



Exact and heuristic methods to solve a bi-objective problem of sustainable cultivation

Angelo Aliano Filho¹ · Helenice de Oliveira Florentino² · Margarida Vaz Pato³ · Sônia Cristina Poltroniere⁴ · João Fernando da Silva Costa⁵

Published online: 14 November 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

This work proposes a binary nonlinear bi-objective optimization model for the problem of planning the sustainable cultivation of crops. The solution to the problem is a planting schedule for crops to be cultivated in predefined plots, in order to minimize the possibility of pest proliferation and maximize the profit of this process. Biological constraints were also considered. Exact methods, based on the nonlinear model and on a linearization of that model were proposed to generate Pareto optimal solutions for the problem of sustainable cultivation, along with a metaheuristic approach for the problem based on a genetic algorithm and on constructive heuristics. The methods were tested using semi-randomly generated instances to simulate real situations. According to the experimental results, the exact methodologies performed favorably for small and medium size instances. The heuristic method was able to potentially determine Pareto optimal solutions of good quality, in a reduced computational time, even for high dimension instances. Therefore, the mathematical models and the methods proposed may support a powerful methodology for this complex decision-making problem.

Keywords Multi-objective optimization · Genetic algorithm · Constructive heuristics and sustainability

1 Introduction

In the current context of sustainable development, the cultivation practices favoring minimization of environmental degradation through alternative ways that avoid the intensive use

✉ Angelo Aliano Filho
angeloaliano@utfpr.edu.br

¹ Departamento Acadêmico de Matemática, Universidade Tecnológica Federal do Paraná, Apucarana, Brazil

² Instituto de Biociências de Botucatu, Universidade Estadual Paulista, Botucatu, Brazil

³ ISEG and CMAFcIO, Universidade de Lisboa, Lisbon, Portugal

⁴ Departamento de Matemática, Universidade Estadual Paulista, Bauru, Brazil

⁵ Universidade Tecnológica Federal of Paraná, Apucarana, Brazil

of chemical products to combat pests besides increasing soil usage are being intensively studied. These practices, if adopted, can prevent an environmental crisis on the planet and make agricultural products healthier.

In this sense, one of the central focuses in crop production discussed of late is the use of measures designed to target sustainable and ecological planning, in the light of environmental degradation that has occurred in recent years. For this reason, the planning of agriculture activities, such as crop rotation, has been gaining prominence in the studies aimed at sustainable cultivation, since it is one of the means of cultivation whose practical principles facilitate ecological and productive agriculture. This practice, once conducted by the rural farmers, brings numerous benefits, since the control of pests, pathogens and weeds are performed biologically, thus decreasing the harmful effects of pesticides to the environment and leading to actions promoting soil recovery, thus making it forever fertile, as stated by Snapp et al. (2005).

In this paper, we present a nonlinear, more specifically, a bi-objective binary quadratic optimization model inspired by Santos et al. (2010), Santos et al. (2011), Santos et al. (2015a) and Aliano Filho et al. (2014). The first three papers referred to presented versions of an integer linear programming model, considering binary variables, constraints of demand, adjacent plots, green manure and fallow, whose aim was to maximize a single objective function (representing the total time of occupation for the planting area or the total profit of the plantation). These papers were focussed on exact methodologies based on branch-and-bound and column generation. In the fourth paper, a different approach was employed. The authors proposed several metaheuristics (evolutionary algorithms, simulated annealing and hybrids) to solve the crop rotation problems presented in Santos et al. (2010), Santos et al. (2011). The objective function, to maximize, represents the total profit in the planning horizon. One of the main features of this paper is to consider a formulation inspired on those works, but here we consider two conflicting objective functions.

The mathematical model considered in this paper deals with two conflicting objectives. The first one, of environmental nature, is defined to minimize the possibility of pest dissemination among the crops in the planting area, along the planning horizon. A quadratic function is proposed to measure the proliferation of pests which depends, essentially, of the crops' botanical families and of the distance between the respective plots. This minimization ensures a decrease in the use of chemical pesticides to combat pests and weeds. Moreover, the maintenance of several crops in the same farm maintains the soil fertile, with high concentration of nitrogen and organic matter, thus reducing the amount of artificial fertilizers. The second objective is purely economic and is defined to maximize the total profit of the plantation along the planning horizon.

To determine Pareto optimal solutions for the bi-objective binary quadratic optimization problem, the weighted sum method was used. The mono-objective problems of this approach were initially solved using an integer quadratic programming solver. In a second phase, a linearization strategy for the original model was applied, thus obtaining a mixed binary linear programming formulation. The respective mono-objective problems were solved using an integer linear programming solver. The high number of integer variables and the computational difficulties arising when running the solvers caused the authors to develop a genetic algorithm (GA) to solve the mono-objective problems of the weighted sum method. Two constructive heuristics were used, with the purpose of improving the performance of the GA. Computational results have shown that this methodology is efficient, in the sense that it obtains potentially Pareto optimal solutions for the bi-objective problem in a CPU time below the exact methods, specially for the high dimension instances. This enables us to assume that

the developed heuristic algorithm is an excellent tool for supporting decision-making in this field.

This work introduces the following innovative aspects:

- The adaptation of a bi-objective binary mathematical programming model for sustainable cultivation;
- A comparative computational study between integer quadratic and linear versions of a mono-objective problem used to determine Pareto optimal solutions for the problem of sustainable cultivation;
- The development and computational testing of a GA, embedding two constructive heuristics which determines potentially Pareto optimal solutions for the problem of sustainable cultivation.

The remainder of this paper is organized as follows. Section 2 briefly presents the problem of sustainable cultivation and correlated studies in the operational research area. Section 3 presents the mathematical modeling of the problem, including a quadratic and a linearized model, all employed to obtain Pareto optimal solutions for the problem. Section 4 describes the GA proposed, including the constructive heuristics necessary for this algorithm. In Sect. 5, 27 real-size instances are also introduced to computationally validate the metaheuristic and the mathematical models. Finally, Sect. 6 describes the conclusions of this work.

2 Sustainable agriculture and crop rotation

According to Tilman et al. (2002), the global demand for food in 2052 will double its figures in 2002. As predicted by these authors, food production designed to guarantee sustainability is a huge challenge for agro-industrial management. New studies and new incentives are required to find a way to improve food production, while preserving environmental integrity and public health.

Parra (2002) presented a work that clarifies the advantages of using the crop rotation technique and shows the benefits of this technique to the natural control of pests, soil protection and water resources. As stated by this author, a sustainable agriculture committed to the environment and natural resources must intensively use crops planted sequentially in the same field. It is an important practice for any sustainable agricultural system because different crops improve the physical, chemical, and biological characteristics of the soil. This practice retains soil fertility and prevents the appearance of disease, insect infestation, and weeds.

According to Havlin et al. (1990), West and Post (2002) and Costa et al. (2014), maintaining and increasing soil productivity depends, essentially, on the management practices of the crops. Usually, they cultivate organic leaf matter and increase nitrogen and carbon fixation by the atmosphere in the soil, all vital components in good production. In this context, crop rotation is an excellent technique to increase soil productivity.

Planning a rotation with diverse crops is not an easy task. Mathematical modeling with operational research techniques can be useful tools in decision-making. Santos et al. (2010) considered the problem of crop rotation in which one must meet known crop demands while respecting ecologically-based production constraints. The authors proposed a binary linear optimization model, where each variable is associated with a feasible crop rotation plan. Due to the large number of variables of the model, a column generation method was used to solve it. Santos et al. (2011) proposed a binary linear optimization model to determine a schedule of crop rotation for each plot, thus maximizing land occupation time. In addition to the restrictions considered in the previous work, planting constraints for adjacent plots and

for crop sequences in the rotations were included. The authors proposed a column generation heuristic to deal with the large number of model variables. In Santos et al. (2015b) a binary programming formulation for this last problem, but considering decisions on plot sizes and schedules was proposed. A branch-price-and-cut algorithm was developed and extensive computational experiments over a set of instances based on real-life data were performed.

Alfandari et al. (2011) proposed a Mixed-Integer Linear Programming model for a multi-period crop rotation optimization problems with demand constraints and incompatibility constraints between cultivation and fallow state on a plot. The model, whose objective was maximize the control agricultural space, was applied to a case study on Madagascan farms.

Frequently, optimization is based on a single monetary criterion, for instance profit maximization. Some authors, as Dury et al. (2012), pointed out the limitations of an approach that focuses exclusively on maximizing returns. Therefore, the growing environmental concerns led researchers to explicitly segment other objectives beyond profitability, such as minimization of total labor, the equipment employed, herbicide usage, losses, pesticide exposure, erosion; and maximization of irrigated areas, organic matters' rate of change, gross margin, among others.

In this new perspective, models with multiple objectives have increasingly been used in agriculture applications, as mentioned by Huang et al. (2011). The multi-objective methodologies are becoming popular due to the flexibility that allows decision-makers to consider all the criteria and objectives simultaneously (Kumar et al. 2017). Dogliotti et al. (2005) developed an approach to support farmers with temporal complex interactions in crop rotations and spatial heterogeneity, revealing a trade-off between the economic and environmental objectives. These authors proposed a mixed integer linear programming model to allocate production activities at a farm with land units differing in soil quality, while minimizing the socioeconomic and environmental objectives, subject to constraints at the farm level.

Hayashi (2000) presented a review of multicriteria analysis applied to agricultural resource management. Methods for selecting multi-attribute discrete alternatives and solving multi-objective planning problems are revised. Annetts and Audsley (2002) proposed multi-objective linear programming models that consider a wide range of farming situations, which permits optimization of profit or environmental outcome, or in fact both. The objective is to identify the best cropping and machinery options which are both profitable and result in improvements to the environment.

The recent paper by Boyabathl et al. (2019) examines crop rotation in a sustainable context taking into account revenue uncertainty and allocation of multiple crops to the plots. The authors have shown that the cultivation of unique crop over the entire planning horizon, as employed in industrial agriculture, leads to a considerable profit loss and soil wear. The paper by Nechi et al. (2019) develops a model of goal programming with several criteria, where stakeholders' preferences are explicitly integrated within a group decision-making process based on consensus and tradeoffs.

In the last 15 years, evolutionary algorithms have been used to support multi-objective cropping plan decisions. Aliano Filho et al. (2014) addressed a crop rotation problem, modeled as a binary linear programming problem. The objective is to maximize the profit from rotation, while incorporating the same ecological constraints of the model presented in Santos et al. (2015b). The authors proposed metaheuristics to solve the problem, as well as greedy constructive heuristics, which inspired the methods presented in this paper.

According to Rossing and Hammer (2006), the main advantage of using genetic algorithms is to produce a set of compromise solutions along the Pareto frontier entailing a low computation effort. These authors used a multi-objective evolutionary algorithm to solve a crop choice model, and applied it in North East Australia. The concept of sustainability

of the cultivating systems is considered in terms of resources used besides other economic aspects, but does not take into account the total profit of the agricultural activity. Sarker and Ray (2009) formulated a crop-planning problem as a multi-objective optimization model and provided useful insights into solutions that are generated by using three different evolutionary strategies. The authors also considered a bi-objective formulation with continuous variables for crop rotation, where the first objective is to maximize the total gross margin (from cultivated plus imported crops) in a single crop year. The second one is to minimize the total working capital required. These are two conflicting economic objectives.

In our paper, we highlight some innovative aspects, such as the adaptation of a crop programming model [inspired in the works Santos et al. (2010, 2011, 2015b) and Aliano Filho et al. (2014)] and introduce two objective functions to balance sustainability and profit. The term “sustainability” is employed in this context because the mathematical model promotes the choice of crops to entry in the planting schedule taking into account environmental degradation, through reduction in intensive pesticide use and the maintenance of soil natural resources.

The methodologies developed provide the manager with a set of several solutions and correspond to distinct scenarios that facilitate his decision-making. We will show that the multi-objective optimization tool enables one to analyze a reduction in environmental impact, along with a modest decrease in profits.

In addition, linked to computational difficulties of the proposed mono-objective (quadratic and linear) models for the weighted sum method, a comparative computational study with different weights given to each function, is performed. To complete this analysis, we conducted a comparison between the exact and heuristic solution methodologies. These aspects show that this study addresses some points that were not covered by the specific literature of this field.

The problem of sustainable cultivation studied can encompass two distinct production scenarios. In the first scenario—cereal cultivation applied in areas of high production—where there are fewer varieties, we highlight soybeans, corn, wheat, rice, and beans as the most common crops. In the second situation, small farmers cultivate on a small scale, in response to a demand from the region, whose focus is placed on vegetables. In this case, the number of cultivated varieties (N) and plots (K) are larger (N can vary from 10 to 20 varieties and K varies from 10 to 15 plots), and the varieties are lettuces, potatoes, tomatoes, peppers, watermelons, and pumpkins, among others.

In summarizing, the problem of sustainable cultivation consists in determining efficient solutions or Pareto optimal solutions to the bi-objective problem of scheduling different varieties of plants/crops in a given set of plots. The planting schedule is characterized by the crops that should be cultivated, the dates and plots where they should be planted, with the purpose of minimizing the possibility of pest proliferation among the crops in the whole planting area and to maximize the profit obtainable in a specified planning horizon. Constraints for crop planting of the same botanical family in alternated plots, for non-overlapping planting and forcing all the plots to share the same cycle duration are also considered.

3 Mathematical modeling

This section is devoted to the quadratic and the linear bi-objective binary mathematical programming models developed for the problem of sustainable cultivation. The constraints associated with non-overlapping of planting constraints and cultivating consecutive plants in

the same plot, were inspired by Santos et al. (2010, 2011, 2015b). The objective function regarding the possibility of pest proliferation is originally proposed by the authors in this paper. The additional components of the formulation considered in this work are based on Aliano Filho et al. (2014).

The indexes and parameters for the models are defined as follows:

Indexes

- i and \bar{i} : related to the crops/cultures;
- j and \bar{j} : related to the plots;
- t : related to the planning horizon periods;
- p : related to the botanical family of plants.

Parameters

- N : number of crops available for planting ($N \geq 2$);
- K : number of available plots;
- T : duration of the planning horizon;
- N_f : number of botanical families of plants ($N_f \geq 2$);
- F_p : set of crops belonging to the family p , $p = 1, \dots, N_f$;
- C_i : duration of crop life cycle i ;
- l_i : profitability of crop i per hectare;
- $area_j$: area of plot j in hectares;
- $P_{i\bar{i}}$: probability of pest infestations between the crops i and \bar{i} . Pest infestation has a higher probability for similar crops and lower probability when the crops are botanically different. Therefore, we propose the following probability of infestation:

$$P_{i\bar{i}} = \begin{cases} 0.9, & \text{if } i \text{ and } \bar{i} \text{ are from the same botanical families,} \\ 0.5, & \text{if } i \text{ and } \bar{i} \text{ are from botanical families of degree between 2 and 3,} \\ 0.1, & \text{if } i \text{ and } \bar{i} \text{ are botanical families with degree equal greater than 4,} \end{cases}$$

where the degree between two crops i and \bar{i} ($i \neq \bar{i}$) is defined as the difference (in module) in the index of the botanical families to which they belong. The decision about three distinct and spaced values for the probability of pest proliferation between two crops came from previous computational experiments performed with the aim of strongly penalizing plantation of similar crops in nearby plots, thus promoting the cultivation of very distinct crops in nearby plots.

- $V_{j\bar{j}}$: the probability of pest infestation from plot j to plot \bar{j} , $j \neq \bar{j}$, and vice-versa. We propose the following formula to calculate $V_{j\bar{j}}$:

$$V_{j\bar{j}} = \frac{1}{1 + d_{j\bar{j}}},$$

in which $d_{j\bar{j}}$ is the shortest distance between plots j and \bar{j} . Note that, if $d_{j\bar{j}} \rightarrow 0$ then $V_{j\bar{j}} \rightarrow 1$; if $d_{j\bar{j}} \rightarrow \infty$ then $V_{j\bar{j}} \rightarrow 0$. This means that neighboring plots are more likely to be simultaneously infected.

The decision variables are defined as follows:

$$x_{ijt} = \begin{cases} 1, & \text{if crop } i \text{ is planted in plot } j \text{ in the period } t, \\ 0, & \text{otherwise,} \end{cases}$$

for all $i = 1, \dots, N$, $j = 1, \dots, K$ and $t = 1, \dots, T$.

A bi-objective quadratic binary formulation for the problem of sustainable cultivation is presented as follows:

$$\text{minimize } z_1 = \sum_{i=1}^N \sum_{\bar{i}=1}^N \sum_{j=1}^{K-1} \sum_{\bar{j}=j+1}^K \sum_{t=1}^T P_{i\bar{i}} \cdot V_{j\bar{j}} \cdot \left(\sum_{r=0}^{C_i-1} x_{ij(t-r)} \right) \cdot \left(\sum_{r=0}^{C_{\bar{i}}-1} x_{\bar{i}\bar{j}(t-r)} \right) \quad (1)$$

$$\text{maximize } z_2 = \sum_{i=1}^N \sum_{j=1}^K \sum_{t=1}^T l_i \cdot \text{area}_j \cdot x_{ijt} \quad (2)$$

$$\text{subject to } \sum_{i=1}^N \sum_{r=0}^{C_i-1} x_{ij(t-r)} \leq 1, \quad j = 1, \dots, K, \quad t = 1, \dots, T \quad (3)$$

$$\sum_{i \in F_p} \sum_{r=0}^{C_i} x_{ij(t-r)} \leq 1, \quad p = 1, \dots, N_f, \quad j = 1, \dots, K, \quad t = 1, \dots, T \quad (4)$$

$$\sum_{i=1}^N \sum_{t=1}^T C_i \cdot x_{ijt} = T, \quad j = 1, \dots, K \quad (5)$$

$$x_{ijt} \in \{0, 1\}, \quad i = 1, \dots, N, \quad j = 1, \dots, K, \quad t = 1, \dots, T, \quad (6)$$

where $t - r \leq 0$ is replaced by $t - r + T$.

The objective function (1), to be minimized, represents the possibility of pest proliferation throughout the planting area, when considering the whole planning horizon. The closer the crops of the same botanical family of plants are, the greater the proliferation. Maximization of objective function (2) aims to optimize the profit of the planting schedule. Note that these objectives conflict with each other. On the one hand, if this last function is maximized, only the crops that provide greater profit will be selected for planting. Hence, the variability of the crops chosen to be planted will be reduced. On the other hand, if only z_1 is minimized, crops of different botanical families offering a lower profit should be chosen, thus resulting in a lower overall profit.

Constraints (3) prevent planting overlapping, that is, constraints which enable a new crop to be inserted in a plot before concluding the C_i periods of cultivation of the previous crop (i) in the plot. Constraints (4) prevent crops of the same botanical family from being cultivated in the same plot in consecutive periods. Constraints (5) ensure that all the T periods of the planning horizon are used in each plot. Finally, constraints (6) impose the domain of all the decision variables of this model.

It is important to note that some previous papers, such as Santos et al. (2010, 2011, 2015b) and Aliano Filho et al. (2014), consider fallow practice as an important requirement in the elaboration of a crop rotation schedule. However, we do not require this practice. From the economic point of view, it is of no interest for farmers to leave an area fallow for a certain period, but rather to always ensure it is occupied with a profitable crop.

To obtain Pareto optimal solutions for this bi-objective problem, we balanced z_1 and z_2 by attributing weights to these functions and minimizing the resulting sum. In this way we solved the mono-objective problem:

$$\begin{aligned} &\text{minimize } z = \lambda \cdot \beta_1 \cdot z_1 + (1 - \lambda) \cdot \beta_2 \cdot (-z_2) \\ &\text{subject to constraints (3)–(6),} \end{aligned} \quad (7)$$

where $\lambda \in [0, 1]$. According to Miettinen (2012), for each value of $\lambda > 0$, the solution to this mono-objective problem is efficient for the problem (1)–(6). The constants β_1 and β_2

selected in the range $[0, 1]$ normalize the objectives. Such constants may be determined by solving the problems:

$$\mathbf{x}_1^* = \arg \min\{z_1 : \text{s.t. (3)–(6)}\} \quad \text{and} \quad \mathbf{x}_2^* = \arg \max\{z_2 : \text{s.t. (3)–(6)}\}, \tag{8}$$

and, calculating:

$$\beta_1 = \frac{1}{z_1(\mathbf{x}_2^*) - z_1(\mathbf{x}_1^*)} \quad \text{and} \quad \beta_2 = \frac{1}{z_2(\mathbf{x}_1^*) - z_2(\mathbf{x}_2^*)}. \tag{9}$$

As observed, the models (1)–(6) and (7) are nonlinear. More precisely, the function z_1 is a quadratic one. The inconvenience of these models is that there is no guarantee that they are convex, thus causing difficulties for the integer quadratic programming solvers. In an attempt to circumvent this problem, we adopted an integer programming modeling technique to linearize the models, which consists in replacing the product $(\sum_{r=0}^{C_i-1} x_{ij(t-r)}) \cdot (\sum_{r=0}^{C_{\bar{i}}-1} x_{\bar{i}\bar{j}\bar{j}t})$ by $z_{\bar{i}\bar{i}\bar{j}\bar{j}t}$ and imposing the linear constraints:

$$\sum_{r=0}^{C_i-1} x_{ij(t-r)} + \sum_{r=0}^{C_{\bar{i}}-1} x_{\bar{i}\bar{j}\bar{j}(t-r)} - z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \leq 1, \tag{10}$$

$$z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \leq \sum_{r=0}^{C_i-1} x_{ij(t-r)}, \tag{11}$$

$$z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \leq \sum_{r=0}^{C_{\bar{i}}-1} x_{\bar{i}\bar{j}\bar{j}(t-r)}, \tag{12}$$

for all $i = 1, \dots, N, \bar{i} = 1, \dots, N, j = 1, \dots, K - 1, \bar{j} = j + 1, \dots, K, t = 1, \dots, T$ where $z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \geq 0$ are the new variables.

The linearized version of model (1)–(6) is presented below:

$$\text{minimize} \quad z_1^l = \sum_{i=1}^N \sum_{\bar{i}=1}^N \sum_{j=1}^{K-1} \sum_{\bar{j}=j+1}^K \sum_{t=1}^T P_{\bar{i}\bar{i}} \cdot V_{j\bar{j}} \cdot z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \tag{13}$$

$$\text{maximize} \quad z_2 = \sum_{i=1}^N \sum_{j=1}^K \sum_{t=1}^T l_i \cdot \text{area}_j \cdot x_{ijt} \tag{14}$$

$$\text{subject to} \quad \sum_{i=1}^N \sum_{r=0}^{C_i-1} x_{ij(t-r)} \leq 1, \quad j = 1, \dots, K, \quad t = 1, \dots, T \tag{15}$$

$$\sum_{i \in F_p} \sum_{r=0}^{C_i} x_{ij(t-r)} \leq 1, \quad p=1, \dots, N_f, \quad j=1, \dots, K, \quad t=1, \dots, T \tag{16}$$

$$\sum_{i=1}^N \sum_{t=1}^T C_i \cdot x_{ijt} = T, \quad j = 1, \dots, K \tag{17}$$

$$\text{constraints (10)–(12)} \tag{18}$$

$$x_{ijt} \in \{0, 1\}, \quad z_{\bar{i}\bar{i}\bar{j}\bar{j}t} \geq 0, \tag{19}$$

$$i = 1, \dots, N, \quad \bar{i} = 1, \dots, N, \quad j = 1, \dots, K - 1, \quad \bar{j} = j + 1, \dots, K, \quad t = 1, \dots, T, \tag{20}$$

where z_1^l is z_1 linearized and $t - r \leq 0$ is replaced by $t - r + T$. The linear reformulation of the weighted sum model is given by:

$$\begin{aligned} &\text{minimize } z = \lambda \cdot \beta_1 \cdot z_1^l + (1 - \lambda) \cdot \beta_2 \cdot (-z_2) \\ &\text{subject to constraints (15)–(20).} \end{aligned} \tag{21}$$

In Sect. 5, we compare the performance of standard software applied to Problem (7) and its linearized version (21) using three values of λ . The processing time of these exact methods corresponding to each of these weights is verified. The Pareto optimal solutions determined are also commented on. However, these exact methods were unable to tackle large instances. In fact, for several instances, the solver did not even find a feasible integer solution to the quadratic Problem (7) and to the linear Problem (21), in one hour’s execution.

4 The genetic algorithm

In this section a genetic algorithm (GA) with two constructive heuristics for the problem of sustainable cultivation is presented. We opted for the metaheuristic GA because it is simple to implement, it has incurred low computational costs and provided good results for combinatorial problems according to Han and Kim (2000), Deb (2001), Rossing and Hammer (2006), Sarker and Ray (2009) and de Oliveira Florentino et al. (2018), among other studies.

The core features of the proposed GA are the solution encoding and mainly the constructive heuristics developed for the initial population, all presented with additional details of this method in the following subsections.

4.1 Solution encoding

In the GA for the problem studied in this work, an individual corresponds to a solution, feasible or otherwise, to the problem. The strategy of encoding that follows was inspired in Aliano Filho et al. (2014) and de Oliveira Florentino et al. (2018). Each individual $\mathbf{I} = (i_{kt}) \in \mathbb{N}^{K \times T}$ is a matrix, and its entry $i_{kt} \in \{1, \dots, N\}$ represents the crop index that is planted in lot k and in period t , $k = 1, \dots, K$ and $t = 1, \dots, T$. This codification has the advantage of being simple to implement and provides all the information required. To facilitate ones understanding of solution encoding, we present an illustrative example.

Example 1 (Illustration of a feasible solution) The data for the crops are shown in “Appendix”, Table 11. Let $N = 20$, $K = 9$ and $T = 12$. A possible solution to this problem could be given by the matrix $\mathbf{I} \in \mathbb{N}^{9 \times 12}$ below:

$$\mathbf{I} = \begin{bmatrix} 11 & 6 & 6 & 8 & 8 & 18 & 18 & 15 & 15 & 17 & 17 & 11 \\ 8 & 16 & 1 & 1 & 13 & 13 & 13 & 9 & 9 & 9 & 20 & 8 \\ 7 & 7 & 17 & 17 & 11 & 11 & 2 & 2 & 2 & 4 & 4 & 7 \\ 18 & 13 & 13 & 13 & 6 & 6 & 11 & 11 & 4 & 4 & 20 & 18 \\ 8 & 19 & 19 & 19 & 7 & 7 & 7 & 6 & 6 & 10 & 20 & 8 \\ 20 & 11 & 11 & 8 & 8 & 5 & 5 & 18 & 18 & 4 & 4 & 16 \\ 9 & 9 & 9 & 4 & 4 & 13 & 13 & 13 & 6 & 6 & 10 & 20 \\ 9 & 2 & 2 & 2 & 6 & 6 & 16 & 10 & 15 & 15 & 9 & 9 \\ 15 & 11 & 11 & 4 & 4 & 1 & 1 & 7 & 7 & 7 & 20 & 15 \end{bmatrix}.$$

In this solution, the planting of crop $i = 6$ is programmed in plot $k = 1$ in period $t = 2$ and it stays in this plot until it is harvested in the period $t = 3$, when the crop $i = 8$ enters in period $t = 4$ and remains until period $t = 5$. Finally, in period $t = 12$, crop $i = 11$ is

planted and occupies the space until its planting period has expired, that is, at $t = 13$ (which corresponds to the first period, according to the mathematical formulation).

4.2 Initial population

The initial population of the GA is carefully created. The basic idea consists in generating individuals with enough diversity to allow the algorithm to combine the varied characteristics of the individuals and produce diverse potentially Pareto optimal solutions to the problem.

This initial population is generated by two different heuristics, named MINPEST and MAXPROFIT. We try to create solutions with distinct characteristics in order to optimize the two conflicting functions, z_1 and z_2 . One of the constructive heuristic attempts to produce solutions that have a low probability of pest proliferation among the crops, by mixing different plants. The other constructive heuristic attempts to obtain high profit-making solutions, by forcing only the more profitable crops to enter the schedule. These two ways of building solutions are necessary for this GA, because the objectives conflict with each other. According to the weight λ given to the objective z_1 , the GA will determine a good solution with respect to the function $z = \lambda \cdot \beta_1 \cdot z_1 + (1 - \lambda) \cdot \beta_2 \cdot z_2$.

Let n be the total number of individuals of this population¹. It is calculated as follows:

- $\frac{n}{2}$ by the MINPEST heuristic;
- $\frac{n}{2}$ by the MAXPROFIT heuristic.

The two constructive heuristics, MINPEST and MAXPROFIT, are presented in the following subsections.

4.2.1 MINPEST heuristic

This heuristic is based on the heuristic in Aliano Filho et al. (2014), which builds a solution to Problem (21) with $\lambda = 0.0$. This heuristic aims to ensure the sequential insertion of different botanical families in the same plot. Each individual satisfies the constraints of the problem, except the period continuity at $t = T$ and $t = 1$. It is possible that the crops of these periods may belong to the same family. Should that happen and a shift of elements (in the same lot) is performed, then this inadmissibility may occur in intermediate periods of the planning horizon. Later, we will consider a way to overcome this drawback.

The pseudocode of this heuristic is in Algorithm 4.1. Initially, for each lot k we sort a permutation \mathcal{P}_k of crops' indexes. We define the sequence of crops \mathbf{v}_k containing the planting schedule for plot k and start to run along vector \mathcal{P}_k and seek the first element whose crop i is not from the same botanical family of the last crop inserted in \mathbf{v}_k . This ensures that the constraints of consecutive planting are always respected. After finding one such crop, it will be inserted in the schedule and \mathbf{v}_k is updated. The construction is made until the period of the new crop to be inserted in \mathbf{v}_k , added to the number of periods of the crops that were already inserted in \mathbf{v}_k is equal to T . Should any crop in \mathcal{P}_k satisfy this, a new permutation is generated.

With the indexes of the crops to be planted in plot k in the T periods, we perform the procedure referred to as *prolongation*, described as follows. Let the row k of the individual \mathbf{I} be denoted by \mathbf{I}_k . This row is built from the vector \mathbf{v}_k . Each crop i is repeated as many times as the duration of its period of cultivation, C_i . This is done until all the crops of \mathbf{v}_k

¹ The number n must be a multiple of 2, because $n/2$ needs to be integer.

are considered. In the end, \mathbf{I}_k has length T . Therefore, the final individual \mathbf{I} is the string concatenation (in rows) of \mathbf{I}_k .

This heuristic produces solutions that always start their cultivation in period $t = 1$ and end in period $t = T$. In general, the solution does not necessarily possess this characteristic, which can be seen in individual \mathbf{I} from Example 1. To produce greater variability in the population solutions, we perform a procedure named *shift* in each row \mathbf{I}_k of \mathbf{I} , with a probability of 70%. If it occurs, we shift the elements of \mathbf{I}_k in a positions to the right of their original positions. Here, a is an integer randomly generated from 1 to T (uniform distribution).

Example 2 (Construction of an individual with the MINPEST heuristic) We will build the scheduling for lot $k = 1, \mathbf{I}_1$, considering the data of the crops displayed in Table 11, $N = 10$ and $T = 12$.

Step 1 We randomly generated $\mathcal{P}_1 = (5, 10, 8, 7, 1, 9, 2, 6, 4, 3)$.

Step 2 Let $i = 5$ be the first crop to be inserted in the schedule. The calendar for planting crop 5 has $C_5 = 2$ duration periods.

Step 3 Let $i = 10$ be the second crop of the permutation. It is inserted in the schedule as it is not from the same botanical family of the previous inserted crop. Then, we are in period $C_5 + C_{10} = 2 + 1 = 3$.

Step 4 The crop $i = 8$ is the third crop inserted in this plot. We already have $C_5 + C_{10} + C_8 = 2 + 1 + 2 = 5$ periods occupied.

Step 5 The next crop of the permutation is $i = 7$. It cannot be inserted in the current period as the last crop planted belongs to the same family. The next crop analyzed is $i = 1$, which is not from the same botanical family. Then, the period occupied is updated to $C_5 + C_{10} + C_8 + C_1 = 2 + 1 + 2 + 2 = 7$.

Step 6 We analyze crop $i = 9$, and it is introduced into the schedule. The current planting period is $C_5 + C_{10} + C_8 + C_1 + C_9 = 2 + 1 + 2 + 2 + 3 = 10$.

Step 7 Analysis of the next crop, $i = 2$, tells us it cannot be inserted in the schedule, because the duration of its cycle is greater than the number of periods needed to complete the cycle, $T = 12$. In sequence, we analyze crop $i = 6$ which possesses precisely the 2 periods required to finish the schedule.

Step 8 Using the *prolongation* operator, we have the vector \mathbf{I}_1 :

$$\mathbf{I}_1 = [5 \ 5 \ 10 \ 8 \ 8 \ 1 \ 1 \ 9 \ 9 \ 9 \ 6 \ 6].$$

(Note that it corresponds to an unfeasible solution, since the first and the last crops of the schedule are both from family 3).

Step 9 If the operator *shift* acts in this vector with $a = 3$ positions, we have a new (unfeasible) schedule for this plot given by:

$$\mathbf{I}_1 = [9 \ 6 \ 6 \ 5 \ 5 \ 10 \ 8 \ 8 \ 1 \ 1 \ 9 \ 9].$$

4.2.2 MAXPROFIT Heuristic

This is a constructive heuristic we have developed in this study. It is specialized in producing solutions with higher profitability. A parameter α is introduced to control the “strength” and “pressure” that one wants to ascribe to the profit of a planting schedule.

Consider a set of crops that the agriculture decision-maker wants to insert in the schedule of plot k , with cardinality $\alpha \geq 2$ (if $\alpha < 2$, there is no a feasible solution regarding the continuity constraints). The more profitable crops from different botanical families must be chosen to compose this set and they are randomly permuted, thus producing the sequence

of crops C_k . Then, the MINPEST heuristic is employed to generate an individual of the GA, i.e., we use the set C_k instead of \mathcal{P}_k for each plot. A smaller value of α produces a solution with a greater profit. In the same way, we apply the *shift* and *prolongation* operators to each plot. The idea supporting this heuristic is the production of more profitable solutions (which also implies an increase in pest infestations) and, consequently, the introduction of diversity in the population of the GA.

Example 3 (Construction of an individual with the MAXPROFIT heuristic) Suppose that $\alpha = 4$, and that the most profitable crops from Table 11, belonging to families 1 and 2, are used to build the planting schedule in $K = 5$ plots. We will only illustrate the construction of one plot. Let the sequence of these crops be $C_1 = (3, 1, 2, 4)$.

Algorithm 4.1 MINPEST heuristic

```

1: Input: crop data
2: for  $k = 1, \dots, K$  do
3:    $\mathbf{v}_k = \emptyset$ 
4:    $duration = 0$ 
5:   Generate a random permutation  $\mathcal{P}_k$  of  $N$  crops.
6:   while  $duration < T$  do
7:     for  $i \in \mathcal{P}_k$  do
8:       if the last crop inserted in  $\mathbf{v}_k$  is not from the same family as crop  $i$  then
9:         if  $C_i \leq T - duration$  then
10:           Insert  $i$  in the sequence  $\mathbf{v}_k$ 
11:            $duration = duration + C_i$ 
12:           if  $duration = T$  then
13:             break
14:           end if
15:         end if
16:       end if
17:     end for
18:     if  $duration \neq T$  then
19:       got to step 7
20:     end if
21:     Apply the prolongation operator with vector  $\mathbf{I}_k$  with the sequence  $\mathbf{v}_k$ 
22:     Apply the shift procedure
23:      $\mathbf{I} = \mathbf{I} \cup \mathbf{I}_k$  (row concatenation)
24:   end while
25: end for
26: Output:  $\mathbf{I}$ , a solution (feasible or not) for the sustainable cultivation problem

```

Step 1 crop $i = 3$ is inserted in the schedule followed by crop $i = 1$. The next crop, $i = 2$, cannot be planted, because it is from the same botanical family as the previous one. Then, crop $i = 4$ is inserted. The sum of periods of the three crops results in $5 < 12 = T$ periods.

Step 2 The insertion of crops continues, starting with the first element of C_1 , which in turn cannot be inserted at this moment. The crops $i = 1$ and $i = 4$ are the next to be allocated to this schedule. The duration of the planting cycle at this stage is $9 < 12 = T$.

Step 3 Starting the scanning once more through the sequence, crops $i = 1$ and $i = 3$ are the last crops to be inserted. Following application of the *prolongation* operator, we have the vector \mathbf{I}_1 :

$$\mathbf{I}_1 = [3 \quad 1 \quad 1 \quad 4 \quad 4 \quad 1 \quad 1 \quad 4 \quad 4 \quad 1 \quad 1 \quad 3],$$

which is infeasible, because crop 3 is planted in sequence (periods 1 and 12).

Step 4 Using the *shift* operator with $a = 4$, we have the following schedule of this plot:

$$\mathbf{I}_1 = [4 \quad 1 \quad 1 \quad 3 \quad 3 \quad 1 \quad 1 \quad 4 \quad 4 \quad 1 \quad 1 \quad 4],$$

which is infeasible for the problem, due to the planting of crop 3.

For the construction of the remaining plots using set \mathcal{C}_1 , we consider other random permutations of the elements from \mathcal{C}_1 . Our concern is to always examine the feasibility of consecutive crops in the intermediate periods, that is, if they are from different families.

The pseudocode of this procedure is similar to the Algorithm 4.1, except for line 5, where the list of crops to be inserted is \mathcal{C}_k .

4.3 Fitness evaluation

The fitness $f(\mathbf{I})$ of each individual \mathbf{I} in the population is given by

$$f(\mathbf{I}) = \lambda \cdot \beta_1 \cdot z_1 + (1 - \lambda) \cdot \beta_2 \cdot (-z_2) + 10^3 \cdot v(\mathbf{I}), \quad (22)$$

where β_1 and β_2 are calculated through the Expression (8), $0.0 < \lambda < 1.0$.

The function $v(\mathbf{I})$ provides the number of violations that corresponds to \mathbf{I} , with regard to the constraints of consecutive planting. Violations occur because the MINPEST and MAX-PROFIT heuristics can determine infeasible solutions as to consecutive planting. The first and last crop of the schedule can be from the same botanical family. Also, following the action of *shift* operator, this inadmissibility can be passed on to intermediate periods. The number of plots in which this occurs is the number of violations $v(\mathbf{I})$ to be considered in the fitness of the individual. Therefore, a feasible individual \mathbf{I} has a null violation, $v(\mathbf{I}) = 0$, and an unfeasible one has $v(\mathbf{I}) > 0$ and tends to be extinct from the population in subsequent generations.

4.4 Selection

The process of selecting the individuals in the population consists of choosing which elements will generate descendants for the next generation. Therefore, $\lfloor \gamma_1 \cdot n \rfloor$ individuals are separated to perform the *crossover*, another genetic operator of this method. The parameter γ_1 , called the selection rate, is defined by the user. The *selection* consists of performing tournaments with two randomly chosen elements of the population, and the one with the lowest fitness is chosen (the best one). The procedure ends when the $\lfloor \gamma_1 \cdot n \rfloor$ individuals are chosen, thus building, the intermediate population.

4.5 Crossover

The main objective of this operator is to produce new solutions from the previously built ones, by exploring the feasible space of the problem and finding more promising solutions with respect to fitness value, i.e., solutions with lower fitness value. This operator randomly chooses $\lfloor \gamma_1 \cdot n \rfloor / 2$ pairs of individuals from the intermediate population. Each pair (\mathbf{F} , \mathbf{M}) is considered a pair of parents father and a mother, respectively, and their genes (the rows of the respective matrix) are swapped between the two parents thus resulting in two offspring (\mathbf{C}_1 and \mathbf{C}_2), each bearing some genetic information from both \mathbf{F} and \mathbf{M} . In the *crossover*,

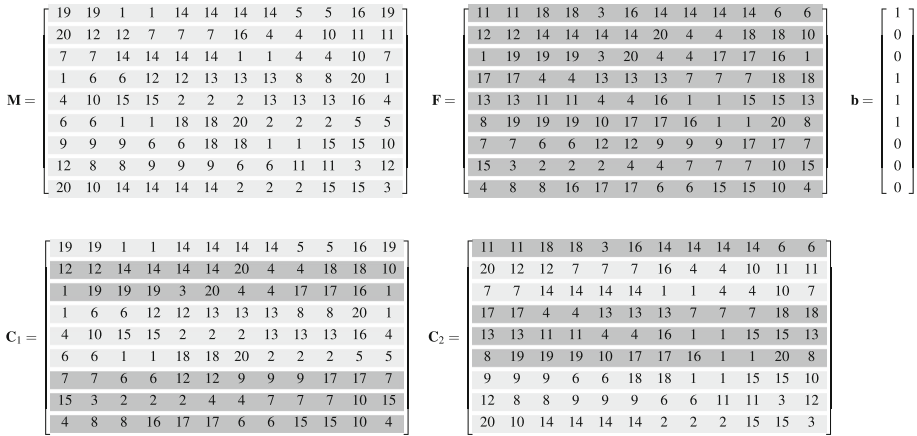


Fig. 1 Illustration of the uniform crossover

a random binary vector $b = (b_k)$ of length K is considered. If $b_k = 0$, then the child C_1 has its k th row copied from the father. If $b_k = 1$, then the k th row comes from the mother. The child C_2 is built inversely, that is, if $b_k = 0$ the information comes from the mother and, otherwise, it comes from the father. This strategy, besides incurring low computational cost, is able to generate different and promising solutions to the problem.

Figure 1 schematically illustrates this uniform operator.

4.6 Mutation

The *mutation* acts on $\lfloor \gamma_2 \cdot n \rfloor$ (where γ_2 is the mutation rate) randomly choosing individuals of the population. The *mutation* avoids the premature convergence of the algorithm. The following function represents the probability for this operator to act in a given generation g of the GA

$$\phi(g) = \frac{1}{1 + 20e^{-g/40}} \tag{23}$$

When $g \rightarrow G$ (where G is the maximum number of generations), $\phi(g) \rightarrow 1$. This means that *mutation* occurs more strongly in the final generations of the GA, when the individuals of the population tend to be similar. For each individual I that mutates, we consider a probability of 50% to replace each row I_k of I . If the replacement occurs, the MINPEST heuristic generates another schedule for I_k .

4.7 Migration

Similar to the *mutation* process, we proposed an operator called *migration*. The purpose is again to ensure diversity in the population, and, consequently, to determine the most promising solutions. A mechanism to replace the $\lfloor \gamma_3 \cdot n \rfloor$ elements of the population (where γ_3 is a given parameter) is used in three moments throughout the course of the GA, more precisely, in generations $g = 0.5G$, $g = 0.7G$ and $g = 0.9G$. In these generations, 50% of the new individuals (immigrants) are generated by the MINPEST heuristic and the remaining 50% by the MAXPROFIT heuristic.

4.8 Update and Elitism

After the *selection*, *crossover*, *mutation*, and *migration*, the population increases by the factor of $1 + \gamma_1$. All the individuals are evaluated, according to Expression (22), and the n best are taken to the next generation. Elitism consists in preserving the best element (called *Elite* and denoted by \mathcal{E}) before the operators act. \mathcal{E} is included in the population before updating at the end of any generation, and constantly updated at each generation. This means that the GA, at least, preserves the best solution achieved in previous generations. Should a better individual than current \mathcal{E} appear in the population, the *elite* individual is updated. The GA method ends its search after G generations.

It is important to note that this GA determines one solution for the problem according to the weight λ given to the objective function z_1 (and $1 - \lambda$ given to z_2). This means that the method determines a potentially Pareto optimal solution of Problem (1)–(6). The approximation of all Pareto optimal solutions would be a computationally expensive task, since the problem can have a large number of such solutions. In the next section, we will demonstrate that each solution demands moderate to high CPU time.

Consequently, we propose to determine only three of these solutions that will give a reasonable representation of different solutions for the Problem (1)–(6). For two of the three solutions, we individually optimize the objectives, thus obtaining the so-called *potentially lexicographic solutions* (Miettinen 2012), when $\lambda = 0.0$ and $\lambda = 1.0$. The third one is obtained by using a combination of the same weight for the objectives (the so called *mixed potentially Pareto optimal solution*, when $\lambda = 0.5$). The three well-spaced values for λ show the conflicting character of the objectives of our problem. A similar procedure was followed in de Oliveira Florentino et al. (2018) with an algorithm whose characteristics, in terms of computational requirements, are similar.

The GA's pseudocode is presented in Algorithm 4.2.

Algorithm 4.2 The GA for Problem (21)

```

1: Input: problem data,  $n$ ,  $G$ ,  $\gamma_1$ ,  $\gamma_2$ ,  $\gamma_3$ 
2: Build the initial population,  $\mathcal{P}$ .
3:  $g = 0$ 
4: while  $g \leq G$  do
5:   Evaluate  $n$  individuals of  $\mathcal{P}$  and identify  $\mathcal{E}$ 
6:   Select  $\gamma_1 \cdot n$  elements from  $\mathcal{P}$  (the intermediate population)
7:   Apply the crossover with individuals from the intermediate population, generating  $\lfloor \gamma_1 \cdot n \rfloor$  children (set  $\mathcal{C}$ )
8:    $\mathcal{P} = \mathcal{P} \cup \mathcal{C}$ 
9:   Apply the mutation operator, with probability  $\phi(g)$  on  $\lfloor \gamma_2 \cdot n \rfloor$  individuals from  $\mathcal{P}$  and update  $\mathcal{P}$ 
10:  if  $g = 0.5G \vee g = 0.7G \vee g = 0.9G$  then
11:    Apply the migration operator on  $\lfloor \gamma_3 \cdot n \rfloor$  individuals from  $\mathcal{P}$ 
12:  end if
13:  Evaluate the new individuals obtained from the crossover, mutation and migration operators
14:  Let  $\bar{\mathcal{E}}$  be the best new individual in  $\mathcal{P}$ 
15:  if  $f(\bar{\mathcal{E}}) < f(\mathcal{E})$  then
16:     $\mathcal{E} = \bar{\mathcal{E}}$ 
17:  end if
18:  Update the population with the  $n$  best elements from  $\mathcal{P} \cup \mathcal{C} \cup \{\mathcal{E}\}$ 
19:   $g = g + 1$ 
20: end while
21: Output:  $\mathcal{E}$ 

```

In the following section, we present the computational results performed to validate the mathematical models with the corresponding exact methodologies, and the heuristic solution method proposed.

5 Computational experiments

The computational experiments were implemented in the MATLAB (2017) software version 2017 in an Intel Core i7 with 8GB of RAM. For comparison purposes, the exact methods were implemented with IBM ILOG CPLEX version 12.8 (CPLEX 2017). Table 9 in the Appendix presents some real characteristics of the 27 instances generated to validate the methodologies. Such instances were semi-randomly generated by combining different values of N , K and T , many of which correspond to real situations. In the same table, we present the number of variables and constraints of the quadratic model (1)–(6) and for its linearized version (13)–(20). The performance of the CPLEX solvers running with these two formulations will be presented in the next tables. The GA parameters, presented in Table 10, were determined by preliminary computational experiments and they were the best cost-benefit choices.

The shape of the plots is illustrated in Fig. 6. Each plot is a square whose sides are 200 meters in length. Therefore, the distance between the plots 1 and 3 is 200 meters, and between plots 5 and 7 is zero, because they are adjacent. For instances with $K < 12$ plots the K first plots were considered, as shown in Fig. 6. The remaining data of the crops, such as planting cycle duration (in periods), the botanical family to which the plant belongs and profitability (per m^2) are presented in Table 11.

5.1 Results

As mentioned above, in the experiments we used three distinct values for the weight in the function z_1 : $\lambda = 1.0$, $\lambda = 0.5$ and $\lambda = 0.0$. With each of the 27 instances considered, the GA was run 20 times.

Tables 1, 2, 3, 4, 5 and 6, indicate the values of the objective functions determined for the quadratic and linear cases for each λ value, along with all the 27 instances generated (z_{CPLEX}). In addition, we provided the average, maximum and minimum values of the GA objective values in the 20 runs. The relative Gap (in %) between the objective values from CPLEX and GA, was measured by using the average values of the GA figures. Hence, this gap was calculated on the basis of the averages of the objective value obtained in GA runs (\bar{z}_{GA}) as follows:

$$\text{Gap} = \frac{\bar{z}_{GA} - z_{CPLEX}}{\bar{z}_{GA}} \times 100.$$

We imposed a time limit of 3600 seconds for CPLEX to determine the optimal solution to Problem (7) (quadratic formulation) or Problem (21) (linearized formulation). The symbol (*) indicates that CPLEX did not determine any feasible integer solution in 3600 seconds.

Table 1 displays the z_1 figures obtained with $\lambda = 1.0$. CPLEX determined a solution for only seven instances with the linearized formulation. With the quadratic formulation, the CPLEX determined a solution for 24 instances and did not find any feasible integer solution for three large instances. Note that, with the non-convex quadratic function (1), the CPLEX solver can stop in a local optimum or in the global optimum.

Table 1 Computational results of the z_1 objective function using $\lambda = 1.0$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	0.113	0.119	0.113	0.113	0.113	0.000	−4.966
2	0.249	0.286	0.256	1.119	0.163	3.018	−10.521
3	3.112	5.730	3.486	7.835	3.226	12.018	−39.162
4	0.294	0.316	0.297	0.301	0.294	1.054	−5.981
5	4.755	4.910	4.807	5.472	4.514	1.094	−2.098
6	*	17.260	10.820	16.630	9.110	*	−37.312
7	0.536	0.525	0.504	0.508	0.501	−5.989	−3.982
8	*	7.803	6.735	10.665	4.528	*	−13.692
9	*	20.350	20.750	29.320	16.890	*	1.966
10	0.079	0.105	0.040	0.050	0.020	−49.495	−61.905
11	*	0.697	0.550	1.400	0.240	*	−21.045
12	*	0.754	0.904	0.702	1.970	*	19.973
13	*	0.227	0.134	0.284	0.090	*	−40.977
14	*	0.979	0.875	1.569	0.400	*	−10.621
15	*	6.587	5.554	9.114	4.360	*	−15.685
16	*	0.783	0.175	0.217	0.135	*	−77.606
17	*	8.183	2.810	4.300	1.290	*	−65.662
18	*	19.210	17.020	21.290	13.040	*	−11.400
19	*	0.039	0.029	0.048	0.009	*	−24.870
20	*	0.517	0.449	1.314	0.130	*	−13.203
21	*	*	2.290	3.700	1.640	*	*
22	*	0.410	0.133	0.260	0.075	*	−67.537
23	*	1.868	1.054	1.700	0.618	*	−43.570
24	*	*	5.700	7.320	4.550	*	*
25	*	0.207	0.206	0.326	0.102	*	−0.435
26	*	8.930	1.710	2.290	1.000	*	−80.850
27	*	*	9.060	10.760	8.020	*	*

The GA solved all the 27 instances. For most instances, this method produced a better solution with respect to the objective z_1 as compared with CPLEX. For example, with instance 10, the GA provided a solution with a 49% and 61% lower proliferation on average, with regard to those determined by CPLEX with the linearized and the quadratic formulations, respectively.

As for the difference, for instance, among the instances and the z_1 value determined, we should highlight certain observations. In the column that presents the results from the GA, note that the differences among instances 1, 10 and 19 consist in the number of crops N , respectively, equal to 5, 10 and 20. As for the z_1 values observed, they decrease as N decreases. This is accountable as there is a greater variability among the crops which leads to more freedom in combining them in the same planting area. On the other hand, if the instances 1, 2 and 3 are compared, we see that they differ only in the number of periods, and the z_1 values increase with/by T . The same happens when K is modified. Note that in

Table 2 Computational results of the z_1 objective function using $\lambda = 0.5$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	2.373	1.518	1.990	2.520	1.896	-16.140	31.137
2	15.810	8.004	8.870	9.880	7.490	-43.896	10.820
3	12.800	8.960	11.770	11.992	11.365	-8.047	31.362
4	14.240	10.910	8.202	9.758	6.729	-42.400	-24.819
5	14.235	10.340	12.895	13.540	11.008	-9.413	24.710
6	*	17.120	22.712	23.569	21.950	*	32.664
7	24.660	18.296	12.730	14.040	11.870	-48.378	-30.422
8	*	24.119	17.900	21.140	17.170	*	-25.784
9	*	23.350	26.356	28.641	23.989	*	12.874
10	2.924	4.560	0.220	0.460	0.101	-92.477	-95.175
11	*	4.173	4.870	5.420	4.210	*	16.708
12	*	8.565	10.890	12.698	9.325	*	27.147
13	*	3.635	0.623	1.117	0.275	*	-82.853
14	*	7.083	8.300	8.660	8.215	*	17.179
15	*	28.132	23.950	33.790	21.900	*	-14.865
16	*	6.462	1.134	2.167	0.619	*	-82.453
17	*	15.290	13.080	15.170	10.440	*	-14.453
18	*	46.190	35.720	46.980	31.840	*	-22.667
19	*	2.243	0.480	0.540	0.300	*	-78.601
20	*	3.696	0.697	1.200	0.356	*	-81.140
21	*	*	5.140	8.460	3.210	*	*
22	*	2.340	0.830	1.130	0.550	*	-64.531
23	*	7.722	2.420	2.980	1.580	*	-68.659
24	*	*	11.610	14.460	9.930	*	*
25	*	3.956	1.510	2.550	0.830	*	-61.833
26	*	12.331	4.950	6.050	3.530	*	-59.857
27	*	*	19.740	24.180	16.240	*	*

instance 9, with $N = 5$ crops, $K = 12$ plots, $T = 24$ periods, the highest z_1 value was attained.

Table 2 illustrates the z_1 values with $\lambda = 0.5$. This means that the (normalized) total profit carries 50% weight in the approximation of the Pareto optimal solutions. The linear and quadratic formulations were solved once more for seven and 24 instances, respectively.

In inserting an objective that is diametrically opposed to minimization of z_1 , the values of this function increase. For example, in the case of instance 26, this increase was 38% with respect to the quadratic formulation (from 8.930 to 12.331). GA performed well considering the average of its objective values as compared to those determined from the linear formulation. For 15 out of the 23 instances solved by CPLEX with the quadratic formulation, the gap was also negative. This means that the GA can generate more promising solutions for this problem with larger dimension instances, where the exact methods fail.

Table 3 Computational results of the objective function z_1 using $\lambda = 0.0$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	9.962	8.710	9.962	9.962	9.962	0.000	14.369
2	19.920	20.000	19.920	19.820	19.920	0.000	-0.400
3	39.840	39.840	39.840	39.840	39.840	0.000	0.000
4	17.220	17.220	17.220	17.220	17.220	0.000	0.000
5	34.440	34.440	34.440	34.440	34.440	0.000	0.000
6	*	69.060	68.880	68.880	68.880	*	-0.261
7	24.560	24.615	24.590	24.630	25.560	0.122	-0.102
8	*	49.244	49.130	49.130	49.130	*	-0.232
9	*	98.530	98.310	98.380	98.270	*	-0.223
10	8.562	8.480	9.960	9.960	9.960	16.328	17.453
11	*	19.125	19.920	19.920	19.920	*	4.157
12	*	38.426	39.840	39.840	39.840	*	3.680
13	*	18.095	17.220	17.260	17.220	*	-4.836
14	*	36.874	34.440	34.440	34.440	*	-6.601
15	*	62.382	68.900	69.060	68.880	*	10.449
16	*	21.720	24.593	24.629	24.569	*	13.227
17	*	43.440	49.150	49.190	49.130	*	13.145
18	*	96.880	98.340	98.510	98.270	*	1.507
19	*	8.562	9.150	11.450	6.880	*	6.868
20	*	19.125	19.920	19.920	19.920	*	4.157
21	*	38.249	39.840	39.840	39.840	*	4.160
22	*	11.895	14.020	18.570	9.720	*	17.865
23	*	36.126	34.440	34.440	34.440	*	-4.667
24	*	*	68.880	68.880	68.880	*	*
25	*	31.720	20.550	25.340	16.460	*	-35.214
26	*	43.440	49.140	49.190	49.130	*	13.122
27	*	*	98.320	98.450	98.270	*	*

Table 3 indicates the z_1 values with $\lambda = 0.0$, that is, only the total profit was optimized. In this way, pest dissemination in those solutions is expected to be greater than in the previous cases. When considering the solutions of the instance 10 produced by GA, the possibility of pest dissemination leaps from 0.04 ($\lambda = 1.0$) to 0.22 ($\lambda = 0.5$) and, ultimately to 9.96 ($\lambda = 0.0$), on average. The same instances that were not solved with $\lambda = 1.0$ and $\lambda = 0.5$ by the CPLEX, when considering the linear formulation, are not solved with the aim of optimizing z_2 , $\lambda = 0.0$. The main justification factor, characteristic of these instances, is the dimension. As for the quadratic formulation, the CPLEX solved 25 instances within the processing time limit. However, the GA did not achieve the same performance in this objective when compared to the results obtained with a greater λ . Our metaheuristic was worse in two of the seven instances solved by CPLEX using the linear formulation, and in 13 out of 25 solved with the quadratic formulation.

Table 4 presents the $z_2 \times 10^{-3}$ values obtained with $\lambda = 1.0$. Here the profit was not optimized. The GA was worse for all instances solved by CPLEX, using the linear formulation,

Table 4 Computational results of the objective function $z_2 \times 10^{-3}$ using $\lambda = 1.0$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	1.193	1.193	1.193	1.193	1.193	0.000	0.000
2	2.360	2.478	2.027	2.120	1.964	14.131	18.220
3	4.544	5.020	4.519	4.805	4.216	0.557	9.986
4	1.900	1.788	1.789	1.789	1.788	5.853	-0.045
5	3.472	3.779	3.179	3.635	2.956	8.439	15.877
6	*	7.574	6.510	7.160	6.130	*	14.048
7	2.385	2.385	2.385	2.385	2.385	0.000	0.000
8	*	5.005	4.148	4.352	3.938	*	17.126
9	*	9.970	8.894	9.589	8.395	*	10.788
10	1.291	1.252	0.770	0.777	0.760	40.352	38.489
11	*	2.621	2.020	2.040	2.000	*	22.930
12	*	4.850	4.300	4.390	4.210	*	11.348
13	*	1.872	1.162	1.194	1.140	*	37.931
14	*	3.580	3.030	3.050	3.010	*	15.363
15	*	7.717	6.410	6.540	6.290	*	16.937
16	*	2.576	1.562	1.590	1.535	*	39.383
17	*	6.077	4.060	4.120	4.030	*	33.190
18	*	11.490	8.620	8.710	8.540	*	24.978
19	*	1.217	0.730	0.780	0.760	*	40.021
20	*	2.510	1.668	1.761	1.608	*	33.546
21	*	*	3.880	4.020	3.730	*	*
22	*	1.903	1.160	1.170	1.140	*	39.056
23	*	3.798	2.530	2.630	2.480	*	33.390
24	*	*	5.843	5.963	5.701	*	*
25	*	2.488	1.550	1.580	1.530	*	37.708
26	*	6.298	3.440	3.580	3.230	*	45.379
27	*	*	7.910	8.120	7.710	*	*

and overcame CPLEX only once, for an instance with a quadratic formulation. As can be observed, on average, the z_2 value obtained by the CPLEX is higher than the one determined by GA.

Table 5 shows the $z_2 \times 10^{-3}$ values for $\lambda = 0.5$. As expected, a relative increase is noted in this function, compared with the results shown in Table 4. The GA was able to solve all the instances but its solutions continue to be worse with regard to z_2 , when compared with the solutions from CPLEX.

Table 6 displays the results for the $z_2 \times 10^{-3}$ values with $\lambda = 0.0$. CPLEX solved seven and 25 instances using the quadratic and linear formulations respectively. Regardless of the objective optimized, the dimension of the problem has a preponderant role. We also note that GA, in the 20 runs, generated solutions with little or no significant difference as to the minimum and maximum values. The potentially Pareto optimal solutions determined to assure maximum profit use an alternation of only a few more profitable crops, thus resulting

Table 5 Computational results of the objective function $z_2 \times 10^{-3}$ using $\lambda = 0.5$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	1.480	1.518	1.462	1.478	1.455	1.216	3.657
2	3.250	2.783	2.760	2.840	2.740	15.077	0.826
3	6.348	5.250	5.690	5.690	5.690	10.365	-8.381
4	2.410	2.356	2.189	2.200	2.176	9.170	7.088
5	5.038	3.987	4.190	4.320	4.160	16.824	-5.092
6	*	7.683	8.450	8.490	8.410	*	-9.983
7	3.420	3.195	2.910	2.930	2.900	14.912	8.926
8	*	5.503	5.560	5.580	5.490	*	-1.032
9	*	10.760	11.413	11.518	11.298	*	-6.069
10	2.083	2.370	1.280	1.340	1.260	38.547	45.992
11	*	3.772	2.790	2.810	2.770	*	26.026
12	*	7.502	5.710	5.780	5.670	*	23.883
13	*	2.769	1.926	1.984	1.901	*	30.444
14	*	5.610	4.170	4.230	4.150	*	25.668
15	*	13.643	8.500	8.530	8.480	*	37.697
16	*	3.958	2.573	5.592	2.553	*	34.985
17	*	8.236	5.580	5.660	5.550	*	32.252
18	*	18.790	11.380	11.450	11.350	*	39.436
19	*	1.994	1.380	1.390	1.370	*	30.789
20	*	3.680	2.640	2.650	2.610	*	28.261
21	*	*	5.540	5.590	5.510	*	*
22	*	2.616	2.040	2.060	2.030	*	22.018
23	*	5.825	3.980	4.110	3.940	*	31.670
24	*	*	8.310	8.400	8.260	*	*
25	*	3.560	2.720	2.750	2.710	*	23.598
26	*	7.729	5.320	5.390	5.280	*	31.168
27	*	*	11.140	11.220	11.080	*	*

in high pest infestation. From a heuristic perspective, the composition of a solution involving a high profit becomes difficult to obtain and computationally expensive. In that case it is more convenient to determine these solutions by employing the exact methods, in cases when they are successful.

Table 7 shows the CPU computational times (in seconds) of each method, for all instances and for the three λ values. CPLEX, when applied with the linear formulation and the objective of minimizing pest dissemination, took one hour for five of the seven instances. In addition, these are feasible or local optima solutions, because CPLEX finalized the search before attaining a stopping criterium different from the time limit. When applied to the quadratic formulation, we have a shorter CPU time, despite having determined only local optimum solutions (known by comparing quadratic CPLEX with GA z values in Table 4). Instances 23 and 26 were the only ones for which CPLEX used the maximum allowed processing time. When λ is equal to 0.5, the CPLEX with the quadratic formulation found local optimal

Table 6 Computational results of the objective function $z_2 \times 10^{-3}$ using $\lambda = 0.0$

Instance	CPLEX		GA			Gap to linear (%)	Gap to quadratic (%)
	Linear	Quadratic	Avg.	Max.	Min.		
1	1.710	1.710	1.710	1.710	1.710	0.000	0.000
2	3.420	3.420	3.420	3.420	3.420	0.000	0.000
3	6.840	6.840	6.840	6.840	6.840	0.000	0.000
4	2.560	2.565	2.560	2.560	2.560	0.000	0.002
5	5.134	5.130	5.130	5.130	5.130	0.078	0.000
6	*	10.260	10.260	10.260	10.260	*	0.000
7	3.420	3.420	3.420	3.420	3.420	0.000	0.000
8	*	6.840	6.840	6.840	6.840	*	0.000
9	*	13.800	13.680	13.680	13.680	*	0.870
10	2.370	2.654	1.760	1.760	1.760	25.738	33.685
11	*	4.740	3.520	3.520	3.520	*	25.738
12	*	9.874	7.040	7.040	7.040	*	28.702
13	*	3.654	2.640	2.640	2.640	*	27.750
14	*	7.002	5.280	5.280	5.280	*	24.593
15	*	14.220	10.560	10.560	10.560	*	25.738
16	*	4.540	3.520	3.520	3.520	*	22.467
17	*	9.380	7.040	7.040	7.040	*	24.947
18	*	18.960	14.080	14.080	14.080	*	25.738
19	*	2.544	1.970	2.010	1.900	*	22.554
20	*	5.087	3.560	3.560	3.560	*	30.018
21	*	8.144	7.120	7.120	7.120	*	14.380
22	*	3.816	2.900	3.010	1.840	*	23.994
23	*	7.632	5.340	5.340	5.340	*	30.023
24	*	*	10.680	10.680	10.680	*	*
25	*	5.087	3.870	9.980	3.750	*	23.930
26	*	10.175	7.120	7.120	7.120	*	30.025
27	*	*	14.240	14.240	14.240	*	*

solutions in a relatively low computational time, when compared to the linearized version (with two global optima corresponding to instances 1 and 4). In the case of $\lambda = 0.0$, this difference in processing time between the linear and the quadratic version is more noticeable. Therefore, the computational difficulty in determining Pareto optimal solutions by the exact methodologies for this problem depends strongly on the weight of the objective functions involved.

As for the GA algorithm, we did not note any interference in its performance with λ —only when N , K or T change. The CPU time is practically proportional to these three parameters. A comparative analysis between CPLEX and GA, shows that the exact methods surpass the metaheuristic with the z_2 objective but lose their efficacy when λ increases. For large instances, GA found potentially Pareto optimal solutions about ten times faster, on average, than CPLEX.

Table 7 Computational CPU time (in seconds) of the CPLEX and GA (on average) to solve each instance

Instance	Linear formulation			Quadratic formulation			Genetic algorithm		
	$\lambda = 0.0$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 0.0$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 0$	$\lambda = 0.5$	$\lambda = 1.0$
1	0.7	106.0	345.9	0.1	12.0	13.0	27.2	18.7	26.4
2	1.9	104.0	3600.0	0.2	35.0	43.0	45.0	31.0	44.0
3	10.0	112.0	3600.0	16.0	107.0	124.0	78.0	53.0	76.0
4	4.8	465.0	899.2	17.6	28.8	32.9	40.0	30.0	42.0
5	23.0	107.0	3600.0	43.9	94.4	118.8	71.0	49.0	67.8
6	*	*	*	252.0	434.0	481.0	180.0	118.0	168.0
7	62.0	156.0	3600.0	10.7	30.7	28.4	67.0	46.0	59.0
8	*	*	*	114.5	250.4	301.7	100.0	71.0	97.0
9	*	*	*	711.0	1630.0	1103.0	354.0	210.0	325.0
10	6.0	97.0	3600.0	0.4	30.0	31.0	28.0	18.0	26.0
11	*	*	*	7.0	412.0	408.0	44.0	31.0	43.0
12	*	*	*	12.0	862.0	808.0	78.0	55.0	79.0
13	*	*	*	6.0	313.0	326.0	43.0	30.0	38.0
14	*	*	*	18.0	1037.0	984.0	93.0	50.0	89.0
15	*	*	*	29.3	987.0	1030.0	163.0	104.0	160.0
16	*	*	*	26.5	753.0	788.0	91.6	59.2	84.7
17	*	*	*	62.2	1100.0	1070.0	220.0	141.0	210.0
18	*	*	*	209.0	1880.0	1848.0	641.0	421.0	628.0
19	*	*	*	2.6	350.0	346.0	39.3	24.0	35.0
20	*	*	*	14.0	1028.0	1017.0	69.0	44.0	67.0
21	*	*	*	1094.0	*	*	189.0	116.0	187.0
22	*	*	*	16.0	982.0	1008.0	65.0	40.0	55.0
23	*	*	*	228.0	3600.0	3600.0	134.0	85.0	129.0
24	*	*	*	*	*	*	397.0	247.0	391.0
25	*	*	*	135.0	1437.0	1477.0	98.0	61.0	84.0
26	*	*	*	1405.0	3600.0	3600.0	105.0	70.0	98.0
27	*	*	*	*	*	*	347.0	222.0	335.0

Table 8 shows the average total number of different crops used in the planting schedule throughout the planning horizon. For $\lambda = 0.0$, the exact methods determined solutions with the two most profitable crops, alternating them with each other (see figures in this table). When increasing λ , there is a greater opportunity to insert different crops to minimize pest infestation, thus enriching the variability of species in the schedule. The GA algorithm determines a planting calendar with a greater crop variability. This justifies why this method determines solutions with less likelihood of pest proliferation compared with CPLEX. For example, the instance 26 has $N = 20$ available crops, from which only six were used in the solution provided by CPLEX while the GA adopted 17.40 different crops, on average, for solutions to this instance.

In the next subsection, we analyze the characteristics of the typical solutions from CPLEX and GA in function of λ , taking an example from instance 23.

Table 8 Number of different crops used in the planting schedule (on average)

Class	Exact (linear)			Exact (quadratic)			Genetic algorithm		
	$\lambda = 0.0$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 0.0$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 0.0$	$\lambda = 0.5$	$\lambda = 1.0$
1	2.00	4.00	3.00	2.00	3.00	3.00	2.00	4.00	3.00
2	2.00	4.00	3.00	2.00	4.00	3.00	2.00	4.20	4.40
3	2.00	5.00	5.00	2.00	5.00	5.00	2.00	3.00	4.20
4	2.00	4.00	4.00	2.00	2.00	3.00	2.00	4.00	3.00
5	2.00	3.00	5.00	2.00	4.00	3.00	2.00	3.90	4.60
6	*	*	*	2.00	4.00	5.00	2.00	3.50	4.90
7	2.00	2.00	5.00	2.00	2.00	3.00	2.00	4.30	3.00
8	*	*	*	2.00	4.00	4.00	2.00	3.00	4.80
9	*	*	*	2.00	2.00	5.00	2.00	3.60	5.00
10	2.00	5.00	7.00	2.00	2.00	3.00	2.00	6.10	2.80
11	*	*	*	2.00	3.00	5.00	2.00	8.80	8.60
12	*	*	*	2.00	4.00	6.00	2.00	9.70	9.80
13	*	*	*	2.00	3.00	3.00	2.00	7.70	2.90
14	*	*	*	2.00	4.00	4.00	2.00	8.90	9.00
15	*	*	*	2.00	3.00	6.00	2.00	9.70	10.00
16	*	*	*	2.00	3.00	3.00	2.00	8.20	2.00
17	*	*	*	2.00	3.00	5.00	2.00	9.20	9.30
18	*	*	*	2.00	2.00	4.00	2.00	9.80	10.00
19	*	*	*	2.00	3.17	3.00	2.00	10.90	6.00
20	*	*	*	2.00	5.00	6.00	2.00	7.60	10.80
21	*	*	*	2.00	*	*	2.00	18.40	18.30
22	*	*	*	2.00	4.00	5.00	2.00	13.30	7.70
23	*	*	*	2.00	5.00	5.00	2.00	15.20	13.60
24	*	*	*	*	*	*	2.00	19.20	19.40
25	*	*	*	2.00	3.00	3.00	2.00	14.80	9.20
26	*	*	*	2.00	6.00	7.00	2.00	17.40	15.30
27	*	*	*	*	*	*	2.00	19.90	20.00

5.2 Analysis of different potentially Pareto optimal solutions

In Figs. 2, 3 and 4 we present potentially Pareto optimal solutions for this problem obtained from CPLEX and GA. The crop data are from instance 23. The numbers presented in these mosaics are the indexes of the crops scheduled to be planted (see Table 11). For example, in Fig. 2a, crop 2 whose duration C_2 is 3 periods, is planted in plot 1 in periods 2 to 4.

Figure 2 presents two typical solutions when trying to minimize z_1 . Figure 2a shows a typical solution from GA, whose associated objective vector is $(z_1, z_2) = (1.1674, 2476.40)^T$. Fig. 2b is the solution from CPLEX (obtained from the quadratic formulation with 4000 seconds of running time) whose objective vector is $(z_1, z_2) = (1.845, 3654.50)^T$. The solution from GA compared with the solution from CPLEX is 58% better with respect to z_1 but 47% worse with respect to z_2 .

These solutions have a different structure. Note that the GA solution used only one crop of 1 period 's duration (crop 20), while the CPLEX solution used crops 3, 10, 16, 20, all

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1	19		2			13			9			19
2		17		14				19		9		
3		8		2			14			9		8
4			13		6		14				2	
5		1		7			19			9		20
6	7	15			19			14				7
7	13		7			9			19			13
8	2		14				7			17		2
9		14			9			7			17	14

(a) Solution provided by GA

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1	5	3				12		17		3	15	5
2		13	16		12		6		1		6	13
3	10		13		16		13		10			13
4		13	16		13		10	15		4		13
5	10		19		3		19				14	
6	15	10		7		20		13				12
7	10		13		20		13		10			13
8	1	3		13		16		4				6
9	4		13		10		7			19		4

(b) Solution provided by CPLEX

Fig. 2 Solutions obtained by CPLEX and by GA using $\lambda = 1.0$

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1	1	3	1	3	1	3	1	3	1	3	1	3
2	1	3	1	3	1	3	1	3	1	3	1	3
3	3	1	3	1	3	1	3	1	3	1	3	1
4	1	3	1	3	1	3	1	3	1	3	1	3
5	3	1	3	1	3	1	3	1	3	1	3	1
6	1	1	3	1	3	1	3	1	3	1	3	1
7	3	1	3	1	3	1	3	1	3	1	3	1
8	1	3	1	3	1	3	1	3	1	3	1	3
9	1	3	1	3	1	3	1	3	1	3	1	3

(a) Solution provided by GA

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1	3	10	3	10	3	10	3	10	3	10	3	10
2	3	10	3	10	3	10	3	10	3	10	3	10
3	3	10	3	10	3	10	3	10	3	10	3	10
4	10	3	10	3	10	3	10	3	10	3	10	3
5	3	10	3	10	3	10	3	10	3	10	3	10
6	3	10	3	10	3	10	3	10	3	10	3	10
7	3	10	3	10	3	10	3	10	3	10	3	10
8	3	10	3	10	3	10	3	10	3	10	3	10
9	3	10	3	10	3	10	3	10	3	10	3	10

(b) Solution provided by CPLEX

Fig. 3 Solutions obtained by CPLEX and GA using $\lambda = 0.0$

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1		7		3		11		8		20		1
2		15		8		5		10		3		18
3	3		1	3		1		3		1		3
4	1	3	1	3	1	3	1	3	1	3	1	3
5		13		2		16		20		18		8
6			1	12		16		9			4	8
7	17		1		11	16	3	12	20	10	17	
8	1		13		20		9		17		3	1
9	3		1	3		1		3		1		3

(a) Solution provided by GA

Plot \ Per.	1	2	3	4	5	6	7	8	9	10	11	12
1	10	3	10	3	10	3	10	3	10	3	10	3
2	16	10	16	10	16	10	16	10	16	10	16	10
3	10	16	3	16	3	16	10	3	10	3	10	3
4	3	10	16	10	16	10	16	10	16	10	3	10
5	10	3	10	16	10	16	10	3	10	3	10	3
6	16	10	16	3	16	3	16	10	3	10	16	10
7	10	16	10	16	3	16	10	3	10	16	10	16
8	3	10	3	10	16	3	16	10	3	10	16	10
9	10	3	10	16	3	16	3	16	10	3	16	3

(b) Solution provided by CPLEX

Fig. 4 Solutions obtained by CPLEX and GA using $\lambda = 0.5$

involving 7 periods of duration. Crop 14 is the one that has the longest duration period among the 20 crops and was planted six times and once, respectively, in the solutions from GA and CPLEX. To minimize z_1 , GA uses the strategy of planting longer duration crops, while CPLEX uses shorter period ones.

Figure 3 presents the solutions using $\lambda = 0.0$. Figure 3a shows a typical solution provided by the GA, with an objective vector of $(z_1, z_2) = (34.44, 5340.40)^T$. Figure 3b shows a solution obtained by CPLEX, whose objective vector is $(z_1, z_2) = (36.12, 7631.50)^T$. An analysis and comparison of these solutions led us to observe that the CPLEX solution is 43% better with respect to objective z_2 , but is 4.8% worse with respect to objective z_1 . It is interesting to note that both solutions use two crops alternately in all plots. However, the GA opts to choose crops 1 and 3 (which provide greater profit and are not from the same family) while the CPLEX determined a solution that uses crops 3 and 10. Although crop 1 ensures a greater profit, its duration is longer when compared to crop 10. CPLEX seeks to plant crops that have a higher per period profit, while GA tries to use the crops with a higher total profit. In considering the CPLEX solutions obtained from two different cultivating strategies (with $\lambda = 1.0$ and $\lambda = 0.0$) we notice that the profit doubles, but pest proliferation increases almost 20 times.

Figure 4 shows typical solutions with $\lambda = 0.5$. Figures 4a, b are associated with the solutions of GA and CPLEX in which the associated objective vectors are $(z_1, z_2) =$

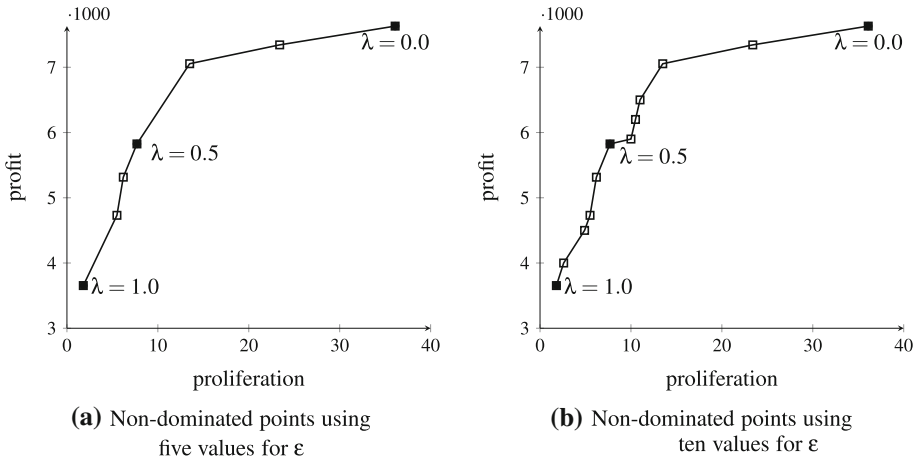


Fig. 5 Other non-dominated points determined for instance 23

$(2.534, 4102.00)^T$ and $(z_1, z_2) = (7.722, 5825.00)^T$, respectively. GA produces a solution which is 67% with respect to objective z_1 , but 42% worse compared with objective z_2 . A remarkable characteristic of the CPLEX solution is the fact that it only uses alternatively crops 3, 10 and 16. GA mixes 16 different types of crops, and this explains the smaller value of the solution from GA for pest proliferation.

In CPLEX solutions, the number of equal crops per period is considerable. For example, in period 1, crop 10 is cultivated in five different plots, while crops 3 and 16 are cultivated in two plots each. Something different occurs with the GA solution. Notably, we have a greater mixture of different crops. Moreover, in analyzing the CPLEX solutions with $\lambda = 1.0$ and $\lambda = 0.5$, we note that, in the second case the profit is 60% greater than with $\lambda = 1.0$. However, the value for pest proliferation quadruples.

It is important to highlight that it would be possible to obtain other Pareto optimal solutions, for instance, using multi-objective exact methods of scalarization. A simple way consists in assigning other weights to the objective functions. An alternative method consists in using the ϵ -constraint method.

To apply the ϵ -constraint method, firstly, we find the lexicographic solutions, thus determining the range of z_2 function in the objective space. Let $z_2 \in [z_2^-, z_2^+]$, where z_2^- and z_2^+ are, respectively, the minimum and maximum profit determined. The next step is to solve the problem of minimizing the possibility of pest proliferation, by imposing a minimal profit of ϵ in the range $[z_2^-, z_2^+]$. When we solve this problem, a Pareto optimal solution can be determined along with the respective non-dominated point (in the objective space). The ϵ values can be attributions of the agricultural manager and decision-maker. Figure 5 shows some additional aspects of the Pareto frontier of this problem, by considering instance 23 and performing five and 10 assignments for ϵ . In these experiments, the ϵ values were equally distributed in the range $[z_2^-, z_2^+] = [3654.00, 7631.50]$, while bearing in mind the quadratic formulation solved by CPLEX. We emphasize in black the non-dominated points already determined with $\lambda = 0.0, 0.5$ and 1.0 . The number of Pareto optimal solutions provided depends on the data of the instance and the CPU time made available by the manager.

These experiments clearly demonstrate that the methodologies, both the heuristic and the exact ones, employed to solve the problem of sustainable cultivation are powerful tools to assist decision-making addressing complex problems such as this one.

6 Conclusions

This work proposes a bi-objective quadratic mathematical programming model to schedule crop cultivation, using a crop rotation approach and constraints to sustainable cultivation. Methods to solve the problem and assist agricultural managers in their decision-making are presented and implemented. The mathematical models with binary variables (the quadratic and linearized version) determine a calendar of crop cultivation in each plot and within a pre-specified planning horizon. Two conflicting objectives are taken into account: (1) minimize the possibility of pest proliferation among the crops and (2) maximize the total profit of the planting schedule. Therefore, these objectives can be balanced, to produce different Pareto optimal solutions that establish a compromise between these two objectives. A multi-objective analysis enables several scenarios of cultivation schedules and safer decision-making.

The paper presents exact techniques of mathematical programming (based on the quadratic model and on the linearized version) to determine Pareto (local) optimal solutions for this problem. We illustrate, through computational tests with semi-random instances of real dimensions, that the exact methods have a limited application. Out of the 27 instances considered, CPLEX solved only seven instances when considering the linearized version, and 24 with the quadratic one, by imposing a maximum limit on the computation time (one hour at the most, as proposed by the agricultural managers). In addition, since the quadratic objective function is not convex, CPLEX may end its search in a local optimum, thus providing a Pareto sub-optimal solution.

To overcome these unfavorable aspects and solve large dimension instances, this work proposes a genetic algorithm. Two constructive heuristics, together with this algorithm are also presented. The experiments showed that this metaheuristic can determine good quality potentially Pareto optimal solutions at a low computational time. Tests with the largest dimension instances (where the exact methods failed to find even a feasible solution) were solved by this algorithm in a computational time below 2 minutes per instance. The results showed that GA determined solutions with a value as to possible pest dissemination 74% lower than the value provided by CPLEX, besides using, on average, one-fifth of the computational time.

In short, the bi-objective methodologies presented aim to establish strategies for different sustainable vegetable production. They preserve a compromise with the economic factor and are therefore applicable in real situations. Consequently, the methodologies can be used to assist the decision-makers in choosing different alternatives of agricultural production for their farms. Finally, this study contributes to the advance of knowledge in the field of Operational Research in a sustainable environment.

Acknowledgements The authors thank the Brazilian institutions FAPESP (Projects n. 2014/01604-0 and 2014/04353-8), CNPq (n.302454/2016-0), FAPESP (n. 2013/07375-0). In addition, we thank the Federal Technological University of Paraná for its support to this research and the translation services provided. The research performed by Margarida Vaz Pato was supported by funding from Fundação para a Ciência e a Tecnologia, Portugal, under projects UID/MAT/04561/2013 and UID/Multi/00491/2013 and by the Research Fund of ISEG. Helenice de Oliveira Florentino was supported by funding from Conselho Nacional de Desenvolvimento Científico e Tecnológico (303267/2011-9).

Appendix

Table 9 presents the characteristics of all instances used.

Table 10 presents the values of the parameters used by the proposed GA to solve the problem.

Table 9 Characteristics of the instances: values for the parameters, number of variables and constraints for models (1)–(6) and (13)–(20)

Instance	Parameters			Linear formulation		Quadratic formulation	
	<i>N</i>	<i>K</i>	<i>T</i>	# Variables	# Constraints	# Variables	# Constraints
1	5	6	6	2430	2400	180	150
2	5	6	12	4860	4794	360	294
3	5	6	24	9720	9582	720	582
4	5	9	6	5670	3825	270	225
5	5	9	12	11,340	7641	540	441
6	5	9	24	22,680	15,273	1080	873
7	5	12	6	10,260	5250	360	300
8	5	12	12	20,520	10,488	720	588
9	5	12	24	41,040	20,964	1440	1164
10	10	6	6	9360	9222	360	222
11	10	6	12	18,720	18,438	720	438
12	10	6	24	37,440	36,870	1440	870
13	10	9	6	22,140	14,733	540	333
14	10	9	12	44,280	29,457	1080	657
15	10	9	24	88,560	58,905	2160	1305
16	10	12	6	40,320	20,244	720	444
17	10	12	12	80,640	40,476	1440	876
18	10	12	24	167,280	80,940	2880	1740
19	20	6	6	36,720	36,402	720	402
20	20	6	12	73,440	72,768	1440	798
21	20	6	24	146,880	145,590	2880	1590
22	20	9	6	87,480	58,203	1080	603
23	20	9	12	174,960	116,397	2160	1197
24	20	9	24	349,920	232,785	4320	2385
25	20	12	6	159,840	80,004	1440	804
26	20	12	12	319,680	159,996	2880	1596
27	20	12	24	639,360	319,980	5760	3180

Table 10 Parameters used for the implementation of the GA for all instances

<i>G</i>	<i>n</i>	γ_1	γ_2	γ_3
300	300	0.8	0.05	0.1

The geometry and layout of plots to create the 27 instances are shown in Fig. 6. The area of each plot was fixed in 4 hectares. In Fig. 6 we illustrate a plantation area divided into 12 plots.

Table 11 presents data for 20 different crops, specifically for duration of the crop cultivation, botanical family that the plant belongs to, and profitability per hectare.

The area of each plot *j* was fixed equal to $area_j = 4$ hectares for all *j*.

Fig. 6 Illustration of the disposition and dimension of the plots considered for the instances

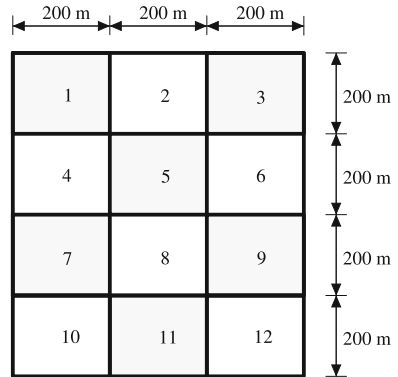


Table 11 Crop data: duration of the crop cultivation, family they belong to and profitability per hectare

Crop	C_i	F_i	$l_i \times \text{ha}^{-1}$
1	2	1	10.0000
2	3	1	9.8947
3	1	2	9.7895
4	2	2	9.6842
5	2	3	9.5789
6	2	3	9.4737
7	3	4	9.3684
8	2	4	9.2632
9	3	5	9.1579
10	1	5	9.0526
11	2	6	8.9474
12	2	6	8.8421
13	3	7	8.7368
14	4	7	8.6316
15	2	8	8.5263
16	1	8	8.4211
17	2	9	8.3158
18	2	9	8.2105
19	3	10	8.1053
20	1	10	8.0000

References

Alfandari, L., Lemalade, J.-L., Nagih, A., & Plateau, G. (2011). A mip flow model for crop-rotation planning in a context of forest sustainable development. *Annals of Operations Research*, 190(1), 149–164.

Aliano Filho, A., de Oliveira Florentino, H., & Vaz Pato, M. (2014). Metaheuristics for a crop rotation problem. *International Journal of Metaheuristics*, 3(3), 199–222.

Annetts, J., & Audsley, E. (2002). Multiple objective linear programming for environmental farm planning. *Journal of the Operational Research Society*, 53(9), 933–943.

Boyabath, O., Nasiry, J., & Zhou, Y. (2019). Crop planning in sustainable agriculture: Dynamic farmland allocation in the presence of crop rotation benefits. *Management Science*, 65(5), 2060–2076.

Costa, A. M., dos Santos, L. M. R., Alem, D. J., & Santos, R. H. (2014). Sustainable vegetable crop supply problem with perishable stocks. *Annals of Operations Research*, 219(1), 265–283.

- CPLEX. (2017). IBM ILOG CPLEX 12.8 Optimizer.
- de Oliveira Florentino, H., Irawan, C., Aliano, A. F., Jones, D. F., Cantane, D. R., Nervis, J. J., et al. (2018). A multiple objective methodology for sugarcane harvest management with varying maturation periods. *Annals of Operations Research*, 267(1–2), 153–177.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.*, Wiley-Interscience series in systems and optimization Chichester: Wiley.
- Dogliotti, S., Van Ittersum, M., & Rossing, W. (2005). A method for exploring sustainable development options at farm scale: A case study for vegetable farms in south Uruguay. *Agricultural Systems*, 86(1), 29–51.
- Dury, J., Schaller, N., Garcia, F., Reynaud, A., & Bergez, J. E. (2012). Models to support cropping plan and crop rotation decisions. A review. *Agronomy for Sustainable Development*, 32(2), 567–580.
- Han, K.-H., & Kim, J.-H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In *Proceedings of the 2000 Congress on Evolutionary Computation, 2000* (Vol. 2, pp. 1354–1360) IEEE.
- Havlin, J., Kissel, D., Maddux, L., Claassen, M., & Long, J. (1990). Crop rotation and tillage effects on soil organic carbon and nitrogen. *Soil Science Society of America Journal*, 54(2), 448–452.
- Hayashi, K. (2000). Multicriteria analysis for agricultural resource management: A critical survey and future perspectives. *European Journal of Operational Research*, 122(2), 486–500.
- Huang, I. B., Keisler, J., & Linkov, I. (2011). Multi-criteria decision analysis in environmental sciences: Ten years of applications and trends. *Science of The Total Environment*, 409(19), 3578–3594.
- Kumar, A., Sah, B., Singh, A. R., Deng, Y., He, X., Kumar, P., et al. (2017). A review of multi criteria decision making (MCDM) towards sustainable renewable energy development. *Renewable and Sustainable Energy Reviews*, 69, 596–609.
- MATLAB. (2017). *version 7.10.0 (R2017a)*. The MathWorks Inc., Natick, Massachusetts.
- Miettinen, K. (2012). *Nonlinear multiobjective optimization* (Vol. 12). New York: Springer.
- Nechi, S., Aouni, B., & Mrabet, Z. (2019). Managing sustainable development through goal programming model and satisfaction functions. *Annals of Operations Research*, 1–20.
- Parra, J. R. P. (2002). *Controle Biológico no Brasil: Parasitóides e Predadores*. São Paulo: Editora Manole Ltda.
- Rossing, W., Hammer, G., et al. (2006). Exploring profit-sustainability trade-offs in cropping systems using evolutionary algorithms. *Environmental Modelling & Software*, 21(9), 1368–1374.
- Santos, L. M., Costa, A. M., Arenales, M. N., & Santos, R. H. S. (2010). Sustainable vegetable crop supply problem. *European Journal of Operational Research*, 204(3), 639–647.
- Santos, L. M., Michelon, P., Arenales, M. N., & Santos, R. H. S. (2011). Crop rotation scheduling with adjacency constraints. *Annals of Operations Research*, 190(1), 165–180.
- Santos, L. M., Munari, P., Costa, A. M., & Santos, R. H. (2015a). A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes. *European Journal of Operational Research*, 245(2), 581–590.
- Santos, L. M., Munari, P., Costa, A. M., & Santos, R. H. (2015b). A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes. *European Journal of Operational Research*, 245(2), 581–590.
- Sarker, R., & Ray, T. (2009). An improved evolutionary algorithm for solving multi-objective crop planning models. *Computers and Electronics in Agriculture*, 68(2), 191–199.
- Snapp, S., Swinton, S., Labarta, R., Mutch, D., Black, J., Leep, R., et al. (2005). Evaluating cover crops for benefits, costs and performance within cropping system niches. *Agronomy Journal*, 97(1), 322–332.
- Tilman, D., Cassman, K. G., Matson, P. A., Naylor, R., & Polasky, S. (2002). Agricultural sustainability and intensive production practices. *Nature*, 418(6898), 671.
- West, T. O., & Post, W. M. (2002). Soil organic carbon sequestration rates by tillage and crop rotation. *Soil Science Society of America Journal*, 66(6), 1930–1946.