Netw Spat Econ (2010) 10:321–343
DOI 10.1007/s11067-009-9120-0

Planning for Agricultural Forage Harvesters and Trucks: Model, Heuristics, and Case Study

Victor Blanco · Luisa Carpente · Yolanda Hinojosa · Justo Puerto

Published online: 4 December 2009

Abstract In this paper we study an actual problem proposed by an agricultural cooperative devoted to harvesting corn and grass. The cooperative uses harvesters for harvesting the crop and trucks for carrying it from the smallholdings to the landowners' silos. The goal is to minimize the total working time of the machinery. Therefore, the cooperative needs to plan both the harvesters and trucks routing. This routing problem simultaneously incorporates the following characteristics: time windows, nested decisions, processing times required to service each facility and the fact that facilities must be visited in clusters. A binary integer linear programming model is proposed to solve this

This research was undertaken with the financial support of *Ministerio Español de Ciencia y Tecnología*, Junta de Andalucía and FEDER through projects SEJ2005-07637-C02-02, CONS-C2-0060 (i-MATH), MTM2007-67433-C02-01, P06-FQM-01366 and the technical support of CESGA, *Centro de Supercomputación de Galicia*.

V. Blanco · J. Puerto (✉)
Departamento de Estadística e IO, Facultad de Matemáticas,
Universidad de Sevilla, 41012 Sevilla, Spain
e-mail: puerto@us.es

V. Blanco
e-mail: vblanco@us.es

L. Carpente
Departamento de Matemáticas, Facultade de Informática,
Universidade da Coruña, 15071 A Coruña, Spain
e-mail: luisacar@udc.es

Y. Hinojosa
Departamento de Economía Aplicada I, Facultad de Ciencias Económicas y Empresariales,
Universidad de Sevilla, 41018 Sevilla, Spain
e-mail: yhinojos@us.es

problem. However, since approaches dealing directly with such formulation lead to considerable computation times, we propose a heuristic alternative solution approach for the problem. The heuristic is applied to the case of the cooperative “Os Irmandiños” with a large number of landowners and smallholdings. We report on extensive computational tests to show that the proposed heuristic approach can solve large problems effectively in reasonable computing time.

Keywords Location-routing · Application: agriculture · Binary linear programming · Heuristic algorithms · Tabu search

1 Introduction

One way to improve regional development in remote areas is to organize individual producers into associations of producers so as to get efficiency in cost operations and stronger position in the markets. In recent years, several regional governments in Spain have promoted the creation of agricultural cooperatives as a tool to obtain better socio-economic regional conditions. This strategy is simultaneously combined with the application of new information technologies and logistics, production planning and marketing tools.

One of these agricultural cooperatives, named “Os Irmandiños” is located in Ribadeo, a Galician city in northwestern Spain (Fig. 1). “Os Irmandiños”

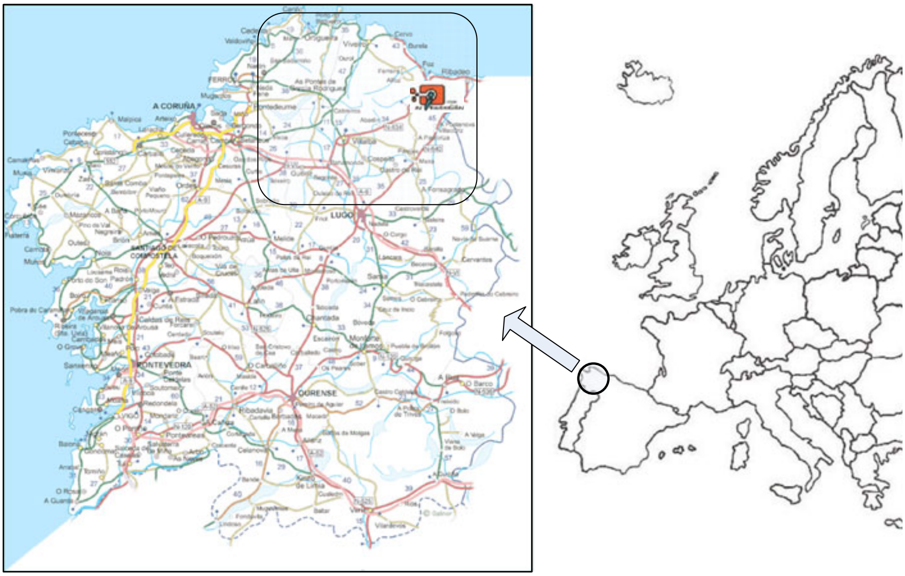


Fig. 1 Location of the cooperative “Os Irmandiños”

has a department of machinery which has to manage the harvesting of grass and corn twice a year, in the spring and autumn seasons, respectively. This cooperative has about 1100 partners in the Spanish provinces of Asturias, La Coruña and Lugo and 350 of them require the services of the the machinery department. The cooperative owns five self-propelled (SP) forage harvesters in its machinery fleet and for its normal operation it rents a number of trucks depending of the season. The SP forage harvesters have a large working area and the five harvesters can harvest together about 5000 ha per year in 3000 fields located at a distance up to 90 km from the cooperative facilities. These machines are the most expensive elements of the fleet and the ones that due to their technical complexity break down the most often. Subsequently these pieces of equipment have the greatest turnover rate and at the same time require the greatest planning effort. One of the goals of the cooperative is to improve its efficiency with respect to daily operation.

In this paper we deal with a real problem facing the above mentioned cooperative. This cooperative has a big number of partners, each of them having one or more smallholdings. The goal is to crop all the smallholdings by using harvesters and to transport the material from the smallholdings to the silos using trucks at a minimum operation cost. This objective must be addressed by designing coordinated efficient routing plans of harvesters and trucks the final goal being to reduce overall cost.

There are four main issues that distinguish this setup from a general routing problem. First of all, there are nested decisions because we have to coordinate the schedules of the harvesters and the trucks (Noam 2001). Second, there exist some kind of time windows associated with the starting time of each owner. These time windows are defined by the requests of the partners and a given tolerance level on the starting date. Third, this problem is related to the so called cluster traveling salesman problem (TSP) because all the smallholdings of the same owner have to be harvested before starting the next partner, but not necessarily by the same machine. This is because the silos in which grass or corn are garnered are opened when the harvest starts and they have to be closed in one or, at most, two days in order to guarantee maintenance conditions. Finally, each smallholding has a processing time which influences the times in the schedule. As far as we know there is no literature considering these elements simultaneously. In spite of that the cluster characteristic can be found in the seminal paper by Chisman (1975) and it also appears together with a simplified version of time windows in Laporte et al. (1996) and Marchal (2005).

The actual methodology followed by the cooperative starts when its technician rents a fixed number of trucks per season and subdivides the entire area into several working areas for the harvesters. Then, he or she assigns one harvester and a number of trucks to each area. The campaigns are organized sequentially in such a way that the harvest is also done sequentially in the different areas and trucks are assigned to pick the crop up in a smallholding whenever it is necessary and provided that there are available trucks. Obviously, every time that a harvester is waiting for a truck (because its hopper is full) the system suffers a delay, that in turns results in an increment of cost. Every week, before the beginning of the activity, there is a plan about the smallholdings to be harvested including the route to follow but nothing is planned for the trucks. This plan can be changed because of break-down of the machines, climatology, rescission of orders or new orders.

Alternatively, we propose a binary linear programming model that describes the problem. Nowadays routing problems appear in many logistics situations. These routing problems request a strong effort in their operational decisions. This is extremely true in the case of nested models where decisions depend on several levels of operation. Nevertheless, it is widely accepted that different tools of Mathematical Programming are suitable for obtaining solutions to this kind of problems.

In a previous paper (Carpente et al. 2009), the authors address a simpler problem that solves the operation planning of harvesters without interaction with the fleet of trucks. In that approach, they attained important savings both in cost and processing time of the machinery fleet but there was no solution model to cope with the coordination of the two vehicle fleets. Here, we extend that model to an integrated framework that incorporates all the ingredients of the actual situation described above. The goal is to minimize the total operation cost. In order to do that, one has to synchronize the schedules of harvesters and trucks so as to minimize the idle (waiting) and working time of the machinery using the given number of rented trucks which in turns is equivalent to minimize cost.

The reader may note that there are two types of cost in the above process. On the one hand, the cost induced by the SP harvesters is an internal cost of the cooperative and is invoiced directly to final users. On the other hand, the cost derived by renting trucks is external and is directly paid by the cooperative. For this reason, the cooperative does not want to lose the control on the number of trucks to be rented and it fixes it as a control parameter of the model. Clearly, there is no technical difficulty to modify the approach to leave this decision to the model introducing some estimated cost per used truck that will be added to objective function. The difference between these two approaches is similar to the one between standard p-median versus plant location problems. Whereas in the former the number of servers is fixed, in the latter this number is a decision variable. From a mathematical point of view, the first model can be transformed into the second moving the corresponding constraint to the objective function with an appropriate cost. The same analysis also applies to our model. In any case, since the number of available trucks to be rented is very limited one can do sensitivity analysis on this number solving different scenarios defined by the number of available trucks and then choosing the cheapest solution which would lead to similar solutions to the ones given by the variable number-of-trucks model.

A Department of Agroforestry Engineering of the University of Santiago de Compostela has designed a geographical information system (GIS) and other different hardware components for the harvesters: GPS, protection against rain and dust and an industrial computer among others. Moreover, they have incorporated software components in the harvesters and in a control center that include location, communication, and management modules. For a more detailed explanation of the data acquisition system of "Os Irmandiños" see Amiama et al. (2008). The GIS provides a digital cartographic base in which all the smallholdings and silos of the partners have a geographic reference. This will enable the cooperative to obtain the distances among them. This information together with the processing time of work in each smallholding (which is a constant that can be known from past sowing) are the main data which we need to solve the problem.

This problem is extremely hard which makes its exact resolution hopeless for medium to large size instances. For this reason, we design and apply a tabu search heuristic (Glover 1989) since they have shown to be effective tools for solving many NP-hard combinatorial optimization problems (Osman and Laporte 1996). Related literature in the context of vehicle routing problems is Laporte (1992), Noam (2001), Ghiani et al. (2003), Marchal (2005), Ando and Taniguchi (2006), and Berbeglia et al. (2007).

The paper is organized as follows. In Section 2 we present the mathematical formulation that models the actual problem. It is a binary linear program with a large number of variables and constraints. Section 3 describes our heuristic proposal to solve the problem: a tabu-like algorithm. There, we specify all the elements of the algorithm: initial solution, neighborhood structure, length of the tabu list and amended objective function. In Section 4, we report on our computational experiments both with respect to the exact and heuristic methods. Finally, we include some conclusions and comments on further research on this topic.

2 Formulation of the problem

In this section we present the mathematical formulation of the model that represents the joint problem of harvesters and trucks for agricultural cooperatives.

First of all, for the sake of simplicity we assume that each smallholding, when processed, exactly fills a harvester up. The reader may note that larger smallholdings can be adequately subdivided in smaller pieces to fulfill the above requirement. In addition, one harvester load is equal to one truck load as actually happens in the real world application.

2.1 Notation and decision variables

The main elements of our formulation are the following:

- Let $N = \{0, 1, \ldots, n, n+1\}$ represent *the set of* $n+2$ *smallholdings*, where 0 and $n+1$ are the fictitious origin and the fictitious final destination of the route, respectively.
- Let $M = \{1, \cdots, m\}$ be *the set of* m *trucks*.
- Let $L = \{1, \cdots, q\}$ be *the set of* q *owners*.

- $P = \{1, \cdots, T\}$ is *the set of all the possible invoiced periods.*
- $H = \{1, \ldots, h\}$ is *the set of h harvesters.*
- d_{ij} denotes the *traveling time for the harvesters between smallholdings i* and j, for all $i, j \in N$.
- a_{ij} denotes the *traveling time for the trucks between smallholdings i* and j (taking into account that the trucks have to visit the silo of smallholding i), for all $i, j \in N$.
- t_i is *the processing time of smallholding i*, for all $i \in N$.
- r_i is *the unloading time of smallholding i* (the time needed to unload the harvester in the truck), for all $i \in N$.
- $F(l)$ is *the set of smallholdings owned by holder l*, for all $l \in L$.
- $S(l)$ is *the period of time in which owner l requests the harvester to start processing his land*, for all $l \in L$.
- $R(l)$ is *the tolerance level* or maximum number of periods of time that owner ℓ can anticipate or delay his activity with respect to his request $S(l)$, for all $l \in L$.

We use two sets of decision variables in our formulation. The first set of variables, $x^c_{ii'k}$, are:

$$x^c_{ii'k} = \begin{cases} 1 & \text{if harvester } c \text{ starts processing smallholding } i \\ & \text{at period } k \text{ and then it goes to process smallholding } i', \\ 0 & \text{otherwise,} \end{cases}$$

for $i, i' \in N, k \in P, c \in H$.

The second set of variables, $y^j_{ii's}$, are similar to the one above but referred to trucks and they are defined as:

$$y^j_{ii's} = \begin{cases} 1 & \text{if truck } j \text{ starts to be loaded from the harvester in smallholding } i \\ & \text{at period } s \text{ and then it goes to smallholding } i' \\ 0 & \text{otherwise,} \end{cases}$$

for $i, i' \in N, s \in P, j \in M$.

In this way, the variables define a route that each harvester and truck has to follow and besides a time planning by specifying the time period in which a harvester starts to process each smallholding or a truck goes to unload the corresponding harvester.

2.2 Constraints

The relationships that describe the real-world model are translated in our formulation via mathematical constraints. These constraints are grouped in five sets:

A) Constraints describing routes and schedules of harvesters.
B) Constraints describing routes and schedules of trucks.
C) Constraints that link harvesters and trucks.

D) Constraints that assure that each owner's smallholdings are processed as a block (cluster constraints).
E) Constraints modeling requests of owners on starting processing times (time windows constraints).

A) Constraints describing routes and schedules of harvesters.

These constraints describe the routes and schedules of harvesters independently of the trucks. The formal description is as follows:

No harvester is allowed to go from a smallholding back to the same smallholding (except the fictitious final smallholding).

$$\sum_{c \in H} \sum_{i \in N \backslash \{n+1\}} \sum_{k} x_{iik}^{c} = 0. \tag{1}$$

Each smallholding is processed by exactly one harvester. This harvester goes exactly to one smallholding after processing the current one (except for the fictitious smallholdings).

$$\sum_{c \in H} \sum_{i' \in N} \sum_{k \in P} x_{ii'k}^{c} = 1 \quad \forall i \in N \backslash \{0, n+1\}. \tag{2}$$

There is exactly one harvester that processes each smallholding regardless of where it comes from (except for the fictitious smallholdings).

$$\sum_{c \in H} \sum_{i \in N} \sum_{k \in P} x_{ii'k}^{c} = 1 \quad \forall i' \in N \backslash \{0, n+1\}. \tag{3}$$

Each harvester reaches the final smallholding exactly once:

$$\sum_{i \in N \backslash \{n+1\}} \sum_{k \in P} x_{in+1k}^{c} = 1 \quad \forall c \in H. \tag{4}$$

Each harvester finishes operation when it reaches the final (fictitious) smallholding.

$$\sum_{k \in P} x_{n+1n+1k}^{c} = 1 \quad \forall c \in H. \tag{5}$$

$$\sum_{c \in H} \sum_{i' \in N \backslash \{n+1\}} \sum_{k \in P} x_{n+1i'k}^{c} = 0. \tag{6}$$

All harvesters start their operations at the initial fictitious smallholding and once the process is started a harvester never comes back to it.

$$\sum_{i' \in N} \sum_{k \in P} x_{0i'k}^{c} = 1 \quad \forall c \in H. \tag{7}$$

$$\sum_{c \in H} \sum_{i \in N} \sum_{k \in P} x_{i0k}^{c} = 0, \tag{8}$$

Each harvester is in continuous operation processing different smallholdings through the planning horizon until the final fictitious smallholding is reached.

$$\sum_{i\in N\setminus\{n+1\}} x^c_{ii'k} \leq \sum_{i''\in N} \sum_{\substack{k'\in P\\ k'>k}} x^c_{i'i''k'} \quad \forall i' \in N\setminus\{0\}\ \forall c \in H,\ \forall k \in P, \tag{9}$$

$$\sum_{i''\in N} x^c_{i'i''k'} \leq \sum_{i\in N\setminus\{n+1\}} \sum_{\substack{k\in P\\ k<k'}} x^c_{ii'k} \quad \forall i' \in N\setminus\{0\},\ \forall c \in H,\ \forall k' \in P. \tag{10}$$

Transitions of each harvester between consecutive smallholdings, in its schedule, fulfill processing, delay (if any), unloading and traveling times.

Note that the time when smallholding i starts to be processed by a harvester is given by:

$$\sum_{c\in H}\sum_{i'\in N}\sum_{k\in P} k\, x^c_{ii'k},$$

whereas the time when a truck starts to be loaded is:

$$\sum_{j\in M}\sum_{i'\in N}\sum_{s\in P} s\, y^j_{ii's}.$$

Then, the delay due to smallholding i, i.e., the time that the harvester is waiting for a truck, once it is full, can be represented as:

$$\begin{aligned}\varepsilon_i &= \sum_{j\in M}\sum_{i'\in N}\sum_{s\in P} s\, y^j_{ii's}\\ &\quad - \sum_{c\in H}\sum_{i'\in N}\sum_{k\in P} (k+t_i)\, x^c_{ii'k} \forall i \in N\setminus\{0, n+1\},\ \varepsilon_0 = \varepsilon_{n+1} = 0.\end{aligned}$$

This delay is always greater than or equal to 0 due to constraint (22).

Therefore, the transition of each harvester between consecutive smallholdings must fulfill:

$$\sum_{k\in P}(k + t_i + \varepsilon_i + r_i + d_{ii'})\, x^c_{ii'k} \leq \sum_{i''\in N}\sum_{k'\in P} k'\, x^c_{i'i''k'},\ \forall i \in N,\ \forall i' \in N,\ \forall c \in H.$$

Note that these inequalities involve quadratic terms in $\varepsilon_i \times x^c_{ii'k}$. In order to transform these quadratic inequalities into linear ones, we introduce a big enough constant K and we rewrite the inequalities in the following form.

$$\begin{aligned}&\sum_{k\in P}(k + K + t_i + r_i + d_{ii'})\, x^c_{ii'k} + \varepsilon_i\\ &\quad \leq \sum_{i''\in N}\sum_{k'\in P}(k' + K)\, x^c_{i'i''k'} + K\,(1 - \sum_{k\in P} x^c_{ii'k}), \forall i \in N,\ \forall i' \in N,\ \forall c \in H.\end{aligned} \tag{11}$$

B) Constraints describing routes and schedules of trucks.

The constraints in this group are similar to the corresponding ones in the previous subsection (A) but referred to trucks. For the sake of completeness, we describe them in full detail in the following:

No truck is allowed to go from a smallholding back to the same smallholding.

$$\sum_{j\in M}\sum_{i\in N}\sum_{s\in P} y^j_{iis} = 0 \tag{12}$$

Each smallholding is processed by exactly one truck. This truck goes exactly to one smallholding after carrying and unloading the crop of the current one (except for the fictitious smallholdings).

$$\sum_{j\in M}\sum_{i'\in N}\sum_{s\in P} y^j_{ii's} = 1 \quad \forall\, i \in N\backslash\{0, n+1\} \tag{13}$$

There is exactly one truck that is assigned to each smallholding regardless of where it comes from (except for the fictitious smallholdings).

$$\sum_{j\in M}\sum_{i\in N}\sum_{s\in P} y^j_{ii's} = 1 \quad \forall\, i' \in N\backslash\{0, n+1\} \tag{14}$$

Each truck reaches the final smallholding exactly once:

$$\sum_{i\in N}\sum_{s\in P} y^j_{in+1s} = 1 \quad \forall\, j \in M \tag{15}$$

Each truck finishes operation when it reaches the final (fictitious) smallholding because once there it does not move anymore.

$$\sum_{j\in M}\sum_{i'\in N}\sum_{s\in P} y^j_{n+1i's} = 0 \tag{16}$$

All trucks start their operations at the initial fictitious smallholding and once the process is started a truck never comes back to it.

$$\sum_{i'\in N}\sum_{s\in P} y^j_{0i's} = 1 \quad \forall\, j \in M \tag{17}$$

$$\sum_{j\in M}\sum_{i\in N}\sum_{s\in P} y^j_{i0s} = 0 \tag{18}$$

Each truck is in continuous operation loading and unloading different smallholdings through the planning horizon until the final fictitious smallholding is reached.

$$\sum_{i\in N} y^j_{ii's} \le \sum_{i''\in N}\sum_{\substack{s'\in P\\ s'>s}} y^j_{i'i''s'} \quad \forall\, i' \in N\backslash\{0, n+1\},\ \forall\, j \in M,\ \forall\, s \in P \tag{19}$$

$$\sum_{i''\in N} y^j_{i'i''s'} \le \sum_{i\in N}\sum_{\substack{s\in P\\ s<s'}} y^j_{ii's} \quad \forall\, i' \in N\backslash\{0,n+1\},\ \forall\, j\in M,\ \forall\, s'\in P \tag{20}$$

Transitions of each truck between consecutive smallholdings, in its schedule, fulfill unloading and travelling times.

$$\sum_{s\in P}(s+r_i+a_{ii'})\, y^j_{ii's} \le \sum_{i''\in N}\sum_{s'\in P} s'\, y^j_{i'i''s'} \quad \forall\, i\in N,\ \forall\, i'\in N\backslash\{n+1\},\ \forall\, j\in M \tag{21}$$

C) Constraints that links harvesters and trucks.

Operations and schedules of harvesters and trucks must be synchronized with respect to the corresponding processing times.

$$\sum_{j\in M}\sum_{i'\in N}\sum_{\substack{s\in P\\ s<k+t_i}} y^j_{ii's} \le 1-\sum_{c\in H}\sum_{i'\in N} x^c_{ii'k} \quad \forall\, i\in N\backslash\{0,n+1\},\ \forall\, k\in P \tag{22}$$

These constraints ensure that if a harvester starts to process a smallholding i at period k, i.e. $\sum_{c\in H}\sum_{i'\in N} x^c_{ii'k}=1$, then no truck can start to be loaded in this smallholding i at any period s prior to the time k plus the processing time t_i of smallholding i.

D) Constraints that assure that each owner's smallholdings are processed as a block (cluster constraint).

$$\sum_{c\in H}\sum_{i\in F(l)}\sum_{i'\in F(l)\backslash\{i\}}\sum_{k\in P} x^c_{ii'k} = |F(l)|-1 \quad \forall\, l\in L \tag{23}$$

Recall that $F(l)$ is the set of smallholdings that belong to owner l. The above constraints state that the number of consecutive movements among smallholdings of the same owner l must be equal to the number of smallholding of this owner minus one, i.e. $|F(l)|-1$. This ensures that all the smallholdings of owner l are processed consecutively.

E) Constraints modeling requests of owners on starting processing times (time windows constraints).

For each owner, there is an admissible time window given by his request to start the activity in his land and a common period of tolerance. None of the smallholdings of this owner is processed before the lower limit of his admissible time window

$$\sum_{c\in H}\sum_{i\in F(l)}\sum_{i'\in N}\sum_{\substack{k\in P\\ k<\max\{1,S(l)-R(l)\}}} x^c_{ii'k} = 0 \quad \forall\, l\in L \tag{24}$$

Constraints (24) together with the following one ensure that the activity of each owner must start within his admissible interval:

$$\sum_{c\in H}\sum_{i\in F(l)}\sum_{i'\in N}\sum_{k=\max\{1,S(l)-R(l)\}}^{\min\{S(l)+R(l),T\}} x^c_{ii'k} \geq 1 \quad \forall l \in L \tag{25}$$

The last constraints impose the binary character on the decision variables.

$$x^c_{ii'k}, y^j_{ii's} \in \{0, 1\} \quad \forall i, i' \in N, k, s \in P, c \in H, j \in M \tag{26}$$

2.3 Objective function

The objective function is to minimize the total time of harvesters' activity plus the delays induced by harvesters waiting for trucks (to unload). For each harvester, this is obtained as the difference between the starting time (period of time when the harvester departs from the fictitious initial smallholding) and the final time (period of time when the harvester reaches the fictitious final smallholding).

$$\min \sum_{c\in H}\left(\sum_{k'\in P} k' x^c_{n+1n+1k'} - \sum_{i\in N}\sum_{k\in P} k x^c_{0ik}\right)$$

The above formulation is designed to solve to optimality the original problem. Nevertheless, this formulation gives rise to an extremely hard integer program. To illustrate its complexity, the reader may note that the overall number of binary variables and constraints are $(n+2)^2\, T\,(h+m)$ and $6+4n+3h+3m+3l+2(n+1)hT+2nmT+(n+1)(n+2)m+(n+1)2h+nT$, respectively. For instance, for an actual problem with 400 smallholdings, 5 harvesters, 5 trucks, 30 owners and 500 time periods, the number of variables and constraints are 808, 020, 000 and 5, 011, 731, respectively. (Note that the cooperative plans its operations on a weekly basis. Thus, the standard number of periods of these problems are around 500. They correspond to one week of work since time is discretized to 5 minutes periods for invoicing reasons. Also it is hard to exceed 400 smallholdings per week.)

As it is shown in Section 4, solving the above formulation to optimality is very cumbersome. This makes it almost impossible to solve the problem for medium to large size instances. For this reason, in the next section, we develop an alternative heuristic approach that provides good quality solutions of the problem.

3 Heuristic algorithm

Local search techniques based on tabu search algorithms have been shown to be effective tools for solving many NP-hard combinatorial optimization problems. They require an initial solution and a neighborhood structure and proceed by transitioning from one solution to another using moves across a number of iterations. Tabu search algorithms take into account a list of forbidden moves, the *tabu list*, trying to skip local optima solutions. A general description of this kind of algorithms can be found in Glover (1989).

In this section we develop a heuristic approach to solve our problem. Our first observation is that to simplify its implementation for escaping of local optima, we considered an enlarged feasible space that during the execution may generate solutions that are not feasible in the original model. In this context, given a set of smallholdings to be processed by different harvesters, a *solution* s is a path without cycles connecting all the smallholdings for each harvester and a plan for a given fleet of trucks that have to carry the crop from the smallholdings to the silos. This planning specifies the time period in which a specific truck serves a certain harvester in a given smallholding. See Fig. 2 for an example. In this figure we depict the optimal sequence of events in a specific plan over a 12-h period. Each event is represented by a bold cross that indicates the time period in which a truck serves a given harvester that processes a smallholding of a given owner. The reader may note that we have omitted the blocks of constraints (24) and (25) defining a feasible solution.

3.1 Elements

Our algorithm is specified by four elements: an initial solution, an amended objective function, a neighborhood structure, and a tabu list. In the following we describe all these elements.

3.1.1 Initial solution

Taking into account that all the owners have a request for the starting period of activity in their lands, the initial solution, s_0, is generated as follows:

– For each harvester assign a geographical area associated to this machine. Sort the owners belonging to this geographical area according to their requests. Ties are randomly broken.
– Sort the smallholdings of each owner according to the order in which they appear in the matrix of traveling times among smallholdings.

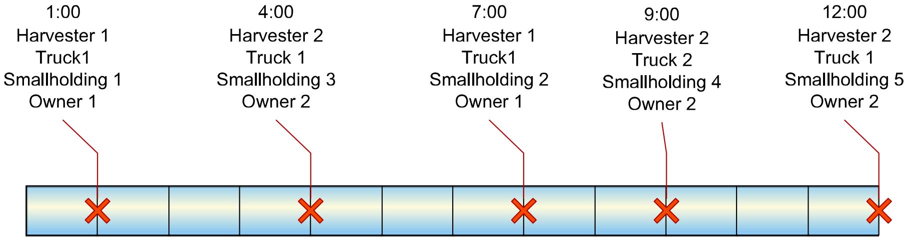


Fig. 2 Scheme of a solution for a problem with two harvesters, two trucks and two owners over a 12-h period

- The above scheme generates routes for the different harvesters. We associate to each of these routes a time schedule for each harvester taking into account the processing time of each smallholding and the traveling times among them. All the harvesters start working at period 0. Sort all the events (time periods in which harvesters require a truck service) in a unique time-line according to the above planning.
- Assign a *free* truck[1] to the next event on the time line, in non-decreasing order of the following list of preferences.
  - Preferences for the assignment of trucks to harvesters:
    1. Find a free truck previously associated to this harvester.
    2. Find a free truck previously associated to a different harvester.
    3. Find a truck in the given fleet that had never been assigned to a harvester.
  - If it is not possible to find a truck satisfying one of the preceding items, then choose among the trucks previously assigned the one that minimizes the waiting time of the harvester (delay). Ties are broken following the above preferences in non-decreasing sequence. Rearrange the events that require a truck service on the time line, taking into account the new delay and repeat this procedure.

3.1.2 Amended objective function

The initial goal is the minimization of the overall *activity time*, i.e. the one used by the harvester for processing smallholdings, plus the delays incurred while waiting for trucks to be unloaded. To this end we evaluate each solution by using an amended objective function that takes into account the time of activity of all the machines plus a weighted sum of the delays (trying to incorporate the importance of the delays) plus a measure of infeasibility (a weighted sum of violated constraints). Given an incumbent solution, s_i, corresponding to an iteration i of the algorithm the amended objective function is:

$$f_{s_i} = Ta_{s_i} + w_1 D_{s_i} + \sum_l w_{2l}\, T_{s_i}(l)$$

where Ta_{s_i} stands for the total activity time spent by the harvesters in that solution s_i (in the planning induced by s_i), D_{s_i} is the accumulated delay of all the harvesters under this solution s_i weighted by the scaling factor $w_1 > 0$ which reflects the importance given to delays, and for each owner l, $T_{s_i}(l)$ is equal to 1 if the constraint given by the request of this owner and its tolerance level is not satisfied and $T_{s_i}(l)$ is equal to 0 otherwise. Moreover, for each l,

[1] We say that a truck is *free* if it has carried the crop to the silo assigned to a given smallholding and it still can meet the new harvester's request after covering the distance between the silo and the new smallholding.

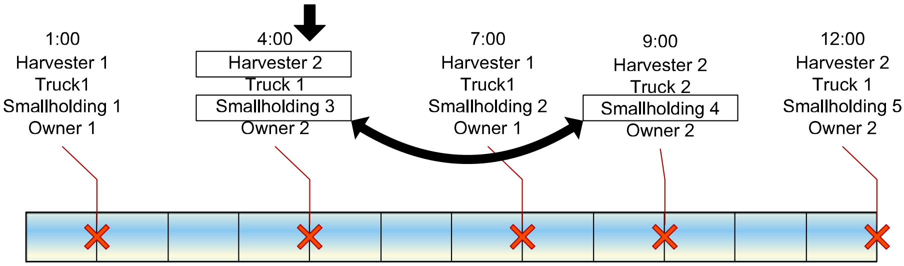


Fig. 3 Example of movement 1

$w_{2l} > 0$ is an arbitrary scaling factor weighting the fulfillment of the requests of the l-th owner.

3.1.3 Neighborhood structure

The structure of our neighborhood is based on three types of random movements:

- **Movement 1**. We select a random harvester and choose two random smallholdings of its route:
  - If they belong to the same owner we exchange their positions in the processing sequence (see Fig. 3).
  - If they do not belong to the same owner we exchange the corresponding owners without modifying the processing sequence of smallholdings within each owner.

  Note that after doing this movement we have to rearrange the requests for truck services and the assignment of the fleet of trucks.
- **Movement 2**. We select two random requests for truck service (independently of the harvester) and if they are served by different trucks, then we interchange these assignments (see Fig. 4). After doing this movement the

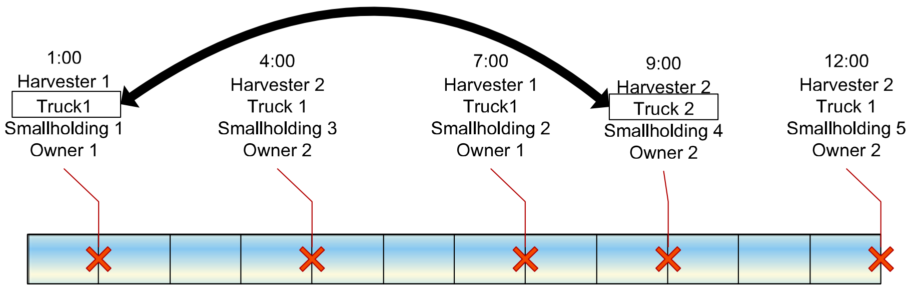


Fig. 4 Example of movement 2

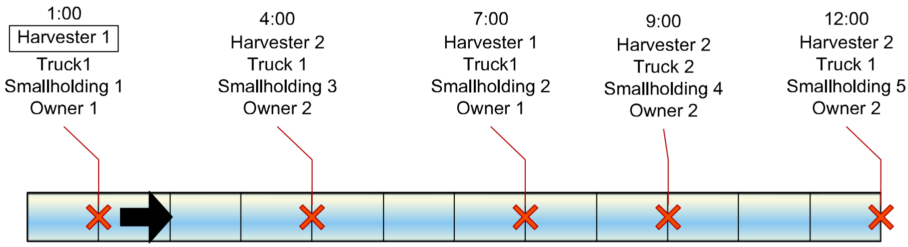


Fig. 5 Example of movement 3

routes of the harvesters are the same as before but we have to redefine the planning.

– **Movement 3**. We select a random harvester and delay the beginning of its activity by a random number of periods between 1 and b, where b is the tolerance level of the first owner that appears in the route of this harvester (see Fig. 5). If $b = 0$ we do not perform any translation.

The size of our neighborhood is limited by a scalar μ that sets the number of neighbors (movements) generated in each iteration. Moreover, we can select the number of movements of each type (1, 2 or 3) to be performed by setting the corresponding percentages.

3.1.4 Tabu list

We maintain a historic list of solutions to avoid cycling. This list follows an FIFO (first in first out) discipline. The length of this list is a parameter, z, fixed by the user that establishes the capacity of the memory (short, medium, long-term memory).

3.2 The algorithm

This algorithm starts with the initial solution s_0 already described. This solution enters the tabu list. In any subsequent iteration, i, the algorithm considers $N'(s_i)$, the neighborhood of s_i obtained by μ (a parameter) movements of the current planning s_i. The type of these μ movements is specified by giving the percentage of movements 1, 2 and 3, respectively. Then it selects a neighbor, $\bar{s}_i \in N'(s_i)$, that provides the minimum amended objective function, f_i. If the corresponding planning is not in the tabu list it becomes the incumbent solution and it enters the list. Notice that when the list is full, using the FIFO strategy, the first element in the list is removed.

If the best neighbor $\bar{s}_i$ of $N'(s_i)$ is in the tabu list, the algorithm will take into account this solution as the current solution only when this planning provides an amended objective function less than or equal to the aspiration function, i.e the objective value of the current solution minus some correction parameter q. Otherwise, the algorithm selects the second best neighbor and proceeds in a similar way. As for the stopping criterion, this algorithm uses a fixed number of iterations. After finishing this number of iterations we select the best planning (solution) found so far.

The implementation of the algorithm has been done according to the pseudo-code in Algorithm 1.

Algorithm 1: Tabu search algorithm.

```
input  : {S(l) : l ∈ L}, {T(l) : l ∈ L}, {t_i : i ∈ N}, {d_ij : i, j ∈ N}, {a_ij : i, j ∈ N},
         {F(l) : l ∈ L}, m, T, I (number of iterations), μ (number of movements),
         percentage of movements 1, 2, and 3, w_1, {w_2l : l ∈ L}, q, length z of tabu list.
output: A local best planning s*;
         period in which each harvester requires a truck service;
         assignments of trucks to harvesters;
         local best total activity time.
begin
    Tabu list is empty;
    s* = s_0 (the initial solution);
    s* is current solution and s* enters tabu list;
    repeat
        let s_i be current solution;
        select best solution s̄_i ∈ N'(s_i);
        if s̄_i is not in tabu list or s̄_i satisfies aspiration function then
            if tabu list is full then
                remove the first element in tabu list;
                s̄_i is current solution and s̄_i enters tabu list s_{i+1} ⟵ s̄_i
            end
        else
            s_{i+1} ⟵ s_i
        end
    until number of iterations ;
    select the best solution s* of the set of all current solutions;
    return s*.
end
```

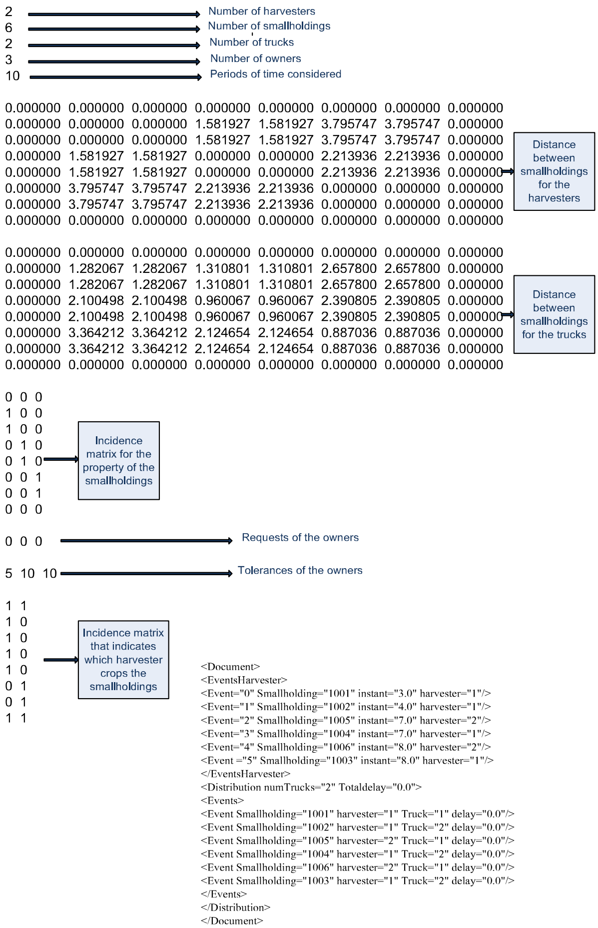
4 Numerical results

Our computational experiments are organized in two blocks. In the first one we compare the quality of our heuristic algorithm with the optimal solution of the IP model. The IP model has been solved with CPLEX 8.1 using a computer with an Intel(R) Xenon(R) processor with CPU E5320 1.86 GHz and 8 GB of RAM memory. The heuristic algorithm has been implemented in JAVA language and solved on a notebook Intel (R) Core (TM) 2 Duo with CPU T7300 2.00 GHz and 2 GB of RAM memory. This comparison is only possible for small instances because due to the high complexity of the exact model CPLEX was unable to solve medium to large instances of this problem. The second block analyzes the quality of the heuristic algorithm by using a battery of trials we have created for this purpose. We have generated 85 file instances that we have encoded with the names O(number of owners)_S(number of smallholdings)_H(number of harvesters)_T(size of the fleet of trucks). For instance, the instance O3_S6_H2_T2 has the data for a problem with three owners, six smallholdings, two harvesters, and two trucks.

Figure 6 shows an example considering these parameters. This figure is organized in two parts. The first one, that covers from the beginning until "< Document >", is devoted to the input data and the second one shows the output of the algorithm. The input data needed by the algorithm are:

– The number of harvesters we want to consider.
– The number of smallholdings that need to be processed.
– The number of trucks that are available to provide service.
– The number of owners that have asked for service.
– Time horizon to be planned. (Period of time we have to plan.) Note that this number is only required for the exact model. This time horizon is usually measured in units of five minutes.

Fig. 6 Example and a solution provided by the algorithm in a trial instance



- A matrix of distances (times) between smallholdings from the point of view of the harvesters. This matrix includes the fictitious initial and final smallholdings. Distances for the harvesters are measured in periods of time, taking into account that harvesters move at a constant velocity of 20 km/h.
- A matrix of distances (times) between smallholdings from the point of view of the trucks taking into account that a truck must visit the silo of the initial smallholding. This matrix includes the fictitious initial and final smallholdings. Distances for trucks are transformed to time assuming that they move at an average velocity of 40 Km/h.
- An incidence matrix that assigns the property of each smallholding to his/her owner.
- Periods of time when owners request for their services.
- Tolerance, i.e. the number of periods of time that we can shift the request of each owner.
- A matrix that assigns smallholdings to harvesters, taking into account that harvesters in the real world operate in separated areas.

For the sake of simplicity, we consider that the processing times for all smallholdings are 1 and that loading times of trucks with crop from harvesters are 0. This assumption is realistic because the sum of both tasks is always close to five minutes which is our discrete time basis.

The output in the second part of Fig. 6 shows a solution that consists of two parts. The first one lists the events, i.e. the requests for service of the smallholdings in a certain period of time. The second one shows the truck that is assigned to cover the corresponding event and the time that the selected harvester has to wait for the truck in that solution (delay). For instance, from the results in Fig. 6, we can see that harvester #1 in smallholding 1001 requests a truck at period 3. This service is provided by truck #1 without delay.

4.1 Exact versus heuristic model

In order to evaluate the quality of the solutions obtained by the heuristic algorithm we solved exactly the IP model for ten small instances. These instances consider 3 owners, 6 or 9 smallholdings, 2 harvesters, and 2 trucks. The parameters for the heuristic are set to 100 iterations, the number of movements per iteration is 20, the parameter q of the aspiration function is 2, and the length of the tabu list is set to 10. The weights w_1 and w_2 for the amended objective function are set to 100 and 1000, respectively. These quantities reflect that we give more importance to satisfy the requests of the owners at the price of possibly generating some delay. We ran the algorithm using six different combinations of moves to generate the neighborhood in each iteration:

- M1 generates 100% of movements type 1.
- M2 generates 100% of movements type 2.

- M1M2 generates 50% of movements type 1 and 50% of movements type 2.
- M1M3 generates 50% of movements type 1 and 50% of movements type 3.
- M2M3 generates 50% of movements type 2 and 50% of movements type 3.
- M1M2M3 generates 35% of movements type 1, 35% of movements type 2, and 30% of movements type 3.

Table 1 shows the average objective function obtained after ten runs of each of the instances that appear in its second column. The first column stands for the instance number to be used in Fig. 7. The third column shows the results of the exact model solved with CPLEX and the following ones show the result of the heuristic algorithm taking into account the six different combinations of movements described above. For these 10 instances only the versions of the algorithm that include movements of type 3 attain always the exact optimal values. Table 2 shows the average running times in seconds corresponding to the results in Table 1 (to solve the IP problem using CPLEX we have considered an upper limit of 20 h of cpu time). We can observe that running times of the heuristic are significantly smaller (several orders of magnitude) than those for the exact method (see also Fig. 7). Taking into account both tables, we can conclude that the combination of the three movements in the tabu search is a promising technique to solve instances with a reasonable computational effort.

4.2 The heuristic algorithm on bigger instances

It is clear from Table 2 that we can not expect to solve real size instances using the exact method. In this subsection we analyze the heuristic version of the algorithm that uses the three types of movements over instances ranging from 5 to 30 owners and 10 to 400 smallholdings. Note that the cooperative makes weekly planning and it is hard to exceed 400 smallholdings per week. We

Table 1 Results provided by the exact model versus the results obtained with the heuristic algorithm taking into account different combination of moves

Objective value (total time of activity of the harvesters + delay)								
Num.	Instances	CPLEX	M1	M2	M1M2	M1M3	M2M3	M1M2M3
1	O3_S6_H2_T2(1)	8	8	8 + 1	8	8	8	8
2	O3_S6_H2_T2(2)	8	8	8 + 1	8	8	8	8
3	O3_S6_H2_T2(3)	9	9	9	9	9	9	9
4	O3_S6_H2_T2(4)	7	7 + 1	7 + 1	7 + 1	7	7	7
5	O3_S6_H2_T2(5)	8	8 + 2	8 + 2	8 + 2	8	8	8
6	O3_S9_H2_T2(1)	11	11 + 4	11 + 2	11 + 2	11	11	11
7	O3_S9_H2_T2(2)	10	10 + 1	10 + 1	10 + 1	10	10	10
8	O3_S9_H2_T2(3)	11	11	11 + 1	11	11	11	11
9	O3_S9_H2_T2(4)	11	11 + 1	11 + 1	11 + 1	11	11	11
10	O3_S9_H2_T2(5)	10	10 + 1	10 + 3	10 + 1	10	10	10

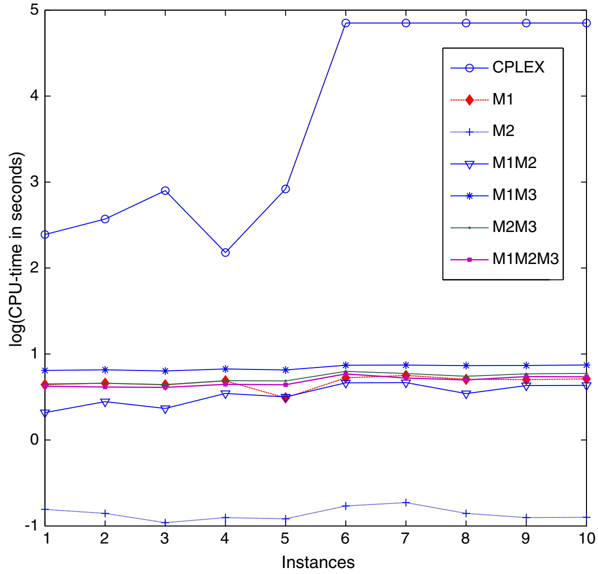


Fig. 7 This figure shows the logarithm of the running times of the different combination of movements with the data of Table 2. Instance number refers to the *Num.* shown in the first column of Table 2

consider five different files for each class of instances and the results in Table 3 report averages and standard deviations of 10 runs of each file, respectively. The first column represents the instances, the second column shows the total time of activity of the harvesters, the third column shows the total delay, the fourth column shows the number of infeasibilities with respect to the tolerance level of the owners, the fifth column (U) shows for each instance the number of files with infeasibilities, and the last column shows the running time of the algorithm in seconds.

The parameters for the heuristic are set to 50 iterations, the number of movements per iteration is 20, the parameter q of the aspiration function is

Table 2 Average running times provided by the exact model and the heuristic algorithm taking into account different combination of moves

Runtime(seconds)								
Num.	Instances	CPLEX	M1	M2	M1M2	M1M3	M2M3	M1M2M3
1	O3_S6_H2_T2(1)	245	4.399	0.156	2.080	6.443	4.478	4.214
2	O3_S6_H2_T2(2)	378	4.571	0.140	2.792	6.551	4.570	4.125
3	O3_S6_H2_T2(3)	810	4.368	0.109	2.325	6.345	4.405	4.089
4	O3_S6_H2_T2(4)	154	4.867	0.125	3.480	6.701	4.901	4.433
5	O3_S6_H2_T2(5)	841	3.089	0.121	3.167	6.521	4.860	4.392
6	O3_S9_H2_T2(1)	72001	5.289	0.171	4.618	7.395	6.287	5.868
7	O3_S9_H2_T2(2)	72000	5.632	0.187	4.651	7.426	5.925	5.256
8	O3_S9_H2_T2(3)	72000	5.070	0.140	3.463	7.332	5.501	4.988
9	O3_S9_H2_T2(4)	72000	5.038	0.125	4.290	7.363	5.872	5.457
10	O3_S9_H2_T2(5)	72001	5.132	0.126	4.321	7.441	5.948	5.427

Table 3 Data obtained by running the heuristic algorithm over the trial instances

Instances	avg/std (activity)	avg/std (delay)	avg/std ($\sum T(l) > 0$)	U	avg/std (runtime(sec.))
O5_S10_H2_T2	15.8/0	0.4/0	0/0	0	4.777/0.210
O5_S15_H3_T2	17.8/0	0.1/0	0/0	0	6.327/0.344
O5_S20_H3_T2	25/0	0.6/0	0/0	0	9.61/0.432
O8_S20_H2_T2	31.6/0	1.7/0.3	0/0	0	10.305/0.470
O8_S30_H3_T2	40.6/0	7.4/0.5	0/0	0	23.874/0.876
O8_S40_H3_T2	54.4/0	20.6/0.4	0.4/0	2	48.276/1.021
O10_S20_H2_T2	36.8/0	2.4/0.1	0/0	0	11.106/0.502
O10_S40_H3_T2	55.4/0	33.2/0.3	0.2/0	1	49.917/1.215
O10_S100_H4_T3	138.2/0	23.6/1.2	0/0	0	592.938/20.125
O20_S40_H2_T2	73.4/0	7.8/0.7	0/0	0	43.412/1.344
O20_S100_H4_T3	141.2/0	54/1.3	0.6/0	1	689.298/24.126
O20_S200_H5_T4	276.8/0	107.2/4.3	0/0	0	3425.125/324.121
O30_S60_H3_T3	111.4/0	15,8/0.2	0/0	0	188.342/6.121
O30_S100_H4_T4	148.8/0	26.2/1.3	0/0	0	691.901/24.452
O30_S200_H5_T5	281.4/0	61.2/2.5	0/0	0	3942.035/500.047
O30_S300_H5_T5	538.8/0	90.8/11.3	1/0	2	5648.08/432.08
O30_S400_H5_T5	580/0	115.7/13.6	3.4/0	4	6230.5/534.41

2, and the length of the tabu list is set to 10. The weights w_1 and w_2 for the amended objective function are set to 100 and 10000,[2] respectively. We ran the algorithm using 35% of movements type 1, 35% of movements type 2, and 30% of movements type 3.

As an evidence of the complexity of the exact model we report on the behavior of a file of the instance O5_S10_H2_T2. In this instance, after 72 h of CPU time, CPLEX obtained a best objective value of 23 whereas the heuristic algorithm obtained a value of 15 in 4 s. (Note that after 72 h of CPU time CPLEX was only able to find some feasible solutions but not the optimal one!.) All files in Table 3 reach a good solution in reasonable time. Instances up to 100 smallholdings are solved in a few seconds. In the remaining cases we could have obtained faster results using a different combination of movements in the algorithm: the more type 2 movements the faster the results obtained. On the contrary, movements of type 1 and 3 increase running times of the algorithm but then it obtains lower delays.

The algorithm does not fulfill the tolerance level constraints (24) and (25) only in 10 files out of the 85 trials. For these cases it would be possible to obtain feasible solutions if we had added new trucks to the problem. For instance, in

[2]We increase the weighting factors w_{2l} of the tolerance level infeasibility according to the size of these bigger instances.

Fig. 8 Evolution of delays and feasibility of file O8_S40_H3_T2(1) with respect to the number of trucks in the problem

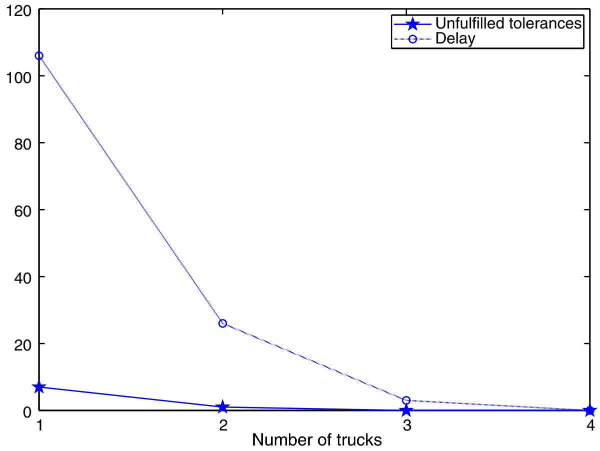


Fig. 8 we show the evolution of the delay and the feasibility in a file of the instance O8_S40_H3_T2. We observe that the solutions improve by adding new trucks.

5 Conclusions

This paper analyzes and solves a real logistic problem that appears in the agricultural cooperative of "Os Irmandiños." This problem consists of coordinating a fleet of self-propelled harvesters with a fleet of trucks used to carry the crop from the smallholdings to the silos. The model we have developed take into account four characteristics that distinguish our approach from a general routing problem: time windows, nested decisions, processing times required to service each facility and the fact that the facilities must be visited in clusters. As far as we know there is no literature considering these elements simultaneously. In this paper we introduce a binary linear programming model in order to minimize the working time of the harvesters in the fleet. Exactly solving the problem for large instances is computationally cumbersome and for this reason we introduce and implement a heuristic algorithm to reduce the computational time. The heuristic has been applied to data estimated using information of "Os Irmandiños." Our experimental results show that the heuristic can be effectively applied with reasonable computational effort.

The real-world implementation of the model is still in progress since the cooperative is currently recording the required data about distances between smallholdings and silos. Needless to say, the application of this methodology in the cooperative will improve the efficiency of the entire process since until now the decision-making process has been done by estimations made according to rough practical rules, from previous experiences. On top of that, our algorithm will also considerably save time to the decision-maker since our method gives automatic plans avoiding the expert technician to be working on complex schedules.

Our future work includes to actually implement the methodology in the cooperative, as soon as they have the available data, to finish a parallel version of the heuristic that will reduce the running times and to compare our tabu heuristic with other heuristic algorithms.

Acknowledgement The authors want to thank Mr. Juan Jesús Sanz Sixto for the JAVA implementation of the heuristic algorithm developed in this paper.

References

Aarts E, Lenstra J (eds) (2003) Local search in combinatorial optimization. Princeton University Press, Princeton

Amiama C, Bueno J, Álvarez C, Pereira J (2008) Design and field test of an automatic data acquisition system in a self-propelled forage harvester. Comput Electron Agric 61(2):192–200

Ando N, Taniguchi E (2006) Travel time reliability in vehicle routing and scheduling with time windows. Networks and Spatial Economics 6(3):293–311

Berbeglia G, Cordeau J, Gribkovskaia I, Laporte G (2007) Static pickup and delivery problems: a classification scheme and survey. TOP 15(1):45–47

Carpente L, Casas-Méndez B, Jácome C, Puerto J (2009) A model and two heuristic approaches for a forage harvester planning problem: a case study. TOP. doi:10.1007/s11750-009-0104-5

Chisman J (1975) The clustered traveling salesman problem. Computers & OR 2(2):115–119

Ghiani G, Guerriero F, Laporte G, Musmanno R (2003) Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies. Eur J Oper Res 151(1):1–11

Glover F (1989) Tabu search, part I. ORSA J Comput 1(3):190–206

Laporte G (1992) The vehicle routing problem: an overview of exact and approximate algorithms. Eur J Oper Res 59(3):345–358

Laporte G, Potvin J, Quilleret F (1996) A tabu search heuristic using genetic diversification for the clustered traveling salesman problem. Journal of Heuristics 2:187–200

Marchal F (2005) A trip generation method for time-dependent large-scale simulations of transport and land-use. Networks and Spatial Economics 5(2):179–192

Noam E (2001) Interconnecting the network of networks. MIT, Cambridge

Osman I, Laporte G (1996) Metaheuristics: a bibliography. Annals of Operations Research 63:511–623