

The Crop Growth Scheduling Problem in Vertical Farming

Alberto Santini¹, Michael Schneider², and Vinicius Greco de Lemos³

¹ Universitat Pompeu Fabra, Barcelona, Spain — alberto.santini@upf.edu

² RWTH Aachen University, Aachen, Germany — schneider@dpo.rwth-aachen.de

³ University of Twente, Enschede, Netherlands — v.grecodelemos@student.utwente.nl

April 21, 2020

Abstract

In this paper, we study the problem of devising an efficient growth schedule for crops growing in shelves inside vertical farming cabinets under controlled growth conditions. By adjusting temperature, humidity, light, and other environmental conditions in different parts of the cabinets, a planner must ensure that crop growth is able to satisfy the demand for the final produce. We propose an integer programming formulation able to capture real-life operational characteristics, including changes of growth conditions on a daily, shelf-by-shelf basis, over a planning horizon of months. We compare four objective functions from which a planner can choose, depending on the concrete operations of the company. A computational study on realistic instances, which we make available as a public dataset, shows that the choice of objective function heavily influences both the difficulty of solving the model with a standard solver and the solution characteristics.

1 Introduction

The stocks of arable land per person are declining worldwide, due to increasing population and urbanisation rates, decreasing water availability, and climate change [4]. Increasing land usage for standard agricultural practices has undesirable effects, such as deforestation, an elevated usage of fertilisers and pesticides, soil degradation and its eventual depletion, low yield per unit of surface, and extensive transportation costs to move produce from the production to the consumption site [2]. All these effects take their toll both economically and, more importantly, on the environment and the well-being of urban and rural communities alike. The large-scale increase in food demand forecast to take place within the next decades has prompted the investigation of alternative production methods. The main aim of these efforts is to increase the yield per square meter while reducing negative effects on the environment and being economically viable [1].

One of the new production methods which is gaining considerable traction is Vertical Farming (VF), i.e., growing crops in vertical stacked layers rather than on the ground [1]. Figure 1 shows three stacked layers hosting mulberry plants. Each shelf provides the plants with nutrients; in the case depicted in the figure, the plants grow without soil and absorb the nutrients directly from water (a growth method known as hydroponics). The shelves also provide light, ventilation and, optionally, controlled temperature and CO₂ levels. The size of the stacks can vary considerably and spans from small shelf cabinets to entire plant factories [8, 7]. Their hosting structures are also diverse and range from specially-built buildings to shipping containers and from reused pre-existing buildings to cabinets no larger than a standard refrigerator. A desirable property of the host structure is that it is isolated from the external environment. This allows the plants to grow in a controlled environment (CE) with regulated levels of light, water, and humidity that, in the most advanced systems, can even vary on a shelf-by-shelf basis. One of the advantages of CE systems is that they are independent of external weather and light conditions and can thus be used in a variety of regions: they are not affected by floods, draughts, and other catastrophic events. They also allow for minimal interaction with the outside environment, sheltering the crops from parasites, pathogens, or heavy metals (all common occurrences in open-air farming) and thus eliminating the need for pesticides and herbicides. Final commercial users, such as restaurants, food markets, or hotels, can accommodate shelf cabinets on their premises, reducing or eliminating any transport cost and the resulting negative effects on the environment.



Figure 1: Stacks of shelves in which mulberries are growing in a hydroponic system, i.e., receiving their nutrients directly from water, without soil. Photograph by Satoshi Kinokuni, distributed under a CC-BY-2.0 license.

On the negative side, growth in a CE is more energetically expensive than open-air farming due to the need of constant artificial lightning. In recent years, however, the energy footprint of CE systems improved due to low-consumption LED lights and the increased efficiency of on-site renewable energy production and storage. In this way, VF may offer opportunities that contribute to achieve the Sustainable Development Goals of the United Nations [11]: *Goal 2: Zero Hunger* could be supported by large-scale VF systems producing staple foods in large quantities in countries with restricted space availability or hostile conditions for open-air farming, while the ability of VF systems to reduce the negative effects of traditional farming as described above can play a role in achieving *Goal 12: Responsible Consumption and Production* and *Goal 15: Life On Land*. Moreover, we note that in pandemic situations like the current outbreak of COVID-19, VF systems may offer the additional autonomy and independence from transport operations that is necessary to implement quarantine measures in affected areas.

Finally, and most important for the definition of our problem, crops growing in CE are not affected by seasonality, and the operators can plan their production to match the demand [2]. The limited capacity of VF shelves, the particular growth schedules that each crop has to follow, and the need to change the environmental conditions on a shelf-by-shelf basis, all constrain how vertical farmers can grow their crops and whether they can meet the demand on time. In this paper, we study the problem of devising an efficient growth schedule for crops growing in VF cabinets under controlled conditions. The decisions involve how to place units of crop on the shelves and how to decide the growth condition of each shelf during a planning horizon of days to months. The aim is to ensure the right conditions during all growth stages of all crops.

The literature on the topic of crop scheduling in VF is scarce. The only work we are aware of is that of Bennell, Martinez, and Potts [3], who use Operational Research techniques for scheduling crops in shelf cabinets. The work stems from an industry collaboration with a local business and has not been published as a technical report or journal paper. The authors study the problem of minimising unmet demand in a growing cabinet in which each shelf can be subject to different lightning and irrigation conditions. Another decision is the growing medium to assign to each shelf, which influences both its capacity and which crops can grow on the shelf. The authors use an Integer Linear Programme (ILP) to precisely describe the problem. The ILP is large (for example, it uses four set of variables, three of which are five-indexed), and although the model is not explicitly provided in their presentation, the authors report that it can solve instances with 3 crops and a 70-day planning horizon, but it is not able to solve a problem with 5 crops within 1 day of computing time.

To tackle realistically-sized instances of the problem, the authors use a heuristic that decomposes the problem into two subproblems. The first tries to minimise unmet demand while satisfying capacity constraints, but it does not allocate crops to shelves. The second subproblem performs the actual allocation, minimising

the number of movements. Thus, the problem has a hierarchical objective function. Because the second subproblem is still computationally challenging, the authors solve it with a rolling-horizon heuristic. Using this approach, they are able to obtain solutions of instances with up to 9 crops, 15 shelves, and a planning horizon of 70 days.

As already noted by Bennell, Martinez, and Potts [3], the problem of crop scheduling in VF has similarities with machine scheduling problems. The analogy links shelves to machines, crop growth phases to jobs, and growth times to processing times. In machine scheduling problems, jobs need be ready at or before their due date, and the planner almost always tries to minimise the total makespan (the time at which the last job ends). Because crops are perishable and backlogs are not admitted, in our problem, crops must be ready for harvest on specific days. The makespan is thus fixed, determined by the last day with demand for a crop. In crop growth, there is no preemption (growth cannot stop and resume at a later time), and different crops can share the same shelf, like in scheduling problems with batches [9]. However, different from classical batching, crops can be added and removed from shelves individually at any time.

Other peculiarities set this problem apart from problems in machine scheduling. For example, crops require specific configurations (e.g., irrigation, CO₂ levels, light cycles) during each growth phase. In this respect, our problem resembles a flow shop problem, in which jobs require a given sequence of operations which have to occur on specific machines. In VF cabinets, however, we do not have a determined shelf for each of these “operations”: shelves can change their configuration daily and provide different growing conditions if the operator adjusts the relevant parameters, such as lightning, water, etc.

We conclude that no other scheduling problem investigated in the literature adequately captures the specificities of growing crops in VF cabinets. Hence, the necessity exists to develop new, dedicated mathematical formulations for this problem.

The main contributions of this paper are the following.

- To the best of our knowledge, we are the first to consider the problem of optimally scheduling crops in VF cabinets. The relevance of the problem stems from the social, environmental, and economic challenges to which VF offers a viable solution, as outlined above.
- We incorporate real-life constraints determined by crop growth conditions and the available infrastructure. Our model is flexible enough to allow for the growth conditions to change daily and on a shelf-by-shelf basis. Growers require such a degree of flexibility to provide the crops with ideal growth conditions and, thus, maximise the quality of their yield.
- We formalise the problem and provide a complete ILP formulation, proposing four models that differ in their objective functions: minimise the number of times crops move from shelf to shelf, minimise the number of times shelves need to change their growth conditions, minimise unmet demand, and minimise the number of shelves used. Decision makers can choose the objective function suitable for their needs.
- We perform an extensive computational study to compare the performance of the models, to determine the size of problem instances solvable via standard solvers, and to compare the operational effects of choosing different objective functions. We use a realistic dataset with instances containing up to six crops, twelve shelves, and a time horizon of 100 days. We conclude that the objective function has a large effect on the difficulty of solving the problem. Moreover, solutions obtained optimising with regards to one objective typically fare poorly with respect to the other objectives.

To study the problem, we formalise it and frame it as an ILP in [Section 2](#). In [Section 3](#), we present the outcome of extensive experiments on realistic instances, providing computational results and managerial insights. We conclude and point out future research directions in [Section 4](#).

2 Mathematical models

To formalise the problem, we introduce the following notation. We consider a set C of crops that grow on a set S of shelves during a time horizon $D = \{1, \dots, \bar{d} - 1\}$. Each element of D represents one day; \bar{d} is the

last day when harvest is possible, and $\bar{d} - 1$ is the last day when growth is possible (assume that crops are harvested at the beginning of the day). We also consider the extended time horizon $D' = \{0\} \cup D$, where we use day 0 to model seeds not yet planted, which will start growth on day 1. On each day $d \in D \cup \{\bar{d}\}$, we have to meet a demand of p_{cd} units for each crop $c \in C$ (with $p_{cd} \in \mathbb{N}$).

A parameter $\delta_{cs} \in \{0, 1\}$ determines compatibility between crops and shelves, taking value 1 iff crop $c \in C$ is able to grow on shelf $s \in S$. For example, some shelves could be not deep or tall enough for growing certain crops. In addition, we define the set of shelves compatible with each crop as $S_c = \{s \in S : \delta_{cs} = 1\}$.

Furthermore, a crop goes through different phases in its growth, each requiring precise conditions, such as temperature level, humidity, and growth medium characteristics. One unit of crop c grows for γ_c days, i.e., it has to spend γ_c days in the VF system. On each day of growth $g \in \{1, \dots, \gamma_c\}$, we require the system to keep the shelf that hosts the unit at condition $k_{c,g} \in K$, where K is the set of possible conditions for the shelves. Practically, K is the set of all feasible combinations of parameters for soil type, temperature, humidity, CO₂, air flow, etc. In real-life applications, the required conditions do not change on a daily basis but only when the crop changes from one *growth phase* to the next, such as germination, seedling growth, etc. The conditions also affect the shelf capacity, i.e., the number of units of crops that can grow on the shelf simultaneously. We denote the capacity of a shelf $s \in S$ under conditions $k \in K$ as q_{sk} , with $q_{sk} \in \mathbb{N}$. Note that the capacity refers to the total number of units of *any* crop that can grow on the shelf at the same time, thereby allowing mixing different crops on the same shelf.

For modelling convenience, we extend the set S with two dummy shelves, obtaining set $S' = \{\sigma, \tau\} \cup S$ and sets $S'_c = \{\sigma, \tau\} \cup S_c$. Element σ represents the *seed vault*, i.e., a virtual location for units of crop before they enter the VF system. Analogously, τ represents the *produce storage*, i.e., the virtual location where units of crop go when they are ready for pick-up. Furthermore, we denote as growth day 0 the last day a unit of crop is in the seed vault. In other words, the set of extended growth days for a crop c is $\{0, 1, \dots, \gamma_c\}$.

Because commercial VF cabinet shelves are often not all different, we can reduce the size of the model by considering the set T of shelf types. Shelves of the same type have the same compatibility with crops and the same capacities. We can then use parameters $\delta_{ct} \in \{0, 1\}$ for compatibility between crop $c \in C$ and shelves of type $t \in T$, and $q_{tk} \in \mathbb{N}$ to denote the capacity of any shelf of type t under condition $k \in K$. Analogously, we can define sets $T' = \{\sigma, \tau\} \cup T$, $T_c = \{t \in T : \delta_{ct} = 1\}$, and $T'_c = \{\sigma, \tau\} \cup T_c$. We denote as n_t , with $n_t \in \mathbb{N}$, the number of shelves of type $t \in T$ available in the VF system.

We use two sets of variables:

- $x_{t_1, d, t_2}^{c, g} \in \mathbb{N}$ is the number of units of crop $c \in C$ in their g -th day of growth ($g \in \{0, \dots, \gamma_c\}$), growing on shelves of type $t_1 \in T'_c$ on day $d \in D'$ and going to shelves of type $t_2 \in T'_c$ on day $d + 1$.
- $y_{t, d, k} \in \mathbb{N}$ is the number of shelves of type $t \in T$ with configuration $k \in K$ on day $d \in D$.

It is helpful to visualise the shelves (as shelf types) and the time horizon as a time-expanded graph, in which paths represent movement of crops while growing. [Figure 2](#) depicts an example with three shelf types, two crops (represented by the dashed and dotted paths, respectively), and no compatibility constraints. The first crop needs to spend three days on a shelf with configuration k_1 , followed by two days on a shelf with configuration k_2 . If we have demand for this crop on day 6, this means the growth needs to start at the beginning of the time horizon and, in fact, the crop leaves the seed vault on day 0 and is already growing on a shelf of type 1 on day 1. The second crop needs four days in the system: during the first two, it needs a shelf with configuration k_1 , and during the second two, a shelf with configuration k_3 . In this example, we assume that the total number of units we are growing does not exceed the capacity of the shelves of type 1 (in configuration k_1), and both crops can be present at the same time on these shelves on day 3.

In the following, we describe in detail a mathematical base model that uses an aggregation of shelves into shelf types. For certain practical settings (e.g., certain objective functions), this assumption has to be dropped, and we give short explanations of how to set up alternative models. We present the constraints of the model in [Section 2.1](#) and propose different possible objective functions in [Section 2.2](#). [Sections 2.3](#) and [2.4](#) explain how to increase the solvability of the base model by fixing variables and adding valid inequalities, respectively.

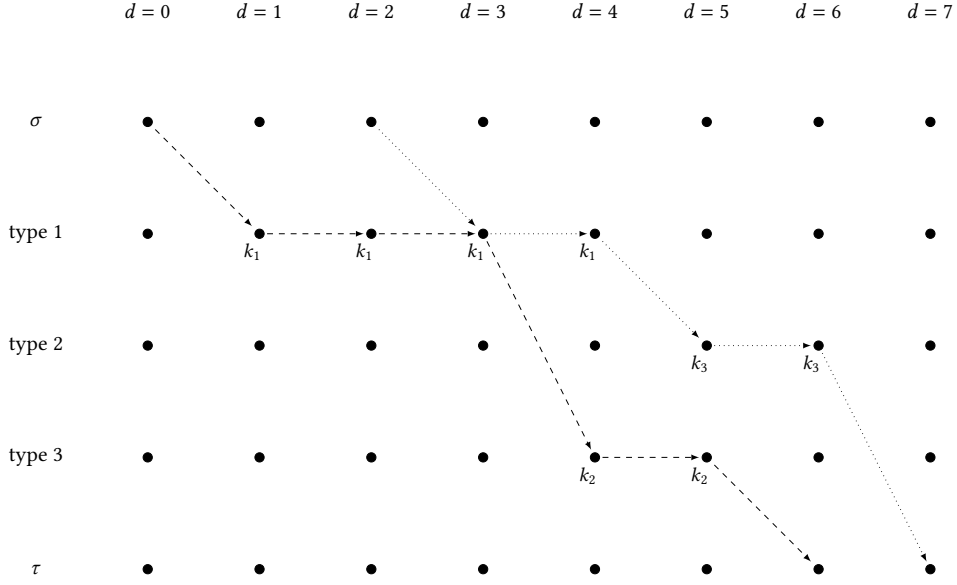


Figure 2: Time-expanded graph whose columns represent days, and rows represent shelf types. The two arrow paths (dashed and dotted) show two crops growing in the system. The labels next to the nodes represent the shelf configurations.

2.1 Constraints

In the following, we describe the constraints to ensure that we satisfy demand and respect capacity constraints and shelf condition compatibility.

Demand satisfaction implies that the correct amount of units of crop reach the produce storage on time:

$$\sum_{t \in T_c} x_{t,d-1,\tau}^{c,y_c} = p_{cd} \quad \forall c \in C \quad \forall d \in D \cup \{\bar{d}\}. \quad (1)$$

Equation (1) guarantees that an amount of units of crop c equal to the demand on day d is sent from any shelf (where it was on day $d - 1$) to the produce storage.

Capacity constraints ensure that a feasible amount of units of crops is grown on any given shelf, on any given day. Recall that shelf capacities are not fixed but depend on the specific configuration used. We can calculate the number of units of crops on shelves of type t_1 with configuration k on a given day d by counting, for example, how many units move out from these shelves, i.e., summing over outgoing arcs.

$$\sum_{t_2 \in T'} \sum_{\substack{c \in C \\ \delta_{ct_1}=1 \\ \delta_{ct_2}=1}} \sum_{\substack{g \in \{1, \dots, y_c\} \\ k_{c,g}=k}} x_{t_1,d,t_2}^{c,g} \leq q_{t_1 k} \cdot y_{t_1,d,k} \quad \forall t_1 \in T \quad \forall d \in D \quad \forall k \in K. \quad (2)$$

Note how equation (2) also serves as a linking constraint, forcing variables y to take non-zero value for a type-day-configuration combination when there are x variables using a shelf of the given type, with the given configuration, on the given day.

Planting constraints: Because the demand determines when the operator needs to harvest a crop, and crop growth lasts a fixed number of days, demand indirectly also determines when the operator will plant the crops.

$$\sum_{t \in T_c} x_{\sigma,d-y_c-1,t}^{c,0} = p_{cd} \quad \forall c \in C \quad \forall d \in \{y_c + 1, \dots, \bar{d}\}. \quad (3)$$

Flow-balance constraints: While equation (1) constrains arcs inbound to the produce storage and equation (3) constrains arcs outbound from the seed vault, the following set of constraints refer to arcs to

and from non-dummy shelves. They make sure to send any amount of crop growing on shelves of one type on a given day, to the same shelf or other shelves for the next day.

$$\sum_{t_2 \in T'_c} x_{t_2, d-1, t_1}^{c, g} = \sum_{t_2 \in T'_c} x_{t_1, d, t_2}^{c, g+1} \quad \forall c \in C \quad \forall g \in \{0, \dots, \gamma_c\} \quad \forall t_1 \in T_c \quad \forall d \in D. \quad (4)$$

Note how [equation \(4\)](#) also ensures that the growth day increases by one each time a day passes.

Configuration constraints ensure that we select no more configurations than there are shelves, for each shelf type on each day.

$$\sum_{k \in K} y_{t, d, k} \leq n_t \quad \forall t \in T \quad \forall d \in D. \quad (5)$$

2.2 Objective functions

Practical applications can vary considerably in the objectives that a planner wants to achieve. In the described VF setting, no single objective function is obviously the correct one to study. Therefore, we investigate a number of meaningful objectives in the following and describe how they can be modelled.

2.2.1 Minimise the number of movements

Moving a crop from a shelf to another one is time-consuming, can damage the crop, or make it undergo unnecessary stress. It would be convenient, then, for crops to stay as much as possible in the same shelf and having the shelf conditions change appropriately to match crop growth phases.

If we use variables aggregated on the shelf types, it is impossible to model each movement of a crop from shelf to shelf. To see why this is the case, consider two crops c_1, c_2 which grow for 2 days ($\gamma_{c_1} = \gamma_{c_2} = 2$), and a VF system with two shelves s_1, s_2 of a single type t , both with large capacities under all growing conditions. Crop c_1 needs to spend one day in condition k_1 , followed by one day in condition k_2 ; crop c_2 needs one day in condition k_1 , followed by one day in condition k_3 .

If we must plant and harvest both crops at the same time, then the following two schedules are both feasible, but the first involves one crop movement while the second involves none:

- Grow both crops c_1, c_2 on shelf s_1 on the first day (in configuration k_1). Then put s_1 in configuration k_2 and keep crop c_1 there, and put s_2 in configuration k_3 and move crop c_2 there.
- Put crop c_1 on shelf s_1 , in configuration k_1 on the first day, and k_2 on the second day. Put crop c_2 on shelf s_2 , in configuration k_1 on the first day, and k_3 on the second day.

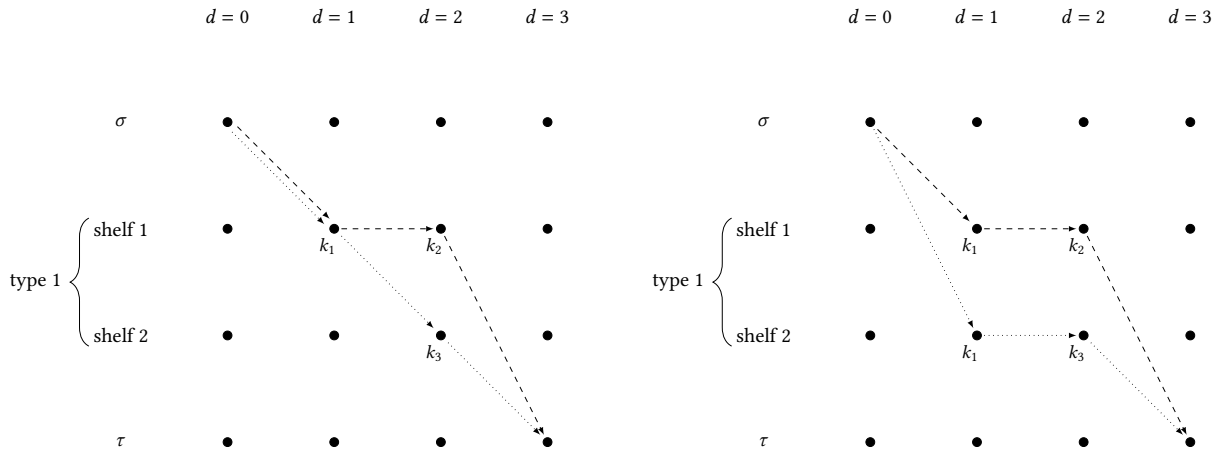


Figure 3: Example of two growth schedules (left and right) which are indistinguishable by looking at the x variables aggregated by shelf type, but are different if we index the x variables over the single shelves.

Figure 3 represents the two possible schedules. Notice how in both cases the x variables would have the same values: $x_{\sigma,0,t}^{c_1,0} = x_{t,1,t}^{c_1,1} = x_{t,2,\tau}^{c_1,2} = x_{\sigma,0,t}^{c_2,0} = x_{t,1,t}^{c_2,1} = x_{t,2,\tau}^{c_2,2} = 1$, with all other x variables equal to zero.

Therefore, to minimise the number of crop movements, we need to use x variables indexed over the single shelves, $x_{s_1,d,s_2}^{c,g} \in \mathbb{N}$ indicating the number of units of crop $c \in C$ which are on shelf $s_1 \in S$ on day $d \in D$ (which is the crop's g -th growth day) and move to shelf $s_2 \in S$ on day $d+1$. Then, we can express the required goal with the minimisation of the following objective function:

$$\sum_{c \in C} \sum_{g=1}^{\gamma_c} \sum_{\substack{s_1, s_2 \in S_c \\ s_1 \neq s_2}} \sum_{d \in D} x_{s_1,d,s_2}^{c,g}. \quad (6)$$

We can easily adjust the constraints noting that the following relation between the shelf and the shelf-type x variables hold:

$$x_{t_1,d,t_2}^{c,g} = \sum_{s_1 \in S_{t_1}} \sum_{s_2 \in S_{t_2}} x_{s_1,d,s_2}^{c,g},$$

where $S_t \subseteq S$ is the set of all shelves of type $t \in T$. We denote the corresponding model as **MinM**.

2.2.2 Minimise the number of reconfigurations

Changing shelf configurations takes time and is prone to errors. In some applications it is advisable to keep shelves in stable conditions and move crops around to the shelf that matches its current requirements. Minimisation of the following objective function reflects this necessity:

$$\sum_{t \in T} \sum_{d=1}^{\bar{d}-2} \frac{1}{2} \sum_{k \in K} |y_{t,d,k} - y_{t,d+1,k}|, \quad (7)$$

where the constant $1/2$ reflects the fact that changing the configuration of one shelf changes the value of two y variables at once. Because no constraint explicitly sets the configuration of an unused shelf, the objective function will make these shelves keep the configuration they had when last used. In this way, we correctly count configuration changes and not “switching on/off” of shelves.

We can linearise equation (7) by replacing variables y with variables $w_{t,d}^{k_1,k_2} \in \mathbb{N}$, counting how many shelves of type $t \in T$ are in configuration $k_1 \in K$ on day $d \in D \setminus \{\bar{d}-1\}$ and then in configuration $k_2 \in K$ on day $d+1$. The objective function becomes

$$\sum_{t \in T} \sum_{d=1}^{\bar{d}-2} \sum_{\substack{k_1, k_2 \in K \\ k_1 \neq k_2}} w_{t,d}^{k_1,k_2}, \quad (8)$$

and we can adjust the constraints noting that we can write variables y as

$$y_{t,d,k_1} = \sum_{k_2 \in K} w_{t,d}^{k_1,k_2} \quad \forall t \in T \quad \forall d \in D \quad \forall k_1 \in K.$$

We denote the corresponding model as **MinR**.

2.2.3 Minimise unmet demand

If it is not guaranteed that a feasible schedule meeting all demand exists, one can decide to keep part of it unsatisfied. In this case, we introduce a new variable $u_{c,d} \in \mathbb{N}$, indicating the amount of unmet demand for crop $c \in C$ on day $d \in D \cup \{\bar{d}\}$. We need to modify equations (1) and (3) as follows:

$$\sum_{t \in T_c} x_{t,d-1,\tau}^{c,\gamma_c} + u_{c,d} = p_{cd} \quad \forall d \in D \cup \{\bar{d}\} \quad \forall c \in C, \quad (9)$$

$$\sum_{t \in T_c} x_{\sigma,d-\gamma_c-1,t}^{c,0} + u_{c,d} = p_{cd} \quad \forall c \in C \quad \forall d \in \{\gamma_c + 1, \dots, p_c\}. \quad (10)$$

Then the objective function becomes

$$\sum_{c \in C} \sum_{d=1}^{\bar{d}} \omega_{cd} u_{c,d}, \quad (11)$$

where $\omega_{cd} \in \mathbb{R}^+$ is the cost of missing one unit of demand of crop c on day d . We denote the corresponding model as **MinUD**.

2.2.4 Minimise the number of shelves used

Moving from the operational to the tactical level, a planner might want to size their VF system and determine what is the smallest number of shelves needed to satisfy their demand in typical scenarios. To do so, we can add a dummy configuration $k_0 \in K$ representing an unused shelf, and a new set of variables $v_t \in \mathbb{N}$ indicating the number of shelves of type $t \in T$ used in the solution. All capacities associated with k_0 will be zero, i.e., $q_{tk_0} = 0$ for all $t \in T$.

The objective function minimises the number of shelves used:

$$\sum_{t \in T} v_t \quad (12)$$

and the following linking constraints ensure variables v_t take the correct values:

$$v_t \geq \sum_{k \in K \setminus \{k_0\}} y_{t,d,k} \quad \forall t \in T \quad \forall d \in D. \quad (13)$$

Because each v_t appears in the minimisation objective function, it will take the smallest value allowed by (13). We denote the corresponding model as **MinS**.

2.2.5 Multi-objective problem

Because no objective function presented above is superior to the others for all applications, a decision maker can consider this problem within a multi-objective optimisation (MOO) approach. In this respect, our model can be adapted using popular multi-objective frameworks, such as hierarchical objectives, weighted sums, ε -constraint and weighted metrics (we refer the reader to recent surveys [6, 5] for further details on MOO). Approaches using MOO appear attractive for real-life applications, considering that—as we report in detail in Section 3.3—optimising with regards to one objective function produces solutions which do not fare well with respect to the other objectives.

2.3 Variable fixing

In our model, the x variables determine the paths in the time-expanded graph, while the y variables assign the configurations to the shelves in a compatible way. As such, a variable $x_{t_1,d,t_2}^{c,g}$ indicates that we use the arc from node (t_1, d) to node $(t_2, d + 1)$ for some units of crop c and that this arc is the g -th in its corresponding path. From this observation, it follows that fixing to zero all the x variables which correspond to infeasible conditions also corresponds to a pruning of the arc set in the graph. In particular, we use the following rules to fix the x variables:

- We neither consider arcs incoming to the seed vault or outgoing from the produce storage, nor arcs from the seed vault not going to non-dummy shelves or coming to the produce storage and not coming from non-dummy shelves:

$$\begin{aligned} x_{t,d,\sigma}^{c,g} = x_{\tau,d,t}^{c,g} = 0 \quad & \forall d \in D' \quad \forall c \in C \quad \forall t \in T_c' \quad \forall g \in \{0, \dots, \gamma_c\}, \\ x_{\sigma,d,\tau}^{c,g} = 0 \quad & \forall d \in D' \quad \forall c \in C \quad \forall g \in \{0, \dots, \gamma_c\}. \end{aligned}$$

- We can remove arcs from the seed vault corresponding to non-zero growth days, or to the produce storage corresponding to unripe crops, or between shelves for infeasible growth days:

$$\begin{aligned} x_{\sigma,d,t}^{c,g} &= 0 \quad \forall d \in D' \forall c \in C \forall t \in T'_c \forall g \in \{1, \dots, \gamma_c\}, \\ x_{t,d,\tau}^{c,g} &= 0 \quad \forall d \in D' \forall c \in C \forall t \in T'_c \forall g \in \{0, \dots, \gamma_c - 1\}, \\ x_{t_1,d,t_2}^{c,g} &= 0 \quad \forall d \in D \forall c \in C \forall t_1, t_2 \in T_c \forall g \in \{0, \gamma_c\}. \end{aligned}$$

- Complementary to the above condition, all arcs with zero growth day have to be outgoing from the seed vault, and all arcs corresponding to ripe crops have to go to the produce storage.

$$\begin{aligned} x_{t_1,d,t_2}^{c,0} &= 0 \quad \forall t_1 \in T' \setminus \{\sigma\} \forall d \in D' \forall t_2 \in T' \forall c \in C, \\ x_{t_1,d,t_2}^{c,\gamma_c} &= 0 \quad \forall t_1 \in T' \forall d \in D' \forall t_2 \in T' \setminus \{\tau\} \forall c \in C. \end{aligned}$$

- We cannot use arcs corresponding to crops which cannot get ripe on time for the last harvest. To this end, we let $d'_c = \max_{d \in D} \{d : p_{cd} > 0\}$ be the last day with some demand for crop $c \in C$, and we set:

$$x_{t_1,d,t_2}^{c,g} = 0 \quad \forall c \in C \forall t_1, t_2 \in T_c \forall d \in \{d'_c - \gamma_c + 1, \dots, d'_c\} \forall g \in \{0, \dots, \gamma_c - (p - d)\}.$$

- We can remove arcs corresponding to unfeasible growth days (when g is greater than d):

$$x_{t_1,d,t_2}^{c,g} = 0 \quad \forall c \in C \forall t_1, t_2 \in T'_c \forall d \in D' : d < \gamma_c \forall g \in \{d + 1, \dots, \gamma_c\}.$$

In addition to the fixing rules described above, we can tighten the upper bound on arcs outgoing from the seed vault and incoming to the produce storage because the demand limits the number of units that are planted and harvested:

$$\begin{aligned} x_{\sigma,d-\gamma_c-1,t}^{c,0} &\leq p_{cd} \quad \forall c \in C \forall t \in T_c \forall d \in \{\gamma_c + 1, \dots, \bar{d}\}, \\ x_{t,d-1,\tau}^{c,\gamma_c} &\leq p_{cd} \quad \forall c \in C \forall t \in T_c \forall d \in D \cup \{\bar{d}\}. \end{aligned}$$

Finally, we can also fix some of the y variables. First notice that, because we know the demand and the growth phases of each crop, we also know how many units of crop will require a given configuration on a given day:

$$\eta_{dk} = \sum_{d'=d+1}^{d+\gamma_c-1} \sum_{\substack{c \in C \text{ s.t.} \\ k=k_{c,\gamma_c-(d'-d-1)}}} p_{cd} \quad \forall d \in D \forall k \in K,$$

where $\eta_{dk} \in \mathbb{N}$ denotes the number of units of any crop which require configuration $k \in K$ on day $d \in D$. We can then fix to 0 all the y variables corresponding to configurations that we do not need:

$$y_{t,d,k} = 0 \quad \forall t \in T \forall d \in D \forall k \in K : \eta_{dk} = 0.$$

2.4 Valid inequalities

A disadvantage of the proposed formulations is their weak linear relaxation, mainly due to the “big M ”-like constraints (2). We describe a set of valid inequalities to strengthen this bound and to speed up the solution of the model.

The first inequality uses parameter η_{dk} to force a minimum number of variables $y_{t,d,k}$ to take value 1 when we need configuration k on a day d . Let $\bar{q}_k = \max_{t \in T} \{q_{tk}\}$ be the largest capacity associated with configuration $k \in K$. Each day, then, we need at least $\lceil \eta_{dk} / \bar{q}_k \rceil$ shelves in configuration k to accommodate the crops which need that configuration. We reflect this with the following valid inequality:

$$\sum_{t \in T} y_{t,d,k} \geq \lceil \eta_{dk} / \bar{q}_k \rceil \quad \forall d \in D \forall k \in K. \quad (14)$$

Note that this inequality is not valid for model *MinUD*. In this model, (14) could make a problem infeasible if there are not enough shelves to accommodate all the crops, when instead the planner could decide not to meet some demand.

In formulations in which we model each shelf independently, we can add “clique-like” constraints to force two crops, requiring two different configurations on a given day, to be on separate shelves. For each day $d \in D$ and configuration $k \in K$, consider the set \mathcal{I}_{dk} of indices (c, g) giving all crops c that require configuration k on day d , being at their g -th day of growth. We would like to use clique constraints to ensure that, on any day d , crops occupying the same shelf s should all be indexed from the same set \mathcal{I}_{dk} . Because the x variables are not binary, it is not possible to enforce such clique constraints, but we can use a weaker form:

$$\sum_{(c,g) \in \mathcal{I}_{dk_1}} \sum_{s' \in S'} x_{s,d,s'}^{c,g} + \sum_{(c,g) \in \mathcal{I}_{dk_2}} \sum_{s' \in S'} x_{s,d,s'}^{c,g} \leq \max\{\eta_{dk_1}, \eta_{dk_2}\} \quad \forall d \in D \quad \forall s \in S \quad \forall k_1, k_2 \in K : k_1 \neq k_2. \quad (15)$$

The left-hand side of equation (15) counts the units of crop on shelf s requiring configuration k_1 or k_2 on day d . The right-hand side limits this number to the largest of the two cumulative demands (of crops growing in configurations, respectively, k_1 and k_2 on day d). For example, if $\eta_{dk_1} > \eta_{dk_2}$ and we assign all units of crop requiring configuration k_1 on day d to shelf s , then equation (15) forces all units of crop requiring configuration k_2 to be on another shelf on that day. These constraints are as strong as clique inequalities only when the right-hand side is 1.

3 Computational experiments

This section describes the computational experiments on an extensive set of benchmark instances (see Section 3.1 for a description) to compare the performance of the different models (Section 3.2) and to investigate the structure of the solutions obtained with different objective functions (Section 3.3).

3.1 Benchmark instances and computational environment

We created a set of benchmark instances based on confidential real-life data, which we published online (see [10]). The base data contains information about crops and characteristics of commercial VF cabinets, and also includes historical demand data. It assumes three types of cabinets containing different numbers of stacked shelves: small ones with seven shelves, medium ones with nine shelves, and large ones with twelve shelves. The shelves can be of two types (short or tall), and Table 1 shows how shelves are distributed in the cabinets. Note that we do not report the capacity of the shelves because it varies with the configuration used (more concretely, it depends on the growth medium).

We have data about six crops, which we denote using letters from *A* to *F*. Crop *A* requires tall shelves, while the other crops can grow on both short and tall shelves. We handle this crop-to-shelf compatibility via parameters δ_{ct} introduced in Section 2. The crops we consider need between 15 and 64 days to grow, and go through two to five growth phases, with each phase corresponding to the need for a different configuration. Table 2 reports the crop data in more detail: Column *total growth time* gives the total time in days that the crop needs to stay in the cabinet (parameter γ_c). Column *#growth phases* is the number of different growth phases the crop goes through. Column *shares config with* indicates which other crops have at least one required configuration in

cabinet	#shelves	#short shelves	#tall shelves
small	7	5	2
medium	9	6	3
large	12	8	4

Table 1: Features of the three types of cabinet considered. Column *cabinet* is the cabinet type. Column *#shelves* report the total number of shelves in the cabinet, which is made up of the number of short and tall shelves, reported respectively in columns *#short shelves* and *#tall shelves*.

crop	total growth time	#growth phases	shares config with	shelf type
A	64	3	—	tall
B	15	2	C, F	short, tall
C	15	2	B, F	short, tall
D	44	5	F	short, tall
E	35	4	—	short, tall
F	35	4	B, C, D	short, tall

Table 2: Real-life data about the crops that we used as a base for instance generation. Column *crop* is the crop identifier. Column *total growth time* gives the total time in days that the crop needs to be in the cabinet (quantity γ_c). Column *#growth phases* is the number of different growth phases the crop goes through; each growth phase requires a different configuration. Column *shares config with* indicates which other crops have at least one required configuration in common with the considered crop (recall that crops which have common configurations can share the same shelf). Column *shelf type* states whether a crop can grow in any shelf or requires a tall one.

common with the considered crop (recall that crops requiring the same configuration at the same time can share a shelf). Column *shelf type* indicates whether a crop can grow in any shelf or requires a tall one.

Given this base data and historical demand patterns, we generate new instances varying the number of crops considered and the time horizon length, and using a demand multiplier to simulate different demand situations:

- We consider all possible combinations of crops, starting from instances with demand for only one crop, then considering the $\binom{6}{2}$ possible ways of selecting two crops, etc. up to instances with demand for all six crops.
- For each crop combination, we generate instances for the large, the medium, and the small cabinet.
- For each choice of crops and cabinet, we generate six instances by multiplying the base demand by a factor of 1.0, 1.2, 1.4, 1.6, 1.8, 2.0.
- For each choice of the three above parameters, we consider time horizons of 60, 80, and 100 days.

Given $2^6 - 1 = 63$ possible ways to choose a crop combination (with at least one crop), 3 cabinets, 6 demand multiplier values, and 3 time horizon lengths, we have a total of 3402 instances. Removing instances with a time horizon of 60 but containing crop *A*, which needs 64 days to grow, leaves us with 2826 instances.

While all instances are feasible for model *MinUD*, this is not true for the other models (*MinM*, *MinR* and *MinS*). Therefore, the first aim of our computational study was to determine which of the 2826 instances are feasible for the other three models (note that an instance is either feasible for all three or none of the models). We coded the models using IBM ILOG Optimization Programming Language and ran them on a cluster equipped with Intel Xeon processors at 2.4GHz, reserving four cores and 4GB of RAM for each run. We used the solver CPLEX 12.7 with a time limit of 1 hour and default settings.

At the end of the runs, we determined that 1441 instances are feasible and 1382 are provably infeasible. We were not able to establish the feasibility of the remaining 3 instances because CPLEX neither proved them infeasible nor produced a feasible solution within the time limit. Table 3 reports the relationship between the instance generation parameters and the number of feasible instances. Increasing the number of crops in tendency decreases the number of feasible instances because crop demands are independent of each other: for example, instances with six crops have roughly twice the demand of instances with three crops. Unsurprisingly, larger cabinets and lower demands lead to a higher number of feasible instances. Finally, longer time horizons correspond to fewer feasible instances because it only takes one day with a surge in demand which the system cannot accommodate to render the whole instance infeasible; a longer time horizon corresponds to more opportunities that there is one such day.

#crops		cabinet		demand mult		time horizon	
value	%feas	value	%feas	value	%feas	value	%feas
1	58.50	small	41.61	1.0	66.03	60	82.80
2	61.39	medium	50.42	1.2	64.54	80	51.94
3	54.33	large	61.04	1.4	57.96	100	34.48
4	41.90			1.6	46.50		
5	26.07			1.8	43.74		
6	19.44			2.0	27.39		

Table 3: Percentage of feasible instances when varying each of the four instance generation parameter. Column *value* indicates the parameter value, while column *%feas* reports the number of feasible instances in percent.

3.2 Performance of the models

In this section, we investigate the performance of each of the four models and analyse the effect of the instance parameters (number of crops, cabinet type, demand multiplier, time horizon) on this performance. Table 4 reports the results of the models based on the set of 1441 instances that we proved to be feasible for all models. The five parts of the table show the results in aggregated form (grouped according to different settings of the instance parameters and as overall aggregate). For each value of instance parameter, column *instances* reports the number of instances aggregated in the row. For each model, column *%feas* reports the percentage of instances for which the respective model found a feasible solution within the 1-hour time limit, column *%opt* the percentage of instances solved to optimality, column *%gap* the average optimality gap in percent, and column *time* the average runtime of CPLEX in seconds. The last two columns are computed based on the instances with a feasible solution found within the time limit.

Of the stricter models, i.e., those models in which demand has to be met, *MinM* shows the worst performance. It provides the fewest instances with a feasible solution (in fact, for the large majority of instances, CPLEX cannot produce a feasible solution within the time limit), the largest optimality gaps, and the highest runtimes. This is due to the fact that, in this model, we cannot aggregate shelves into shelf types, leading to considerably more variables and constraints. A higher number of crops (beyond four), larger cabinets, higher demands, and longer time horizons have a clear negative effect on the ability of the model to find feasible solutions. The effect of the instance parameters on solution quality is rather small and not always monotonous, which can be explained by the generally weak performance of the model on nearly all instances.

With regards to finding feasible solutions, we observe a clear improvement for model *MinR* and again for *MinS*. However, even for *MinS*, instances exist for which no feasible solution can be determined within the time limit. This highlights the necessity of alternative solution approaches to provide the planner with at least one feasible solution to implement. While the instance parameters have only a very slight impact on the feasibility of solutions for *MinS*, for *MinR*, a higher number of crops, larger cabinets, and higher demands clearly increase feasibility (the effect of longer time horizons is unclear). A possible explanation is that larger cabinets correspond to a larger solution space, which can be harder to explore but can make it easier to find a feasible solution.

Concerning solution quality, *MinR* finds more optimal solutions within clearly shorter runtimes than *MinS* while average gaps are slightly larger. The performance difference between these two models strongly depends on instance parameters. A higher number of crops clearly improves solution quality for the former model while strongly decreasing it for the latter. This is due to the poor behaviour of variables y in the linear relaxation of all models, which is caused by “big-M” constraints (2). This behaviour is exacerbated when shelves operate near full capacity and few shelves can be empty, e.g., when considering a large number of crops or longer time horizons. In this case, many y variables will be non-zero and will take fractional values close to, but strictly less than one in the linear relaxation, thus requiring more branching. In model *MinS*, the objective function indirectly penalises variables y through variables v , which therefore tend to assume fractional values more often, leading to worse performance. Previously removed: Larger cabinets have a slightly

parameter		instances			MinM			MinR			MinS			MinUD				
name	value	%feas	%opt	%gap	time	%feas	%opt	%gap	time	%feas	%opt	%gap	time	%feas	%opt	%gap	time	
#crops	1	178	28.65	15.17	28.69	1899.30	61.80	56.74	6.80	11.99	100.00	100.00	0.00	5.89	100.00	100.00	0.00	0.71
	2	442	37.56	2.71	79.40	3400.08	75.11	69.00	5.21	72.49	98.42	82.58	0.37	714.17	100.00	99.32	0.09	28.76
	3	489	40.70	0.20	95.33	3600.00	74.85	71.17	2.30	172.69	98.98	45.40	2.68	2141.60	100.00	97.96	0.24	93.58
	4	264	40.91	0.00	98.35	3600.00	75.00	72.35	0.94	209.26	97.73	12.50	7.06	3264.14	100.00	96.97	0.29	157.99
	5	61	22.95	0.00	96.69	3600.00	85.25	81.97	1.13	417.50	98.36	1.64	10.76	3589.91	100.00	93.44	0.56	387.86
	6	7	0.00	0.00	—	3600.00	100.00	85.71	3.57	753.95	100.00	0.00	15.96	3600.00	100.00	85.71	1.19	558.28
cabinet	small	391	41.43	4.35	75.60	3273.87	70.33	67.52	1.99	18.29	98.72	59.59	3.12	1591.38	100.00	99.49	0.08	32.86
	medium	475	42.74	2.95	87.15	3393.94	72.42	68.00	5.60	53.88	98.53	53.89	3.00	1767.17	100.00	98.95	0.13	57.25
	large	575	30.09	1.57	90.46	3497.95	77.57	72.00	2.49	299.21	98.78	53.91	2.52	1742.44	100.00	96.70	0.33	152.76
demand mult	1.0	311	50.80	3.54	86.58	3399.20	63.02	60.13	2.82	147.99	98.39	40.51	6.26	2250.36	100.00	98.07	0.19	96.42
	1.2	304	46.71	3.29	86.16	3429.93	65.79	61.18	4.92	166.05	97.37	48.68	3.52	1916.93	100.00	97.37	0.28	120.99
	1.4	273	38.10	1.83	88.00	3467.99	75.09	71.43	3.76	94.55	98.90	56.04	2.12	1729.95	100.00	98.53	0.15	84.04
	1.6	218	27.52	1.83	83.51	3393.07	82.11	78.44	2.61	140.36	100.00	62.39	1.41	1467.36	100.00	99.54	0.07	43.78
	1.8	206	23.30	3.88	75.83	3106.17	81.07	74.27	3.19	157.94	99.51	65.53	0.80	1360.96	100.00	96.60	0.37	134.90
	2.0	129	20.16	1.55	72.06	3346.49	91.47	84.50	2.37	202.66	98.45	78.29	0.29	858.51	100.00	100.00	0.00	6.46
time horizon	60	461	42.52	2.60	86.56	3432.42	79.39	78.09	1.14	43.57	99.78	74.40	1.45	1051.89	100.00	98.92	0.13	56.38
	80	589	39.05	3.57	84.38	3340.80	70.63	65.87	4.14	119.48	98.30	53.99	2.78	1771.87	100.00	97.45	0.26	108.92
	100	391	28.64	1.79	82.28	3422.71	72.38	64.71	5.11	322.84	97.95	35.29	4.60	2405.31	100.00	98.47	0.17	96.49
Overall		1441	37.34	2.78	84.74	3391.23	73.91	69.47	3.37	147.43	98.68	55.45	2.84	1709.58	100.00	98.20	0.19	88.74

Table 4: Performance comparison of the four models for varying instance parameters. Column *instances* reports how many instances, out of those identified as feasible in Section 3.1, correspond to the setting specified under *parameter*. Column *%feas* indicates the number of instances (in percent) for which the model produced a feasible solution. Column *%opt* lists the number of instances solved to optimality in percent. Column *%gap* reports the average optimality gap in percent, and column *time* is the average solution time in seconds. The last two columns are computed based on the instances with a feasible solution found within the time limit.

positive effect on *MinR* and a slightly negative effect on *MinS*; higher demands increase solution quality for both models.

Contrary to these observations, the less strict model *MinUD* always produces feasible solutions with ease (for example, a solution not growing anything is feasible). It also shows the highest percentage of instances solved to optimality and the lowest average gaps, confirming that solving the model with a commercial solver is a valid approach to the problem when minimising unmet demand. A higher number of crops and larger cabinets decrease solution quality while the effect of higher demands and longer time horizons is small and unclear.

3.3 Effect of different objective functions

In this section, we investigate the impact of optimising with regards to a certain objective function on the structure of the resulting solutions. More precisely, we want to see to what extent the characteristics stipulated by the other objective functions can already be witnessed in the solutions, i.e., whether the studied objectives are synergic or contradictory. To this end, we recorded the number of crop movements, reconfigurations, and used shelves in the solutions of all models. Figures 4a to 4c report the information about, respectively, the number of reconfigurations per shelf, the number of movements per unit of crop, and the fraction of shelves used in the solutions found by the different models.

In all the figures, we note that only the model minimising the given objective achieves satisfactorily results. In Figure 4a, the models which do not minimise the number of reconfiguration give solutions with a median of up to twenty reconfigurations per shelf and model; *MinS* even has outliers with up to 60 reconfigurations per shelf. Intuitively, indeed, if a planner wants to minimise the number of used shelves, he has to juggle as many configurations as possible in the few active shelves, which makes these two objectives contradictory. We can observe the same effect in Figure 4b: when minimising the number of shelves the planner has to frequently move around crops. Finally, Figure 4c shows that when shelf minimisation is not explicitly demanded by the objective function, the models try to use as many shelves as possible: all models except *MinS* have median shelf usage of 100%. To summarise, a multi-objective approach seems very interesting for a practitioner who wants to achieve different desirable objectives simultaneously because he cannot rely on the assumption that optimising with regards to one objective will produce solutions of acceptable quality with respect to any other objective.

4 Conclusions

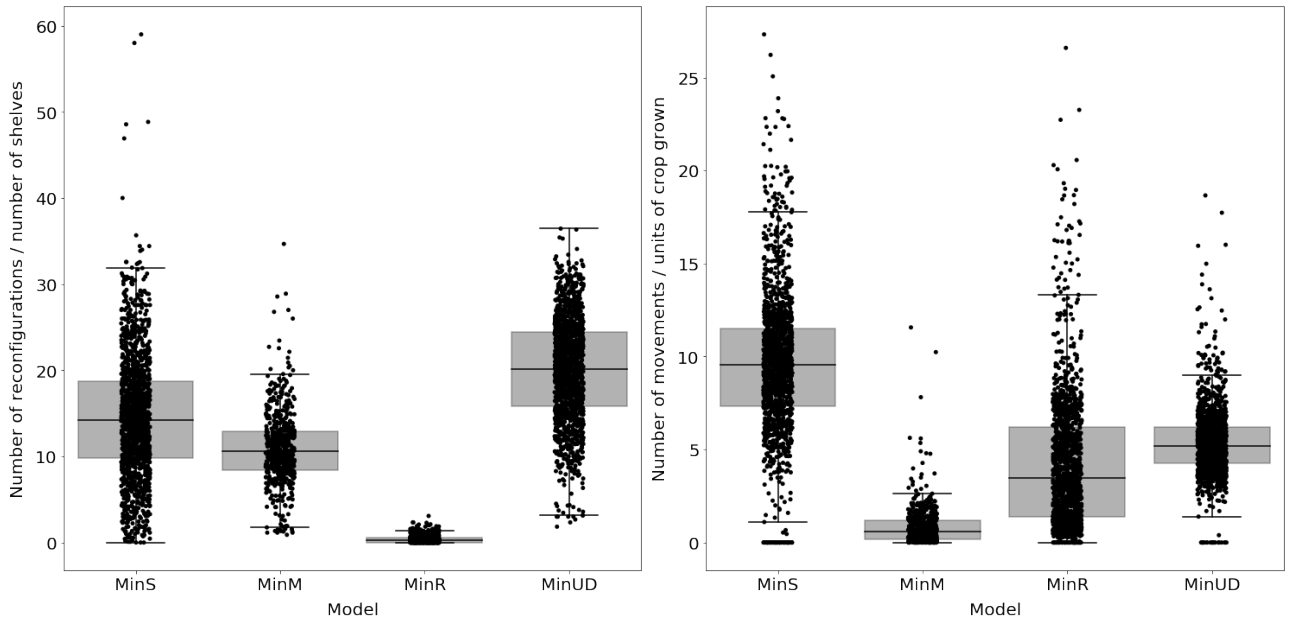
In this paper, we present four mathematical models for scheduling crops in a vertical farming system, which are strengthened using variable fixing and valid inequalities. The models mainly differ in their objective to minimise, respectively, movements, reconfigurations, shelf usage, or unmet demand. Numerical experiments on a large set of benchmark instances based on real-world data show that the performance of the models when solved with a standard solver is diverse and strongly dependent on instance data. We can conclude that using the model as practical decision support is only recommendable for the objective of minimising unmet demand. Developing suitable algorithms for the other objectives is an interesting area for future research. We also find that none of the objectives steers the solutions to be acceptable with regards to any of the other objectives, which makes multi-objective optimisation techniques another fruitful avenue for future investigations.

Acknowledgements

The authors are extremely grateful to Julia Bennell, Toni Martinez, and Chris Potts for sharing with us the slides of their presentation at the MIC'17 conference [3].

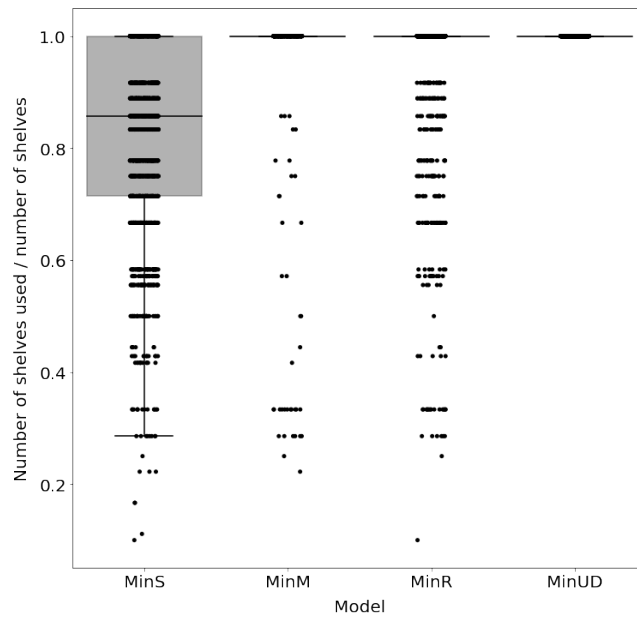
References

- [1] Andrew Beacham, Laura Vickers, and James Monaghan. "Vertical farming: a summary of approaches to growing skywards". In: *The Journal of Horticultural Science and Biotechnology* 94.3 (2019), pp. 277–283. DOI: [10.1080/14620316.2019.1574214](https://doi.org/10.1080/14620316.2019.1574214).



(a) Number of reconfigurations per shelf.

(b) Number of crop movements per unit of crop grown in the system.



(c) Share of shelves used.

Figure 4: Each box spans the second and the third quartiles, with whiskers extending to the rest of the distribution (excluding outliers). The horizontal black line inside each box depicts the median. Each dot represents one instance.

- [2] Kurt Benke and Bruce Tomkins. “Future food-production systems: vertical farming and controlled-environment agriculture”. In: *Sustainability: Science, Practice and Policy* 13.1 (2017), pp. 13–26. DOI: [10.1080/15487733.2017.1394054](https://doi.org/10.1080/15487733.2017.1394054).
- [3] Julia Bennell, Toni Martinez, and Chris Potts. “Scheduling for the growing of crops to meet demand”. In: *MIC’17. Session T2.2: Scheduling*. Proceeding of the 12th Metaheuristics International Conference (Universitat Pompeu Fabra, July 4–7, 2017). Barcelona, 2017. URL: <https://easychair.org/smart-program/MIC’2017/2017-07-06.html#talk:46211>.
- [4] Nina Fedoroff. “Food in a future of 10 billion”. In: *Agriculture & Food Security* 4.1 (2015), pp. 1–10. DOI: [10.1186/s40066-015-0031-7](https://doi.org/10.1186/s40066-015-0031-7).
- [5] Nyoman Gunantara. “A review of multi-objective optimization: methods and its applications”. In: *Cogent Engineering* 5.1 (2018). DOI: [10.1080/23311916.2018.1502242](https://doi.org/10.1080/23311916.2018.1502242).
- [6] Deb Kalyanmoy. “Multi-objective optimization”. In: *Search methodologies*. Ed. by Edmund Burke and Graham Kendall. 2014, pp. 403–449. DOI: [10.1007/978-1-4614-6940-7_15](https://doi.org/10.1007/978-1-4614-6940-7_15).
- [7] Toyoki Kozai. *Smart plant factory: the next generation indoor vertical farms*. 2018. DOI: [10.1007/978-981-13-1065-2](https://doi.org/10.1007/978-981-13-1065-2).
- [8] Toyoki Kozai, Genhua Niu, and Michiko Takagaki. *Plant factory: an indoor vertical farming system for efficient quality food production*. 2015. DOI: [10.1016/C2014-0-01039-8](https://doi.org/10.1016/C2014-0-01039-8).
- [9] Chris N Potts and Mikhail Y Kovalyov. “Scheduling with batching: a review”. In: *European journal of operational research* 120.2 (2000), pp. 228–249. DOI: [10.1016/S0377-2217\(99\)00153-8](https://doi.org/10.1016/S0377-2217(99)00153-8).
- [10] Alberto Santini. *Instances for the crop growth scheduling problem in vertical farming*. 2020. DOI: [10.5281/zenodo.3759681](https://doi.org/10.5281/zenodo.3759681). URL: <https://github.com/alberto-santini/crop-growth-scheduling-vf>.
- [11] UN General Assembly. *Transforming our world: the 2030 agenda for sustainable development*. Tech. rep. A/RES/70/1. United Nations, Oct. 21, 2015. URL: <https://www.refworld.org/docid/57b6e3e44.html>.