Computers and Electronics in Agriculture 103 (2014) 75–81

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Harvest-order planning for a multiarm robotic harvester

B. Zion [a,*], M. Mann [b], D. Levin [a], A. Shilo [a], D. Rubinstein [b], I. Shmulevich [b]

[a] Institute of Agricultural Engineering, Agricultural Research Organization, The Volcani Center, Bet Dagan, Israel
[b] Faculty of Civil and Environmental Engineering, Technion-Israel Institute of Technology, Haifa, Israel

ARTICLE INFO

Article history:
Received 26 July 2013
Received in revised form 14 February 2014
Accepted 18 February 2014

Keywords:
Robotic harvest
Multiarm robot
Harvest planning
Melon

ABSTRACT

A multiarm robotic harvester is being developed for two-dimensional crops such as melons. A number of Cartesian manipulators, mounted in parallel on a rectangular frame, traverse laterally across the crop bed as the frame moves along it. The robotic arms reach down to pick melons and place them on adjacent lateral conveyors. The coordinates of the fruits to be harvested are assumed to be known prior to harvest so that the robot gets a bank of targets in local coordinates. In this paper, we describe the algorithms developed and used to plan the assignment of melons to be harvested by each of a number of arms in a collaborative way so that the maximum number of fruits will be harvested by a given number of arms. Under practical kinematic conditions, the fruits and the manipulators' capabilities can be modeled as a task of coloring an interval graph, and a greedy algorithm known to produce an optimal solution for a *k colorable sub-graph problem* is used. Under faster manipulator performance, an approximation algorithm based on heuristics and a local search was shown to produce near-optimal harvest assignments. The algorithms are used to facilitate the design of the robot using simulations of the effects of robot speed, number of arms, manipulator's lateral acceleration and fruit handling time on the harvest. The simulations enable economic optimization of the design of such robotic harvesters, taking into account the costs of robotic arms, labor and operation time and the value of the crop.

1. Introduction

One of agricultural engineers' main objectives is to reduce labor in agriculture, due to a drastic reduction in the availability of workers for agriculture worldwide. Full or partial mechanization and automation have been successful in almost all fields and crops. However, many laborious practices and operations are still conducted manually. Of these, the most complex in terms of economically feasible automation is the harvesting of perishable and sensitive fruit.

Advances in the automation of industrial operations have inspired agricultural engineers to develop robots for various agricultural tasks. However, despite the high degree of mechanization and automation in agriculture and the relatively intensive scientific and commercial activity in this field in the past three decades, very few robots have evolved beyond the research stage (Kapach et al., 2012; Sakai et al., 2008), and none are widely used in open fields. The robotic harvesters' limited success is due mainly to the complexity of the context—the terrain, environment (e.g. dust, humidity, temporal and spatial variability in lighting) and mission (e.g. variability, selectivity, throughput requirements) – which results in low fruit-picking success ratios or operations that are too slow to be economically relevant. A comprehensive review of all attempts to develop robots for agricultural operations is beyond the scope of this paper, but a few are described herein to illustrate the difficulty involved in achieving viable application.

Early studies on robotic harvesting were reported in the 1980s in Japan (Kawamura et al., 1984). A robot was mounted on a small four-wheel car which moved among the tomato plants to be harvested. The end effector consisted of a suction pad that attached to a fruit, separating it from neighboring fruit. Two parallel plate fingers held the fruit at its peduncle and picked it off. The robot managed to pick 91% of 23 fruits at various stages of maturity and the picking cycle took 15 s (Monta et al., 1998). Murakami et al. (1999) developed a robotic cabbage harvester; under field testing, only 40–71% of the mature cabbage heads were picked and it took 32 s to pick one head of cabbage. Umeda et al. (1999) developed a robot for harvesting watermelons using a long truss manipulator operated from a side furrow and equipped with a vacuum gripper. It managed to pick 10 of 15 fruits weighing up to 13 kg. Muscato et al. (2005) described a research project aimed

* Corresponding author. Address: Institute of Agricultural Engineering, Agricultural Research Organization, The Volcani Center, Bet Dagan 50250, Israel. Tel.: +972 3 9683450; fax: +972 3 9604704.
E-mail address: boazz@volcani.agri.gov.il (B. Zion).

http://dx.doi.org/10.1016/j.compag.2014.02.008
0168-1699/

at developing a robotic orange harvester. The picking cycle took 8.7 s per fruit and they concluded that the robot's performance was not good enough to replace manual picking. Foglia and Reina (2006) developed a robot for harvesting radicchio. In laboratory tests, the average harvest cycle took 6.5 s (including identification, approach, cutting and picking a radicchio head). Baeten et al. (2008) tested a tractor-mounted robot designed for harvesting apples. It required stabilization of the robot in its position in front of the tree (using two hydraulic cylinders) and all-around cover over the robot and the tree. It was able to detect and harvest 80% of the apples with a cycle time of 8–10 s.

In the 1990s, a robotic melon harvester was developed at the Institute of Agricultural Engineering of The Volcani center, Israel (Edan et al., 2000). It consisted of a mobile rectangular platform drawn by a tractor on which a Cartesian robotic manipulator was mounted. The manipulator could access any location across the crop bed from within the frame of the platform which traveled above it. A gripper was mounted at the end of the manipulator (Wolf et al., 1990), which was designed to enclose a melon between two rings, detach it from the vine and place it on one of two longitudinal conveyors mounted along either side of the platform. A gray-scale computer vision system was developed to detect melons in front of the platform and guide the gripper toward them. The optimal manipulator trajectory and sequence of melons to harvest was defined by the traveling salesman algorithm. Under certain conditions, the robot was stopped to enable harvesting clusters of melons. Field tests with two cultivars (approximately 100 fruits each) showed that 94% and 91% of the respective melons were reached by the robot arm and 86% were successfully picked. An average of 20% false detections and picks occurred. The average cycle time at a forward speed of 0.02 m/s and manipulator speed of 0.75 m/s was 15 s. Considering the very slow speed of the robot (which was limited by the melon identification system), limited performance and the fact that melons have to be selectively harvested, the robot was not economically viable. Later, the performance of a melon harvester with a single Cartesian arm and two arms arranged in three different configurations (sequential, parallel and tandem) was simulated (Edan and Miles, 1993). The tandem configuration was the fastest, followed by the parallel and sequential modes and the single arm for all actuator speeds evaluated. However, in their tandem configuration, the second arm harvested the melons that the first left behind, rather than following a pre-arranged collaborative order.

Thus the concept of completely autonomous agricultural field robots failed so far by and large to become widely spread and new approaches have been sought in recent years to improve their performance. One approach is to combine human workers and robots synergistically. Hwang and Kim (2003) suggested a system for remotely operating a robot in a greenhouse for various operations, including pruning, watering, pesticide application, and harvest of watermelons. Bechar and Edan (2003) found that the collaboration of a human operator and robot applied to images acquired in melon fields increased target detection from 80% (robot only) to 94% and reduced the time required for detection by 20%.

In the present study, we suggest another approach as an intermediate stage toward the final goal of fully autonomous robotic melon harvesters: separating the sensing system from the robot. Assuming the harvest targets (melons in this case) are mapped in the field prior to harvest (by, for example, an accurate GPS) and the robot gets a bank of targets in local coordinates, its speed of operation is no longer limited by the sensing method, but by its mechanical capabilities and the fruit's sensitivity. Separate operations allow the strongest link-the robotic harvester-to operate independently from the weakest link-the sensing. In addition, such a robot can use multiple arms to speed up harvest. This approach can be used when target locations are fixed in space and independent of each other (which is not the case, for example, in citrus trees, where a fruit is picked, its branch changes position due to the lighter load), and in particular for crops which can be modeled as two-dimensional (e.g. melons, watermelons, pumpkins, etc). If the precise two-dimensional coordinates of the targets are known, the robot can get to the targets and harvest them in a preplanned order at any time of day-or even night.

Herein we describe the robot layout and present the algorithms used to plan the assignment of melons to be harvested by each of a number of arms. This is done collaboratively to ensure that the maximum number of fruits will be efficiently harvested by a minimal number of arms. The success of such an approach might pave the way to a viable melon robot harvester which would drastically reduce the 70 working days required to manually harvest a single hectare of melons (Levkovitch and Kaplan, 2010).

2. Materials and methods

2.1. The robot

The robot under development consists of a 2-m wide rectangular frame which spans the melon bed (Fig. 1). A number of Cartesian manipulators, 3 DOF each, can be mounted in parallel on the frame, such that they can traverse over the row, reach down to pick a melon, lift it and place it on an adjacent lateral conveyor. Each lateral conveyor serves a manipulator and conveys the melons picked by that manipulator to one of two longitudinal conveyors, which move all harvested melons to a platform at the back where they are received and packed. Placing a lateral conveyor next to each robotic manipulator can drastically reduce its pick-and-place cycle time as compared to a setup which uses only longitudinal conveyors (Edan et al., 2000). The manipulator can move laterally relative to the robot direction and up and down to reach melons and lift them. It cannot move in the direction of the robot's progress (only a short displacement to place a picked melon on the conveyor). At the present study the robot moves at a constant speed along the row (either self-propelled or towed). The assumption is that the robot position and heading are known (using, for example, RTK_GPS) and updated at a high enough frequency and precision for the melon coordinates to be known and mapped before the robot is sent to harvest them. The distance between the parallel manipulators has no effect on the robot's functionality and is kept minimal while still allowing free manipulator movement. The idea is that each of the robotic arms is assigned specific melons to harvest in a collaborative manner that will result in the maximum number of melons being harvested using a given number of arms under given kinematic conditions.

2.2. The problem: creating harvest assignments for a multiarm robot

Suppose N melons need to be harvested from a crop bed by a k-arm robot. Each arm can sequentially reach melons along the field and pick them (hereafter termed 'path'). Once an arm has picked a melon, it can or cannot reach another melon, depending on the robot's kinematic parameters (forward speed-Vy, maximum allowed lateral acceleration of the arm-a_x, and time taken to pick a melon-T), place it on a conveyor and get back to a 'ready-to-pick' position ('handling time' hereafter). Assigning harvest orders to the k arms so that under specific kinematic conditions the harvest will be optimal is a combinatorial problem. Like the multiple Salesman Problem which looks for optimal routes for a fleet of vehicles that need to serve a number of customer, it is NP hard (Mitrovic-Minic and Krishnamurti, 2002) A straightforward way of solving the problem would be to compute all possible paths for a single arm and then search for the combination of k paths $\{P_1, P2,\ldots,P_k\}$ that

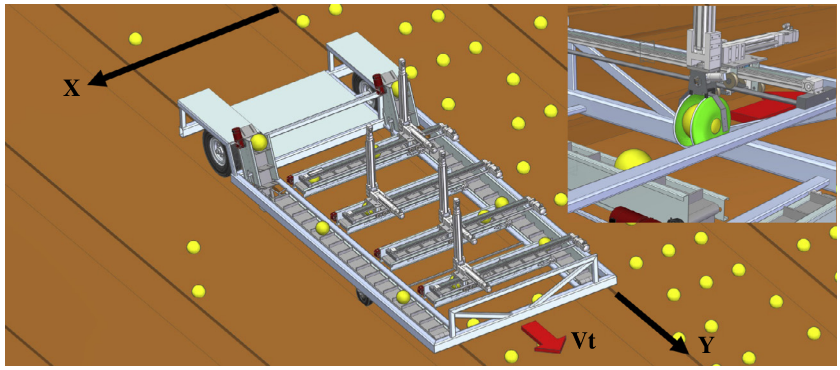

Fig. 1. Layout of a 4-arm robotic harvester and a zoom in view of the manipulator and the gripper. Each arm is served by a lateral conveyor belt onto which it places picked melons. The lateral conveyors move the melons to the longitudinal conveyors, which convey them to a platform at the back, where they are received and packed.

yields the maximum number of harvested melons. Harvest assignments to the arms can then be derived from these k paths. However, this combinatorial task requires exponential time to solve. For example: a field with only 64 melons, spread at a density of 4 melon/m^2 and harvested by robotic arms with a sideways acceleration of 5 m/s^2 and a melon handling time of 5 s, can generate more than 300 possible paths. For a 5-arm robot, the number of 5-path combinations which have to be compared in the search for the optimum is on the order of 10^{12} and it becomes impossible to compute in a reasonable time.

2.3. A solution: modeling the melon field as an interval graph

Consider ripe melons grown on a crop bed that is 2 m wide and L m long and their coordinates in the world are given. Using the robot's kinematic settings (forward speed, lateral acceleration and melon handling time) one can calculate all the coordinates an arm can reach from any given location. Clearly, once the arm picks a melon, it can reach another melon ahead of it only if the time it takes to traverse the lateral distance between them is shorter than (or equal to) the time it takes the robot (which progresses at a constant speed) to cover the longitudinal distance AND longer than the melon-handling time (during which the gripper is occupied lifting a melon, placing it on a conveyor and getting back to a ready-to-pick position). These kinematic conditions section the field ahead of an arm location into zones in which melons are, or are not accessible (Fig. 2). If all of the possible connections between melons are drawn, a graph of 'nodes' (melons) connected by 'edges' (symbolizing all possible robot arm paths from melon to melon according to the kinematic conditions) is generated (Fig. 2a). By converting this graph into one in which only the disconnected melons (those which cannot be sequentially accessed by the same robot arm) are connected by edges, a complement graph is generated (Fig. 2b). Such a graph can be treated by techniques and algorithms used in graph coloring. If the robot arms are considered 'colors', graph theory can be used to assign optimal harvest orders to k robotic arms, in the same way that graph-coloring algorithms are used to color the nodes of such graphs so that no two connected nodes share the same color (since connected melons in Fig. 2b–d cannot be picked by the same arm). This is known as *the maximum k-colorable subgraph problem* (Yannakakis and Gavril, 1987). In the following section we show that under some (practical) kinematic conditions, the graph becomes an interval graph and the optimal solution can be adopted from known rapid algorithms used to color such graphs.

An interval graph $G = (V,E)$, where V is a set of vertices from which the corresponding set of intervals (edges, E) begin, describes intersections between intervals on the real line (Gilmore and Hoffman, 1964; Yannakakis and Gavril, 1987). A graph G such as Fig. 2b represents an interval graph if and only if its vertices V can be arranged sequentially ($v1$, $v2$, ..., vn) along the real line (using the longitudinal coordinate y as the real line) in a way that if vi *adj* vk (i.e. connected vertices), it implies that vi *adj* vj for $i < j < k$ (Weisstein, 2013). To check if this condition for such a graph to represent an interval graph is met, we consider the robot's direction of motion as a real line and sequentially order the melons according to their y-coordinate in the field. If the handling time is very short (e.g. 0 s), then the field is sectioned (clear and dashed regions) by the curve generated for the forward and lateral motions of the robot and arm, respectively (Fig. 2b). This sectioning pattern shows that a melon further down field is connected to (0,0) while two melons closer to it are not. This situation can occur even when the handling time T is 1.5 s (Fig. 2c) and it violates the *iff* condition which defines an interval graph. However, if the handling time is longer than the time it takes an arm to traverse the width of the bed (and thus the position at which the arm is back at a ready-to-pick position is further downfield than the end of the inaccessible region below the orange curve (Fig. 2d), then the field is sectioned into rectangles because the handling time is the limiting time. In the rectangle closer to the arm position (up to the dashed blue line in Fig. 2d), none of the melons can be picked, and beyond that rectangle (further downfield than the dashed blue line), all of the melons can be reached and picked. This pattern of field sectioning conforms to the condition that makes the connections graph an interval graph.

If the manipulator is limited to a lateral acceleration of 1 or 5 m/s^2, the time it takes it to traverse a 2 m bed width is 2.83 or 1.26 s, respectively. In both examples, this time is shorter than the practical handling time currently required for picking the fruit, lifting it off the ground, placing it on a conveyor and getting back to a ready-to-pick position (which we estimate to be at least 3 s). Therefore, for a robot which operates under realistic and practical kinematic conditions, assignment of harvest order to the different arms can be planned using algorithms used to realize interval graphs at $O(n)$ time.

One such algorithm is the greedy algorithm which is known to produce an optimal solution of interval graph coloring (Yannakakis and Gavril, 1987). One can use *k-colorable* procedures to find the number of arms needed to harvest all melons. But to determine whether a given number of robotic arms is economically-optimal and not only mathematically-optimal requires calculating the maximum number of melons which can be harvested using k robotic arms (because, for example, if k arms harvest 99% of the melons, there may be no justification for using $k + 1$ arms to pick an extra 1%). We used a greedy algorithm to solve the *maximum k-colorable subgraph problem* operated on interval graphs. At each node of the graph, the algorithm checks which colors are not adjacent to

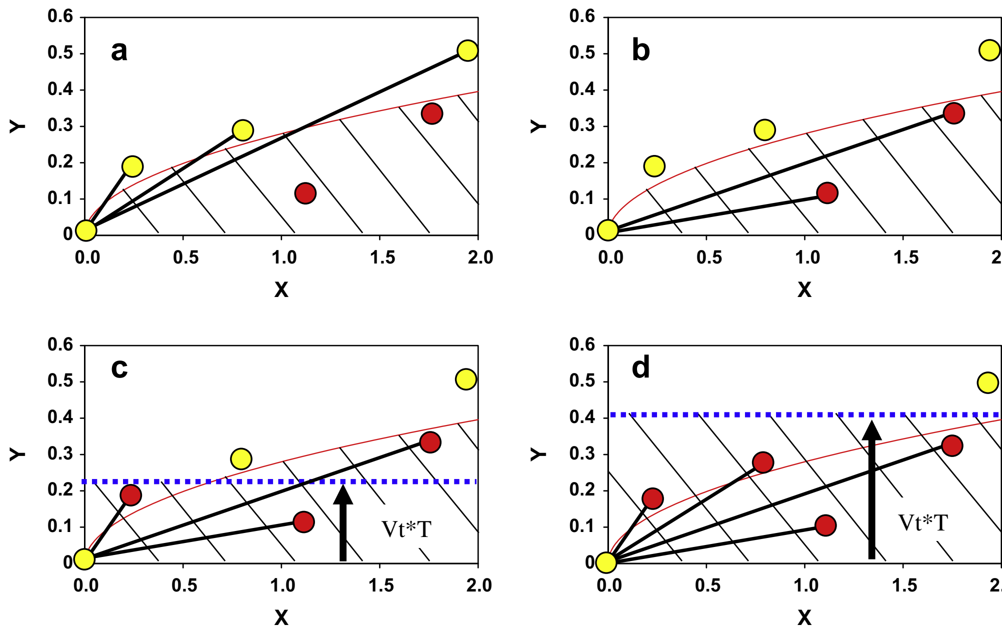

Fig. 2. Melon connectivity graph (a) in a small part (0.6 m long) of a 2-m wide field and its complement graph (b). Yellow melons are accessible from (0,0) and red are not. In the complement graphs (b–d), inaccessible melons are connected. The orange curve separates the field into two regions (clear and dashed) in which melons can or cannot be reached from (0,0), respectively, according to the kinematic setup. In the examples shown, the robot speed *Vt* = 0.14 m/s, manipulator acceleration *a* = 1 m/s^2 and melon handling time *T* = 0 s in (a, b), *T* = 1.5 s in (c) and *T* = 3 s in (d). For presentation simplicity only connections from (0,0) are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

it; of these, it assigns the color whose forward end of the interval is closest to the node being colored.

2.4. An alternative algorithm: 'taxis dispatcher'

Where the problem cannot be modeled as the mission of coloring an interval graph (e.g. if the handling time is reduced to 1 s or less) and thus the greedy algorithm described above cannot be implemented to produce an optimal work plan (WP, *k* lists of melons to be harvested by the *k* arms), another approach has to be used to assign missions to the arms. Combinatorial problems with no exact optimal solution are usually handled by algorithms which approximate the solution in reasonable time. Such is the 'taxi dispatcher' algorithm developed for our robotic harvester. The name is borrowed from the role of a dispatcher at a taxi station who gets requests for taxis from passengers at different locations and checks which taxis are available to fulfill each request. Similarly, stepping one melon at a time in the order of their location along the field, for a given melon *i*, the algorithm checks which of the *k* robotic arms can reach and harvest it (according to their current positions and the kinematic setup). As opposed to the greedy algorithm mentioned in Section 2.3, which assigns the closer available color (arm) to the next node (melon), here all options are considered and for each of the arms that can pick this melon, a new possible WP is generated. Therefore, at melon *i*, a maximum of *k* new WPs can stem from any given WP generated up to the previous melon. The process continues until the last melon is checked.

At each melon *i*, after all possible new WPs have been generated, the concurrent arm positions in all possible WPs are compared. If sets of *k* positions are found to be identical up to a permutation, the superior WP (the one which harvested the most melons to that point) is kept, and all other WPs are omitted. A permutation occurs if in different WPs, the same *k* melons are harvested by *k* arms but in a different order. As an example, consider a robot with 3 arms: if in one WP arm 1 is assigned to melon 36, arm 2 to melon 37, and arm 3 to melon 38 and in another WP, arm 1 is assigned to melon 37, arm 2 to melon 38 and arm 3 to melon 36, then the positions of the arms are considered identical up to a permutation in terms of the robot's ability to harvest melons ahead of the robot.

Under some kinematic conditions, the computation is feasible but rather long. For example, in a field with 1024 melons, spread at a density of 4 melon/m^2 and harvested by a robot with 8 arms with a lateral acceleration of 1 m/s^2, a forward speed of 0.5 m/s and melon-handling time of 2 s, it took approximately 150 s to compute the best WPs which harvest only 76% of the melons. However, when the field is long, the number of melons is large, and the kinematic conditions facilitate high harvest rates, the number of WPs generated can be very large and the computation load increases exponentially.

To further reduce the computation load, the effect of the initial positions of the robot's arms on the harvest of a given field and the characteristics of WPs generated by the algorithm were studied. A set of melon coordinates was randomly generated using a uniform distribution and a given density ρ. An additional set of *k* random points was generated as the initial positions of *k* arms in front of that field. The algorithm computed the best WPs to harvest the melons and repeated itself 10 times with the same melon coordinates but different initial positions. The differences between the 10 repetitions were studied by comparing the best WPs generated by each of them, and by operating a running window of 16 melons along the field, in which the harvest ratio was calculated (out of 16 possible melons) and compared between repetitions. The results showed that the influence of the arms' initial positions is minute and limited to the beginning of the field. In fact, in each of the tests conducted in a field of 1024 melons, a robot with 4 arms, 5 m/s^2 lateral acceleration, 0.04–0.14 m/s forward speed, and 4–5 s handling time, the algorithm converged to an identical best WP (different best WP for different kinematic conditions) from the third melon on. Differences between the WPs generated from different initial positions sometimes occurred but only for the first two melons. For example, for a 6-arm robot, at 0.5 m/s forward speed, 1 m/

s^2 acceleration of the arms and 1 s handling time, the best WP harvested 1620 melons (out of 2048) and the worst 1618. The difference between these WPs was generated for the first two melons in the field.

Stepping from one melon to the next, the algorithm splits the WPs generated at each step into all possible WPs according to the arms' ability to reach the next melon. However, most of these WPs are inferior to the final best WP. Noting that the positions of the arms at the end of each step are initial positions for the next step, and since the initial position test showed that their effect is minute and limited to the nearest few melons, one can neglect inferior WPs and carry on to the next step with only the best WPs. This is a local search which carries on with a few options, rather than just one, to increase the chances of finding the final optimal WP.

To verify this hypothesis and determine the effect of the number of WPs carried on from one step to the next, the algorithm was tested repeatedly while moving from step to step with only the 1, 2, 10, 100 or 500 best WPs. The harvest ratios were compared and showed that for 2, 3, 4, 6, or 8 arms, robot speeds of 0.04–0.14 m/s, arm acceleration of 5 m/s^2 and 4 or 5 s melon-handling times, there was no difference in harvest ratios if a single best WP or 500 superior WPs were carried to the next step. When the handling time was set to 0 s, carrying only a single superior WP from step to step reduced the harvest by 33 melons relative to keeping two WPs (938 vs. 971 melons harvested out of 1024), and keeping 10 WPs or more resulted in 973 melons harvested. Thus, the conclusion from the simulation was that the computation time of the WP can be drastically reduced to a few seconds without practically compromising the harvest quality by filtering out many inferior WPs at every local search step along the field and carrying only 10 superior WPs to the next step.

Since 'sacrificing' a melon (skipping it intentionally even though it can be reached) might enable harvesting more than one other melon which otherwise would not be reachable, the algorithm was modified to accommodate this option: in addition to the options of harvesting a melon by each of the arms that can reach it, the option of skipping it was added. As a result, a maximum of $k + 1$ new WPs (and not k as before) could stem from any given WP as the computation progressed to the next melon. Results showed that for very short handling times (close to 0 s), allowing intentional skipping of melons improved the harvest. For example, for a 4-arm robot with a forward speed of 0.5 m/s, lateral acceleration of 5 m/s^2 and handling time of 0 s, the modified algorithm harvested 925.5 melons (averages of 10 repetitions) compared to 922.3 melons harvested before the modification. When the handling time was longer than 1 s, no difference was found.

The complete algorithm is summarized in the following pseudo code:

(1) consider next melon in the field
(2) check all current WPs for arms that can pick it
(3) extend all WPs:
  (a) assign each of the accessible arms to pick this melon
  (b) assign "0" to this melon- not to be picked
(4) compare all new WPs for arms' positions permutations:
  (a) discard inferior WPs with permuted positions
(5) compare remaining WPs by their harvest rate:
  (a) discard all WPs but the best ten
(6) check if this was the last melon:
  (a) if YES – select best WP and END
  (b) if NO – GOTO step 1

To test how well this algorithm approximates the optimal WP, it was first compared to the results of the optimal greedy algorithm which is known to produce an optimal solution under kinematic conditions which conform to the interval graph definition. Under a wide set of robotic and kinematic setups (Vy = 0.04–0.14 m/s, a = 2–5 m/s^2, T = 2–5 s and k = 3–8 arms), there was only a single case in which the greedy algorithm outperformed the taxi dispatcher algorithm, by a single melon out of 1024. Though that proves that the taxi dispatcher algorithm does not produce optimal WPs, it indicates that for all practical purposes it is a valid algorithm which is near-optimal. Since the *k-colorable subgraph problem* is NP-hard (unless the problem can be modeled as an interval graph), the taxi dispatcher algorithm was used to generate practical close-to-optimal WPs under conditions which could not be modeled as interval graphs.

3. Results: robot performance

3.1. Effects of operation variables on the harvest

The algorithms described in Section 2 were used to simulate the effects of all robotic operation variables (robot forward speed, number of arms, their lateral acceleration, melon-handling time and crop density) on the harvest. Each simulation took from less than a second to a few seconds, depending on the set of variables tested. The effect of number of robotic arms at various robot forward speeds and melon-handling times is shown in Fig. 3. If the crop density is 4 melon/m^2, then using a 4-arm robot will limit the forward speed to 0.02 m/s if all melons are to be harvested. Using 8 arms will enable a forward speed of 0.06 m/s and even 0.1 m/s if the harvest ratio can be slightly compromised to 99.1%. If the melon-handling time can be reduced to 2 s, a robot with 8 arms can travel at 0.14 m/s and harvest 99.9% of the melons if the manipulator's lateral acceleration is 4 m/s^2 (Fig. 4). The handling time interacts with the forward speed and manipulator acceleration as possible limiting factors of accessibility from one melon to the next. It must be shorter than the time it takes the robot to cover the forward distance between two melons and at the same time, the arm must be able to traverse the lateral distance between them. At very slow robot speeds, a handling time of up to 5 s is short enough and hardly affects the harvest ratio but at higher speeds, it becomes dominant and prevents access to many melons (Fig. 5).

3.2. Proof of synergistic robotic arm collaboration

To demonstrate the collaboration between the robotic arms of this robot design, the following test was conducted: the field was split longitudinally into two or four strips of identical width. Two

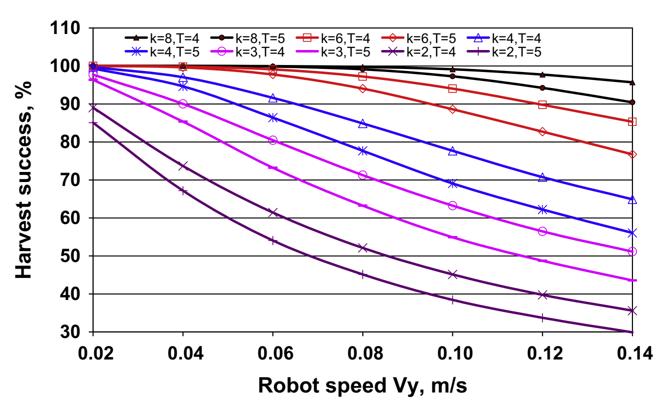

Fig. 3. Effects of number of robotic arms at various robot forward speed and melon-handling times (T = 4 and 5 s) on the harvest at an arm lateral acceleration of 5 m/s^2, and crop density of 4 melon/m^2.

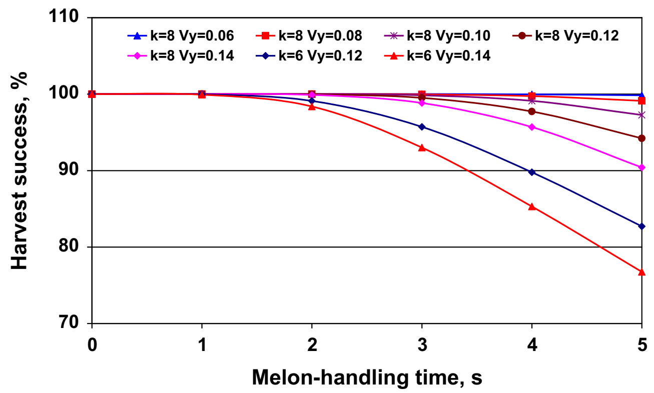

Fig. 4. Effects of melon-handling time and robot speed (Vy) on harvesting by an 8-arm or 6-arm robot (k) with arm lateral acceleration of 4 m/s^2, and a density of 4 melon/m^2.

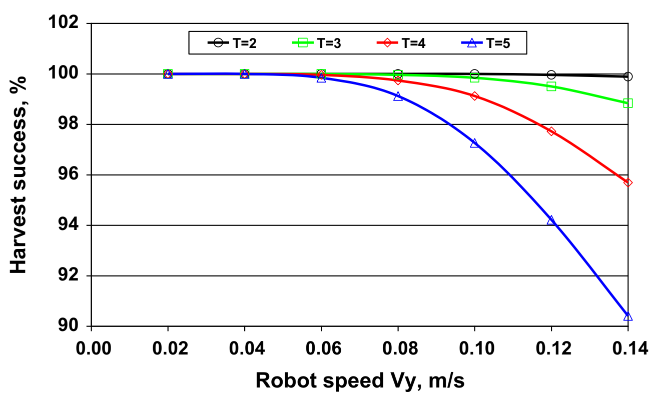

Fig. 5. Effect of melon-handling time (T) and robot forward speed on the harvest ratio of an 8-arm robot with arm lateral accelerations of 4 or 5 m/s^2 (same result), and a density of 4 melon/m^2.

arms of a 4-arm robot were allocated to harvesting one strip and the other 2 arms harvested the other strip (half of the field for each pair of arms). Alternatively, each of the 4 arms was assigned to harvest one of four split strips. In all tests, the same coordinates of 1024 melons were used.

When the field was not split, the 4-arm robot harvested 86.4% of the melons (density of 4 melon/m^2, robot forward speed of 0.1 m/s, arm lateral acceleration of 5 m/s^2 and melon handling time of 3 s). When the field was split into two strips, it harvested 75.7% and when split into 4 strips, only 62.3% of the melons were harvested. Similar trends were found for different setups.

4. Discussion and summary

To date, robotic harvesters have not been widely successful despite the scientific and commercial activity in this field in the past three decades. Their limited success has been attributed mainly to the complexity of the context, namely terrain, environment, and mission, and the crop's value. Robotic harvesters attempting to deal autonomously with these complexities have failed to be economically feasible. One of the concepts emerging in recent years is to deal with this technological evolution gridlock by forming synergistic human–machine collaborations. The approach proposed in this paper for the robotic harvesting of melons and similar crops (e.g. watermelons, pumpkins) is to separate the fruit-identification stage from the harvest stage by mapping the target fruit a priori. This approach enables selective harvesting of the ripe fruits and higher harvest ratios than have been reached to date (due to limitations of computer-vision-based identification and of technological ripeness-sensing capabilities). The fruits can be mapped manually, using accurate GPS or computerized vision systems combined with other sensing technologies for ripeness detection if and when they become sufficiently accurate.

Two algorithms were used to assign missions to each of k arms of a multiarm robotic harvester. In most practical cases, the mission assignment can be modeled as a *k-colorable subgraph problem* from graph theory and a greedy algorithm used to solve the coloring of interval graphs to produce the optimal harvest order assignments for the arms in less than a second. If much faster manipulators will be developed in the future (and the gripper should be such that will not damage the fruits), the taxi dispatcher algorithm can be used to produce near-optimal harvest plans in with very short computation times (up to a few seconds). The two algorithms were used to simulate and demonstrate the effects of robot forward speed, number of arms, their lateral acceleration and melon-handling time on the harvest. For any given melon density and distribution, these algorithms can be used to simulate the harvest and select a desirable robot design (number of arms and operation variables).

The suggested robot design, which enables synergistic collaboration between its arms, resulted in a better harvest ratio than the previously tested approaches of splitting the field into sections and letting each arm harvest a section, or using a tandem configuration in which the second arm harvests melons that the first has left behind (Edan and Miles, 1994). By considering collaboration between all the arms while planning the harvest, an optimal harvest can be achieved. In contrast, Bozma and Kalalioglu (2012) proposed an approach for a multi robot pick-and-place task in which every robot optimizes its own action while considering the actions of only its closest neighboring robots. They found out that the number of items picked off a conveyor belt did not increase with respect to a "selfish" approach in which every robot optimizes its task regardless of other robots. In contrast to previous conclusions (Edan and Miles, 1994), this study shows that using more than two arms increased the harvest ratio and enabled faster robot speeds. Furthermore, limiting the arms to lateral motions and mounting lateral conveyors next to each arm shorten the cycle times normally required to discharge picked melons directly on longitudinal conveyors. In parallel to this work we study a slightly different robot design in which the manipulators are not fixed to the robot's rectangular frame in the direction it progresses (Y in Fig. 1), but rather have 3 DOF. As a result, the lateral conveyors have to be removed and the manipulators place the picked melons on the two longitudinal conveyors (Mann et al., in press-a, in press-b). A comparison of the efficiency of the two designs is planned for the near future. The robot can be drawn by a tractor or self-propelled. In the present work, the assumption was that it progresses at a constant speed. However, in subsequent work, the forward speed will be controlled and the two approaches will be compared.

The algorithms developed and presented here, facilitate economic optimization of the design of such robots taking into account the costs of robotic arms, operation time, labor cost and the value of the crop. This is the scope of a future publication.

References

Baeten, J., Donné, K., Boedrij, S., Beckers, W., Claesen, E., 2008. Autonomous fruit picking machine: a robotic apple harvester. Field Service Robotics 42, 531–539.
Bechar, A., Edan, Y., 2003. Human-robot collaboration for improved target recognition of agricultural robots. Ind. Robot 30, 432–436.
Bozma, H.I., Kalalioglu, M.E., 2012. Multirobot coordination in pick-and-place tasks on a moving conveyor. Robot. Computer-Integrated Manuf. 28, 530–538.
Edan, Y., Miles, G.E., 1993. Design of an agricultural robot for harvesting melons. Trans. ASAE 36, 593–603.

Edan, Y., Miles, G.E., 1994. Systems-engineering of agricultural robot design. IEEE Trans. Syst. Man Cybern. 24, 1259–1265.

Edan, Y., Rogozin, D., Flash, T., Miles, G.E., 2000. Robotic melon harvesting. IEEE Trans. Robot. Automat. 16, 831–835.

Foglia, M.M., Reina, G., 2006. Agricultural robot for radicchio harvesting. J. Field Robot. 23, 363–367.

Gilmore, P.C., Hoffman, A.J., 1964. A characterization of comparability graphs and of interval graphs. Can. J. Math. 16, 539–548.

Hwang, H., Kim, S.-C., 2003. Development of multi-functional tele-operative modular robotic system for greenhouse watermelon. IEEE/ASME Int. Conf. Adv. Intelligent Mech., 1344–1349.

Kapach, K., Barnea, E., Marion, R., Edan, Y., Ben-Shahar, O., 2012. Computer vision for fruit harvesting robots – state of the art and challenges ahead. Int. J. Comput. Vision Robot. 3, 4–34.

Kawamura, N., Namikawa, K., Fujiura, T., Ura, M., 1984. Study on agricultural robot (Part 1). J. Japanese Soc. Agric. Mach. 46, 353–358.

Levkovitch, G., Kaplan, M., 2010. Production cost in labor days in selected labor-intensive agricultural crops. In: Scientist, T.C. (Ed.), Ministry of Agriculture and Rural Development, Israel.

Mann, M.P., Shmulevich, I., Rubinstein, D., Zion, B., 2014. Motion planning of a mobile Cartesian manipulator for optimal harvesting of 2-D crops. Trans. ASABE, in press.

Mann, M.P., Zion, B., Rubinstein, D., Linker, R., Shmulevich, I., 2014. Minimum time kinematic motions of a Cartesian mobile manipulator for a fruit harvesting robot. J. Dynam. Syst. Measurement Control-Trans. ASME, in press.

Mitrovic-Minic, S., Krishnamurti, R., 2002. The Multiple Traveling Salesman Problem with Time Windows: Bounds for the Minimum Number of Vehicles. School of Computing Science. Simon Fraser University, Burnaby, Canada.

Monta, M., Kondo, N., Ting, K.C., 1998. End-effectors for tomato harvesting robot. Artif. Intell. Rev. 12, 11–25.

Murakami, N., Otsuka, K., Inoue, K., Sugimoto, M., 1999. Development of robotic cabbage harvester. (Part 2). Field test by developed hand. J. Japanese Soc. Agric. Mach. 61, 93–100.

Muscato, G., Prestifilippo, M., Abbate, N., Rizzuto, I., 2005. A prototype of an orange picking robot: past history, the new robot and experimental results. Ind. Robot 32, 128–138.

Sakai, S., Iida, M., Osuka, K., Umeda, M., 2008. Design and control of a heavy material handling manipulator for agricultural robots. Auton. Robots 25, 189–204.

Umeda, M., Kubota, S., Iida, M., 1999. Development of "STORK", a watermelon-harvesting robot. Artif. Life Robot. 3, 143–147.

Weisstein, E.W., 2013. Interval Graph. Wolfram MathWorld.

Wolf, I., Bar-Or, J., Edan, Y., Peiper, U.M., 1990. Developing grippers for a melon harvesting robot. American Society of Agricultural and Biological Engineers, St. Joseph, MI, USA.

Yannakakis, M., Gavril, F., 1987. The maximum k-colorable subgraph problem for chordal graphs. Inform. Process. Lett. 24, 133–137.