UC Davis

UC Davis Previously Published Works

Title

Deterministic predictive dynamic scheduling for crop-transport co-robots acting as harvesting aids

Permalink

<https://escholarship.org/uc/item/7h04p54t>

Authors

Peng, C
Vougioukas, SG

Publication Date

2020-11-01

DOI

10.1016/j.compag.2020.105702

Peer reviewed

Deterministic predictive dynamic scheduling for crop-transport co-robots acting as harvesting aids

Chen Peng (彭晨)^{*,*}, Stavros G. Vougioukas^{*}

^a University of California, Davis, Department of Mechanical and Aerospace Engineering, One Shields Ave, Davis, CA 95616-5270, USA

^b University of California, Davis, Department of Biological and Agricultural Engineering, One Shields Ave, Davis, CA 95616-5270, USA

Abstract

Manual harvesting of fresh-market crops like strawberries is very labor-intensive. Apart from picking fruits, pickers spend significant amounts of time carrying full trays to a collection station at the edge of the field. Small teams of harvest-aid robots that help large picking crews by transporting empty and full trays can increase harvest efficiency by reducing pickers' non-productive walking times. However, robot sharing among the crew may introduce non-productive waiting delays between the time a tray becomes full and when a robot arrives to collect it. Reactive robot scheduling cannot eliminate mean waiting times because pickers must wait for a robot to travel the distance from the collection station to them. Predictive scheduling is better suited to this task, because if the time and location that a pickers' tray will fill are known to the scheduler in advance, a robot can start moving toward that location before the tray becomes full; hence, waiting times due to robot travel can be reduced or eliminated.

In this paper, dynamic predictive scheduling was modeled for teams of robots carrying trays during manual harvesting. The times and locations of the tray-transport requests were assumed to be known exactly (deterministic predictions). Near-optimal scheduling was implemented to provide efficiency upper-bounds for any predictive scheduling algorithms that incorporate uncertainty in the predictive requests. Robot-aided harvesting was simulated using manual-harvest data collected from a commercial picking crew. Scheduling performance was studied as a function of the number of robots – for a given crew size – with robot speed as a parameter. Additionally, the effect of the earliness of the availability of the predictions on performance was studied.

Experimental results showed that both reactive and predictive scheduling did not improve the mean non-productive time significantly relative to manual harvesting, when only four robots were used. Actually, deploying fewer than four robots led to worse non-productive time. However, introducing five to eight robots decreased mean non-productive time drastically, and when ten or more robots were used, non-productive time was reduced by 64.6% (reactive scheduling) and up to 93.7% (predictive scheduling) with respect to all-manual non-productive time. The efficiency increases were 15% and 24%, respectively. It was also verified that reactive dispatching always performed worse than deterministic predictive scheduling. Also, when the robot-to-picker ratio was larger than approximately 1:3, the waiting time and efficiency plateaued, i.e., did not improve further, regardless of how early the prediction was available to the scheduler. The reason is that the mean waiting time is lower bounded by the sum of mean travel time and tray exchange time, which are both constant. Although the above results represent upper-

bounds for performance – since predictions were perfect - they indicate that tray-transport robots acting as harvest aids can increase harvesting efficiency significantly when scheduled properly.

1. Introduction

Manual harvesting of fresh-market specialty crops, like strawberries, table grapes, cherry tomatoes and berry fruits is costly and labor-intensive. For example, the cost of labor for strawberry harvesting in California’s Central Coast represents 55% of the total operating cost per acre (Bolda, Tourte, Murdock, & Sumner, 2016). The crops mentioned above are planted in equally spaced parallel rows with furrows/aisles between them. Pickers walk inside the rows or furrows and harvest selectively, based primarily on ripeness criteria (see Fig. 1a, for a strawberry harvesting example). Each picker places the harvested crop in a small container (e.g., tray, bag or wheelbarrow) (Fig. 1b). When the container becomes full, the picker walks and transports it to a collection station at the field’s headland (Fig. 1c) and takes an empty container to resume picking. Selective harvesting requires advanced perception, to detect and select harvestable fruits; significant dexterity and skill, to harvest them without damaging them, and speed, to work fast, in a cost-effective manner. Building machines that meet all these requirements is challenging. Robotic harvesters for these crops have either not been developed at all (e.g., table grapes), or existing prototypes have not successfully replaced pickers in real-world open field production (e.g., strawberries).

Fig. 1 a) Pickers walk and harvest inside furrows; b) a picker places strawberry in plastic containers inside a tray that lies on a picking cart (Seyyedhasani, Peng, Jang, Vougioukas, 2020a); c) trays are transported to the headland.

1.1 Scheduling teams of tray-transport robots

As an alternative to fully automated picking, mechanized labor aids aim to increase worker productivity by making it easier/faster to pick or by reducing the non-productive parts of the harvesting cycle. For example, to reduce the time pickers spend walking for container transportation during harvesting the crops mentioned above, teams of mobile, tray-transporting robots are investigated, with strawberries serving as a case-study crop (USDA REEIS, 2013). During robot-aided harvesting, each picker walks inside a furrow, collects ripe fruits, and puts them in a standard-sized tray (Fig. 2b) located on an instrumented cart (Fig. 2a), in the same way as in all-manual harvesting. These carts are equipped with load cell sensors to detect the weight of the tray and GPS module to observe the motion of the carts (Khosro Anjom, Vougioukas, Slaughter, 2018a). The cart system informs the robots – via wireless transmission - when and where a tray becomes full (Khosro Anjom, & Vougioukas, 2019). A full tray results in a *tray-transport request* to the robot team. The robots travel between the collection station and

the pickers to bring empty trays to pickers and transport the full trays with the harvested produce from pickers to the collection station. One version of these robots is designed with a compact form-factor to allow travel inside narrow picking furrows (Fig. 2c); as a result, payload capacity is limited to one-tray.

Fig. 2 a) picker places strawberries in plastic containers inside a tray that lies on a picking cart; b) A standard picking tray with known capacity; c) Crop-transport robot traveling inside a furrow (Seyyedhasani, Peng, Jang, Vougioukas, 2020a).

In-field automated tray logistics can reduce the non-productive walking time of pickers. For strawberries, walking time has been measured to reach up to 22% of the harvest time (Khosro Anjom, Vougioukas & Slaughter, 2018b); higher inefficiencies are often reported, anecdotally. For reasons of cost effectiveness, robot team size must be kept small, and, therefore, each robot must serve many pickers. However, robot-sharing among the crew may introduce waiting time, i.e., a delay between the instant when a tray-transport request arises (tray becomes full) and when a robot arrives to collect the tray. Hence, efficient robot dynamic scheduling is essential to ensure that the overall reduction in walking time is larger than the waiting time introduced by robot operation. In reactive scheduling (Blazewicz, et al., 2019), a machine/vehicle/robot is allocated to a task only after the scheduler receives the task request. In the context of tray-transport robots for harvesting, Jang (2018) and Seyyedhasani, Peng, Jang and Vougioukas (2020b) showed that in reactive scheduling, picker waiting time is reduced when the robot-to-pickers ratio increases; however, above a certain ratio, adding more robots does not reduce further the waiting time. The reason is that a picker's waiting time is at least as much as the time needed for the robot to travel the distance from the collection station to the picker. Predictive scheduling policies are better suited to this task (Ritzinger, Puchinger & Hartl, 2016), because, if the time and location of a pickers' next tray-transport request is known to the scheduler in advance, a robot can be dispatched – and start moving toward that location – before the tray becomes full; hence, waiting times due to robot traveling can be reduced or eliminated.

Two issues are of primary importance when predicting the location and time of tray-transport requests: *uncertainty* and *availability earliness*. Uncertainty can be detrimental for predictive scheduling algorithms that assume perfect information (Bertsimas et al., 1991; Blazewicz, et al., 2019). Dynamic stochastic scheduling algorithms that incorporate prediction uncertainty can improve performance (Bent et al., 2004; Ichoua, Gendreau, Potvin, 2006; Blazewicz, et al., 2019). However, to evaluate and compare such algorithms (existing and new ones) in the context of robot-aided harvesting, one needs to compare them against a baseline, i.e., the “best possible” situation. The best possible situation – from an uncertainty point of view – is predictive scheduling when tray-transport request predictions are perfect, without uncertainty. Availability earliness refers to how early a prediction becomes available to the scheduling algorithm before the actual request takes place. Khosro Anjom and Vougioukas (2019)

showed that the earlier a tray-transport prediction is made – while a tray is being filled – the larger the uncertainty will be. Additionally, even if predictions are perfect, tray-transport request predictions arriving too late (too close to the actual request) will result in increased picker waiting times, because robots need time to travel to the location of the request.

1.2 Research goals and contributions

The first goal of this paper is to model and implement an optimal deterministic *predictive* scheduling algorithm for tray-transport requests during harvesting, and study its performance for different robot-to-picker ratios. The second goal is to explore the effect of prediction availability earliness on the mean waiting time of the pickers, under optimal deterministic predictive scheduling.

To achieve these goals, a strawberry harvesting simulator was developed based on a hybrid systems approach that models the harvest-related activities and motions of all agents (pickers and robots) involved in harvesting. A major contribution of this work lies in the development of a mathematical model for optimal deterministic predictive scheduling in the context of manual harvesting, with robots offering automated tray logistics (Section 3). The model was solved using exact and approximate methods, and the solver was embedded into the simulator, as a separate module. Another contribution lies in the extensive simulation experiments that were performed based on actual strawberry manual harvesting data; the analyses that were conducted to study the effect of robot-to-picker ratio and prediction availability earliness on scheduling performance, and the explanation of the performance curves that resulted (Section 7).

The rest of this paper is organized as follows: Section 2 describes a model for manual picking activities and robot tray-transport operations in strawberry harvesting. In Section 3 a mathematical model for predictive scheduling is developed, and in Section 4, an exact and an approximate algorithm to solve the model are reviewed and adopted. Section 5 presents briefly the simulator and the embedded tray-transport request and predictive scheduling modules. Section 6 presents the experimental design of our experiments and section 7 presents the results and their discussion. Finally, the main conclusions of this paper are summarized in section 8.

2. Modeling of harvesting activity with crop-transporting robots

The strawberries are typically planted in equally spaced parallel raised plant beds with furrows and aisles between them that accommodate human and machine traffic (Fig. 3a). The field headlands are reserved for collection/parking/inspection stations and traffic of people, forklifts and trucks involved in the handling and transportation of the harvested crops. A typical harvesting block consists of approximate 80~120 furrows in the length of about 100 meters. Each harvesting block is divided into two sections by a splitting line connecting the middle points of each furrow (block upper section is above blue dotted line in Fig. 3b). The field can be modeled using two points at the edges of each furrow, two points to represent the field split line, and one point for each collection station. The picker crew starts harvesting one section and transits together to the other in the field block. Before harvesting, there are several pre-allocated collection stations evenly distributed along the headland (Fig 3b). Harvesting starts from one corner of a block section and advances toward the other corner. The active collection station transits gradually as the picking crew moves.

Fig. 3 a) Layout of a typical raised-bed strawberry field; 3b) schematic figure of the strawberry harvesting field block with two sections (upper and lower); furrows; plant beds; field split line, and collection stations.

In the envisioned crop-transport robotic aiding harvesting, each picker enters an unoccupied furrow to start picking strawberries selectively from the plants on the raised beds on each side of that furrow. When their trays are full, they will wait at that location for the transporting robot before resuming picking. The dispatched robot starts from the active collection station, drives an empty tray to the assigned picker's full tray location and takes the full tray back to the active collection station, where they wait for next dispatching.

In our previous work (Seyyedhasani, Peng, Jang & Vougioukas, 2020a), a discrete-time hybrid system was developed to model and simulate the activities and motions of all agents involved in all-manual and robot-aided harvesting. A Finite State Machine (FSM) was utilized to model the discrete operating states/modes of the agents and the transitions between the operating modes. In this paper, for the purposes of simulating predictive scheduling, the FSM was simplified (since all-manual harvesting was not modeled). The activities of a picker during robot-aided harvesting were classified into seven discrete operating states/modes (Table 1), and the operations of a tray-transport robot into eight states (Table 2). The operating states of pickers and robots, and the possible transitions amongst them are shown in Fig. 4.

Operating state	Action
START	A picker leaves the collection station with an empty tray in hand, to start picking.
WALK_TO_FURROW_ENTRANCE	A picker walks in the headland, toward an empty (unoccupied) furrow.
WALK_TO_FURROW_SPLITLINE	A picker walks inside an empty (unoccupied) furrow until the field's split line is reached.
PICK	A picker is picking inside a furrow, with direction from the field split line toward the collection station.
WAIT FOR ROBOT ARRIVAL	A picker waits (idle), with a full tray, for a robot to come.
EXCHANGE_TRAYS	A picker takes the empty tray brought by the robot, and places a full tray on the robot.
STOP	A picker stops picking after the last tray is picked by a robot.

Table 1 States defined to represent a picker's operating states during robot-aided harvesting

Operating state	Action
START	Robot at the collection station starts operation with no tray on it.
AVAILABLE	Robot with one empty tray on it is waiting at the collection station to be dispatched to a tray-transport request.
TRAVEL_TO_PICKER	Robot travels from collection station – carrying an empty tray – toward the location of the transport request.
WAIT_UNTIL_TRAY_FILLS	Robot arrives at the location of the tray-transport request and waits for the picker to finish harvesting.
EXCHANGE_TRAYS	Robot is idle while picker exchanges the empty tray with a full tray.
TRANSPORT_FULL-TRAY	Robot travels toward the collection station to deliver a full tray.
IDLE_IN_QUEUE	Robot with a full tray waits in a queue at the collection station to have its tray unloaded, and an empty tray loaded.
STOP	Robot stops its operation at the collection station after the last tray has been unloaded.

Table 2 States defined to represent a robot's operating states during robot-aided harvesting

Fig. 4 Two coupled finite state machines model human picker and robot operations during robot-aided harvesting, where robots transport full and empty trays.

Following our previous work (Seyyedhasani, Peng, Jang & Vougioukas, 2020a), the operation of the p th picker at any time step k , in operating state S_p is represented by a continuous state vector, $\mathbf{X}_{p,k} = (x_{p,k}, y_{p,k}, W_{p,k})$ that contains as state variables the picker position coordinates $x_{p,k}$, $y_{p,k}$, and the weight $W_{p,k}$ of the pickers' tray. The elapsed time $T_{p,k}$ inside the current state is also part of S_p . The variables are updated with state-dependent difference equations that model picker kinematics (Eq 1, Eq 2), crop harvesting (Eq 3) and elapsed time (Eq 4):

$$x_{p,k+1} = x_{p,k} + \Delta t V_{S_p} \cos \theta_{S_p} \quad (\text{Eq 1})$$

$$y_{p,k+1} = y_{p,k} + \Delta t V_{S_p} \sin \theta_{S_p} \quad (\text{Eq 2})$$

$$W_{p,k+1} = W_{p,k} + \Delta t \hat{p}_{s_p} \quad (\text{Eq 3})$$

$$T_{p,k+1} = T_{p,k} + \Delta t \quad (\text{Eq 4})$$

In state S_p , V_{s_p} is the pickers' travel speed, θ_{s_p} is the picker's heading direction, and $\hat{p}_{s_p}$ is the fruit picking rate. The parameters are assumed constant within S_p , i.e., mean values are used over the duration of the state. Parameters V_{s_p} , $\hat{p}_{s_p}$ depend on individual picker performance, yield, and external factors that cannot be modeled (e.g., answering a phone call, stretching), and thus, their values are modeled as stochastic variables. The travel direction θ_{s_p} is set by the direction of the furrow or the headland that the picker travels in, depending on the state. The walking speed V_{s_p} is the same inside states "WALK_TO_FURROW_ENTRANCE" and "WALK_TO_FURROW_SPLITLINE", because the picker walks carrying a cart and tray to relocate (not pick), and is equal to a value v^w ; inside state "PICK", V_{s_p} is equal to a slow moving speed, v^p . V_{s_p} is equal to zero in operating states "WAIT_FOR_ROBOT_ARRIVAL" and "EXCHANGE_TRAYS". If the capacity of the tray is m^c , and the time spent in filling the i th tray (in state "PICK") is Δt_i^{pick} , the picking rate, $\hat{p}_{s_p}$ in the "PICK" state can be estimated as $\hat{p}_{s_p} = m^c / \Delta t_i^{pick}$. Distributions for stochastic parameters v^w , v^p and Δt_i^{pick} can be estimated experimentally (see Section 7.1).

At any time step k , the operation of the r th robot in operating state S_r is represented by: a continuous state vector, $\mathbf{X}_{r,k} = (x_{r,k}, y_{r,k})$ that contains the robot's position coordinates; the elapsed time $T_{r,k}$ inside the current operating state, and the coordinates of the robot's current target location, $X_{t,k}, Y_{t,k}$. Simple robot kinematics are modeled by Eq 1, and Eq 2, where θ_{s_r} is the robot's moving direction; time is updated using Eq 4 ("r" is used instead of "p" in the subscripts). When the robot is idle, the target coordinates $X_{t,k}, Y_{t,k}$ are equal to its current coordinates. In the "TRAVEL_TO_PICKER" state, $X_{t,k}, Y_{t,k}$ are the coordinates of the dispatch location assigned by the scheduler; in the "TRANSPORT_FULL_TRAY" state, $X_{t,k}, Y_{t,k}$ are the coordinates of the current collection station. Robot speed V_{s_r} is equal to a parameter v_r that is assumed constant in all operating states in which the robot is travelling. The robot's traveling direction is along the direction of the headland or the furrow. The time duration of states "IDLE_IN_QUEUE" and "EXCHANGE_TRAYS" are assumed to be constant.

3. Deterministic Predictive Scheduling of Harvest-aid Robots

Deterministic predictive scheduling of a team of crop-transport robots is a variant of the well-known Capacitated Vehicle Routing Problem (CVRP) (Toth et al., 2002, Vougioukas et al., 2012). However, in the case of strawberry harvesting, each robot can only serve one tray-transport request, with each robot route starting and ending at the collection station. Hence, the scheduling of these crop-transport robots can be modeled as a parallel machine scheduling problem (PMSP) with the objective of minimizing the mean of waiting times of all transport requests (Lawler, E.L., et al., 1993). Essentially, servicing a tray-transport request with one robot is the equivalent of executing one individual job by one machine in PMSP; robots correspond to independent parallel machines. The optimal criterion of the scheduling system is to minimize the mean waiting time, ΔT^w , of all tray-transport requests.

In the envisioned robot-aided strawberry harvesting operation, each picker from a set $\mathcal{S}^p = \{P_1, P_2, \dots, P_Q\}$ of Q pickers harvests fruits in a tray that lies on a picking cart. When the tray fills, a robot F_k from a team of M identical transport robots $\mathcal{S}^f = \{F_1, F_2, \dots, F_M\}$ brings an empty tray to the

picker and carries the full tray to a collection station; the station's coordinates $\mathbf{L}^s$ are known. The robot scheduling algorithm has access to a set of predicted tray-transport requests $\mathcal{S}^{\mathcal{R}} = \{R_1, R_2, \dots, R_N\}$, where $0 \leq N \leq Q$ and computes an updated schedule every time a new predicted request enters $\mathcal{S}^{\mathcal{R}}$ and there is a robot available to be dispatched.

Let us assume that at some time t , a new predicted request R_i is generated and entered into $\mathcal{S}^{\mathcal{R}}$. R_i contains the following (known) information: (1) a prediction of the remaining time interval Δt_i^f (with respect to t) until the tray becomes full with fruit and ready to be transported, and (2) the predicted coordinates $\mathbf{L}_i$ of the cart's (and picker's) locations at time t_i^f when the tray becomes full ($t_i^f = t + \Delta t_i^f$). The travel time required for a robot to travel from the collection station location at $\mathbf{L}^s$ to $\mathbf{L}_i$ – and back – is computed by approximating the path by one straight line segment on the headland and another straight line segment inside the furrow that corresponds to $\mathbf{L}_i$ (Fig. 5). The one-way traveled distance D_{si} is the corresponding Manhattan distance from $\mathbf{L}^s$ to $\mathbf{L}_i$ along the path. The corresponding one-way travel time is $\Delta t_i^u = D_{si}/v^r$, where v^r is the robot's speed (assumed to be constant and the same for all robots).

Fig. 5 To compute robot travel time when serving a request, the robot's path from the collection station location ($\mathbf{L}^s$) to the tray-transport request location ($\mathbf{L}_i$) – and back – is approximated by a straight line segment on the headland and another straight line segment inside the furrow that corresponds to $\mathbf{L}_i$

Let us now assume that the scheduling algorithm selects robot F_k to serve some request R_i and dispatches the robot at some time t_{ki}^d (Fig. 6). The robot will arrive at $\mathbf{L}_i$ at time instant t_{ki}^a :

$$t_{ki}^a = t_{ki}^d + \Delta t_i^u \quad (\text{Eq 5})$$

Then, the waiting time for request R_i to be served by robot F_k will be Δt_{ki}^w :

$$\Delta t_{ki}^w = \max(t_{ki}^a - t_i^f, 0) \quad (\text{Eq 6})$$

The reason for using the 'maximum' operator is because the scheduling is predictive, and therefore, it is possible that robot F_k arrives at $\mathbf{L}_i$ before the predicted tray fill-up time, t_i^f ; in this case, there is no waiting time, i.e., Δt_{ki}^w is zero. Once the tray is full and the robot is at $\mathbf{L}_i$ the picker will take the empty tray from the robot and will load the full tray on the robot (and then resume picking). This corresponding

time is assumed constant and equal to Δt^L . Next, the robot will transport the tray back to the collection station; the required time will be Δt_i^u . When the robot arrives at the collection station it will take Δt^{UL} seconds until all previously arrived robots in the queue are served, the full tray it carries is unloaded and an empty tray is loaded on it, so that it is ready to be dispatched again. Obviously, Δt^{UL} is stochastic and depends on factors such as the number of robots and the working speed of the worker(s) at the collection station. Since we don't know the statistics of this variable, we cannot model it effectively as being stochastic; instead, we treat it as a deterministic parameter and select its value based on prior work. Seyyedhasani, Peng, Jang & Vougioukas, (2020b) calculated the frequency histogram of the corresponding time spent by workers at the collection station, when a crew of 25 pickers harvested. The histogram was heavily skewed toward the origin; 50% of the time, pickers spent at most 15 s waiting in line. Given that the number of robots will be small (much smaller than 25), and the fact that Δt^{UL} is typically much smaller than the transport time Δt_i^u , hence variations in its value have an even smaller effect – proportionally – on the total time between requests (Eq. 7), the value of 15 s was used in this work as a conservative (pessimistic) value for Δt^{UL} .

The total time, Δt_i^p , required by a robot to serve request R_i and be available to serve another request is shown in (Eq 7). So, the *completion time* of service of request R_i by robot F_k , is the time instant is (Eq 8).

$$\Delta t_i^p = 2\Delta t_i^u + \Delta t^L + \Delta t^{UL} \quad (\text{Eq 7})$$

$$t_{ki}^c = t_{ki}^d + \Delta t_i^p \quad (\text{Eq 8})$$

At any time, t , robot F_k is either serving some request R_i or is idle. In the former case, the robot is available to be dispatched again, after a time interval Δt_k^A (Eq 9) that is needed to serve request R_{i-1} .

$$\Delta t_k^A = \max(0, t_{k(i-1)}^c - t) \quad (\text{Eq 9})$$

If the robot is idle at the collection station, it is available to be dispatched immediately ($\Delta t_k^A = 0$).

Fig. 6 Timelines of request R_i served by robot F_k

Let us now focus on the robot dispatch time t_{ki}^d . Scheduling becomes more effective as the number of requests in $\mathcal{S}^R$ is larger (Lu, Sitters, Stougie, 2003). If the robot dispatch time is delayed by some amount Δt_i^r (also referred to as ‘release delay’, i.e., $t_{ki}^d \geq t + \Delta t_i^r$), each new request that may

arrive during Δt_i^r will cause an updated schedule to be computed. However, too long a release delay may increase the request waiting time, as the robot could depart too late. To select Δt_i^r , one can note that if the robot arrives early at L_i ($t_{ki}^a < t_i^f$), it will have to wait idle until the tray is full. The greatest value for Δt_i^r that eliminates robot idle time at L_i^f is:

$$\Delta t_i^r = \max(\Delta t_i^f - \Delta t_i^u, 0) \quad (\text{Eq 10})$$

The introduction of the release delay ensures that the robot will not arrive at L_i before Δt_i^f and the calculation of picker waiting time (Eq 6) can be simplified as $\Delta t_{ki}^w = t_{ki}^a - t_i^f$. By combining Eqs (6, 7, 8, 10), Δt_{ki}^w can be expressed as Eq (11).

$$\Delta t_{ki}^w = t_{ki}^a - t_i^f = t_{ki}^c - \Delta t_i^p + \Delta t_i^u - (t + \Delta t_i^f) = t_{ki}^c - \Delta t_i^p - \Delta t_i^r - t \quad (\text{Eq 11})$$

The timelines integrating all temporal items in a schedule of request R_i served by F_k are shown in Fig. 7.

Fig. 7 Timelines of request R_i served by robot F_k

At any time t , Δt_i^p and Δt_i^r are obtained from $(L_i, \Delta t_i^f)$, which are independent from the schedule. Hence, minimizing the sum of the waiting times of all generated requests ($\sum_1^M \sum_1^N \Delta t_{ki}^w$) is equivalent to minimizing the sum of completion times, $\sum_1^M \sum_1^N t_{ki}^c$ of all requests in $\mathcal{S}^R$. A feasible schedule solution includes two parts: the *assignments* of all generated requests $\mathcal{S}^R$ to robots $\mathcal{S}^F$, and the *schedules* of each robot F_k . An $M \times N$ matrix $\mathbf{Z}$ represents the assignments of $\mathcal{S}^R$ on $\mathcal{S}^F$. Each element z_{ki} of $\mathbf{Z}$ is 1 if request R_i is served by robot F_k ; otherwise, z_{ki} is 0. The schedule for each robot F_k is composed of a list of dispatching tuples. Each dispatching tuple includes the dispatching location L_i and the dispatching time t_{ki}^d .

The mathematical model of predictive scheduling of crop-transport robots is expressed with equations (11-18), which define a minimization problem with associated constraints:

- (1) Schedules should consider all the generated requests $\mathcal{S}^R$ (Eq 12).
- (2) Each request is assigned only to one robot (Eq 13).
- (3) The robot is not dispatched until the release constraint of scheduled request (Eq 16).
- (4) The robots start from the active collection station and end with the same one in the planning schedule (Eq 17).

- (5) Each robot serves only one picker when dispatched, and no preemption is allowed, i.e., a robot dispatched to a request R_i cannot be re-assigned to another request R_j (Eq 19).

$$\min \sum_{k=1}^M \sum_{i=1}^N t_{ki}^C$$

s. t.

$$\sum_{k=1}^M \sum_{i=1}^N z_{ki} = N \quad (\text{Eq 12})$$

$$\sum_{k=1}^M z_{ki} = 1 \quad (\text{Eq 13})$$

$$t_{ki}^C = t_{ki}^d + \Delta t_i^p \quad (\text{Eq 14})$$

$$\Delta t_i^r = \max(\Delta t_i^f - \Delta t_i^u, 0) \quad (\text{Eq 15})$$

$$t_{ki}^d \geq t + \Delta t_i^r + \Delta t_k^A \quad (\text{Eq 16})$$

$$\Delta t_i^p = 2\Delta t_i^u + \Delta t^L + \Delta t^{UL} \quad (\text{Eq 17})$$

$$\Delta t_i^u = \frac{D_i}{v_r} \quad (\text{Eq 18})$$

$$\forall R_i, R_j \in \mathcal{S}_{\mathcal{R}}, \text{ and } i \neq j, [t_{ki}^C - \Delta t_i^p, t_{ki}^C) \cup [t_{kj}^C - \Delta t_j^p, t_{kj}^C) = \emptyset \quad (\text{Eq 19})$$

4. Exact and Approximate Solution Methods

The above formulated parallel machines scheduling problem with release dates (Nessah et al, 2007, 2008) is a variant of the well-known job shop problem (JSP) (Lawler, 1993). Following symbol notations defined by Lawler et al. (1993), the problem can be expressed as $Pm|r_i|\sum C_i$ where Pm represents identical parallel machines, r_i means that the i th job cannot be processed until its release time, and $\sum C_i$ represents that the objective criterion is to minimize the sum of the completion times of all jobs. Du et al. (1991) proved that this problem is NP-hard in a strong sense and hence the optimal solution cannot be obtained in polynomial time. In this paper, an exact algorithm, Branch and Bound search (BAB) (Nessah, et al., 2007) was implemented to get the optimal solution of the formulated problem at the cost of long computation time. Meanwhile, a sub-optimal - but faster - approximation algorithm was also implemented, because scheduling must be performed in real-time in the field with limited compute resources, and deterministic scheduling is an important component of stochastic predictive algorithms that incorporate uncertainty, such as the scenario-based method (Bent et al., 2004). The BAB algorithm converges to the global optimal solution of each modelled problem, while the efficient approximate algorithm, converges to a sub-optimal solution; both were applied and compared. The scheduling algorithms were implemented in C++ and a Python wrapper (Cython) was used to package the algorithm into a Python version that was embedded in the simulator.

4.1 Exact Branch and Bound (BAB)

The BAB algorithm computes a systematic enumeration of the candidate solutions using state-space search (Land et al., 1960). Bounds of the optimized function combined with the value of the current best solution enable the algorithm to “prune” part of candidate search branches to avoid exploring parts of subtrees. During the search process, the lower bound (possible best value of the objective function) of

each node is calculated, and if the lower bound is higher than the current best solution, the node and its subtree will be ignored. Hence, the efficiency of the BAB algorithm depends on the accurate and efficient estimation of the lower bound of the subspace in the searching process.

In this paper, two lower bounds were applied by relaxing certain constraints. One lower bound was obtained by relaxing the release constraints of jobs. When the release time of each job is removed, all jobs can be processed immediately. The optimal solution to this no-release-constraint problem was optimally solvable by the policy of Shortest Processing Time (SPT) (Conway et al., 1967), which means that the machines serve jobs with the shortest process time first. The second lower bound was achieved by allowing job splitting and preemptive dispatching, which converted the problem to $Pm|pmtn, r_i| \sum C_i$ (pmtn means that preemptive processing is allowed in the scheduling). The optimal solution to this problem was achieved by the policy of the Shortest Remaining Processing Time (SRPT) (Nessah et al., 2007): at any time, a released job with the shortest remaining process time is simultaneously processed on all the available machines. The processing is interrupted if another job becomes available with a processing time strictly shorter than the remaining processing time of the job in the process.

4.2 SRPT-convert approximation algorithm

Approximation algorithms compute approximate solutions to NP-hard problems with provable guarantees on the distance – given some metric - of the returned solution to the optimal one. The approximation of such algorithms is always guaranteed to be within a multiplicative or additive factor of the optimal solution even in the worst cases. In this work, the efficient CONVERT algorithm was used (Phillips et al., 1998) to solve the original $Pm|r_i| \sum C_i$ problem. CONVERT firstly relaxes the original problem by allowing preemption and job-splitting, thus turning it into a $Pm|pmtn, r_i| \sum C_i$ problem, which can be solved in polynomial time. Then, the solution of the relaxed problem is adjusted, to generate a solution to the original problem.

5. Implementation of Harvest Simulator

Simulator software was developed to simulate robot-aided strawberry harvesting based on the hybrid systems model presented in Section 2. This simulator constitutes a significant extension and adaptation of the simulator developed by Seyyedhasani, Peng, Jang & Vougioukas, (2020a) to incorporate the predictions of requests and the predictive scheduling of robots. The architecture of the simulator is shown in Fig. 8. The simulator is initialized with the geometrical description of the strawberry field (furrow endpoints, split line, collection station locations), the picker crew and robot team parameters, and the initial locations of pickers, robots and active collection station. The “Picker operations” and “Robot operations” modules implement the coupled hybrid system models of the pickers and robots, respectively. The “Crop, crew & collection station distribution” module updates the status of each furrow (harvested/unharvested/currently harvested) and the active collection station, and also calculates the sequence of furrows picked by the crew (after harvesting from a furrow, a picker moves to the furrow of the closest unharvested bed). During simulated harvesting, the picker states are input to the “Tray-transport request prediction” module that computes predicted transport requests, and the predicted requests and current robot states are used by the “Predictive scheduling” module to compute robot schedules and output dispatch commands to the robots.

Fig. 8 Architecture of integrated harvesting simulator and predictive scheduling system.

The simulator uses a global time variable t to represent the current time of the harvesting activity; time starts at $t = 0$ s and increases by Δt (0.5s was used). Before the start of each tray picking, the harvesting parameters ($v_p, v_w, \Delta t^{pick}$) are sampled randomly using their respective experimentally derived frequency histograms. The states of pickers and robots are updated at each time step and the simulation terminates when the entire field block is harvested. There are four pre-allocated collection stations on each half of the split block, and at any point in time, only one collection station is active (the one closest to the crew). This procedure is standard practice in commercial harvesting and reflects what was done when the harvesting data were collected.

Fig. 9 presents a snapshot of the visualization of the robot-aided harvesting process. The currently active collection station is shown as a red star symbol. The locations where trays became full (tray-transport request locations) are marked as green hexagons and are exactly the same as the robot pickup locations (marked as golden crosses), since request predictions are perfect.

Fig. 9 Visualization of robot-aiding harvesting simulator in a commercial strawberry field

5.1 Tray-transport request prediction module

As soon as a picker starts picking – and filling the tray – a mechanistic model (Khosro Anjom, & Vougioukas, 2019) is used to predict online the time Δt_k of the tray-transport request (when tray fills up). An estimate, $\widehat{v_{sp}}$, of the mean of the actual moving speed v_{sp} of the picker is also computed from the trajectory of the picker’s cart. The location of the tray-transport request is predicted as $\Delta t_k \times \widehat{v_{sp}}$; if the cart will not become full by the end of the current furrow, the prediction is not inserted in the request set $\mathcal{S}^{\mathcal{R}}$ because it is not possible to predict the next furrow the picker will enter to continue picking and filling the tray. To study the effect of the timing of the availability (to the scheduler) of request predictions, the tray fill-ratio (FR) was used as a proxy of the availability earliness. The FR is defined as the current weight of a tray divided by the tray’s maximum weight (capacity); it is zero when the tray is empty and one when the tray becomes full. The reason that weight was used instead of time is that in real-world operation, the time to fill a tray is unknown until the tray becomes full. However, tray capacity is known in advance (trays are standard), and the weight of each tray is transmitted to the robot scheduling system in real-time; hence, FR can be computed in real-time.

In the simulator, the tray-transport request prediction module has access to the pickers’ states at each timestamp, so the location and time of the tray-transport request can be predicted perfectly as soon as the pickers start picking (at $FR=0$). In the real world, perfect “prediction” is only possible when $FR=1$, which is equivalent to reactive scheduling, since the robots are scheduled when the tray becomes full. To evaluate the effect of prediction earliness on predictive scheduling, the same harvesting operations were simulated for FR values ranging from zero to one.

5.2 Predictive scheduling module

At each time step, the predictive scheduling module receives and checks the updated robot states and tray transport requests. The scheduling algorithm is executed only when there are robots available in

the collection station and new predictive transport requests enter the tray-transport request set $\mathcal{S}^{\mathcal{R}}$, the scheduling module will solve the modeled problem and store the computed schedules in a schedule table. Each schedule consists of the available robot index F_k , the request location $\mathbf{L}_i$, and the dispatch time t_{ki}^d ; the dispatch command is sent to the scheduled robot when actual time equals dispatch time. After a dispatching command is sent, the corresponding request and schedule are removed from $\mathcal{S}^{\mathcal{R}}$ and from the schedule table, respectively.

6. Experimental Design

All simulation results were based on data gathered in a strawberry field block that was harvested manually in the morning of June 26th, 2018, in Santa Maria, CA, USA. The harvesting block consisted of 100 furrows spaced 1.65 meters apart. Each furrow was 100 m long, and the size of the picking crew was 25 pickers, and one worker at the collection station.

6.1 Evaluation Metrics

In both the all-manual and robot-aided strawberry harvesting, productive time per tray – denoted as Δt_i^{ef} – is defined in the same way: it is the time required by a picker to fill the i th tray to its capacity, starting from an empty tray. Productive time includes picking and walking to relocate to a new furrow to resume picking when the tray cannot be finished in the current furrow. Non-productive time per tray – denoted as Δt_i^{fe} – is defined as the time interval required to exchange a full tray with an empty tray. In manual strawberry harvesting, Δt_i^{fe} includes the picker’s walking time to transport the full tray to the unloading station, the waiting time in a queue to deliver the tray and get an empty one, and the walking time required to return to the previous position to resume picking. In contrast, in robot-aided harvesting, Δt_i^{fe} is the sum of the time (Δt_i^w) the picker spends waiting for a robot to arrive in state “WAIT_FOR_ROBOT_ARRIVAL”, plus the time (Δt^L) needed to place the full tray on the robot and take an empty tray from the robot, in state “EXCHANGE_TRAYS”. Δt_i^w is highly dependent on the robot scheduling policy, whereas Δt^L is small and is assumed to be constant. The mean non-productive time, ΔT^{fe} , when N trays are harvested, can be computed by Eq (20):

$$\Delta T^{fe} = \frac{1}{N} \sum_{i=1}^N \Delta t_i^{fe} = \frac{1}{N} \sum_{i=1}^N (\Delta t_i^w + \Delta t^L) \quad (\text{Eq 20})$$

The mean harvesting efficiency, E_{ff} , when harvesting N trays with or without robots, is defined as the averaged sum of ratios of productive time over total time spent for each tray; it is calculated by Eq (21):

$$E_{ff} = \frac{1}{N} \sum_{i=1}^N \frac{\Delta t_i^w + \Delta t^L}{\Delta t_i^w + \Delta t^L + \Delta t_i^{fe}} \quad (\text{Eq 21})$$

The mean waiting time calculated over N trays is given by Eq (20).

$$\Delta T^w = \frac{1}{N} \sum_{i=1}^N \Delta t_i^w \quad (\text{Eq 22})$$

E_{ff} can be used to evaluate the overall performance of all-manual and robot-aided harvesting, and ΔT^w can be used to evaluate the performance of the scheduling algorithm.

6.2 Determination of sufficient number of Monte Carlo runs

Since the parameters (v^w , v^p , Δt^{pick}) of the picker operating states are stochastic variables, harvesting is simulated using a Monte-Carlo approach. The parameters are sampled - before the start of each tray picking - from the experimentally derived frequency histograms that approximate the respective probability distributions (Figures 5-7). Next, the number M of Monte-Carlo repetitions that are needed to achieve a given value of desired relative precision for the waiting time will be derived (Figliola, R.S. et al., 2014).

Let a picker's waiting time for the i th tray be Δt_i^w , and let $\bar{\Delta t}^w$ be the sample mean waiting time for all N trays; $\bar{\Delta t}^w$ is predicted by one execution of the harvest simulation:

$$\bar{\Delta t}^w = \frac{1}{N} \sum_{i=1}^N \Delta t_i^w \quad (\text{Eq 23})$$

The sample standard deviation of the waiting time computed from N trays is:

$$S^w = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\Delta t_i^w - \bar{\Delta t}^w)^2} \quad (\text{Eq 24})$$

The variation in the sample statistics is characterized by a normal distribution of the sample mean values about the true mean. The variance of the distribution of mean values can be estimated through the standard deviation of the sample mean $S^{\bar{\Delta t}^w}$ shown in Eq (25) (Figliola, R.S. et al., 2014).

$$S^{\bar{\Delta t}^w} = \frac{S^w}{\sqrt{N}} \quad (\text{Eq 25})$$

$\bar{\Delta t}^w$ will be different each time the harvest simulation is executed ($\bar{\Delta t}^w$ is a random variable). If harvesting is simulated M times, a pool of $\sum_{i=1}^M N_i$ measurements will be available. The pooled mean $\langle \bar{\Delta t}^w \rangle$ of M datasets is calculated from Eq (26):

$$\langle \bar{\Delta t}^w \rangle = \frac{\sum_{j=1}^M N_j \bar{\Delta t}_j^w}{\sum_{j=1}^M N_j} \quad (\text{Eq 26})$$

The standard deviation $\langle s^w \rangle$ of the data pool is given by Eq (27), where $v_j = N_j - 1$:

$$\langle s^w \rangle = \sqrt{\frac{v_1 (s_1^w)^2 + v_2 (s_2^w)^2 + \dots + v_M (s_M^w)^2}{v_1 + v_2 + \dots + v_M}} \quad (\text{Eq 27})$$

The pooled standard deviation of $\langle \bar{\Delta t}^w \rangle$ is expressed as $[S^{\bar{\Delta t}^w}]$ in Eq (26). $[S^{\bar{\Delta t}^w}]$ represents the absolute precision of the estimation of the true mean ΔT^w :

$$[S^{\Delta t_w}] = \frac{\langle S^w \rangle}{\sqrt{\sum_{j=1}^M N_j}} \quad (\text{Eq 28})$$

Finally, the relative precision $[p^r]$ of our estimation of the true mean is given by Eq (29).

$$[p^r] = [S^{\Delta t_w}] / \langle \Delta t^w \rangle \quad (\text{Eq 29})$$

In this work, a relative precision equal to 1% was deemed adequate. The number M of Monte-Carlo repetitions to achieve 1% relative precision was determined experimentally and is reported in Section 7.2.

7. Experimental Results and Discussion

7.1 Estimation of picker parameters

Distributions of the stochastic picker parameters (v^w , v^p , Δt^{pick}) for manual picking were generated from data collected in a commercial strawberry field during the high-yield season, in the morning of June 28th, 2018. Recorded data included other harvest-related quantities, such as the non-productive time for each tray, Δt_i^{fe} . The field blocks were in Santa Maria, California [34.947, -120.524] and covered approximately 2.58 ha. Flags were placed before harvesting in the picking fields, at known – measured – distances from each other. The activities of the picking crew were recorded using digital cameras placed at appropriate locations. Then, a human observer watched the videos with a timer to record the time intervals when pickers crossed consecutive flags. Walking speeds were estimated by dividing distances over time, and picking times were recorded directly from the chronometer values. The derived frequency distributions for v^p , v^w , and Δt^{pick} are shown in Fig. 10, Fig. 11 and Fig. 12, respectively.

Fig. 10 Frequency histogram of picker walking speed (v^p) during picking (inside state "PICK").

Fig. 11 Frequency histogram of picker walking speed (v^w) while picker relocates carrying the tray and cart (in states "WALK_TO_FURROW_ENTRANCE" and "WALK_TO_FURROW_SPLITLINE").

Fig. 12 Frequency histogram of the pickers' picking time (Δt^{pick}).

7.2 Determination of sufficient number of Monte Carlo runs

As explained in Section 6.2, when more Monte-Carlo runs are executed for the same harvesting scenario, higher accuracy can be achieved for the estimate of true mean $\langle \bar{\Delta t}^w \rangle$. To quantify the relative precision given different times of measurements, one harvesting scenario was simulated that involved six robots, $FR=0.5$, robot speed at 1.5m/s, and the SRPT-CONVERT scheduling algorithm. The relative accuracy $[p^r]$ was computed as a function of increasing number of Monte-Carlo runs. Fig. 13, shows that 100 executions resulted in a relative accuracy of approximately 1%, which corresponds to 0.081s of absolute precision, given that $\langle \bar{\Delta t}^w \rangle$ was 8.1s. Therefore, the evaluated metrics (ΔT^w , ΔT^{fe}) in all our experiments were estimated based on the pooled means of $M=100$ Monte-Carlo runs.

Fig. 13 Relative precision of the estimated mean waiting time $\langle \Delta t^w \rangle$ as a function of the number of simulation repetitions for Monte-Carlo sampling

7.2 Comparison of Exact and Approximate Solutions

The approximate (SRPT) and the exact algorithm (BAB search) were used to compare the resulting picker mean waiting times for the same harvesting configuration. Table 3 shows the relative discrepancy of the approximate solutions compared to the exact ones, for various FRs, and the corresponding mean running times of the harvest simulations of the entire field with the two scheduling algorithms. The largest discrepancy was 6.8%, and in terms of absolute value, it corresponded to a difference of 1.5 seconds in mean waiting time. However, the BAB algorithm required long computation times, and was 26 to 40 times slower than the approximate algorithm.

<i>FR</i>	Picker mean waiting time with SRPT	Picker mean waiting time with BAB	<i>Discrepancy in waiting times between BAB and SRPT (%)</i>
0.3	1.5 min	40 min	2.0
0.5	1.2 min	32 min	3.3
0.7	0.8 min	28 min	6.8
0.8	0.7 min	26 min	3.1
0.9	0.5 min	20 min	2.3

Table 3 Discrepancy between the picker mean waiting time that resulted with the exact scheduling algorithm (BAB) and the approximate algorithm (SRPT)

Since the goals of this work are to investigate changes in performance as the number of robots increases and the FR varies, small errors in absolute waiting times were not deemed as important. On the other

hand, each Monte-Carlo simulation repeats the harvest operation 100 times, and therefore, computational speed was more important than absolute accuracy. Therefore, the approximate SRPT algorithm was used for all simulation results.

7.3 Reactive and predictive schedule performance vs. number of robots

It is expected that as the number of robots (robot-to-picker-ratio) increases and request predictions become available earlier (FR decreases), non-productive time (ΔT_{fe}) and waiting time (ΔT^w) will decrease, causing efficiency to increase. To quantify this behavior, simulations were executed with increasing numbers of robots and FRs. Fig. 14 shows the mean non-productive time as a function of the robot-to-picker-ratio, with FR as a parameter, and robot speed equal to 1.5 m/s. The dotted line shows the baseline, i.e., the mean non-productive time for all-manual harvesting, which was 84.5 s, based on data collected in the field (Section 7.1). The curve at FR=1 corresponds to reactive scheduling based on the Shortest Remaining Processing Time criterion (Sections 4.1, 4.2).

Fig. 14 Mean non-productive time as a function of the number of robots, with different Fill-Ratios (FRs); robot speed is 1.5 m/s. The manual harvesting non-productive time (84.5 s) was measured in the field, with a 25-people commercial harvesting crew.

When only four robots were used, reactive and predictive scheduling did not improve non-productive time significantly. Actually, deploying fewer than four robots led to worse non-productive time, and hence these data points were not included. Introducing five to eight robots decreased non-productive time drastically, and when ten or more robots were used, waiting time was reduced by 64.6% (reactive scheduling) and up to 93.7% (predictive scheduling, with early prediction at FR=0.8).

As expected, when reactive scheduling was used, non-productive time plateaued (at 29.9 s) when more robots (in this case ten) were deployed, because picker waiting time cannot be less than the robot travel time from the collection station to the picker. The same was true for predictive scheduling; however, the availability of request predictions reduced non-productive time down to 5.3 s, when predictions were available early enough (FR=0.8). Even very late access to request predictions (FR=0.95)

improved significantly the non-productive time (15.5 s). It was noticed that when FR was less than 0.8, robot-aided harvest performance did not improve (hence those curves are not depicted); the reason will be discussed in Section 7.4. The curve at FR=0.8 (and even FR=0) plateaued at 5.3 s.

Fig. 15 shows harvesting efficiency (E_{ff}) curves, for all-manual harvesting (black dotted horizontal line), and for robot-aided harvesting with varying number of robots and availability earliness (FR) parameters. The efficiency curves exhibit a “reciprocal” behavior to the non-productive time curves (Δt_i^{fe} is in the denominator of the efficiency terms, in Eq. 19) and the above discussion about non-productive time vs. robot-to-picker ratio applies to them too.

Fig. 15 Mean harvest efficiency as a function of the number of robots, with different FRs; robot speed is 1.5 m/s

Next, an analysis is presented to explain the plateaus of the picker waiting (non-productive) time in reactive scheduling, in Fig. 14. Each picker needs to wait for the serving robot to traverse a finite distance D_i starting from the collection station to the predicted i th full-tray request location. Given enough robots, if reactive scheduling is used, ΔT^w can be approximated to be the mean of this distance divided by robot speed, plus the constant tray exchange time. During the simulations, the distances D_i were recorded. These distances depend on the dimension of the field and on yield distribution. Their histogram is shown in Fig. 16.

Fig. 16 Frequency histogram of the distance D_i from the collection station to the location of the full-tray request; the mean value is 39.74 m

The mean running distance was $\bar{D} \approx 39.74$ m. Since tray loading time (Δt^L) was assumed constant – equal to 5 s, in this study - the value of ΔT^{fe} at the “plateau” - for reactive scheduling - can be estimated from Eq (30).

$$\Delta T_{fe} \approx \frac{\bar{D}}{v^r} + 5 \approx 31.13s \quad (\text{Eq 30})$$

The result matches approximately the mean of non-productive time of the curve labeled as “FR = 1.0”, in Fig. 14. When predictive scheduling is used early enough (FR<0.8) and many robots are available, the robots start traversing D_i early enough to reach pickers before or exactly when the tray becomes full. Hence, the value of the plateau is dominated by the loading time ($\Delta t^L=5$ s).

7.4 Predictive schedule performance vs. availability earliness of request predictions

Fig. 17 shows the relationship between the picker mean waiting time ($\langle \bar{\Delta t}^w \rangle$) and prediction availability earliness (FR), when robot speed is 1.5 m/s. As it can be seen in the figure, $\langle \bar{\Delta t}^w \rangle$ was almost independent of FR, when FR was smaller than approximately 0.8, but it started increasing when FR became larger than this value; the rate of increase became larger as FR got closer to one.

Fig. 17 Mean waiting time, $\overline{\Delta t}^w$, as a function of FR, with different the number of robots; robot speed is 1.5 m/s

To explain this behavior, we introduce the concept of the “earliness” of a predicted request R_i , at the instant it enters the requests set $\mathcal{S}^{\mathcal{R}}$. Given R_i , the one-way travel time from the collection station to the request location L_i^f , is Δt_i^u , which can be calculated by dividing the corresponding travel distance D_i by the robot speed v^r . A request is characterized as “early” if it meets the condition $\Delta t_i^u \leq \Delta t_i^f$ when entering $\mathcal{S}^{\mathcal{R}}$, i.e., when the robot travel time to the predicted location of the request is shorter than the remaining time, Δt_i^f , needed to fill the tray. When a request is “early” and a robot is available to be dispatched immediately, the corresponding picker will not need to wait for the arrival of a robot at all; instead, the robot may have to wait for the picker to finish. If $\Delta t_i^u > \Delta t_i^f$, then R_i is a “late” request, because even if a robot is available to serve R_i immediately, that picker will still need to wait for the robot, for a time interval equal to $(\Delta t_i^u - \Delta t_i^f)$. The value of Δt_i^u depends on robot speed, and the value of Δt_i^f is determined by FR (if FR=1, $\Delta t_i^f = 0$; if FR=0, $\Delta t_i^f = \Delta t^{pick}$, else $0 < \Delta t_i^f < \Delta t^{pick}$).

Based on the scheduling model presented in section 3, a robot will not be dispatched to a request until the request’s release constraint is satisfied. Therefore, early requests, with an FR value below a certain FR threshold, cannot change the current schedule, and thus will not affect the performance of the system (see Appendix for the formal proof, and derivation of the threshold equation (Eq. A.8). We can calculate this threshold for our example, as follows. Since the robot speed is 1.5m/s, $(\Delta t_j^u)_{max}$ can be calculated by the largest possible one-way transport distance, which corresponds to the distance when the collection station is at the center of the headland, and the tray becomes full at the most distance corner of the field from the collection station. Given the dimensions and geometry of the field, and the spacing and dimensions of the beds and furrows, this distance is equal to 71 m. Given that the mean picking time is 275.5s (from the histogram of Fig 12), the FR threshold, according to (Eq A.8) is 0.83. ΔT^w would be

expected to start increasing significantly when FR is above this threshold, a behavior that is seen in Fig 18.

As mentioned above, the robot's one-way travel time Δt_i^u for request R_i depends on robot speed v_r ; higher speeds result in smaller one-way travel times and $(\Delta t_i^u)_{max}$ also increases. Hence, FR threshold increases. In Fig. 19, one can see that the turning points of FR^t (start increasing point for wait time) of multiple plots does shift right (increase) as the robot speed increases. These turning points of FR^t are listed into the second column in Table 4. FR^t are nearly consistent with the FR thresholds estimated from Eq. A.8. Given different robot velocities, one can estimate the FR thresholds shown in third column in Table 4. FR thresholds gave an instructive earliness for predictive module to generate the predictive requests.

Fig. 19 Mean picker waiting time ($\overline{\Delta t}^w$) as a function of FR, at various robot speeds; robot-to-picker ratio is 6/25

Robot speed (m/s)	FR^t	FR thresholds	Discrepancy (%)
1.0	0.75	0.74	1.3%
1.2	0.8	0.79	1.3%
1.5	0.85	0.83	2.4%
1.7	0.87	0.85	2.3%
2.0	0.90	0.87	3.3%

Table 4 FR^t from the simulation results, $\widehat{FR}^t$ estimated from Eq 29 and their discrepancy percentage

Based on the above results, we conclude that FR does not affect predictive scheduling performance significantly, as long as it is below a certain threshold.

8. Summary and conclusions

In this paper, strawberry harvesting under the assistance of tray-transporting robots was investigated. In particular, dynamic predictive robot scheduling was modeled and implemented, assuming perfect prediction of the locations and times of the next tray-transport requests (deterministic predictions). Also, the influence of the earliness of prediction availability on scheduling performance was studied. The study was based on a harvesting simulator that modeled human pickers and transport robots and utilized manual harvesting model parameters estimated from data collected during harvesting of commercial strawberry fields.

Experimental results showed that to achieve significant reduction in picker mean non-productive time and harvest efficiency, the robot-to-picker ratio had to be above 1:5 (5 robots for 25 pickers); deploying fewer than four robots led to worse non-productive time. When ten or more robots were used, non-productive time was reduced by 64.6% (reactive scheduling) and up to 93.7% (predictive scheduling) with respect to all-manual non-productive time. The corresponding efficiency increases were 15% and 24%. Reactive dispatching ($FR = 1$) performed always much worse than deterministic predictive scheduling ($FR < 1$), because robot travel time to the pickers contributed significantly to picker waiting times. Also, when the robot-to-picker ratio was larger than approximately 1:3 (8 robots for 25 pickers) the waiting time and efficiency plateaued, regardless of how early the prediction was available to the scheduler (i.e., how small FR was). The reason is that, if a robot is always available to serve a predicted request, the picker mean waiting time is the sum of mean travel time plus the tray exchange time, which are both constant; when predictions are made very early, waiting time is lower-bounded by the tray exchange time.

When the robot/picker ratio is up to 1/4 under the predictive scheduling policy *SRPT-Convert*, the non-productive time is reduced around 40% to the reactive scheduling and over 70% compared to manual harvesting. The influence of request prediction earliness, FR, on the predictive scheduling was also analyzed and we found that FR starts affecting the performance of the predictive scheduling after a FR threshold is reached. The FR threshold can be estimated in advance, given a specific harvesting configuration.

9. Appendix

Here, we show that early requests, with an FR value below a certain threshold, cannot change the current schedule, and thus will not affect the performance of the system.

Let $\mathcal{S}^P$ be the set of all the next tray-transport requests. The size of $\mathcal{S}^P$, $|\mathcal{S}^P|$ is equal to Q , the number of pickers. Given an FR value, the set $\mathcal{S}^R$ of the predicted next tray-transport requests is a subset of $\mathcal{S}^P$ ($\mathcal{S}^R \subseteq \mathcal{S}^P$). When FR is 0, $\mathcal{S}^R$ is equal to $\mathcal{S}^P$, at any time instant. Let $\mathcal{S}^{R'}$ be set of requests that have not been predicted yet, i.e., they are not available to the scheduler; $\mathcal{S}^{R'}$ is the complement set $\mathcal{S}^P \setminus \mathcal{S}^R$. The size of $\mathcal{S}^{R'}$ depends on FR. When FR is too small, some of the early requests - with too late release times - don't need to be considered at the instant when the scheduler dispatches the robot to the request whose release constraint has been reached. At a dispatching instant, the scheduler decides to schedule robot k to request R_i . We assume this dispatch command will not be affected by ignoring any requests in

$S^{R'}$, so for $\forall R_j \in S^{R'}$, sum of wait time of schedule $\{R_j, R_i\}$ must be larger than sum of wait time of schedule $\{R_i, R_j\}$ for robot k . We can get wait times for these two possible schedules in Table A1, based on Eq 9.

Schedule	Wait time of R_i	Wait time of R_j	Sum of wait times
$\{R_j, R_i\}$	$\Delta t_j^p + \Delta t_j^r + \Delta t_i^u - \Delta t_i^f$	0	$\Delta t_j^p + \Delta t_j^r + \Delta t_i^u - \Delta t_i^f$
$\{R_i, R_j\}$	$\Delta t_i^u - \Delta t_i^f$	Δt_i^p	$\Delta t_i^u - \Delta t_i^f + \Delta t_i^p$

Table A1. Wait times of two possible schedules: $\{R_j, R_i\}$ and $\{R_i, R_j\}$

The sum of wait times of schedule $\{R_j, R_i\}$ must be larger than that of $\{R_i, R_j\}$; therefore:

$$\Delta t_j^p + \Delta t_j^r + \Delta t_i^u - \Delta t_i^f \geq \Delta t_i^u - \Delta t_i^f + \Delta t_i^p \quad (\text{Eq A.1})$$

The expression can be simplified further:

$$\Delta t_j^p + \Delta t_j^r \geq \Delta t_i^p \quad (\text{Eq A.2})$$

Substituting $\Delta t_j^r = \Delta t_{pick}^i(1 - FR_j) - \Delta t_j^u$ into Eq A.2, we get:

$$\Delta t_j^p + \Delta t_{pick}^i(1 - FR_j) - \Delta t_j^u \geq \Delta t_i^p \quad (\text{Eq A.3})$$

where FR_j is the fill ratio of request R_j . Using Eq 15 for R_j and R_i , Eq A.3 can be written as:

$$\Delta t_j^u + \Delta t_{pick}^i(1 - FR_j) \geq 2\Delta t_i^u \quad (\text{Eq A.4})$$

Solving for FR_j results in Eq A.5.

$$FR_j \leq 1 - \frac{2\Delta t_i^u - \Delta t_j^u}{\Delta t_{pick}^i} \quad (\text{Eq A.5})$$

This inequality must always exist, even when the right-hand side is at minimum, i.e.:

$$FR_j \leq \left(1 - \frac{2\Delta t_i^u - \Delta t_j^u}{\Delta t_{pick}^i}\right)_{min} \quad (\text{Eq A.6})$$

Approximating $\Delta t_i^u \approx \Delta t_j^u$ and $\Delta t_{pick}^i \approx \overline{\Delta t^{pick}}$, we get Eq A.7.

$$FR_j \leq \left(1 - \frac{\Delta t_j^u}{\overline{\Delta t^{pick}}}\right)_{min} \quad (\text{Eq A.7})$$

When Δt_j^u takes its maximum value, the right-hand side is minimized:

$$FR_j \leq 1 - \frac{(\Delta t_j^u)_{max}}{\overline{\Delta t^{pick}}} \quad (\text{Eq A.8})$$

Therefore, when a request in $S^{R'}$ meets Eq A.8, it can be ignored at the dispatching instant.

Acknowledgments

Part of this work was funded by NIFA-USDA Grant 11423498, which supported the project “FRAIL-bots: Fragile cRop hArvest-aiding mobiLe robots” under the National Robotics Initiative; partial funding was also provided by NIFA Hatch/Multi-State Grant 1001069.

References

- Baker, K.R., Trietsch, D., 2013. Principles of sequencing and scheduling. John Wiley & Sons.
- Baptiste, P., Jouglet, A., Savourey, D., 2008. Lower bounds for parallel machine scheduling problems. *International Journal of Operational Research*, 3(6), pp.643-664.
- Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6), pp.977-987.
- Bertsimas, D.J., Van Ryzin, G., 1993. Stochastic and dynamic vehicle routing with general demand and interarrival time distributions. *Advances in Applied Probability*, 25(4), pp.947-978.
- Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G. and Weglarz, J., 2019. Handbook on scheduling. Springer International Publishing.
- Bolda, M. P., Tourte, L., Murdock, J., Sumner, D., 2016. Sample costs to produce and harvest strawberries. Available at <https://coststudies.ucdavis.edu/current/>. Accessed 15 August 2019.
- Conway, M. and Maxwell, W., 1967. *Miller, Theory of Scheduling*. Reading: Addison Wesley.
- Du, J., Leung, J.Y., Young, G.H. 1991. Scheduling chain-structured tasks to minimize makespan and mean flow time. *Information and Computation*, 92(2), pp.219-236.
- Figliola, R.S., Beasley, D.E., 2014. *Theory and design for mechanical measurements*. John Wiley & Sons.
- Seyyedhasani, H., Peng, C., Jang, W.J. and Vougioukas, S.G., 2020a. Collaboration of human pickers and crop-transporting robots during harvesting–Part I: Model and simulator development. *Computers and Electronics in Agriculture*, 172, p.105324.
- Seyyedhasani, H., Peng, C., Jang, W.J. and Vougioukas, S.G., 2020b. Collaboration of human pickers and crop-transporting robots during harvesting–Part II: Simulator evaluation and robot-scheduling case-study. *Computers and Electronics in Agriculture*, 172, p.105323.
- Ichoua, S., Gendreau, M. Potvin, J.Y., 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40(2), pp.211-225.
- Jang, W.J., 2018. Investigation on the Harvest-aid Robot Scheduling Problem and the Implementation of Its Simulation Platform. M.Sc. Thesis. University of California, Davis.
- Khosro Anjom, F., Vougioukas, S. G., Slaughter, D.C., 2018a. Development and Application of a Strawberry Yield-Monitoring Picking Cart. *Computers and Electronics in Agriculture*. 155: 400-411.
- Khosro Anjom, F., Vougioukas, S.G., Slaughter, D.C., 2018b. Development of a linear mixed model to predict the picking time in strawberry harvesting processes. *Biosystems Engineering*, 166, pp.76-89.
- Khosro Anjom, F., Vougioukas, S.G., 2019. Online prediction of tray-transport request time using mechanistic grey box models for improved scheduling of robotic strawberry harvest-aids. *Biosystems Engineering*, 188, pp.265-287.
- Land, A. H., Doig, A. G., 1960. "An automatic method of solving discrete programming problems". *Econometrica*. 28 (3). pp. 497–520.
- Lawler, E.L., Lenstra, J.K., Kan, A.H.R. and Shmoys, D.B., 1993. Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science*, 4, pp.445-522.
- Lu, X., Sitters, R.A., and Stougie, L., 2003. A class of on-line scheduling algorithms to minimize total completion time. *Operations Research Letters*, 31(3), pp.232-236.
- Nessah, R., Chu, C., Yalaoui, F., 2007. An exact method for Pmsds/ri/ $\sum i=1nCi$ problem. *Computers & operations research*, 34(9), pp.2840-2848.

735 Nessah, R., Yalaoui, F., Chu, C., 2008. A branch-and-bound algorithm to minimize total weighted completion time
736 on identical parallel machines with job release dates. *Computers & Operations Research*, 35(4), pp.1176-1190.
737 Phillips, C., Stein, C., Wein, J., 1998. Minimizing average completion time in the presence of release dates.
738 *Mathematical Programming*, 82(1-2), pp.199-223.
739 Ritzinger, U., Puchinger, J. and Hartl, R.F., 2016. A survey on dynamic and stochastic vehicle routing problems.
740 *International Journal of Production Research*, 54(1), pp.215-231.
741 Vougioukas, S.G., Spanomitros, Y., Slaughter, D., 2012. Dispatching and routing of robotic crop-transport aids for
742 fruit pickers using mixed integer programming. In 2012 Dallas, Texas, July.
743 USDA REEIS 2013. [https://reeis.usda.gov/web/crisprojectpages/1000459-nri-small-frail-bots-fragile-crop-harvest-](https://reeis.usda.gov/web/crisprojectpages/1000459-nri-small-frail-bots-fragile-crop-harvest-aiding-mobile-robots.html)
744 [aiding-mobile-robots.html](https://reeis.usda.gov/web/crisprojectpages/1000459-nri-small-frail-bots-fragile-crop-harvest-aiding-mobile-robots.html). Accessed, December 1, 2019.