Computers and Electronics in Agriculture 224 (2024) 109162

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Crop planting layout optimization in sustainable agriculture: A constraint programming approach

Tommaso Adamo [a], Lucio Colizzi [b,*], Giovanni Dimauro [b], Emanuela Guerriero [a], Deborah Pareo [a]

[a] Dipartimento di Ingegneria dell'Innovazione- University of Salento, Via Monteroni, Lecce 73100, Italy
[b] Dipartimento di Informatica - University of Bari Aldo Moro, Via E. Orabona, 4, Bari 70125, Italy

ARTICLE INFO

Keywords:
Constraint programming
Optimization crop planting layout
AI planning
Smart Agriculture
Intercropping systems

ABSTRACT

In sustainable agriculture, intercropping systems represent a valuable approach. These systems involve placing mutually beneficial plant types in close proximity to each other, with the goal of exploiting biodiversity to reduce pesticide and water usage, as well as improve soil nutrient utilization. Despite its potential, the optimization of intercropping systems has received limited attention in previous studies. One of the first steps in the design of an intercropping system is the solution of the crop planting layout problem, which involves meeting crop demand while maximizing positive interactions between adjacent plants. We perform a complexity analysis of this problem and solve it through constraint programming, an artificial intelligence technique, which relies on automated reasoning, constraint propagation and search heuristics. To this aim, we present two constraint programming models based on integer variables and interval variables, respectively. Through a computational study on real-life instances, we examine the impact of different modelling approaches on the difficulty of solving the crop planting layout problem with standard constraint programming solvers. This research work has also provided the groundwork for a sowing robotic arm (under development), aiming to automate intercropping systems and assist farm workers.

1. Introduction

Food security has been achieved in the 20th century thanks to industrial agriculture, characterized by just a few profitable crops, with the nutrient cycles externally regulated.

Nowadays, there is wide consensus about the enormous environmental footprint and the low climate resilience of intensive monoculture, which outweigh its positive aspects in the long run. In monoculture farming, the field is managed as a single unit. This allows for the execution of cultivation tasks with efficient large-scale machinery. However, since a large extent of genetically uniform crops is rare in nature, farmers need to maintain stable yields through the use of synthetic fertilizers and crop protection products, such as pesticides. It is now known that such input intensification, along with low-diversity agricultural systems, contributes to soil erosion, water pollution, atmospheric carbon levels, and diminishes biodiversity (Goulson et al., 2015; Tilman et al., 2011; Tissier et al., 2016; Kinzig et al., 2006).

Crop diversification. A sustainable alternative to intensive monoculture appears to be *crop diversification*, i.e. the practice of cultivating in a given area more than one crop belonging to different species. The main underlying idea is that, in natural systems, ecosystem functions are heavily influenced by species richness (Malézieux, 2012). By providing additional ecosystem services, *crop diversification* helps to stabilize food production over time (Renard and Tilman, 2019) and prevent toxic pesticides and synthetic fertilizers (Duru et al., 2015), avoid excessive tillage and preserve soil, water and biodiversity (Barot et al., 2017; Beillouin et al., 2019; Tamburini et al., 2020; Beillouin et al., 2021). Benefits of crop diversification in industrial agriculture is being investigating not only from an agro-ecological perspective but also in terms of its socio-economical impacts as well as its burden in terms of managerial complexity. In particular, the main challenge for an agronomic advisor is to determine a trade-off between the maximization of agro-ecological services and the minimization of management complexity (Ditzler et al., 2021).

The main goal of crop diversification is the maximization of positive interactions among different crop species. To achieve this goal, there are two main agronomic strategies: *crop rotation* and *intercropping*. **Crop rotation** involves *temporal diversification*. Specifically, it aims to minimize the impact of monoculture farming by reducing its intensity, which refers to the duration of time during which the same crop is grown on a specific land plot. The crop rotation calendar of a land plot

* Corresponding author.
E-mail address: lucio.colizzi@uniba.it (L. Colizzi).

https://doi.org/10.1016/j.compag.2024.109162
Received 16 November 2023; Received in revised form 6 May 2024; Accepted 9 June 2024
Available online 26 June 2024

indicates the period from planting to harvesting for each crop in the rotation, as well as fallow periods and the planting of green manure crops (Leoni et al., 2015). Crop rotation offers various benefits, including mitigation of soil damage and preservation of soil resources (Dogliotti et al., 2003; Venter et al., 2016), as well as improved weed suppression (Weisberger et al., 2019). **Intercropping systems** are based on *spatial diversification* achieved by simultaneously cultivating mutually beneficial plant types in close proximity to each other. The optimization goal is to maximize positive spatial interactions. In Khanal et al. (2021), the authors provide a review of the main existing metrics used in the scientific literature to assess intercropping systems, in terms of direct and private costs and benefits, as well as indirect and social costs and benefits.

Strip intercropping. In intercropping systems, crops can be grown together according to three main layouts (Bybee-Finley and Ryan, 2018). **Mixed intercropping** refers to grow several crop species with no particular spatial configuration. **Relay intercropping** refers to staggered planting of two or more crops so that only parts of their life cycles overlap. In this article, we focus on **strip intercropping** layout, where two or more crop species are grown adjacent to one another. Plants are arranged in long and narrow strips, consisting of one or more rows. Such layout facilitates independent crop management by existing farm machinery and promotes ecological interactions between adjacent strips (Juventia et al., 2022). Strip intercropping experienced popularity in the United States during the latter half of the 20th century as a means to address soil erosion and reduce reliance on mineral fertilizers (Francis et al., 1986). Moreover, it has been extensively utilized in China for numerous decades (Li et al., 2020). Despite this, the adoption of strip intercropping remained limited in Europe. Potential explanations for this disparity include not only cultural variations but also systemic differences in intercropping practices. Indeed, in Europe, most systems are mixed intercropping with cereal/legume combinations harvested together for animal feed. The potential of strip intercropping was highlighted in a comprehensive global meta-analysis comparing various intercropping methods in Yu et al. (2015). The analysis revealed a 25% increase in yield for strip cropping systems compared to sole crops, along with a higher land equivalent ratio for strip cropping compared to mixed intercropping. Other recent meta-analysis show that strip intercropping reduces both pest infestation (Tajmiri et al., 2017) and disease incidence (Zhang et al., 2019).

Implementing strip intercropping. Ditzler and Driessen (2022) discussed about the socio-technical lock-ins encountered by farmers when they try to implement an intercropping system. Such challenges are present at all levels of production, from field to market. Indeed intercropping practices often require strategic investments, such as acquiring specialized machinery and making adjustments to management, processing, and marketing methods (Morel et al., 2020). From this perspective, strip intercropping offers a notable implementation advantage over other intercropping techniques. Indeed strip configurations can be adjusted to accommodate the working width of available on-farm machinery, while still facilitating interaction among crops placed in adjacent rows.

In this paper we focus on the Crop Planting Layout Problem (CPLP), which represents one of the initial steps in the design of an intercropping system. The goal of the CPLP is to fulfil known crop demands while maximizing positive interactions between neighbouring plants. We specifically focus on CPLP in strip intercropping systems, where each row can accommodate multiple crop species. This is a relatively new farming method aimed to enhance the delivery of agro-ecological services by reducing the size of homogeneous field units, each meant as a strip portion planted with a single crop species. As the resolution of in-field crop diversity increases, agroecosystems become increasingly capable of self-regulation (Van Apeldoorn et al., 2011). This places the farmer in a role that is less focused on externally regulating nutrient cycles and crop protection, but more on facilitating agroecological processes to obtain harvestable yields (Philip Robertson et al., 2014; Storkey et al., 2015; Tittonell et al., 2016). As the management role of the farmer becomes more complex and knowledge-intensive, the importance of using quantitative tools to support the design of cropping systems also increases. Current quantitative methods already prescribe knowledge-intensive decisions, primarily regarding crop selection, rotation, and land-area assignment (Dogliotti et al., 2003; Groot et al., 2010; Martin et al., 2019). To the best of our knowledge, the Crop Planting Layout Problem in intercropping systems has only recently been addressed for the first time with a (semi-quantitative) approach by Juventia et al. (2022). In particular, the authors have proposed a framework for the integration of practical implementation-oriented criteria with the solution of the CPLP in strip intercropping. The proposed framework aims to provide a systematic exploration and evaluation of strip intercropping configurations. The authors presented a case study in which about 100,000 layouts were automatically generated by two quantitative tools, i.e., ROTAT and RotaStrip. The generated layouts were then filtered to retain configurations that met a set of operating constraints. The filtered configurations were ranked according to some indicators. Finally, the researchers selected 16 Pareto-optimal layouts and visually presented them to farmers for the final selection. One of the main limitations of the framework proposed by Juventia et al. (2022) lies in the use of quantitative methods, originally designed for farms with one main crop per year and, therefore, not suitable for dealing with the combinatorial optimization structure of the CPLP in strip intercropping systems. This paper aims to overcome this drawback by applying Constraint Programming (CP), an Artificial Intelligence (AI) planning technique, to formulate and solve optimally the CPLP in strip intercropping systems. In particular, the Crop Planting Layout Problem is (declaratively) modelled through CP by encoding decisions as *variable* starting and ending positions of *homogeneous* field units, each corresponding to a strip portion planted with a single crop species. Relationships among variables (i.e. decisions) are expressed through mathematical and symbolic constraints.

The constraint programming paradigm. The advantage of solving combinatorial optimization problems through CP is two-fold. From a modelling point of view, in a constraint language complex relationships can be expressed with declarative, compact and flexible models (Focacci et al., 2002). From a problem solving perspective, CP solvers implement a *model-and-run* paradigm, allowing to obtain good quality solutions with a reasonable computational burden (Adamo et al., 2016). During the solution process, constraints interact through shared variables. Searching for an optimal solution is interleaved with constraint reasoning. In particular a constraint propagation (filtering) algorithm infers removal of infeasible values from variable domains. A constraint is considered propagated if no more infeasible values can be inferred for the variables involved in the constraint. The iterative process underlying constraint propagation is finite and incomplete, i.e., it does not guarantee the pruning of all infeasible values. For this reason, propagation is interleaved with search. When propagation reaches a fixed point, a branching step partitions the problem into subproblems, defined by instantiating a variable to its values in its domain. Specifically, in the CPLP, a variable instantiation represents the assignment of a *homogeneous* field unit to a portion of strip starting and ending at given positions. Each assignment defines a CPLP subproblem. After the branching step, the CP solver continues its search by solving all generated CPLP subproblems. In constraint programming, the optimization goal is implemented as a sequence of feasibility problems. As soon as a feasible CPLP solution is reached, the corresponding objective function value is computed. Then a bounding constraint is added into the constraint pool, requiring that further feasible layouts must have a higher value of total positive interactions.

It is worth noting that there are combinatorial problems that are more easily solved using Mixed Integer Linear Programming (MILP). This is the case of *pure* problems that are likely to be tackled with

Mathematical Programming approaches, leveraging their (well studied) geometrical structure. A typical example is the Travelling Salesman Problem where Branch-and-Cut approaches (Applegate et al., 1998) outperform CP based approaches. Nevertheless, this gap is significantly reduced when, for example, time-interval based constraints are considered (Schreye, 1999). Broadly speaking, Constraint Programming has shown to be more robust and efficient than the MILP approach for modelling and solving a large panel of combinatorial optimization problems, where decisions correspond to intervals with *variable* starting and ending values. This is due to two main reasons:

- CP interval-based models are concise and very expressive, since CP declarative languages separate interval-based decisions from logical aspects. As a result, CP interval-based models occupy less memory than their MILP counterparts, which require a huge amount of (additional) auxiliary variables to model logical aspects.
- The inference underlying CP interval-based models is very efficient due to the conditional domain maintained in interval-based decisions, which naturally allows conjunctive reasoning between constraints. The inference underlying MILP solvers relies on generating and adding new constraints, contributing to further increase the size of the MILP model.

As aforementioned in the Crop Planting Layout problem, the main decision component prescribes the assignment of a *homogeneous* field unit to a portion of strip starting and ending at given positions. This implies that CPLP is suitable to be successfully formulated as a CP interval-based model. For a more in-depth analysis of the relationship between these two *model-and-run* approaches, i.e. Constraint Programming and Mixed Integer Linear Programming, interested readers may refer to Focacci et al. (2002) and Bockmayr and Kasper (1998).

2. Objectives

The main goals of this paper can be summarized as follows.

- We introduce and motivate the CPLP as a discrete optimization decision problem. In particular we deal with strip intercropping systems, where each row is modelled as a sequence of homogeneous field units, each corresponding to a strip portion planted with a single crop species. The Crop Planting Layout Problem aims to determine the size of each homogeneous field unit so that to satisfy known crop demands and maximize positive interactions between adjacent field units.
- We explore for the formulation and solution of the CPLP, the application of constraint programming, an AI planning technique. Two alternative constraint programming models are proposed. The former makes use of interval variables and can be run by a restricted set of solvers, typically equipped with a scheduling-dedicated modelling language; the latter is based on integer variables and (basically) solver-independent.
- To ensure that the use of AI in agriculture has a positive environmental impact, it is essential to develop approaches that balance effectiveness with the use of computational resources (Rolnick et al., 2022). For these reasons we also study the computational complexity, both theoretically and empirically, of the proposed approach. The computational results showed that the proposed AI-based approach is not energy-intensive to run. It takes only a few minutes to generate high-quality solutions using computational resources typically available on a laptop or smartphone.
- As stated in Ditzler et al. (2023), the user-friendliness of automatic planning tools plays a central role to enhance the way of thinking of stakeholders involved in the design and implementation of strip intercropping systems. To this aim, the proposed optimization approach has been successfully integrated into an AI-based platform as a REpresentational-State-Transfer-Application Programming Interfaces (Rest-API) service, which can be seamlessly invoked through a dedicated mobile application.

The remainder of this paper is organized as follows. Section 3 reviews previous related contributions. Section 4 provides a problem definition along with an illustrative example. Section 5 presents two constraint programming models. In Section 6 a complexity analysis is performed. Section 7 describes system implementation details. In Section 8 is presented a critical evaluation of the proposed approach. Section 9 concludes this paper and discusses possible future research works.

3. Previous related works

In this paper we are interested in the combinatorial optimization issues implied by the higher managerial complexity of crop diversification. To this aim, in the following we review main contributions, where crop diversification has been modelled as a discrete optimization problem.

Literature on optimization of crop rotations. In the literature, determining the crop rotation calendar has been modelled as a scheduling problem known as the Crop Rotation Scheduling Problem (CRSP). There are several contributions addressing solution approaches for CRSP. Clarke (1989) proposes one of the first optimization model dealing with combinatorial aspects of crop rotation. Dogliotti et al. (2003) propose an approach for handling environmental constraints for crop succession in each crop rotation. Alfandari et al. (2015) prove the $\mathcal{NP}$-hardness of the CRSP, when objective function is the minimization of the land used to cover crops demand. In dos Santos et al. (2010), the CRSP is modelled as a 0–1 linear program, maximizing land use and taking into account demand constraints. The authors propose a solution approach based on column generation. A similar solution approach is adopted by dos Santos et al. (2011) to solve a model for the CRSP without demand constraints. The contribution by dos Santos et al. (2011) is the first one to introduce adjacency constraints, i.e. preventing that crops of the same botanical family be cultivated in adjacent plots during the same period. In Aliano Filho et al. (2014), the model proposed by dos Santos et al. (2011) is adapted in order to take into account demand constraints and maximize the profit of planted crops. Metaheuristics are devised as solution approaches. The contribution by Regis Mauri (2019) improves the mathematical model proposed by dos Santos et al. (2011), so as to make it more general and easier to be solved by a commercial solver. In addition, five different relaxation approaches are devised to find high-quality bounds and solutions for the CRSP. Recently, Benini et al. (2023) provide a formal characterization and the complexity analysis of the CRSP based on sequences of k consecutive crops. They propose Integer Linear Programming models tailored for the best practices in Mediterranean pedo-climatic contexts, where k is set to 3. The experimental campaign with real data demonstrates that the proposed models can efficiently solve real-life instances.

Literature on optimization of intercropping systems. Intercropping has been widely practised in smallholder cropping systems. For these reasons, optimization of intercropping systems has been addressed by only a few contributions. According to Czárán and Bartha (1990), the individual-based plant models can be classified as either *grid-based* or *neighbourhood-based*. Each model has its definition of *zone-of-occupancy*, meant as the space required by the plant to allow for proper root development and access to sunlight. In grid-based models the space is discretized into a grid of cells where plants may be placed. The zone-of-occupancy of a plant corresponds to a set of contiguous cells, whilst plant interactions are defined by empirical rules. In neighbourhood models, each individual plant has a zone-of-occupancy modelled in the continuous space, and plant interaction models are based on analytical models. Avigal et al. (2021) formulate the CPLP as a nonlinear program, where the zone-of-occupancy is modelled as a circular zone, whilst

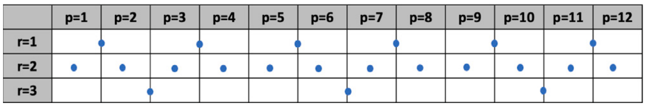


Fig. 1. A planting area with $R = 3$ rows, $N = 12$ positions. Blue circles represent fertigation points located at the centre of the served area.

positive interactions between pair of plants are encoded as coefficients of a matrix, referred to as companionship values. The authors do not provide any details about the solver used to determine the optimal solution. A simulation approach is adopted to tune the input parameter, modelling the maximum level of overlap between plants in the non-linear program. Ding et al. (2020) formulate the CPLP in a continuous space with the aim of determining rows spacing in a strip intercropping layout. Each row can be associated to only one species. The interaction model is neighbourhood-based and the CPLP is solved by a (heuristic) genetic algorithm and rows spacing is determined with the goal of maximizing light interception. Computational tests have been carried out with only two species (soybean and maize) and a given inter-planting ratio. To the best of our knowledge the present work is the first contribution dealing with the CPLP where the plant interaction model is grid-based.

4. Problem formulation and background

In this section we introduce notation underlying the encoding of a CPLP solution along with an illustrative example. Then it is provided a formal definition of the CPLP.

Notation for the planting area. In a strip intercropping system based on a grid-based interaction model, the planting area is partitioned into a grid of cells with R (equally spaced) rows (i.e. strips), each consisting of N positions. We denote with (r, p) the pth cell on row r, with $r = 1, \dots, R$ and $p = 1, \dots, N$. Each row is equipped with a set of fertigation points equally spaced. On the rth row, each fertigation point serves f_r contiguous cells. This implies that each row r has $\left\lfloor \frac{N}{f_r} \right\rfloor$ fertigation points. Fig. 1 shows a planting area consisting of three rows, each one partitioned into twelve cells ($N = 12$). The number of fertigation points is respectively equal to six, twelve and three, i.e. $f_1 = 2$, $f_2 = 1$ and $f_3 = 4$.

Notation for the set of species. We denote with H the number of different species available to be planted. Each crop h is characterized by a demand of d_h units, with $h = 1, \dots, H$. Each unit of crop h requires at least o_h contiguous cells. Since each plant has to be assigned to exactly one fertigation point, each unit of crop h requires $g_{rh} = \max\{o_h, f_r\}$ contiguous cells, when assigned to row r. In a CPLP solution, each single strip is modelled as a sequence of clusters, each corresponding to a strip portion occupied by only one species. The length of a cluster is expressed in terms of number of positions. For each crop h the farmer provides bounds on the cluster length, i.e. a lower bound $c_h \geq o_h$ and an upper bound $C_h \leq N$. The farmer controls through these bounds the layout resolution in terms of crop diversity. In particular, in a CPLP solution, shorter clusters correspond to higher resolution of crop diversity. Nevertheless, the three-year experiment conducted by Ditzler et al. (2023) has shown that ecological objectives can be achieved without compromising production objectives by increasing crop diversity resolution, up to a certain threshold. Beyond this point, the experiment showed that it is unclear whether a balance between ecological and production goals can be maintained. Ditzler et al. (2023) also observed that this uncertainty is primarily attributed to existing agronomic and technological constraints that limit the production capacities of highly diverse cropping systems in industrialized contexts. The bounds on cluster length aim to model the farmer's knowledge of the trade-off between crop diversity and production goals, accounting for agronomic and technological constraints on production capacities. For example, let us consider a planting area with only two crop species, h_1 and h_2, where $o_{h_1} = o_{h_2} = 1$. To satisfy agronomic and technological constraints, the farmer may require that each cluster occupies a portion of the strip equal to at least one-quarter and one-third, respectively, i.e., $c_{h_1} \geq \left\lfloor \frac{N}{4} \right\rfloor$ and $c_{h_2} \geq \left\lfloor \frac{N}{3} \right\rfloor$. On the other hand, to balance ecological and production goals, the farmer may also impose constraints on the maximum crop diversity resolution of a single strip. This can be modelled by specifying the maximum number of distinct clusters that can be placed on a single strip. Below, we provide some examples of how these technological constraints and limits on the crop diversity resolution of a single strip can be encoded using parameters c_{h_1}, c_{h_2}, C_{h_1} and C_{h_2}.

- If the farmer requires that no more than one cluster can be placed on a single strip, then we have $c_{h_1} = c_{h_2} = C_{h_1} = C_{h_2} = N$.
- If the farmer requires that no more than two distinct clusters can be placed on a single strip, then we have $c_{h_1} = c_{h_2} = \left\lfloor \frac{N}{2} \right\rfloor$ and $C_{h_1} = C_{h_2} = N$.
- If the farmer requires that no more than three two distinct clusters must be placed on a single strip, then we have $c_{h_1} = \left\lfloor \frac{N}{4} \right\rfloor$, $c_{h_2} = \left\lfloor \frac{N}{3} \right\rfloor$, and $C_{h_1} = C_{h_2} = N$.

Finally, we observe the higher the upper bound C_h, the greater the variability of a planting layout in terms of distinct cluster length values. For example, due to technological constraints, the farmer may impose a unique cluster length for each crop h, i.e. $c_h = C_h$ with $h = 1, \dots, H$.

Cluster placement policy on a single row. Let consider a cluster consisting of δ_h units of crop h to be planted on row r, where the first N_r cells have been previously assigned to other clusters, with $N_r < N$. The cluster placement policy states that, when placed on row r, the cluster occupies an interval of positions from the cell $(r, N_r + 1)$ to the cell $(r, N_r + \lambda_{rh})$. Algorithm 1 computes the cluster length λ_{rh} as follows. First it computes the minimum number of fertigation points needed for placing δ_h plants of crop h on row r. This value is determined by Algorithm 1 at line 1, as the minimum integer number n_{rh} that satisfies the following inequality:

$$g_{rh} \times \delta_h \leq n_{rh} \times f_r.$$

Then the cluster length is set by Algorithm 1 at line 2 as follows:

$$\lambda_{rh} = n_{rh} \times f_r.$$

Finally Algorithm 1-line 3 checks if the length λ_{rh} satisfies upper and lower bound constraints, that is:

$$c_h \leq \lambda_{rh} \leq \min\{C_h, N - N_r\}.$$

If the feasibility check fails (Algorithm 1-line 4), the policy declares the cluster placement unfeasible for the row r (i.e. $\lambda_{rh} = -1$).

Algorithm 1: Computing cluster length

Input: g_{rh}, f_r, δ_h, C_h, c_h
Output: λ_{rh}

1 $n_{rh} \leftarrow \left\lceil \frac{g_{rh} \times \delta_h}{f_r} \right\rceil$;
2 $\lambda_{rh} \leftarrow n_{rh} \times f_r$;
3 **if** $(\lambda_{rh} > \min\{C_h, N - N_r\}) \vee (\lambda_{rh} < c_h)$ **then**
4 $\quad \lambda_{rh} \leftarrow -1$
5 **end if**

Fig. 2 reports an example corresponding to the placement of three clusters each consisting of three units of crop $\bar{h}$ with $o_{\bar{h}} = 3$, $c_{\bar{h}} = o_{\bar{h}}$ and $C_{\bar{h}} = N$. The planting area is empty, i.e. $N_r = 0$ for $r = 1, 2, 3$. The lengths λ_{rh} computed by Algorithm 1 are equal to

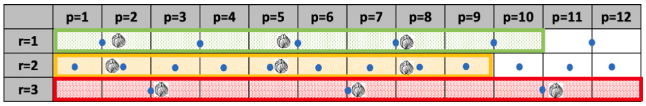


Fig. 2. Example of three clusters placed on the planting area of Fig. 1.

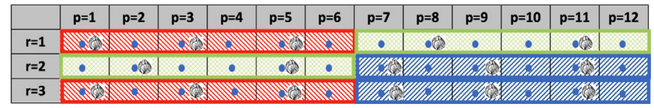


Fig. 3. Example of a CPLP solution with three crops: tomatoes (red cells), broccoli (green cells), peppers (blue cells).

- 10 positions for the first row where $f_1 = 3$, i.e.

$$n_{1\bar{h}} = \left\lceil \frac{\max\{3,2\} \times 3}{2} \right\rceil = 5$$

$$\lambda_{1\bar{h}} = 5 \times 2 = 10$$

- 6 positions for the second row where $f_2 = 1$, i.e.

$$n_{2\bar{h}} = \left\lceil \frac{\max\{3,1\} \times 3}{1} \right\rceil = 9$$

$$\lambda_{2\bar{h}} = 9 \times 1 = 9$$

- 12 positions for the third row where $f_3 = 4$, i.e.

$$n_{3\bar{h}} = \left\lceil \frac{\max\{3,4\} \times 3}{4} \right\rceil = 3$$

$$\lambda_{3\bar{h}} = 3 \times 4 = 12$$

The cluster placement is feasible for all three rows, i.e. $3 \leq \lambda_{r\bar{h}} \leq 12$ with $r = 1, 2, 3$.

Notation for interactions. The farmer evaluates feasible crop planting layouts in terms of the positive interactions triggered by placing in proximity mutual beneficial crops. We consider two cells (r, p) and (r', p) to be *adjacent* if they belong to *adjacent* rows and share a boundary that is not a discrete set of points. Each crop planting layout is evaluated in terms of interactions occurring between adjacent cells. To this aim we use a *score* matrix $A \in \mathbb{R}^{H \times H}$, where the integer coefficient a_{hk} is an empirical measure of the symmetric relationship between plants of species h and k, with $a_{hk} = a_{kh}$ and $h, k = 1, \ldots, H$. More formally an interaction occurs when species h and k occupy cells (r, p) and $(r+1, p)$, with $r = 1, \ldots, R-1$ and $p = 1, \ldots, N$. The coefficient a_{hk} is an integer value, which models an interaction between crops h and k that might be either mutual beneficial ($a_{hk} > 0$) or not-beneficial ($a_{hk} < 0$) or neutral ($a_{hk} = 0$). In the following, we will use both the terms *companionship value* and *score value* to refer to the coefficient a_{hk}. We also assume that the farmer provides a set S of crop pairs. Each pair $(h, k) \in S$ corresponds to a *no-adjacency constraint*, stating that it is not feasible to place at adjacent positions crops h and k. A relevant example of no-adjacency constraints is provided by allelopathy, i.e. a crop h releases toxic chemicals that inhibit growth of another crop k. Such not-beneficial interactions are modelled with $a_{hk} = a_{kh} = -M$, with M a large positive value.

An illustrative example. Fig. 3 reports an example of a feasible layout. The planting area has three rows with twelve positions, i.e. $R = 3$ and $N = 12$. The colour of each vertex denotes the associated crop. Regarding planting capacity of each row, the planting area of Fig. 3 refers to the basic case where each cell is served by one fertigation point, i.e. $f_r = 1$ with $r = 1, 2, 3$. The crop demands refer to three crops: tomatoes (red cells), broccoli (green cells) and peppers (blue cells). The demands are 6 units of tomatoes (i.e. $d_1 = 6$), 4 of broccoli (i.e. $d_2 = 4$) and 2 of peppers (i.e. $d_3 = 2$). The level of occupancy of the three species is respectively 2 cells for tomatoes (i.e. $o_1 = 2$), 3 cells for broccoli (i.e. $o_2 = 3$), 2 cells for peppers (i.e. $o_3 = 2$). Lower and upper bounds on cluster length are $c_h = o_h$ and $C_h = N$, with $h = 1, 2, 3$. The layout in Fig. 3 prescribes two clusters for each crop with each cluster spanning 6 positions. As far as the layout score is concerned, for each pair of crops h and k the coefficient $a_{hk} \in \{1, -100, 0\}$. In particular each coefficient models an interaction which might be either mutual beneficial ($a_{hk} = 1$) or not-beneficial ($a_{hk} = -1$) or neutral ($a_{hk} = 0$). The layout of Fig. 3 triggers interactions that are either neutral or beneficial, with a total score of 12. In particular, the first twelve pairs of adjacent cells (i.e., $((r, p); (r+1, p))$ with $r = 1, 2$ and $p = 1, \ldots, 6$) contribute to the total score with the twelve mutual beneficial interactions between tomatoes and broccoli (i.e., $a_{12} = a_{21} = 1$). The remaining twelve pairs of adjacent cells do not contribute to the total score due to neutral interactions. Specifically, peppers and broccoli have a neutral interaction (i.e., $a_{23} = a_{32} = 0$), occurring between the remaining six pairs of adjacent cells belonging to the first two rows (i.e., $((1, p); (2, p))$ with $p = 7, \ldots, 12$). A neutral interaction also occurs between adjacent cells of the second and third row (i.e., pairs of cells $((2, p); (3, p))$ with $p = 7, \ldots, 12$) since they have all been occupied by the same species (i.e., $a_{hh} = 0$ with $h = 1, \ldots, 3$).

Problem definition. Given a planting area and crop demands, a feasible crop planting layout prescribes an assignment of crops to cells compliant with the cluster placement policy, so that to satisfy crop demands, planting capacity of rows, no-adjacency constraints as well as constraints on minimum and maximum cluster length. The crop planting layout problem aims to determine a feasible crop planting layout that maximizes the total score of crop combinations.

5. Constraint programming models

This section is devoted to the formulation of two (alternative) CP models. The proposed models share the following set of parameters to be computed in a pre-processing phase for each pair (r, h), with $h = 1, \ldots, H$ and $r = 1, \ldots, R$.

First, tighter bounds c'_{rh} and C'_{rh} on cluster lengths are computed as follows:

$$c'_{rh} = \min_{i \in [c_h, C_h]} (i | i \mod g_{rh} = 0),$$

$$C'_{rh} = \max_{i \in [c_h, C_h]} (i | i \mod g_{rh} = 0).$$

To model the constraint on the maximum cluster length, we compute the minimum distance $\tilde{c}_{rh}$ between two clusters of species h planted on the same strip, that is:

$$\tilde{c}_{rh} = \min(c'_{rk} | k \in \{1, \ldots, H\} \setminus \{h\}).$$

Given row r, the maximum number of clusters of crop h that can be placed on the row r can be computed by taking into account the demand d_h, cluster length constraints and row planting capacity as follows:

$$\eta_{rh} = \left\lfloor \frac{\min\{d_h \times g_{rh}, \left\lfloor \frac{N}{g_{rh}} \right\rfloor\} + \tilde{c}_{rh}}{c'_{rh} + \tilde{c}_{rh}} \right\rfloor.$$

Table 1 summarizes all symbols used in this section.

5.1. A CP model based on interval variables

Interval variables are a modelling feature provided by some CP solvers, e.g. Google ORTools (Perron and Furnon, 2022) and IBM CP Optimizer (IBM ILOG, 2023). Even though such modelling concepts are typically exploited to represent activities in scheduling problems, they are a mathematical abstraction of intervals with *variable* starting and ending values. For these reasons, the use of interval variables can be extended (in quite natural way) to model decisions in the Crop Planting Layout Problem.

Interval variables. An interval variable x is a decision variable whose domain consists of subsets of type $\{\perp\} \cup \{[a, b) \mid a, b \in \mathbb{Z}, a \leq b\}$. Some of the basic constraints on interval variables allow: to limit the possible positions of an interval variable; to define precedence relations between two interval variables. An important characteristic of interval variables is that they can be *optional*. Thus solving a constraint programming model with optional interval variables also means prescribing which optional interval variables will be instantiated (i.e. $x := [a, b)$) and which interval variables will not be present (i.e. $x := \perp$). In particular when the interval variable x is present in the solution (i.e. $x \neq \perp$), its domain is defined by a tuple of ranges:

$$([\sigma_{min}, \sigma_{max}], [\epsilon_{min}, \epsilon_{max}], [\lambda_{min}, \lambda_{max}]).$$

The range $[\sigma_{min}, \sigma_{max}] \in \mathbb{Z}$ represents the domain of the starting position. Similarly, $[\epsilon_{min}, \epsilon_{max}] \in \mathbb{Z}$ and $[\lambda_{min}, \lambda_{max}] \in \mathbb{Z}$ denote, respectively, the domain of the ending position and length. The (internal) consistency of the variable x is based on the internal consistency of the ranges defining its domain. When the bounds of such ranges assume inconsistent values (for instance because $\sigma_{min} > \sigma_{max}$ or because $\epsilon_{min} - \sigma_{max} > d_{max}$), if presence status of the interval has been already set to selected (i.e. $x \neq \perp$), then a failure is triggered. Otherwise, the interval presence status is automatically set to unselected (i.e. $x := \perp$).

As far as the CPLP is concerned, we recall that the parameter η_{rh} represents the maximum number of *no-overlapping* clusters of crop h that can be placed on the row r, with $h = 1, \ldots, H$, $r = 1, \ldots, R$. In particular, we model with the optional interval variable x_{rhi} the ith interval of positions on row r occupied by crop h, whose domain is represented by the tuple of ranges $([1, R], [1, N], [c'_{rh}, C'_{rh}])$, with $h = 1, \ldots, H$, $r = 1, \ldots, R$ and $i = 1, \ldots, \eta_{rh}$. In the following we denote with $\lambda(x)$, $\sigma(x)$ and $\epsilon(x)$ functions that return, respectively the length, the first position and the final position of an interval variable x.

Sequence Variables. A CPLP solution prescribes for each row a sequence of (no-overlapping) interval variables. To represent this component of the solution, the proposed model exploits the notion of sequence variable, a decision variable whose value corresponds to a permutation of (present) interval variables belonging to the set $\mathcal{A}$. More formally, let assume that all variables in $\mathcal{A}$ have been instantiated, with $n = |\mathcal{A}|$. A permutation π of $\mathcal{A}$ is a function $\pi : \mathcal{A} \to [0, n]$, with the length of the permutation equal to the number of variables present in $\mathcal{A}$. The domain of a sequence variable $\boldsymbol{s}$ defined on $\mathcal{A}$ is the set of all possible permutations π. For example if $\mathcal{A} = \{x, x'\}$ is a set of two interval variables with x being present and x' optional, the domain $\Pi(\mathcal{A}) = \{(x), (x, x'), (x', x)\}$ of the sequence $\boldsymbol{s}$ defined on $\mathcal{A}$ consists of 3 permutation values $\{\pi_1, \pi_2, \pi_3\}$ such that $\pi_1(x) = 1$, $\pi_1(x') = 0$, $\pi_2(x) = 1$, $\pi_2(x') = 2$, $\pi_3(x) = 2$, $\pi_3(x') = 1$. Constraints on sequence variables model rules for cutting off *unfeasible* permutations or for specifying the order of intervals in the permutation in terms of relative position of their start and end values. We model with $\boldsymbol{s}_r$ a sequence variable with $\boldsymbol{s}_r \in \Pi(\mathcal{A}_r)$, where $\mathcal{A}_r$ is the set of interval variables x_{rhi} associated to row r, i.e. $\mathcal{A}_r = \{x_{rhi} | h = 1, \ldots H, i = 1, \ldots, \eta_{rh}\}$ and $r = 1, \ldots, R$.

Integer variables. Finally we denote with δ_{rhi} the number of plants placed in the ith interval of positions on row r occupied by crop h, with $h = 1, \ldots, H$, $r = 1, \ldots, R$ and $i = 1, \ldots, \eta_{rh}$.

Global constraints. A powerful modelling tool concerns *global constraints*, which allow a concise description of the problem as well as efficient and effective propagation through special purpose inference algorithms. In particular we exploit the global constraint $noOverlap(s_r, M^r, After)$, which defines a sequence s_r of non-overlapping interval variables, with each interval variable in the sequence constrained to end before the start of all its successors in the sequence. The transition distance matrix $M_r : \{1, \ldots H\} \times \{1, \ldots H\} \to \mathbb{Z}^+$ defines the minimal number of positions that must separate two clusters in the row r. Since we aim to forbid overlapping of clusters in each row r and enforce the minimal distance between consecutive clusters, we define the matrix M_r as follows:

$$m_{rhk} = \begin{cases} 0 & h \neq k, \\ \widetilde{c}_{rh} & h = k, \end{cases}$$

with $h, k = 1, \ldots, H$ and $r = 1, \ldots, R$.

CP solvers offer constraints over interval variables that make easy to handle complex relationship without using complex expressions of logical connectors. On the other hand CP community lacks of a set of such constraints that is shared by all constraint programming solvers. For these reasons, we formulate the first model by using constraints extracted from IBM's CP Optimizer. This implies that it may not be possible to run the exact same model in another solver such as OR Tools, because these features might require different names.

$$\max \sum_{h=1}^{H} \sum_{k=1}^{H} a_{hk} \times \sum_{r=1}^{R-1} \sum_{i=1}^{\eta_{rh}} \sum_{j=1}^{\eta_{r+1,k}} overlapLength(x_{rhi}, x_{r+1,k,j}, 0) \tag{1}$$

s.t.

$$\sum_{r=1}^{R} \sum_{i=1}^{\eta_{rh}} \delta_{rhi} = d_h \qquad h = 1, \ldots, H \tag{2}$$

$$\lambda(x_{rhi}) = \left\lceil \frac{g_{rh} \times \delta_{rhi}}{f_r} \right\rceil \times f_r \qquad h = 1, \ldots, H, i = 1, \ldots, \eta_{rh}, r = 1 \ldots, R \tag{3}$$

$$noOverlap(\boldsymbol{s}_r, M_r, After) \qquad r = 1 \ldots, R \tag{4}$$

$$noOverlap([x_{rhi}]_{i=1}^{\eta_{rh}}, [x_{r+1,kj}]_{j=1}^{\eta_{r+1,k}}) \qquad r = 1 \ldots, R-1, (h, k) \in S \tag{5}$$

The objective function (1) states that the model aims to maximize the total companionship. In particular the $overlapLength(x, x', absVal)$ method calculates the length of the overlap between interval variables x and x'. When interval variable x or x' is absent, the function returns the value $absVal$. Constraints (2) requires that the CPLP solution satisfies the demand of crop $h = 1, \ldots, H$. Constraints (3) determine the cluster length according to the placement policy (Algorithm 1: line 1–line 2). Constraints (4) forbid overlapping of clusters in each row r and enforce the minimal distance between consecutive clusters.

No-adjacency constraints are modelled as no-overlapping constraints by (5).

5.1.1. Boosting propagation and search

Propagation algorithms aim to identify a feasible solution by filtering out the domain of each decision variable into a single value. We start by observing that a constructive heuristic for CPLP should determine first cluster lengths satisfying constraints (2)–(3) and, then, cluster first-positions satisfying no-overlapping constraints (4)–(5). This implies that the feasibility of a CPLP solution is mainly due to length values of the selected intervals. For these reasons we annotate the model with

$$strong(\lambda(x_{rhi})) \qquad r = 1, \ldots, R \quad h = 1, \ldots, H \quad i = 1, \ldots, \eta_{rh}.$$

to encourage the solver to enrich stronger (higher inference) constraints on length variables. Assigned an integer value to δ_{rhi}, constraints (3) consider domain values of interval lengths individually, with the propagation algorithms dynamically discovering which values in each domain either satisfy or violate constraints (3). Nevertheless we can precompute such values and store them in the sets L_{rh}, where

$$L_{rh} = \{0\} \cup \{\lambda \mid \lambda \in [c'_{rh}, C'_{rh}] : \lambda \ mod \ f_r = 0\}$$

with $r = 1, \ldots, R$ and $h = 1, \ldots, H$. Then the solver can be enriched with higher inference by adding the boolean constraints (6). In particular constraints (6) determine whether the interval length $\lambda(x_{rhi})$ is contained within the value set L_{rh}.

$$allowedAssignments(\lambda(x_{rhi}), L_{rh}) \tag{6}$$

with $r = 1, \ldots, R$, $h = 1, \ldots, H$ and $i = 1, \ldots, \eta_{rh}$.

After propagation, if there is at least one variable with multiple values, branching (value instantiation) takes place and the search continues. Efficient branching strategies play a key role in reaching a feasible solution within shorter computational time. Problem structure(s) can be exploited in boosting the search procedure by avoiding to explore equivalent search space. To this aim we consider the other main component of a CPLP solution: sequencing decisions. No overlapping constraints (4)–(5) rely on sequence variables associated to interval variables. In particular, permutations in the domain of sequence variables are symmetric in nature, since we can *easily* generate a new permutation from another one. Indeed given a CPLP solution, if we swap on the same row two clusters of the same crop with the same length, then the total companionship of the new solution will not change. In order to prune off such symmetric solutions we add the following constraints, where $h = 1, \dots, H$, $r = 1, \dots, R$ and $i = 1, \dots, \eta_{rh}$

$$endBeforeStart(x_{rh,i-1}, x_{rhi}, \widetilde{c}_{rh}) \quad i \geq 2 \tag{7}$$

$$x_{rhi} \neq \perp \Rightarrow x_{rh,i-1} \neq \perp \quad i \geq 2 \tag{8}$$

$$before(s_r, x_{rh,i-1}, x_{rhi}) \quad i \geq 2 \tag{9}$$

$$endOfPrev(s_r, x_{rhi}, 1) = \sigma(x_{rhi}) \tag{10}$$

$$startOfNext(s_r, x_{rhi}, N+1) = \epsilon(x_{rhi}) \tag{11}$$

The main underlying idea is to model precedence relationship between clusters of crop h on row r. Constraints (7) and (8) guarantee consistency between conditional domains of interval variables x_{rhi} and its immediate predecessor $x_{rh,i-1}$. In particular constraints (7) exploit the $endBeforeStart(x, x', c)$ method, stating that c is the minimum distance (i.e. number of positions) between the end of cluster x and the start of cluster x'. It imposes the inequality $\epsilon(x) + c \leq \sigma(x')$. Constraints (8) require that the interval variable x_{rhi} is selected only if the interval variable $x_{rh,i-1}$ is also present in the solution. Constraints (9) exploit the $before$ method in order to require that, given two instantiated interval variables $x_{rh,i-1} \neq \perp$ and $x_{rhi} \neq \perp$ with $i \geq 2$, the interval x_{rhi-1} has to appear before the interval $x_{rh,i}$. Other intervals may be ordered in between the two. Constraints (10) and (11) model consistency among domains of interval variables included in the sequence s_r. In particular constraints (10) state that the starting position of the interval variable x_{rhi} has to be equal to the end of the interval variable that precedes interval x_{rhi} in sequence variable s_r. The method $endOfPrev$ returns value 1 when interval x_{rhi} is present (i.e. $x_{rhi} \neq \perp$) and is the first interval of sequence s_r. When the interval x_{rhi} is not present (i.e. $x_{rhi} = \perp$), $endOfPrev$ returns the constant value 0. Similarly constraints (11) state that the ending position of the interval variable x_{rhi} has to be equal to the start of the interval variable that is next to interval x_{rhi} in sequence variable s_r. The method $startOfNext$ returns value $N+1$ when interval x_{rhi} is present (i.e. $x_{rhi} \neq \perp$) and is the last interval of sequence s_r. When the interval x_{rhi} is not present (i.e. $x_{rhi} = \perp$), $startOfNext$ returns the constant value 0.

5.2. A CP model based on integer variables

We propose an alternative (solver-independent) CP model, based on integer variables z_{rhj} modelling, for each crop h, the starting position of the jth cluster on row r, with $h = 1, \dots, H$, $r = 1, \dots, R$ and $j = 1, \dots, \eta_{rh}$. Its domain is the finite set $\{0, \dots, N\}$. In particular, if $z_{rhj} = 0$ then the corresponding cluster is not selected by the current CPLP solution. If $z_{rhj} \neq 0$, then the integer variable $y_{rhj} > 0$ represents the length of the corresponding cluster, with $y_{rhj} \in \{c'_{rh}, \dots, C'_{rh}\} \bigcup \{0\}$. A null length (i.e. $y_{rhj} = 0$) is implied by a not selected cluster (i.e. $z_{rhj} = 0$). Given two adjacent rows r and $r+1$, if the CPLP solution selects two clusters associated to two distinct crops h and k, with $h \neq k$, then the local companionship is *proportional* to the number $\theta(h, k, r, i, j)$ of *adjacent* positions computed as:

$$\theta(h,k,r,i,j) = \max\{\min\{z_{rhi} + y_{rhi}, z_{r+1,k,j} + y_{r+1,k,j}\} - \max\{z_{rhi}, z_{r+1,k,j}\}, 0\},$$

Table 1
Index sets, parameters and decision variables.

Parameters	
H	Number of different species, with index $h \in \{1, \dots, H\}$ and $k \in \{1, \dots, H\}$
R	Number of strips, with index $r \in \{1, \dots, R\}$
N	Number of positions for each strip, with index $p \in \{1, \dots, N\}$
f_r	Number of contiguous cells served by each fertigation point on rth row
d_h	Demand of the species h
o_h	Occupancy of species h
g_{rh}	Number of contiguous cells requested for crop h, equal to $\max\{o_h, f_r\}$
c_h	Lower bound on the cluster length of crop h provided as input by the farmer
C_h	Upper bound on the cluster length for each crop h provided as input by the farmer
N_r	Cells that have been previously assigned to other clusters on row r
n_{rh}	Minimum number of fertigation points such that $g_{rh} \times \delta_h \geq n_{rh} \times f_r$
λ_{hr}	Cluster length, computed as $n_{rh} \times f_r$
a_{hk}	Coefficient of companionship between species h and k
M	A large positive value
c'_{rh}	A tighter lower bound on the cluster length of crop h computed as a multiple of g_{rh}
C'_{rh}	A tighter upper bound on the cluster length of crop h computed as a multiple of g_{rh}
$\tilde{c}_{rh}$	Minimum distance between two clusters of species h in a strip r
η_{rh}	Maximum number of clusters of crop h that can be placed on row r with index $j \in \{1, \dots, \eta_{rh}\}$
$\mathcal{A}_r$	Set of interval variables x_{rhi} associated to row r
M_r	Matrix defining the minimal number of positions that must separate two clusters in the row r
L_{rh}	Set of values of interval lengths multiples of f_r
Index sets	
$\mathcal{S}$	Set of no-adjacency crop pairs
Interval variables	
x_{rhi}	The ith interval of positions on row r occupied by crop h
s_r	A sequence variable in $\Pi(\mathcal{A}_r)$
Integer variables	
z_{rhj}	Starting position of the jth cluster on row r for each crop h, with $j = 1, \dots, \eta_{rh}$
y_{rhj}	Length of the jth cluster on row r for each crop h
δ_{rhj}	Units of crop h placed in the jth cluster on row r

with $r = 1, \dots, R-1$, $h, k = 1, \dots, H$, and $i, j = 1, \dots, N$. For example, given two adjacent rows r and $r+1$, if two clusters occupy, respectively, the intervals $[z_{rhi}, z_{rhi} + y_{rhi}] = [2, 7]$ and $[z_{r+1kj}, z_{r+1kj} + y_{r+1kj}] = [3, 9]$, then

$$\theta(h,k,r,i,j) = \max\{\min\{7,9\} - \max\{2,3\}, 0\} = 4.$$

We still denote with δ_{rhj} the number of plants of crop h included in the jth cluster placed on row r, with $h = 1, \dots, H$, $r = 1, \dots, R$ and $i = 1, \dots, \eta_{rh}$. The Crop Planting Layout Problem can be formulated as:

$$Maximize \quad \sum_{h=1}^{H} \sum_{k=1}^{H} a_{hk} \times \sum_{r=1}^{R-1} \sum_{i=1}^{\eta_{rh}} \sum_{j=1}^{\eta_{r+1,k}} \theta(h,k,r,i,j) \tag{12}$$

s.t.

$$z_{rhj} = 0 \implies y_{rhj} = 0 \quad h = 1, \dots, H, r = 1, \dots, R, j = 1, \dots, \eta_{rh} \tag{13}$$

$$z_{rhj} + y_{rhj} \leq N + 1 \quad h = 1, \dots, H, r = 1, \dots, R, j = 1, \dots, \eta_{rh} \tag{14}$$

$$\sum_{r=1}^{R} \sum_{j=1}^{\eta_{rh}} \delta_{rhj} = d_h \quad h = 1, \dots, H \tag{15}$$

$$y_{rhj} = \left\lceil \frac{g_{rh} \times \delta_{rhj}}{f_r} \right\rceil \times f_r \quad h = 1, \dots, H, j = 1, \dots, \eta_{rh}, r = 1 \dots, R \tag{16}$$

$$z_{rhj} \neq 0 \wedge z_{rki} \neq 0 \Rightarrow z_{rhj} + y_{rhj} \leq z_{rki} \vee z_{rki} + y_{rki} \leq z_{rhj} \quad h, k = 1, \dots, H, r = 1, \dots, R, j = 1, \dots, \eta_{rh}, i = 1, \dots, \eta_{rk} \tag{17}$$

$$z_{rhi} \neq 0 \wedge z_{rhi+1} \neq 0 \implies z_{rhi} + y_{rhi} + \tilde{c}_{rh} \leq z_{r,h,i+1} \quad h = 1, \dots, H, r = 1, \dots, R, i = 1, \dots, \eta_{rh} - 1 \tag{18}$$

$$z_{rhi} \neq 0 \wedge z_{r+1,k,j} \neq 0 \implies z_{rhi} + y_{rhi} \leq z_{r+1,k,j} \vee z_{rhi} \geq z_{r+1,k,j} + y_{r+1,k,j} \quad i = 1, \dots, \eta_{rh}, j = 1, \dots, \eta_{r+1,k}, r = 1, \dots, R-1, (h,k) \in \mathcal{S} \tag{19}$$

$$z_{rhi} \neq 0 \Rightarrow z_{rh,i-1} \neq 0 \qquad r = 1,\dots,R, h = 1\dots,H, i = 2,\dots\eta_{hr} \tag{20}$$

The objective function (12) states that the CPLP aims to maximize the total companionship. Constraints (13) model the relationship between cluster starting position and the corresponding length. Inequalities (14) model upper bound on the ending positions of clusters. Equalities (15) represent the demand satisfaction constraints. Constraints (16) model the cluster length according to the placement policy (Algorithm 1: line 1–line 2). Constraints (17) model the no-overlapping constraints for pair of clusters placed on the same row. Similarly constraints (18) model the minimum *distance* between two clusters placed on the same row and associated to the same crop. No-adjacency constraints are modelled by (19). Finally constraints (20) aim to model symmetry breaking by modelling a precedence relationship between the clusters of each crop placed on the same row.

6. Complexity

Let us denote with d-CPLP, the decision version of the CPLP, meant as the problem of determining if there exists a feasible solution satisfying all crop demand. Let us assume that the d-CPLP does not admit an answer yes (i.e. it is not guaranteed that a feasible solution meeting all crop demand exists). In this case one can decide to unmeet part of the demand and introduce a new variable u_h, denoting the amount of unsatisfied demand for crop h, with $h = 1,\dots,H$. The demand constraints and the objective function become:

$$\sum_{r=1}^{R}\sum_{j=1}^{\eta_h} \delta_{rhj} + u_h = d_h \qquad h = 1,\dots,H,$$

$$\min \sum_{h\in H} \omega_h \times u_h, \tag{21}$$

where ω_h is the cost of one unit of unmet demand of crop h, with $h = 1,\dots,H$. In particular when all costs ω_h are positive, solving the d-CPLP is equivalent to asking for any feasible solution of cost zero under (21). Therefore, a reduction to d-CPLP suffices for the total companionship function.

Theorem 1. *d-CPLP is strongly NP-complete.*

Proof. We prove the thesis by reduction from the strongly $\mathcal{NP}$-complete 3-PARTITION problem (Garey and Johnson, 1979), which can be defined as follows.

Instance. We are given an integer $b \in \mathbb{N}$ and a set of $3n$ integers $Q = \{q_1,\dots,q_{3n}\}$ satisfying the following two conditions

i $\sum_{\ell=1}^{3n} q_\ell = n \times b$
ii $b/4 < q_\ell < b/2\ \forall \ell \in \{1,\dots,3n\}$.

Problem. Find whether there exists a partition of Q into n subsets $Q_1,\dots,Q_n$ that satisfies $\sum_{q\in Q_{\ell'}} q = b$, $\forall \ell' \in \{1,\dots,n\}$.

Given any instance of 3-PARTITION, we build an instance of d-CPLP with a set $\mathcal{H}$ of n crops, i.e. $\mathcal{H} = \{h_1,\dots,h_n\}$. All crops have the same demand b, i.e. $d_h = b$ with $h \in \mathcal{H}$. Each crop h is characterized by a single value of cluster length, with $c_h = C_h = N$, and each *unit* of crop h occupies a single position on a row, i.e. $o_h = 1$ with $h \in \mathcal{H}$. As a result each row can be associated to only one cluster (*no-overlapping constraints*). The planting area consists of $3n$ rows $r_1,\dots,r_{3n}$ and each row r_ℓ is equipped with q_ℓ fertigation points, i.e. $q_\ell = \left\lfloor \frac{N}{f_{r_\ell}} \right\rfloor$. This implies that q_ℓ is the planting capacity of row r_ℓ, meant as the maximum number of plants that can be associated to row r_ℓ, with $\ell \in \{1,\dots,3n\}$. As reported in Fig. 4 a d-CPLP instance can be modelled on a complete bipartite graph G, where the set of vertices is partitioned into two subsets. The former corresponds to the vertices on the left in Fig. 4 and models rows (*strip-vertices*). The latter corresponds to the vertices on the right and represents crops (*crop-vertices*). Each crop-vertex h has b units of outflow. Each strip-vertex ℓ has q_ℓ units of maximum inflow. One unit of flow sent on edge $\{h,\ell\}$ models that one unit of crop h has been assigned to row ℓ, with a consumption of one unit of its capacity q_ℓ. A feasible d-CPLP solution corresponds to an integer flow in G, sending out all $n \times b$ units from the crop-vertices while respecting each planting capacity q_ℓ and no-overlapping constraints, i.e., the flow entering a strip-vertex ℓ originates from nodes all associated to the same crop, with $\ell = 1,\dots,3n$.

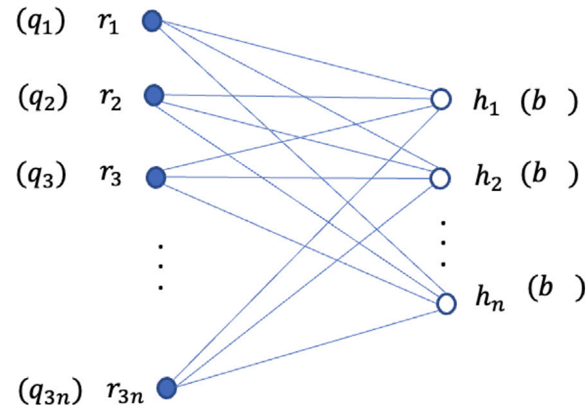


Fig. 4. Reduction of 3-PARTITION to d-CPP.

Let us assume that the d-CPLP instance built above admits answer yes. Given the flow on G corresponding to a feasible CPLP solution, we have that each strip-vertex must receive flow from at most one crop-vertex, i.e. there is room for planting a single cluster on each strip. Moreover each crop-vertex must send flow to at least 3 strip-vertices, i.e. $b/4 < q_\ell < b/2\ \forall \ell \in \{1,\dots,3n\}$. Since $\sum_{\ell=1}^{3n} q_\ell = n \times b$, each strip-vertex ℓ has to receive exactly q_ℓ units of flow. As a result the b outflow units of each crop-vertex must be sent to exactly three strip-vertices. In other words one can find a subset of exactly three strip-vertices whose capacities sum to b, and all these n subsets are disjoint, i.e. the given instance of the 3-PARTITION admits answer yes.

We now demonstrate that if the given 3-PARTITION is a yes-instance then the d-CPLP admits answer yes. By hypothesis, there exists a feasible 3-PARTITION solution, that is a partition of Q into n subsets $Q_1,\dots,Q_n$ that satisfy $\sum_{q\in Q_h} q = b$, $\forall h \in \{1,\dots,n\}$. Since $\sum_{j=1}^{3n} q_\ell = n \times b$ and $b/4 < q_\ell < b/2\ \forall \ell \in \{1,\dots,3n\}$, we have that each subset Q_h consists of three elements. One can determine an integer flow in G, such that each crop-vertex $h \in \{1,\dots,n\}$ has an outflow of b units, reaching three strip-vertices with capacity $q_\ell \in Q_h$, with $\ell \in \{1,\dots,3n\}$. Because the subsets Q_h are disjoint, the inflow of strip-vertex ℓ consists of q_ℓ units, all coming from the same node h. Therefore, the d-CPLP instance has answer yes.

We have proved that the d-CPLP admits answer yes to the instance built above if and only if the given 3-PARTITION instance is a yes-instance. Hence the thesis is proved. □

7. System implementation details

This work is part of a research project aiming to automate the design and control of an intercropping system according to the Industry 4.0 paradigm (Lezoche et al., 2020). The proposed solution approach has been integrated as AI-cloud-service in a microservice oriented platform, developed by the team of researchers involved in the project. This section focuses on the implementation details of the system. Fig. 5 provides a high-level overview of the whole system architecture. In particular, its components include a front-end mobile application, a gateway service, a relational database, and two back-end micro-services. The CPLP and

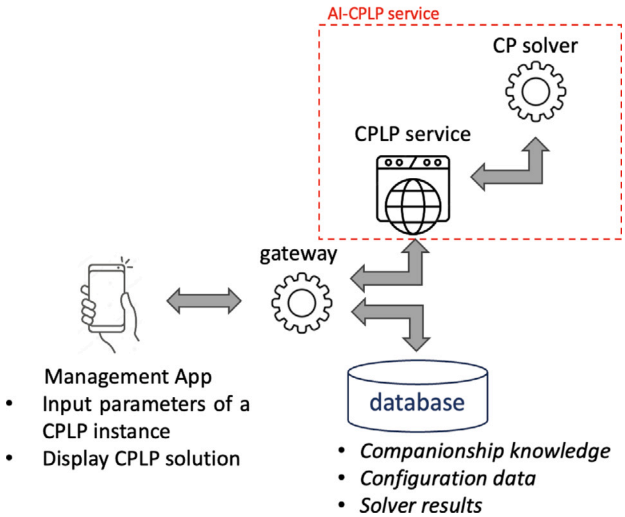


Fig. 5. System architecture.

CP services constitute the core component of the system, referred to as AI-CPLP. In the following we give a general description of each component.

The management mobile application is a user-friendly native application developed for both iOS and Android using Microsoft Xamarin. It is responsible for sanitizing the end user input before invoking the back-end. There is no direct connection between the application and the AI-CPLP back-end. Instead, the mobile application communicates with a gateway service in order to provide some essential information: the irrigation system configuration, the selected plant species and the farmer's preferences in terms of minimum and maximum cluster length for each selected species. Fig. 6 reports some representative screens captured by the mobile client application.

The Gateway is a public-exposed cloud service written in PHP which performs two tasks. First of all, it collects data from the mobile application and securely stores it in a relational database. Secondly, it triggers the AI-CPLP back-end whenever the end user send a request through the application. It is worth noting that all communication messages are in JSON format.

The relational database stores the knowledge base of our system and has been implemented using the open-source MariaDB database management system. It stores every end user request and response. Moreover, it contains the score matrix A. In particular, we refer to the spatial diversity scoring system proposed by Juventia et al. (2022), where each score value a_{hk} is obtained by combining two types of indicators. The first indicator combines farmers' experience with data from literature on intercropping practices. The aim is to model the effects of crops on each other's yield and pest and pathogen attack. An indicator value of −1 (undesirable effect), 0 (neutral effect), and +1 (desirable effect) is assigned. We augmented the first indicator with no-adjacency constraints, encoded in the knowledge base with a score value $a_{hk} = -100$. The second indicator takes into account the spreading of farm operations and soil disturbance. In particular, the indicator value assesses whether a crop pair has sowing or harvesting periods that are at least three weeks apart. In the second indicator this is represented by a minimum score of 0 (i.e., overlapping sowing and harvesting periods), +1 (i.e., either the sowing or the harvesting period overlaps), or +2 (i.e., no overlap between the sowing and harvesting periods of the two crops).

The CPLP micro-service has been coded in Java using Spring Boot framework responsible for back-end validation, CPLP problem instance encoding and resolution through the invocation of CP solver micro-service.

The CP solver micro-service runs an off-the-shelf black-box CP solver in order to provide a solution for a given CPLP problem instance. We implemented the CP model based on integer variables (12)–(20) using MiniZinc (Nethercote et al., 2007), an open-source framework equipped with a *solver-independent* CP modelling language. Performances are strongly related to the algorithm used as back-end solver for filtering, propagation and search. In particular, the MiniZinc framework compiles a CP model into FlatZinc, a low-level language designed to be easily interfaceable to constraint solvers, which might be either external or internal. We chose the (external) Google OR-Tools (Perron and Furnon, 2022) solver. Since 2013, Google OR-Tools has consistently been a top-performing solver in the MiniZinc Constraint Modelling and Optimization Competition (MiniZinc, 2023), an annual international competition that promotes research and development of constraint programming technologies. Concerning the CP model based on interval variables (1)–(11), we used the Optimization Programming Language (OPL) (Van Hentenryck, 1999), a proprietary modelling language equipped with a repertoire of scheduling functions tailored for the CP Optimizer (IBM ILOG, 2023), the commercial CP solver developed by IBM ILOG.

7.1. Computational campaign

The aim of our computational experiments was to assess the performance of the proposed models on realistic instances. All test files are available at https://cloud-simple.it/cplp. All the experiments were run on a standalone Linux machine with an Intel Core i7 processor composed by 4 cores clocked at 2.5 GHz and equipped with 16 GB of RAM. We have tested the proposed models on a set of 572 instances. As aforementioned this work is part of a research project aiming to automate the design and control of intercropping systems. To this aim demands and parameters inherent the planting areas were provided by farmers involved in the research project as stakeholders. In particular, test instances refer to two types of planting area: the former has $R = 6$ rows with a number of positions $N \in \{33, 60, 100, 135\}$ for each row, the latter has $R = 8$ rows and can only accommodate at most $N = 100$ or 135 plants on each row. Each position was equipped with a fertigation point, i.e. $f_r = 1$ with $r = 1, \dots, R$. The demand concerns 8 horticulture species, each characterized by its own occupancy and level of companionship with the other species, as detailed in Table 2. It is worth noting that the level of plant occupancy has been chosen based on the farmer's experience with intercropping systems. In particular, the distance between adjacent strips and the levels of occupancy have been selected with the aim of maximizing light interception. Each instance refers to H different species with $H \in \{2, 4, 6, 8\}$. Cluster length constraints follow five different policies, detailing the percentage of coverage of a single cluster, i.e.:

$\mathcal{P}_1$ $c_h = C_h = N$,

$\mathcal{P}_2$ $c_h = \frac{N}{2}$ and $C_h = N$,

$\mathcal{P}_3$ $c_h = \frac{3N}{8}$ and $C_h = \frac{N}{2}$,

$\mathcal{P}_4$ $c_h = \frac{N}{4}$ and $C_h = N$,

$\mathcal{P}_5$ $c_h = \frac{N}{4}$ and $C_h = \frac{N}{2}$

with $h = 1, \dots, H$. For any feasible combination of parameters ($H, R, N, \mathcal{P}$), we generated up to five instances. In particular, no instances were generated for the infeasible combinations $(8, 6, 100, \mathcal{P}_1)$ and $(8, 6, 135, \mathcal{P}_1)$, whilst only one instance was generated for each combination $(8, 8, N, \mathcal{P}_1)$ with $N \in \{100, 135\}$. Farmers also required that no-adjacency constraints were exploited to model the *antagonism* between species, meant as pair of species (h, k) with $a_{hk} = -100$ and $h, k = 1, \dots, H$. In particular we generated two types of instances for each combination $(H, R, N, \mathcal{P})$, labelled as $Hard$ and $Soft$. The $Hard$ instances encoded *plant antagonism* with *hard* no-adjacency constraints,

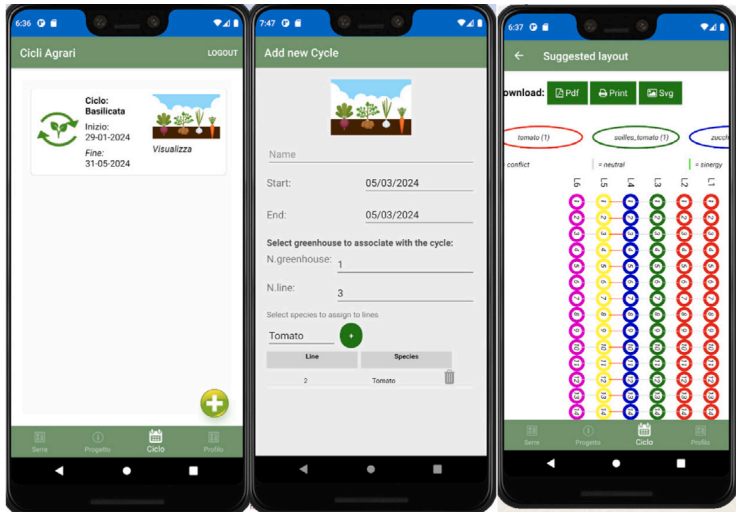


Fig. 6. Details of the management mobile app of Fig. 5.

i.e. $S = \{(h,k) \mid a_{hk} = -100, h = 1\ldots, H, k = 1,\ldots,H\}$. The $Soft$ instances refers to $soft$ no-adjacency constraints, i.e. $S := \emptyset$. This leads to a computational campaign carried out on 1144 instances. In the following we briefly refer to

- the model based on interval variables as IV model;
- the model based on integer variables as IP model.

Instances were solved with a time limit of 480 s. The CP-optimizer solver was able to determine a feasible solution for all instances of both variants of IV model, with 440 out of 572 instances solved to optimality. The OR-Tools was not able to determine at least one feasible solution for 21 instances ($Hard$) and 32 instances ($Soft$). As far as the IP model is concerned, the number of instances solved to optimality by OR-Tools was 386 ($Hard$) and 345 ($Soft$). Fig. 7 details results also for time limits lower than 480 s. Results show that CP-Optimizer required at least 120 s to determine at least one feasible solution for all instances of the IV model, whilst the remaining time was spent to improve the solution quality. It is worth noting that solutions violating adjacency constraints are feasible for the $Soft$ instances of both models. Such an increase in the number of feasible solutions worsens the success rate of OR-Tools, whilst it has no significant impact on the success rate of CP-Optimizer. In particular for the 21 $Hard$-instances and 32 $Soft$-instances of the IP model, no feasible solution was found even when we increased the time limit to 3600 s. In order to investigate which are the complicating constraints for such instances of the IP model, we report in Table 3 the success rates of IP model for each policy. Results show that OR-Tools solver was able to find at least one feasible solution for all instances of IP model with a minimum cluster length not lower than 37% of the number N of row positions, i.e. policies $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. Nevertheless, a lower minimum cluster length makes difficult to find a feasible solution for instances of IP, i.e. $\mathcal{P}_4$ and $\mathcal{P}_5$. The worst success rate occurred for $Soft$ instances with $\mathcal{P}_4$ and $\mathcal{P}_5$, corresponding to instances with the largest set of feasible solutions. Additional tables are available at https://cloud-simple.it/cplp. Results show that for both $Hard$ and $Soft$ variants of IV model, CP-Optimizer took (on average) less than 20 s to find the optimal solution for about 420 instances out to 572 and less than 35 s to determine at least one feasible solution for all 572 instances. On the other hand the OR-Tools determined a feasible solution within 202 (166) s on average for 540 (551) out of 572 instances of the $Soft$ ($Hard$) variant of IP model.

We have observed that the most challenging instances occur when using the cluster length policy $\mathcal{P}_5$, where the total number of feasible clusters ranges from 2 to 4 for each strip. Given that we considered 8 crop species and 8 strips, we can assert that the proposed approach

Table 2
Score matrix of species and their respective occupancy. Symbols + and − correspond to $a_{hk} = 1$ and $a_{hk} = -100$, respectively. Empty cells refer to the default value $a_{hk} = 0$.

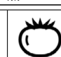
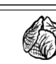
species								
tomato			−	−		+		
zucchini				−		+		
pepper	−			+		+		
cucumber	−	−	+			+	−	
eggplant						+		
lettuce	+	+	+	+	+			+
pumpkin				−				
beetroot						+		
occupancy	2	3	2	3	2	1	5	1

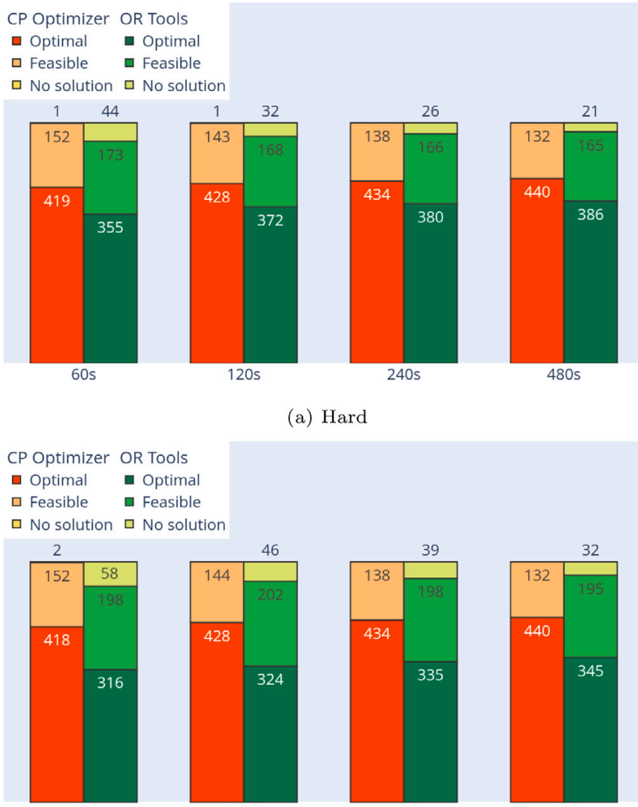
Table 3
Feasibility OR-Tools: policies.

$\mathcal{P}$	Count	Hard				Soft			
		60 s	120 s	240 s	480 s	60 s	120 s	240	480 s
1	92	92	92	92	92	92	92	92	92
2	120	120	120	120	120	120	120	120	120
3	120	120	120	120	120	120	120	120	120
4	120	108	110	114	115	102	108	111	114
5	120	88	98	100	104	80	86	90	94
Total	**572**	**528**	**540**	**546**	**551**	**514**	**526**	**533**	**540**

is well-suited for instances with a total number of feasible clusters not exceeding 256. By adjusting the lower bound on cluster length c_h and reducing the number of species H, our model can efficiently determine the optimal layout for larger planting areas. For instance, it can determine the optimal layout for a planting area with 128 strips and 4 crop species using the cluster length policy $\mathcal{P}_1$. However, for scenarios with a higher number of feasible clusters, our approach may not guarantee the determination of high-quality CPLP solutions within a few minutes using computational resources typically available on a laptop or a smartphone.

8. Discussion

In this paper, we introduced an application of AI planning techniques for supporting the design of strip intercropping layouts. Given a planting area and known crop demands, the optimization goal was



(a) Hard

(b) Soft

Fig. 7. Success rates for $Hard$ and $Soft$ instances.

to maximize crop diversity, while considering production and technological constraints. We also delved into studying the computational complexity of the proposed AI-based approach, both theoretically and empirically. Initially, we demonstrated the strong $\mathcal{NP}$-completeness of the considered optimization problem, regardless of the objective function under consideration. Subsequently, we carry out a computational analysis to evaluate the performance of the proposed models on real-world datasets consisting of instances with up to eight crops requiring 1080 plants. The computational results demonstrated that the AI-based solution approach is sustainable, since it is not energy-intensive to run (Rolnick et al., 2022). Indeed, it determines high-quality solutions in just a few minutes by requiring computational resources typically available on a laptop or smartphone. Options exist for making the proposed approach more general by incorporating other diversity dimensions in the scoring system.

Recently (Juventia et al., 2022) devised a framework for the systematic exploration and evaluation of spatio-temporal strip intercropping layouts. The main limitation of the proposed approach relies on the use of quantitative tools not designed to cope with multiple dimensions in crop diversifications. To overcome this limitation, the framework requires user supervision to automatically filter layouts generated by quantitative methods. Our optimization approach can be augmented, in quite natural way, to include temporal crop diversification as follows. Given a partition of the planning horizon in T time periods, the proposed CP formulations should take into account temporal dimension by defining the time copy (r, p, t) of cell (r, p) in period t, with $r = 1, \dots R$, $t = 1, \dots, T$ and $p = 1, \dots, N$. Then an asymmetric score value $a'_{h,k}$ quantifies interactions between crop h and crop k assigned, respectively, to the time copies (r, p, t) and $(r, p, t+1)$ of cell (r, p). If it is required that crop h has not to be assigned to the same cell for two consecutive periods, then the negative interaction $a'_{hh} = -100$ is added in the knowledge base. The total score is then obtained by adding both temporal and spatial scores of crop combinations in the layout. It is worth noting that, agronomic research in the past mainly addressed temporal aspects at annual scale. Embedding temporal diversity into our CP models would enable the evaluation of spatio-temporal interactions at a finer time resolution, such as sowing and harvesting. Our modelling approach can be extended to include further crop diversity dimensions. For example recently (Ditzler et al., 2021) devised a conceptual framework for a scoring system with three diversity dimensions, which are space, time and genes. The genetic diversity was meant to model single and mixed cultivar. Our CP formulation can take into account such third diversity dimension by adding the mixed cultivar to the set of crop species. If crop h is a mixed cultivar, then a positive interaction $a_{hh} = 1$ is reported in knowledge base. In a more recent contribution, Ditzler et al. (2023) carried out a three-year experiment to assess the potential of high-resolution intercropping system. To interpret empirical data, the authors relied on the scoring system proposed in Ditzler et al. (2021). These recent contributions suggest that an interesting area for future research is the development of ad-hoc automatic planning tools designed to optimize multi-dimension crop diversity at field scale. Indeed, although our CP model can be naturally extended to incorporate dimensions such as time and genetic diversity, solving it with standard solvers may become energy-intensive due to the exponential increase in the number of constraints and decision variables.

As observed by Ditzler et al. (2023), strip intercropping can, for the most part, be implemented using common agricultural machinery. To implement the layout prescribed by the proposed algorithm, currently available planting seedling machines can be utilized (Williames, 1997). These machines are equipped with seedbeds, each accommodating from 60 to 120 young seedlings ready for transplantation. Each seedbed is typically a flat, raised area of soil where seeds are sown for germination and seedling growth before being transplanted to their final growing location. Seedlings are arranged in rows within the seedbed, with adequate spacing between each seedling to allow for proper root development and access to sunlight. Currently, the sowing phase is carried out by *seeding roll machines* designed for single-crop seedbeds. The farmers involved in the project collaborated with a robotics developer to test a proof-of-concept of a sowing robotic arm. Fig. 8 shows how the system architecture of Fig. 5 has been extended with a mobile app. This app has been specifically designed to retrieve a previously saved CPLP solution and convert it into a robot work plan. The cell rows underlying the CPLP solution are mapped with the positions of the seedbeds, and then each crop-to-cell assignment is converted into a pick-and-place action. The resulting work plan is then sent to the robotic arm. The seeds are made available on a rotating plate, equipped with compartments corresponding to the distinct crop species. Using a pneumatic end effector, the robotic arm picks up seeds of one crop from the rotating plate and places them into the positions of the seedbed assigned to the crop in the CPLP solution. Fig. 9 shows the proof-of-concept of the sowing robotic arm made by 3D printing, as well as the interface of the mobile app.

Finally it is worth noting that the research project was conducted participatorily with farmers interested in experimenting with strip intercropping systems. One of the participating farmers successfully used the AI-based platform to design and implement a strip intercropping layout in the greenhouse shown in Fig. 10, comprising 5 rows, 5 clusters and 5 species. According to the farmer, this marks the initial stride towards integrating the research project's outcomes into a large-scale industrialized setting.

9. Conclusions

This paper proposes a new AI-based approach for automating intercropping systems. By presenting two constraint programming models, we have successfully demonstrated the determination of optimal crop

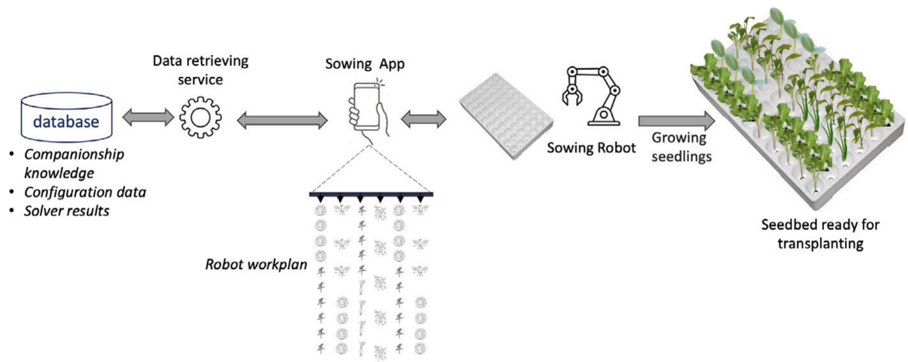


Fig. 8. How to integrate a sowing robotic arm in the system architecture of Fig. 5.

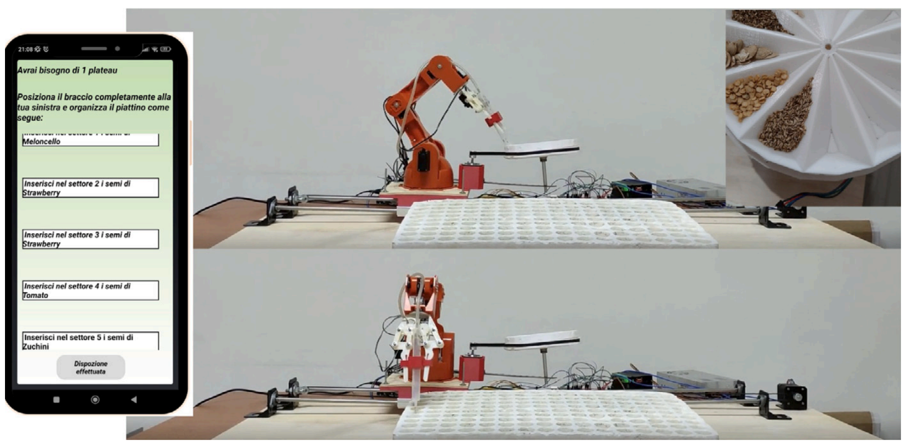


Fig. 9. The proof-of-concept of the sowing robotic arm and the sowing mobile app.

Fig. 10. The test case.

planting layouts with the objective of maximizing the positive interactions between plants placed in close proximity. The proposed models are tailored for strip intercropping systems where the layout of fertigation points is divided into rows and the plant interaction model is grid-based. We prove that the considered problem is strongly $\mathcal{NP}$-complete. Integrating the proposed planning approach within an AI-based platform has enabled the development of an intelligent decision-making service, that can be seamlessly invoked by farmers through a dedicated mobile application. Numerical experiments on a set of benchmark instances, based on real-world data, show that standard solvers provide a good tradeoff between solution quality and computational resource utilization. In particular, the two proposed models were both able to solve instances where no more than two different species are placed on the same row. When it is feasible to place a higher number of species on a single row, the model based on interval variables outperformed the model based on integer variables. This suggests that the difficulty of the decision problem is mainly due to the assignment decision component, prescribing how to arrange plants on each strip. We finally observe, the proposed decision-making service has been developed under the assumption that the scoring system encodes only spatial crop diversification. An emerging research area is represented by the development of optimization tools for incorporating multiple diversity dimensions at the field scale as part of cropping system planning. A promising research direction is to extend the proposed AI-based planning approach to include crop diversification at several dimensions, including spatial, temporal, genetic, and fertigation diversity. This introduces additional constraints to the (hard) assignment decisions and suggests that developing ad-hoc optimization algorithms is an interesting area for future research.

CRediT authorship contribution statement

Tommaso Adamo: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Lucio Colizzi:** Writing – original draft, Validation, Supervision, Software, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Giovanni Dimauro:** Validation, Methodology, Formal analysis, Conceptualization. **Emanuela Guerriero:** Writing – original draft, Validation, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Deborah Pareo:** Writing – review & editing, Validation, Software, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors are grateful to all farmers involved in the survey for their collaboration for providing data and information used in the experimentation. This work was partly supported by Ministero dell'Università e della Ricerca (Progetti DM 1062 del 10/08/2021 - Azione IV.4 - Azione IV.6 - cod. 02-G-14860-1) and Regione Puglia (Italy) (Progetto di ricerca SIMPLe (Sistemi Integrati autoMatizzati Per la coLtivazione Indoor) PSR 2014–2020 Puglia -Articolo 35 del Regolamento (VE) n. 13052013 Misura 16 Cooperazione - Sottomisura 16.2 - CUP:B89J20000100009 - Sostegno a progetti pilota e allo sviluppo di nuovi prodotti, pratiche, processi e tecnologie). This support is gratefully acknowledged.

References

Adamo, T., Calogiuri, T., Ghiani, G., Grieco, A., Guerriero, E., Manni, E., 2016. Neighborhood synthesis from an ensemble of MIP and CP models. In: Learning and Intelligent Optimization: 10th International Conference, LION 10, Ischia, Italy, May 29–June 1, 2016, Revised Selected Papers 10. Springer, pp. 221–226. http://dx.doi.org/10.1007/978-3-319-50349-3_15.

Alfandari, L., Plateau, A., Schepler, X., 2015. A branch-and-price-and-cut approach for sustainable crop rotation planning. European J. Oper. Res. 241 (3), 872–879.

Aliano Filho, A., Florentino, H., Pato, M., 2014. Metaheuristics for a crop rotation problem. Int. J. Metaheuristics 3 (3), 199–222.

Applegate, D., Bixby, R., Chvátal, V., Cook, W., 1998. On the solution of traveling salesman problems. Doc. Math. 645–656, URL http://eudml.org/doc/233207.

Avigal, Y., Deza, A., Wong, W., Oehme, S., Presten, M., Theis, M., Chui, J., Shao, P., Huang, H., Kotani, A., et al., 2021. Learning seed placements and automation policies for polyculture farming with companion plants. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 902–908.

Barot, S., Allard, V., Cantarel, A., Enjalbert, J., Gauffreteau, A., Goldringer, I., Lata, J.-C., Le Roux, X., Niboyet, A., Porcher, E., 2017. Designing mixtures of varieties for multifunctional agriculture with the help of ecology. A review. Agron. Sustain. Dev. 37, 1–20.

Beillouin, D., Ben-Ari, T., Makowski, D., 2019. A dataset of meta-analyses on crop diversification at the global scale. Data Brief 24, 103898.

Beillouin, D., Ben-Ari, T., Malézieux, E., Seufert, V., Makowski, D., 2021. Positive but variable effects of crop diversification on biodiversity and ecosystem services. Global Change Biol. 27 (19), 4697–4710.

Benini, M., Blasi, E., Detti, P., Fosci, L., 2023. Solving crop planning and rotation problems in a sustainable agriculture perspective. Comput. Oper. Res. 159, 106316.

Bockmayr, A., Kasper, T., 1998. Branch and infer: A unifying framework for integer and finite domain constraint programming. INFORMS J. Comput. 10 (3), 287–300.

Bybee-Finley, K.A., Ryan, M.R., 2018. Advancing intercropping research and practices in industrialized agricultural landscapes. Agriculture 8 (6), 80.

Clarke, H.R., 1989. Combinatorial aspects of cropping pattern selection in agriculture. European J. Oper. Res. 40 (1), 70–77.

Czárán, T., Bartha, S., 1990. The effect of spatial pattern on community dynamics; A comparison of simulated and field data. In: Progress in Theoretical Vegetation Science. Springer, pp. 229–239.

Ding, W., Fan, C., Xu, L., Xie, T., Liu, Y., Wan, Z., Max, N., 2020. Optimization method to obtain appropriate spacing parameters for crop cultivation. Int. J. Agric. Biol. Eng. 13 (1), 146–152.

Ditzler, L., van Apeldoorn, D.F., Schulte, R.P., Tittonell, P., Rossing, W.A., 2021. Redefining the field to mobilize three-dimensional diversity and ecosystem services on the arable farm. Eur. J. Agron. 122, 126197.

Ditzler, L., Driessen, C., 2022. Automating agroecology: How to design a farming robot without a monocultural mindset? J. Agric. Environ. Ethics 35 (1), 2.

Ditzler, L., Rossing, W.A., Schulte, R.P., Hageman, J., van Apeldoorn, D.F., 2023. Prospects for increasing the resolution of crop diversity for agroecosystem service delivery in a Dutch arable system. Agric. Ecosyst. Environ. 351, 108472.

Dogliotti, S., Rossing, W., van Ittersum, M., 2003. Rotat, a tool for systematically generating crop rotations. Eur. J. Agron. 19 (2), 239–250.

dos Santos, L.M.R., Costa, A.M., Arenales, M.N., Santos, R.H.S., 2010. Sustainable vegetable crop supply problem. European J. Oper. Res. 204 (3), 639–647.

dos Santos, L.M.R., Michelon, P., Arenales, M.N., Santos, R.H.S., 2011. Crop rotation scheduling with adjacency constraints. Ann. Oper. Res. 190 (1), 165–180.

Duru, M., Therond, O., Martin, G., Martin-Clouaire, R., Magne, M.-A., Justes, E., Journet, E.-P., Aubertot, J.-N., Savary, S., Bergez, J.-E., et al., 2015. How to implement biodiversity-based agriculture to enhance ecosystem services: A review. Agron. Sustain. Dev. 35, 1259–1281.

Focacci, F., Lodi, A., Milano, M., 2002. Mathematical programming techniques in constraint programming: A short overview. J. Heuristics 8, 7–17.

Francis, C., Jones, A., Crookston, K., Wittler, K., Goodman, S., 1986. Strip cropping corn and grain legumes: A review. Am. J. Altern. Agric. 1 (4), 159–164.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability. Vol. 174, freeman San Francisco.

Goulson, D., Nicholls, E., Botías, C., Rotheray, E.L., 2015. Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. Science 347 (6229), 1255957.

Groot, J.C., Jellema, A., Rossing, W.A., 2010. Designing a hedgerow network in a multifunctional agricultural landscape: Balancing trade-offs among ecological quality, landscape character and implementation costs. Eur. J. Agron. 32 (1), 112–119.

IBM ILOG, 2023. User's manual for CP Optimizer: v22.1.1. URL https://www.ibm.com/docs/en/icos/22.1.1.

Juventia, S.D., Norén, I.L.S., Van Apeldoorn, D.F., Ditzler, L., Rossing, W.A., 2022. Spatio-temporal design of strip cropping systems. Agric. Syst. 201, 103455.

Khanal, U., Stott, K.J., Armstrong, R., Nuttall, J.G., Henry, F., Christy, B.P., Mitchell, M., Riffkin, P.A., Wallace, A.J., McCaskill, M., et al., 2021. Intercropping: evaluating the advantages to broadacre systems. Agriculture 11 (5), 453.

Kinzig, A.P., Ryan, P., Etienne, M., Allison, H., Elmqvist, T., Walker, B.H., 2006. Resilience and regime shifts: Assessing cascading effects. Ecol. Soc. 11 (1).

Leoni, C., Rossing, W., van Bruggen, A.H., 2015. CHAPTER 4.2: Crop rotation. In: Plant Diseases and their Management in Organic Agriculture. Am Phytopath Society, pp. 127–140.

Lezoche, M., Hernandez, J.E., Díaz, M.d.M.E.A., Panetto, H., Kacprzyk, J., 2020. Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture. Comput. Ind. 117, 103187.

Li, C., Hoffland, E., Kuyper, T.W., Yu, Y., Zhang, C., Li, H., Zhang, F., van der Werf, W., 2020. Syndromes of production in intercropping impact yield gains. Nat. Plants 6 (6), 653–660.

Malézieux, E., 2012. Designing cropping systems from nature. Agron. Sustain. Dev. 32, 15–29.

Martin, E.A., Dainese, M., Clough, Y., Báldi, A., Bommarco, R., Gagic, V., Garratt, M.P., Holzschuh, A., Kleijn, D., Kovács-Hostyánszki, A., et al., 2019. The interplay of landscape composition and configuration: New pathways to manage functional biodiversity and agroecosystem services across Europe. Ecol. Lett. 22 (7), 1083–1094.

MiniZinc, 2023. The MiniZinc challenge. URL https://www.minizinc.org/challenge.html.

Morel, K., Revoyron, E., San Cristobal, M., Baret, P.V., 2020. Innovating within or outside dominant food systems? Different challenges for contrasting crop diversification strategies in Europe. PLoS One 15 (3), e0229910.

Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G., 2007. MiniZinc: Towards a standard CP modelling language. In: Principles and Practice of Constraint Programming–CP 2007: 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007. Proceedings 13. Springer, pp. 529–543.

Perron, L., Furnon, V., 2022. OR-tools: v9.5. URL https://developers.google.com/optimization/.

Philip Robertson, G., Gross, K.L., Hamilton, S.K., Landis, D.A., Schmidt, T.M., Snapp, S.S., Swinton, S.M., 2014. Farming for ecosystem services: An ecological approach to production agriculture. BioScience 64 (5), 404–415.

Regis Mauri, G., 2019. Improved mathematical model and bounds for the crop rotation scheduling problem with adjacency constraints. European J. Oper. Res. 278 (1), 120–135.

Renard, D., Tilman, D., 2019. National food production stabilized by crop diversity. Nature 571 (7764), 257–260.

Rolnick, D., Donti, P.L., Kaack, L.H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A.S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al., 2022. Tackling climate change with machine learning. ACM Comput. Surv. 55 (2), 1–96.

Schreye, D.D., 1999. Solving TSP with time windows with constraints. In: Logic Programming: Proceedings of the 1999 International Conference on Logic Programming. pp. 515–529.

Storkey, J., Döring, T., Baddeley, J., Collins, R., Roderick, S., Jones, H., Watson, C., 2015. Engineering a plant community to deliver multiple ecosystem services. Ecol. Appl. 25 (4), 1034–1043.

Tajmiri, P., Fathi, S.A.A., Golizadeh, A., Nouri-Ganbalani, G., 2017. Effect of strip-intercropping potato and annual alfalfa on populations of Leptinotarsa decemlineata Say and its predators. Int. J. Pest Manage. 63 (4), 273–279.

Tamburini, G., Bommarco, R., Wanger, T.C., Kremen, C., Van Der Heijden, M.G., Liebman, M., Hallin, S., 2020. Agricultural diversification promotes multiple ecosystem services without compromising yield. Sci. Adv. 6 (45), eaba1715.

Tilman, D., Balzer, C., Hill, J., Befort, B.L., 2011. Global food demand and the sustainable intensification of agriculture. Proc. Natl. Acad. Sci. 108 (50), 20260–20264.

Tissier, M.L., Handrich, Y., Robin, J.-P., Weitten, M., Pevet, P., Kourkgy, C., Habold, C., 2016. How maize monoculture and increasing winter rainfall have brought the hibernating European hamster to the verge of extinction. Sci. Rep. 6 (1), 25531.

Tittonell, P., Klerkx, L., Baudron, F., Félix, G.F., Ruggia, A., Van Apeldoorn, D., Dogliotti, S., Mapfumo, P., Rossing, W.A., 2016. Ecological intensification: Local innovation to address global challenges. Sustain. Agric. Rev. 19, 1–34.

Van Apeldoorn, D.F., Kok, K., Sonneveld, M.P., Veldkamp, T., 2011. Panarchy rules: rethinking resilience of agroecosystems, evidence from Dutch dairy-farming. Ecol. Soc. 16 (1).

Van Hentenryck, P., 1999. The OPL Optimization Programming Language. MIT Press.

Venter, Z.S., Jacobs, K., Hawkins, H.-J., 2016. The impact of crop rotation on soil microbial diversity: A meta-analysis. Pedobiologia 59 (4), 215–223.

Weisberger, D., Nichols, V., Liebman, M., 2019. Does diversifying crop rotations suppress weeds? A meta-analysis. PLoS One 14 (7), e0219847.

Williames, G.A., 1997. Apparatus and method for transferring seedlings from plant trays. US Patent 5, 644, 999.

Yu, Y., Stomph, T.-J., Makowski, D., van Der Werf, W., 2015. Temporal niche differentiation increases the land equivalent ratio of annual intercrops: A meta-analysis. Field Crops Res. 184, 133–144.

Zhang, C., Dong, Y., Tang, L., Zheng, Y., Makowski, D., Yu, Y., Zhang, F., van Der Werf, W., 2019. Intercropping cereals with faba bean reduces plant disease incidence regardless of fertilizer input; A meta-analysis. Eur. J. Plant Pathol. 154, 931–942.