Computers & Industrial Engineering 135 (2019) 456–466

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A collaborative control protocol for agricultural robot routing with online adaptation

Puwadol Oak Dusadeerungsikul*, Shimon Y. Nof

PRISM Center and School of Industrial Engineering, Purdue University, IN, USA

ARTICLE INFO

Keywords:
Collaborative control theory
Collaboration
Greenhouse
Traveling salesmen problem
Sensors
Adaptive search algorithm

ABSTRACT

Stress in crops is one of the major concerns in precision agriculture because it indicates the emergence of disease and damage in plants. Detecting the stress condition of a plant early is critical. A system that can monitor the condition of plants is a desirable solution. In this work, Collaborative Control Theory is utilized to construct a new system, ARS (agricultural robotic system) which synchronizes humans, a mobile robot, and a variable set of sensors to effectively perform the monitoring and detection tasks. A key protocol for that system, which combines routing algorithm, adaptive search algorithm, and collaboration control framework has been developed and validated, and is presented in this article. By using greenhouse as a case study structure, the protocol routes a robot to visit the sampled locations by using a genetic algorithm. In addition, the search algorithm can be guided by the predictive characteristics of the crops' stress, which can spread to other plants according to sunlight, airflow direction, and other known conditions. Based on simulation experiments, the results indicate with statistical significance that (1) the routing algorithm increases the number of successful detections of existing stressed plants by 45.77% compared to monitoring without this routing algorithm. (2) The adaptive search algorithm improves the number of successful detections of stressed plants by 71.88% compared to a system without the adaptive search algorithm. (3) The new protocol developed in this research yields the highest overall robotic efficiency, compared with a system without collaborative control framework.

1. Introduction

Ability to detect stresses in crops in early stages is crucial in precision agriculture since it can help prevent plants from developing damaging disease. Diseases, insects, and weeds damage approximately 40% of food production in the world (Oerke & Dehne, 2004). So far, the techniques that are applied to handle this situation are not effective. Lacking an effective system to monitor stress conditions of crops leads to ambiguous and wrong decisions of implementing over or under amounts of water, fertilizers, and pesticide, resulting in wasted effort, money and time. Therefore, an effective system which can detect stresses in plants early enough to prevent them from uncontrolled spreading of disease and damages is widely considered to be a beneficial solution.

Currently, agricultural processes involve high labor-intensive work. Workers perform monitoring, inspecting, farming, and harvesting tasks (Khan, Martin, & Hardiman, 2004). For inspection tasks, workers walk into the plot and sample locations where to inspect. Each worker usually walks approximately 20 km per day to perform the task. Because of limitations of workers and available time, however, the inspecting process is inaccurate, and the detection of stresses is frequently too late.

The improvement of farming equipment will drastically change the way farmers work (Grad et al., 2014). Because of the improvement, farmers can obtain relatively higher productivity at lower cost (Mare & Mare, 2015). Therefore, researchers have focused on better techniques and equipment to improve the productivity of agricultural production systems. Researchers tend to apply robots to work for people in agriculture tasks (Edan, 1999; Edan, Han, & Kondo, 2009). The new agricultural machinery is expected to become a major contributor to the precision of agriculture system (Eaton, Katupitiya, & Siew, 2008).

Nowadays, agricultural robotics is typically used for weeding, spraying, irrigation (Cheein and Carelli, 2013). A survey of agriculture automation has been done by Edan, Han, and Kondo (2009). Advancement of precision agriculture, however, is not as fast as predicted (McBratney, Whelan, Ancev, & Bouma, 2004). For precision agriculture, certain research problems need to be explored. Since an agricultural system has many factors to consider, such as weather for each day, humidity, stress in plant, quality of soil, to name but a few, it presents a challenge for researchers in diverse disciplines. Therefore,

* Corresponding author.
E-mail addresses: pdusadee@purdue.edu (P.O. Dusadeerungsikul), nof@purdue.edu (S.Y. Nof).

https://doi.org/10.1016/j.cie.2019.06.037
Received 27 August 2018; Received in revised form 22 March 2019; Accepted 16 June 2019
Available online 17 June 2019
0360-8352/

Nomenclature

Variables and Abbreviations

AD	association and dissociation
ARS	agricultural robotic system
$ARS_{Single}(t)$	single ARS' agent at time t
$ARS_{Group}(t)$	group of ARS' agents at time t
AS	adaptive Search
Avg.	average
BM	best matching
C	conflict
c_{ij}	travel time from i to j
CCP-ED	collaborative control protocol for early detection of stress in plants
CCT	collaborative Control Theory
CFT	collaborative fault tolerance
CLM	collaborative life-cycle manufacturing
CRM	collaboration requirement matrix
CRP	collaboration requirement planning
CTR	collaborative telerobotics
CVC	collaborative visualization and comprehension
n	total number of infected locations found
d_i	number of infected locations found during operation at location i
E	error
ELOCC	emergent lines of collaboration and command
EPCR	error prevention and conflict resolution
EWP	e-Work parallelism
GA	genetic algorithm
KISS	keep it simple, system
L_i	i^{th} level of adaptive search
$M, \bar{M}$	partition of set of integer i
m, n	number of sampled locations in the greenhouse
$N_r(t)$	set of constraints r at time t
ORE	overall robotic effectiveness
P	performance
p_j	inspection time at location i
p_j'	extra processing time at node j
q_j	searching time at location i
q_j'	extra searching time at node j
R_i	agent or group of agents i
S	success
$S_{ARS_{Single}}(t)$	state of single ARS' agent at time t
$S_{ARS_{Group}}(t)$	state of group of ARS' agents at time t
SD	standard deviation
s_i	number of inspected locations at location i
T	total available time
T'	total processing time
T_i	task i
TSP	traveling salesman problem
U	utilization
x_{ij}	ARS traverse from i to j
y_i	ARS perform adaptive search at location i
α_j	probability to fail to search at node j
β_j	probability to fail to inspect at node j

precision agriculture will consist of various advanced technologies, such as information system, operation research, machinery and robotics, and information management (Gebbers & Adamchuk, 2010).

The research reported here focuses on developing and validating a protocol to monitor effectively stress in greenhouse plants. It is part of a larger project, seeking to benefit from human-robot stress monitoring for early detection of stress, thus preventing its propagation and conversion to disease (see Acknowledgement at the end). The major observation is that existing protocols lack ability to collaborate with parties that comprise the monitoring system. Moreover, current protocols have not relied on captured real characteristics of the monitored plants. Therefore, this work focuses on the following aspects of monitoring crops in greenhouses: (1) Establishing a framework for monitoring plants by a robotic system. (2) Collaboration between three parties; human (one or more), a mobile robot, and sensors. (3) Routing algorithm and search algorithm that can capture real characteristics of plants.

Section 2 summarizes previous research in the above areas. The design of the collaborative control protocol, including Collaborative Control Theory and adaptive search algorithm, is explained in Section 3. Section 4 describes experiments of greenhouse robotic simulation. Conclusions and future work direction are discussed in Section 5.

2. Background

2.1. Robotics in the agriculture

Research on agricultural robotics for monitoring conditions of crops is not a new idea. The main focus of researchers has been to find methods to optimize the monitoring and detecting processes. Table 1 summarizes previous works. In the first place, robots were used to help farmers performed routine operations such as fertilizing, irrigation, and harvesting (Reid, Zhang, Noguchi, & Dickson, 2000; Keicher & Seufert, 2000). After that, robots in agriculture were used to improve, enhance, and control activities such as monitoring, inspection, and detection diseases. Astrand and Baerveldt (2002) used the autonomous mobile robot to control weed in open field. In addition, robots and remote sensing are used for gathering information about status of crops and environment conditions (Diker & Bausch, 2003; Nagasaka et al., 2004). Bochtis et al. (2011) designed a system which can create plan for monitoring crops. Xue, Zhang, and Grift (2012) developed vision-based guidance system to navigate a robot in cornfield. Spekken and Bruin (2013) designed a method to minimize maneuvering and servicing time for agricultural machine. The research about self-driven robots has been developed since the anticipation of future aging society and population decline (Iida et al., 2013). After that, Ishibashi, Iida, Suguri, and Masuda (2013) presented an idea of using web application to create remote monitoring robotic system. In addition, a system that automatically navigates a robot to spaying pesticide was developed by Ko et al., in 2015. In 2016, Zou et al. gave an idea of how to deploy sensors in open field to monitor crops intelligently which is a different approach to monitor crops. Lastly, Chebrolu et al. in (2017), used a robot with an RGB-D sensor to capture information of beet fields.

All of researchers' work aims to enhance the ability to control and monitor crops in fields in different perspectives. The development of robotic machinery in agriculture moves us closer to precision agriculture. They optimize the way workers work in fields by using robots, sensors, and machinery. They also improve productivity, increase accuracy and reduce waste of system, helping farmers achieve better crops yield. An important characteristic of crops that can contract disease to nearby locations, however, has not yet been considered. Moreover, collaboration among parties in agriculture monitoring system is also an important part that can expedite and improve systems' performance. Lastly, handling conflicts and errors in agriculture robotic systems has also not yet been considered. Therefore, this article develops a collaborative control protocol that takes the mentioned characteristics of plants' stress into consideration. Based on such knowledge, routing and adaptive search can be achieved with more advanced and effective results.

Table 1
Recent research on agricultural robotics for crop monitoring (sample).

Agricultural Monitoring Purpose (Source)	Focus of Approach	Functions Included			
		Operation Time at Node	Routing & Adaptive Search	Conflicts & Errors Prevention	Cyber-Supported Collaboration
Weed Control (Astrand & Baerveldt, 2002)	An autonomous mobile robot with vision systems works in outdoor environment.	Partial	No	Partial	No
Maize crops (Diker & Bausch, 2003)	Using remote sensing to estimate in-season plant and fertilizer in soil.	Partial	No	No	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	Partial	No	No	No
Weed monitoring (Bochtis et al., 2011)	A system which can plan and manage the monitoring task of a field.	Partial	No	No	No
Corn fields (Xue et al., 2012)	A variable field-of-view machine vision method guides a robot to move between fields.	No	No	Yes	No
Field operations (Spekken & Bruin, 2013)	Minimize non-productive movements and maneuvering by agricultural machinery.	Yes	No	No	No
Agricultural Robot (Ishibashi et al., 2013)	Remote web-based monitoring system to order and control a robot in the field.	No	No	No	Partial
Greenhouse pesticide spraying (Ko, Ryuh, Kim, Suprem, & Mahalik, 2015)	Driving strategies for autonomous agricultural mobile robot.	No	No	Yes	No
Agricultural monitoring (Zou, Yang, Tang, Xiao, & Zhao, 2016)	Algorithm to minimize sensor deployment nodes for intelligent monitoring.	No	No	No	No
Sugar beet fields (Chebrolu et al., 2017)	A robot recording plant classification, localization and mapping carrying a multi-spectral camera and an RGB-D sensor.	No	Partial	Yes	No
Greenhouse crops (This article)	Collaborative control protocol and adaptive algorithms to monitor stress of individual plants.	Yes	Yes	Yes	Yes

2.2. Collaborative control theory

Collaborative Control Theory (CCT) includes principles and framework for engineers to design a complex system with multiple agents (Nof, 2007). With a good design, better performance (relatively more reliable, shorter time, and cost-effective) can be achieved. Collaboration among parties in system will enable design of effective e-Work system (Nof, Ceroni, Jeong, & Moghaddam, 2015). Moreover, an important ability which enables better performance of e-Work is sharing information among agents with collaboration protocol. All agents in a particular system will have the same mutual goal as to improve the system and its performance. In any system, it can have one or combination of the following collaborations: mandatory, optional, and concurrent collaboration. There are nine CCT design principles that can be applied in this research: collaboration requirement planning (CRP), e-Work parallelism (EWP), keep it simple, system (KISS), error prevention and conflict resolution (EPCR), collaborative fault tolerance (CFT), association and dissociation (AD), emergent lines of collaboration and command (ELOCC), best matching (BM), and collaborative visualization and comprehension (CVC).

CCT has been applied extensively in various aspects to design multi-agent complex systems. For example, Zhong, Wachs, and Nof (2013) applied the theory to create CTR, telerobotics framework for collaborative life-cycle manufacturing (CLM). The CTR framework considers a protocol which handles conflicts and errors in human-robot system for nuclear application. Asynchronous cooperation requirement planning also applied CCT for having better robotic performance in agriculture task (Zhong, Nof, & Berman, 2015). In addition, best matching protocol in CCT can be used in assembling e-work networks to better utilize existing equipment, parts, and suppliers (Velasquez & Nof, 2009). Moreover, CCT best matching allocation protocols for capacity and demand sharing optimized overall yield for supply network (Moghaddam, and Nof, 2014; 2016). Additionally, resources sharing-based framework for cyber-physical system used CCT to enable flexibility of modeling and control (Nayak, Levalle, Lee, & Nof, 2016).

Based on the wide range of CCT applications, it can be applied to precision agriculture process to obtain better system performance and efficiency. CCT principles are used to develop crops monitoring process. As shown in this article, it helps improve efficiency, by significantly saving time and cutting cost.

2.3. Traveling salesman problem for robotic monitoring and inspection

In general, if we need to develop a path for visiting every desired node once, 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 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 TSP because it can provide an acceptable solution within limited time, even though the optimal solution is not guaranteed by solving the problem with a heuristic approach. (Potvin, 1996, Nagata & Kobayashi, 2013).

3. Methodology

The objective of this article is to develop a protocol that can capture the characteristics of individual plants and coordinate among all collaborating participants in the system to work most effectively. In this section, we demonstrate the design of collaborative control protocol for early detection of stress in plants. The protocol has three main parts:

collaborative control protocol, routing algorithm, and adaptive search algorithm.

3.1. Task description

The agricultural robotic system (ARS) comprises humans (one or more), a robot mounted on a remote-controlled or autonomous mobile cart, and multiple 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 perspective, the robot that moves through required locations needs to be equipped with arms and sensors to inspect the conditions at each spot (Edan & Miles, 1994). A robot will carry several types of sensors which contain detection agents. Routing a robot in a greenhouse environment is required to perform planning in an unstructured but predictable, knowledge-based environment.

Where agricultural 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. Therefore, a representative sample of the plants in each area is typically selected for assessment and inspection. Using this approach, the chosen sample of plants can be assumed to represent that local area, thus saving time and cost. After the samples of plants are selected, a robot is guided from the Robot Base Point (see Fig. 1) to visit each selected sample location 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 certain 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 certain 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 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). Fig. 1 illustrates the situation described above.

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

The protocol is derived from Collaborative Control Theory (CCT) principles. The system components (parties) 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 (components) need to collaborate intelligently to effectively perform the task. As all agents of the system are working together, this is “Mandatory Collaboration,” or collaborate as required.

Collaborative control protocol is designed to integrate agents (parties) in ARS system to work seamlessly. The protocol design starts with creating collaboration requirement planning (CRP) which helps the designer to allocate tasks to agents. After the planning of tasks has been 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.

3.2.1. Collaboration requirement planning (CRP)

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

CRP-I: Plan Generation

Referring to 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 Collaboration requirement matrix (CRM) is necessary. CRM can be expressed as follow.

$$A(R_i)xT_i \rightarrow CRM \quad (1)$$

A(Ri) denotes the set of resources i available for each task. T_i denotes as tasks and R_i denotes available agent or group of agents. The matrix will generate CRM which contain CRM(R_i, T_i) = 0 when resource R_i is not available for task T_i and CRM(R_i, T_i) = 1 when resource R_i is available for task T_i.

For this situation, at each location, the robot inspects several

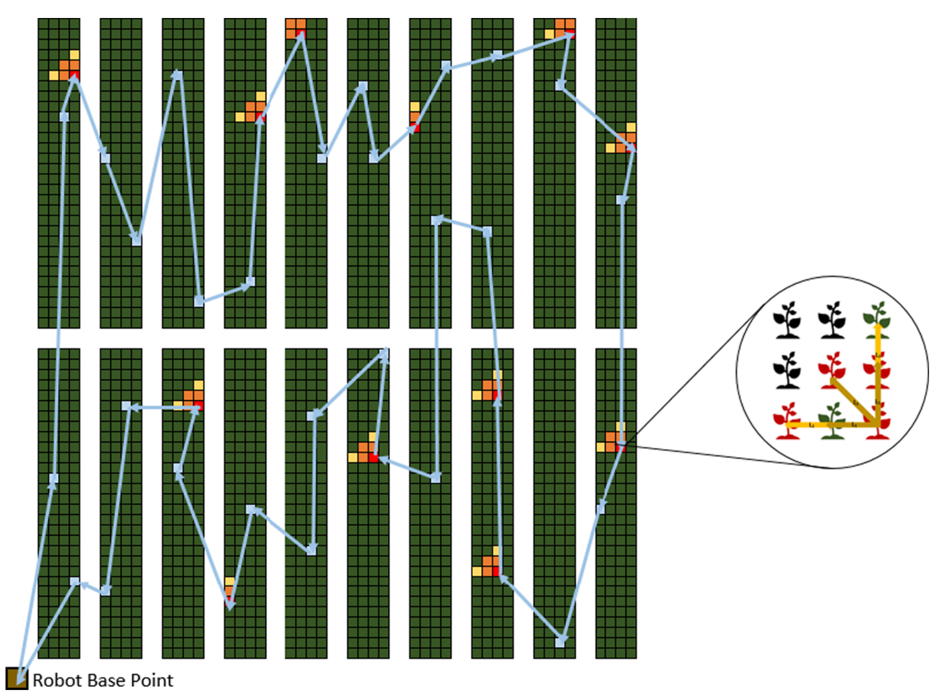

Fig. 1. Agricultural robotic system operation; North and West propagation directions assumed as given by experts for this crop season.

parameters with respect to the defined task and only one robot is required in this ARS. The system will have several types of tasks T_i as follows.

T_1 = compute the routing
T_2 = moving to the location
T_3 = measure the parameter
T_4 = error and conflict checking
T_5 = stress status checking
T_6 = decision making

For the agent or group of agents R_i, one can define the following team.

R_1 = {robot}
R_2 = {sensors}
R_3 = {human}
R_4 = {robot, sensor}
R_5 = {robot, human}
R_6 = {sensor, human}
R_7 = {robot, sensor, human}

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

$$CRM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

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 monitoring task needs to be generated. The plan will define route and parameters that are needed to be measured by sensors. Therefore, the plan will define a route of a mobile robot, order of locations to be visited, order of information to be obtained at each location, and assignment of sensor (s) to measure parameters. In addition, 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.

3.2.2. Error prevention and conflict resolution (EPCR)

In the ARS, there are potential conflicts and errors. 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.

$$\exists\, E\,[ARS_{Single}(t)],\quad if\quad S_{ARS_{Single}}(t) \overset{Dissatisfy}{\rightarrow} N_r(t) \quad (2)$$

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

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

$$\exists\, C\,[ARS_{Group}(t)],\quad if\quad S_{ARS_{Group}}(t) \overset{Dissatisfy}{\rightarrow} N_r(t) \quad (3)$$

Where C is conflict, $ARS_{Group}(t)$ is group of ARS′ agents at time t, $S_{ARS_{Group}}(t)$ is state of group of ARS′ agents 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 2.

In every system, conflicts and errors are unavoidable. 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, real performance of protocol can be analyzed. In addition, how to develop EPCR algorithm to solve conflicts and errors effectively can be future research in ARS area.

3.2.3. Elements in CCP-ED

CCP-ED's objective is to utilize resource available to detect stress in greenhouse crops. The following sections describe CCP-ED in details and Fig. 3 presents CCP-ED workflow.

Decision variables:

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

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

Parameters:

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

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

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

$$p_j\,' = extra\ \ processing\ \ time\ \ at\ \ node\ \ j$$

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

$$q_j\,' = extra\ \ searching\ \ time\ \ at\ \ node\ \ j$$

Table 2
Major potential errors and conflicts examples in ARS planning and control.

Type	Example	Collaborator(s)	State	Constraint
Error	Path error	Human	Inputs wrong data, or does not run the algorithm properly	The objective of the routing
Error	Movement or routing error	Robot	Robot cannot move according to the 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 the initial route	Human/operator objective
Conflict	Information conflict	Human and sensor	The human does not receive the information on time	Sensor's objective/capacities
Conflict	Time measuring conflict	Robot and sensor	The robot tends to move to a new location, but sensor has not yet finished measuring the parameter	Robot's and sensor(s)' objectives/ capacities
Conflict	Transition conflict	Human, robot, and sensor	The sensor information is not sent by a robot to the human	Robot's capacity; Sensor's capacity
Conflict	Sensor conflict	Multiple sensors	Two sensors provide conflicting data for the same parameter measurement	Sensor's capacity
Conflict	Human conflict	Two or more humans	Decision from two humans are different	Human's capacity

$T = total\ \ avible\ \ time$

$T' = total\ \ processing\ \ time$

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

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

3.2.4. CCP-ED steps

Step 1 Sample n nodes

Step 2 Use genetics algorithm to develop the TSP route for n nodes. Travelling 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:

$$\min z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{4}$$

Subject to:

$$\sum_{i} x_{ij} = 1;\ \ for\ \ all\ \ j \tag{5}$$

$$\sum_{j} x_{ij} = 1;\ \ for\ \ all\ \ i \tag{6}$$

$$\sum_{i\in M}\sum_{j\in \bar{M}} x_{ij} \geq 2;\ \ for\ \ all\ \ i\ \ and\ \ j \tag{7}$$

$$x_{ij}\ \ binary\ \ for\ \ all\ \ i\ \ and\ \ j \tag{8}$$

- Objective function (4) is to minimize cost of traveling from i to j.
  - o 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 sum of a tour that has smallest cost c_{ij}.
- Constraint (5) is the constraint that forces all node has only one incoming arc.
  - o For all node, j will have only one arc from node i.
- Constraint (6) is the constraint that forces all node has only one outgoing arc.
  - o For all node, i will have only one arc out to node j.
- Constraint (7) is sub-tour elimination constraint.
  - o Let M and $\bar{M}$ be the partition of integer $i = \{1, 2\ldots n\}$ so that $M \cap \bar{M} = \varnothing$ and $M \cup \bar{M} = i$. When we partition group of nodes as mentioned above, this constraint ensures that every partition has at least 2 arcs. It means at least 1 arc-in and 1 arc-out for each partition. Because of this constraint, one can ensure that the sub-tour is eliminated.
  - o Fig. 2 (modified from Bellmore & Nemhauser, 1968) shows how constraint (7) can eliminate sub-tours. Which (a) has a sub tour since there is no path from M to $\bar{M}$ but (b) and (c) do not have any sub-tour, since there are 2 or more arcs from M to $\bar{M}$.

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 location j with probability of re-inspection 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 the decision y_i for search with probability of re-searchingα_j, $T' = T' + q_i$

$$y_i = \begin{cases} 1 & searching\ \ for\ \ the\ \ surroundingarea\ \ for\ \ the\ \ suspected\ \ plant \\ 0 & otherwise \end{cases}$$

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**.

Fig. 3 demonstrates workflow of CCP-ED as describing in the step by step protocol.

3.3. Routing algorithm

To guide a mobile robot to visit sampled locations, an effective routing algorithm which can create an optimal or near optimal tour is needed. An effective routing algorithm can save traveling time for mobile robot and allow ARS system to spend more time on finding infected plants. In this work, genetic algorithm is applied to find a tour for a mobile robot.

3.3.1. Algorithm steps

Step 1 Generate initial population: Initial locations population (tour) is randomly generated by having the same probability of choosing each path. The number of initial locations is 10 times larger than the size of the sampling locations (Storn, 1996). Each chromosome in this genetic algorithm is a list of locations that a mobile robot will potentially visit. In addition, all initial populations are feasible solution with different fitness value. 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 chance to select 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 probability of successful crossover equal to one minus expected errors in the system.

Step 4 Mutation: mutation is performed by switching randomly two locations (two genes) in each chromosome. The probability to successfully mutate is equal to one minus expected conflict in the system.

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

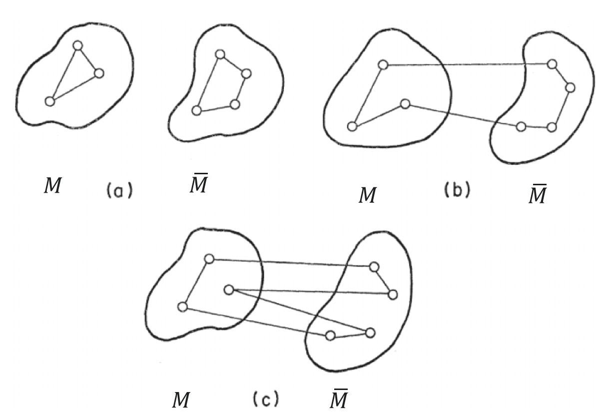

Fig. 2. Sub-tour elimination (modified from Bellmore & Nemhauser, 1968).

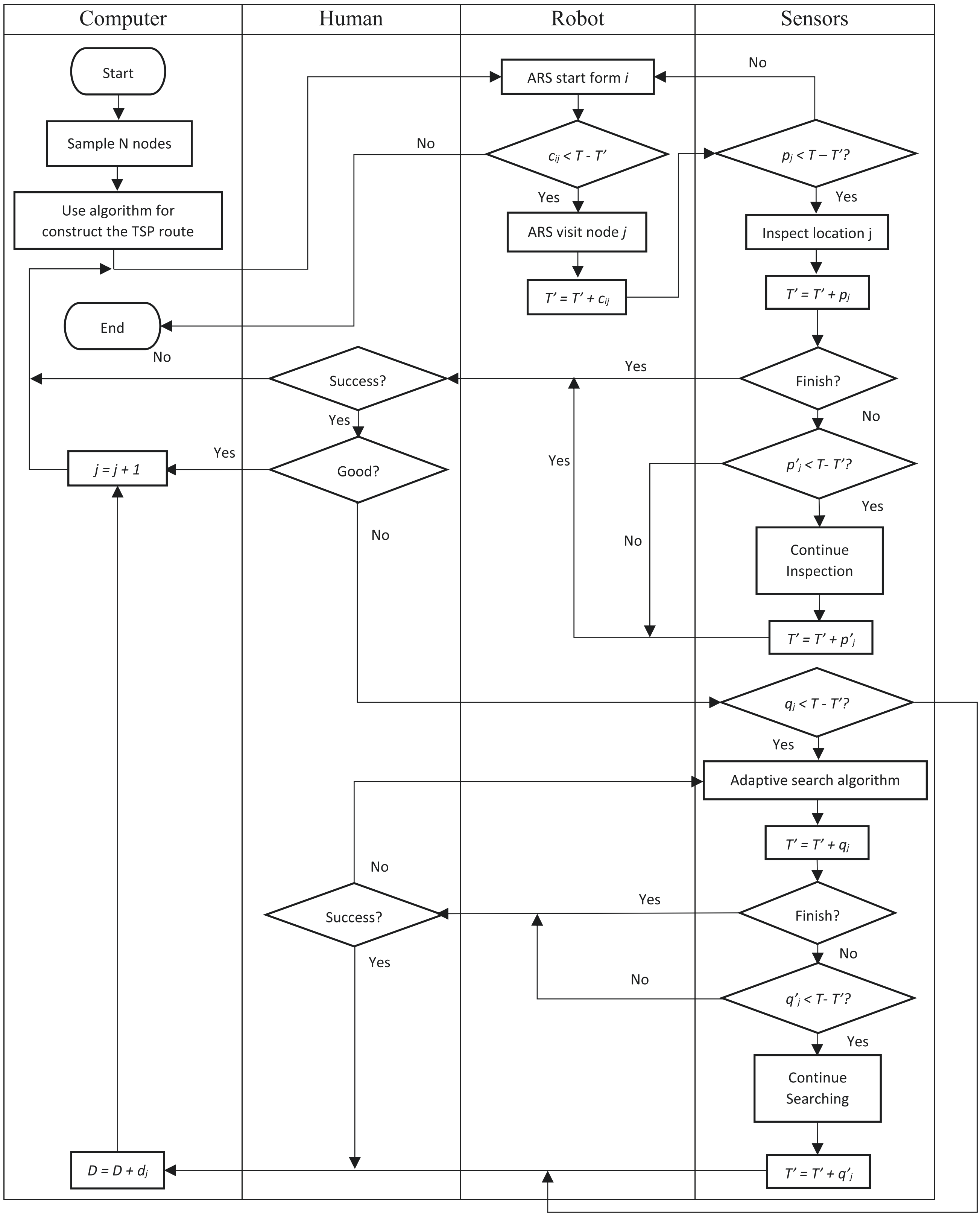

Fig. 3. CCP-ED Workflow.

After performing all five steps, the process repeats until it meets stopping criterion (three times number of sampling locations). The result is a planned tour for a mobile robot to actually visit and monitor stress conditions of inspected plants at each of those locations.

3.4. Adaptive search (AS) algorithm

Adaptive search (AS) algorithm is an important part of 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 in scientifically predictable directions. Stress and disease will more likely spread in directions influenced by sunlight and airflow, as discussed earlier. Therefore, given this knowledge, we can construct the adaptive search algorithm which can reflect real characteristics of plant stress or disease propagation. When the first infected plant is detected, the adaptive search algorithm will be activated to further explore high potential locations (i.e., locations with high risk of stress). In addition, if the fraction of stress plant from the first operation is beyond a threshold, the second search which help to indicate severity of such area will be activated. The search algorithm is adaptive, based on the new information just obtained, type of disease, type of plant, stress severity in plant, season of the year, and other environmental conditions.

Suppose Northern and Western directions of plant in greenhouse are more likely to have similar stress symptoms, based on prior pathological knowledge. Therefore, the adaptive search algorithm should further inspect plants in the given directions once the first infected plant is found. (Fig. 4)

3.4.1. Algorithm steps

Step 1 Sensors inspect plant at sampled location

Step 2 If the sampled plant has abnormal condition (sign of stress, diseases, etc.), adaptive search algorithm will be activated by starting to search for 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 host.

By performing 4 steps of 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. Adaptive search algorithm will be activated once sensors found the first signs of stress in plant. This procedure will save time for searching in unlikely plants' locations.

3.5. Protocol performance metrics

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

1. Stressed or infected location found

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

$$D = \sum_{i=1}^{n} d_i \tag{9}$$

D = total number of infected locations found
d_i = number of infected locations found at location i
n = number of sampled locations in the greenhouse

2. 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 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.

$$ORE = UxPxS \tag{10}$$

Where

$$U = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}(p_i + q_i + c_{ij})}{T} \tag{11}$$

$$P = \frac{\sum_{i=1}^{n}(p_i + q_i)}{\sum_{j=1}^{m}\sum_{i=1}^{n}(p_i + q_i + c_{ij})} \tag{12}$$

$$S = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} s_i} \tag{13}$$

U = utilization
P = performance
S = success
c_{ij} = travel time from i to j
d_i = number of infected plants found during operation at location i
m, n = number of sampled locations in the greenhouse
p_i = inspection time at location i
q_i = searching time at location i
s_i = number of inspected plants at location i
T = total available time

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

4. Experiments

To assess and validate the CCP-ED designed in this research, computer simulation is used to test its effectiveness. With the same amount of resources (and time), a protocol which finds the highest number of existing stressed or already infected plants and has the highest ORE would be preferred.

4.1. Experimental design

The experiment is meant to methodically determine effectiveness of the CCP-ED. By assuring that a greenhouse has healthy plants, and effectively protecting or recovering unhealthy plants inside, the protocol is implemented to find the stressed or infected locations and the ORE of the system is measured. A greenhouse simulation with a mobile robot which is mounted with sensors is the platform. To compare performance of the developed protocol, six alternative protocol designs which represent different combinations of algorithm are tested (Table 3).

Assumptions and parameters

1. The greenhouse consists of 10x5 subspaces and each subspace is

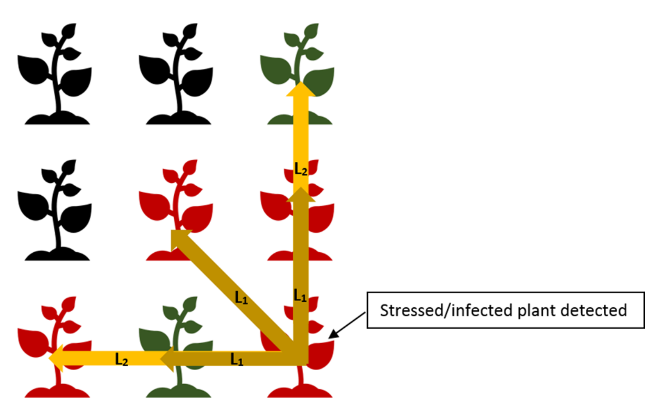

Fig. 4. Adaptive search algorithm when propagation directions are known from prior pathological findings.

Table 3
Alternative protocols design.

Protocol design No.	Routing algorithm	Search algorithm
1	Genetic Algorithm	Adaptive Search
2	Genetic Algorithm	None
3	Genetic Algorithm	Always Search
4	Random Routing	Adaptive Search
5	Random Routing	None
6	Random Routing	Always Search

composed of 30x4 locations.

2. The infected will not propagate between subspaces which means that the search algorithm needs to inspect only the locations within a given subspace.
3. Two hours of operation time (120 min) with the inspection time distributed as normal distribution N(20/60, 5/60) minutes and searching time distributed as normal distribution N(80/60, 20/60) minutes.
4. The probability of conflict between sensors' reading is 5% and error is 10%.
5. The conflict cost time is normally distributed N(5/60, 1/60) for inspection and N(20/60, 4/60) for searching.
6. The speed of the robot movement is 50 m/minute. (Ji, 2014)
7. Using rectilinear distance from location to location.
8. The sampling location will represent the status of surrounding area.

5. Results and analysis

With 100 replications of simulation, the results are shown in Fig. 5 and Fig. 6. Table 4 and Table 5 summarize the matrices from the simulation runs. The developed protocol had the significantly highest number of infected locations found, even though standard deviation of this protocol is also relatively higher than others.

From the results, the routing algorithm can improve the number of infected locations found by 37.74% compared with the alternate design. The adaptive search algorithm can improve the number of infected locations found by 71.88%.

ORE of the developed protocol also higher than others which means the highest productivity of a robotic 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.

T-test results of the number of infected location (Table 6) have been done with null hypothesis are the number of infected locations found by developed protocol and other protocol are the same. With 99.5% confidence interval, the null hypothesizes are rejected. The developed protocol can find significantly larger number of existing infected location than other protocol.

In addition, *t*-test (Table 7) indicates that the null hypothesizes "ORE of the developed protocol is the same with other protocols" are rejected at 99.5% statistically significant level. Therefore, ORE of the developed protocol is significantly higher than other protocols.

6. Conclusions and discussions

A collaborative control protocol for agricultural robot routing with online adaptation is established for monitoring the condition of plant in a greenhouse. The collaborative control protocol combines three main parties (humans, a mobile robot, and sensors) in the system together by utilizing CCT. The objective of the system is to monitor condition of plants with limited resources. Errors and conflicts of the system are considered in the developed protocol. The protocol expands system

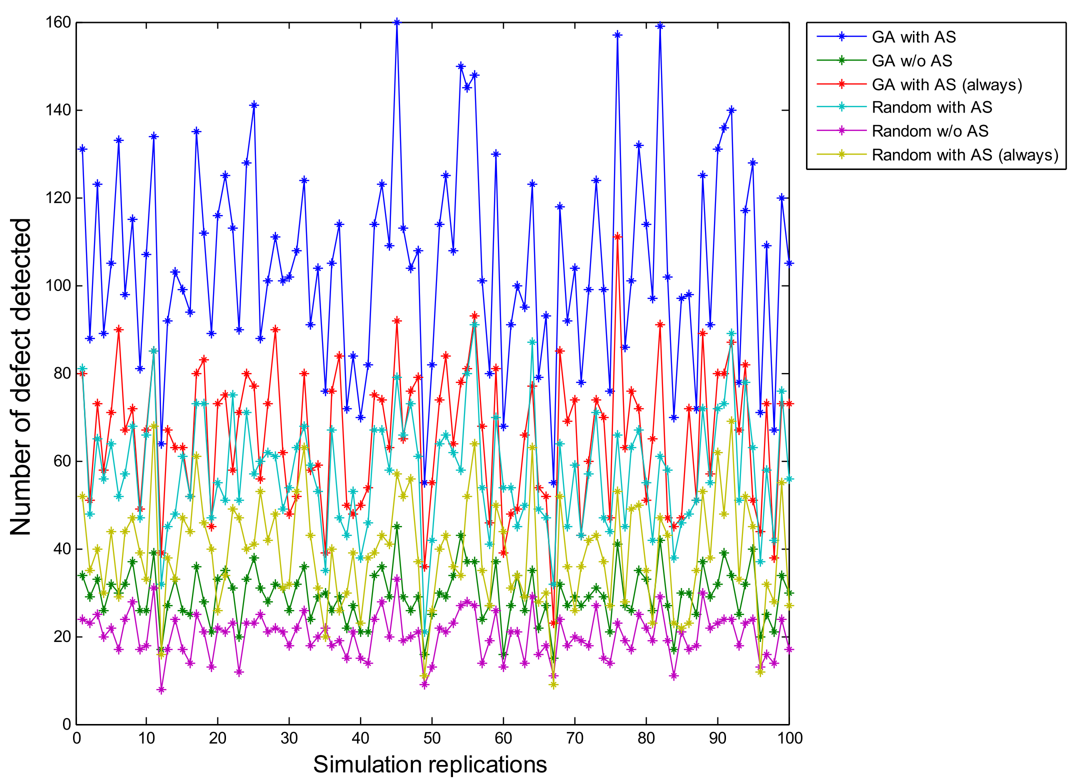

Fig. 5. Number of stressed/infected plants in each run.

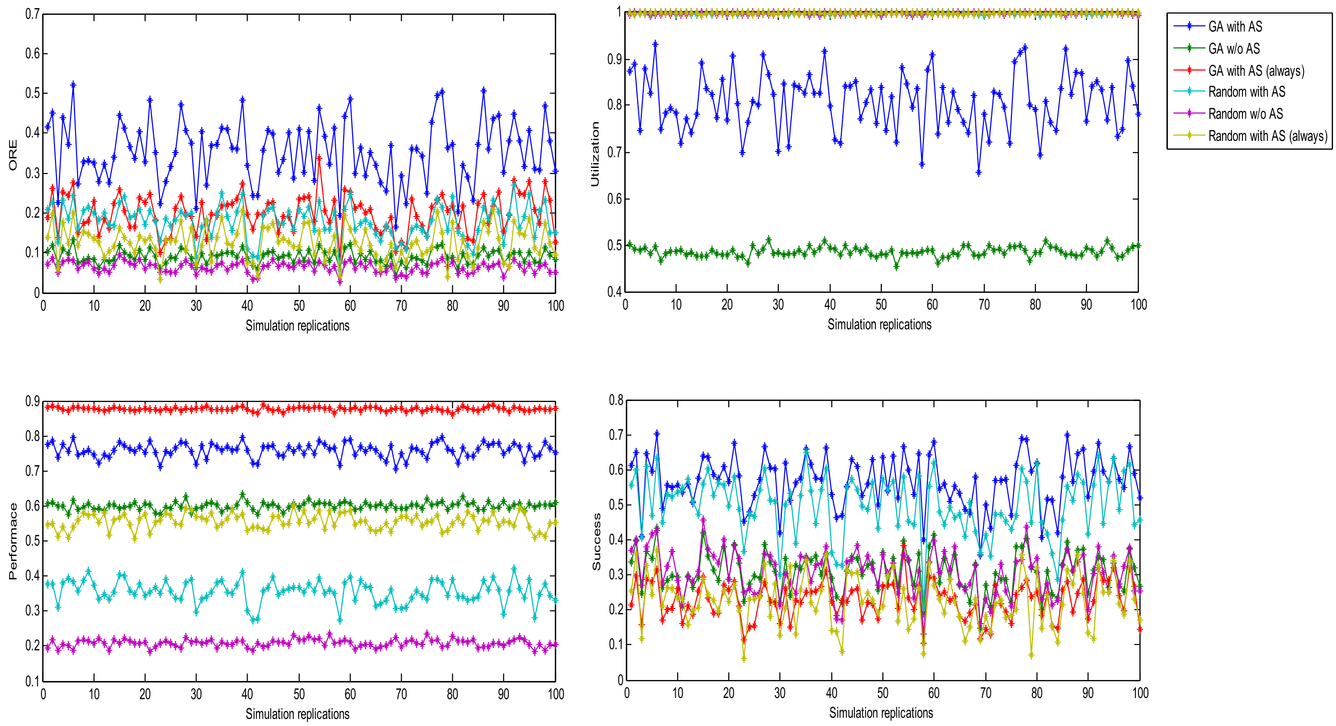

Fig. 6. Overall Robotic Efficiency.

Table 4
Data collected form simulation.

	Protocol 1*	Protocol 2	Protocol 3	Protocol 4	Protocol 5	Protocol 6
Avg. infected found	105.97	29.80	65.97	57.47	20.17	39.00
SD of infected found	25.15	6.16	17.12	13.03	4.62	13.08
Avg. ORE	36.74%	9.59%	20.47%	9.59%	6.60%	12.38%
SD of ORE	8.27%	2.02%	4.00%	2.02%	1.60%	3.89%

* Preferred design.

Table 5
Overall robotic efficiency.

	Protocol 1	Protocol 2	Protocol 3	Protocol 4	Protocol 5	Protocol 6
Avg. Utilization	83.06%	50.34%	99.89%	99.80%	99.77%	99.85%
Avg. Performance	76.37%	61.27%	87.87%	35.73%	21.49%	55.82%
Avg. Success	57.09%	31.11%	23.32%	48.61%	30.81%	22.21%
Avg. ORE	36.74%	9.59%	20.47%	9.59%	6.60%	12.38%

Table 6
t-test of the equality of infected locations found.

	H_0 Hypothesis	P value
1	Average of infected locations found is the same in Protocol 1 and Protocol 2.	< 0.005
2	Average of infected locations found is the same in Protocol 1 and Protocol 3.	< 0.005
3	Average of infected locations found is the same in Protocol 1 and Protocol 4.	< 0.005
4	Average of infected locations found is the same in Protocol 1 and Protocol 5.	< 0.005
5	Average of infected locations found is the same in Protocol 1 and Protocol 6.	< 0.005

*At significance level 0.05, based on P values of < 0.05, all null hypotheses shown are rejected

design under the cooperative parties. In this work, performance matrices which are number of infected locations and overall robotic efficiency (ORE) are used to measure performance of the protocol. ORE will capture overall capability of the robotic system in three aspects; utilization, performance, and successfulness. With the higher ORE, the system will better utilize available resources.

The methodology is developed and validated in this work. Routing algorithm can save traveling time by creating a tour for a mobile robot. Time saved by routing algorithm can be used for performing inspecting and monitoring task. Routing algorithm improved number of infected location found by 45.77%. Adaptive search algorithm captures scientifically known characteristics of plant which can propagate disease to the nearby locations in specific directions and can improve detection rate by 71.88%. By having too sensitive search algorithm, however, a

Table 7
t-test of the equality of ORE.

	H_0 Hypothesis	P value
1	Average of ORE is the same in Protocol 1 and Protocol 2.	< 0.005
2	Average of ORE is the same in Protocol 1 and Protocol 3.	< 0.005
3	Average of ORE is the same in Protocol 1 and Protocol 4.	< 0.005
4	Average of ORE is the same in Protocol 1 and Protocol 5.	< 0.005
5	Average of ORE is the same in Protocol 1 and Protocol 6.	< 0.005

*At significance level 0.05, based on P values of < 0.05, all null hypotheses shown are rejected.

system will waste time to search at the areas that do not have, or have lower risk of locating an infected plant. Therefore, search algorithm should be activated only when sensors found the suspicious area.

For recommendations, parameters can be adjusted for reflecting on other situations. In the future, researchers can pursue the most effective sampling policy by using historical data to select proper locations for each run. Moreover, researchers can further explore situations where a single robot is not sufficient. Operational area might be relatively too large for one robot, or tasks assigned for the robot are complicated and need more than one robot to perform the operations. Protocol for multi-robots operating system would be another future challenge for researchers. Lastly, research about EPCR algorithm to solve potential conflicts and errors in the system before they occur can be an important research question open for further exploration and discoveries.

Acknowledgements

The research presented here is supported by PRISM Center for Production, Robotics, and Integration Software for Manufacturing & Management at Purdue University. In addition, the research on ARS is supported by BARD project Grant# IS-4886-16R, "*Development of a Robotic Inspection System for Early Identification and Locating of Biotic and Abiotic Stresses in Greenhouse Crops*" with Co-PIs A. Bechar (ARO, Israel); Y. Tao (UMD); and S.Y. Nof. An abbreviated version of this article was submitted and presented in 2018 IISE Annual Conference.

References

Astrand, B., & Baerveldt, A. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots, 13*, 21–35.
Bellmore, M., & Nemhauser, G. L. (1968). The traveling salesman problem: A survey. *Operations Research, 16*(3), 538–558.
Bochtis, D. D., Sørensen, C. G., Jorgensen, R. N., Norremark, M., Hameed, I. A., & Swain, K. C. (2011). Robotic weed monitoring, acta Agriculturae scandinavica, section B. *Soil & Plant Science, 61*(3), 202–208.
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.
Cheein, F. A., & Carelli, R. (2013). Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE Industrial Electronics Magazine, 7*(3).
Diker, K., & Bausch, W. C. (2003). Radiometric field measurements of maize for estimating soil and plant nitrogen. *Biosystems Engineering, 86*(4), 411–420.
Eaton, R., Katupitiya, J., & Siew, K. W. (2008). *Autonomous farming: Modeling and control of agricultural machinery in a unified framework. 15th International Conference on Mechatronics and Machine Vision in Practice*.
Edan, Y., & Miles, G. E. (1994). Systems engineering of agricultural robot design. *IEEE Transactions on System, Man, and Cyber, 24*(8), 59–65.
Edan, Y. (1999). Food and agriculture robotics. In S. Y. Nof (Ed.). *Handbook of industrial robotics*. New York: John Wiley.
Edan, Y., Han, S., & Kondo, N. (2009). Automation in Agriculture. In S. Y. Nof (Ed.). *Springer Handbook of Automation* (pp. 1095–1128). Berlin, Heidelberg: Springer.
Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science, 327*(5967).
Grad, I., Mateoc-Sirb, N., Manescu, C., Mateoc, T., Brebu, C., & Toth, A. (2014). Study on precision agriculture in Romania. *Agricultural Management, 16*(1), 238–240.
Iida, M., Uchida, R., Zhu, H., Suguri, M., Kurita, H., & Masuda, R. (2013). Path-following control of a head-feeding combine robot. *Engineering in Agriculture, Environment and Food, 6*(2), 61–67.
Ishibashi, Maya, Iida, Michihisa, Suguri, Masahiko, & Masuda, Ryohei (2013). Remote monitoring of agricultural robot using web application. *IFAC Proceedings Volumes, 46*(18), 138–142. https://doi.org/10.3182/20130828-2-SF-3019.00047.
Ji, C. (2014). Vision information acquisition for fruit harvesting robot and development of robot prototype system. Ph.D. Dissertation, China Agricultural University (pp. 72–78).
Keicher, R., & Seufert, H. (2000). Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture, 25*, 169–194.
Khan, A., Martin, P., & Hardiman, P. (2004). Expanded production of labor-intensive crops increases agricultural employment. *California Agriculture, 58*(1).
Ko, M. H., Ryuh, B., Kim, K. C., Suprem, A., & Mahalik, N. P. (2015). Autonomous greenhouse mobile robot driving strategies. *System Integration Perspective: Review and Application IEEE/ASME Transactions on Mechatronics, 20*(4).
Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research, 21*(2), 498–516.
Mare, H., & Mare, F. (2015). A decision support model for the adoption of precision agriculture practices. *International Journal of Agricultural Management, 5*(1/2), 31–37.
McBratney, A., Whelan, B., Ancev, T., & Bouma, J. (2004). *Future directions of precision agriculture. 7th International Conference on Precision Agriculture*. USA: Minneapolis.
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.
Moghaddam, M., & Nof, S. Y. (2016). Real-time optimization and control mechanisms for collaborative demand and capacity sharing. *International Journal of Production Economics, 171*, 495–506.
Nagasaka, Y., Zhang, Q., Grif, T. E., Kanetani, Y., Umeda, N., & Kokuryu, T. (2004). An autonomous field watching-dog robot for information collection in agricultural fields. *An ASAE/CSAE Meeting Presentation* (pp. 4215–4222). .
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.
Nayak, A., Levalle, R. R., Lee, S., & Nof, S. Y. (2016). Resource sharing in cyber-physical systems: Modeling framework and case studies. *International Journal of Production Research, 54*(23).
Nof, S. Y. (2007). Collaborative control theory for e-Work, e-Production, and e-Service. *Annual Reviews in Control, 31*, 281–292.
Nof, S. Y., Ceroni, J., Jeong, W., & Moghaddam, M. (2015). *Revolutionizing Collaboration through e-Work, e-Business, and e-Service.* ACES Series: Springer eBooks.
Oerke, E. C., & Dehne, H. W. (2004). Safeguarding production—losses in major crops and the role of crop protection. *Crop Protection, 23*, 275–285.
Potvin, J. Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research, 63*(1), 339–370.
Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture, 25*, 155–167.
Spekken, M., & Bruin, S. (2013). Optimized routing on agricultural fields by minimizing. *Precision Agriculture, 14*, 224–244.
Storn, R. (1996). On the usage of differential evolution for function optimization. *Proc. Biennial Conf. North Amer. Fuzzy Inform. Process. Soc.* (pp. 519–523). .
Velasquez, J. D., & Nof, S. Y. (2009). Best-matching protocols for assembly in e-work networks. *International Journal of Production Economics, 122*, 508–516.
Xue, J., Zhang, L., & Grift, T. E. (2012). Variable field-of-view machine vision based row guidance of an agricultural robot, Variable field-of-view machine vision based row guidance of an agricultural robot. *Computers and Electronics in Agriculture, 84*, 85–91.
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.
Zhong, H., Nof, S. Y., & Berman, S. (2015). Asynchronous cooperation requirement planning with reconfigurable end-effectors. *Robotics and Computer-Integrated Manufacturing, 34*, 95–104.
Zou, S., Yang, F., Tang, Y., Xiao, L., & Zhao, Y. (2016). Optimized algorithm of sensor node deployment for intelligent agricultural monitoring. *Computers and Electronics in Agriculture, 127*, 76–86.