $\mathcal{J}$)	75
m	Task (m = 1, 2, 3, …, $\mathcal{M}$)	72
n	Resource (n = 1, 2, 3, …, $\mathcal{N}$)	72
r	Constraint (r = 1, 2, 3, …, $\mathcal{R}$)	74
t	Time (t = 1, 2, 3, …, $\mathcal{T}$)	74

Parameters

α_j	Probability to fail to search at node j	76
$ARS_{Group}(t)$	ARS at multiple agents at time t	74
$ARS_{Single}(t)$	ARS at single agent at time t	74
β_j	Probability to fail to inspect at node j	76
c_{ij}	Travel cost from i to j	75
$N_r(t)$	Constraints r at time t	74

D	Total number of infected locations found	76
d_i	Number of infected plants found during operation at location i	76
M, N	Number of sampled locations in the greenhouse	85
$NIPF$	Number of infected plants found	84
ORE	Overall Robotic Effectiveness	84
P	Performance rate	84
p_j	Processing time at node j	75
p'_j	Extra processing time at node j	75
q_j	Searching time at node j	75
q'_j	Extra searching time at node j	76
$Resource_n$	Resource n	72
S	Success rate	84
$S_{ARS_{Group}}(t)$	Group of ARS' agents at time t	74
$S_{ARS_{Single}}(t)$	ARS' single agent at time t	74
T	Total available time	76
T'	Total processing time	76
$Task_m$	Task m	72
U	Utilization rate	84

Variables

x_{ij}	1 if the ARS traverse from i to j, 0 otherwise	75
y_i	1 if the ARS search at location i, 0 otherwise	75

CHAPTER 5

Indices

i	Location (i = 1, 2, 3, …, $\mathcal{I}$)	102

Parameters

A	Adaptive search is activated	93

A'	Adaptive search is not activated	94
α	Type 1 Error	90
β	Type 2 Error	90
C_H	Cost of inspecting the healthy plant	93
C_I	Inspection cost	92
C_o	Overage cost	93
C_S	Cost of inspecting the infected plant	93
C_u	Underage cost	93
M	Location	94
m_0	Infected started location	97
p	Probability that the plant develops diseases	97
P_H	Profit from finding healthy plant	92
P_S	Profit from finding stressed plant	92
q	Probability to propagate in a specific direction	97
R	Rejection region (Reject the null hypothesis)	91
R_0	Critical ration	94
S	Plant has stress	92
S'	Plant does not have stress	91

CHAPTER 6

Indices

m	Task (m = 1, 2, 3, …, $\mathcal{M}$)	111
n	Resource (n = 1, 2, 3, …, $\mathcal{N}$)	116
r	Robot (r = 1, 2, 3, …, $\mathcal{R}$)	109

Parameters

$C_{baseline}$	Total operation cost from the baseline procedure	125
C_{CRP-H}	Total operation cost from CRP-H	125
c_{mn}	Cost for tasks m performed by robot (or team of robots) n	111
C_{max}	System makespan	116

$\Delta Cost$	Cost-saving	125
$\Delta Time$	Time-saving	125
N_R	Number of robot or team of robots	115
N_T	Number of tasks	115
p_m	processing time of task m	117
R_r	Robot type r	109
S, S'	Schedule S, S'	119
$W_{baseline}$	Total weighted completion time from the baseline procedure	125
W_{CRP-H}	Total weighted completion time from CRP-H	126
w_m	Priority of task m	117

Variables

x_{mn}	1 if task m is performed by a robot or team of robots n, 0 otherwise	115
y_{mt}	1 if task m start at time t, 0 otherwise	116

ABSTRACT

Food security is a major concern of human civilization. A way to ensure food security is to grow plants in a greenhouse under controlled conditions. Even under careful greenhouse production, stress in plants can emerge, and can cause damaging disease. To prevent yield loss farmers, apply resources, e.g., water, fertilizers, pesticides, higher/lower humidity, lighting, and temperature, uniformly in the infected areas. Research, however, shows that the practice leads to non-optimal profit and environmental protection.

Precision agriculture (PA) is an approach to address such challenges. It aims to apply the right amount or recourses at the right time and place. PA has been able to maximize crop yield while minimizing operation cost and environmental damage. The problem is how to obtain timely, precise information at each location to optimally treat the plants. There is scant research addressing strategies, algorithms, and protocols for analytics in PA. A monitoring and treating systems are the foci of this dissertation.

The designed systems comprise of agent- and system-level protocols and algorithms. There are four parts: (1) Collaborative Control Protocol for Cyber-Physical System (CCP-CPS); (2) Collaborative Control Protocol for Early Detection of Stress in Plants (CCP-ED); (3) Optimal Inspection Profit for Precision Agriculture; and (4) Multi-Agent System Optimization in Greenhouse for Treating Plants. CCP-CPS, a backbone of the system, establishes communication line among agents. CCP-ED optimizes the local workflow and interactions of agents. Next, the Adaptive Search algorithm, a key algorithm in CCP-ED, has analyzed to obtain the optimal procedure. Lastly, when stressed plants are detected, specific agents are dispatched to treat plants in a particular location with specific treatment.

Experimental results show that collaboration among agents statistically and significantly improves performance in terms of cost, efficiency, and robustness. CCP-CPS stabilizes system operations and significantly improves both robustness and responsiveness. CCP-ED enabling collaboration among local agents, significantly improves the number of infected plants found, and system efficiency. Also, the optimal Adaptive Search algorithm, which considers system errors and plant characteristics, significantly reduces the operation cost while improving performance.

Finally, with collaboration among agents, the system can effectively perform a complex task that requires multiple agents, such as treating stressed plants with a significantly lower operation cost compared to the current practice.

CHAPTER 1. INTRODUCTION

1.1 Research Motivation

To secure food production, the farmer grows plants in a greenhouse. While greenhouse plants are well nurtured and are provided with almost perfect conditions to grow, they are far from precision controlled. Fluctuations in environmental parameters such as temperature, humidity, and airflow cause stress and lead to some initial state of diseases in plants. In the US, crop stresses which are caused by environmental condition change causes damage more than $200 billion worth between 1980 and 2012 (Suzuki, Rivero, Shulaev, Blumwald, & Mittler, 2014). To minimize the severity of the production loss in crops, an abnormal condition of plants such as stresses must be detected, localized, and treated early (Gueroui & Labraoui, 2015). Currently, farmers apply resources (i.e., water, fertilizer, and pesticide) evenly over the agriculture area as aim to prevent diseases and yield loss (Mandal & Ghosh, 2000). It, however, wastes resources, money, and time as previous research reports that over or under water may fail to boost crops yield (Drechsel et al., 2015). In addition, improper fertilizing can damage both financial return and crop production (Ribaudo et al., 2012). Hence, uniform resource management in agriculture may not lead to the desired outcome as it damages both profit and environment.

Precision Agriculture (PA), which aims to apply the right amount of resources to the right location at the right time by utilizing information technologies, is a promising approach which gains much attention now. PA can ensure the proper treatment for each plant while minimizing unnecessary resources used (Thompson, Bir, Widmar, & Mintert, 2019). Also, the ability to detect and localize stress in plants early can strengthen PA benefit as the cost of treatment in the early state is typically lower (Mahlein, 2016). Moreover, to deal with an unexpected situation in the agricultural area, smart agriculture and advanced technologies with PA, called Agriculture Robotic System (ARS), are necessary to enhance the advantages. Farmers should be able to work and monitor crops remotely, and, with a decision support system, they can deal with a current and unexpected situation correctly. Therefore, the smart and precision agriculture will reduce the overall cost of food production while ensuring human food security and minimizing damage to environment.

Definition 1.1. *Agriculture Robotics System* (ARS) is a smart agriculture system that utilizes the strength of agents to optimize system performance with limited resources.

The definition addresses the objectives and challenges of ARS. Advanced technologies such as sensors and agriculture robotics together with new solutions such as Cyber-Physical System (CPS), Internet of Things and Internet of Services (IoT/IoS), Task Administration Protocol (TAP), Collaborative Control Theory (CCT), and Precision Collaboration (PCol) must be utilized to develop a system and its protocol for supporting the ARS. The ARS consists of multiple sensors, human operators, robots, algorithms, and other supporting tools. Because each system agent has its strengths and weaknesses, collaboration among agents is critical. For example, although sensors have advantages to monitor and collect data which can be analyzed to create a proper treatment for each plant (Gongal, Amatya, Karkee, Zhang, & Lewis, 2015), they cannot work without agricultural robotics and human operators (Min Hyuc, Beom-Sahng, Kyoung Chul, Suprem, & Mahalik, 2015). Besides, an algorithm that provides output for a particular situation also requires input from sensors. Lastly, communications among system agents will be less efficient and contain conflicts and errors (C&E), if no synchronization workflow is established. Therefore, ensuring collaboration among agents is necessary to provide the optimal outcome of the agricultural system.

System analytics and operations research have been utilized for more than five decades to help make better decisions in business, production, supply chain, and healthcare. Decisions — especially in a structured and error-free environment — are studied extensively and in many cases, are solved to optimality. On the other hand, the unstructured system is still a challenge for researchers. Because of complexity, the system usually trades optimality for higher flexibility or lower cost. In this study, an agricultural system, which is relatively unstructured and prone to error by nature, is selected as a representative of unstructured system. Therefore, this study not only address issues in agriculture robotic system, but also the optimal operations in an unstructured system are solved.

1.2 General Assumptions

In this section, the general assumptions which are assumed throughout the study are discussed. There are four types of assumptions; research scope assumption, modeling assumption, agent assumption, and terminologies assumption.

1.2.1 Assumption 1: System scope assumption

The ARS comprises humans, robots, and sensors. Humans are the decision makers who will solve complex, unanticipated real-time problems. The mobile robot will be guided to selected, assigned locations for inspecting plant samples at those locations. From system engineering perspectives, the robot that moves through required locations needs to be equipped with arms and sensors to monitor the conditions at each spot (Edan & Miles, 1994). The robot will carry several types of sensors which contain detecting agents.

Where crops are grown, especially in relatively large areas, it is hard to inspect every single plant to check its status and detect whether it is under stress. Hence, every day, a representative sample of plants is selected for assessment and monitoring. Utilizing the plant sampling approach, the chosen sample of plants can be assumed to represent that local area, thus saving time and cost. Robot-Human Base Point (Figure 1.1) is the location where robots and humans are located. Data collected during the monitoring process are transferred to the Robot-Human Base Point for further analysis (if needed). Figure 1.1 presents the description above, which is the environment of the research in this study.

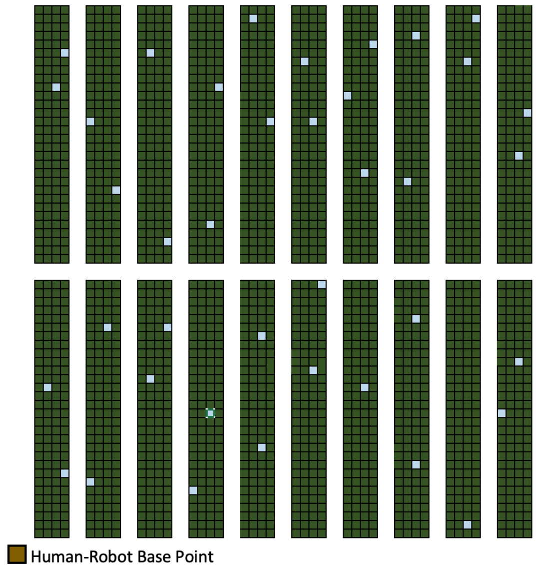

Figure 1.1. System scope assumption

1.2.2 Assumption 2: Modeling assumption

In this study, input parameters of the models such as the number of agents, agent speed and capabilities, greenhouse structure, other modeling inputs are assumed to be known as the parameters that can be measured before designing the system.

Moreover, the parameters are also assumed to be constant and not changed over time. In other words, there are no learning effect impacts on the parameters. With this assumption, in the situation that humans and machines are involved in the monitoring system, the system and analysis are for the worst-case scenario.

The uncertainties of the input parameters, however, are considered in models. The distribution of the input parameter represents the uncertainties of parameters. For example, inspection time, travel speed, and the robot's arm movement speed are the parameters that are assumed to have a normal distribution.

Some parameters, such as C&E and stressed plant locations, are assumed to exist in the system. The exact location, status, and value, however, are unknown. This assumption reflects the real situation of the system, where parameters have existed in the system, but the system designers do not know the real value.

1.2.3 Assumption 3: Agents assumption

The third assumption relates to agents and the capabilities of agents. In the study, agents such as humans, robots, and sensors are employed to perform a specific task. As discussed earlier and presented in Table 1.1, humans are assumed to be superior to other agents in terms of real-time decision making. On the other hand, robots are good at repetitive tasks. Lastly, sensors are responsible for collecting a massive amount of data. Base on the task descriptions, agents are assumed to be able to work independently. For example, robots can work by their capability regardless of whether other agents are working (but it may cause an inferior performance). Robots can move and approach plants even though sensors are ready to operate or not. Also, some agents such as sensors and IoT/IoS may need support from other agents to perform a completed process, but there are no condition sequences between agents' operations.

With the independent working condition, the system agents are able to work in parallel, which supports the e-Work Parallelism principle in CCT. Because of the parallelism, the system operation time can be minimized, unlike sequential operation. The system, however, has a higher probability of containing C&E. Thus, the Error Prevention and Conflict Resolution design principle of CCT must be appropriately utilized to balance the benefit gains from the parallel work design and loss from the C&E in the system.

Table 1.1. Main role and responsibility of agents

Main agents	**Roles of agent**
Humans	Real-time decision-makers; Solve unexpected situations
Robot, manipulator, and mobile robot cart	Move into a greenhouse; Approach specific parts of plants; Computing and responding to the command;
Sensors	Collect data; Transmit data

1.2.4 Assumption 4: Terminologies assumption

There are common terms that are used throughout the studies. First, the term ARS refers to the entire monitoring system, not just a single agent. The reason is that an agent in the system cannot work to deliver the desired output. The system agents need to work in order to achieve the system goal collaboratively. Therefore, ARS must represent the entire system as a team.

In ARS, three main agents are humans, robots, and sensors. Humans include farmers, researchers, and experts who can make a judgment for most of the situation. Humans have higher knowledge comparing to other agents.

Robots refer to the robot cart, robot's arms, and other manipulators that move into the greenhouse, approach plants in multi-direction, and connect to the command from humans in the case of emergencies. The robots also include treatment robots that are dispatched in the case of stress and disease treatment.

Lastly, sensors refer to multi-spectral cameras, thermal sensors, video cameras, high-definition cameras, and other IoT/IoS devices, which responsible for collecting data and detecting the condition of plants.

1.3 Research Problem and Research Questions

1.3.1 Research problem

To increase food security, preventing production loss of crops is crucial. Because plants usually develop stress before the diseases emerge, detecting stress as early as possible, and treat the stress precisely, are not only prevent yield loss from crops diseases but also reduce the production cost of food. The key challenge is, therefore, with the limited resources available, to identify stress locations in order to design the proper treatment before diseases developed.

1.3.2 Research questions

The following research questions (RQ) are defined for attempting to address the research problem. The RQs are as follows.

Research question 1 (RQ_1)

How can we design and develop the CPS framework, which can combine algorithms, sensors, robots, humans, and other agents to work effectively and facilitate real-time communications for the greenhouse system?

Research question 2 (RQ_2)

How can we design the cyber collaborative protocol for the agents in a greenhouse under the CPS framework to perform their job to maximize the system performance, reflect real-time characteristics of plants, and utilize available time most effectively, even optimally, for the earliest detection of stress in crops?

Research question 3 (RQ_3)

How can we effectively utilize the new information found during the inspection process to provide the optimal collaborative interaction procedures between agents to maximize system performance during the monitoring process?

Research question 4 (RQ_4)

How can we develop a protocol which effectively and collaboratively manages agents to treat the stressed plants that may require a specific type of agent or interaction among multiple agents?

1.4 Dissertation Structure

The dissertation is organized as follows: Chapter 2 provides a summary of current research related to the topic. Chapter 3 introduces the CPS for the agricultural robotic framework and the validation of the benefits of CPS in agriculture. Chapter 4 presents the mechanism at the agent level, which collaboratively works by the collaborative control protocol. Chapter 5 presents an analysis of the agent procedure. Chapter 6 describes the methodologies and validation of collaborative agent procedures to respond to emergencies. Finally, Chapter 7 presents conclusions, discussion, and future challenges.

CHAPTER 2. LITERATURE REVIEW

2.1 Agriculture Robotics System (ARS)

For decades, researchers and engineers have utilized agriculture robotics for working in an agricultural field. Because of the advancement in sensors and robotic technology, the ideas of applying robots in the unstructured area such as agricultural field and assist other agents such as humans and sensors become feasible (Belforte, Deboli, Gay, Piccarolo, & Ricauda Aimonino, 2006; Gay, Piccarolo, Aimonino, & Deboli, 2008; McIntosh, 2015). As automation and robots are good at repetitive tasks and able to work in the long period of time continuously, they usually are responsible for routine operations such as irrigation, harvesting, and inspection, and help or replace farmers (Keicher & Seufert, 2000; Reid, Zhang, Noguchi, & Dickson, 2000). Also, in extreme weather conditions, robots with smart technology can work with reliable results (Pedersen, Fountas, Have, & Blackmore, 2006). Hence, automation and robots are commonly used for the agriculture system in various fields.

Greenhouse automation and its operating system are the focus as the greenhouse environment capture challenges form not only an open field environment but also indoor issues. In a greenhouse, plots are organized as in the open field but have limited space to maneuver and visualize. Also, GPS and connection ability in a greenhouse are limited. Automation and robots which work in the greenhouse, thereby, are designed and prepare for such challenges and constraints.

The typical greenhouse operations which are focused by researchers on applying automation and robots are harvesting robots, data collection and crop inspection robots, environmental control systems, spraying robots, and yield improvement robots. The following section describes and explains each of them.

2.1.1 Harvesting Robot

Even though researchers and engineers put substantial effort into developing a precision harvesting system for fruits and vegetables, the success rate still low, and the issue still challenges (Bac, Henten, Hemming, & Edan, 2014). A series of harvesting system of agricultural products such as oranges, tomatoes, and strawberries which are developed for the past 30 years shows development and attempts of harvesting robotic system. Examples of harvesting approaches for PA are apple harvesting robot (Silwal et al., 2017), strawberry yield monitoring and picking carts (Khosro Anjom, Vougioukas, & Slaughter, 2018), orchard robotic bin handling (Ye et al., 2018), and kiwi automation technology (Mu, Liu, Cui, Fu, & Gejima, 2018).

With the current approach, fruits and vegetables harvesting robots are far from commonly used in a practical setting. The main challenges are the diversity of agricultural products, limitation of the machine to deal with the real-time situation (e.g., inaccurate operation, high cost, slow procedure, and inflexible tools) (Spekken & Bruin, 2013).

2.1.2 Data collection and crops inspection robots

Using PA, data for each location are critical for analyzing and predicting crop output, such as fruit mass or number of leaves (Figure 2.1). Robots, therefore, are utilized for collecting data at each location in the agriculture area. For example, a watch-dog robot (Nagasaka et al., 2004), and a weed control robot (Astrand & Baerveldt, 2002) are sensor integrated robots that can both work in the field and collect data at the same time.

Not only the data collection task, but robots with sensors are also able to perform crop inspection and disease detection tasks (Liao et al., 2017). In the past, it requires workers to walk and inspect each plant, which yields low accuracy and unreliable outcome. Agricultural robotics, however, improve the speed and accuracy of the inspection system because of an improvement of automation accuracy and sensor performance (i.e., more reliable, faster, more accurate, and requires less energy) (Sai, Fan, Yuliang, Lei, & Yifong, 2016). Moreover, by using sensors which can perform a non-contact inspection, contamination, and spread of diseases are minimized. Figure 2.2 shows diseases and stress monitoring by the robot in the laboratory.

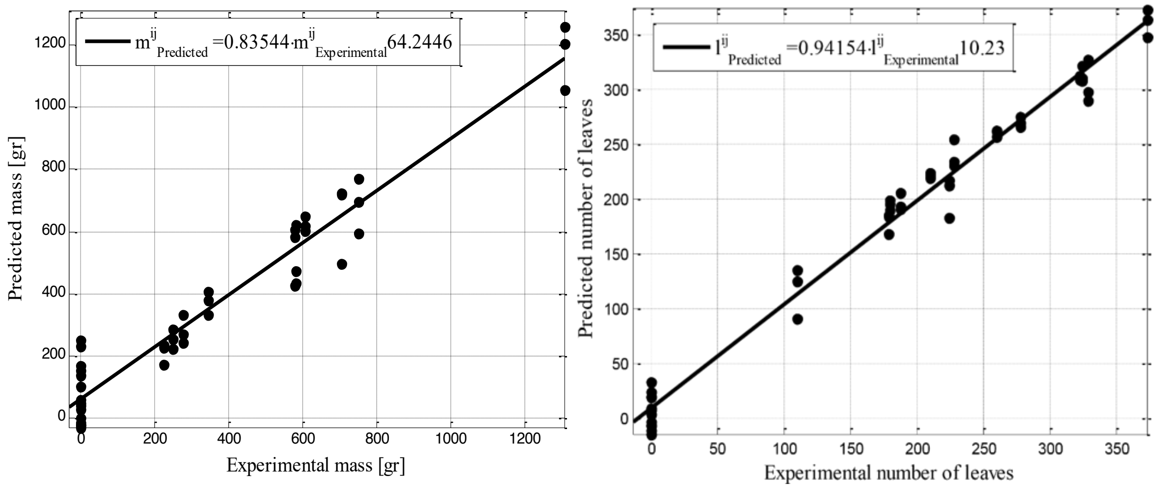

Figure 2.1. (Left) Fruit mass prediction; (Right) Number of leaves prediction (Finkelshtain, Bechar, Yovel, & Kósa, 2017)

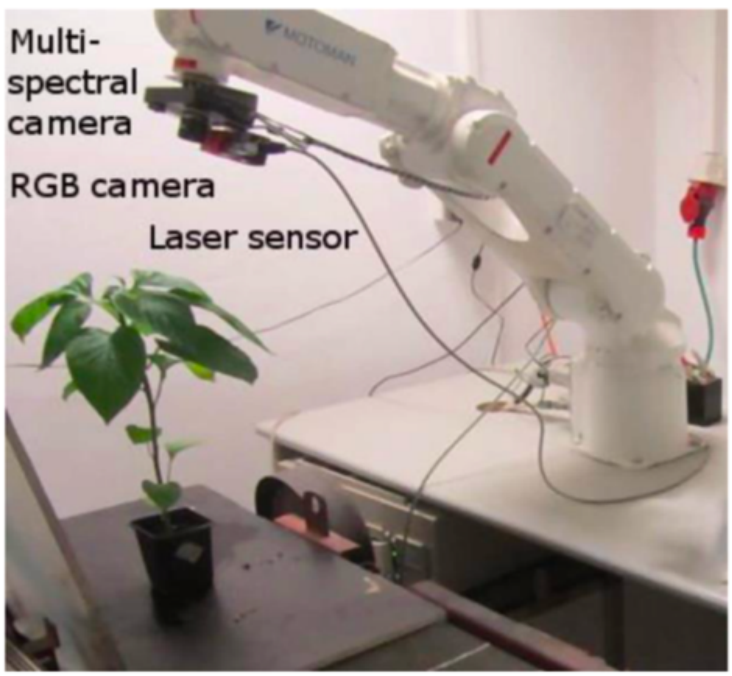

Figure 2.2. Disease and stress monitoring by the robot (Schor et al., 2016; Schor et al., 2017)

2.1.3 Environmental Control System

In a greenhouse, environmental parameters such as light density, temperature, humidity, CO_2 concentration, and airflow can be manipulated accurately (Simonton, 1990) because of the sensor and microcomputer (Kondō, Monta, & Noguchi, 2011). Therefore, farmers can extend the growing period, shorten the production cycle, and maximize crops output. Climate control models in a greenhouse which require various input variables influence both weather condition inside greenhouse and growth rate of crops. The models utilize proportional integral derivative control, cascade control, nonlinear control, predictive control, and adaptive control in order to have a closed-loop system (Albright, 2002; Bailey, 2006; Essahafi & Lafkih, 2018). Because of the feedback loop, the models can also provide a suggestion for parameter setting and change controlled parameters through wireless sensors and auto-aerial control system in the greenhouse (Diker & Bausch, 2003; Ferentinos, Katsoulas, Tzounis, Bartzanas, & Kittas, 2017; Manfreda et al., 2018).

Internet network also supports a connection in a greenhouse control system. Researchers develop automation and robots which are connected and managed through mobile/web applications and the internet to perform specific tasks (Chebrolu et al., 2017; Ishibashi, Iida, Suguri, & Masuda, 2013). The internet network in the greenhouse also enables human-in-the-loop design, which can both simplify the system and improve system performance in dealing with unexpected situations (Bechar, Meyer, & Edan, 2009). For example, a human, robot, and sensors design system for monitoring greenhouse crops show superior performance when humans are integrated into the system (Guo, Dusadeerungsikul, & Nof, 2018).

2.1.4 Spraying Robot

An automation sprayer (Figure 2.3) can apply specific fertilizer, pesticide, liquid chemical, or water to a specific crop location in the greenhouse (Min Hyuc et al., 2015). With an unmanned spraying robot, farmers have minimized the exposure to chemical substances. The advancement of unmanned robots such as drone which can take-off and landing in the limited space has a very high potential in the near future to use in agricultural context (Cappelleri & McArthur, 2019).

Figure 2.3. An autonomous sprayer (Dar, Edan, & Bechar, 2011)

A machine visualization system (Figure 2.4) and a selective sprayer generating map and guidance for automation to location/site-specific treatment have been developed to solve an issue about autonomous control for unmanned sprayer (Kunz, Weber, Peteinatos, Sökefeld, & Gerhards, 2018; Xue, Zhang, & Grift, 2012). Moreover, a review of alternative spraying methods and solutions which can be applied to the PA context is provided (Bechar & Vigneault, 2016, 2017).

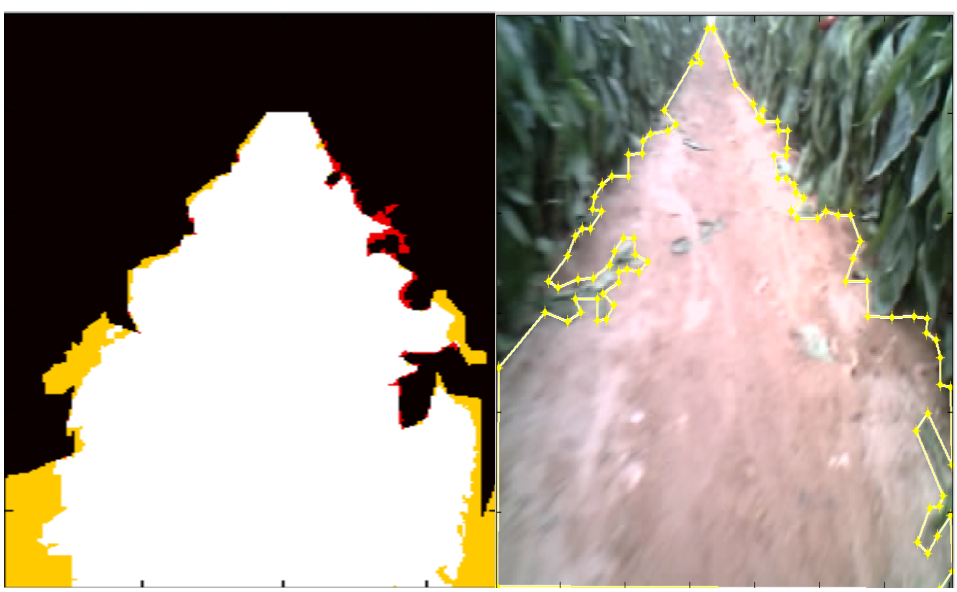

Figure 2.4. Machine visualization for mapping in the greenhouse (Dar et al., 2011)

2.1.5 Yield Improvement Robot

The last category is of ARS is the yield improvement robot. The yield improvement robot is automation, which helps farmers increase the quality and quantity of agriculture outputs. Examples of the robots in this category are tree pruning robot and seedling production automation. Figure 2.5 shows a tree pruning robot that can both calculate the path for reaching the tree branch as well as prune the targeted tree branch accordingly. In addition, the robot can suggest the tree branches to prune in order to optimize agriculture outputs. The machine operations are first, calculate the optimal trajectory (Figure 2.5 Left) and then prune the unnecessary branches by using end-effector (Figure 2.5 Right).

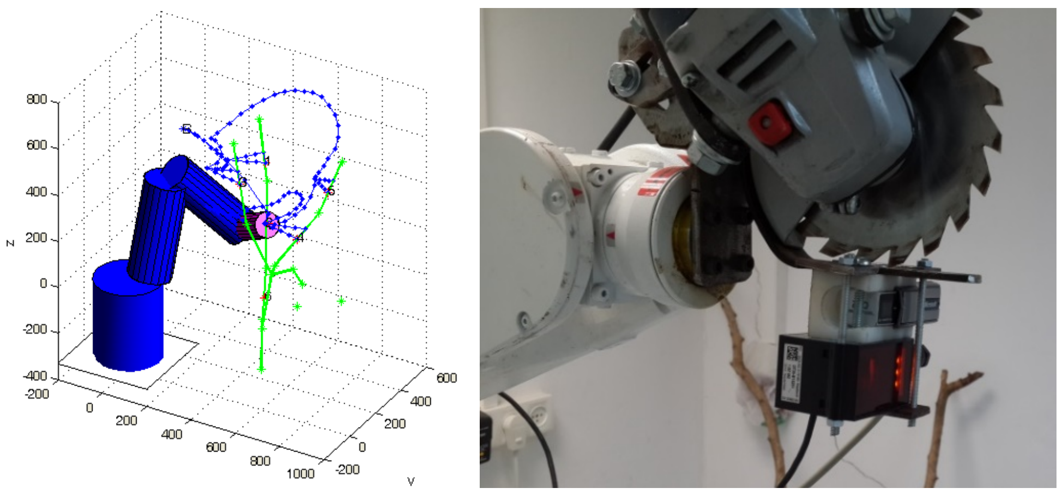

Figure 2.5. (Left) Computing optimal trajectory tree pruning; (Right) End-effector for pruning (Bechar, Nof, & Wachs, 2014)

Another example of yield improvement automation is seedling production automation. As seedling production is an essential factor in ensuring the high-quality agriculture product, attempts to develop an automation for seedling production are prevalent. Machines that are either semi-automated, such as selecting seed, seeding, transplanting, grafting, cutting, and sticking (Kondo & Ting, 1998) or fully automated (Mitsuhashi, Yamazaki, & Shichishima, 1994) have been designed and developed to help farmers perform such tasks. A nursery greenhouse containing systems such as the germination chamber, tray support system, irrigation system, heating system, and cooling system are developed to ensure the optimal growing conditions for young seedlings (Balliu, Sallaku, & Nasto, 2017). Besides, robots have been applied and commercialized to various operations such as grafting and transplanting as they can improve the product quality, crop yield, and disease resistance. Also, the IoT/IoS in the greenhouse becomes an essential element in seedling production automation as young seedling are sensitive to abiotic and biotic stresses (Gonzalez-Amarillo et al., 2018).

Although robots and agricultural machinery can help farmers improve task performance, they cannot overcome obstacles that prevent PA in real implementation. Challenges mainly come from constraints and characteristics of agricultural tasks which are relatively unstructured (Edan, Han, & Kondo, 2009), and may cause C&E in the system. Taking a crop monitoring process as an example, the automation needs to deal with unclear data during the inspection process. More importantly, the system may contain the discrepancy of data, which leads to C&E during the operation. With the current approach, the automation cannot deal with such expected or unexpected C&E. Also, the capability to respond to the diversities of products, interactions among agents, and requests to the real-time information are limited. Lastly, knowledge-based response and operation predictability by historical data are not utilized effectively. Hence, the current automation approach suffers from dealing with mentioned challenges, and PA cannot be implemented in the practical setting.

Therefore, to overcome such shortcomings, a new support system is necessary. Such support systems are expected to improve productivity, reliability, safety, and continuity of the system, optimize work methods, increase accuracy, reduce waste, and as a result, provide better crop quality and higher yield (Dusadeerungsikul & Nof, 2019).

2.2 Cyber-Augmented System in Agriculture

2.2.1 Cyber-Physical System (CPS) in Agriculture

In agricultural CPS, robots are considered to help humans perform an operational task such as moving in a greenhouse and approach the suspicious areas. Sensors which are faster in detection than other devices are mounted on the robot to perform inspection task. Sensors transmit data gathered from plants to agricultural experts, which remotely monitor conditions of crops in real-time. The experts can consider a plant to deal with the situation at each particular location. For example, a new mobile robot may be dispatched for taking pictures or videos and transmitting by WIFI or 4G technologies for further investigation, or a robot may order to treat the plant by applying fertilizer to the location.

The cyber-support system for agriculture described above can be defined as an agricultural Cyber-Physical System (CPS). The CPS, which utilizes information/knowledge technologies, cybernetics, and high-performance computers, can effectively combine and connect agents and features, i.e., wireless communication, real-time control, intensive computing, and brain-inspired models.

Because of the advancement and re-evaluation of information and communication technologies, CPS has been extensively used in the 21st century in various complex systems such as medical device operation, traffic control, safety engineering, infrastructure control, and agricultural robotics (Zhong & Nof, 2015). With support from the cyber-augmented system and abilities to compute and operate in real-time, CPS can solve an unexpected situation spontaneously. For instance, a CPS and multi-sensor system support a precision pesticide spraying task by minimizing the amount of chemical applied to plants (Stark, Rider, & Chen, 2013). Because of the complexity of CPS architecture, a three layers system, the physical layer, the network layer, and the decision layer; is detailed designed and applied to PA (Nie, Sun, & Li, 2014). Lastly, utilizing the benefit from real-time response and advancement of new technologies, a CPS framework of monitoring, detecting, and responding cyber-physical framework has been designed and validated to have superior performance than the traditional approaches (Guo et al., 2018). Researchers are extensively used CPS developments for PA as presented in the current research articles (Biradar & Shabadi, 2017; Cimino et al., 2017; Dong, Vuran, & Irmak, 2013; Goap, Sharma, Shukla, & Rama Krishna, 2018; Morimoto, 2018; Perez-Exposito, Fernandez-Carames, Fraga-Lamas, & Castedo, 2017). An agricultural CPS enables system to have better intelligence and real-time control; therefore, move the system another step closer to PA objective.

2.2.2 Internet of Things and Internet of Services (IoT/IoS) in agriculture

The advancement of technologies narrows down a gap between a physical object and cyber layers. The self-contained components (IoT) and the self-contained actions/services (IoS) are essential links to bridge such gaps. IoT/IoS, which allows the decentralized and distributed control enables researchers to intergrade isolated systems as well as connect the physical and cyber system together (Moghaddam & Nof, 2018). Moreover, to deal with a current problem in which its complexity grows exponentially, the system needs abilities to receive new information and reflect the situation emerged (Alberts, 2014; C.-Y. Huang, Ceroni, & Nof, 2000; Putnik et al., 2013). IoT/IoS, which are components and equipment in a system, can support such requirements by obtaining/transmitting data from/to network systems (Wu, Dai, & Dai, 2013) and adapting themselves according to the situation in real-time. Also, because of IoT/IoS, components, services, and human operators have effectively connected to the network and allow human-in-the-loop design. With utilizing IoT/IoS advantages, researchers can explore new system designs and overcome the existing challenges.

Because of the mentioned IoT/IoS capabilities, IoT/IoS are applied in various applications such as smart transportation, environment monitoring, warehouse, and factory of the future (Dusadeerungsikul, He, Sreeram, & Nof, 2020; Madakam, Ramaswamy, & Tripathi, 2015). IoT/IoS is also considered to be a support operation in smart and precision agriculture. IoT/IoS are also utilized as detection agents and connection agents in a greenhouse monitoring system (Dusadeerungsikul, Nof, Bechar, & Tao, 2019; Guo et al., 2018). With the active collaboration and connection among agents by IoT/IoS, the system shows superior performance in terms of stress detection, robustness, and response time.

2.3 Collaborative Control Theory (CCT)

2.3.1 Design principle of CCT

Collaborative Control Theory (CCT) is the design principles for the multi-agent system. With the nine CCT principles which facilitate multiple agent interactions and collaboration, engineers can effectively design a complex system that has higher performance. The nine CCT principles are 1) Collaboration Requirement Planning (CRP), 2) e-Work Parallelism (EWP), 3) Keep It Simple, System (KISS), 4) Error Prevention and Conflict Resolution (EPCR), 5) Collaborative Fault Tolerance (CFT), 6) Association and Dissociation (AD), 7) Emergent Lines of Collaboration and Command (ELOCC), 8) Best Matching (BM), and 9) Collaborative Visualization and Comprehension (CVC) (Nof, Ceroni, Jeong, & Moghaddam, 2015).

Engineers and researchers have been applied CCT principles in various fields to design the response and control mechanism. Some examples are to respond to a market uncertainty of demand and supply, an algorithm for prevention and detection C&E, network design, telerobotic collaboration, and protocol designs (Chen & Nof, 2012; Ko & Nof, 2012; Moghaddam & Nof, 2014; Nof, 2007; Zhong, Wachs, & Nof, 2013).

In agriculture fields, researchers and engineers have been applied CCT to design and plan agriculture tasks. For example, Asynchronous Cooperation Requirement Planning, which can improve harvesting and grasping performance and cost of agriculture robots, has been designed and developed (Zhong, Nof, & Berman, 2015). CPS framework with CCT principles is designed for Monitor, Detection, and Response in CPS for greenhouse crops (Guo et al., 2018). The designed framework yields better detection performance and response to an unexpected situation. Also, in the Collaborative Control Protocol for Early Detection of Stress in Crop (Dusadeerungsikul & Nof, 2019), the designed protocol for managing multiple agriculture agents utilizes CCT for assigning tasks to agents and preventing/resolving system error/conflicts.

Based on the examples mentioned, the full range applications of CCT are explored and yield better planning/control operation and service systems. To achieve PA, CCT methods and approaches can support and utilize (Nof, 2015). Table 2.1 shows PA tasks, CCT methods, as well as their challenges.

Table 2.1. Challenge in Precision Agriculture with CCT principles and methods

PA Task	Related CCT Principle(s)	CCT Method(s)	Challenge(s)
Harvesting	CRP; EWP; KISS; EPCR; AD; BM	Humans-Robots-Sensors team	Agent collaboration in CPS for given tasks in PA
Data collection and crops inspection	CRP; EPCR; CFT; ELOCC; BM; CVC	Humans-Robots-Sensors team algorithm and protocol	Multiple agent collaboration; conflicts prevention and errors resolution;
Environmental control	CRP; EWP; EPCR	DHM-R tools	Collaborative machine learning
Spraying	CRP; EWP; EPCR; CVC	Swarms robot	Collaboration for precision operation
Yield improvement	EWP; EPCR; AD; ELOCC	Demand and Capacity Sharing	Could communication for collaborative control decision support system

2.3.2 Task Administration Protocol (TAP)

Integrations, communications, and interactions among agents distributed and decentralized have increased as the complexity of the system and the problem growth. Hence, an agent collaboration protocol, called Task Administration Protocol (TAP), is required for effective works and operations in the system (Nof et al., 2015). TAP, which has an objective to maximize system performance, is a coordination and collaboration protocol managing agents, algorithms, and databases with rules, heuristics, and interaction procedures. Three main planning and control elements in TAP are Task i ($Task_i$), Resource j ($Resource_j$), and Time k ($Time_k$). $Task_i$ is an activity or operation which requires agents ($Resource_j$) to operate. $Resource_j$ is an agent which can perform a given $Task_i$. $Time_k$ is a specific point of time and duration of time which $Task_i$ will perform by the assigned $Resource_j$.

TAP has helped engineers and researchers to design the effective workflow and coordination procedure in various fields such as workflow design, task allocation design, time-out design, and Petri-net model design (Anussornnitisarn, Nof, & Etzion, 2005; Ko & Nof, 2008; Nof et al., 2015). With the TAP framework, the TAP protocol outperforms a non-TAP protocol because of the active synchronization among agents (Ko & Nof, 2010). In PA, the TAP framework is also applied to design protocol for early detection stress in greenhouse crops, called Collaborative Control Protocol for Early Detection of Stress in Plant (Dusadeerungsikul & Nof, 2019). The protocol manages both system agents (humans, robots, and sensors) and algorithms and yields superior performance in resource allocation and utilization.

2.3.3 Precision Collaboration (PCol)

A collaboration support system which can work, collaborate, coordinate, and communicate precisely and smartly, called Precision Collaboration (PCol), is considered as a useful element for PA. The support system can improve the interaction between inputs/outputs and its performance measurement (Bechar et al., 2014; Bechar, Nof, & Wachs, 2015).

In PCol, the system can include physical, logical, and virtual agents to perform a given task by collaboration procedure. Because of PCol that can prevent and resolve system C&E, the system with complex interactions will be robust and can work effectively. Therefore, to achieve PCol, it requires an integrated communication, synchronization, control, information, and analysis to have a PCol for an exact position, identification, recognition, location, dimensions, proximity, awareness, and other attributes (Bechar, Wachs, Lumkes, & Nof, 2012).

Figure 2.6 presents the Human-Computer-Robot (HCR) PCol framework (Nof, 2015). There are four possible collaborations; Human-Human (H-H), Human-Computer (H-C), Human-Robot (H-R), and Robot-Robot (R-R) (Dusadeerungsikul, Nof, & Bechar, 2020). For example, a simple object targeting and monitoring tasks for a specific location (Figure 2.7) require R-R type collaboration – robot manipulator and robot cart. The laser sensor is utilized for spotting location for a precise target, which helps the collaboration more efficient. On the other hand, mapping, localizing, and human detection tasks (Figure 2.8) are more complex and need multiple collaboration types; H-C collaboration – human operator and computer/sensor detection, H-R collaboration – human operator and robot cart, and R-R collaboration – robot manipulator and robot cart. In this case, a laser sensor is utilized for scanning maps, communication between agents, transmit signal, and detecting an object.

As presented in Figure 2.6 and examples, a laser is an essential technology for PCol. It has the unique ability in accurately and rapidly transfers signal, data, and energy. The roles of laser in PCol are presented in Figure 2.6.

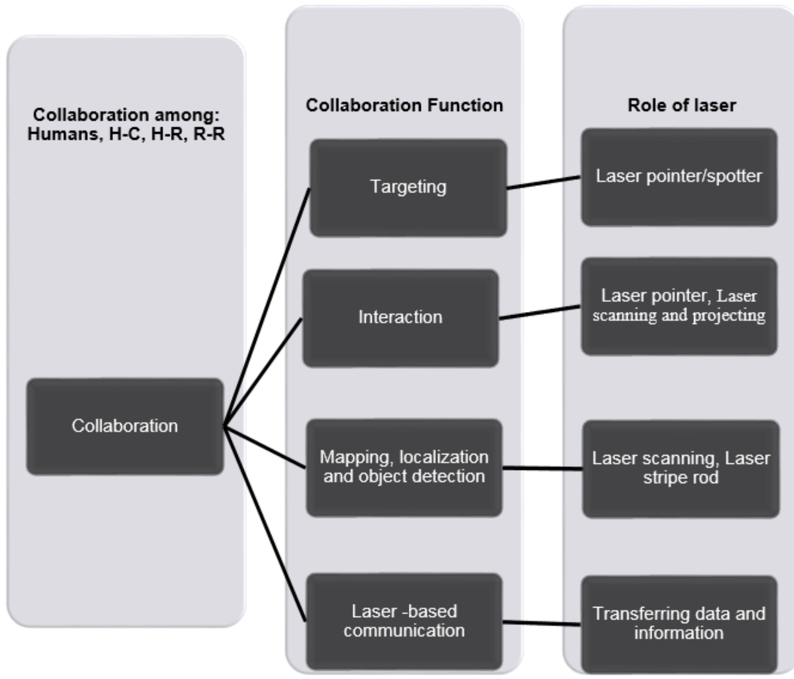

Figure 2.6. Precision Collaboration Support Framework (Nof, 2015)

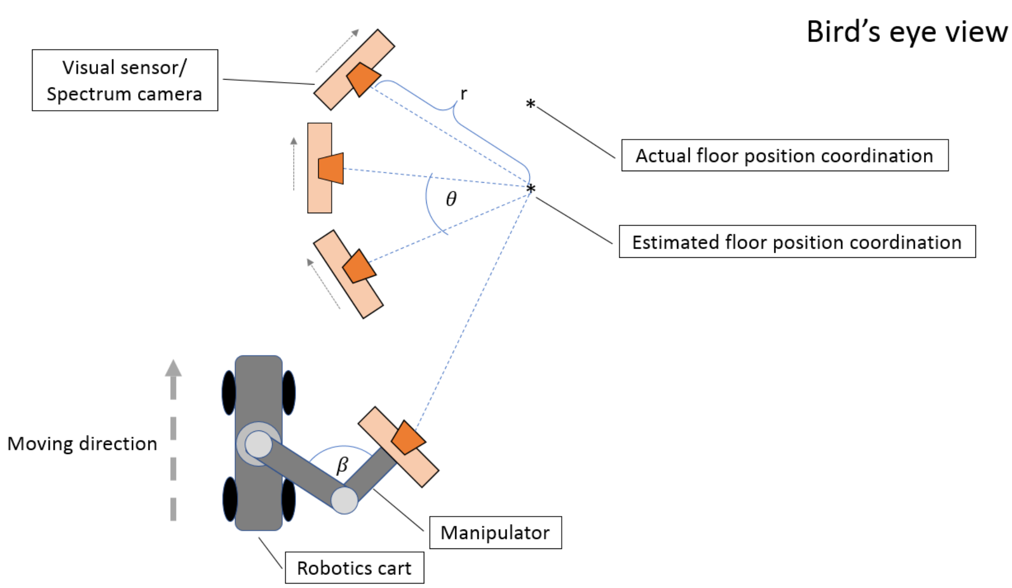

Figure 2.7. Object targeting and monitoring

Figure 2.8. Mapping, localizing, and human detection tasks

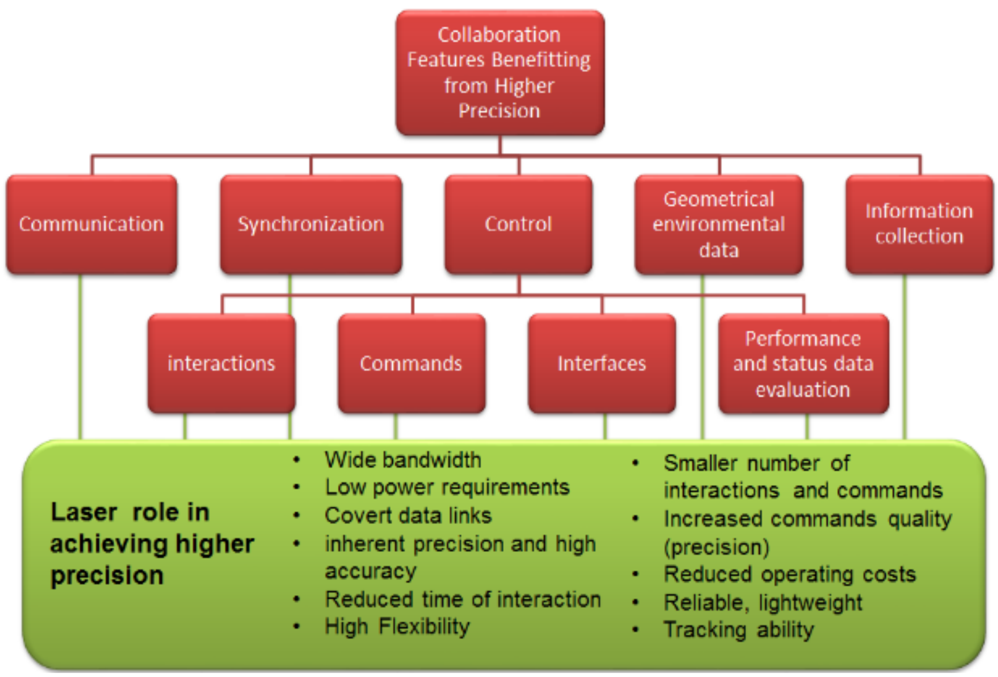

Figure 2.9. Collaboration support features and their precision requirements (Bechar et al., 2012)

2.4 Optimization in Agriculture

2.4.1 Traveling Salesman Problem for Robotic Monitoring and Inspection

In general, if we need to develop a path for visiting every desired node once, the Traveling salesman problem (TSP) is used to describe this situation. TSP is a well-known NP-hard problem that many researchers have explored extensively in numerous variations. In TSP, given n integer nodes and n-dimensional square matrix of the distance between nodes, the objective is to find a tour that visits each location once with the lowest total cost (Bellmore & Nemhauser, 1968). Solving real cases, TSP is relatively difficult to apply. Methods such as dynamic programming, branch and bound, and other heuristics can be used for solving TSP. Heuristics approaches were developed to find a good feasible solution in different scenarios (Lin & Kernighan, 1973). Genetic algorithm (GA) is a popular algorithm to obtain the solution for a combinatorial optimization (Jun, Lee, & Chun, 2019). Research has been explored GA to solve TSP because it can provide an acceptable solution within a limited time, even though the optimal solution is not guaranteed by solving the problem with a heuristic approach (Nagata & Kobayashi, 2013; Potvin, 1996). Other bio-inspired algorithms, such as ant colony system and neural network, are also developed to solve TSP. Both algorithms provide similar performance in terms of the quality of the result and numeric computing (Xiao, Tao, & Chen, 2012). Besides, a novel bio-inspired algorithm named elephant search algorithm can be applied to solve TSP with stable performance comparing to other metaheuristics. The performance of the algorithm in terms of fitness value, however, is inferior to other algorithms (Deb et al., 2016).

Currently, researchers utilize TSP and its variance for solving problem in various areas such as last-mile delivery and logistics. For example, the integrating of drone and track with TSP and routing problem has been shown the significant improvement of the system performance (Jeong, Song, & Lee, 2019; Kitjacharoenchai et al., 2019).

Consider this study, a monitoring plants in a greenhouse task has a similarity to TSP. A mobile robot is required to move to assigned locations for inspecting conditions of plants as part of RQ_2. If we apply the TSP concept to solve plant monitoring tasks, we can obtain higher system performances, which are measured with system performance metrics.

2.4.2 Object Search by Robot

Object search is one of the major tasks for service robots (Miura, Kadekawa, Chikaarashi, & Sugiyama, 2016). Recently, a mobile robot is required to not only move to the assigned location, but it is also required to have the ability to detect and recognize objects at a specific location (Sjo, Galvez Lopez, Paul, Jensfelt, & Kragic, 2009). To find an object, the robot must move and examine various parts of the environment. An active search for an object by robot needs to combines tasks of localization, mapping, and motion control by maintaining effectiveness in the system (Makarenko, Williams, Bourgault, & Durrant-Whyte, 2002). Blind search is a common and typical search procedure for the unknown environment. The robot starts from a specific location and expands the search area from that location until it meets with an object. A system can be trained by some guidance or knowledge-based information to assist the direction of robot search or move (Sadhu, Abhiram, Chandan, Madhu, & Shreedarshan, 2013). By combing knowledge-based about the environment, search performance can be improved. Recently, research about a single robot and multiple robots search problems have been done in various fields (Kulich, Preučil, & Bront, 2014). For example, The development of a search algorithm by robot's manipulation utilizing a greedy algorithm has shown the significant improvement of search quality in some certain conditions (Dogar, Koval, Tallavajhula, & Srinivasa, 2014).

Moreover, if maps or prior knowledge of the environment are known, localizing and detecting the object can be done faster and more accurately, comparing with the unknown environment (Boussard & Miura, 2013). Also, for the cases that objects are required not to contact by a robot's arm, sensors typically integrate into the robot to serve this requirement.

There are three hierarchical search levels; local search, global search, and exploration, which serve to improve search performance from different perspectives (Sprute, Pörtner, Rasch, Battermann, & König, 2017). Learning and detection algorithms for multi-object tracking are developed and tested. The learning part of the algorithm has been proven to improve search performance by differentiating noise and target (Malagi & Rangarajan, 2016).

Research about object searches in robots has been explored in various contexts such as military, household, automotive industry, and the healthcare system. The concept of object search, however, is rarely used in an agricultural environment. It is a high potential to apply the concept of the blind search for an object to find stresses or diseases in plants as, in principle, they are close to each other. Also, a combination of search algorithms and knowledge-based information about directions of plant disease propagation can improve search performance and reflect the real characteristics of a plant. Moreover, the search algorithm needs to adapt itself according to the new information found to indicate the severity of an area.

2.4.3 Optimal Cost Balancing in Agriculture

Optimal cost balancing in agriculture is a practical and useful problem in the monitoring process. During the monitoring and inspection process, the decision to stop monitoring in the current location and move to another location strongly relates to monitoring cost and the benefit gains from continue inspecting or moving to the next location. The longer time the robot inspects in the area, the higher the confidence level at the location. On the other hand, spending too much time at one location might not be desirable as the available time of the system is limited.

The question described earlier is close to a classical problem in optimization, called Newsvendor Model (NVM). The NVM is utilized mainly on inventory management problems (Qin, Wang, Vakharia, Chen, & Seref, 2011). The generic setting of the problem is as follows. Consider a supplier, a vendor, and customers, at the beginning of the period, the vendor is interested in the stock policy (Q), which can yield the optimal profit. The demand of the product is a random variable ψ that has stochastic characteristics with the probability density function (PDF); $f(\psi)$ and cumulative density function (CDF); $F(\psi)$. The selling price of the product is fixed as P per unit. The vendor assumed to be unlimited capacity and zero lead time from the supplier. Lastly, inventory cannot be leftover across the period.

If the amount of stock Q is larger than the demand ψ, the remaining inventory $(Q - \psi)$ needed to be sold as salvage with price v. On the other hand, if the demand ψ is larger than the stock Q, thee unfulfilled demand $(\psi - Q)$ will cost u to the vendor. It can be assumed that $u > v$ for the obvious reason. The cost c of the product is assumed to be fixed. Therefore, the end of period profit for the vendor can be presented as in Equation (2.1).

$$\pi(Q,\psi) = \begin{cases} (P-\psi)Q - u(\psi - Q); if\ \psi \geq Q \\ P\psi + v(Q-\psi) - cQ;\ if\ \psi < Q \end{cases} \tag{2.1}$$

To maximize the total expected profit, it can be shown as follows.

$$\begin{aligned} \mathrm{E}[\pi(Q,\psi)] &= (P+u-c)\int_{Q}^{\infty} Qf(\psi)d\psi + u\int_{Q}^{\infty} \psi f(d)d\psi \\ &+(P-v)\int_{Q}^{\infty} \psi f(\psi)d\psi - (c-v)\int_{Q}^{\infty} Qf(\psi)d\psi \end{aligned} \tag{2.2}$$

Where $E[\cdot]$ represent the expectation function. Equation (2.2) can be shown as a concave function in Q (Silver, 1998). Therefore, the optimal value of Q (represent as optimal order Q^*) is as follow.

$$\begin{aligned} (Q^*) &= \frac{p+u-c}{p+u-v} \\ &= \frac{C_u}{C_o + C_u} \end{aligned} \tag{2.3}$$

Where C_u represent underage cost or cost from unsatisfied demand, which is equal to $p - c + u$. C_o is the overage cost or cost from having leftover inventory, which is equal to $c - v$.

NVM has been extended in many areas, such as in multi-product considerations, developing a pricing and discounting policy, and multi-period consideration (Khouja, 1999). Researchers have been investigated and improved by relaxing constraints so that NVM can have more practical implications. For example, when the demand information is incomplete, NVM can be used to provide the order quantity that minimizes the maximum regret of not acting optimally. The result is suited for the application, which requires robustness but not a conservative solution (Andersson, Jörnsten, Nonås, Sandal, & Ubøe, 2013; Perakis & Roels, 2008). NVM can also be utilized for the order quantity, which synchronizes vendor and manufacturer. The generalized NVM for the coordination problem provides the structure and connection between vendor and manufacturer so that the overall cost is minimized (Weng, 2004).

Even though NVM gives a promising suggestion of the inventory policy (Whitin, 1955), applications are mainly on the supply chain setting such as airline, and other perishable items

(Cachon, 2003). There is a scant work that applies the NVM in the agriculture setting. As capacity and time can also be considered as a perishable resource, utilizing them most effectively is essential in PA. Not only can the ARS spend time on the critical locations, but the overall production cost is also expected to be minimized.

2.4.4 Multi-agent System Optimization

The multi-agent system comprises of a set of agents that have the same or different capabilities and interact with each other by a protocol defining rules and procedures (Barbati, Bruno, & Genovese, 2012). The multi-agent system is studied extensively in engineering, economics, and social sciences (Billari, Fent, Prskawetz, & Scheffran, 2006). The multi-agent system is also utilized in the production system (Cowling, Ouelhadj, & Petrovic, 2003). The primary motivation for adopting a multi-agent system in the industry is the possibility of reducing production costs by having agents (e.g., robots) working faster and in parallel (Brogårdh, 2007). Cooperative work reduces processing time and thus improves the efficiency of operations. Instead of having single powerful, and complicated robots, a group of small yet simple robots, is easier to implement (Khamis, Hussein, & Elmogy, 2015) and yield superior performance (Nof et al., 2015). When cooperating machine sets share the same functionality, the overlap of machine capability introduces redundancy to the system. The benefit is especially significant at process bottlenecks, in which cases a single failure can disrupt the entire workflow.

Moreover, the overlap of machine capability through collaboration adds another dimension of flexibility due to the additional tasks that can be performed by the cooperating machine set (Rajan & Nof, 1996). When the parallel machines have different capabilities, the combination of skillset enables the team to accomplish tasks that cannot be performed by any single one. Another argument is based on the relative simplicity in design. Having simple robots can be simpler and cheaper to implement than having a single powerful, and complicated robot.

The multi-agent system aims to exploit each agent's capabilities to achieve system objectives either independently, or through collaboration. In such systems, tasks can be assigned either to individual robots or cooperative teams of robots with enhanced capabilities (Ceroni & Nof, 1999). The challenging problem in a multi-agent system is the Multi-Robot Task Allocation (MRTA) problem, especially when it comes to heterogeneous, unreliable robots equipped with different types of sensors and actuators (Khamis et al., 2015). Effective MRTA means forming teams under the task constraint and requirement in an optimal way. This problem can be seen as finding the task-to-agent assignment to achieve the overall system goals. The following challenges, therefore, need to be addressed when designing a solution to an MRTA problem (Parker, 1999): (1) How to assign a set of tasks to a set of robots? (2) How the robot teaming is coordinated efficiently and reliably? (3) How to make the robot teams adapt autonomously to dynamic changes in the environment?

To address and solve the challenges in multi-agent optimization systematically, workflow optimization protocol is utilized developed and used as a critical tool (Nof et al., 2015; Tkach, Edan, & Nof, 2017) with the combination of optimization techniques such as linear programming, mixed integer programming, and dynamic programming. Even though the multi-agent system has been utilized in various fields, it rarely applies to the relatively unstructured environment, such as agriculture. Taking the development of the multi-agent optimization into account, its benefits to the agricultural task is noticeable.

2.5 Summary of Research Gaps

From the literature survey, four main gaps are found. The first gap (G_1) comes from agent's communication. In a current monitoring system, the communication procedure between agricultural agents is not well established. It leads to the independent work among agents. Because of lack of communication, the system is inefficient and creates C&E during monitoring process.

The second gap (G_2) is the non-optimal interaction between agents. Resulting from G_1, agents do not well communicate; hence, it leads to a non-optimal interaction among them. The current agricultural robotics procedure emphasizes on effective movements of agents (robots and machinery). The interactions between agents are ignored by the current literature. Without an optimal interaction between agents, the system cannot provide the highest performance and contains a lot of inefficient actions.

Next, the third gap (G_3) deals with the information obtained during the monitoring process. The current approaches of agriculture robotic system, epically monitoring system, ignore to utilize real-time information. Because the current approaches mostly utilize pre-programmed agents, the approaches rarely make adjustment after obtaining new data during plant inspections. New data collected, however, may provide the insight of the specific locations and lead to the better solution of the system.

Lastly, the fourth gap (G_4) is the collaboration gap. Because most of the current approaches aim to utilizing a single agent (such as robot or machine) to perform a task, the collaboration has been ignored. Moreover, in the case of the multi-agent systems, each agent has its specific role which rarely collaborates to others. Additionally, the non-collaborative approach typically leads to higher cost comparing to collaborative approach as the system requires a specific agent for a specific task. Moreover, the collaborative approach usually requires weaker (less expensive) agents, but, because of the collaborative procedure, the complex tasks can be accomplished. Table 2.2 maps the research gap with literature section, RQs, and methodologies presented in the later sections. Moreover, Table 2.3 presents the sample of previous work and indicates drawback of each approach.

Table 2.2. Summary of the research gaps corresponding with the RQs and methodologies

Research Gaps	**Literature sections**	**RQs**	**Methodologies**
Communication	Section 2.1, 2.2, 2.3	RQ_1, RQ_2, RQ_3, RQ_4	Chapter 3: Collaborative Control Protocol for Cyber-Physical System (CCP-CPS)
Interaction	Section 2.1, 2.2, 2.3, 2.4.1, 2.4.2, 2.4.4	RQ_2, RQ_3, RQ_4	Chapter 4: Collaborative Control Protocol for Early Detection of Stress in Plants (CCP-ED)
Utilization new Information	Section 2.1, 2.4.3	RQ_2, RQ_3	Chapter 5: Optimal Inspection Profit for Precision Agriculture
Collaboration	Section 2.1, 2.2, 2.3, 2.4.4	RQ_1, RQ_2, RQ_4	Chapter 6: Multi-Agent System Optimization in Greenhouse for Treating Plants

Table 2.3. Recent research on agricultural robotics for crop monitoring (sample)

Agricultural Monitoring Purpose	**Focus of Approach**	**Research Gap Included**			
		G_1	**G_2**	**G_3**	**G_4**
Weed Control (Astrand & Baerveldt, 2002)	An autonomous mobile robot with vision systems works in the outdoor environment.	No	Partial	No	No
Maize crops (Diker & Bausch, 2003)	Using remote sensing to estimate in-season plant and fertilizer in the soil.	No	Yes	Partial	No
Agricultural fields (Nagasaka et al., 2004)	An autonomous watch-dog robot is equipped with a camcorder and GPS to record information about the monitored crops	Yes	No	No	No
Weed monitoring (Barbati et al., 2012)	A system that can plan and manage monitoring task in a field.	No	No	Yes	No
Cornfields (Xue et al., 2012)	A variable field-of-view machine vision method guides a robot to move between fields.	Yes	Yes	Partial	No
Field operations (Spekken & Bruin, 2013)	Minimize non-productive movements and maneuvering by agricultural machinery.	Yes	Yes	No	No
Agricultural Robot (Ishibashi et al., 2013)	Remote web-based monitoring system to order and control a robot in the field.	Yes	No	Yes	No
Greenhouse spraying (Min Hyuc et al., 2015)	Driving strategies for autonomous agricultural mobile robots.	No	No	Partial	No
Agricultural monitoring (Sai et al., 2016)	Algorithm to minimize sensor deployment nodes for intelligent monitoring.	Partial	No	Yes	No
Sugar beet fields (Chebrolu et al., 2017)	A robot carries a multi-spectral camera and an RGB-D sensor for recording, classifying and localizing plants.	Yes	Yes	No	No
Growth rate of crops (Liao et al., 2017)	An IoT/IoS-base system to monitor a growth rate of greenhouse crops	Yes	No	Yes	No
Greenhouse crops (This study)	Collaborative control protocol and algorithms to monitor and treat the stress of individual plants.	Yes	Yes	Yes	Yes

CHAPTER 3. COLLABORATIVE CONTROL PROTOCOL FOR CYBER-PHYSICAL SYSTEM (CCP-CPS)

The first part is the development of a protocol called Collaborative Control Protocol for Cyber-Physical System (CCP-CPS). The protocol is the backbone of the ARS, which has three main agents; humans, robots, and sensors. In this chapter, the design of the protocol is discussed. Also, the protocol is validated by the experiments to investigate the impacts of CPS on the agricultural system, especially the monitoring system. Note that the methodologies, protocols, and algorithms designed in the later chapter are the supporting part in enhancing the performance of CCP-CPS.

3.1 Collaborative Control Protocol for Cyber-Physical System (CCP-CPS) design

A CPS oriented framework and workflow is modified from Monitoring, detecting, and responding – cyber-physical system framework (MDR-CPS) by Guo, Dusadeerungsikul, & Nof (2018) and improved by Dusadeerungsikul, Nof & Bechar (2020). The main objective is to show the significance of the CPS integration greenhouse as, in PA, the greenhouse needs CPS to operate collaboratively, effectively communicate, and exchange information.

3.1.1 CPS framework for PA

Without CPS, a monitoring system cannot perform its full capacity. Figure 3.1 illustrates the CPS framework. Figure 3.1 and Figure 3.2, which derived from TAP, show the system has two layers, Task layer (T layer) and Resource layer (R layer). At any time $Time_t$, the two layers are connected and communicate by CPS. R layer is the agent or $Resource_n$ which needs to perform a given task $Task_m$ in the T layer. As agents in the R layer are distributed and decentralized, they require effective communication and information exchange to operate on the task. Arrows in Figure 3.2 represent the necessary communication between agents and tasks. Therefore, CPS, which is the middle node, is necessary to establish a reliable and capable connection between the T layer and R layer. In CPS node, it has four main parts Wireless Sensor Network (WSN), Cloud platform, Transmission mode, and Agricultural robots and other actuators. The following section will elaborate on each part in CPS for agricultural monitoring tasks.

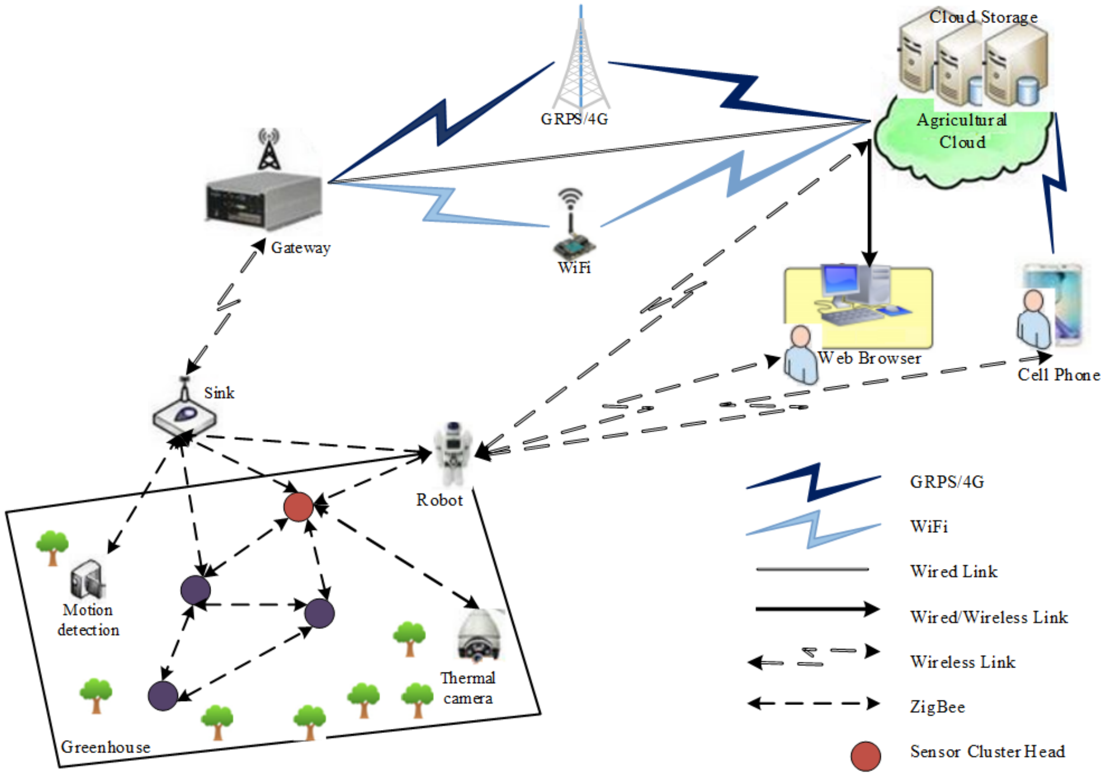

Figure 3.1. CPS framework for monitoring plants conditions in a greenhouse (Guo et al., 2018)

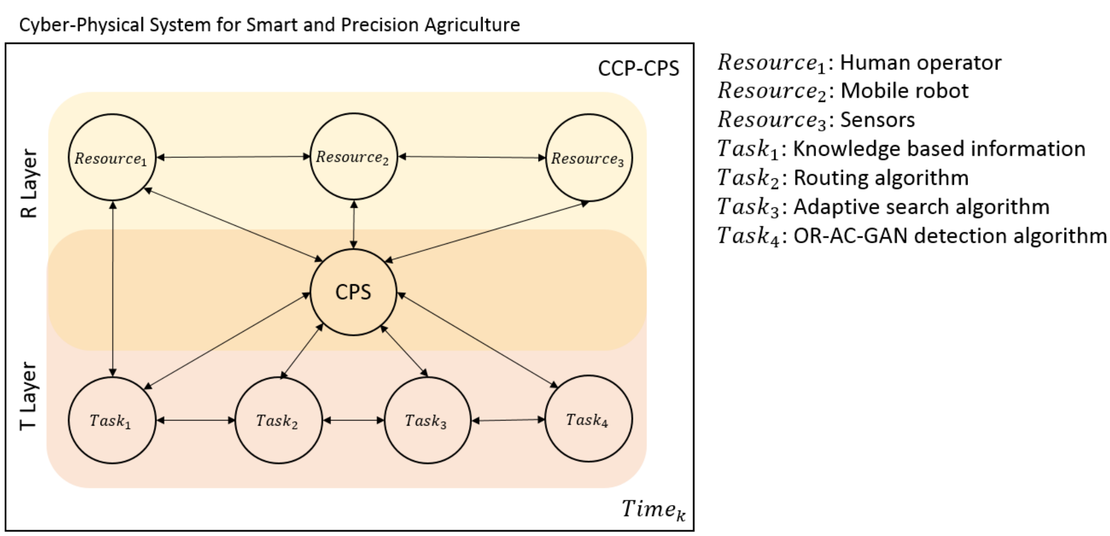

Figure 3.2. CPS Framework for greenhouse monitoring system

Wireless sensor network (WSN) deployment

Sensors are the essential components of the overall system for environmental monitoring and control. As the crop conditions inside greenhouses are moderated, the implementation of wireless sensor technologies is more accessible than in outdoor applications. The deployment of WSN refers to the deployment of sensor nodes located in greenhouses to provide information about environmental parameters that influence the development of the crops. Also, the WSN receives and sends the signal from the local sensors mounted on the robot. There are different kinds of heterogeneous agricultural sensors in model: monitoring the environmental parameters or abiotic stresses, such as humidity, temperature, pressure, CO_2 density, sunshine density, and water levels; monitoring diseases or static biotic stresses on crops, for instance, bacteria, fungi, viruses; monitoring mobile biotic stresses on crops, such as insects, rats, and other unexpected intruders; monitoring chemical insults, such as crops nutrients, soil PH value and pesticides influence on crops. The sensor types are shown in Table 3.1.

Table 3.1. Different sensors monitoring different parameters

Type	**Monitoring category**	**Monitoring parameters**	**Monitoring devices**
1	Environmental parameters	Temperature, humidity, CO_2 density, sunshine density, water levels	Stationary sensors
2	Soil parameters	Soil water density, soil pesticides, soil temperature, soil PH value, soil compaction	Stationary sensors
3	Crops growth parameters	Stress in a plant, Leaves' temperature, leaves' humidity, crony temperature, stem micro change, fruits swelling	Sensors installed on mobile robots
4	Crops diseases	Bacteria, fungi, viruses	Stationary sensors + HD Cameras +mobile robots
5	Intruders	Insects, rats, bugs, worms	

Parameters of type 1 and type 2 in Table 3.1 are simpler to monitor by stationary sensors or wireless sensor networks. Parameters of type 3 are usually monitored by sensors that are difficult to be fixed on leaves or stems where they need to be monitored. In our scenario, sensors for monitoring type 3 are installed on mobile robots, and robots navigate to the crops which are needed to be checked. Parameters of type 4 and 5 are usually to be sensed or captured by optical/thermal sensors or HD cameras; however, they are limited by computing ability and memory, photos transmitted may not be clear enough to be analyzed. Therefore, humans sometimes still need to dispatch a robot to obtain more information.

Cloud platform

The agricultural cloud platform in our model is used in the agricultural field based on a few server clusters. It contains two components, which are cloud storage and cloud computing/expert system, and not only stores a great deal of sensing data, but provides services, such as crop diseases analysis, intruders' alarm, and stresses identified. Agricultural cloud is constructed based on cloud computing theory and technology with the advantage of low cost, keeping and maintaining rich resources, and reducing the burden on farmers. According to the trained database, an expert system can determine whether there are particular stresses on crops or plants. For instance, if there is a disease on plant leaves, cameras take and transmit images to the cloud platform, after expert system processing, and an appropriate alarm is delivered to farmers. It is the prior and current art of machine learning and artificial intelligence. We do not focus on how to establish and train the expert system in this study. Here we focus on how the agricultural CPS can be controlled by collaborative workflow to monitor, detect, and respond to agricultural stresses.

Transmission mode

The network layer provides routing and data aggregation services. As shown in Figure 3.1, in the framework of CPS, sensors transmit data to sink nodes, cameras, and sink nodes connect to the gateway through wireless links. The gateway connects the agricultural cloud by GPRS/4G, Internet, WIFI, or local area networks. Human operators or farmers can access agricultural data through a web browser or smartphone, which enables them to resolve a problem in real-time. The detailed transmission technologies are shown in Table 3.2.

Table 3.2. Communication technologies applied in CCP-CPS

Connect methods	Application areas	Connection attribute
ZigBee	Sensors to sensors; sensors to sink nodes; robots to sensors or sink nodes	Wireless link
GPRS/4G, Internet, WIFI, LAN	Gateway to the agricultural cloud; agricultural cloud to the web browser (users) or smartphone; robots to a gateway	A wired or wireless link

Agricultural robots and other actuators

Sensors monitor the physical environment, and actuators activate physical processes. The terminal computation module contains basic executive rules of an actuator and has a small storage capacity of real-time data. For instance, if temperature, humidity, and solar radiation do not match the preset parameters, the interfaces between software and hardware trigger the corresponding hardware/actuation equipment to adjust automatically. Therefore, we pay much attention to agricultural robots in our model. The design of an agricultural robot is to aid detection in particular situations for particular stresses. Though sensors can do much of the monitoring work and can obtain pictures or photos, they are limited by power, fixed location, and transmission ability. For instance, suppose the agricultural expert system finds out there may be an abnormal situation, such as fungi on plant leaves, according to photos obtained by cameras. It is challenging to decide what kind of disease, and at what scale it happens, because the data transmitted by WSN are insufficient, or unclear. That is why farmers need a mobile robot to reach the nodes which convey ambiguous images in that scenario. When the robot arrives at the given location, cameras installed on the robot could take more pictures or record a video and transmit them to an agricultural expert system to carry out further analysis. That is a reasonable requirement for the robot because it can be designed as a mobile service with powerful computing ability, large memory, and sufficient electrical power. According to the descriptions above, we can determine the computational structure of an agricultural robot for CCP-CPS, as shown in Figure 3.3. The purpose of the robot computer is to run the necessary software for interfacing with the robot platform and sensors, sensor information processing, mission planning and execution, navigation, implementation control, user interface, network communication.

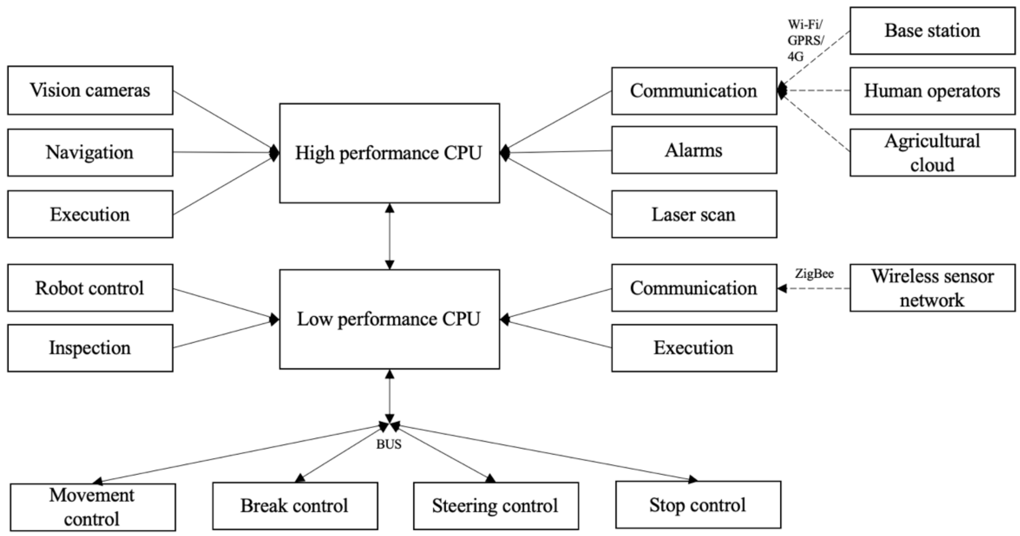

Figure 3.3. Computational structure for CCP-CPS agricultural robot

3.1.2 CCP-CPS design

The workflow for the CCP-CPS framework is shown in Figure 3.1. The details of the workflow can be described as following.

Step 1 CCP-CPS system initialization.

Step 2 Sensors have been triggered to monitor the greenhouse.

Step 3 Sensed data have been transmitted to the agricultural cloud.

Step 4 Data have been saved in the agricultural cloud storage.

Step 4 An agricultural expert system has analyzed

Step 6 If no stresses have been found, no action.

Step 7 If confirmed stresses have been detected, the corresponding mechanism will be triggered.

Step 8 If an agricultural expert system cannot confirm because photos are unclear or insufficient, then produce an informed alarm signal and deliver it to humans.

Step 8.1 Humans order a robot to navigate to a given area to collect more data upon receiving the alarm signal.

Step 8.2 The robot is launched to navigate when it receives an order from humans.

Step 8.3 The robot takes photos or records videos when it reaches the given crops in terms of the order.

Step 8.4 The robot transmits the data to the agricultural cloud when it finishes the collection.

Note: for **Step 4** and **Step 5**, the Agricultural expert system detects again according to the new data; if stresses can be confirmed, the procedure turns to **Step 7**, otherwise **Step 8**.

If human operators receive two consecutive alarm signals for a given plant or sensor packet, they can decide whether to order a robot to check the area or check by a farmer. The pseudo-code is shown below. The workflow is described by a chart with conversion conditions, as shown in Figure 3.4.

Protocol 3.1: CCP − CPS

```
CCP − CPS system initialization
Sensors have been triggered to monitor the greenhouse
Sensed data have been transmitted to the agricultural cloud
Data have been saved in the agricultural cloud storage
The agricultural expert system has analyzed data
IF No stresses have been found, DO
        No action
ELSE IF Confirmed stresses have been detected, DO
        The corresponding mechanism is triggered
ELSE IF Agricultural expert system cannot confirm, DO
        Alarm signal and deliver it to humans
        Humans order a robot to collect more data
        The robot is launched to navigate
        The robot takes photos or records video
        Robot transmits the data to the agricultural cloud
END IF
Protocol Terminated
```

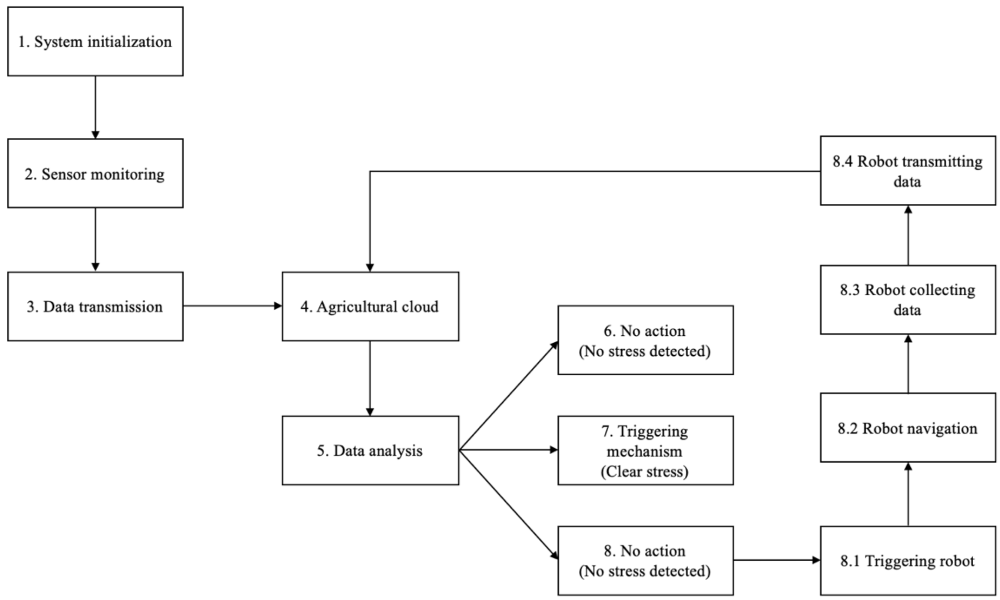

Figure 3.4. CCP-CPS relationship diagram

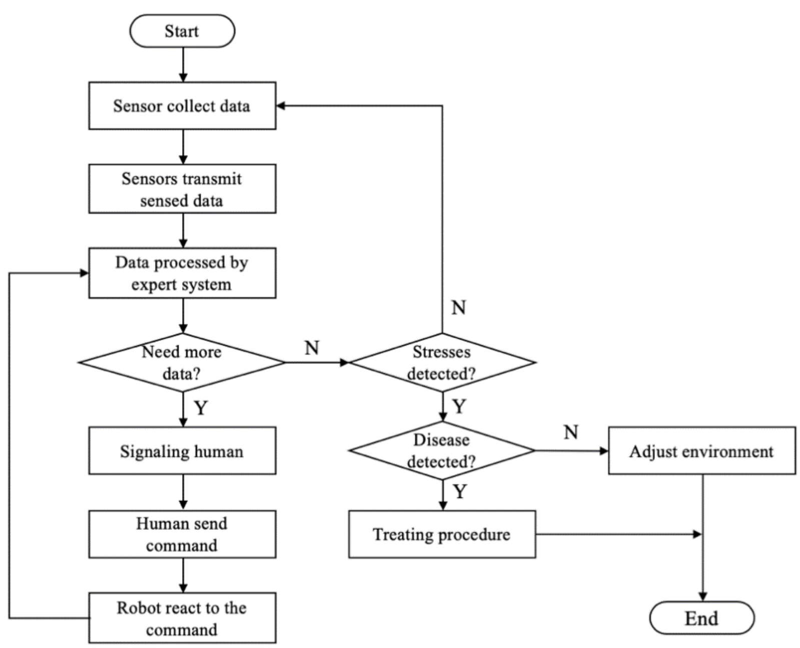

Figure 3.5. CCP-CPS workflow chart

3.1.3 Collaborative control theory in CCP-CPS

Collaborative architecture for CCP-CPS

To achieve the required tasks, collaborations are needed to utilize resources available in the system effectively. Collaborations will enable a more efficient system with fewer errors. In the model, collaboration can be categorized into five levels: collaborative sensing, collaborative data processing, collaborative communication, collaborative acting, and collaborative control (Figure 3.6).

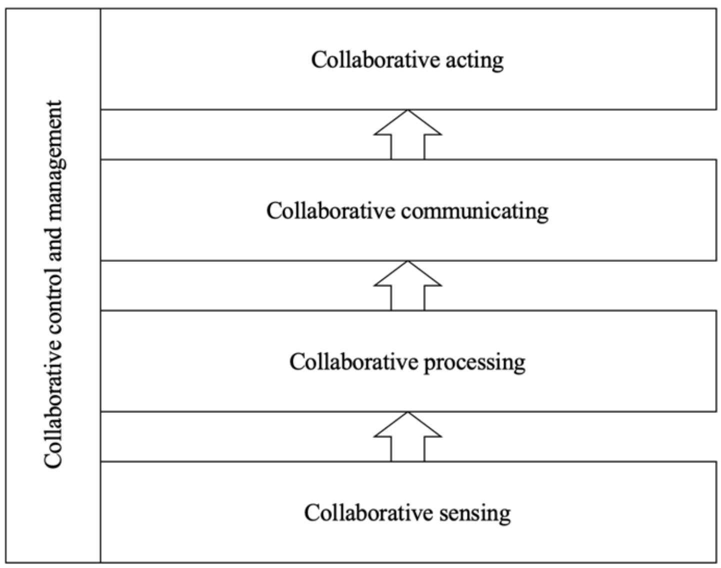

Figure 3.6. Collaborative architecture for CCP-CPS

Collaborative sensing happens with sensors measure sensing environment data and static plant physical attribute data. When data are measured, a mobile robot that is equipped with sensors needs to approach the target and sense attributes such as leaves' temperature, soil humidity, disease on stems. Collaborative processing means a mobile robot is equipped with a computer that can analyze data for application needing high-performance computing high-resolution image processing, video processing, and pattern recognition. This collaboration is critical when the operation areas are far from the base point or when results from processing are needed in real-time control to activate remedy actions. Collaborative communication is a significant part of CCP-CPS. In many situations, a mobile robot downloads data from the sensor, and at the same time, it receives data from sensors that are installed on it. All data are combined and transmitted to the expert system or human operation to diagnose disease of the plant on time. Collaborative acting requires human, a robot, and sensors to coordinate to perform monitoring, detecting, and responding effectively. The collaborative acting is critical for enabling real-time control for detecting stress in plants and achieving high-quality results. Collaborative control requires the different control mechanisms to work synchronously to achieve a specific task, effectively use resources, provide safe operations, and control the fault-tolerant mechanisms.

Collaboration requirement plan (CRP)

For the workflow and referring to the CRP principle in CCT, two stages of CRP, CRP-I, and CRP-II are deployed. In CRP-I, a detailed plan is generated, based on the work objectives and available resources. In CCP-CPS, CRP is utilized to assign and manage tasks and resources in CPS. Figure 3.2 presents the initial assigning tasks to resources.

In the second stage CRP-II plan execution & revision, Figure 3.7 shows an application of the CRP plan in agricultural CPS. CRP-II executes the plan generated by CRP-I real-time and revises the plan following spatial and temporal challenges, changes, and constraints. The purpose of CRP-II is to assign or reassign tasks to resources. CRP-II identifies all existing plans for each task, resolves the conflicts between the plans, and supports collaboration within a plan.

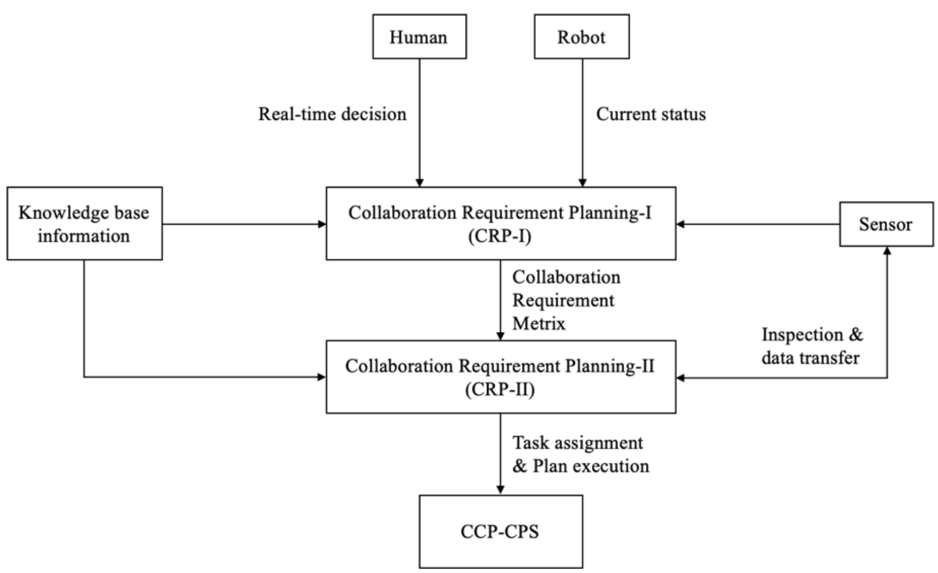

Figure 3.7. CRP architecture for agricultural CCP-CPS

Error prevention and conflict resolution (EPCR)

EPCR Principle deals with cyber-supported detection of errors and conflicts among collaborating agents, and the cost associated with resolving the detected errors and conflicts. Naturally, any system that cannot overcome its errors and conflicts effectively will get out of control and eventually collapse.

Refer to the definition of C&E, in the CCP-CPS model, the errors and conflicts may emerge at any device or cell. In this study, C&E by sensors, robots, and humans are mainly considered.

The following example illustrates an error and conflict produced by a robot. Suppose the robot navigates at a greenhouse at 10 a.m. according to a scheduled routine. During the process, it receives an order from a farmer to arrive at a given location where data delivered by a stationary sensor were found to be suspicious. Define a navigating task as $Task_1$, and a locating task as $Task_2$, and two agents are matching them, respectively, define a conflict for the robot at time *t* as follows.

$$\exists(Task_1(t) \cap Task_2(t)) \xrightarrow{agent\ performs > 1\ task} C\left[ARS_{Group}(t)\right]$$

When a conflict (C) occurred in the ARS at multiple agents at time t ($ARS_{Group}(t)$), first, there is a mechanism to detect the conflict; for instance, the robot can call an interrupt to report the conflict to the farmer or control center. The following steps are specified for the algorithm to resolve a conflict in this scenario, and the result presents in Figure 3.8. The robot will respond to the emergent event (new order) first, even during the process of pre-scheduled navigation.

The followings are the Resolve conflict in robot tasks algorithm together with the pseudo-code.

Step 1 Detection. A control module on the robot calls an interrupt to report the conflict to the farmer or control center.

Step 2 Identification. According to the definition of conflict, a control center identifies it is a conflict.

Step 3 Diagnostics. Control center determines the type, magnitude, time, and cause of the out-of-control status.

Step 4 Control center analyzes, predicts, and prevents propagation of the conflict.

Step 5 Conflict resolution. The control center resolves the conflict with a method that defines every event priority. For example, a mechanism is set to the event's priorities. Define regulations as:

- Regular events' priorities are lower than emergent events.
- Earlier emergent events' priorities are higher than later emergent events' priorities. (It can be changed with emergency priority code if needed.)
- A robot can execute only one event task (order) at time t.

Step6 Exception handing. Managing exceptions, i.e., constructive deviations from the process. In this example, the robot will respond to the emergent event (new order) first, even during the process of pre-scheduled navigation. Figure 3.8 shows the process.

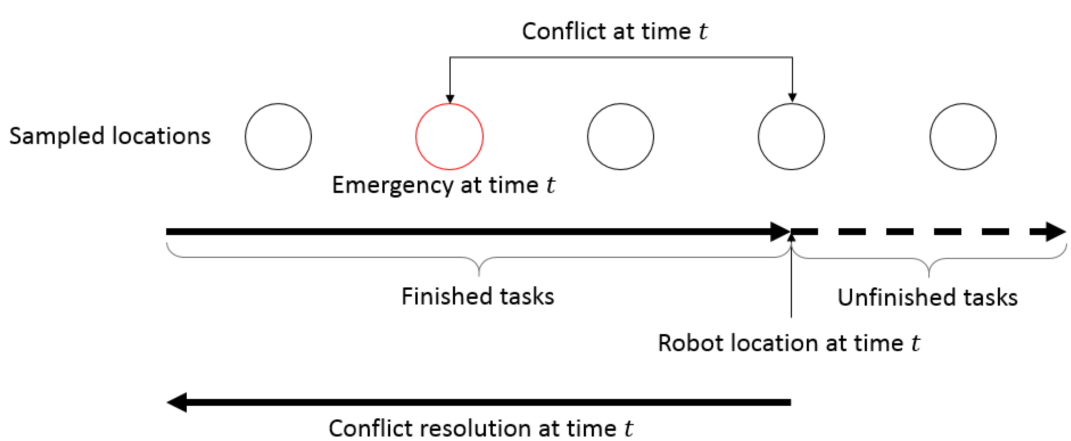

Figure 3.8. Robot conflict propagation and the resolution process (Guo et al., 2017)

The pseudo-code of the Resolve conflict in the robot tasks algorithm is presented below.

Algorithm 3.2: Resolve conflict in robot tasks

```
Detection
    A control module on the robot calls an interrupt
    Report the conflict to the farmer or control center
Identification
    Control center identifies a conflict
Diagnostics
    Determines type, magnitude, time and cause
    Analyzes, predicts, and prevents propagation of conflict
Define regulations
    Regular events' priorities are lower than emergent events
    Earlier emergent events' priorities are higher than later
        It can be changed with emergency priority code if needed
    Robot can execute only one event task (order) at time t
Exception handing
    Managing exceptions, i.e., process constructive deviations
Algorithm STOP
```

3.1.4 Monitoring alternatives

In order to understand the impact of CPS, two schemes for monitoring greenhouse crops are considered; 1) CPS scheme for monitoring greenhouse crops, 2) Non-CPS scheme for monitoring greenhouse crops. Because CPS that allows real-time control and communication among system agents, the first scheme can include humans, robot, and sensors which is the CCP-CPS design. As Non-CPS scheme has a limitation in connection, communication, and collaboration from human to robot and sensor agents, the system can only have a mobile robot equipped with multiple sensors and no human in the design. Next, each scheme will be described in detail.

CPS Scheme

In the CPS scheme, three main agents (humans, robots, and sensors) are collaborating as a system. Hyperspectral sensors are mounted on a mobile robot, and it reaches assigned locations to inspect plants. Figure 3.9 (left) presents the CPS scheme workflow of operation procedure and operation time of each step (in parentheses). As illustrates in Figure 3.9 (right), the monitoring process by the CPS scheme begins at a Base Point where human operators work, and a mobile robot starts. The robot travels which takes time ($Time_1$) to the assigned location. At the location, the inspection process is performed. $Time_2$, $Time_3$, $Time_4$, and $Time_5$ are used for localization, obtaining images, moving manipulator, and sensors operation, respectively. After parameters are measured, and signals are transmitted to human operators who take $Time_6$ to determine the quality of the data, the human operators may, in turn, assign the robot to re-measure the parameter if C&E happens during the operation or measure additional parameters such as additional images or temperature for a better decision or conclusion about the status of such plant. Otherwise, the robot moves to the next location, and the process repeats until the robot visits all locations.

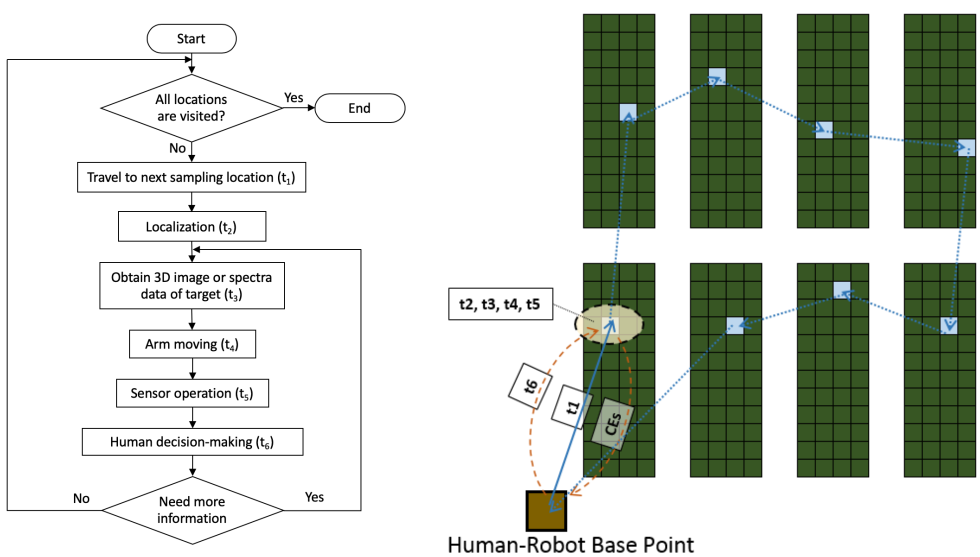

Figure 3.9. CPS Scheme

Non-CPS Scheme

In the Non-CPS scheme, the system consists of a mobile robot and multiple sensors that are mounted on the robot. Figure 3.10 (left) describes the Non-CPS scheme workflow and its operation time (in parentheses). Also, the pictorial of the scheme is illustrated in Figure 3.10 (right). Similar to the CPS scheme, the monitoring process starts at the Base Point. With the same conditions of the robot, it takes $Time_1$ to move to the assigned location. Robot and sensors require the same amount of time to perform the inspection process; $Time_2$, $Time_3$, $Time_4$, and $Time_5$. In this scheme, however, the real-time response about data quality is not possible because of no CPS facilitating; thus, no response from human operators. The robot will not transmit information to the Base Point, but it will move to the next location immediately. The process will reiterate until all locations are visited, and the robot moves back to the Based Point location. At Base Point, data will be transferred to a computer, and if additional information is needed, the robot will be dispatched again to obtain the missing data from the specific locations which will take time to re-operate as $Time_1'$, $Time_2'$, $Time_3'$, $Time_4'$, and $Time_5'$.

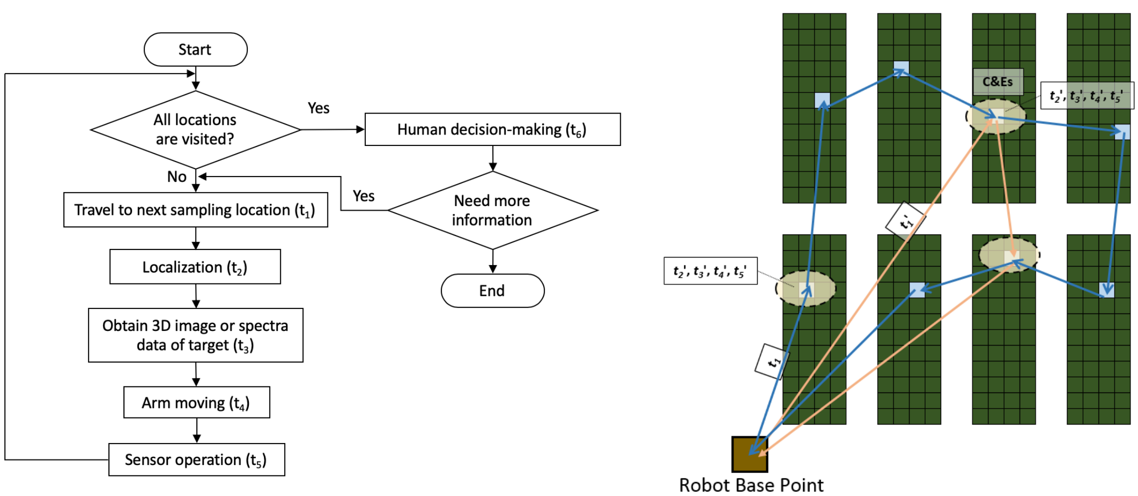

Figure 3.10. Non-CPS Scheme

3.2 Experiments and results

Three experiments by computer simulation have been conducted to evaluate different perspectives and performance (Sanchez & Wan, 2015) of the two monitoring frameworks; CPS scheme and Non-CPS scheme.

3.2.1 Experiment 3.1 – Total operation time

The first experiment aims to compare the total operation time of the two schemes. In the experiment, the same parameters (i.e., movement speed, inspection rate, and C&E rate), target locations, and maps are assigned to both schemes. The two schemes need to complete the assigned tasks, and the total operation time for each scheme are captured. The scheme which has lower total operation time is considered as a superior scheme.

Results and analysis

The results from the 100 computer simulation replications are presented in Table 3.3. While the results show similar outcomes from the two schemes in the average operation time (0.16% different), the standard deviations (SD) are significantly different (810.34% different). To compare the differences between both operation times, the t-test is conducted with the null hypothesis indicates the two schemes have the same operation time. Because the p-value is less than 0.005, the null hypothesis is rejected. Therefore, the CPS scheme has statistically lower in average operation time than the Non-CPS scheme at a 99.5% confidence level.

Table 3.3. Experiment 3.1 results

	CPS Scheme	**Non-CPS Scheme**
Average operation time (second)	873.20	874.61
SD of operation time (second)	0.29	2.64

3.2.2 Experiment 3.2 – Conflicts and error tolerance

The second experiment tests the two schemes with different probability of C&E. Because, in agriculture context, C&E can be expected, the scheme which has lower sensitivity to C&E or, in other words, has higher C&E toleration is the preferred scheme. By having C&E toleration, the total operation time will not increase, or increases increase with the lower rate. In the experiment, the two schemes which are assigned with the same map, tasks, and parameters are given a probability of C&E from 0% (C&E free, ideal system) to 90% with a 10% increment step. Lastly, the total operation time for both schemes is measured and analyzed.

Results and analysis

The results from the 100 computer simulation experiments are shown in Figure 3.11. The results show that, for both schemes, the total operation time is increased when the system has a higher probability of C&E. The average operation time of the CPS scheme, however, increases at a slower rate compares to the Non-CPS scheme. The T-test is conducted to compare the average operation time of both schemes at each C&E level. Because the p-value at each C&E level is less than 0.005, the CPS scheme has statically lower operation time than the Non-CPS scheme with a 99.5% confidence level.

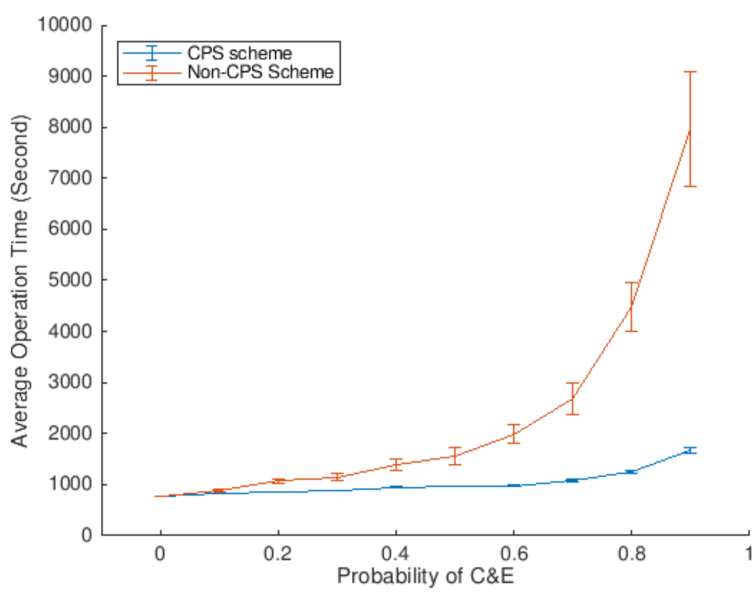

Figure 3.11. Experiment 3.2 results

3.2.3 Experiment 3.3 - Emergency response

The third experiment has an objective to capture emergency response performance. Because emergencies such as plant disease, unexpected plant conditions, and a new request from human operators can happen in a greenhouse, the scheme which can respond faster and more accurately would prevent yield loss and be considered as a better scheme.

In the experiment, human operators assume to know critical information of plants and diseases such as characteristics of infected plants, characteristics of plant diseases, and severity of diseases (Mahlein, 2016). Hence, in the CPS scheme, after an emergency, e.g., plant disease, happens, the robot will receive an emergency procedure from human operators or a knowledge-based expert system that the robot can respond and follow immediately. Non-CPS scheme, however, the robot does not have any specific procedure or direction from human operators or an expert system. Therefore, the robot needs to randomly inspect plants surrounding the infected location to find the propagation direction before responding to the emergency.

Results and analysis

The result from 100 computer simulation experiments is presented in Table 3.4. The average response time to an emergency and SD of the CPS scheme is significantly lower than the other. When CPS is applied, the robot can focus only on the direction of disease propagation provided by human operators or expert systems. On the other hand, without CPS, the robot needs to inspect plants one by one, which is less efficient and therefore requires a longer completion time. Moreover, with the assumptions about human operators' knowledge, not only time and cost of operation are reduced, the environmentally friendly crop production system is promoted as pesticides are minimizing to use (Gebbers & Adamchuk, 2010).

Table 3.4. Experiment 3.3 results

	CPS Scheme	**Non-CPS Scheme**
Average operation time (second)	1122.05	13061.56
SD of operation time (second)	60.65	1485.971

3.2.4 Conclusion and discussion

The study demonstrates a significant relative advantage of CPS in CCP-CPS. Because of the framework, communication, and collaboration among CCP-CPS agents, humans, a mobile robot, and sensors are possible and useful. The study demonstrates a framework, workflow, as well as application of CCT and CPS in an agricultural greenhouse setting. The framework includes humans, a mobile robot, and sensors. An agricultural CPS aims at collaborative monitoring, detection, and responses to stresses at identified locations. The workflow of the CPS environment is designed and tested in the study.

To validate the model for the integration and collaboration of CCP-CPS, computer simulation is utilized to perform the experiments. The results show that integrating a CPS environment with the CCP, a system can reduce the total operational time for monitoring and detection, thanks to faster and more effective communication among participating agents. Also, the system has relatively higher fault tolerance. With conflicts and errors occurring in a system, collaboration among agents in the system can endure them better and stabilize the continuity, hence the available operational time of the system. In other words, it can operate more harmoniously. Lastly, a CPS can enable better emergency response compared to the alternative system designs. Because collaboration among agents in the system is managed under the CPS environment, CCP's performance is effectively improved.

CHAPTER 4. COLLABORATIVE CONTROL PROTOCOL FOR EARLY DETECTION OF STRESS IN PLANTS (CCP-ED)

In this chapter, a protocol which can capture plant characteristics and coordinate among all collaborating participant in the system to work most effectively is developed and validated. Collaborative control protocol for early detection of stress in plants (CCP-ED) has three main algorithms: routing algorithm, adaptive search algorithm, and detection algorithm (Dusadeerungsikul & Nof, 2019; Dusadeerungsikul, Nof, & Bechar, 2018). Recent research about robotics in agriculture does not consider collaboration yet. Therefore, the current system cannot overcome some key issues such as C&E, efficiency, and production cost. CCP-ED has been developed to solve this issue by focusing on collaboration among agents in the system (humans, robots, and sensors). Besides, by considering C&E during experiments, it will reflect the real system condition. Lastly, the adaptive search algorithm is developed to reflect plant characteristics.

4.1 Task description

As discussed in the system scope assumption (Section 1.2.1), the ARS is to monitor the status of plants in the greenhouse. Humans, robots, and sensors are work as a team by having decision-makers (humans), a facilitator (robot), and inspectors (sensors). As a robot will move in a greenhouse and carry sensors that have detection agent, a robot tour which guide direction of a robot in a greenhouse environment is required to perform planning in an unstructured but predictable, knowledge-based environment.

After all, locations to visit is selected, a robot is guided from the Human-Robot Base Point (see Figure 4.1) to the locations by the robot routing algorithm. There, it will acquire sensor data about any stress among the sampled local plants. Data collected from each location can indicate the potential of the crops as either being under control (meaning, unstressed) or not. When a plant displays unusual stress, surrounding plants may already have the same problem.

Given scientifically established stress and disease behaviors for certain predictable diseases, the stress and disease will more likely spread in specific known directions. Such directional spread may be influenced by sunlight and other light sources, by airflow direction, and other causes.

Suppose, for example, that Northern and Western greenhouse directions tend to have relatively more stressed plants, given one stressed plant was found at a specific inspected location. Hence, the adaptive search algorithm needs to further check at the surrounding plants in those directions. The adaptive search algorithm cannot have information about the stress of a particular plant until it reaches each sampled location. The algorithm needs to be adaptive based on new information found during the operation period (possibly updated by remote experts). Figure 4.1 illustrates the situation described above.

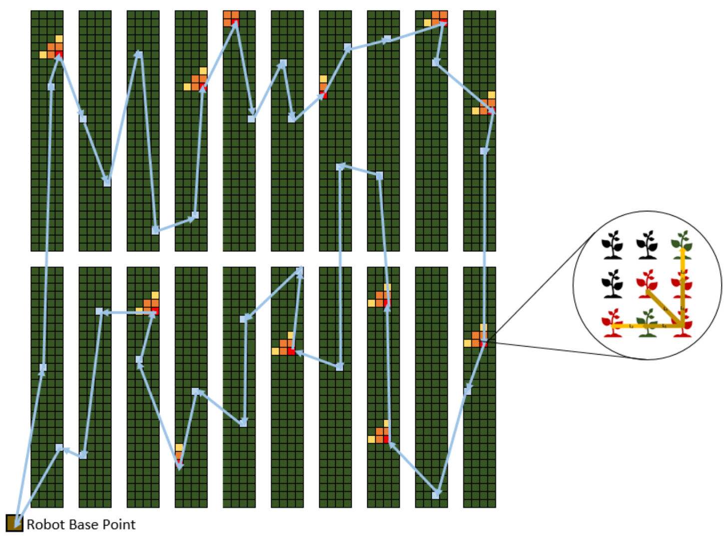

Figure 4.1. Agricultural robotic system operation; North and West propagation directions assumed as given by experts for this crop season

4.2 CCP-ED

The protocol is derived from the Collaborative Control Theory (CCT) principles. The system components are considered as agents that have the mutual goal of saving cost (time) while finding as early as possible the maximum number of stressed or already infected crops. The system agents need to collaborate intelligently to perform the task effectively. As all agents of the system are working together, this is “Mandatory Collaboration,” or collaborate as required.

The collaborative control protocol is designed to integrate agents in the ARS system to work seamlessly. The protocol design starts with creating Collaborative Requirement Planning

(CRP), which helps the designer to allocate tasks to agents. After the planning of tasks has completed, potential conflicts and errors in the system and its operations are discussed. Lastly, the step by step protocol which combines the routing algorithm and adaptive search algorithm is explained.

4.2.1 Collaboration Requirement Planning (CRP)

By applying the CRP concept to design the ARS system, we can obtain as follows.

CRP-I: Plan Generation

Referring to the CCT framework, the initial route can be mapped with CRP-I, which is the planning phase. In order to develop the CRP-I, establishing the requirements generating from the Collaboration Requirement Metrix (CRM) is necessary. CRM can be expressed as follow.

$$Avilable\left(Resource_j\right) \times Task_i \rightarrow CRM \tag{4.1}$$

$Avilable\left(Resource_j\right)$ denotes the set of $Resource_i$ available for each $Task_i$. $Task_i$ denotes as tasks and $Resource_n$ denotes available agent or group of agents. The matrix will generate CRM which contain $CRM\left(Task_i, Resource_j\right) = 0$ when resource $Resource_i$ is not available for task $Task_i$. and $CRM\left(Task_i, Resource_j\right) = 1$ when resource $Resource_j$ is available for task $Task_i$.

For this situation, at each location, the robot inspects several parameters to the defined task, and only one robot is required in this ARS. The system will have several types of tasks $Task_i$ in CCP-ED.

$Task_1$ = compute the routing

$Task_2$ = moving to the location

$Task_3$ = measure the parameter

$Task_4$ = error and conflict checking

$Task_5$ = stress status checking

$Task_6$ = decision making

For the agent or group of agents $Resource_i$, one can define the following team:

$Resource_1$ = robot

$Resource_2$ = sensors

$Resource_3$ = human

$Resource_4$ = team of robot and sensor

$Resource_5$ = team of robot and human

$Resource_6$ = team of sensor and human

$Resource_7$ = team of a robot, sensor, and human

Therefore, the CRM matrix, which has a group of agents in a row and task in a column, is derived as follows to define what collaboration modes are feasible.

$$CRM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

CRP-II: Plan Execution & Revision

CRP-II is the execution phase that obtains the plan from CRP-I. When more information is obtained during the process, the plan from CRP-I can be adjusted, which is the CRP-II role.

In this situation, the initial plan created for routing and inspecting tasks needs to be generated. The plan will define the route and parameters that are needed to be measured by sensors.

Therefore, the plan will define a route of a mobile robot, an order of locations to be visited, an order of information to be obtained at each location, and assignment of a sensor(s) to measure parameters. Also, the plan will be updated over time based on new information found during the monitoring process, and the sequence of the location will be updated.

4.2.2 Error Prevention and Conflict Resolution (EPCR)

In the ARS, there are potential conflicts and errors. CCP-ED will take potential conflicts and errors into account by having conflict and error rates in the protocol and during experiments. Having conflict and error rates, the real performance of the protocol can be analyzed.

Error prevention and conflict resolution (EPCR) principle will help to resolve conflicts and errors as early as possible. Errors occur when the input, output, or intermediate result of ARS does not meet specifications or expectations. Error is defined as follows.

$$\exists E\left[ARS_{Single}(t)\right], if\ S_{ARS_{Single}}(t) \xrightarrow{Dissatisfy} N_r(t) \tag{4.2}$$

Where E is an error, $ARS_{Single}(t)$ is ARS' single agent at time t, $S_{ARS_{Single}}(t)$ is the state of ARS agent at time t, and $N_r(t)$ is the set of constraints, r, at time t.

Moreover, conflict refers to the difference between the information, goals, plans, tasks, operations, or activities of the collaborating agents. Conflict is defined as follows.

$$\exists C\left[ARS_{Group}(t)\right], if\ S_{ARS_{Group}}(t) \xrightarrow{Dissatisfy} N_r(t) \tag{4.3}$$

Where C is conflict, $ARS_{group}(t)$is a group of ARS' agents at time t, $S_{ARS_{Group}}(t)$ is the state of the group of ARS agent at time t, and $N_r(t)$ is the set of constraints, r, at time t.

If dissatisfaction of conflicts or errors is found, conflicts or errors are detected and need to be solved. According to the definition of errors and conflicts, potential errors and conflict are described in Table 4.1.

Table 4.1. Major potential errors and conflicts examples in ARS planning and control

Type	Example	Collaborator(s)	State	Constraint
Error	Path error	Human	Human inputs wrong data and parameters	The objective of the routing
Error	Routing error	Robot	Robot cannot move according to routing plan	Robot's goal
Error	Measuring error	Sensor(s)	Sensor measures the wrong parameter	Sensor's goal
Conflict	Command conflict	Human and Robot	Human commands the robot to deviate from initial route	Human/operator objective
Conflict	Information conflict	Human and sensor	Human does not receive information on time	Sensor's objective/capacities
Conflict	Time measuring conflict	Robot and sensor	Robot moves to a new location, but sensor has not yet finished measuring	Robot's and sensor(s)' objectives/capacities
Conflict	Transition conflict	Human, robot, and sensor	A robot does not send information to human	Robot's capacity; Sensor's capacity
Conflict	Sensor conflict	Multiple sensors	Two sensors provide different results	Sensor's capacity
Conflict	Human conflict	Two or more humans	A decision from two humans are different	Human's capacity

4.2.3 Elements in CCP-ED

CCP-ED's objective is to utilize resources available to detect stress in greenhouse crops. The following sections describe CCP-ED in detail.

Decision variables:

$$x_{ij} = \begin{cases} 1 \text{ if } ARS \text{ traverse from } i \text{ to } j \\ 0 \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if } ARS \text{ search at location } i \\ 0 \text{ otherwise} \end{cases}$$

Parameters:

$c_{ij} = travel\ cost\ from\ i\ to\ j$

$p_j = processing\ time\ at\ node\ j$

$p'_j = extra\ processing\ time\ at\ node\ j$

$q_j = searching\ time\ at\ node\ j$

$q'_j = extra\ searching\ time\ at\ node\ j$

$\beta_j = probability\ to\ fail\ to\ inspect\ at\ node\ j$

$\alpha_j = probability\ to\ fail\ to\ search\ at\ node\ j$

$T = total\ avible\ time$

$T' = total\ processing\ time$

$D = Total\ number\ of\ infected\ locations\ found$

$d_i = number\ of\ infected\ plants\ found\ during\ operation\ at\ location\ i$

4.2.4 CCP-ED design

This section, CCP-ED step, is presented together with pseudo-code as below. The protocol which aims to detect stress in crops has three main algorithms; routing algorithm, adaptive search algorithm, and stress detection algorithm. The stress detection algorithm (Wang et al., 2019) is utilized and not be discussed in this study as the focus is on the ARS collaborative system. First, the protocol steps are presented as follows by pseudo-code for the protocol.

Collaborative control protocol for early detection of stress in plants (CCP-ED) steps

Step 1 Sample n nodes

Step 2 Create a tour for n nodes by routing algorithm

Step 3 If $c_{ij} < T - T'$, visit node j and $T' = T' + c_{ij}$

Else End algorithm

Step 4 If $p_j < T - T'$, inspection at j with probability of reinspection β_j and $T' = T' + p_j$

Else go to **Step 3**

Step 5 If the sensor finished the task go to **Step 6**

Else, if $p'_i < T - T'$, spend p'_j to finish the measuring task and $T' = T' + p'_j$

Step 6 Checking the quality of data obtained from the sensor, if good, go to **Step 7**.

Else, re-measure the data and $T' = T' + p_j + p'_j$

Step 7 If the status of node j is good or $q > T - T'$, then go to **Step 10**

Else, make decision y_i for search with a probability of re-searching α_j, $T' = T' + q_i$

$$y_i = \begin{cases} 1 \; searching \; for \; the \; surrounding \; area \; for \; the \; suspected \; plant \\ 0 \; otherwise \end{cases}$$

For searching the surrounding area, the ARS will use time q_i

Step 8 If the sensor finished the task go to **Step 9**

Else, if $q'_i < T - T'$, spend q'_j to finish the searching task and $T' = T' + q'_j$

Step 9 Checking the quality of data obtained from the sensor, if good, go to **Step 10**.

Else, re-measure the data and $T' = T' + q_j + q'_j$

Step 10 Update $D = D + d_j$, then do to **Step 3**.

The pseudo-code for CCP-ED is presented below.

Protocol 4.1: CCP − ED

```
CCP − CE initialization
Routing algorithm generates an initial tour
FOR i from 1 to #locations
        Detection Algorithm investigate location i
        IF Detection Algorithm found stress, DO
                Adaptive Search is activated
        END IF
ELSE FOR
Protocol Terminated
```

A routing algorithm for agriculture robotics algorithm

As the route of the robots can be represented as a TSP, traveling salesman problem can be formalized as follows.

Let

$$x_{ij} = \begin{cases} 1 \; if \; the \; ARS \; traverse \; from \; i \; to \; j \\ 0 \; otherwise \end{cases}$$

$c_{ij} = cost \; from \; i \; to \; j$

Objective function:

$$\min z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.4}$$

Subject to:

$$\sum_{i} x_{ij} = 1 \, ; for \; all \; j \tag{4.5}$$

$$\sum_{j} x_{ij} = 1 \, ; for \; all \; i \tag{4.6}$$

$$\sum_{i \in s} \sum_{j \in \bar{s}} x_{ij} \geq 2; for \; all \; i \; and \; j \tag{4.7}$$

$$x_{ij} \; binary \; for \; all \; i \; and \; j \tag{4.8}$$

The objective function (4) is to minimize the cost of traveling from *i* to *j*.

- If $x_{ij} = 1$, meaning that there is a tour from *i* to *j*. Therefore, in TSP, one would like to minimize the cost of travel from node *i* to *j* by the sum of a tour that has the smallest cost c_{ij}.

Constraint (5) is the constraint that forces all nodes has only one incoming arc.

- For all nodes, *j* will have only one arc from node *i*.

Constraint (6) is the constraint that forces all nodes has only one outgoing arc.

- For all nodes, *i* will have only one arc out to node *j*.

Constraint (7) is sub-tour elimination constraint.

Let S and $\bar{S}$ be the partition of the integer $i = (1, 2 \ldots n)$ so that $S \cap \bar{S} = \emptyset$ and $S \cup \bar{S} = i$. When we partition a group of nodes, as mentioned above, this constraint ensures that every partition has at least two arcs. It means at least one arc-in and one arc-out for each partition. Because of this constraint, one can ensure that the sub-tour is eliminated.

Figure 4.2 (from Bellmore and Nemhauser, 1968) shows how constraint (7) can eliminate sub-tours. Which (a) has a sub tour since there is no path from S to $\bar{S}$ but (b) and (c) do not have any sub-tour since there are two or more arcs from S to $\bar{S}$.

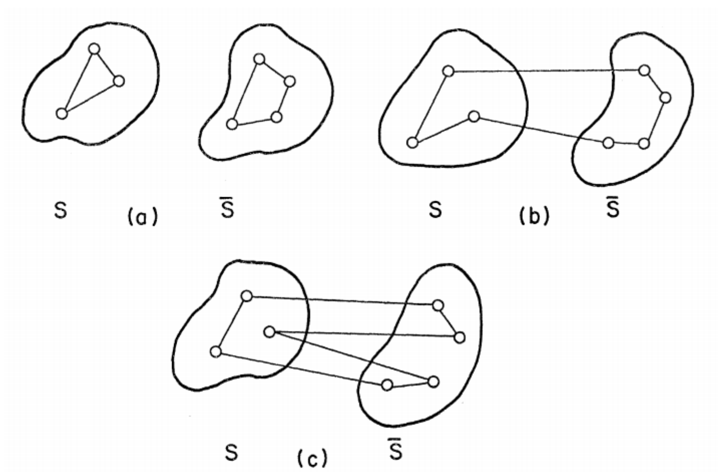

Figure 4.2. Sub-tour elimination

The mathematical model presented above is an NP-hard problem that cannot be solved in polynomial time. To guide a mobile robot to visit sampled locations, an effective routing algorithm that can create an optimal or near-optimal tour is needed. An effective routing algorithm can save traveling time for mobile robots and allow the ARS system to spend more time on finding infected plants. In this work, a genetic algorithm is applied to find a tour for a mobile robot. The algorithm steps and pseudo code are as follow.

Step 1 Generate initial population: 1,000 initial population (tour) are randomly generated by having the same probability of choosing each path. Each chromosome in this genetic algorithm is the list of locations that a mobile robot will visit. Also, all initial populations are feasible solutions with different fitness values. No initial tours are eliminated.

Step 2 Select parents: roulette wheel selection rule is used in the algorithm to select a parent based on fitness value (total distance). Roulette wheel selection rule will give more chances to select a chromosome with better fitness value. Only the selected chromosome will move to crossover (**Step 3**) and mutation (**Step 4**).

Step 3 Crossover: single-point crossover is implemented in the algorithm with 0.9 probability of successful crossover.

Step 4 Mutation: mutation is performed by switching randomly two locations (two genes) in each chromosome. The probability to successfully mutate is 0.9.

Step 5 Evaluation: all new offspring from the mutation step will be evaluated. Only the offspring, which has better fitness value (shorter distance), will replace the parent.

After performing all five steps, the process repeats for 300 iterations (stopping criterion). The result is a planned tour for a mobile robot to visit and monitor the stress conditions of inspected plants. The algorithm can be translated to pseudo-code as follows.

Algorithm 4.1: Routing algorithm for agriculture robotics

Initialize
Parameter
$n \leftarrow$ #locations $\times 10$
m
FOR i from 1 to n **DO**
    $route_i \leftarrow$ random generation
    $Fitted_i \leftarrow$ The total distance of each route
END FOR
FOR j from 1 to m **DO**
    Select m route base on the roulette wheel
    Single crossover
    Mutation
    FOR k from 1 to $m/2$ **DO**
        $Fitted_k \leftarrow$ The total distance of each route
    END FOR
END FOR
Terminate Algorithm

Adaptive search algorithm

The adaptive search algorithm (AS) is an essential part of the collaborative control protocol to indicate the severity of disease at plants in a greenhouse. Based on the behavior of a given plant, stress and disease usually propagate at scientifically predictable directions. Stress and disease will more likely spread at directions influenced by sunlight and airflow, as discussed earlier. Therefore, given this knowledge, we can construct an adaptive search algorithm, which can reflect the real characteristics of plant stress or disease propagation.

Suppose Northern and Western directions of the plant in the greenhouse are more likely to have similar stress symptoms; therefore, the adaptive search algorithm should further inspect plants in the given directions once the first infected plant is found in Figure 4.3.

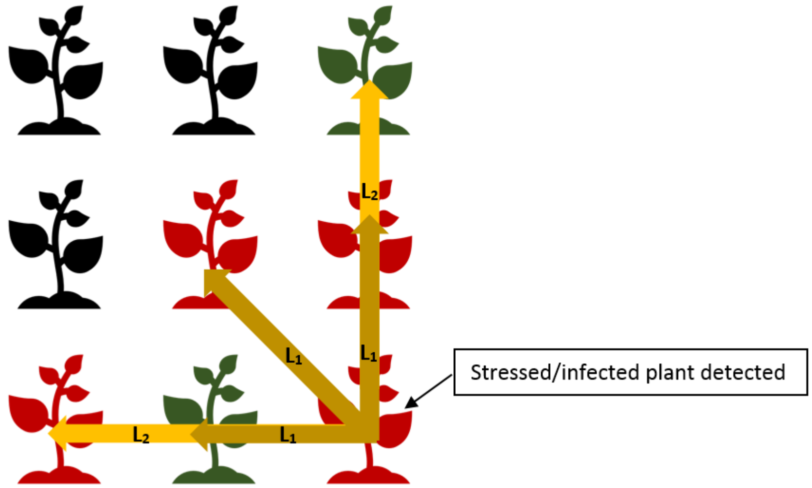

Figure 4.3. Adaptive search when propagation directions are known from previous research

Based on the information discussed above, the Adaptive search algorithm steps and pseudo code are as follows.

Step 1 Sensors inspect plant at the sampled location

Step 2 If the sampled plant has an abnormal condition (sign of stress, diseases), the adaptive search algorithm will be activated by starting to search at the first potential locations (L_1).

Step 3 After performing the first search operation, if more than half of the plants inspected are also infected, the second search operation (L_2) will be activated.

Step 4 All information about infected locations will be sent to the host.

By performing four steps of the adaptive search algorithm, farmers can know the magnitude and number of stressed or already infected plants, hence plan a precise, localized, and safe mitigation procedure. The adaptive search algorithm will be activated once sensors found the first signs of stress in the plant. This procedure will save time for searching in unlikely plants' locations. The algorithm can be translated to pseudo-code as follows.

	Algorithm 4.2: Adaptive Search Algorithm
1.	Initialize
2.	Parameter
3.	$i \leftarrow$ plant status
4.	$\boldsymbol{IF}\ i = stress\ \boldsymbol{DO}$
5.	The adaptive search algorithm is activated (L_1)
6.	$l \leftarrow$ #stress locations
7.	$\boldsymbol{IF}\ l >$ Treshold $\boldsymbol{DO}$
8.	The adaptive search algorithm is activated (L_2)
9.	$\boldsymbol{END\ IF}$
10.	$\boldsymbol{END\ IF}$
11.	Information transmitted
12.	Terminate Algorithm

4.3 Experiments

4.3.1 Experimental setting

The CCP-ED is applied to computer simulation in order to validate the effectiveness of the protocol. With the same operational resources (time, agent capacity, and the number of agents), the protocol is tested against alternative protocols. Table 4.2 shows the protocols used in the experiment.

Table 4.2. Alternative protocols design

Protocol design No.	Routing algorithm	Search algorithm	Inspection Algorithm
1 (CCP-ED)	Genetic Algorithm	Adaptive Search	OR-AC-GAN
2	Genetic Algorithm	Always Search	OR-AC-GAN
3	Genetic Algorithm	None	OR-AC-GAN
4	Random Routing	Adaptive Search	OR-AC-GAN
5	Random Routing	Always Search	OR-AC-GAN
6	Random Routing	None	OR-AC-GAN

4.3.2 Performance metrics

The following metrics are used for measuring the performance of the protocol in different aspects.

Number of Infected Plants Found (NIPF)

The protocol aims to find existing stressed or infected locations in a greenhouse correctly. The first metric is the total number of existing stressed/infected locations found.

$$NIPF = \sum_{i=1}^{n} d_i \tag{4.9}$$

NIPF = total number of infected plants found

d_i = number of infected locations found at location *i*

n = number of sampled locations in the greenhouse

Overall Robotic Effectiveness - ORE

Overall Robotic Effectiveness (ORE) measures the overall detection ability of the robotic system. The measurement comprises three main components: Utilization (U), Performance (P), and Success (S). Each of the components measures a different aspect of the system.

- Utilization (*U*) measures the proportion of uptime in the total available time for a robot in the ARS system.
- Performance (*P*) measures the time that a robot performs work (finding infected plants) during its uptime.
- Success (*S*) measures the percentage of successful operations that have been completed by the robot, meaning the proportion of infected plants found out of the total number of plants inspected by this robot.

$$ORE = U \times P \times S \tag{4.10}$$

Where

$$U = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} (p_i + q_i + c_{ij})}{T} \tag{4.11}$$

$$P = \frac{\sum_{i=1}^{N}(p_i + q_i)}{\sum_{j=1}^{M}\sum_{i=1}^{N}(p_i + q_i + c_{ij})} \tag{4.12}$$

$$S = \frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} s_i} \tag{4.13}$$

U = utilization

P = performance

S = success

T = total available time

p_i = inspection time at location i

q_i = searching time at location i

c_{ij} = travel time from i to j

d_i = number of infected plants found during operation at location i

s_i = number of inspected plants at location i

M, N = number of sampled locations in the greenhouse

The above metrics will be applied in the computer simulation experiments, which are described in the next chapter.

4.3.3 Experimental Results

From 100 computer simulation replication experiments, the results are presented in Figure 4.4 and Figure 4.3. The results show that CCP-CPS finds a significantly higher Number of Infected Plants Found (NIPF). The designed protocol can improve the NIPF by 73% compared with the baseline protocol (Protocol#6).

To test the difference in a mean of NIPF, ANOVA tests are conducted, and results are shown in Table 4.3. The ANOVA results show that at least one protocol design yields the difference in a mean of NIPF at 0.005 significant level. The more in-depth analysis of protocol performance is performed by conducting the Fisher Pairwise comparison tests. The result indicates that CCP-CPS yields the highest NIPF, which is the main objective of ARS, indicating stress locations in the greenhouse.

After that, Overall Robotic Efficiency (ORE), which derives from Utilization (U), Performance (P), and Success (S) of the system (Dusadeerungsikul & Nof, 2019) is calculated and analyzed to compare resource used. ANOVA test indicates that, based on p-value, which less than 0.005, at least one protocol has a difference in ORE. Then, the Fisher Pairwise comparison tests are performed, and results indicate the CCP-CPS provides the highest ORE value. Because of the highest ORE (ORE = 30.23%), the CCP-CPS can be considered as a protocol that can most effectively utilize resource (time) given to the system.

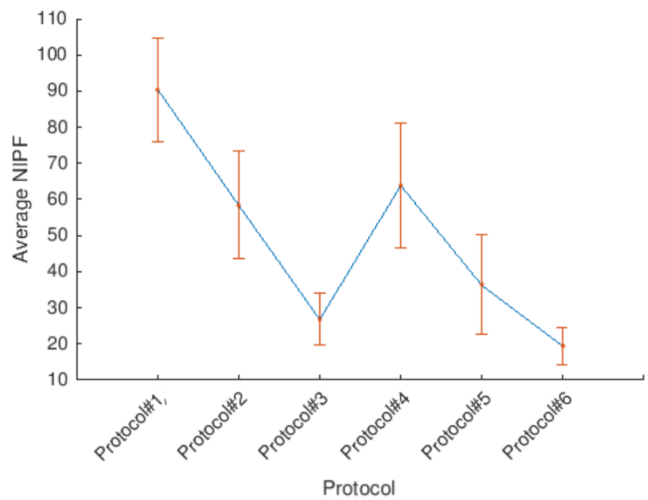

Figure 4.4. NIPF under each protocol design

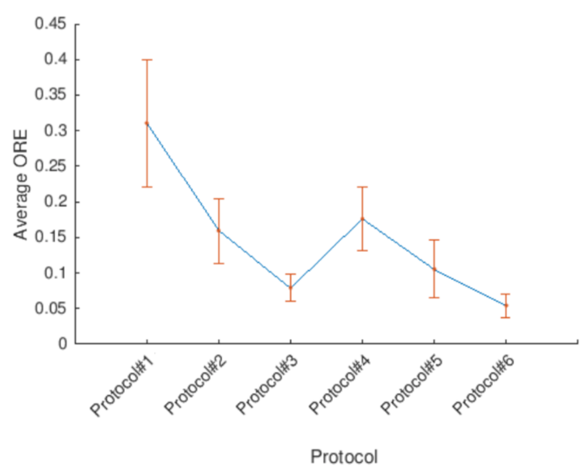

Figure 4.5. ORE under each protocol design

Table 4.3. ANOVA analysis

	ANOVA Analysis	P-value
1	The average of NIPF is the same among protocol designs	<0.005
2	The average of ORE is the same among protocol designs.	<0.005

*At significance level 0.05, all null hypotheses shown are rejected

Analysis of ORE

With 100 replications of simulation, ORE and its component, U, P and S, for each scenario are shown in Figure 4.6. The ORE of the developed protocol is higher than others which means the highest productivity of a monitoring system. Although protocol 3, 4, 5, and 6 have the highest utilization, the system stopped before visiting all nodes since it spent most of the time on an unnecessary task such as traveling or searching the area which has no potential.

Overall, the developed protocol outperforms others by detecting the more existing infected locations and utilizes most of resources available to detect infected locations. It visited all the assigned locations before it stopped since the utilization is less than 100%. The high performance indicates that, for the given time, the developed protocol uses time to perform inspection task (not for traveling). Lastly, the high successfulness means adaptive search algorithm can successfully capture character of disease in plant.

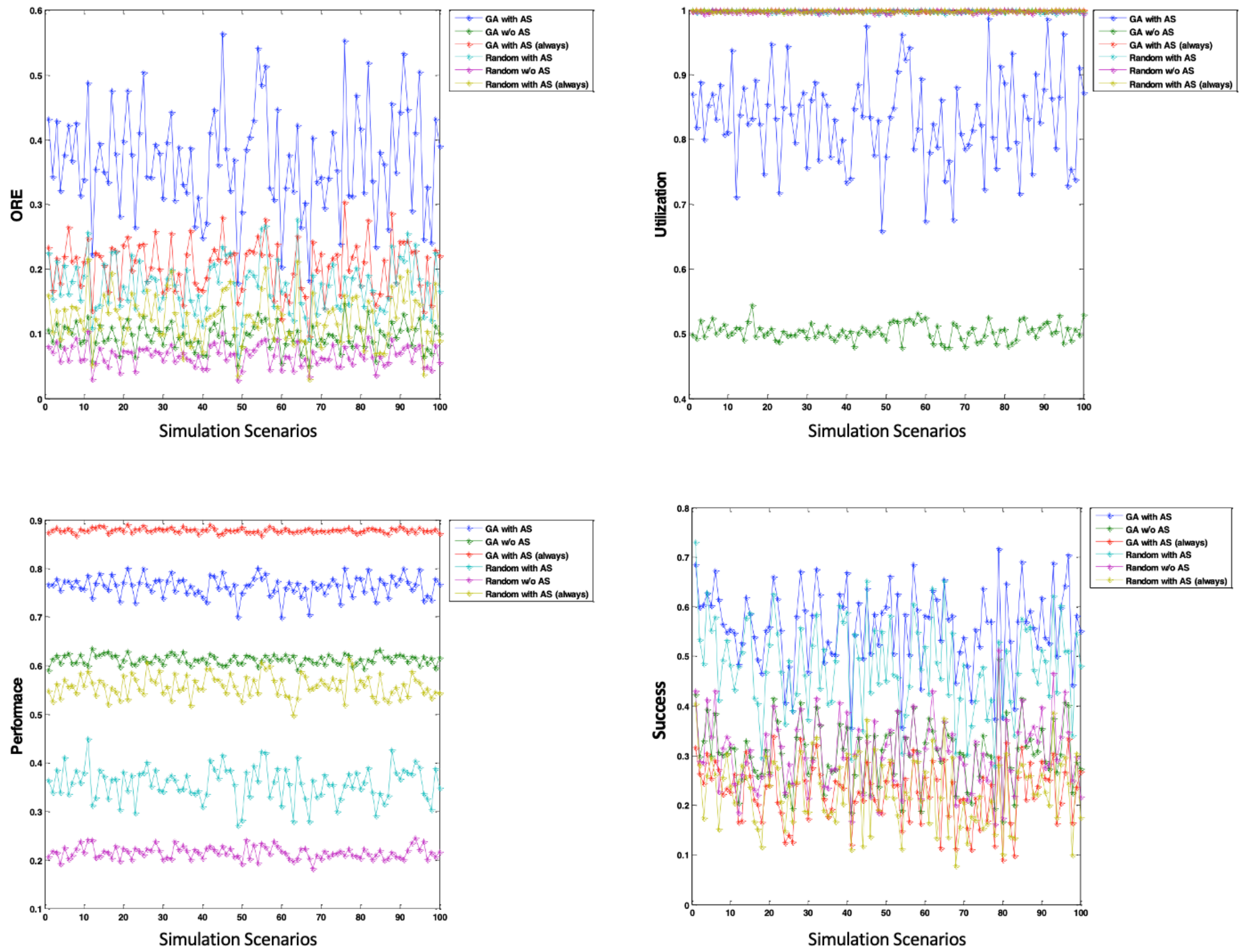

Figure 4.6. ORE and its components

4.3.4 Conclusion and Discussion

In this study, the CCP-ED is developed to enhanced CCP-CPS capability. The CCP-ED aims to monitor the condition of greenhouse crops by the system with humans, a mobile robot, and sensors, which are the local agent and does not involve in CCP-CPS. The CCP-ED is validated and compared with alternative protocols. CCP-ED utilizes CCT, TAP, and PCol to have a useful system under limited resources. CPS and IoT/IoS, which are mainly discussed, are the connection and communication elements of CCP-ED. The protocol is composed of three algorithms, namely, Routing algorithm, Adaptive search algorithm, and Stress detection algorithm.

Two metrics used for capturing monitoring and detection performance are NIPF and ORE. The NIPF presents the performance of the monitoring system, and ORE shows the effectiveness of the system.

Computer simulations are conducted to test the designed protocol, and results show its superior performance. The Routing algorithm, which can save travel time for the agent by 11%, allows the robot to perform more valuable tasks – inspection and monitoring tasks. Also, the Adaptive search, which connects to the disease propagation database and is activated after the Stress detection algorithm trigger enhances system performance by focusing on the locations which have a high potential of stressed/infected plants. Hence, the robot effectively utilizes the given time and improves mean NIPF by 73%.

The study has shown the significance of knowledge-based information and effective collaboration among system agents. The knowledge-based information helps the system to save cost and time, enhance the collaboration between system agents, and prepare for undesirable issues in agricultural fields, which are relatively unstructured. Machine learning, deep learning, and artificial intelligence techniques can be applied to improve CCP-ED. Moreover, CPS enables agents in the system to have effective communication, collaboration, decentralize decision making, distributed control, and real-time response are required for productive smart and precision agriculture. The second case study will demonstrate the importance of CPS in smart and precision agriculture.

CHAPTER 5. OPTIMAL INSPECTION PROFIT FOR PRECISION AGRICULTURE

This chapter aims to enhance the capability of AS. The previous research has shown that AS has a promising ability to improve the performance of the monitoring system. Therefore, it is worthwhile to investigate the algorithm more in-depth to find the optimal policy of AS. The optimal AS is expected to take the dynamic of a system, such as errors that can be occurred and the difference in disease characteristics into account. Therefore, a search policy, call Dynamic Adaptive Search (D-AS), is developed and tested in this chapter to be integrated into the CCP-ED as an optimal search policy.

In this chapter, first, two types of error in the monitoring system which impact the performance of AS are discussed. Because of the two types of error, the system contains uncertainties that lead to additional cost of the system, overage cost, and underage cost. The proof of the optimal adaptive search balancing overage and underage cost is presented and discussed to deal with system errors. Experiments are conducted to investigate the optimal policy and, lastly, the sensitivity analysis has been conducted.

5.1 Type of system errors in AS

As discussed earlier, in the agricultural environment, there are potential errors in the plant monitoring system. With the null hypothesis of the stress detection algorithm is a plant that does not have stress (H_0: $Plant\ is\ healthy$), there are two types of errors in the system; Type 1 Error (α), and Type 2 Error (β). Figure 5.1 shows the relationships and conditions of the errors in the system.

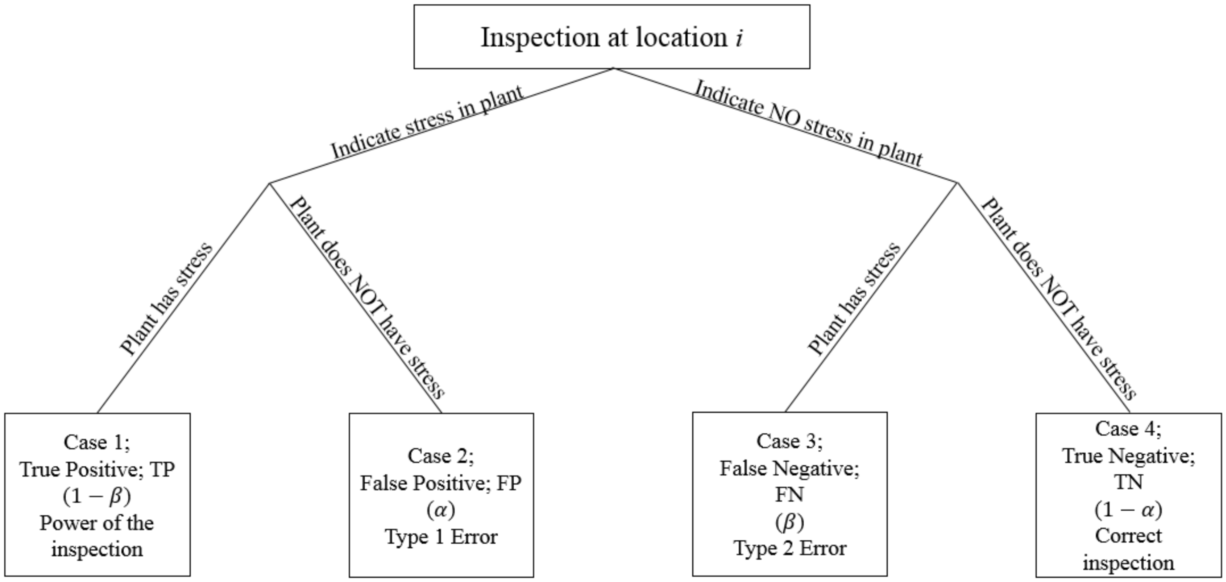

Figure 5.1. Type of errors in a monitoring system

Type 1 error (α) or False Positive (FP) happens when the stress detection algorithm indicates stress in the plant when in truth is none. Without error, the stress detection algorithm must indicate True Negative (TN). Therefore, the probability of making a type 1 error in the monitoring system is presented in Equation (5.1).

$$\alpha = P(R|S) \tag{5.1}$$

Where

$$R = Rejection\ region\ (Reject\ the\ null\ hypothesis)$$
$$S' = Plant\ does\ not\ have\ stress$$

Type 2 error (β) or False Negative (FN) is an error when the stress detection algorithm does not indicate stress in a plant when the truth is that the plant has stress. Without error, the stress detection algorithm must indicate True Positive (TP). So, the probability of making a type 2 error in the monitoring system is shown in Equation (5.2).

$$\beta = 1 - P(R|S) \tag{5.2}$$

Where

$$R = Rejection\ region\ (Reject\ the\ null\ hypothesis)$$

$$S = Plant\ have\ stress$$

Moreover, the power of the monitoring system, which is the ability to detect when it has stress in a plant is presented in Equation (5.3).

$$Power\ of\ monitoring\ system = 1 - \beta = P(R|S) \tag{5.3}$$

In PA, plants are necessary to inspect to ensure the healthiness or prepare treatment if needed. The system errors in the monitoring process, however, will lead to misinterpretation of information. For example, a plant can be seen as unhealthy and needs further inspection for obtaining the impact of the diseases or treatment, while, in fact, the plant is healthy. In this case, time and resources are spent too much than it should be, which increases the overall operation cost. Therefore, the system that can deal with the errors by balancing between the risk of error and cost of an inspection is needed to minimize the operation cost.

5.2 Monitoring Profit

In the monitoring process, the system will gain system profit (or loss) during the operation. Monitoring profit defines as a benefit from gaining new information to the system. The monitoring profit from finding stress in crops is denoted as P_S and the profit from finding healthy plant denoted as P_H. Because plant that has stress needs the treatment to prevent yield loss, the system will gain more profit and information when indicating plant with stress than healthy plant; then it is reasonable to assume that $P_S > P_H$.

Also, there will be an inspection cost (C_I) incurred in every step of inspection. The system better to save operation costs (meaning not to operate) if the plant is healthy, but if the plant has stress, the system needs to inspect and indicate such location. Therefore, from this analysis, it is safe to assume that $P_S > C_I > P_H$.

Moreover, when the system has errors (either Type 1 or Type 2), there will be an additional cost associated with them. In the next section, errors are quantified to operation costs to capture the impact of each error type.

5.2.1 Overage Cost

The Overage cost (C_o) happens when the search algorithm spends extra time inspecting in the location, which is already sure to have the disease.

Lemma 5.1: Type 1 error incurs the Overage cost (C_o) and the amount of C_o is equal to $\alpha \times C_H$ where C_H denotes of cost of inspecting the healthy plant, $C_I - P_H$.

Proof: At each location that plant does not have diseases, cost of expanding the search is equal to.

$$C_o = P(A|\, S') \times C_H \tag{5.4}$$

Where

$$A = Adaptive\ search\ is\ activated$$

$$S' = Plant\ does\ not\ have\ stress$$

$$C_H = Cost\ of\ inspecting\ healthy\ plant$$

By definition,

the $P(activate\ search \mid plant\ does\ not\ have\ disease)$ is a Type 1 error.

$$C_o = \alpha \times C_H \tag{5.5}$$

■

5.2.2 Underage Cost

The underage cost (C_u) occurs when the search algorithm does not spend enough time to inspect the location in which information about the disease is not sufficiently clear.

Lemma 5.2: Type 2 error incurs the Underage cost (C_u) and the amount of C_u is equal to $\beta \times C_S$ where C_S denotes the cost of not inspecting the infected plant, $P_S - C_I$.

Proof: At each location that plants have diseases, the cost of not expanding the search is equal to.

$$C_u = P(A' \mid S) \times C_S \tag{5.6}$$

Where

$$A' = Adaptive\ search\ is\ notactivated$$

$$S = Plant\ has\ stress$$

$$C_S = Cost\ of\ not\ inspecting\ infected\ plant$$

By definition,

the $P(do\ not\ activate\ search \mid plant\ has\ disease)$ is Type 2 error.

$$C_u = \beta \times C_S \tag{5.7}$$

■

5.3 Optimal Balancing

In order to optimize system profit by balancing C_o and C_u. **Theorem 5.1** is proposed as follows.

Theorem 5.1 Optimal Expansion of Dynamic Adaptive Search: Assuming a cost of inspection healthy plant and cost of not inspect infected plant are equal, the optimal expansion for an infected location is when the search algorithm inspects at location m such that the cumulative distribution function (CDF) is greater than or equal to critical ration (R_0).

$$R_0 = P(M \leq m^*) = F(m^*) \geq \frac{\beta}{\beta + \alpha} \tag{5.8}$$

Proof:

At the location M, the algorithm should expand search to m location while

$$P(M \leq m)C_o < P(M > m)C_u$$

From Lemma 1 and Lemma 2,

$$P(M \leq m) \times \alpha \times C_H < P(M > m) \times \beta \times C_S$$

From assumption, $C_H = C_S$

$$P(M \leq m) \times \alpha < \mathrm{P}(M > m) \times \beta$$

The algorithm should stop expanding search at m^* which is the optimal location when

$$P(M \leq m^*) \times \alpha \geq P(M > m^*) \times \beta$$

$$P(M \leq m^*) \times \alpha \geq (1 - P(M \leq m^*)) \times \beta$$

$$F(m^*) = P(M \leq m^*) \geq \frac{\beta}{\alpha + \beta} \tag{5.9}$$

■

Therefore, from **Theorem 5.1**, the optimal expansion to maximize profit gain is when the search algorithm expands the search to the first m such that the CDF, $F(m^*)$, is less than or equal to critical ration, R_0.

5.3.1 Dynamic Adaptive Search Algorithm

Develop from **Theorem 5.1**, Dynamic Adaptive Search Algorithm (D-AS) is present as follows.

Algorithm 5.1: Dynamic Adaptive Search

Initialize
Parameter
Calculate Critical Ratio
FOR all location m_i in monitoring plan ***DO***
    Inspect m_i
    $i \leftarrow 0$
    IF m_i is infected ***DO***
        FOR $R < R_0$ ***DO***
            Inspect m_{i+1}
            $R = R + P(i+1)$
            $i = i + 1$
        END FOR
    ELSE
        WHILE $R < 1 - R_0$ ***DO***
            Inspect m_{N_i}
            $R = R + P(i+1)$
            $i = i + 1$
        END WHILE
    END IF
END FOR
Terminate Algorithm

5.3.2 Propagation Probability Map Model

As the D-AS requires CDF of the infected plant, $P(X < m^*)$, to determine the optimal search progression. In this section, the theory and algorithm of how to model and determine the probability of each location are presented.

Lemma 5.3: Because of environmental conditions, the plant may generate diseases on its own. Once the plant is infected, the disease can propagate to the nearby locations in some specific directions based on the type of disease, sunlight, seasons of the year, and airflow. Moreover, the disease has a maximum distance to propagate.

Theorem 5.2 Diseases Propagate Probability: The probability of location n is infected from location m_0 if m_0 is infected is $P(n\ is\ infected)$.

Where

$$P(n\ is\ infected) = \begin{cases} \dfrac{pq^{M-1}(1-q^n)}{1-q} & ; m_M \leq n, m_M > m_0 \\ 0;\ otherwise & \end{cases} \tag{5.10}$$

Proof: Based on **Lemma 5.3** and Figure 5.2, the disease can be modeled as follow.

Greenhouse Location	Probability that m_n is infected form m_0
m_M	p^{M-1}
...	...
m_n	p^{n-1}
...	...
m_3	p^3
m_2	p^2
m_1	p
m_0	1

Propagation direction

Figure 5.2. Disease propagation

Let p denotes the probability that the plant develops diseases. q is the probability to propagate in a specific direction. The disease can propagate at most M locations (to m_M) from the original location m_0. Therefore, the probability of a plant $m_M \neq m_0$ is infected by the first location m_0 is as follows.

$$P(n\ is\ infected) = \frac{pq^{n-1}}{\sum_{i=0}^{M-1} pq^i}$$

Consider $\sum_{i=0}^{M-1} pq^i$

$$\sum_{i=0}^{M} pq^i = pq^0 + pq^1 + pq^2 + \cdots + pq^{M-1}$$

$$= p(1 + q^1 + q^2 + \cdots + q^{M-1})$$

Let

$$\omega = 1 + q^1 + q^2 + \cdots + q^{M-1}$$

$$q\omega = q + q^2 + q^3 + \cdots + q^M$$

$$(1 - q)\omega = 1 - q^M$$

$$\omega = \frac{1 - q^M}{1 - q}$$

Therefore,

$$P(n\ is\ infected) = \begin{cases} \dfrac{pq^{n-1}(1-q)}{1-q^M}; m_M \le n, m_M > m_0 \\ 0;\ otherwise \end{cases}$$

■

Algorithm 5.2: To generate CDF at each location in a greenhouse map, the **Algorithm 5.2** can be utilized.

	Algorithm 5.2: Probability Map
1.	Initialize
2.	Parameter
3.	$n \leftarrow 0$
4.	$\boldsymbol{WHILE}$ Initial location $n \le$ maximum distance M $\boldsymbol{DO}$
5.	Calculate the probability of m_n according to Equation (4)
6.	$\boldsymbol{END\ WHILE}$

5.4 Experiments, Results, and Analysis

In this section, numerical experiments are conducted using computer simulation. The proposed algorithm, D-AS, is compared against the alternatives in terms of cost and system performance. Lastly, the sensitivity analysis is conducted to analyze the robustness of the algorithm.

5.4.1 Experimental Setting

The experiment involves 100 plants in the greenhouse for the initial inspection, which are sampled randomly. Table 5.1summarizes the details of simulation parameters.

Table 5.1. Summary of parameter settings

Parameter	Value
Number of simulations runs	100
Type 1 error	$\alpha = 10\%$
Type 2 error	$\beta = 30\%$
The maximum distance of disease to propagate	$m_M = 5\ locations$
Inspection profit of inspecting and found stress	$P_S = 5\ cost\ unit/location$
Inspection profit of inspecting and not found stress	$P_H = 1\ cost\ unit/location$
Inspection cost	$C_I = 3\ cost\ unit/location$

5.4.2 Search policies alternatives

The proposed algorithm, D-AS, is validated against other alternative policies – Static Adaptive Search, Always Search, and None Search. The first two search procedure is categorized as an adaptive search, while the last two options are non-adaptive search. Each of the AS policy is described in detail as follow.

Dynamic Adaptive Search

The Dynamic Adaptive Search (D-AS) is the proposed method and described earlier in the methodology section. The algorithm considers C&E in a system that can adjust search progression according to new information found. The D-AS has described in **Algorithm 5.1**. Moreover, **Algorithm 5.2** is required for calculating CDF at each location. Figure 5.3 represents **Algorithm 5.1**, **Algorithm 5.2**, and the linkage.

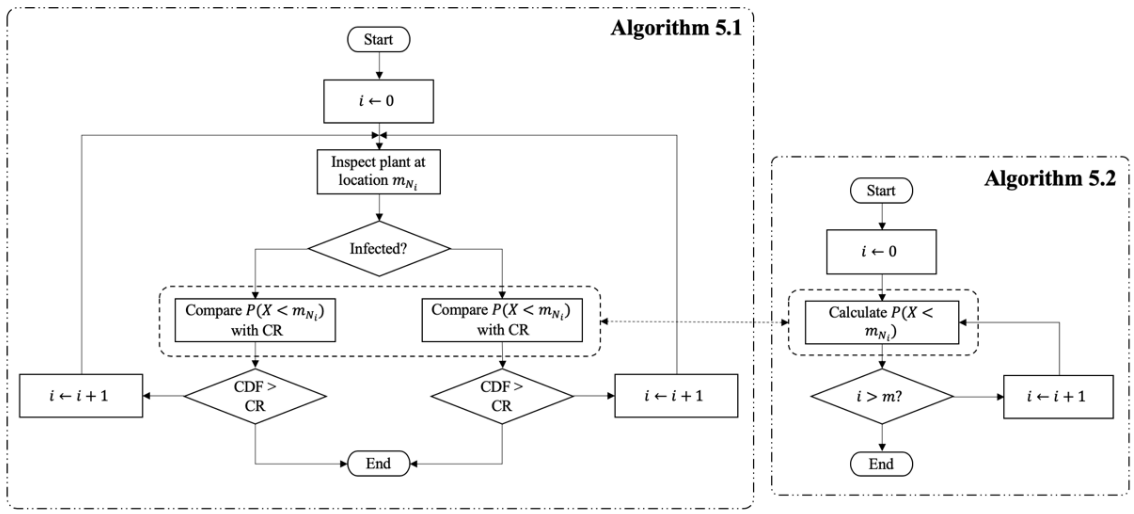

Figure 5.3. Dynamic Search Algorithm

Static Adaptive Search

The Static Adaptive Search (S-AS) is modified from (Dusadeerungsikul & Nof, 2019). The S-AS has linked to knowledge-based information, which helps the algorithm to emphasize the specific direction as the knowledge-based information contains the disease propagation direction. Even though the algorithm can adapt based on new information found, the progression of how further in search is predetermined. The algorithm is present as in **Algorithm 5.3**.

Algorithm 5. 3: Statics Adaptive Search

Initialize
Parameter
FOR all location m_i in monitoring plan ***DO***
Inspect m_i
    IF m_i is infected ***DO***
        Inspect m_{i+1} *and* m_{i+2}
        Calculate infected ratio R_I
        IF $R_I > \tau$ ***DO***
            Inspect m_{i+3} *and* m_{i+4}
        END IF
    END IF
END FOR

Always Search

The Always Search algorithm represents the system without real-time information. The system agent cannot determine whether the inspected location is infected or not. The system, however, has scientific knowledge about the disease propagation direction. Therefore, the system will always search in such a direction to gain the most information.

	Algorithm 5.4: Always Search
1.	Initialize
2.	Parameter
3.	***FOR*** all location m in monitoring plan ***DO***
4.	Inspect $m_i, m_{i+1}, m_{i+2}, m_{i+3}, m_{i+4}$
5.	***END FOR***

None Search

The None Search algorithm represents the current practice of farmers. As the current practice does not utilize the knowledge-based, the worker will inspect only the assigned location according to the monitoring plan. Algorithm 5 presents the None Search algorithm.

	Algorithm 5.5: None Search
1.	Initialize
2.	Parameter
3.	***FOR*** all location m in monitoring plan ***DO***
4.	Inspect m_1
5.	***END FOR***

5.4.3 Computational Experiment and Results

System Performance Analysis

In this section, the system performance will be analyzed by using a computer simulation experience. The matrices that are utilized to capture system performance are True Positive Ratio (TPR) and True Negative Ratio (TNR).

True Positive Ratio

True Positive Ratio (TPR) indicates how many TP found from the total positive location indicated (both TP and FP). The objective of the system is to indicate TP locations; therefore, the system that has higher TPR is considered as a better system. TPR can be calculated as present in Equation (5.11).

$$TPR = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FP_i} \quad (5.11)$$

Where

$$N = Number\ of\ initial\ inspected\ locations$$
$$TP_i = True\ Positive\ found\ from\ the\ initial\ location\ i$$
$$FP_i = False\ Positive\ \ found\ from\ the\ initial\ location\ i$$

True Negative Ratio

True Negative Ratio (TNR) represents the ratio between TN and total negative location indicated (both TN and FN). The system that has higher TNR is the preferred system because the system can indicate the true healthy plant and not false, indicating the healthy plant. TNR can be calculated as present in Equation (5.12).

$$TNR = \frac{\sum_{i=1}^{N} TN_i}{\sum_{i=1}^{N} TN_i + FN_i} \quad (5.12)$$

Where

$$N = Number\ of\ initial\ inspected\ locations$$
$$TN_i = True\ Negative\ found\ from\ the\ initial\ location\ i$$
$$FN_i = False\ Negative\ found\ from\ the\ initial\ location\ i$$

Experimental Results

Table 5.2 shows results from experiments. In general, the D-AS outperforms (or as good as) other policies in terms of TPR. On the other hand, TNR results are very similar among different policies. The ANOVA test and Post-hoc Bonferroni test are performed. Results show that, in the low diseases which have a low probability of developing or propagating, such as p or q equal to 30%, the D-AS performs None Search policy, and they outperform the other two policy with 95% confidence level. On the other hand, when the diseases developed or spared quickly, the D-AS performs as good as an S-AS with a 95% confidence level. The Always Search policy, which is the most conservative approach, the TPR is lower than other policies. On the other hand, in terms of TNR, the Always Search outperforms other policies with a 95% confidence level.

In conclusion, as the D-AS can adapt to the different input, it modifies the optimal search policy at each situation in terms of TPR. For example, if the input indicates the low propagation rate (q equal to 30%), the D-AS has changed itself to the None Search policy as the results are statistically the same. The D-AS, however, expands search further if the propagation rate is high.

Table 5.2. Performance analysis

$\boldsymbol{p}$ and $\boldsymbol{q}$		**D-AS**		**S-AS**		**Always Search**		**None Search**	
		$\boldsymbol{TPR}$	$\boldsymbol{TNR}$	$\boldsymbol{TPR}$	$\boldsymbol{TNR}$	$\boldsymbol{TPR}$	$\boldsymbol{TNR}$	$\boldsymbol{TPR}$	$\boldsymbol{TNR}$
$p = 30\%$	Avg	0.557	0.947	0.434	0.962	0.216	0.987	0.558	0.939
$q = 30\%$	Std	0.070	0.030	0.064	0.020	0.036	0.006	0.066	0.034
$p = 30\%$	Avg	0.538	0.944	0.487	0.955	0.270	0.983	0.562	0.943
$q = 60\%$	Std	0.083	0.027	0.088	0.022	0.070	0.010	0.070	0.033
$p = 30\%$	Avg	0.546	0.943	0.542	0.943	0.343	0.974	0.560	0.943
$q = 90\%$	Std	0.110	0.031	0.113	0.030	0.124	0.016	0.075	0.033
$p = 60\%$	Avg	0.625	0.915	0.562	0.939	0.353	0.973	0.624	0.911
$q = 30\%$	Std	0.132	0.068	0.107	0.030	0.111	0.015	0.130	0.070
$p = 60\%$	Avg	0.645	0.910	0.596	0.927	0.390	0.968	0.662	0.893
$q = 60\%$	Std	0.123	0.052	0.120	0.039	0.124	0.018	0.141	0.079
$p = 60\%$	Avg	0.640	0.904	0.638	0.903	0.448	0.953	0.689	0.881
$q = 90\%$	Std	0.143	0.066	0.144	0.067	0.174	0.040	0.143	0.081
$p = 90\%$	Avg	0.730	0.820	0.649	0.901	0.457	0.953	0.728	0.818
$q = 30\%$	Std	0.165	0.179	0.136	0.064	0.163	0.037	0.164	0.178
$p = 90\%$	Avg	0.728	0.857	0.672	0.889	0.485	0.947	0.758	0.768
$q = 60\%$	Std	0.147	0.091	0.142	0.069	0.170	0.039	0.173	0.216
$p = 90\%$	Avg	0.705	0.853	0.702	0.852	0.530	0.921	0.781	0.730
$q = 90\%$	Std	0.159	0.129	0.159	0.126	0.205	0.083	0.176	0.234

Cost Analysis

In this section, the cost to operate D-AS is compared with the alternative AS by using Monitoring Profit. The computer simulation experience is used for analyzing each algorithm.

Monitoring Profit

Monitoring Profit (MP) shows profit by the information gain from expanding search with respect to the operation cost. At each location, the new information about the status of the plant will lead to the benefits gain of the system. It, however, also has a cost of the inspection, such as location as well. The MP captures the difference between benefit and cost. Hence, the system which has higher MP is preferred. The calculation of MP is presented in Equation (5.13).

$$MP = \sum_{i=1}^{N} (P_S \times TP_i + P_H \times TN_i) - NC_I \tag{5.13}$$

Where

$$P_S = Benefit\ gain\ if\ inspection\ locaiton\ has\ stress$$
$$P_H = Benefit\ gain\ if\ inspection\ locaiton\ does\ not\ has\ stress$$
$$N = Number\ of\ inspection$$
$$C_I = Inspection\ cost$$

Note that

$$P_S > C_I > P_H$$

Experimental Results

Table 5.3 presents the results of the experiments. The D-AS indicates the best performance among the search policy. ANOVA test and Post-hoc Bonferroni test are performed to investigate the statistical difference among policies. Results show that, with a 95% confidence level, in most of the case, the D-AS is statistically higher MP than other policies.

Only the case that p and q are low (30%), the D-AS, and None Search have the same MP. The D-AS yields the same result as None Search because, as mentioned before, in the low propagation rate case, D-AS is performed as the None Search. It is intuitively as the diseases hardly propagate to the next location. The algorithm should not expand to the nearby location even though the first location is found infected as the cost of expanding might exceed the expected information gain.

On the other hand, the Always Search policy has the lowest MP. The reason is that the policy investigates further locations no matter what the information obtained. It makes the system has too much cost and yields the lowest MP.

Table 5.3. Cost analysis

$\boldsymbol{p}$ and $\boldsymbol{q}$		**D-AS**	**S-AS**	**Always Search**	**None Search**
		$\boldsymbol{MP}$	$\boldsymbol{MP}$	$\boldsymbol{MP}$	$\boldsymbol{MP}$
$p = 30\%$	Avg	22.120	-31.941	-326.521	22.120
$q = 30\%$	Std	4.298	5.157	5.786	4.298
$p = 30\%$	Avg	21.530	-2.500	-275.802	21.421
$q = 60\%$	Std	5.449	6.517	8.160	4.254
$p = 30\%$	Avg	37.580	37.533	-180.507	22.401
$q = 90\%$	Std	8.507	8.479	12.376	4.307
$p = 60\%$	Avg	51.867	51.065	-174.790	51.864
$q = 30\%$	Std	7.367	8.233	11.573	7.367
$p = 60\%$	Avg	91.250	83.232	-130.288	68.928
$q = 60\%$	Std	8.697	9.522	12.339	7.746
$p = 60\%$	Avg	132.043	131.060	-29.307	80.467
$q = 90\%$	Std	11.783	11.735	16.375	7.817
$p = 90\%$	Avg	144.446	105.966	-23.417	105.966
$q = 30\%$	Std	11.508	9.188	15.778	9.188
$p = 90\%$	Avg	176.685	171.125	20.670	125.315
$q = 60\%$	Std	12.292	11.143	16.169	9.707
$p = 90\%$	Avg	228.291	227.758	126.102	140.076
$q = 90\%$	Std	14.298	14.306	19.695	9.913

Sensitivity Analysis

This section aims to analyze the sensitivity and robustness of the proposed algorithm. The D-AS is compared with the alternative options when the conditions deviate from the assumptions in Table 5.1. Robustness is an essential characteristic of an algorithm which operates in the agricultural environment because the environment is less structured and has high fluctuations. The algorithm has higher robustness (or less sensitive) to the change in the parameters considered to be a better option. In the analysis, four parameters, p, q, α and β, have varies to see changes in the metrics. The sensitivity is defined as follows.

$$Sensitivity = \frac{Y(x)}{Y(x^*)} \tag{5.14}$$

Where

$$Y(x) = Result\ from\ assumed\ parameters$$
$$Y(x^*) = Result\ from\ actual\ parameters\ (Optimal\ result)$$

Experimental Results

From the simulation experiment, sensitivity results are presented in Table 5.4, Table 5.5, Table 5.6, and Table 5.7. Table 5.4 presents the sensitivities of different search policies when p deviate from the known parameters while Table 5.5 shows the sensitivities when q is different. Table 5.6 and Table 5.7 demonstrate the sensitivities when α and β have deviated from the setting.

Although the deviations of parameters are range from 10% to 25%, D-AS provides results that no more than 10% from the optimal setting. In other words, the D-AS is robust and gives near-optimal results at the situation that input has uncertainty.

Other algorithms are less robust compared to the D-AS. Always Search is the most sensitive algorithm, especially when p and q are lower than the known value.

Table 5.4. Sensitivity analysis 1

%Deviation of p		D-AS			S-AS			Always Search			None Search		
		TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP
+10%	Avg	1.018	0.988	1.002	1.016	0.992	1.074	1.031	0.995	1.414	1.021	0.978	1.076
	SD	0.007	0.024	0.009	0.016	0.027	0.017	0.014	0.031	0.021	0.000	0.018	0.005
+25%	Avg	1.027	0.980	1.044	1.025	0.986	1.124	1.051	0.992	1.963	1.032	0.962	1.129
	SD	0.034	0.052	0.036	0.045	0.059	0.047	0.043	0.066	0.051	0.027	0.041	0.028
-10%	Avg	1.046	0.971	1.019	1.045	0.979	1.214	1.084	0.988	3.878	1.055	0.946	1.216
	SD	0.018	0.073	0.032	0.029	0.083	0.050	0.045	0.094	0.065	0.010	0.060	0.025
-25%	Avg	1.070	0.962	1.109	1.069	0.973	1.324	1.120	0.985	6.764	1.081	0.932	1.320
	SD	0.020	0.090	0.019	0.001	0.103	0.045	0.039	0.119	0.076	0.024	0.078	0.015

Table 5.5. Sensitivity analysis 2

%Deviation of q		D-AS			S-AS			Always Search			None Search		
		TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP
+10%	Avg	1.009	0.995	1.043	1.008	0.996	1.045	1.015	0.998	0.549	1.013	0.991	1.052
	SD	0.012	0.022	0.017	0.017	0.024	0.020	0.021	0.026	0.023	0.009	0.021	0.015
+25%	Avg	1.015	0.990	1.076	1.011	0.994	1.071	1.020	0.997	0.438	1.023	0.982	1.102
	SD	0.034	0.044	0.039	0.042	0.050	0.047	0.049	0.053	0.050	0.025	0.041	0.031
-10%	Avg	1.024	0.986	1.019	1.021	0.991	1.124	1.038	0.995	0.320	1.031	0.975	1.148
	SD	0.043	0.064	0.054	0.052	0.071	0.062	0.063	0.075	0.068	0.039	0.060	0.048
-25%	Avg	1.034	0.982	1.069	1.032	0.987	1.191	1.059	0.993	0.248	1.040	0.968	1.195
	SD	0.049	0.084	0.066	0.055	0.089	0.071	0.067	0.097	0.082	0.053	0.081	0.064

Table 5.6. Sensitivity analysis 3

%Deviation of α		D-AS			S-AS			Always Search			None Search		
		TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP
+10%	Avg	1.006	0.997	1.032	1.006	0.998	1.036	1.011	0.999	0.896	1.008	0.995	1.037
	SD	0.011	0.018	0.015	0.012	0.020	0.016	0.015	0.021	0.018	0.011	0.018	0.015
+25%	Avg	1.012	0.995	1.062	1.012	0.997	1.071	1.022	0.998	0.821	1.015	0.991	1.072
	SD	0.022	0.039	0.030	0.025	0.040	0.032	0.030	0.042	0.036	0.021	0.038	0.030
-10%	Avg	1.016	0.992	1.089	1.015	0.994	1.103	1.031	0.997	0.763	1.021	0.986	1.105
	SD	0.037	0.056	0.046	0.040	0.056	0.049	0.047	0.061	0.054	0.035	0.054	0.046
-25%	Avg	1.020	0.990	1.114	1.018	0.992	1.132	1.038	0.996	0.716	1.025	0.981	1.137
	SD	0.051	0.071	0.062	0.056	0.071	0.066	0.064	0.078	0.072	0.050	0.070	0.062

Table 5.7. Sensitivity analysis 4

%Deviation of β		D-AS			S-AS			Always Search			None Search		
		TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP	TPR	TNR	MP
+10%	Avg	1.006	0.998	1.024	1.006	0.999	1.028	1.009	0.999	0.952	1.006	0.997	1.028
	SD	0.009	0.016	0.014	0.009	0.016	0.014	0.012	0.017	0.016	0.009	0.016	0.014
+25%	Avg	1.013	0.997	1.051	1.013	0.997	1.060	1.021	0.999	0.910	1.015	0.994	1.056
	SD	0.008	0.031	0.025	0.008	0.031	0.027	0.020	0.034	0.031	0.010	0.031	0.027
-10%	Avg	1.015	0.995	1.072	1.015	0.996	1.084	1.025	0.998	0.873	1.018	0.991	1.084
	SD	0.022	0.046	0.039	0.021	0.046	0.041	0.035	0.050	0.045	0.023	0.046	0.040
-25%	Avg	1.013	0.993	1.089	1.012	0.995	1.102	1.025	0.997	0.843	1.018	0.988	1.109
	SD	0.033	0.060	0.053	0.034	0.060	0.056	0.051	0.066	0.060	0.036	0.061	0.053

5.5 Conclusion and Discussions

In this chapter, the AS is considered to account for an effective monitoring process focusing on the high potential locations. The D-AS, which aims to balance between inspection cost and system profit as well as system errors, is developed. Two theorems, namely Optimal Expansion of Dynamic Adaptive Search and Diseases Propagation Probability, are derived to support the D-AS. The Optimal Expansion of Dynamic Adaptive Search theorem indicates the expansion level of D-AS to maximum the system benefits. The Probability at Location, which Diseases Propagate theorem provides the CDF at each location to indicate the relative chance diseases are propagated. The procedure is validated against other heuristics and current practice. The results indicate that the D-AS outperforms other heuristics by giving the relatively the same infected/non-infected locations found per inspection, but with the lower cost (higher monitoring profit). Sensitivity analysis shows if the parameters deviate from known value, the D-AS still provide the near-optimal solutions.

CHAPTER 6. MULTI-AGENT SYSTEM OPTIMIZATION IN GREENHOUSE FOR TREATING PLANTS

In this chapter, a protocol called Collaboration Requirement Planning protocol for HUB-CI (CRP-H) is presented (Dusadeerungsikul, Sreeram, et al., 2019). The objective of CRP-H is to improve the plant treatment process in greenhouse by applying cyber collaboration to the system and synchronize system agents. As CRP-H which is a TAP integrated with CRP has two main parts, namely CRP-I (Optimization) and CRP-II (Harmonization), the CRP-I aims to minimize the total operation cost, CRP-II has an objective to sequence agents to work smoothly.

6.1 Cyber Collaborative for Plant Treatment (C2T)

One of the primary operations in PA is applying proper farming resources such as fertilizer, water, and pesticide at a specific location. Consider the objectives of the C2T system as follows: (1) apply fertilizer at given locations with the lowest total operational cost; (2) minimize the total weighted completion time; (3) minimize makespan. The second and third objectives are solved by scheduling. Since plant states are not identical, there are relative priorities associated with each location. For example, higher priority locations can have more severe time constraints than regular priority locations. Also, considering that the system can receive unexpected/emergency requests which can often impact the current schedule, it is advantageous to minimize total completion time as the objective function since the system will complete the highest number of tasks earlier during the planned period. Thus, if there is a new request during the operation, it will have a lesser impact on the overall schedule.

For the rest of the article, the following model is used: The C2T contains two unique types of robots (R_r): R_1 and R_2. There are three possible robot teaming options: R_1, R_2, $R_1 + R_2$; Tasks can be executed either by R_1 or R_2 separately, or by a collaborative team of both R_1 and R_2, $(R_1 + R_2)$. Task types are distinguished based on the ability of either robot or team to accomplish the task, and the performance of the task for each robot/team is monitored.

6.1.1 Cyber Collaborative Greenhouse Operation Types

There are two types of operations in C2T; Planned operation and Unplanned operation. Because of the dynamic environment in the greenhouse, not every operation can schedule beforehand. There is a high potential to receive an additional request during the normal operation process. Hence, the greenhouse should be able to deal with an unplanned request. The following describes details for each operation type.

(1) Planned operation. Under normal operation, the operation starts with the receipt of the task list, indicating the type of farming resources, location, and priority. The task list is a list of locations to treat, which usually receives beforehand according to farming schedule and material planning. The type of package information will specify the handling procedure, which indicates which type and number of a robot(s) are needed. The location is the destination of the robot or team of the robot within the greenhouse. Naturally, the farther the distance of the plant location from the robot-human basepoint, there will be higher operation costs (Liu, Li, Yang, Wan, & Reha, 2011) and longer time required. Lastly, the priority value of a given location indicates its severity of the plant status. Relatively higher priorities (higher priority weight) indicate stricter time constraints for this location, and hence, the tasks associated with this package need to be accomplished earlier than others.

(2) Unplanned operation. During the monitoring process, the system may receive additional requirements called Unplanned Request (UR). The UR must be integrated into the current plan. Typically, re-optimizing the entire array of the remaining tasks along with the UR would provide the best results. However, given that the re-optimizing process usually comes with additional cost, if the additional cost exceeds the saving from re-optimization, the system should have the decision support systems in place to ensure that an entire schedule re-optimization is not required.

6.1.2 Type of Plant Treatment Tasks

In the operation process (either planned or unplanned operation), the task in a greenhouse by robots can be categorized into five types, which are explained as follows as well as in Table 6.1.

Type 1: No-bottleneck. The first type is the simplest task, which can be performed by either R_1, R_2 or the collaborative team, $(R_1 + R_2)$. This is the most common task type in greenhouses.

Type 2: Robot 1 is critical. In this task type, R_1 is necessary to complete the task. The task can either be completed by R_1 or the collaborative team, $(R_1 + R_2)$.

Type 3: Robot 2 is critical. Similar to Type 2, R_2 is necessary for this task type. The task can either be completed by R_2 or the collaborative team, $(R_1 + R_2)$.

Type 4: Complementary task. The fourth type of task requires both robots working together but cannot be accomplished individually, for example, the complex task which cannot be performed alone by either robot but can be performed when they work collaboratively.

Type 5: Constraint task. In this task type, the task can be performed by R_1 or R_2 individually, but not collaboratively as a team. Examples of such tasks can include tasks where the location has space constraints such as narrow aisles.

Table 6.1. Task type in C2T

Task Type	**Robot 1** $(\boldsymbol{R_1})$	**Robot 2** $(\boldsymbol{R_2})$	**Team of robot 1 and robot 2** $(\boldsymbol{R_1 + R_2})$
1	+	+	+
2	+	-	+
3	-	+	+
4	-	-	+
5	+	+	-

6.1.3 Task Operational Cost

The cost for each task to be performed by the robots or their collaborative teams is defined in the CRM. As an assumption, the cost of individual robot allocations is strictly cheaper than when performed by a team. The example of CRM is shown below where c_{mn} is a cost for tasks m performed by robots (or team of robots) n. For task types 1, 2, and 3, if a team of robots performs the task, the operational cost is the total operation cost performed by the individual robots, which is strictly higher.

$$CRM = \begin{bmatrix} c_{11} & \cdots & c_{m1} \\ \vdots & \ddots & \vdots \\ c_{1n} & \cdots & c_{mn} \end{bmatrix} \tag{6.1}$$

6.1.4 Task Operational Time

The operational task time is the total time to complete the task, which includes receiving the order, picking the package, moving to the location, storing the package, and moving back to the base location. For task types 1, 2, and 3, if a team of robots performs the task, the operational time is assumed to be strictly lesser. In other words, the team of robots can perform the task at a faster rate than an individual robot operation.

6.2 Cyber Collaborative Greenhouse System Architecture

The designed C2T system has three main agents; human operators, robots, and sensors. The system receives input from human operators via a spatial-visual programming software called VIPO (G. Huang et al., 2020). VIPO allows human operators to allocate system tasks in the spatial context within an interface, which can minimize the learning time. An output from VIPO is a computer script that indicates the type of package, location to store, and priority. VIPO symbolizes the human in the loop (HITL) components of this article, where human operators can schedule and allocate tasks to the robots based on their expertise.

There are two types of input from the human operator to VIPO - normal input and additional input. The normal input which enters the system before the operation begins will generate to the planned operation, and additional input will create UR. The input will feed to HUB-CI, a system brain model, which can manage tasks in real-time and deciding to maximize system performance.

HUB-CI maintains the Collaboration Requirement Planning protocol for HUB-CI (CRP-H). CRP-H connects CRP to TAP for both allocating tasks to agents (by CRP) and optimize workflow and interaction among system agents (by TAP). The CRP, which is the fundamental design principle in CCT, has two main modules; optimizer (CRP-I) and harmonizer (CRP-II). The optimization module is responsible for assigning task(s) to a robot or a team of robots (one to one or one to many matching). The harmonization module takes care of sequencing and scheduling a robot or team of robots to perform tasks.

After the inputs are processed by HUB-CI, utilizing CRP-H, the generated plan will be executed and simulator. Besides, the plan will also be distributed to other system agents. The system architecture is presented in Figure 6.1.

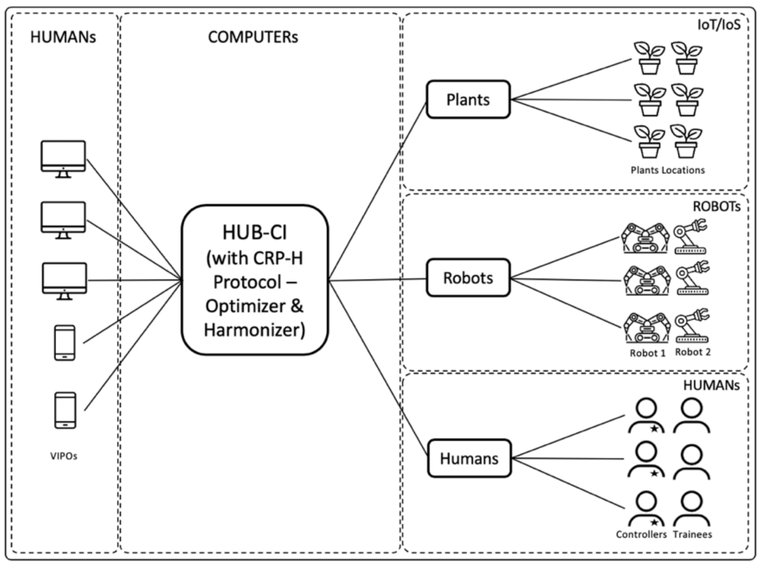

Figure 6.1. Cyber Collaborative Greenhouse system architecture

6.3 CRP-H Protocol design

For effective coordination, a Collaboration Requirement Planning protocol for HUB-CI called CRP-H is developed in this section. The CRP-H is the workflow optimization and collaboration protocol in HUB-CI for task allocation and task scheduling/sequencing. Figure 6.2 presents the CRP-H and its components.

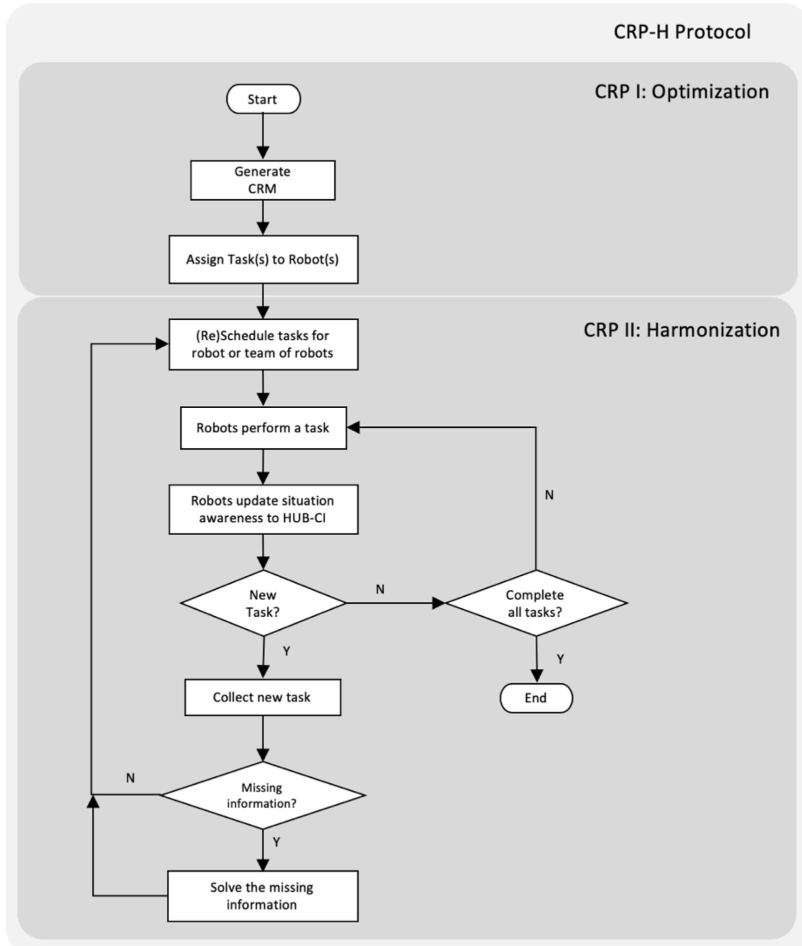

Figure 6.2. CRP-H protocol

As mentioned before, CRP-H has two modules, Optimization (CRP-I) and Harmonization (CRP-II). The following section will describe each part of CRP-H in detail.

6.3.1 CRP-I Optimization

The CRP-I aims to minimize total operating costs for planned operations. The mathematical model for CRP-I is presented as follows.

6.3.2 A mathematical model for CRP-I

Mathematical model

Let

$$c_{mn} = cost\ for\ task\ m\ is\ performed\ by\ a\ robot\ or\ team\ of\ robots\ n$$

$$x_{mn} = \begin{cases} 1\ if\ task\ m\ is\ performed\ by\ a\ robot\ or\ team\ of\ robots\ n \\ 0\ otherwise \end{cases}$$

$$N_R\ =\ Number\ of\ robot\ or\ team\ of\ robots$$

$$N_T = Number\ of\ task$$

$$N_m\ = Number\ of\ task\ in\ type\ j$$

$$Maximum\ loading\ factor = \max\left(\left\lceil\frac{N_T}{N_R}\right\rceil, \max_m(N_m)\right)$$

Objective function

$$min\ z\ = \sum_m \sum_n c_{mn} x_{mn} \tag{6.2}$$

Subject to

$$\sum_n x_{mn} = 1\ ; \forall m \tag{6.3}$$

$$\sum_m x_{mn} \leq K\ ; \forall n \tag{6.4}$$

$$x_{mn} \in \{1,0\}, \forall m, n \tag{6.5}$$

$$m = 1,2, \dots, \mathcal{M}$$

$$n = 1,2, \dots, \mathcal{N}$$

As mentioned before, the objective of the optimizer is to minimize total operational costs presenting in Equation (6.2). c_{mn} is the operational cost defined in the CRM. The CRM will update continuously with the new information received from IoT/IoS agents and human operators.

The first constraint, Equation (6.3), ensures that a robot or team of robots will perform all tasks. The second constraint, Equation (6.4), ensures that the robot or team of robots will not be overloaded.

In the planned operation, the input data (task, distance, and priority) are received in advance. CRP-I, which requires more computational power and processing time due to the volume of data being processed, and hence can be initiated ahead of the actual operation.

The output from CRP-I is the assignment of the task(s) to robot or team of robots. Two types of assignments from the CRP-I is (1) collaborative assignment and (2) non-collaborative assignment. The collaborative assignment is the task that assigns for a team of robot ($R_1 + R_2$) while the non-collaborative assignment is the task which assigns to a single robot (R_1or R_2).

The output, however, does not indicate the sequence of tasks for each team of agents. Therefore, the harmonization module (CRP-II) is necessary.

6.3.3 CRP-II Harmonization

The second phase of CRP-H is harmonization. The harmonizer has the main objective of the sequencing task at each robot or team of robots concerning minimizing makespan (C_{max}) and total summation of weighted completion time ($\sum w_i C_i$). The objective is selected to ensure that in case of unexpected conditions such as new UR, robot delays, operation conflicts, and errors, the high priority tasks are scheduled as early as possible to minimize heavier losses due to re-optimization.

6.3.4 A mathematical model for CRP-II

The mathematical model for harmonization can be presented as below.

Mathematical model

Let

$$y_{mt} = \begin{cases} 1\ if\ task\ m\ start\ at\ time\ t \\ 0\ otherwise \end{cases}$$

$$p_m = processing\ time\ of\ task\ i$$

$$w_m = priority\ of\ task\ i$$

Objective function

$$\min z = \sum_{m} \sum_{t} w_m (t + p_m) y_{mt} \tag{6.6}$$

Subject to

$$\sum_{t=0}^{C_{\max}-1} y_{mt} = 1\,; \forall m \tag{6.7}$$

$$\sum_{m=1}^{\mathcal{M}} \sum_{u=(\max(t-p_m,0)}^{t-1} y_{mu} = 1\,; \forall t \tag{6.8}$$

$$y_{mt} \in \{0,1\}; \forall m, t$$

$$m = 1,2, \dots, \mathcal{M}$$

$$t = 0,1,2, \dots, C_{\max} - 1$$

The objective function, Equation (6.6), ensures the minimization of total weighted completion time. The first constraint, Equation (6.7), ensures that any task has a single starting point. The second constraint, Equation (6.8), ensures that only one task can be performed at a time. By solving the above mathematical model, the schedule that minimizes the total weighted completion time is generated. The minimization of makespan is ensured by having a non-delay schedule.

It, however, needs significant computational power and time. Therefore, an algorithm for CRP-II, called Collaborative Robots Scheduling (CRS), is developed.

6.3.5 Collaborative Robots Scheduling (CRS) Algorithm

As the harmonization module provides real-time control and adaptation from the new information, harmonization will be performed at the local agent(s) levels to provide efficient responsiveness. Solving the mathematical model presented before which contains a large number of decision variables ($t \times y_{mt}$) requires powerful computational power, which is difficult given time constraints and local agent limitations. With the objective of harmonizer, an algorithm called Collaborative Robots Scheduling (CRS) is developed to ensure that the optimal schedule is achieved locally at lower computational costs.

The CRS utilizes the advantages of the Weighted Shortest Processing Time first (WSPT) algorithm that yields the optimal solution for the total weighted completion time problem by developing multi-level of WSPT. The CRS algorithm is as follows.

	Algorithm 6.1: CRS Algorithm
1.	Initialize
2.	Parameter
3.	Schedule collaborative assignment tasks according to WSPT (called collaborative schedule)
4.	Schedule non-collaborative assignment tasks according to WSPT (called non-collaborative schedule)
5.	Combine the collaborative schedule with the non-collaborative schedule while the release time of non-collaborative schedule is makespan (C_{max}) of the collaborative schedule
6.	Terminate algorithm

Theorem 6.1: The optimal schedule for collaborative robots

The CRS yields the optimal total weighted completion time for each robot and team of robots.

Proof. At each particular robot (and a team of robots), the problem becomes $1||\sum w_m C_m$ problem.

Let task m has a weight w_m, the time to process p_m, and task m' has a weight $w_{m'}$, the time to process $p_{m'}$ so that $\frac{w_m}{p_m} < \frac{w_{m'}}{p_{m'}}$.

Suppose schedule S', which starts at time t contradicted with WSPT rule. There will be at least one pair of the task such that $\frac{w_m}{p_m} < \frac{w_{m'}}{p_{m'}}$. However, task m is placed before m' (Figure 6.3). The total weighted completion time of S' is $\gamma + w_m(t + p_{m'}) + w_{m'}(t + p_m + p_{m'})$ where γ is the total weighted completion time of all tasks except m and m'.

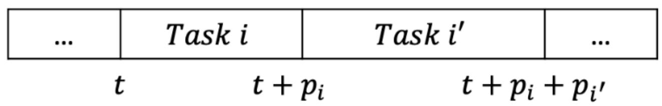

Figure 6.3. Schedule S'

If task m and m' are interchanged and produce schedule S (Figure 6.4), the total weighted completion time of S is $\gamma + w_m(t + p_{m'}) + w_{m'}(t + p_m + p_{m'})$. Note that γ of S and S' are the same as all tasks except m and m' reaming unchanged.

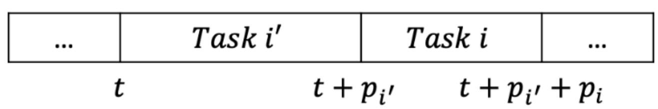

Figure 6.4. Schedule S

Because $\frac{w_m}{p_m} < \frac{w_{m'}}{p_{m'}}$, then $w_m(t + p_{m'}) + w_{m'}(t + p_m + p_{m'}) > w_m(t + p_{m'}) + w_{m'}(t + p_m + p_{m'})$ Moreover, the new schedule, which follows WSPT, has a strictly lesser objective function.

Moreover, since a non-collaborative task requires the strictly longer processing time (by definition of the task) and has strictly lower priority (because it uses only a single robot), to schedule the task optimally, the non-collaborative schedules must be released after the completion of the collaborative schedule.

■

Theorem 6.2: The guarantee optimal makespan for collaborative robot schedule

The optimal makespan C_{max}^{*} of the system is.

$$C_{\max}^{*} = \sum_{k \in R_1 + R_2} p_k + \max\left(\sum_{m = R_1} p_m , \sum_{m' = R_2} p_{m'} \right) \tag{6.9}$$

Proof. There are two sub-schedules for the makespan; makespan form collaborative schedule and from the non-collaborative schedule.

First, consider a robot team $(R_1 + R_2)$. Regardless of the sequence of tasks, the makespan of the robot team equals to the total processing time of task assigned to the team $(\sum_{k \in R_1 + R_2} p_k)$.

Next, consider a single robot agent (R_1 and R_2). Each robot can work individually, and the makespan of all single robot agent become maximum total completion time of all robots $(\max(\sum_{R_1} p_i , \sum_{R_2} p_{i'}))$. Also, the earliest time that each robot can start working is immediately after the collaboration task has done. Therefore, the optimal makespan for the robotic schedule is.

$$C_{\max}^{*} = \sum_{k \in R_1 + R_2} p_k + \max\left(\sum_{m = R_1} p_m , \sum_{m' = R_2} p_{m'} \right)$$

■

6.4 Human Role and Human-in-the-loop Design for C2T

Human support has become an essential aspect of production system design, assessment, and improvement (Zhang, Schmidt, Schlick, Reuth, & Luczak, 2008). In the C2T, human subject matter expert (SME) is considered as an intelligent agent who is capable of providing decision-making capabilities in real-time (Dusadeerungsikul & Nof, 2019). SME's can fill in the missing information in tasks which contain incomplete information and thus restructure the system. For example, the system might come across packages that might have incomplete or erroneous data. In such cases, it is up to the human operators to use their expertise to provide the missing data for the packages, which can include appropriating the priority weights of the packages, updating the plant location, adding these packages to the queue once the information is complete. The humans interact with the system in real-time, and the performance metrics are collected appropriately. In the experiments, the SME's are simulated re-prioritizing a fixed proportion of the task queue, and these added changes occur in real-time. Thus, the system considers the dynamic impact of modeling humans as physical agents capable of leveraging expertise for decisions and thus can prove to be more robust than entirely autonomous systems.

6.5 Experiments

To validate the protocol and algorithm designed, three computer simulations experiments are constructed as follows.

6.5.1 General Experimental Setting

Mode of operation

For all cases, the greenhouse scenario with the monitoring tasks is simulated. Any task can be performed in three ways: either by R_1, R_2, or $R_1 + R_2$.

Time

In a greenhouse scenario, the operation can be simplified into a single metric – processing time. The processing time for each task by one of the three-team is derived from travel distance with the assumption that the collaborative team achieves a faster processing rate than the task done by either R_1 or R_2.

Capability

As mentioned earlier, different tasks have distinct operational requirements. The robots' ability to complete a task is assigned to the five potential types of operation.

Weight

Weight denoting the priority level between 0 to 10 is assigned to each task. Weight 10 being the highest priority, while the task with weight 0 does not have a deadline.

Cost

The cost of operation is directly proportional to time with the fixed cost. Each robot has a fixed price for every time unit in operation. The unit cost of the collaboration team is the summation of the participating agents.

6.5.2 Modeling Procedure

The simulation experiments are constructed in MATLAB. Given the above experimental settings, the collaborative tasks incur higher costs for a faster processing rate. The performance metrics are the total operation cost, makespan (C_{max}) and weighted completion time ($\sum w_m C_m$). In tune with CRP, the CRM that reflects the status of the system is updated in real-time. Hence, CRM is a dynamic matrix that is administrated and updated by the administrator of the system.

6.5.3 Experiment 6.1: Performance Analysis in Planned Operations

Experiment 6.1 aims to compare the designed protocol with the standard procedure (Baseline). The system description is as follows.

System Description

Agents: 2 robots (R_1 and R_2) and 1 robot team ($R_1 + R_2$.)

Tasks: 100 tasks with different priorities and available immediately at the beginning of the operation time.

Operation procedures: Two types of operation procedures.

(1) CRP – H: The designed protocol.

(2) Baseline procedure: Random task to robot/team of robot assignment with First come first serve (FCFS) scheduling

6.5.4 Experiment 6.2: Performance Analysis with Unplanned Operations

Experiment 6.2 aims to compare the performance of the CRP-H with the baseline when unplanned operations happen (e.g., UR). The designed protocol is tested against the standard procedure to deal with unplanned operations (Baseline). The system description is as follows.

System Description

Agents: 2 robots (R_1 and R_2) and 1 robot team ($R_1 + R_2$.)

Tasks: 90 tasks available immediately with different priorities and ten tasks added after the beginning of the operation.

Operation procedures: Two types of operation procedures.

(1) CRP – H: The designed protocol with re-schedule the new task according to CRP-II

(2) Baseline procedure: Utilizing CRP-I for the task to robot/team of robot assignment with First come first serve (FCFS) for the new tasks

6.5.5 Experiment 6.3: Performance Analysis with HITL design

Experiment 6.3 allows human experts to be involved in the decision-making process. The CRP-H, which allows with human-in-the-loop features, is validated against the non-human procedure. The system description is as follows.

System Description

Agents: 2 robots (R_1 and R_2) and 1 robot team ($R_1 + R_2$.)

Tasks: 90 tasks available immediately and ten tasks without priority added after the beginning of the operation.

Operation procedures: Two types of operation procedures.

(1) CRP – H: The protocol with HITL.

(2) Baseline procedure: The protocol without HITL.

6.6 Results

6.6.1 Experiment 6.1 Results

In experiment 6.1, the result from the random task to agent assignment with FCFS rule is used as the design baseline. Table 6.2 and Figure 6.5 summarizes two performance metrics based on the result of 100 operation runs. The average total operation cost and the average weighted completion time of the CRP-H is 11.84% and 37.11% lower than the baseline, respectively. At the significance level 0.05, two sample standard t-tests ($p<0.0001$) confirm that the CRP-H significantly outperforms the baseline both in terms of total operating cost and total weighted completion time. CRP-H yields statistically significant lower makespan than the baseline as the result of the optimization objective of cost and priority. Importantly, the makespan from the CRP-H has met the guarantee optimal makespan from Theorem 2.

Table 6.2. Result of Experiment 6.1

	CRP-H	**Baseline**	**% Difference**
Average Total Operational Cost ($)	4916.02 (235.97)	5576.10 (267.65)	11.84%*
Average Total Weighted Completion time (sec)	1.61×10^5 (2.01×10^4)	2.56×10^5 (2.87×10^4)	37.11%*
Average Makespan (sec)	893 (75.91)	901 (58.57)	0.89%*

Note: Standard deviations are given in parentheses, * Statistically significant at ($p<0.0001$)

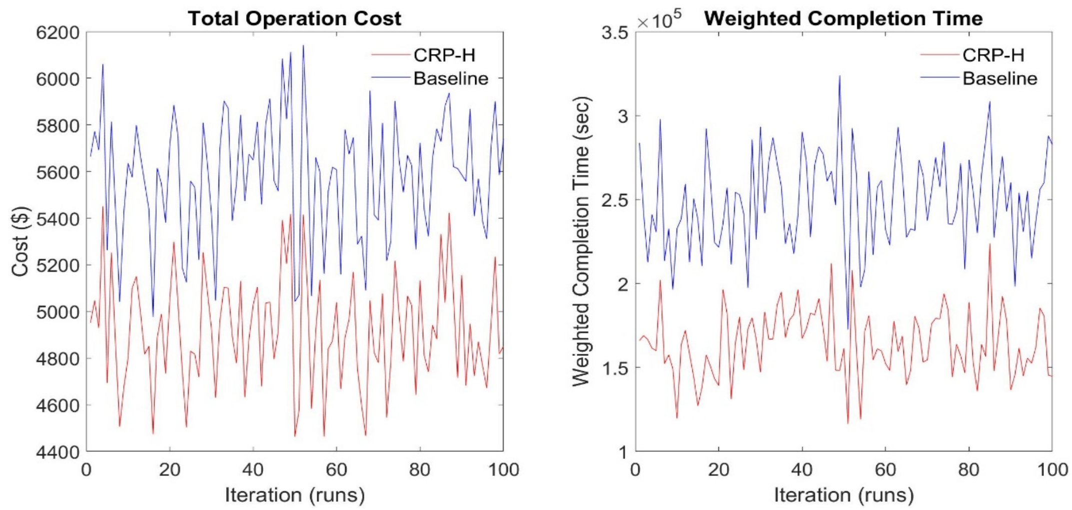

Figure 6.5. Experiment 6.1 results

Next, to see the impact of the saving from the CRP-H, the cost and time saving is calculated, as shown in Equation (6.10) and (6.11).

Cost-saving

$$\Delta Cost = C_{Basedline} - C_{CRP-H} \tag{6.10}$$

Where

$\Delta Cost$ = Cost saving

$C_{Basedline}$ = Total operation cost from the baseline procedure

C_{CRP-H} = Total operation cost from CRP-H

Time-saving

$$\Delta Weighted\ Completion\ Time = W_{Basedline} - W_{CRP-H} \tag{6.11}$$

Where

$\Delta Weighted\ Completion\ Time$ = Time saving

$W_{Basedline}$ = Total weighted completion time from the baseline procedure

W_{CRP-H} = Total weighted completion time from CRP-H

Figure 6.6 shows the cost and time-saving at various task loads. The results show that cost-saving is linearly proportional (R^2 =0.997) to the increasing number of tasks, and, interestingly, the time saving is in a polynomial relationship, non-linear relationship (R^2 =0.981) with the number of tasks in the queue. The results also validate the robustness of the CRP-H in a multi-robot task allocation problem as both cost and time saving are always positive.

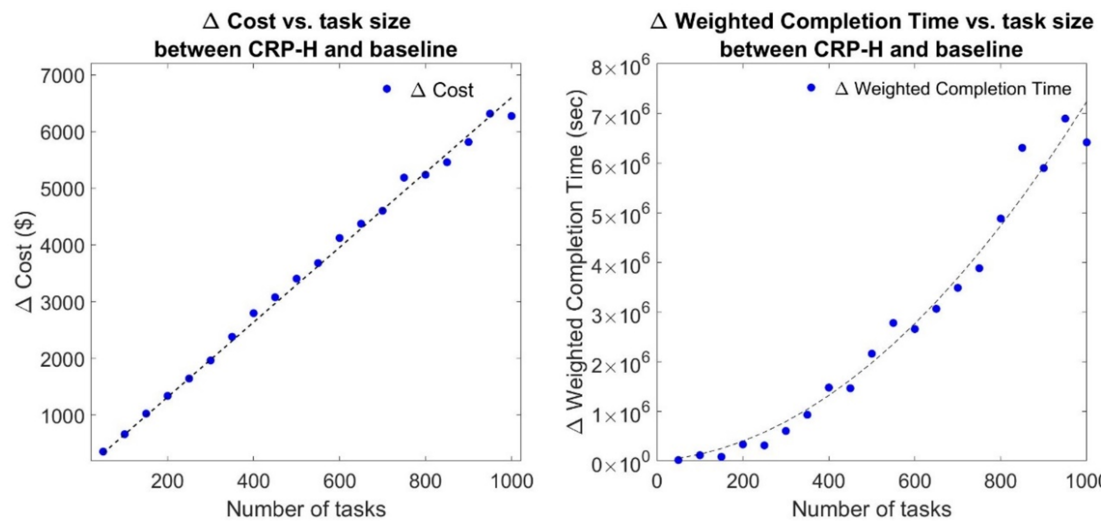

Figure 6.6. Experiment 6.1: Cost and time-saving at different number of tasks

6.6.2 Experiment 6.2 Results

In Experiment 6.2, 10% of the tasks are added to the queue after the robot-task assignment and optimizing the schedule has performed. The goal of the experiment is to understand the impact of dynamic changes in the environment and determine the criteria for re-optimization. Table 6.3 and Figure 6.7 compare the performance metrics from the result of 100 operation runs. Even though the difference in total operation cost arises from the 10% of the tasks that are inserted after initiating the work sequence (due to the fact that the majority of the task sequence had been optimized under CRP-I), at significant level of 0.05, the average operational cost of CRP-H is lower (p=0.04) than the baseline. Also, the weighted completion time of CRP-H is 10.7% lower (p<0.0001) than the baseline. In the actual working scenario, the criteria for re-optimization is when the saving in cost and weighted completion time is more significant than the fixed optimization cost. CRP-H also provide a statistically significant lower makespan than the baseline. Note that the makespan by the CRP-H in the experiment 2 is larger than the guaranteed bound from Theorem 2. The makespan from the theorem captures only planned operation while the actual makespan reflects the URs.

Table 6.3. Result of Experiment 6.2

	CRP-H	**Baseline**	**Difference %**
Average Total Operational Cost ($)	4916.02 (235.97)	4984.31 (234.49)	1.37%*
Average Total Weighted Completion time (sec)	1.61×10^5 (2.01×10^4)	1.84×10^5 (2.11×10^4)	12.50%*
Average Makespan (sec)	893 (75.91)	897 (75.35)	0.45%*

Note: Standard deviations are given in parentheses, * Statistically significant at ($p<0.0001$)

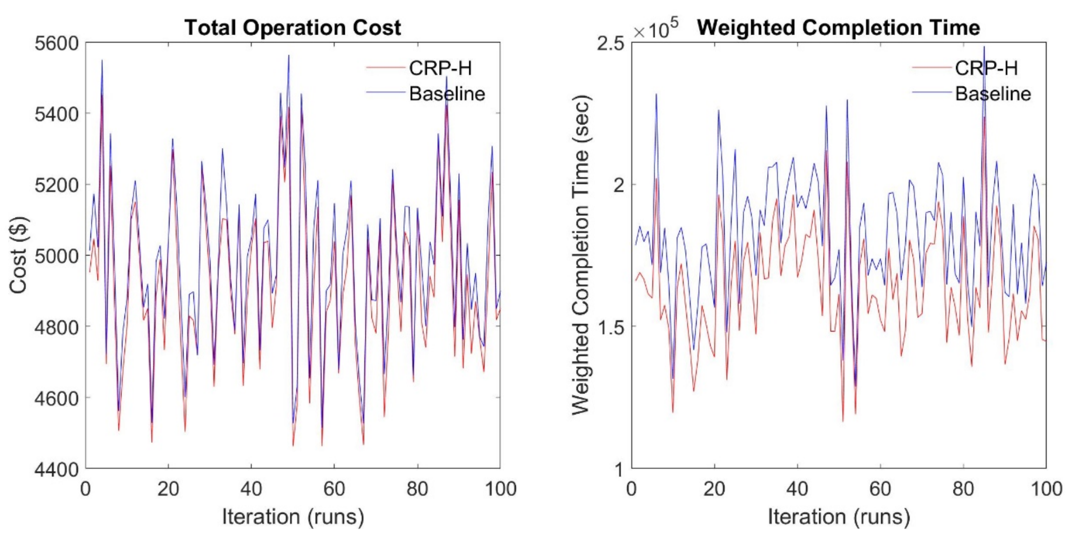

Figure 6.7. Experiment 6.2 results

6.6.3 Experiment 6.3 Results

In Experiment 6.3, a certain percentage of tasks received during the operation process (unplanned requests, UR) arrive with missing information, such as priority or deadlines. The experiment studies the impact of human intervention in the task allocation and sequencing problem. For this experiment, the human agent assigns the missing priority of tasks before the optimization, as compared to the baseline state where these tasks are added at the end of queue with minimal priority. The results for 10% of UR are shown in Table 6.4 and Figure 6.8, which suggests that HITL design significantly improves both the total operational cost and weighted completed time when compared to the baseline ($p<0.0001$). An additional human involvement cost for the reprioritization tasks is assumed to be zero, since the involvement can be considered part of routine human tasks. On the other hand, if there is an additional human involvement cost, the cost should be lower than a cost threshold (the difference between total operation cost from CRP-H and baseline) to be considered as a cost-effective situation.

Table 6.4. Result of Experiment 6.2

	CRP-H	**Baseline**	**Difference %**
Average Total Operational Cost ($)	5071.68 (239.46)	5437.81 (267.65)	6.73%*
Average Total Weighted Completion time (sec)	1.61×10^5 (2.00×104)	1.83×10^5 (2.12×104)	10.23%*
Average Makespan (sec)	893 (19.32)	897 (11.35)	0.45%*

Note: Standard deviations are given in parentheses, * Statistically significant at ($p<0.0001$)

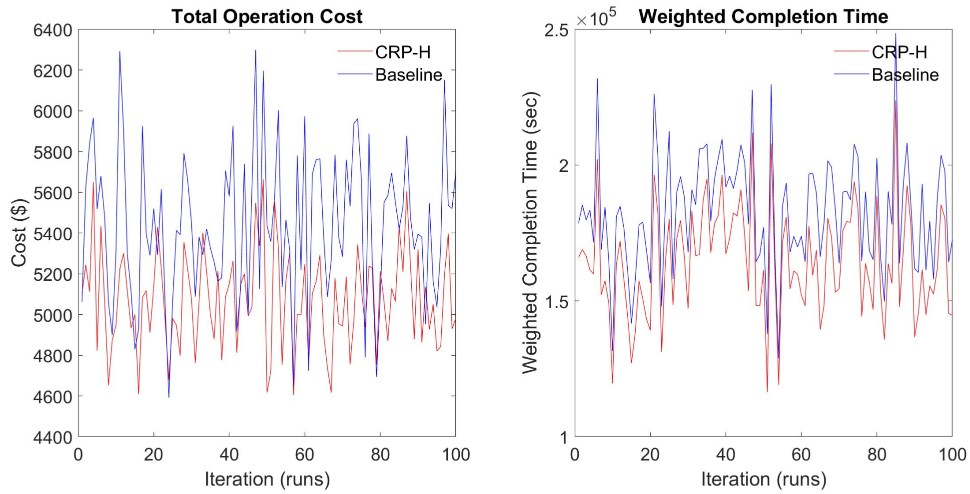

Figure 6.8. Experiment 6.3 results

Figure 6.9 shows the cost difference at various UR percentages. The results show the cost differences are linearly proportional (R2 = 0.998) to the increasing percentage of unplanned tasks, which suggests that with increasing unplanned or unexpected events, CRP-H improves cost reduction via augmented stabilization of performance.

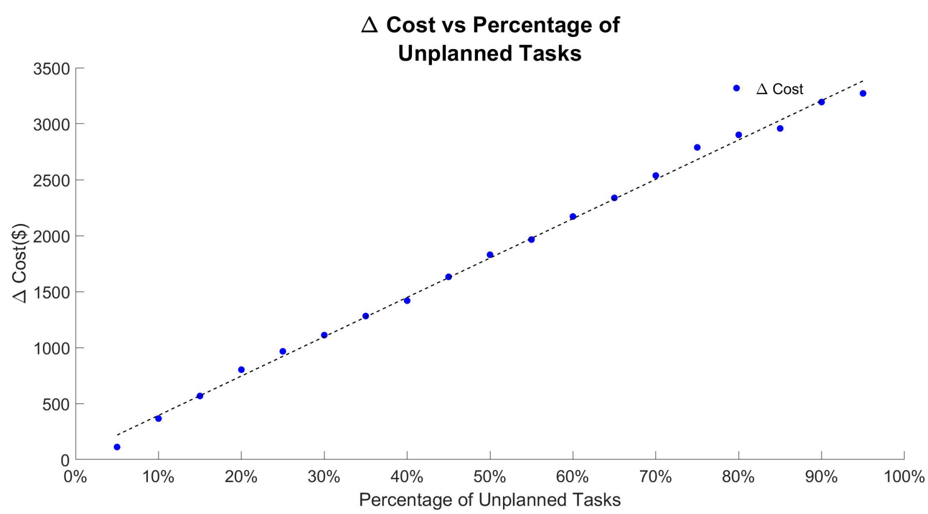

Figure 6.9. Cost differences between CRP-H and baseline when percentage of UR increases

6.7 Conclusion and Discussion

As stresses are usually the initial states of diseases, detecting and treating them early is critical. In this chapter, a new protocol called CRP-H is designed, developed, and validated to address such a challenge. By studying a cyber collaborative system, five types of tasks are defined to represent the greenhouse functional tasks. CRP-H is composed of two main parts: CRP-I for a task(s) to robot(s) assignment optimization; and CRP-II for task sequencing, scheduling, and harmonization. CRP-I operates at the global level of the C2T, which requires high computational time while CRP-II operates at the local-agent level, which has only limited computational power to respond to dynamic change in the greenhouse.

Two theorems are presented in this article. Theorem 6.1, the optimal schedule for collaborative robots proves that the CRS algorithm provides the optimal schedule for collaborative robots in the greenhouse. Theorem 6.2, the guarantee optimal makespan for collaborative robot schedule, can provide the optimal makespan of the collaborative robot schedule. The guaranteed optimal makespan has utilized information available at the time. If the system obtains URs after makespan is calculated, the actual system makespan is greater than or equal to the optimal makespan from the Theorem 6.2.

Based on the three experiments, observations indicate that the Collaboration Requirement Planning protocol for HUB-CI (CRP-H) delivers superior performance in terms of total operation cost, makespan, and total weighted completion time when compared to a common practice in today's operations. Lower operational cost is enabled by the use of the CRP-I part of the CRP-H, which optimally assigns tasks to the robot(s). Moreover, total weighted completion time and makespan are minimized because of the CRP-II part of the CRP-H, which can update the execution schedule in real-time, by the collaborative cyber connectivity with IoT/IoS devices' information. The experimental results also show that HITL design can help the system become more robust, given that the versatility of human decision-making is appropriately applied. HITL Systems help to deal with new and emergent errors such as diseases in plants, without a significant increase in cost.

Importantly, as presented in Figure 6.6, the experiments also show that when the number of tasks received by the C2T is increased, the CRP-H can linearly improve the performance of the system in terms of total operation cost. Also, the total weighted completion time is improved in the polynomial relationship to the number of tasks. In addition, Figure 6.9 shows the cost saving from having HITL can linearly save the operation cost when percentage of UR increases.

In the practical implications, the CRP-H can save both money and time for the system without additional investment. CRP-I can be performed in the background before the operations begin. CRP-II, which requires only small computational power due to the simple rules of algorithm, can support the system by adjusting the operation during the ongoing process to respond to the unplanned requests. Additionally, with HITL design, the system can overcome the unexpected situation such as missing data with the minimal incremental cost. On the other hand, if the additional cost due to HITL (e.g., worker cost) exceeds the difference between CRP-H and the baseline case, then it is not attractive to maintain HITL. It, however, must be mentioned that if the human input were not included, the tasks would either incur hidden costs from erroneous from missing information, or cost from wrong treatments, both of which are undesired outputs. Human agents can identify erroneous cases and assign missing priority to them, thus preventing further and future losses in cases.

CHAPTER 7. CONCLUSION AND DISCUSSION

7.1 Summary and Original Contributions

Food security has drawn a lot of attention from society now. Because of food security, researchers and farmers grow plants in greenhouse to protect them from undesirable conditions and to maximize the crops output. Plant, however, are far from precision controlled. Moreover, they are fostered and manipulated to produce crops in the unusual cycle. Such conditions, in many cases, causes plants to develop stresses or diseases, leading to loss of crops yields. To minimize the loss, farmers utilize uniform resource management i.e., apply water, fertilizer, pesticide evenly over the greenhouse field. The uniform resource management approach, however, has been proven to be the non-optimal solution for food production in both profit and environmental protection.

The alternative solution to the food security is PA. PA has a high potential in the near future. Many international companies are developing and testing real-world applications of PA. PA aims to apply the right amount of resources such as water, fertilizer, and pesticide to a specific location at the right time so that the farming resources are minimized and environmental is protected.

The critical point of PA is how to use limited resources effectively. The resources are not only water, fertilizer, and pesticide, but also time and effort which are put in the farming process. Moreover, information obtained during the farming process must be utilized effectively in order to achieve the optimal output in PA. As the farming equipment and automation are limited in terms of operation hours, the procedure to maximize information gained within the limited time is essential.

In this study, the monitoring system which inspects the status of the plant in the greenhouse is the focus. Because knowing the status of the plant in a specific location can provide directions on how to allocate resources properly, the monitoring system is necessary. The study has four sub-studies that aim to improve the monitoring process. The first study, which utilizes CPS for PA, is the backbone of the following studies as it develops the high-level picture of the monitoring system. The second study is the detailed operations of how local agents work together to perform the collaborative monitoring task. The third study analyzes the performance of the local agent's algorithm and provides the optimal procedure to reduce time while maximizing the information gain. Lastly, the fourth study is to design the optimal procedure after receiving information about the status of plants.

In order to have an effective monitoring system, the study first explores the roles and implementation of CPS in a greenhouse. Because of the newly emerged approaches, namely CPS, IoT/IoS, TAP, CCT, and PCol, CCP-CPS can perform collaboratively and effectively. The CCP-CPS utilizing CCT, TAP, and PCol is developed for coordinating and optimizing system agents (humans, a mobile robot, and sensors), which can increase the productivity of the system to achieve its goal. Working collaboratively, which tends to minimize resources used, is essential when resources are limited. Also, CPS and IoT/IoS help the system to have effective real-time communication and control, which also improves system performance.

Three experiments are conducted in the first study to investigate the different perspectives of CCP-CPS. The first experiment aims to compare the operation cost for the CPS and Non-CPS (current practice) schemes. The result shows that with the CPS, the system yields a significantly lower operation cost by 23.5%. Because of CPS enables real-time communication and response, costs which relate to traveling and delaying are minimized. Next, the second experiment aims to test the robustness of both schemes. With changing system C&E, the results show that the operation time from the non-CPS scheme significantly increases when compared to the CPS scheme. In the agriculture context, which is relatively unstructured by nature, a system that can tolerate to C&E in the system is preferred. Hence, the CPS scheme shows superior performance than the non-CPS scheme. Lastly, the third experiment investigates the performance when the robot receives the emergency request. The experimental result indicates that, with CPS, the response time of the robot is significantly lower than the non-CPS scheme.

Even though the CPS designed scheme provides thee relatively superior performances in terms of cost, robustness, and time, the scheme does not indicate how the local agents interact with each other. Hence, a new protocol called CCP-ED is developed in the second study. The protocol aims for managing local agents, namely humans, robots, and sensors, to inspect plants. Also, the protocol has three main algorithms, routing algorithm, adaptive search algorithm, and stress detection algorithm. The algorithms enable agents to perform the assigned tasks.

Two performance metrics are applied to capture the performance of CCP-ED; NIPF and ORE. The experimental results indicate that the CCP-ED yields the better in terms of infected locations found and efficiency. CCP-ED can significantly improve NIFP, which represents the performance of the system. Also, CCP-ED yields significantly higher ORE, which represents the system efficiency. As ORE consists of three components; U, P, and S, the CCP-ED, which results in 85.3% ORE, is considered a healthy system. Lower than 80% of ORE would be considered as an inefficient system as the system wastes time for performing unnecessary tasks. On the other hand, nearly 100% of ORE would not be desirable, especially in the agriculture system, as it lacks flexibility. Lacking flexibility would result in the incapable of acting to the change in the system, which may lead to the inability to respond to urgent situations.

The CCP-ED provides insight into how the collaborative system operates. It would be more meaningful to further investigate one of the core algorithms, the Adaptive Search algorithm, to suggest the optimal procedure of the system. In the third study, the objective is to investigate the optimal procedure of the AS to save the inspection time and provide the maximum information gain. Moreover, to reflect the real system situation, the procedure will include C&E, which impacts the decision of AS. The inspection process may have errors (either type 1 or type 2), which can affect the performance of the system significantly. Therefore, the D-AS is developed to improve the search procedure by incorporating system errors and disease's characteristics, such as disease generated probability and disease propagation probability.

Two theorems are developed and proved based on the observation and finding in the system characteristics. The first theorem, Optimal Expansion of Dynamic Adaptive Search, indicates the optimal expansion of D-AS with respect to the character of disease as well as error in the system. The second theorem, Diseases Propagate Probability, indicates the probability of diseases found at each particular location.

Three experiments are conducted to have a more in-depth investigation of the D-AS. In the first experiment, performance analysis aims to investigate inspection performance in terms of TPR and TNR. TPR and TNR capture the quality of inspection ratio (number of true infected/true healthy plant found against complete inspection). Therefore, the superior search policy must have a higher value of TPR and TNR. The experimental results show that the D-AS provide the significantly better (or the same) as the alternative search policies. As the D-AS adapts to the situation of disease, in the situations in which diseases are rare to generate and propagate, the D-AS will perform as None Search policy.

On the other hand, when the probability of disease to generate and propagate is higher, the D-AS will expand the search accordingly. As a result, it keeps the D-AS at least as good as other search policies, and in most of the case, the D-AS perform significantly better in term of TPR. In terms of TNR, the D-AS does not perform significantly differently from others.

The second experiment, cost analysis, investigates the performance in terms of cost, and benefit from the inspection process. MP is utilized to measure the inspection profit for each search policy. As the information gain during the inspection process is very important for planning resources and treatment for crops, the more useful information the system gain, the better the system is. Results show that D-AS increases MP by 12.8% compare to the current practice (None-Search policy). It means that using the D-AS as a search policy can yield the maximum inspection profit to the system.

Lastly, to investigate the sensitivity of the D-AS and other search policies, the third experiment is conducted. By having the actual parameters (such as the probability of disease emerges and the probability of disease propagation) deviates from the setting, changes in results from the optimal solutions are defined as the sensitivity of the algorithm. The experimental results show that D-AS provides a robust result (non-sensitive). When the input parameters are deviated by 25%, the results deviate from the optimal solution by 4.9%. On the other hand, the other search policies are less robust to the change in the input parameters (more sensitive). As discussed earlier, in the agricultural context, the system has high potential to have deviation of inputs. For example, the probability of disease to generate may not be exact to the setting parameter. As a result, the system which can work efficiently even though the deviation of the setting exists is preferred. Moreover, the system may contain errors as errors are typically found in every system. The D-AS, which balances between over inspection (by type 1 error) and under inspection (by type 2 error), can provide the optimal procedure for AS.

After inspecting and identify the stress locations, the next important task is to send the robot to treat or remove the plants. Therefore, the last protocol, CRP-H, is designed, developed, and validated to address the task. Five types of tasks are defined to represent greenhouse functional tasks. CRP-H is composed of two main parts: CRP-I and CRP-II. The CRP-I which performs in a global level aims to optimally assign task(s) to robot(s) (many to many matching) by minimizing the total operation cost. As a result, the CRP-I requires high computational power because of the complexity of the problem. On the other hand, CRP-II which aims to response to the real-time information needs to operate at the local-agent level which has only limited computational power.

Two theorems are derived from the analysis of observations. Theorem 6.1, Optimal Schedule for Collaborative Robots, indicates that the CRS algorithm yields the optimal schedule for collaborative robots in the greenhouse. Next, theorem 6.2, Guaranteed Optimal Makespan for a Collaborative Robot Schedule, proves the optimal makespan (lower bound) of the collaborative robot schedule in greenhouse.

In order to validate the performance of CRP-H, three experiments are conducted. Three performances of the CRP-H are captured by three main metrics, total operation cost, makespan, and weighted completion time. The experimental results show that CRP-H yields superior performance comparing to the current practice. CRP-I enable the operation cost saving because the CRP-I assign task(s) to robot(s) optimally. In addition, the CRP-II which obtains real-time updated information and execution schedule by the collaborative cyber connectivity with IoT/IoS devices in greenhouse schedules tasks according to CRS algorithm and non-delay schedule policy so that the makespan and total weighted completion time are minimized. Lastly, the experimental results indicate the benefit of the HITL design that can increase the robustness of the system. Because the human experts can deal with the unexpected situations such as priority of the task, the total operation cost is minimized.

The original contributions of the study are:

1. *Contribution 1*; A CPS framework for the agricultural systems, a representative of an unstructured system. Also, the framework is adaptable to other unstructured systems.
2. *Contribution 2*; A collaborative control protocol which can synchronize operations and communications at the local level agents containing simple operation algorithms.

3. *Contribution 3*; Analysis of operations in an unstructured system that some system information is unknown in advanced.
4. *Contribution 4*; An interaction optimization procedure for agents in unstructured systems, which enables to logically leverage the capabilities of agents and yield a relatively superior system in performing complicated tasks

Table 7.1 shows the relationship of each chapter to the research questions and original contribution of the study. By having CCP-CPS, which utilizes the CPS scheme, it is possible to have a collaborative system which composes of multiple agents (i.e., humans, a mobile robot, sensors, IoT/IoS) in the system, address RQ_1. The CCP-ED is the detailed operation of each agent in the system addressing RQ_2. The information obtains from CCP-CPS and CCP-ED can be most utilized by using D-AS, which addressing RQ_3. Lastly, once the system can localize the unhealthy plants, CRP-H is used to treat plants in the locations effectively, addressing RQ_4.

Table 7.1. Relationship between research questions, contributions, and dissertation structure

Research Question		Contributions	Relative Sections
RQ_1	How can we design and develop the CPS framework, which can combine algorithms, sensors, robots, humans, and other agents to work effectively and facilitate real-time communications for the greenhouse system?	Contribution 1 and 3	3.1 Collaborative Control Protocol for Cyber-Physical System (CCP-CPS) design 3.1.1 CPS framework for PA 3.1.2 CCP-CPS design 3.1.3 Collaborative control theory in CCP-CPS
RQ_2	How can we design the cyber collaborative protocol for the agents in a greenhouse under the CPS framework to perform their job to maximize the system performance, reflect real-time characteristics of plants, and utilize available time most effectively, even optimally, for the earliest detection of stress in crops?	Contribution 2 and 4.	4.2 Collaborative Control Protocol for Early Detection of Stress in Plants (CCP-ED) 4.2.1 Collaboration Requirement Planning (CRP) 4.2.2 Error Prevention and Conflict Resolution (EPCR) 4.2.3 Elements in CCP-ED 4.2.4 CCP-ED design
RQ_3	How can we effectively utilize the new information found during the inspection process to provide the optimal collaborative interaction procedures between agents to maximize information gains during the monitoring process?	Contribution 3 and 4	5.1 Type of system errors in AS 5.2 Monitoring Profit 5.2.1 Overage Cost 5.2.2 Underage Cost 5.3 Optimal Balancing 5.3.1 Dynamic Adaptive Search Algorithm 5.3.2 Propagation Probability Map Model
RQ_4	How can we develop a protocol which effectively and collaboratively manages agents to treat the stress plants that may require a specific type of agent or interaction among multiple agents?	Contribution 2 and 4	6.1 Cyber Collaborative for Plant Treatment (C2T) 6.2 Cyber Collaborative Greenhouse System Architecture 6.3 CRP-H Protocol design 6.4 Human Role and Human-in-the-loop Design for C2T

7.2 Limitation and Future Research Directions

PA now gains more attention to becoming a crucial part of solving the food security issue. The existing knowledge is not sufficient to address challenges and issues in PA. Therefore, this study developed concepts, methodologies, theories, and applications that can be applied to improve and optimize the operation system in PA. To this end, the following research directions are recommended to strengthen the work and investigate limitations of the study in the optimization in PA.

1. *Unexpected events or requests to the system*: The monitoring system of greenhouse crops can be requested to deal with unexpected events or requests such as add the additional location to the existing plan, re-inspect the previous locations, or delay from the impaction process. Future research can be the designing of protocols and algorithms to respond to the requests.
2. *Utilizing historical data*: During the monitoring and inspection process, massive data are collected by sensors. The data can be analyzed and provide meaningful insight. The insights are, for example, the change in directions of the disease's propagation, factors that impact the disease, and the states of stress in the plant.
3. *Integration of learning algorithms*: As mentioned earlier, the massive data collected every day are very valuable and can be used for training the agents to become more efficient in performing tasks. Researchers can consider problems about how to integrate the learning algorithm to the monitoring system.
4. *Additional system agents*: Two sub-directions can be considered in this direction; adding the additional type of agents or adding the same type of agent. Adding more types of agents can improve the agent system performance. The additional type of agent would be drones, field workers, or other types of robots. As the different type of agents has different strength (and weaknesses), the collaboration of agent would strengthen the system performance and capability. On the other hand, the more significant number of agents in the system will lead to a higher complexity of the system as well. Therefore, the more complex protocol which can deal with the multi-agent system is required. Another direction is to add the existing type of agent to the system. Researchers can consider a larger scale of problems, such as a greenhouse where multiple robots are needed. With multiple robots, it can be considered

as another variation of vehicle routing problem, the vehicle routing problem for service operations. Developing a protocol for multi-robot working for a large greenhouse will be an interesting research question to the researchers.

5. *Conflicts and errors*: As conflicts and errors can happen in every system, the system which can resolve conflicts and prevent errors is considered as an efficient system. In the study, some types of conflicts and errors are considered. The future research should address issues about the resolution and prevention of the conflicts and errors in the monitoring system, for example, the delay of the information or inconsistency of the information received.

REFERENCES

Alberts, D. S. (2014). Systems Agility Quotient (AQ). *INSIGHT, 17*(2), 10-13. doi:10.1002/inst.201417210

Albright, L. D. (2002). Controlling greenhouse environments. *Acta Horticulturae, 578*(578), 47-54. doi:10.17660/ActaHortic.2002.578.4

Andersson, J., Jörnsten, K., Nonås, S. L., Sandal, L., & Ubøe, J. (2013). A maximum entropy approach to the newsvendor problem with partial information. *European Journal of Operational Research, 228*(1), 190-200. doi:10.1016/j.ejor.2013.01.031

Anussornnitisarn, P., Nof, S., & Etzion, O. (2005). Decentralized control of cooperative and autonomous agents for solving the distributed resource allocation problem. *Int. J. Prod. Econ., 98*(2), 114-128. doi:10.1016/j.ijpe.2004.05.015

Astrand, B., & Baerveldt, A.-J. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots, 13*(1), 21-35. doi:10.1023/A:1015674004201

Bac, C. W., Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888-911. doi:10.1002/rob.21525

Bailey, B. (2006). Natural and mechanical greenhouse climate control. *Natural and mechanical greenhouse climate control*(710), 43-53.

Balliu, A., Sallaku, G., & Nasto, T. (2017). Nursery management practices influence the quality of vegetable seedlings. *Italus Hortus, 24*, 39-52.

Barbati, M., Bruno, G., & Genovese, A. (2012). Applications of agent-based models for optimization problems: A literature review. *Expert Systems With Applications, 39*(5), 6020-6028. doi:10.1016/j.eswa.2011.12.015

Bechar, A., Meyer, J., & Edan, Y. (2009). An objective function to evaluate performance of human-robot collaboration in target recognition tasks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 39*(6), 611-620. doi:10.1109/TSMCC.2009.2020174

Bechar, A., Nof, S. Y., & Wachs, J. P. (2014). Precision collaboration and advanced integration using laser systems and techniques. In S. Y. Nof, A. M. Weiner, & G. J. Cheng (Eds.), *Laser and Photonic Systems: Design and Integration* (pp. 165-200): CRC Press.

Bechar, A., Nof, S. Y., & Wachs, J. P. (2015). A review and framework of laser-based collaboration support. *Annual Reviews in Control, 39*, 30-45. doi:10.1016/j.arcontrol.2015.03.003

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering, 149*(C), 94-111. doi:10.1016/j.biosystemseng.2016.06.014

Bechar, A., & Vigneault, C. (2017). Agricultural robots for field operations. Part 2: Operations and systems. *Biosystems Engineering, 153*, 110-128. doi:10.1016/j.biosystemseng.2016.11.004

Bechar, A., Wachs, J., Lumkes, J., & Nof, S. (2012). *Developing a human-robot collaborative system for precision agricultural tasks.* Paper presented at the Eleventh international conference precision agriculture.

Belforte, G., Deboli, R., Gay, P., Piccarolo, P., & Ricauda Aimonino, D. (2006). Robot design and testing for greenhouse applications. *Biosystems Engineering, 95*(3), 309-321. doi:10.1016/j.biosystemseng.2006.07.004

Bellmore, M., & Nemhauser, G. L. (1968). The traveling salesman problem: A survey. *Operations Research, 16*(3), 538-558. doi:10.1287/opre.16.3.538

Billari, F. C., Fent, T., Prskawetz, A., & Scheffran, J. (2006). *Agent-Based Computational Modelling Applications in Demography, Social, Economic and Environmental Sciences* (1st ed. 2006. ed.): Heidelberg : Physica-Verlag HD : Imprint: Physica.

Biradar, H. B., & Shabadi, L. (2017). Review on IOT based multidisciplinary models for smart farming. In (Vol. 2018-, pp. 1923-1926): IEEE.

Boussard, M., & Miura, J. (2013). Observation planning for object search by a mobile robot with uncertain recognition. In (Vol. 194, pp. 101-110): Springer Verlag.

Brogårdh, T. (2007). Present and future robot control development—An industrial perspective. *Annual Reviews in Control, 31*(1), 69-79. doi:10.1016/j.arcontrol.2007.01.002

Cachon, G. P. (2003). Supply chain coordination with contracts. *Handbooks in Operations Research and Management Science, 11*(C), 227-339. doi:10.1016/S0927-0507(03)11006-7

Cappelleri, D. J., & McArthur, D. R. (2019). Unmanned aerial vehicles capable of environmental interaction. In: Google Patents.

Ceroni, J. A., & Nof, S. Y. (1999). Robotics terminology. In S. Y. Nof (Ed.), *Handbook of industrial robotics*: John Wiley.

Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., & Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research, 36*(10), 1045-1052. doi:10.1177/0278364917720510

Chen, X. W., & Nof, S. Y. (2012). Constraint-based conflict and error management. *Engineering Optimization, 44*(7), 821-841. doi:10.1080/0305215X.2011.613466

Cimino, D., Ferrero, A., Queirolo, L., Bellotti, F., Berta, R., & De Gloria, A. (2017). A low-cost, open-source cyber physical system for automated, remotely controlled precision agriculture. In (Vol. 409, pp. 191-203): Springer Verlag.

Cowling, P., Ouelhadj, D., & Petrovic, S. (2003). A multi-agent architecture for dynamic scheduling of steel hot rolling. *Journal of Intelligent Manufacturing, 14*(5), 457-470. doi:10.1023/A:1025701325275

Dar, I., Edan, Y., & Bechar, A. (2011). American society of agricultural and biological engineers annual international meeting 2011, ASABE 2011. In (Vol. 1, pp. 288-302): American Society of Agricultural and Biological Engineers.

Deb, S., Fong, S., Tian, Z., Wong, R., Mohammed, S., & Fiaidhi, J. (2016). Finding approximate solutions of NP-hard optimization and TSP problems using elephant search algorithm. *An International Journal of High-Performance Computer Design, Analysis, and Use, 72*(10), 3960-3992. doi:10.1007/s11227-016-1739-2

Diker, K., & Bausch, W. C. (2003). Radiometric field measurements of maize for estimating soil and plant nitrogen. *Biosystems Engineering, 86*(4), 411-420. doi:10.1016/j.biosystemseng.2003.08.016

Dogar, M., Koval, M., Tallavajhula, A., & Srinivasa, S. (2014). Object search by manipulation. *Autonomous Robots, 36*(1-2), 153-167. doi:10.1007/s10514-013-9372-x

Dong, X., Vuran, M. C., & Irmak, S. (2013). Autonomous precision agriculture through integration of wireless underground sensor networks with center pivot irrigation systems. *Ad Hoc Networks, 11*(7), 1975-1987. doi:10.1016/j.adhoc.2012.06.012

Drechsel, P., Heffer, P., Magen, H., Mikkelsen, R., Singh, H., & Wichelns, D. (2015). Managing water and nutrients to ensure global food security, while sustaining ecosystem services. *IDEAS Working Paper Series from RePEc*.

Dusadeerungsikul, P. O., He, X., Sreeram, M., & Nof, S. Y. (2020). Multi-agent system optimization in factories of the future: Cyber collaborative warehouse case study.

Dusadeerungsikul, P. O., & Nof, S. Y. (2019). A collaborative control protocol for agricultural robot routing with online adaptation. *Computers & Industrial Engineering, 135*, 456-466. doi:10.1016/j.cie.2019.06.037

Dusadeerungsikul, P. O., Nof, S. Y., & Bechar, A. (2018). *Detecting stresses in crops early by collaborative robot-sensors-human system automation.* Paper presented at the IISE Annual Conference and Expo 2018, Orlando, FL.

Dusadeerungsikul, P. O., Nof, S. Y., & Bechar, A. (2020). Smart collaborative robotics and CPS for smart and precision agriculture. In A. Castrignanò (Ed.), *Agricultural Internet of Things and Decision Support for Smart Farming*.

Dusadeerungsikul, P. O., Nof, S. Y., Bechar, A., & Tao, Y. (2019). *Collaborative control protocol for agricultural cyber-physical system.* Paper presented at the 25th International Conference on Production Research 2019, Chicago, IL.

Dusadeerungsikul, P. O., Sreeram, M., He, X., Nair, A., Ramani, K., Quinn, A. J., & Nof, S. Y. (2019). Collaboration requirement planning protocol for HUB-CI in factories of the future.

Edan, Y., Han, S., & Kondo, N. (2009). Automation in agriculture. In *Springer handbook of automation* (pp. 1095-1128): Springer.

Edan, Y., & Miles, G. E. (1994). Systems engineering of agricultural robot design. *IEEE Transactions on Systems, Man, and Cybernetics, 24*(8), 1259-1265. doi:10.1109/21.299707

Essahafi, M., & Lafkih, M. A. (2018). Comparison between two adaptive controllers applied to greenhouse climate monitoring. *International Journal of Advanced Computer Science and Applications, 9*(1), 341-346. doi:10.14569/IJACSA.2018.090147

Ferentinos, K. P., Katsoulas, N., Tzounis, A., Bartzanas, T., & Kittas, C. (2017). Wireless sensor networks for greenhouse climate and plant condition assessment. *Biosystems Engineering, 153*, 70-81. doi:10.1016/j.biosystemseng.2016.11.005

Finkelshtain, R., Bechar, A., Yovel, Y., & Kósa, G. (2017). Investigation and analysis of an ultrasonic sensor for specific yield assessment and greenhouse features identification. *An International Journal on Advances in Precision Agriculture, 18*(6), 916-931. doi:10.1007/s11119-016-9479-0

Gay, P., Piccarolo, P., Aimonino, D. R., & Deboli, R. (2008). International Conference, Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agro-Food Systems, 15-17 September 2008, Ragusa, Italy. In. Ragusa, Italy: Ragusa Safety Health Welfare.

Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science, 327*(5967), 828. doi:10.1126/science.1183899

Goap, A., Sharma, D., Shukla, A. K., & Rama Krishna, C. (2018). An IoT based smart irrigation management system using Machine learning and open source technologies. *Computers and Electronics in Agriculture, 155*, 41-49. doi:10.1016/j.compag.2018.09.040

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture, 116*, 8-19. doi:10.1016/j.compag.2015.05.021

Gonzalez-Amarillo, C. A., Corrales-Munoz, J. C., Mendoza-Moreno, M. A., Gonzalez Amarillo, A. M., Hussein, A. F., Arunkumar, N., & Ramirez-Gonzalez, G. (2018). An IoT-Based traceability system for greenhouse seedling crops. *IEEE Access, 6*, 67528-67535. doi:10.1109/ACCESS.2018.2877293

Gueroui, A., & Labraoui, N. (2015). Concepts and evolution of research in the field of wireless sensor networks. *arXiv.org, 7*(1). doi:10.5121/ijcnc.2015.7106

Guo, P., Dusadeerungsikul, P. O., & Nof, S. Y. (2018). Agricultural cyber physical system collaboration for greenhouse stress management. *Computers and Electronics in Agriculture, 150*, 439-454. doi:10.1016/j.compag.2018.05.022

Huang, C.-Y., Ceroni, J. A., & Nof, S. Y. (2000). Agility of networked enterprises — parallelism, error recovery and conflict resolution. *Computers in Industry, 42*(2), 275-287. doi:10.1016/S0166-3615(99)00076-7

Huang, G., Rao, P. S., Wu, M., Wadhawan, M., Nof, S. Y., Chandrasegaran, S., . . . Quinn, A. J. (2020). *VIPO: Spatial-Visual Programming with Functions for Robot-IoT Workflows.* Paper presented at the ACM CHI 2020.

Ishibashi, M., Iida, M., Suguri, M., & Masuda, R. (2013). Remote monitoring of agricultural Robot using web application. *IFAC Proceedings Volumes, 46*(18), 138-142. doi:10.3182/20130828-2-SF-3019.00047

Jeong, H. Y., Song, B. D., & Lee, S. (2019). Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics, 214*, 220-233. doi:10.1016/j.ijpe.2019.01.010

Jun, S., Lee, S., & Chun, H. (2019). Learning dispatching rules using random forest in flexible job shop scheduling problems. *International Journal of Production Research, 57*(10), 3290-3310. doi:10.1080/00207543.2019.1581954

Keicher, R., & Seufert, H. (2000). Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture, 25*(1), 169-194. doi:10.1016/S0168-1699(99)00062-9

Khamis, A., Hussein, A., & Elmogy, A. (2015). Multi-robot task allocation: A review of the state-of-the-art. *Studies in Computational Intelligence, 604*, 31-51. doi:10.1007/978-3-319-18299-5_2

Khosro Anjom, F., Vougioukas, S. G., & Slaughter, D. C. (2018). Development and application of a strawberry yield-monitoring picking cart. *Computers and Electronics in Agriculture, 155*, 400-411. doi:10.1016/j.compag.2018.10.038

Khouja, M. (1999). The single-period (news-vendor) problem: literature review and suggestions for future research. *Omega, 27*(5), 537-553. doi:10.1016/S0305-0483(99)00017-1

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M. A., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering, 129*, 14-30. doi:10.1016/j.cie.2019.01.020

Ko, H. S., & Nof, S. Y. (2008). Modeling task administration protocols for human and robot e-Workers. *Handbook of Digital Human Modeling*.

Ko, H. S., & Nof, S. Y. (2010). Design of protocols for task administration in collaborative production systems. *International Journal of Computers, Communications and Control, 5*(1), 91-105. doi:10.15837/ijccc.2010.1.2466

Ko, H. S., & Nof, S. Y. (2012). Design and application of task administration protocols for collaborative production and service systems. *International Journal of Production Economics, 135*(1), 177-189. doi:10.1016/j.ijpe.2011.06.005

Kondo, N., & Ting, K. C. (1998). *Robotics for bioproduction systems*. Saint Joseph, MI: American Society of Agricultural Engineers.

Kondō, N., Monta, M., & Noguchi, N. (2011). *Agricultural robots : mechanisms and practice* (English ed. ed.). Sakyo-ku, Kyoto : Melbourne, Vic., Australia: Kyoto University Press ; Trans Pacific Press.

Kulich, M., Preučil, L., & Bront, J. J. M. (2014). Single robot search for a stationary object in an unknown environment. In (pp. 5830-5835): Institute of Electrical and Electronics Engineers Inc.

Kunz, C., Weber, J., Peteinatos, G., Sökefeld, M., & Gerhards, R. (2018). Camera steered mechanical weed control in sugar beet, maize and soybean. *An International Journal on Advances in Precision Agriculture, 19*(4), 708-720. doi:10.1007/s11119-017-9551-4

Liao, M.-S., Chen, S.-F., Chou, C.-Y., Chen, H.-Y., Yeh, S.-H., Chang, Y.-C., & Jiang, J.-A. (2017). On precisely relating the growth of Phalaenopsis leaves to greenhouse environmental factors by using an IoT-based monitoring system. *Computers and Electronics in Agriculture, 136*, 125-139. doi:10.1016/j.compag.2017.03.003

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research, 21*(2), 498-516.

Liu, J., Li, C., Yang, F., Wan, H., & Reha, U. (2011). *Production planning for semiconductor manufacturing via simulation optimization.* Paper presented at the Proceedings of the 2011 Winter Simulation Conference (WSC).

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A literature review Journal of Computer and Communications.

Mahlein, A.-K. (2016). Plant disease detection by imaging sensors - parallels and specific demands for precision agriculture and plant phenotyping. *Plant disease, 100*(2), 241. doi:10.1094/PDIS-03-15-0340-FE

Makarenko, A. A., Williams, S. B., Bourgault, F., & Durrant-Whyte, H. F. (2002). *An experiment in integrated exploration.* Paper presented at the IEEE International Conference on Intelligent Robots and Systems.

Malagi, V. P., & Rangarajan, K. (2016). Multi-object tracking in aerial image sequences using aerial tracking learning and detection algorithm. *Defence Science Journal, 66*(2), 122-129.

Manfreda, S., McCabe, M. F., Miller, P. E., Lucas, R., Pajuelo Madrigal, V., Mallinis, G., . . . Ruiz Perez, G. (2018). On the use of unmanned aerial systems for environmental monitoring. *10*(4). doi:10.3390/rs10040641

McIntosh, P. (2015). Agricultural robotics: Here come the agribots. *Maximum Yield news letter.*

Min Hyuc, K., Beom-Sahng, R., Kyoung Chul, K., Suprem, A., & Mahalik, N. P. (2015). Autonomous greenhouse mobile robot driving strategies from system integration perspective: Review and application. *IEEE/ASME Transactions on Mechatronics, 20*(4), 1705-1716. doi:10.1109/TMECH.2014.2350433

Mitsuhashi, T., Yamazaki, A., & Shichishima, T. (1994). *Automation of plant factory.* Paper presented at the Proc. 4th SHITA symposium, Tokyo, Japan.

Miura, J., Kadekawa, S., Chikaarashi, K., & Sugiyama, J. (2016). Human-robot collaborative remote object search. In (Vol. 302, pp. 979-991): Springer Verlag.

Moghaddam, M., & Nof, S. Y. (2014). Combined demand and capacity sharing with best matching decisions in enterprise collaboration. *International Journal of Production Economics, 148*, 93-109. doi:10.1016/j.ijpe.2013.11.015

Moghaddam, M., & Nof, S. Y. (2018). Collaborative service-component integration in cloud manufacturing. *International Journal of Production Research, 56*(1-2), 677-691. doi:10.1080/00207543.2017.1374574

Morimoto, E. (2018). *What is cyber-physical system driven agriculture?-Redesign of big data for outstanding farmer management.* Paper presented at the 2018 ASABE Annual International Meeting.

Mu, L., Liu, H., Cui, Y., Fu, L., & Gejima, Y. (2018). Mechanized technologies for scaffolding cultivation in the kiwifruit industry: A review. *Information Processing in Agriculture, 5*(4), 401-410. doi:10.1016/j.inpa.2018.07.005

Nagasaka, Y., Zhang, Q., Grift, T. E., Kanetani, Y., Umeda, N., & Kokuryu, T. (2004). *An autonomous field watching-dog robot for information collection in agricultural fields.* Paper presented at the ASAE Annual International Meeting 2004.

Nagata, Y., & Kobayashi, S. (2013). A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *INFORMS Journal on Computing, 25*(2), 346-363. doi:10.1287/ijoc.1120.0506

Nie, J., Sun, R. Z., & Li, X. H. (2014). A precision agriculture architecture with Cyber-Physical Systems design technology. *Applied Mechanics and Materials, 543-547*(Vehicle, Mechatronics and Information Technologies II), 1567-1570. doi:10.4028/www.scientific.net/AMM.543-547.1567

Nof, S. Y. (2007). Collaborative control theory for e-Work, e-Production, and e-Service. *Annual Reviews in Control, 31*(2), 281-292. doi:10.1016/j.arcontrol.2007.08.002

Nof, S. Y. (2015). Innovations in precision collaboration that matter to precision agriculture Robotics. In: BARD Workshop.

Nof, S. Y., Ceroni, J., Jeong, W., & Moghaddam, M. (2015). *Revolutionizing Collaboration through e-Work, e-Business, and e-Service* (1st ed. 2015. ed.): Berlin, Heidelberg : Springer Berlin Heidelberg : Imprint: Springer.

Parker, L. E. (1999). Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, 5*(1), 5-19. doi:10.1080/10798587.1999.10750747

Pedersen, S., Fountas, S., Have, H., & Blackmore, B. (2006). Agricultural robots—system analysis and economic feasibility. *An International Journal on Advances in Precision Agriculture, 7*(4), 295-308. doi:10.1007/s11119-006-9014-9

Perakis, G., & Roels, G. (2008). Regret in the newsvendor model with partial information. *Operations Research, 56*(1), 188-203. doi:10.1287/opre.1070.0486

Perez-Exposito, J. P., Fernandez-Carames, T. M., Fraga-Lamas, P., & Castedo, L. (2017). 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). In (Vol. 2018-, pp. 662-669): IEEE.

Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research, 63*(3), 337-370. doi:10.1007/BF02125403

Putnik, G., Sluga, A., Elmaraghy, H., Teti, R., Koren, Y., Tolio, T., & Hon, B. (2013). Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. *CIRP Annals - Manufacturing Technology, 62*(2), 751-774. doi:10.1016/j.cirp.2013.05.002

Qin, Y., Wang, R., Vakharia, A. J., Chen, Y., & Seref, M. M. H. (2011). The newsvendor problem: Review and directions for future research. *European Journal of Operational Research, 213*(2), 361-374. doi:10.1016/j.ejor.2010.11.024

Rajan, V., & Nof, S. (1996). Cooperation Requirements Planning (CRP) for multiprocessors: Optimal assignment and execution planning. *Theory and Applications, 15*(4), 419-435. doi:10.1007/BF00437605

Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture, 25*(1-2), 155-167. doi:10.1016/S0168-1699(99)00061-7

Ribaudo, M., Delgado, J., Hansen, L., Livingston, M., Mosheim, R., & Williamson, J. (2012). *Nitrogen Use in U.S. Agriculture: Implications and Management*: Nova Science Publishers, Inc.

Sadhu, C., Abhiram, M. H., Chandan, B., Madhu, G., & Shreedarshan, K. (2013). *Cognitive Learning Enabled Real Time Object Search Robot.* Paper presented at the 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE).

Sai, Z., Fan, Y., Yuliang, T., Lei, X., & Yifong, Z. (2016). Optimized algorithm of sensor node deployment for intelligent agricultural monitoring. *Computers and Electronics in Agriculture, 127*(C), 76-86. doi:10.1016/j.compag.2016.06.001

Sanchez, S. M., & Wan, H. (2015). *Work smarter, not harder: a tutorial on designing and conducting simulation experiments.* Paper presented at the 2015 Winter Simulation Conference (WSC).

Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., & Berman, S. (2016). Robotic disease detection in greenhouses: Combined detection of powdery mildew and tomato spotted wilt virus. *IEEE Robotics and Automation Letters, 1*(1), 354-360. doi:10.1109/LRA.2016.2518214

Schor, N., Berman, S., Dombrovsky, A., Elad, Y., Ignat, T., & Bechar, A. (2017). Development of a robotic detection system for greenhouse pepper plant diseases. *An International Journal on Advances in Precision Agriculture, 18*(3), 394-409. doi:10.1007/s11119-017-9503-z

Silver, E. A. (1998). *Inventory management and production planning and scheduling* (3rd ed. ed.). New York: Wiley.

Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics, 34*(6), 1140-1159. doi:10.1002/rob.21715

Simonton, W. (1990). Automatic geranium stock processing in a robotic workcell. *Trans. ASAE, 33*(6), 2074-2080.

Sjo, K., Galvez Lopez, D., Paul, C., Jensfelt, P., & Kragic, D. (2009). Object search and localization for an indoor mobile robot. *CIT Journal of Computing and Information Technology, 17*(1), 67. doi:10.2498/cit.1001182

Spekken, M., & Bruin, S. (2013). Optimized routing on agricultural fields by minimizing maneuvering and servicing time. *An International Journal on Advances in Precision Agriculture, 14*(2), 224-244. doi:10.1007/s11119-012-9290-5

Sprute, D., Pörtner, A., Rasch, R., Battermann, S., & König, M. (2017). Ambient assisted robot object search. In (Vol. 10461, pp. 112-123): Springer Verlag.

Stark, B., Rider, S., & Chen, Y. (2013). Optimal pest management by networked unmanned cropdusters in precision agriculture: A cyber-physical system approach. *IFAC Proceedings Volumes, 46*(30), 296-302. doi:10.3182/20131120-3-FR-4045.00019

Suzuki, N., Rivero, R. M., Shulaev, V., Blumwald, E., & Mittler, R. (2014). Abiotic and biotic stress combinations. In (Vol. 203).

Thompson, N., Bir, C., Widmar, D., & Mintert, J. (2019). Farmer perceptions of precision agriculture technology benefits. *Journal of Agricultural and Applied Economics, 51*(1), 142-163. doi:10.1017/aae.2018.27

Tkach, I., Edan, Y., & Nof, S. Y. (2017). Multi-sensor task allocation framework for supply networks security using task administration protocols. *International Journal of Production Research, 55*(18), 5202-5224. doi:10.1080/00207543.2017.1286047

Wang, D., Vinson, R., Holmes, M., Seibel, G., Bechar, A., Nof, S., & Tao, Y. (2019). Early detection of tomato spotted wilt virus by hyperspectral imaging and Outlier Removal Auxiliary Classifier Generative Adversarial Nets (OR-AC-GAN). *Sci Rep, 9*(1), 4377. doi:10.1038/s41598-019-40066-y

Weng, Z. K. (2004). Coordinating order quantities between the manufacturer and the buyer: A generalized newsvendor model. *European Journal of Operational Research, 156*(1), 148-161. doi:10.1016/S0377-2217(03)00003-1

Whitin, T. M. (1955). Inventory control and price theory. *Management Science, 2*(1), 61-68. doi:10.1287/mnsc.2.1.61

Wu, D. F., Dai, L. Q., & Dai, H. C. (2013). Application of Internet of Thing technology in ecological-environmental protection of long-distance water diversion project. In (Vol. 353-356, pp. 2563-2566): Trans Tech Publications.

Xiao, R., Tao, Z., & Chen, T. (2012). An analytical approach to the similarities between swarm intelligence and artificial neural network. *Transactions of the Institute of Measurement and Control, 34*(6), 736-745. doi:10.1177/0142331211402903

Xue, J., Zhang, L., & Grift, T. E. (2012). Variable field-of-view machine vision based row guidance of an agricultural robot. *Computers and Electronics in Agriculture, 84*(C), 85-91. doi:10.1016/j.compag.2012.02.009

Ye, Y., He, L., Wang, Z., Jones, D., Hollinger, G. A., Taylor, M. E., & Zhang, Q. (2018). Orchard manoeuvring strategy for a robotic bin-handling machine. *Biosystems Engineering, 169*, 85-103. doi:10.1016/j.biosystemseng.2017.12.005

Zhang, X., Schmidt, L., Schlick, C. M., Reuth, R., & Luczak, H. (2008). A human task-oriented simulation study in autonomous production cells. *International Journal of Production Research, 46*(18), 5013-5041. doi:10.1080/00207540701222768

Zhong, H., & Nof, S. Y. (2015). The dynamic lines of collaboration model: Collaborative disruption response in cyber–physical systems. *Computers & Industrial Engineering, 87*, 370-382. doi:10.1016/j.cie.2015.05.019

Zhong, H., Nof, S. Y., & Berman, S. (2015). Asynchronous cooperation requirement planning with reconfigurable end-effectors. *Robotics and Computer Integrated Manufacturing, 34*, 95-104. doi:10.1016/j.rcim.2014.11.004

Zhong, H., Wachs, J. P., & Nof, S. Y. (2013). A collaborative telerobotics network framework with hand gesture interface and conflict prevention. *International Journal of Production Research, 51*(15), 4443-4463. doi:10.1080/00207543.2012.756591

VITA

PUWADOL DUSADEEERUNGSIKUL

EDUCATION

Purdue University West Lafayette, IN

Ph.D. in Industrial Engineering (*Advisor:* Professor Shimon Y. Nof) 2020

Georgia Institute of Technology Atlanta, GA

Master of Science in Industrial Engineering 2015

Chulalongkorn University Bangkok, Thailand

Bachelor of Engineering, Industrial Engineering, Honors 2011

RESEARCH EXPERIENCE

Researcher, PRISM Center, Purdue University 2016 – *present*

Plant Monitoring System for Precision Agriculture

Lead an interdisciplinary and multi-institutional project funded by the U.S.–Israel Binational Agricultural Research and Development Fund (BARD) grant on developing a human-robot stress monitoring system for food plants by collaborating with agricultural scientists from the Agricultural Research Organization (Israel) and bioengineers from the University of Maryland.

Cyber Collaboration Factories of the Future

Research and manage NSF grant-supported project on developing a Factory of the Future, dealing with new technologies, workforce, and production demand

Complex Systems/System-of-Systems Service Design for Rapid Decision-Making

Managed and conducted a collaborative research project with aerospace engineers to develop a service system responding to system reliability and security disruptions

TEACHING EXPERIENCE

Teaching Assistant, School of Industrial Engineering, Purdue University

IE332 Computing in Industrial Engineering

IE484 Integrated Production Systems II

IE579 Design and Control of Production and Manufacturing Systems

IE590 Structured Engineering Innovation

WORK EXPERIENCE

Industrial Engineer, Menlo Worldwide Logistics 2013 – 2014

Initiated development of improvement projects in Menlo Warehouses and truck delivery systems

Supply Planner, Johnson & Johnson 2011 – 2013

Managed End-to-End supply chain for Carefree Pantiliner products range (both in plant and in sub-contracting manufacturing)

Internship, ExxonMobil 2010

Standardized ExxonMobil Thailand's truck planning process, dealt with over 500 gas stations and improved a planning process by designing a more effective procedure, reducing waste and non-value-added work

PUBLICATIONS

Journal Articles

J2. **Dusadeerungsikul, P. O.**, & Nof, S. Y. (2019). A collaborative control protocol for agricultural robot routing with online adaptation. *Computers & Industrial Engineering*, 135, 456-466.

J1. Guo, P., **Dusadeerungsikul, P. O.**, & Nof, S. Y. (2018). Agricultural cyber-physical system collaboration for greenhouse stress management. *Computers and Electronics in Agriculture*, 150, 439-454.

Under Review

J3. **Dusadeerungsikul, P. O.**, He, X., Sreeram, M., & Nof, S. Y. Multi-agent system optimization in factories of the future: Cyber collaborative warehouse case study.

Book Chapters

B1. **Dusadeerungsikul, P. O.,** Nof, S. Y., & Bechar. A. (2019). Smart collaborative robotics and CPS for smart and precision agriculture. In A. Castrignanò (Ed.), *Agricultural Internet of Things and Decision Support for Smart Farming*.

Conference Proceedings

C3. **Dusadeerungsikul, P. O.,** Sreeram, M., Nair, A., He, X., Ramani, K., Quinn, A., & Nof, S. Y. (2019, August). Collaboration requirement planning protocol for HUB-CI in factories of the future. 25th ICPR Conference, Chicago, IL, *Procedia Manufacturing*.

C2. **Dusadeerungsikul, P. O.**, Nof, S. Y., & Bechar. A., Tao, Y. (2019, August). Collaborative control protocol for agricultural cyber-physical system. 25th ICPR Conference, Chicago, IL, *Procedia Manufacturing*.

C1. **Dusadeerungsikul, P. O.**, Nof, S. Y., & Bechar. A. (2018, May). Detecting stresses in crops early by collaborative robot-sensors-human system automation. IISE Conference 2018, Orlando, FL, *Proceedings of IISE Conference.*