A Model for Scheduling a Fleet of Autonomous Electric Agricultural Robots

Mateus Vilela Souza, Bruno Bachelet, Thiago Noronha, Loïc Yon, Christophe Duhamel

A Model for Scheduling a Fleet of Autonomous Electric Agricultural Robots

Mateus Souza
Université Clermont Auvergne, CNRS, LIMOS
FR-63000 Clermont-Ferrand, France
mateus.vilela_souza@uca.fr

Bruno Bachelet
Université Clermont Auvergne, CNRS, LIMOS
FR-63000 Clermont-Ferrand, France
bruno.bachelet@uca.fr

Thiago Noronha
Univ. Federal de Minas Gerais
Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte, Brazil
tfn@dcc.ufmg.br

Loïc Yon
Université Clermont Auvergne, CNRS, LIMOS
FR-63000 Clermont-Ferrand, France
loic.yon@isima.fr

Christophe Duhamel
Université Le Havre Normandie
FR-76063 Le Havre, France
christophe.duhamel@univ-lehavre.fr

ABSTRACT

Autonomous and electric robots appeared in agriculture to perform activities on the fields. The management systems dedicated to farms thus need to integrate software for the scheduling of a fleet of such robots. As agricultural systems are by nature subject to unpredictable events, it seems necessary to provide a solution for online scheduling. As a first step toward a global solution, we propose a model for a sub-problem that targets to provide the next route of each robot of the fleet in order to satisfy a subset of the demands for activities on fields emitted by the farmers. In this work, we present a mixed integer linear programming formulation for the sub-problem using the prize-collecting concept called here *Prize-Collecting Robot Scheduling Problem with Time Window and Precedence Constraints*. Specific constraints are introduced to ensure that PC-RSP provides viable solutions for the targeted online process.

KEYWORDS. Agriculture, scheduling, prize-collecting

Topics: PO in the agriculture and environment, Logistics and Transport

1. Introduction

Recent advances in robotics bring autonomous, small and light robots to work in agriculture Duckett et al. [2018]. Their introduction enables improving the efficiency of agricultural operations and reducing the environmental impacts Bochtis et al. [2014], like the decrease of wastes, the focus on areas of great needs and potentially reducing fertilizer costs Blackmore et al. [2005]. The traditional planning methods for agricultural operations need to be supplemented by new scheduling features, like route planning and sequential task scheduling Bochtis et al. [2014] mainly to adapt to the introduction of these robots. Lately, the farmers use *Farm Management Information Systems* (FMIS), sophisticated and complex systems to support production management Fountas et al. [2015].

In this context, we consider a farm composed of plots, and a warehouse where a fleet of robots is stored, set up (their tools, called here *equipments*, can be changed to enable a robot to perform different tasks), and maintained. Each plot represents an area with cultivation activities, and it is assumed here to have single entry and exit points. In each plot, a set of demands for activities that must be performed by the robots is emitted by the farmers. These demands have many requirements, which include: compatible robot - equipment pairing to perform the demand, a demand has to be achieved by only one robot (set up with one equipment only), a time window in which the activity must be performed, precedence constraints between demands, a priority level for each demand, and a preference date for the activity to be started.

There are different types of robots and equipments, and their numbers are limited. Equipments can only be installed or uninstalled at the warehouse, and the list of tools that are compatible with each robot is known. The time to install or to uninstall an equipment on a robot depends on both the types of the robot and the equipment. Robots can be set up with only one equipment at a time and can thus perform only a set of activity kinds once outside the warehouse to satisfy demands. We consider here electric robots operating on batteries. These batteries have to be recharged periodically at the warehouse. Besides, each different *configuration* (*i.e.* a pair of robot and equipment) has a specific battery discharging rate (assumed to be linear here) and performances (speeds to move and work). The energy and time spent by each configuration to execute the demands, and to move between locations, are deterministic and known beforehand (it can be provided by tools like Cariou et al. [2017]).

The decision problem consists here in managing and scheduling the robots, set up with equipment, to perform a set of demands in the farm. It is dynamic, *i.e.* new demands appear on the fly. Moreover, the execution of the demands are subject to unpredictable events, like: breakage of robot or equipment, unfavorable weather, unforeseen demands. Notice that the FMIS should react to these unpredictable events by emitting new demands Bachelet et al. [2021]. Thus, the problem needs to be handled by an online algorithm, because decisions must be made without the full knowledge of the future Karp [1992].

To model this scheduling problem, we propose an online approach, the *Robot Scheduling Problem with Time Window and Precedence Constraints* (RSP-TWP), a part of an ongoing project Belhassena et al. [2021]. RSP-TWP is an iterative approach where the problem is decomposed in a sequence of sub-problems named *Prize-Collecting Robot Scheduling Problem with Time Window and Precedence Constraints* (PC-RSP) presented here. PC-RSP generates a single route for each robot in order to achieve some set of demands. A *route* is the trajectory of a robot (more precisely a configuration) that leaves the warehouse, moves sequentially to several plots to handle some demands before returning to the warehouse. In the global RSP-TWP problem, the PC-RSP sub-problem is called each time a robot returns to the warehouse and recharges its battery, in order

to perform another trip with the same, or another equipment. As input, this problem needs mainly the availability dates of robots and equipments for a new route, and the current set of unscheduled demands.

We present here PC-RSP for which we propose a Mixed Integer Linear Programming (MILP) formulation. The performance of solving this MILP is evaluated for different instances, in order to show how this approach is amenable to be used in the perspective of the global online approach RSP-TWP.

2. Related Work

According to Pinedo et al. [2015], two types of scheduling problems exist: *static* where the schedule is not expected to change much over time, which can be considered to be the case of the PC-RSP problem analyzed in this work, as it only provides the next routes of the robots; and *dynamic* where the schedule is expected to change frequently, which is the case of the global RSP-TWP problem that should deal with unpredictable events.

Allahverdi et al. [2008] propose a review on scheduling problems. Our problem can be considered closely related to the job-shop problem, where the robots are the machines and the demands are the jobs. It is possible to consider in this case that our problem contains setup times, if we consider the travel times of the robots to move between the locations (warehouse and plots). However, our problem could also be considered close to a Vehicle Scheduling Problem (VSP), that is also known as Vehicle Routing Problem (VRP) Cordeau et al. [2002]. The robots are the vehicles that need to be scheduled to satisfy demands.

In our problem, each demand has to be satisfied inside a certain time window. Kontoravdis and Bard [1995] present solutions for a VRP with Time Windows (VRP-TW), but they aim at minimizing the overall traveled distance and the number of used vehicles, and not to satisfy the maximum number of clients, or demands in our case. They propose to solve the problem with a Greedy Randomized Adaptive Search Procedure (GRASP). Tests on real instances were achieved, and the results are competitive in time with exact methods and reached good solutions (*i.e.* that handle all the demands with an acceptable number of vehicles and a reasonable traveled distance) in almost all cases.

Bruglieri et al. [2015] consider a VRP-TW with the management of the energy of the vehicles (E-VRP-TW) dealing with the challenging of robot battery energy administration too. They handle the possibility of partial recharge of the vehicles' batteries and propose a MILP formulation. A Variable Neighborhood Search Branching (VNS-B) has been designed to obtain solutions of good quality in reasonable computational times. Schneider et al. [2014] address the Electric VRP-TW and Recharge Stations (E-VRP-TW-RS). In this problem, there are recharge stations where the vehicles can refill their battery in the path of the route. A hybrid heuristic has been designed that combines Variable Neighborhood Search (VNS) with Tabu Search (TS). The tests showed a high performance and the positive effects of the hybridization of strategies.

The prize-collecting concept guides the design of the objective function of PC-RSP. This approach is notably used for the Prize-Collecting Traveling Salesman Problem (PC-TSP), as presented in Fischetti and Toth [1988]. They search for a route for a vehicle to visit a subset of cities and use the concept of prize to introduce a reward on cities that are visited. PC-TSP is simpler in comparison to our PC-RSP problem, as we need to handle many robots and equipments to find several routes, and we propose to introduce a prize associated to the achievement of a demand. A branch-and-bound algorithm is proposed in Fischetti and Toth [1988] to solve optimally the PC-TSP problem that showed good performances on most of the tested instances.

Stenger [2012] addresses the Prize-Collecting Vehicle Routing Problem with Non-Linear cost (PC-VRP-NL) with the possibility of hiring outsource resources. This work also adapts the strategy of prize-collecting to a real problem, which is similar to our work only in using prizes and in the generation of routes. An integer linear model is presented like a VNS algorithm to solve the problem. After tests on instances adapted from classical VRP instances, it was demonstrated that the proposed VNS strategy exhibited a strong performance. Stenger et al. [2013] can be considered the continuation of Stenger [2012], expanding the scope of PC-VRP-NL to the multi-depot case. This work deals with different aspects that are not taken into account in our PC-RSP problem: a non-linear objective function, multi-depot and the possibility of outsourcing the service to satisfy its customers. It was proposed an Adaptive Variable Neighborhood Search (A-VNS) algorithm to solve the problem, and benchmark instances, both for single-depot and multi-depot. In both cases, it was observed good performance of the proposed algorithm.

The multi-vehicle version of the Time-Dependent Prize-Collecting Arc Routing Problem (TD-PC-ARP) is presented in Black et al. [2013]. This work not only implements the strategy of prizes, but also includes the modeling of time, like in our problem, but without time windows for each demand. Two metaheuristics are presented, one based on VNS and the other one on TS. VNS had better performance on small instances and TS on large ones. Li and Tian [2016] proposes a Two-Level Self-Adaptive Variable Neighborhood Search (TL-SA-VNS) algorithm to address the PC-VRP problem. They deal with the concept of prizes to satisfy clients and the generation of routes, but not with batteries, configuration of robots and equipments as we need to. They propose MILP to generate lower bounds for the problem, and a self-adaptive strategy to analyze the neighborhood to increase the probability of finding better solutions, and thus better upper bounds. The proposed metaheuristic showed better performances (*i.e.* lower gaps on obtained solutions) than Multi-Start Iteration Local Search (MS-ILS), Multi-Start Local Search (MS-LS), and Particle Swarm Optimization (PSO).

Tang and Wang [2006] also addresses the PC-VRP problem, but with new constraints and an objective function that is a combination of factors. The objective takes into account the minimization of the number of used vehicles and the overall traveled distance, and the maximization of the number of visited customers. In our model, the objective considers only one factor, the number of handled demands, pondered by their priority and the proximity of their scheduled starting date with their preference date. This adapted PC-VRP is solved with an Iterated Local Search (ILS) algorithm based on a Very Large-Scale Neighborhood (VLSN) using cyclic transfer. This algorithm is compared with another ILS approach and a Multi-Start based on VLSN. It shows good performances by providing better solutions than the two other ones for the tested instances.

3. MILP Formulation

We propose here a MILP for the PC-RSP problem. Let D be the set of demands for activities ($n = |D|$), R the set of robots and E the set of equipments. The objective is to schedule the next route of each robot in R, while addressing some of the demands in their time window and satisfying the precedence constraints between demands. The robots leave the warehouse fully charged and must return to the warehouse at the end of each route, respecting the battery capacity. Besides, in its current route, the robot is equipped with one single equipment from E that is used to serve all the demands in that route. Each demand has a preferred date to be served and a priority. The objective is to find a schedule where demands are carried out as close as possible to their preferred date, and favoring the demands according to their priority. We propose to model this objective with triangular prize functions.

As PC-RSP is foreseen to be part of an online process, it needs the availability date $a^r \in \mathbb{R}$

of each robot $r \in R$, meaning when robot r will be back at the warehouse and ready for a new route (*i.e.* fully recharged and unequipped) each time it is called. The model also requires the availability date $b^e \in \mathbb{R}$ of each equipment $e \in E$ and an up-to-date set of demands (*i.e.* the yet unscheduled demands in the global online process).

Henceforth, we use the following terminology in this paper. A **plot** is an area of open land, especially one planted with crops or pasture, typically bounded by hedges or fences, with specific entry and exit points. It can also be called *field* or *area*. We consider here a single entry and exit point. The **warehouse** is the place where robots are recharged, and equipment are set up and removed from the robots. A **location** can be a plot or the warehouse. A **farm** is a set of locations, here we consider a single warehouse with several plots. An **activity** is an operation performed by a robot on a plot, such as irrigation, plowing, spraying, sowing, etc. A **demand** is a request for robots to perform an activity on a plot, and describes all the requirements to carry out this activity (e.g. the configurations able to carry out the activity, time window in which the activity must be achieved, etc.). An **equipment** is installed on a robot to make it ready to carry out a given kind of activity. It can also be called a *tool*. A **configuration** is a pair of compatible robot and equipment. A **route** is a circuit starting at the warehouse where a robot is first set up with one equipment and refilled, then sent successively to several plots to perform the same activity to satisfy many demands, and ultimately returns to the warehouse to uninstall the equipment from the robot.

3.1. Parameters

The battery capacity of robot $r \in R$ is given by $c^r \in \mathbb{R}$. The recharging rate of robot $r \in R$ is given by $\rho^r \in \mathbb{R}$. It is assumed that this rate is the same regardless the battery charge. The equipments compatible with robot $r \in R$ are given by $E(r)$ and the robots that are compatible with equipment $e \in E$ are given by $R(e)$.

As a demand is associated with a plot, we propose to model a set of demands as a graph where nodes represent demands (and consequently locations). Let $G = (V, A)$ be the corresponding directed graph where $V = D \cup \{0\} \cup \{n+1\}$, where 0 and $n+1$ both represent the (same) warehouse. It is assumed that each route begins at 0 and ends at $n+1$. The set of nodes $A = \{(i,j) : i,j \in D, i \neq j\} \cup \{(0,j) : j \in D\} \cup \{(i,n+1) : i \in D\}$ and $\overline{A} = \{(i,j) : i,j \in D, i \neq j\}$.

The time for robot $r \in R$, if installed with an equipment $e \in E$, to carry out demand $i \in D$ is given by $\delta_i^{re} \in \mathbb{R}$. δ_0^{re} is the time to install equipment $e \in E(r)$ on robot $r \in R$ at the warehouse and δ_{n+1}^{re} is the time to uninstall the equipment $e \in E(r)$ from the robot $r \in R$ at the warehouse. Earliest starting date for serving demand $i \in D$ is $l_i \in \mathbb{R}$, preferred starting date for serving demand $i \in D$ is $f_i \in \mathbb{R}$ and the latest starting date for serving demand $i \in D$ is $u_i \in \mathbb{R}$.

The subset of demands that precede demand $j \in D$ is $P_j \subset D$, *i.e.* a robot can start j only after all demands $i \in P_j$ are achieved. The priority of demand $i \in D$ in comparison with the other demands is given by $\psi_i \in \mathbb{R}$. This is the way to allow the farmer to express the relative importance of the demands.

The set of configurations that can carry out demand $i \in D$ is $C_i \subseteq R \times E$. The configurations that go out of the warehouse (C_0) and back to the warehouse (C_{n+1}) are assumed to be: $C_0 = C_{n+1} = \mathcal{C}$, where $\mathcal{C} = \bigcup_{i \in D} C_i$ is the set of all valid configurations. $C_{ij} = C_i \cap C_j$, for all $(i,j) \in A$. The time for configuration $(r,e) \in \bigcup_{i \in D} C_i$ to move from location $i \in V$ to location $j \in V$ is $d_{ij}^{re} \in \mathbb{R}$. The energy spent by configuration $(r,e) \in \bigcup_{i \in D} C_i$ to move from location $i \in V$ to location $j \in V$ is $e_{ij}^{re} \in \mathbb{R}$. The energy spent by configuration $(r,e) \in \bigcup_{i \in D} C_i$ to carry out demand $i \in D$ is $\epsilon_i^{re} \in \mathbb{R}$. We also set $\epsilon_0^{rt} = 0$ for all $(r,e) \in C_0$, and $\epsilon_{n+1}^{rt} = 0$ for all $(r,e) \in C_{n+1}$.

3.2. Decision Variables (Discrete)

We define the decision variables x, such that $x_{ij}^{re} = 1$ if configuration $(r, e) \in C_{ij}$ carries out demand $i \in D \cup \{0\}$ and then moves to location $j \in D \cup \{n+1\}$, $x_{ij}^{re} = 0$ otherwise.

3.3. Auxiliary Variables (Continuous)

We define the auxiliary variables z, y and w. $z_i = 1$ if demand $i \in D$ is served, or $z_i = 0$ otherwise. $y^{re} = 1$ if robot $r \in R$ uses equipment $e \in E(r)$ to serve the demands assigned to a route, or $y^{re} = 0$ otherwise. These variables do not need to be integer because their values are induced by x. $w_i^{re} \in \mathbb{R}$ is the date when configuration $(r, e) \in C_i$ starts the activity of demand i. In the case (r, e) does not carry out i, $w_i^{re} = 0$. This is not necessarily the date when r arrives at i, because it might need to wait for the opening of the time window of i to start performing the activity.

3.4. Objective Function

We define a prize function Z_i for each demand $i \in D$ that depends on the date w_i when the activity starts, $w_i = \sum_{(r,e)\in C_i} w_i^{re}$. h_i^- is how much the service of demand $i \in D$ will be advanced relatively to its preferred date, such that $h_i^- = \max(0, f_i - w_i)$. h_i^+ is how much the service of demand $i \in D$ will be delayed relatively to its preferred date, such that $h_i^+ = \max(0, w_i - f_i)$. They are meaningful only if demand i is carried out. The maximum value of Z_i is ψ_i (the priority of demand i) when $w_i = f_i$ (the preferred date) and linearly decreases with w_i moving away from f_i to reach 0 when reaching the limits (l_i and u_i). The objective is thus to maximize the sum of the prize functions Z_i as expressed by (1).

$$\max Z = \sum_{i \in D} \psi_i \left(z_i - \frac{h_i^-}{f_i - l_i} - \frac{h_i^+}{u_i - f_i}\right) \tag{1}$$

3.5. Constraints

$$\sum_{(i,j)\in A} \sum_{(r,e)\in C_{ij}} x_{ij}^{re} = z_i \qquad \forall i \in D \tag{2}$$

$$\sum_{e \in E(r)} y^{re} \leq 1 \qquad \forall r \in R \tag{3}$$

$$\sum_{r \in R(e)} y^{re} \leq 1 \qquad \forall e \in E \tag{4}$$

$$\sum_{(\ell,j)\in A} x_{\ell j}^{re} - \sum_{(i,\ell)\in A} x_{i\ell}^{re} = \begin{cases} y^{re}, & \text{if } \ell = 0 \\ 0, & \forall\, \ell \in D \\ -y^{re}, & \text{if } \ell = n+1 \end{cases} \qquad \forall \ell \in D, \forall (r,e) \in \bigcup_{k \in D} C_k \tag{5}$$

$$\sum_{(i,j)\in A} \sum_{(r,e)\in C_i} (\epsilon_i^{re} + e_{ij}^{re}) x_{ij}^{re} \leq c^r \qquad \forall r \in R \tag{6}$$

$$z_i \geq z_j \qquad \forall j \in D, \forall i \in P_j \tag{7}$$

$$\sum_{(r,e)\in C_i} \left(w_i^{re} + \sum_{(i,\ell)\in A} \delta_i^{re} x_{i\ell}^{re}\right) \leq \sum_{(r,e)\in C_j} w_j^{re} + M a_{ij}(1 - z_j) \qquad \forall j \in D, \forall i \in P_j \tag{8}$$

$$w_i^{re} + \delta_i^{re} + d_{ij}^{re} \leq w_j^{re} + M b_{ij}^{re}(1 - x_{ij}^{re}) \quad \forall (i,j) \in A, \forall (r,e) \in C_{ij} \tag{9}$$

$$l_i \sum_{(i,j)\in A:(r,e)\in C_{ij}} x_{ij}^{re} \leq w_i^{re} \leq u_i \sum_{(i,j)\in A:(r,e)\in C_{ij}} x_{ij}^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{10}$$

$$h_i^- \geq f_i z_i - \sum_{(r,e)\in C_i} w_i^{re} \qquad \forall i \in D \tag{11}$$

$$h_i^+ \geq \sum_{(r,e)\in C_i} w_i^{re} - f_i \qquad \forall i \in D \tag{12}$$

$$w_0^{re} \geq b^e y^{re} \qquad \forall (r,e) \in \mathcal{C} \tag{13}$$

$$w_0^{re} \geq a^r y^{re} \qquad \forall (r,e) \in \mathcal{C} \tag{14}$$

$$x_{ij}^{re} \in \{0,1\} \qquad \forall (i,j) \in A, \forall (i,j) \in A, \forall (r,e) \in C_{ij} \tag{15}$$

$$0 \leq z_i \leq 1 \qquad \forall i \in D \tag{16}$$

$$0 \leq y^{re} \leq 1 \qquad \forall (r,e) \in \mathcal{C} \tag{17}$$

$$h_i^- \geq 0 \qquad \forall i \in D \tag{18}$$

$$h_i^+ \geq 0 \qquad \forall i \in D \tag{19}$$

Constraints to manage the robots and equipments are given by constraints (2) to (4). Constraints (2) require that a demand must be served by at most one robot. Constraints (3), together with (5), ensure that a scheduled robot uses the same equipment to serve all the demands on its route. Constraints (4) manage the availability of each equipment. Constraints to model the routes of the robots, dealing with their energy and the time windows of the demands are given by (5) to (10). The flow conservation constraints (5) impose that the route of each robot begins and ends at the warehouse. Constraints (6) enforce that each robot has enough energy to serve all the demands on its route and to return to the warehouse. Constraints (7) and (8) impose the precedence between demands. Subtour elimination constraints (9) link variables w and x. Constants Ma_{ij} and Mb_{ij}^{re} are defined in Section 3.7. The restrictions on time windows are imposed by (10). They also ensure that any robot waits until the opening of the time window of a demand before starting the activity. The variables h^- and h^+ - used in the objective function - are determined by constraints (11) and (12), respectively. The domain of variables w is also lower bounded by a^r and b^e, any time a robot r or an equipment e is used in a route by constraints (13) and (14), respectively. The domain of variables x, z, y, h^-, and h^+ are defined by constraints (15) to (19).

3.6. Towards Online Scheduling

As explained previously, PC-RSP will be called iteratively during the online process. We propose to add constraints (20) to (26) in order to enforce that each non-scheduled demand can be carried out in the next iteration of PC-RSP (not necessarily all of them simultaneously) if the schedule defined by variables x is performed as expected. For this, the following continuous auxiliary variables are defined: $\lambda^e \in \mathbb{R}$ that is the next date when equipment $e \in E$ will be available (*i.e.* when returning to the warehouse); $\mu^r \in \mathbb{R}$ that is the next date when robot $r \in R$ will be available (*i.e.* when returning to the warehouse); and γ_i^{re}, such that $\gamma_i^{re} = 1$ if configuration $(r,e) \in C_i$ could handle demand $i \in D$ in the next iteration (*i.e.* next time PC-RSP will be called), assuming that the schedule defined by variables x will be performed as expected, and $\gamma_i^{re} = 0$ otherwise.

$$\lambda^e \geq w_i^{re} + \delta_i^{re} + d_{i,n+1}^{re} + \delta_{n+1}^{re} - (1 - x_{i,n+1}^{re}) Mc_i^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{20}$$

$$\mu^r \geq w_i^{re} + \delta_i^{re} + d_{i,n+1}^{re} + \delta_{n+1}^{re} - (1 - x_{i,n+1}^{re})Md_i^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{21}$$

$$\begin{aligned}\xi^r = & \sum_{(0,j)\in A:(r,e)\in C_{0j}} (\epsilon_0^{re} + e_{0j}^{re})x_{0j}^{re} + \sum_{(i,j)\in \overline{A}:(r,e)\in C_{ij}} (\epsilon_i^{re} + e_{ij}^{re})x_{ij}^{re} \\ & + \sum_{(i,n+1)\in A:(r,e)\in C_{i,n+1}} (\epsilon_i^{re} + e_{i,n+1}^{re} + \epsilon_{n+1}^{re})x_{j,n+1}^{re} \qquad \forall r \in R\end{aligned} \tag{22}$$

$$\mu^r \geq w_i^{re} + \delta_i^{re} + d_{i,n+1}^{re} + \frac{\xi^r}{\rho^r} - (1 - x_{i,n+1}^{re})Me_i^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{23}$$

$$\lambda^e + \delta_0^{re} + d_{0i}^{re} \leq u_i + (1 - \gamma_i^{re})Mf_i^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{24}$$

$$\mu^r + \delta_0^{re} + d_{0i}^{re} \leq u_i + (1 - \gamma_i^{re})Mg_i^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{25}$$

$$\sum_{\forall (r,e)\in C_i} \gamma_i^{re} \geq 1 - z_i \qquad \forall i \in D \tag{26}$$

$$\lambda^e \geq b^e \qquad \forall e \in E \tag{27}$$

$$\mu^r \geq a^r \qquad \forall r \in R \tag{28}$$

$$\gamma_{ij}^{re} \in \{0,1\} \qquad \forall (i,j) \in A, \forall (r,e) \in C_{ij} \tag{29}$$

Constraints (20) determine a lower bound of when equipment e will be available after being used in a scheduled route (*i.e.* when it will be available for the next route). Constraints (21) determine a lower bound of when robot r will be available for the next route, considering the time to uninstall an equipment from it. Constraints (22) compute the total energy ξ^r of robot r to perform the scheduled route. Constraints (23) determine a lower bound of when robot r will be available for the next iteration, considering the time to recharge the battery. Constraints (24) bound λ^e if equipment e is able to satisfy demand i in the next iteration. Constraint (25) bound μ^r if robot r is able to satisfy demand i in the next iteration. Constraints (26) ensure that any non-scheduled demand can be carried out in the next iteration, at least by one available configuration. The domains of variables λ and μ are lower bounded by constraints (27) and (28) respectively. And the domain of the variables γ is defined by constraints (29).

3.7. Constants

$$Ma_{ij} = u_i + \max_{(r,e)\in C_i} (\delta_i^{re}) \qquad \forall j \in D, \forall i \in P_j \tag{30}$$

$$\begin{aligned} Mb_{ij}^{re} &> u_j && \forall (0,j) \in A, \forall (r,e) \in C_{0j} \\ Mb_{ij}^{re} &= u_i + \delta_i^{re} + d_{ij}^{re} && \forall (i,j) \in A : i \neq 0, \forall (r,e) \in C_{ij} \end{aligned} \tag{31}$$

$$Mc_i^{re} = u_i + d_{i,n+1}^{re} + \delta_{n+1}^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{32}$$

$$Md_i^{re} = u_i + d_{i,n+1}^{re} + \delta_{n+1}^{re} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{33}$$

$$Me_i^{re} = u_i + d_{i,n+1}^{re} + \frac{c^r}{\rho^r} \qquad \forall i \in D, \forall (r,e) \in C_i \tag{34}$$

$$Mf_i^{re} = \delta_0^{re} + d_{0i}^{re} + \max_{j\in D, \dot{r}\in R:\forall (\dot{r},e)\in C_j} \left(b^e + u_j + \delta_j^{\dot{r}e} + d_{j,n+1}^{\dot{r}e} + \delta_{n+1}^{\dot{r}e}\right) \quad \forall i \in D, \forall (r,e) \in C_i \tag{35}$$

$$Mg_i^{re} = \delta_0^{re} + d_{0i}^{re} + \max_{j\in D, \dot{e}\in E:\forall (r,\dot{e})\in C_j} \left(a^r + u_j + \delta_j^{r\dot{e}} + d_{j,n+1}^{r\dot{e}} + \delta_{n+1}^{r\dot{e}} + \frac{c^r}{\rho^r}\right) \quad \forall i \in D, \forall (r,e) \in C_i \tag{36}$$

Equations (30) to (36) define the constants Ma_{ij}, Mb_{ij}^{re}, Mc_i^{re}, Md_i^{re}, Me_i^{re}, Mf_i^{re} and Mg_i^{re} used in some constraints presented previously. They are set here to have a complementary slack as small as possible in the model.

4. Instances Generation

In order to generate complex instances for the PC-RSP problem, the data of Solomon's instances for the VRP-TW problem (Solomon [1987]) were used, those instances have 100 customers organized in clusters. The instances are grouped in two kinds of clusters, $C1$ and $C2$, where $C1$ instances have clusters more dense than those in $C2$. Customers' time windows are used for the demands and the deposit becomes the warehouse.

The robot set is composed of 3 robots, one of each kind, $R = \{1, 2, 3\}$, and the equipment set is composed of 3 equipments, one of each kind, $E = \{a, b, c\}$. With these two sets, 6 possible configurations are created, $C = \{(1, a), (1, b), (2, a), (2, c), (3, b), (3, c)\}$.

c^r is equal to the vehicle's capacity from Solomon's instances. ρ^r is equal to c^r divided by 8 hours. a^r and b^e are set to 0. $G = (V, A)$ is the graph from Solomon's instances. C_i is randomly built from C, such that $|C_i| = 3$. δ_i^{re} is the service time from Solomon's instances. δ_0^{re} and δ_{n+1}^{re} are set to 5 minutes. d_{ij}^{re} are the distances from Solomon' instances. e_{ij}^{re} is set to 0. ϵ_i^{re} is the weight (demand) of each customer from Solomon's instances. There is no precedence between demands.

In order to compute the preferred date and time window of each demand in the generated instances, it is necessary to build a viable solution (without time window constraints) for the problem, *i.e.* a set of routes for configurations that satisfies all the demands. From this solution, f_i is set to the time a configuration starts to carry out demand i, and randomly l_i is set to $f_i - U[0, 24]$ and u_i to $f_i + U[0, 24]$. ψ_i is randomly set to $U[1, 100]$.

5. Computational Results

Instances	Upper bound	Lower bound	Relative gap	Percentage of satisfied demands
InstancePCRSP-C101	3315.00	688.42	382%	12%
InstancePCRSP-C102	3271.00	197.07	1560%	3%
InstancePCRSP-C103	3310.00	198.25	1570%	4%
InstancePCRSP-C104	3566.00	612.22	482%	9%
InstancePCRSP-C105	3215.50	20.00	15977%	1%
InstancePCRSP-C106	3431.00	152.00	2157%	2%
InstancePCRSP-C107	3375.00	695.32	385%	10%
InstancePCRSP-C108	3245.89	332.08	877%	7%
InstancePCRSP-C109	3324.67	0.00	*	0%
InstancePCRSP-C201	4984.00	39.00	12679%	1%
InstancePCRSP-C202	5200.00	211.00	2364%	6%
InstancePCRSP-C203	5151.00	0.00	*	0%
InstancePCRSP-C204	5559.00	120.67	4507%	3%
InstancePCRSP-C205	5034.00	0.00	*	0%
InstancePCRSP-C206	5539.00	0.00	*	0%
InstancePCRSP-C207	5378.00	76.00	6976%	2%
InstancePCRSP-C208	4972.00	91.00	5364%	1%

Table 1: Results of MILP solving with execution time limit of 300 seconds.

* Not possible to calculate the gap, lower bound set to 0.

Instances	Upper bound	Lower bound	Relative gap	Percentage of satisfied demands
InstancePCRSP-C101	3315.00	918.74	261%	19%
InstancePCRSP-C102	3271.00	699.85	367%	10%
InstancePCRSP-C103	3309.00	439.86	652%	8%
InstancePCRSP-C104	3557.00	1262.27	182%	17%
InstancePCRSP-C105	3215.50	388.00	729%	6%
InstancePCRSP-C106	3431.00	930.28	269%	15%
InstancePCRSP-C107	3364.50	1228.87	174%	20%
InstancePCRSP-C108	3245.89	999.74	225%	19%
InstancePCRSP-C109	3324.00	737.02	351%	13%
InstancePCRSP-C201	4984.00	776.90	542%	17%
InstancePCRSP-C202	5200.00	935.42	456%	17%
InstancePCRSP-C203	5151.00	587.55	777%	10%
InstancePCRSP-C204	5559.00	559.58	893%	10%
InstancePCRSP-C205	5034.00	116.00	4240%	3%
InstancePCRSP-C206	5539.00	668.57	728%	12%
InstancePCRSP-C207	5378.00	210.00	2461%	3%
InstancePCRSP-C208	4972.00	994.34	400%	18%

Table 2: Results of MILP solving with execution time limit of 3600 seconds.

Tables 1 and 2 show results of solving the MILP model for PC-RSP with a time limit of respectively 300 and 3600 seconds. The first field is the name of the instances in Solomon's set. The next two fields are the upper bound and the lower bound, *i.e.* the so far best integer solution found. The fourth field is the relative gap between the two bounds. The last field is the percentage of demands satisfied by the best found solution.

In Table 1, we can observe that it was not possible to find a viable solution for 3 instances in $C2$ and 1 instance in $C1$ within the time limit. In Table 2, a viable solution is found for all the instances within the time limit. The gaps are smaller in $C1$ than in $C2$ in Table 1. And the gaps are mostly smaller in $C1$ than in $C2$ in Table 2. This suggests that the density of the clusters is a factor that makes the problem harder to be solved.

The comparison between Table 1 and Table 2 shows an improvement in the lower bounds with the increase of the time limit of execution. The improvement in the upper bound is very small, indicating that the linear relaxation of the model is not very good. The improvement of the gap of all instances can be observed, influenced mostly by lower bound improvement.

The tests were made on an Intel Xeon E5405 at 2.0 GHz, 16 GB of 667 MHz DDR2 RAM memory, and 4 cores (hyper-threading not activated). CPLEX 20.1 was used for linear program solving.

6. Conclusion

We proposed here a MILP formulation for the problem that we called *Prize-Collecting Robot Scheduling Problem with Time Windows and Precedence Constraints* (PC-RSP). This problem is to find the next route for each robot of a fleet, in order to satisfy some demands for activities on plots in a farm. It is part of a more global problem, the *Robot Scheduling Problem with Time Window and Precedence Constraints Problem* (RSP-TWP) that targets to produce a complete schedule to satisfy all the demands for activities in a given period of time.

To solve this problem, we consider solving PC-RSP repeatedly. Moreover, as agricultural systems are by nature subject to unpredictable events, we have foreseen to also use PC-RSP in a global online scheduling process. It is therefore necessary to solve PC-RSP within a short time, we assess less than 5 minutes.

As one can observe in Table 1, the proposed MILP for PC-RSP does not make possible to get a viable solution for all the instances within this time limit, showing the necessity of studying heuristics in future works to solve PC-RSP. However, from these results, it appears that the structure of the graph representing the locations of the farm has an impact on the difficulty to solve the problem. Studies to improve the model in order to find better upper bounds more rapidly should be done to solve PC-RSP more efficiently with MILP. After these studies, a method to solve PC-RSP for the global RSP-TWP problem will be considered.

Acknowledgments

This work was sponsored by a public grant overseen by the French National Research Agency as part of the ”Investissements d’Avenir” through the IDEX/I-SITE initiative CAP 20-25 (16-IDEX-0001).

References

Allahverdi, A., Ng, C. T., Cheng, T. E., and Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032.

Bachelet, B., Battistoni, P., Bimonte, S., Cariou, C., Chalhoub, G., Coutarel, F., and Tricot, N. (2021). Towards an architecture for online scheduling of autonomous robots in agriculture: Open issues. *International Journal of Smart Vehicles and Smart Transportation*, 4(2):to appear.

Belhassena, A., Battistoni, P., Souza, M., Laneurit, J., Moussa, R., Bimonte, S., Wrembel, R., Abouqateb, M., Cariou, C., Chalhoub, G., AndrÃ©, G., Sebillo, M., and Bachelet, B. (2021). Towards an architecture for agricultural autonomous robots’ scheduling. In *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pages 194–203, Gold Coast, Australia. IEEE Computer Society Press.

Black, D., Eglese, R., and Wøhlk, S. (2013). The time-dependent prize-collecting arc routing problem. *Computers and Operations Research*, 40:526–535.

Blackmore, S., Stout, B., Wang, M., and Runov, B. (2005). Robotic agriculture - the future of agricultural mechanisation? In *European Conference on Precision Agriculture*, volume 5, pages 621–628, Uppsala, Sweden. Wageningen Academic Publishers.

Bochtis, D. D., Sørensen, C. G., and Busato, P. (2014). Advances in agricultural machinery management: a review. *Biosystems Engineering*, 126:69–81.

Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A matheuristic for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 55:89–92.

Cariou, C., Gobor, Z., Seiferth, B., and Berducat, M. (2017). Mobile robot trajectory planning under kinematic and dynamic constraints for partial and full field coverage. *Journal of Field Robotics*, 34(7):1297–1312.

Cordeau, J.-F., Desaulniers, G., Desrousiers, J., Solomon, M. M., and Soumis, F. (2002). Vrp with time windows. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, chapter 7, pages 157–193. SIAM, Philadelphia, USA.

Duckett, T., Pearson, S., Blackmore, S., Grieve, B., Chen, W.-H., Cielniak, G., Cleaversmith, J., Dai, J., Davis, S., Fox, C., et al. (2018). Agricultural robotics: The future of robotic agriculture. Technical report, UK-Robotics and Autonomous Systems (UK-RAS) Network.

Fischetti, M. and Toth, P. (1988). An additive approach for the optimal solution of the prize collecting traveling salesman problem. *Vehicle Routing: Methods and Studies*, 231:319–343.

Fountas, S., Carli, G., Sørensen, C. G., Tsiropoulos, Z., Cavalaris, C., Vatsanidou, A., Liakos, B., Canavari, M., Wiebensohn, J., and Tisserye, B. (2015). Farm management information systems: current situation and future perspectives. *Computers and Electronics in Agriculture*, 115:40–50.

Karp, R. M. (1992). On-line algorithms versus off-line algorithms: How much is it worth to know the future? In *IFIP Congress*, volume 12, pages 416–429, Madrid, Spain. International Computer Science Institute.

Kontoravdis, G. and Bard, J. F. (1995). A grasp for the vehicle routing problem with time windows. *ORSA Journal on Computing*, 7(1):10–23.

Li, K. and Tian, H. (2016). A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. *Applied Soft Computing*, 43:469–479.

Pinedo, M., Zacharias, C., and Zhu, N. (2015). Scheduling in the service industries: an overview. *Journal of Systems Science and Systems Engineering*, 24(1):1–48.

Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Stenger, A. (2012). The prize-collecting vehicle routing problem with non-linear cost: integration of subcontractors into route design of small package shippers. In *1st International Conference on Operations Research and Enterprise Systems*, volume 1, pages 265–273, Vilamoura, Algarve, Portugal. SciTePress.

Stenger, A., Schneider, M., and Goeke, D. (2013). The prize-collecting vehicle routing problem with single and multiple depots and non-linear cost. *EURO Journal on Transportation and Logistics*, 2(1-2):57–87.

Tang, L. and Wang, X. (2006). Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. *The International Journal of Advanced Manufacturing Technology*, 29:1246–1258.