Expert Systems With Applications 162 (2020) 113792

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A framework for multi-objective optimization of virtual tree pruning based on growth simulation

Damjan Strnad [a,*], Štefan Kohek [a], Bedrich Benes [b], Simon Kolmanič [a], Borut Žalik [a]

[a] Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia
[b] College of Science, Purdue University, 150 N. University St, West Lafayette, IN 47907, USA

ARTICLE INFO

Keywords:
Virtual tree pruning
Multi-objective optimization
Growth simulation
Simulated annealing
NSGA-II

ABSTRACT

We present a framework for multi-objective optimization of fruit tree pruning within a simulated environment, where pruning is performed on a virtual tree model, and its effects on tree growth are observed. The proposed framework uses quantitative measures to express the short-term and long-term effects of pruning, for which potentially conflicting optimization objectives can be defined. The short-term objectives are evaluated on the pruned tree model directly, while the values of long-term objectives are estimated by executing a tree growth simulation. We demonstrate the concept by using a bi-objective case, where the estimated light interceptions of the pruned tree in the current and the next season are used to define separate optimization objectives. We compare the performance of the multi-objective simulated annealing and the NSGA-II method in building the sets of non-dominated pruning solutions. The obtained Pareto front approximations correspond to diverse pruning solutions that balance between optimizing either objective to different extents, which indicates a potential for new applications of the multi-objective pruning optimization concept.

1. Introduction

Fruit tree pruning is an important horticultural process that improves the light conditions for fruit development within the crown, and regulates the vegetative growth of a tree. Pruning helps to establish the balance between the fruit quality and quantity through consecutive harvesting seasons (Ikinci, Kuden, & Ak, 2014; Olesen, Menzel, McConchie, & Wiltshire, 2013; Villasante, Godoy, Zoffoli, & Ayala, 2012). The absence of proper tree pruning makes trees susceptible to diseases, may reduce crop quality, or pronounce irregularity of yield (alternate bearing) (Ersin, 2017; Giulivo, 2011; Maggs, 1963; Mohammadi, Mahmoudi, & Rezaee, 2013). Pruning is, therefore, an essential skill for fruit growers.

Recently, computer simulations have been developed for modeling physiological responses of trees to different horticultural practices, based on the source–sink model of resource exchange (Allen, Prusinkiewicz, & DeJong, 2005; Balandier, Lacointe, Le Roux, Sinoquet, Cruiziat et al., 2000; Fišer, Ravi, Benes, Shi, & Hirst, 2015; Kang, Fišer, Shi, Sheibani, Hirst et al., 2016; Lescourret, Moitrier, Valsesia, & Génard, 2011; Xia, Li, & Huang, 2009). Various simulated environments with integrated 3D visualization for support in teaching of tree training techniques have been introduced, allowing the user to manipulate a virtual tree in an interactive manner (Cokelaer, Fumey, Guédon, & Godin, 2010; Kohek, Guid, Tojnko, Unuk, & Kolmanič, 2015; Lang, & Lang, 2007; Lopez, Favreau, Smith, & DeJong, 2010; Yang, Chen, Hua, Kang, & Dong, 2015). Such tools complement field training, as they enable the user to observe and compare the effects of various actions on a tree interactively.

The EduApple virtual tree pruning tool (Kohek et al., 2015; Kolmanič, Tojnko, Unuk, & Kohek, 2017) allows active training support by guiding the user's actions towards a desired result concerning the selected pruning criteria, such as the lighting conditions within the crown (Strnad, & Kohek, 2017). The algorithmic pruning optimization in EduApple enables active user guidance during the educational use of the tool, but can also be used to pursue and analyze new pruning techniques in a virtual environment. However, the existing pruning optimization considers only the immediate gains of pruning, expressed as a single quantitative objective, which is not necessarily aligned with the long-term goals of pruning.

In this paper, we present a conceptual framework for multi-objective pruning optimization that considers both immediately observable and delayed pruning effects. This study aims to assess the efficiency of multi-objective optimization (MO) methods in finding the sets of non-dominated pruning solutions, and to indicate new opportunities for educational and analytical use of existing tools. Our approach uses tree

* Corresponding author.
E-mail addresses: damjan.strnad@um.si (D. Strnad), stefan.kohek@um.si (Š. Kohek), bbenes@purdue.edu (B. Benes), simon.kolmanic@um.si (S. Kolmanič), borut.zalik@um.si (B. Žalik).

https://doi.org/10.1016/j.eswa.2020.113792
Received 13 January 2020; Received in revised form 9 June 2020; Accepted 23 July 2020
Available online 28 July 2020

growth simulation to perform stochastic sampling of future pruned tree development, and builds a set of non-dominated pruning solutions with respect to the pre-growth and post-growth objectives. We demonstrate the concept using a bi-objective case, where the estimated light intake of the pruned tree is used to define the evaluation criteria. The methodological contributions of this paper are:

1. introduction of heuristic constraints into the optimization procedure, which reduces the dimension of the search space,
2. sampling of the stochastic growth simulation model for evaluating the delayed effects of pruning, and
3. introduction of multi-objective pruning optimization that constructs a set of non-dominated solutions with respect to the pre-growth and post-growth objective values.

2. Related work

Simulation of tree response to manipulation has been modeled to a varying degree of detail in previous studies. A model of carbon assimilation, distribution, and storage in peach trees is presented in L-PEACH (Allen et al., 2005; Lopez et al., 2010). Interactive visualization allows user intervention (fruit thinning and pruning) using a daily time step. Different components of the model control tree architecture (implemented using L-systems), movement of carbon (using an electric circuit analogy), and plant organ functionality.

SIMWAL is a detailed structural–functional model of a walnut tree with simulation of climate conditions and reactions to pruning (Balandier et al., 2000). It uses separate sub-models to compute light interception, photosynthate production, and partitioning of resources among growth, respiration, and reserve demand. The architectural submodel in SIMWAL determines bud break probabilities and the extent of new growth as a response to pruning.

IMapple (Fišer et al., 2015; Kang et al., 2016) is a structural–functional model for the Golden Delicious apple tree, which uses precise geometric representation to incorporate direct illumination and self-shadowing in the computation of leaf irradiances. The simulation of resource transportation is based on the same model that is used in L-PEACH. User actions on the tree model are not considered in IMapple.

VCHERRY (Lang & Lang, 2007) is an application for predicting the short-term and long-term effects of pruning decisions on the yield and fruiting of cherry trees. The model allows simulation of multi-season tree development under varying pruning regimes. The amount of vegetative and reproductive growth can be modulated by selection of different rootstocks and orchard/climate parameters.

Lescourret et al. introduced QualiTree (2011), a generic fruit tree model for simulating the effects of different cultivation practices on the development and quality of the fruit. The model considers carbon and water processes from the state of the bloom to the end of fruit development, and produces the estimated distributions of different fruit quality parameters.

A model of branching responses to different pruning schemes in apple trees, based on a hidden semi-Markov chain model of different types of bud formation, was presented by Xia et al. (2009). The educational goal of the developed application is similar to that of EduApple.

Existing work does not address optimization of virtual tree pruning with respect to a given set of quantifiable goals, which are often contraindicating due to complex interactions implemented in the simulation models. In this paper, a framework for such multi-objective pruning optimization is presented. The concept of pruning optimization was used previously in automated rule-based pruning, but the goal there was to optimize the parameters of the pruning branch identification method in order to achieve the target pruning branch proportion (Karkee, Adhikari, Amatya, & Zhang, 2014).

3. Background

In this section, an overview of the tree representation and growth simulation in EduApple is presented first, followed by a formal definition of the pruning optimization problem.

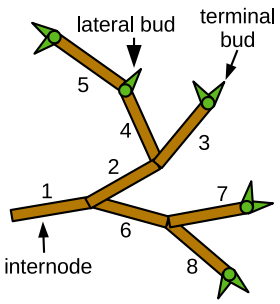

Fig. 1. Elements of tree structure with depth-first internode ordering.

3.1. The EduApple tree growth and training simulator

EduApple (Kolmanič et al., 2017) is a software tool for computer-aided support in the horticultural education of tree training techniques. It allows the user to perform actions on a virtual tree model and observe its effects on subsequent tree growth and development immediately. It can be used to supplement field practices, with the advantage of providing instant feedback and allowing an unlimited number of harmless trial-and-error learning cycles. To this end, the tool uses an integrated tree growth model, which is parameterized to allow simulation of different behaviors. In order to demonstrate the concept of multi-objective pruning optimization, the EduApple model has been extended with an empirical model of resource allocation for reproductive and vegetative growth.

The tree is modeled as a set of chains of connected branch segments (Fig. 1). The basic building block of a tree skeleton is a *metamer*, which is also the smallest unit of growth. The metamer consists of an *internode* and two buds, the terminal and the lateral bud, which expand into new branch segments upon growth. Thickness, age, and the number of descendants are registered and stored for each internode. Internodes represent potential locations of pruning cuts, and are indexed uniquely by using a depth-first ordering. Buds are labeled in the same way. We denote the set of all internode indices of a tree model $\mathcal{T}$ by $\mathcal{I}^{\mathcal{T}}$ and the set of its bud indices by $\mathcal{B}^{\mathcal{T}}$.

The tree growth is simulated in discrete seasonal steps. The extent of growth is determined by the amount of accumulated resources R, which is calculated for a tree of age A as:

$$R = C_1 \min\{A, 12\} + \left(C_2 \tanh\left(\frac{0.2}{A}\right) + 2\right) \sum_{i \in \mathcal{B}} q_i. \tag{1}$$

The factors C_1 and C_2 are model parameters. The first term on the right-hand side of Eq. (1) approximates the fixed resource storage of a tree, while the second one represents the photosynthetic product, compiled from aggregated bud illuminances q_i. The efficiency for utilizing a photosynthesis product is modeled using a decreasing hyperbolic tangent function to reduce tree growth with age (Kohek et al., 2015).

The light exposure of each bud i is calculated by reducing the full exposure value $q_i = 1$ in proportion to the estimated shadow which is cast on the bud by the upper tree parts (Benes, 1996; Měch, & Prusinkiewicz, 1996; Pałubicki et al., 2009). To simplify the calculation, the shadow cast by a metamer is approximated by a downward expanding conical shadow volume (Kohek et al., 2015).

An empirical model of resource competition for reproductive and vegetative growth was introduced in EduApple, and used for the demonstration of the multi-objective pruning concept in this study. It models resource allocation on a high level to simulate the disrupted balance of reproductive and vegetative growth, which can occur in unmanaged trees. The model is based on the assumptions of the allocational theory that states that the resource supply is fixed, and its allocation among competing plant functions is mutually exclusive (Bazzaz, Ackerly, & Reekie, 2000). The resource allocation model

is parameterized to allow simulating different behaviors. The amount of resources r_f for reproductive growth is determined by allocating a constant amount C_3 for each of N_f flower buds:

$$r_f = N_f(C_3 - C_4 \cdot A). \tag{2}$$

The depletion of tree resources by reproductive growth is reduced by a constant yearly amount of C_4 per flower bud, which allows modeling of the non-linear relationship between the resource allocation and the plant's vegetative size (Bazzaz et al., 2000). The values of C_3 and C_4 are adjustable model parameters. The remaining resources r_v are dedicated to vegetative growth:

$$r_v = \max\{0, R - r_f\}. \tag{3}$$

The growth proceeds by redistributing the vegetative resources back to the buds. The amount of resources r_i received by each vegetative (i.e., non-flowering) bud $i \in \mathcal{B}_{veg}$ is, in part, proportional to its light exposure. In this way, the tree expands primarily in the parts that are well lit, which is an important element of tree self-organization (Pałubicki et al., 2009). The resource allocation is affected additionally by the topological and geometrical properties of the branch at the bud position, such that upward oriented buds with shorter topological distance to the roots receive an increased share of the resources (Kohek et al., 2015).

The last step of growth is stochastic bud shooting, in which a bud i can grow a sequence of $\lfloor r_i \rfloor$ metamers in a single cycle. Only buds that gather enough resources to develop at least one metamer can eventually grow, with different probabilities of shooting for one-year-old terminal buds, one-year-old lateral buds, and older buds. The growing buds are replaced by a corresponding number of new branch segments. The differentiation of flowering and vegetative buds is also performed at their creation, with a bud becoming a flowering one with probability $0.02 \leq p_f \leq 0.05$.

3.2. Pruning as a multi-objective optimization problem

The goal of pruning is to improve the light conditions within the tree crown (immediate, short-term effect), and to maintain the balance of reproductive and vegetative growth (delayed, long-term effect). Both of these conditions are disrupted in unmanaged or poorly managed trees, where pruning is performed as a corrective measure. In this paper, the concept of multi-objective pruning optimization is demonstrated using an illustrative case, where the achievement of both pruning goals is estimated by calculating the tree's light interception before and after the growth. The pre-growth illumination of pruned tree's flower buds is used to estimate the light conditions for fruit development within the crown, while the post-growth illumination of buds is used to estimate the extent of vegetative growth, which will provide the capacity for the next season's production. For this purpose, we define the *light intake* $F(\mathcal{T})$ of a given tree model $\mathcal{T}$ as a function of its flower buds' light exposures:

$$F(\mathcal{T}) = \sum_{i \in B_f^{\mathcal{T}}} q_i^2 \tag{4}$$

Here, $B_f^{\mathcal{T}} \subset B^{\mathcal{T}}$ denotes the set of flower buds. The light intake in Eq. (4) is proportional to both the number and the illumination of flower buds, thereby combining the quantitative and qualitative criteria of fruit production. The use of squared light values emphasizes the preference for higher illuminations of the buds.

The light intake objective is a proxy measure for estimating the balance of vegetative and reproductive growth achieved by pruning. There are two complementary effects of pruning that influence the objective values: the first effect is a direct melioration of buds' light exposures, while the second one is the regulation of vegetative growth, which provides the basis for the development of flower buds in the next season. After pruning, the immediate objective value decreases due to the removal of flower buds, although the loss is partially compensated by improved light conditions of the remaining buds. At the same time, the removal of buds also reduces the amount of accumulated resources, but allocates more resources to the vegetative growth of new wood, which increases the future objective value. By using the pre-growth and post-growth light intake values as separate objectives to be maximized, we obtain a multi-objective pruning optimization problem. The interplay of potentially contraindicating effects of pruning on the immediate and delayed objective values presents the non-trivial element of the problem, and indicates the existence of multiple non-dominated pruning solutions (i.e., a non-dominated solution set (Collette, & Siarry, 2003)).

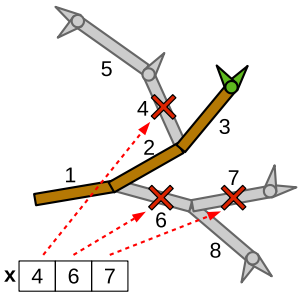

Fig. 2. Correspondence between a pruning vector **x** and a pruned tree model.

We represent the tree pruning as a sequence of cuts, where each item in the sequence is a cut location that corresponds to one of the tree's internodes. Any pruning can be described formally by a vector of internode indices

$$\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle, x_i \in I^{\mathcal{T}}.$$

The list of potential cut locations can be reduced by using additional pruning constraints (Section 4.1). Fig. 2 shows an example of a mapping between the cut sequence and the corresponding pruned tree model.

The same pruning can correspond to many different cut sequences, because one or more cuts in the pruning description can be redundant. This happens when an internode and any of its descendants are included in the cut set, because the effect of a subcut on the same branch is neutralized by the corresponding supercut. A situation of this sort is depicted in Fig. 2 for cuts 6 and 7. Such redundancy in pruning representation is a useful property for pruning optimization, because individual cuts can be deactivated and reactivated by the search procedure, resulting in the dynamic size of the pruning solution (i.e., the number of effective cuts).

4. Methodology

In this section, we describe a general multi-objective pruning optimization framework with in-the-loop growth simulation for evaluating the accomplishment of short-term and long-term pruning objectives. We exemplify the concept with a bi-objective case of maximizing the two-year light intake (Section 3.2), where two different methods are used for optimization. The first one is a multi-objective version of simulated annealing, which uses a local search heuristic to construct a set of non-dominated pruning solutions. The second one is a popular MO method called NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002).

4.1. The pruning heuristics for constrained optimization

Unconstrained pruning optimization considers every internode as a potential cut location, which results in a vast number of possible solutions. Introducing heuristic constraint increases search efficiency by reducing the size of the solution space. We propose the following heuristics to constrain the optimization:

- cut locations are limited to branches from a specified age span $[a_{\min}, a_{\max}]$,
- only cuts that result in a pruned mass (i.e., the number of removed internodes and buds) above some threshold $m_{\min}$ are considered,
- the number of cuts in a single pruning is constrained to a range $[d_{\min}, d_{\max}]$, and
- pruning of a branch is performed only immediately after the fork (i.e., on the first internode of the main branch extension or the lateral branch).

Some of the proposed heuristics mimic aspects of practical pruning; they can, for example, prevent pruning of main branches, which are important for tree structure, or promote thinning cuts at branch bases (Harris, 1994). Setting the minimum pruning mass of a cut is an ad hoc heuristic that reduces the search space by excluding cuts that do not contribute significantly to the solution. The limited number of cuts is both a practical constraint, because it prevents inflicting too much damage to a tree, and an efficient one, because large solution vectors often contain a lot of redundancy.

The first three constraints are parameterized by numerical bounds, while the last one is a binary constraint. By setting the appropriate parameter values, the user can select an arbitrary combination of the above constraints. The constraints are used in the search initialization phase to filter out the internodes when constructing the list of possible cut locations (see Section 4.2), and to limit the size of the solution vectors during optimization. The adherence of solution vectors to the constraints is ensured by limiting all perturbations of pruning vectors to the list of allowed cut locations, and by preventing explicitly the change of a solution vector that would result in its size being outside of the prescribed bounds.

4.2. The proposed optimization framework

Fig. 3 shows the overview of the proposed pruning optimization framework, which links the procedures in the context of bi-objective light intake maximization. The framework consists of an initialization phase, a solution evaluation phase, and an optimization phase. The input to the algorithm is the tree model description (Section 3.1). The output of the procedure is the set of non-dominated pruning solutions, each described by a vector of cut locations (Section 3.2) and the corresponding multi-objective value vector.

In the initialization phase, the input tree model $\mathcal{T}$ is first screened to produce the list $\mathcal{I}^{\mathcal{T}}_{\text{cut}}$ of candidate cut locations, which is a subset of all internodes $\mathcal{I}^{\mathcal{T}}$, filtered with respect to the employed pruning constraints (Section 4.1). The set of non-dominated pruning solutions is initially set to an empty set. The optimization starts from a randomly initialized population of pruning solution vectors $\mathcal{P}^{(0)} = \{\mathbf{x}_i^{(0)}\}$, which are of size $d = \lfloor (d_{\min} + d_{\max})/2 \rfloor$. The initial pruning vectors are generated by sampling d-times randomly from $\mathcal{I}^{\mathcal{T}}_{\text{cut}}$. The initial population is then set as the current candidate population, and the procedure enters the evaluation phase.

During the evaluation phase, the tree $\mathcal{T}$ is pruned separately according to each pruning vector $\mathbf{x}_i^{(t)} \in \mathcal{P}^{(t)}$ in the current candidate population $\mathcal{P}^{(t)}$. The short-term objective value of $\mathbf{x}_i^{(t)}$ is next computed as the light intake of the pruned tree $\mathcal{T}_{/\mathbf{x}_i^{(t)}}$ using Eq. (4). To evaluate the expected light intake in the next season, the growth simulation is executed multiple times (the inner loop in Fig. 3), and the light intake of each grown tree is computed. The results of all simulations are averaged to obtain a more reliable estimate of the long-term objective value. The evaluation phase concludes by concatenating the short-term and long-term objective values into the bi-objective value vector (Section 4.3).

The exact steps of the optimization phase depend on the employed MO method. In general, the solution vectors of the current candidate population are first tested for inclusion in the active set of non-dominated solutions, and the non-dominated front of objective vectors

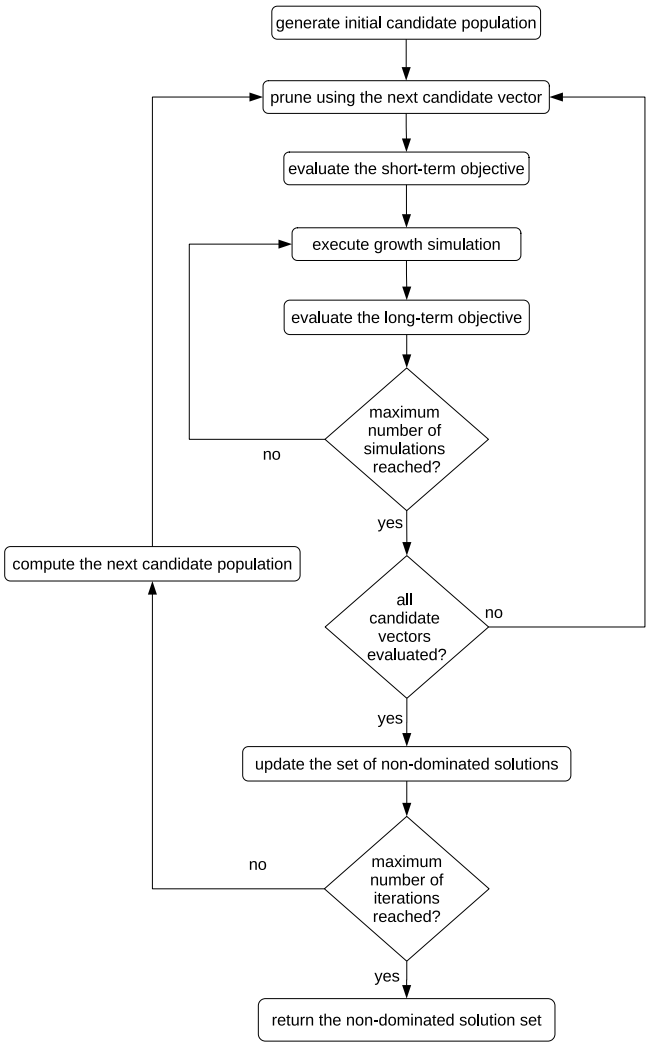

Fig. 3. Flowchart of the proposed pruning optimization framework.

is updated accordingly. A selection process is next performed on the current candidate population to determine its active subset. Unless the prescribed number of optimization iterations has been reached, the active subset is modified and repopulated according to the rules of the optimization method. In this way, the next candidate population $\mathcal{P}^{(t+1)}$ is obtained, and the procedure loops back to the evaluation phase. Upon the outer loop termination, the final non-dominated set is returned as a result.

4.3. Multi-objective evaluation function

The value of a pruning solution is determined by two objectives, expressed by the light intake of the pruned tree before and after growth. The short-term objective value is calculated as the light intake of the pruned tree by evaluating Eq. (4) directly. The long-term objective value for the pruned tree is estimated by executing the tree growth simulation and evaluating the objective function of the grown tree. Because the growth model is stochastic, the future light intake is averaged over multiple growth runs to obtain a more reliable estimation of the true objective value. We used 20 runs in our experiments. The principle of multi-objective function evaluation for some given solution vector $\mathbf{x}$ is illustrated in Fig. 4.

We define the bi-objective value $\mathbf{f}(\mathcal{T}, \mathbf{x})$ of a pruning solution $\mathbf{x}$, when applied to the tree $\mathcal{T}$, as a pair:

$$\mathbf{f}(\mathcal{T}, \mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x})\right), \tag{5}$$

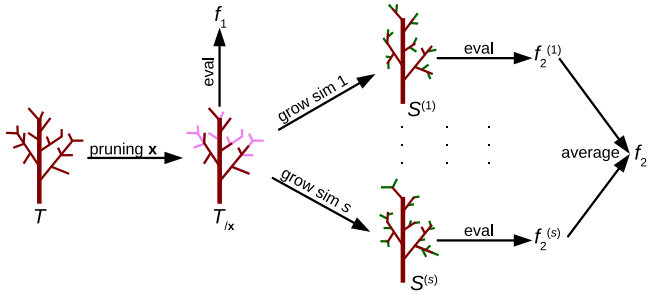

Fig. 4. The flow of multi-objective function evaluation.

where

$$f_1(\mathbf{x}) = F(\mathcal{T}_{/\mathbf{x}})$$
$$f_2(\mathbf{x}) = \langle F^*(\mathcal{T}_{/\mathbf{x}})\rangle. \tag{6}$$

Here, $\mathcal{T}_{/\mathbf{x}_t}$ denotes the tree $\mathcal{T}$ pruned according to the vector of cuts $\mathbf{x}$, while $\langle F^*\rangle$ denotes the expectation of the post-growth light intake averaged over s growth simulations:

$$\langle F^*(\mathcal{S})\rangle = \frac{1}{s}\sum_{i=1}^{s} F(\mathcal{S}^{(i)}), \tag{7}$$

where $\mathcal{S}^{(i)}$ is the tree resulting from the ith growth simulation on $\mathcal{S} = \mathcal{T}_{/\mathbf{x}}$. To reduce the variance of future light intake in Eq. (7) due to random differentiation of flower buds at growth, their values are calculated by performing the summation in Eq. (4) over all one-year old buds, and scaling the result by the flower bud probability p_f.

4.4. The pruning optimization methods

We compare the performance of two optimization methods in this study. The first method is a multi-objective version of simulated annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983). SA is simple to implement, and was used as a reference in this study because it demonstrated comparative performance to the population-based methods in single-objective pruning optimization (Strnad & Kohek, 2017). The second method is NSGA-II, a popular MO algorithm that uses non-dominated sorting of candidate solutions, and preserves population diversity by calculating their crowding distances (Deb et al., 2002).

The goal of the optimization is to build a set $\mathcal{X}$ of non-dominated pruning solutions, which is maintained separately from the active population. In the bi-objective case from Eq. (5), solution $\mathbf{x}_i$ is dominated by solution $\mathbf{x}_j$ ($\mathbf{x}_j \succ \mathbf{x}_i$), if $\mathbf{x}_j$ is at least as good as $\mathbf{x}_i$ in both objectives and strictly better than $\mathbf{x}_i$ in at least one of them:

$$\mathbf{x}_j \succ \mathbf{x}_i \Leftrightarrow \forall k \in \{1,2\} : f_k(\mathbf{x}_j) \geq f_k(\mathbf{x}_i) \wedge \exists k \in \{1,2\} : f_k(\mathbf{x}_j) > f_k(\mathbf{x}_i). \tag{8}$$

4.4.1. Multi-objective simulated annealing

Simulated annealing is a local optimization meta-heuristic that operates with a single solution vector. The adaptation of SA for the multi-objective case is presented in Algorithm 1.

The optimization starts from a random initial solution vector $\mathbf{x}$ and proceeds by modifying the solution in an attempt to improve its value. In each iteration, one or more local changes are applied to the *current* or *active solution* $\mathbf{x}$ to obtain a new *candidate solution* $\mathbf{x}'$. The candidate solution is evaluated according to the multi-objective function $\mathbf{f}$, and tested for inclusion into the active set of non-dominated solutions $\mathcal{X}$. If the candidate solution is non-dominated, the set $\mathcal{X}$ is updated by adding $\mathbf{x}'$ and removing the solutions dominated by $\mathbf{x}'$. The accepted candidate solution is also set as the new active solution. On the other hand, if the candidate solution is dominated and, thus, not added to $\mathcal{X}$, it still becomes the new active solution with probability:

$$p = e^{-\frac{\Delta\mathbf{f}}{T}}, \tag{9}$$

Algorithm 1 Multi-objective simulated annealing

Input: number of iterations N, initial temperature T_0, extra mutation rate Mr, modification probability distribution $\mathbf{P}$, number of iterations M for random restart
Output: non-dominated set of solutions $\mathcal{X}$

procedure SA($N, T_0, Mr, \mathbf{P}, M$)
  $T \leftarrow T_0$
  $\mathbf{x} \leftarrow$ random initial solution
  compute the objective value $\mathbf{f}(\mathbf{x})$
  $\mathcal{X} \leftarrow \{\mathbf{x}\}$
  $m \leftarrow 0$
  while $T > 0$ **do**
    if $m < M$ **then**
      $\mathbf{x}' \leftarrow$ modify $\mathbf{x}$ using Mr and $\mathbf{P}$
      $m \leftarrow 0$
    else
      $\mathbf{x}' \leftarrow$ random solution
    end if
    compute the objective value $\mathbf{f}(\mathbf{x}')$
    if $\nexists \mathbf{z} \in \mathcal{X} : \mathbf{z} \succ \mathbf{x}'$ **then**
      update $\mathcal{X}$ with $\mathbf{x}'$
      $\mathbf{x} \leftarrow \mathbf{x}'$
      $m \leftarrow 0$
    else
      $\mathbf{x} \leftarrow \mathbf{x}'$ with probability $e^{-\Delta\mathbf{f}/T}$
      $m \leftarrow m + 1$
    end if
    $T \leftarrow T - 1/N$
  end while
  return $\mathcal{X}$
end procedure

where T is the so-called temperature parameter, which is set to initial value T_0 and reduced slowly towards zero. The usual cooling regime is linear, such that the temperature reduction ΔT is set to $1/N$, where N is the maximum number of optimization iterations. In this study, we define $\Delta\mathbf{f}$ as the minimum L^1 distance of $\mathbf{x}'$ from any point in $\mathcal{X}$ that dominates it:

$$\Delta\mathbf{f}(\mathcal{T}, \mathbf{x}', \mathcal{X}) = \min_{\substack{\mathbf{z}\in\mathcal{X} \\ \mathbf{z}\succ\mathbf{x}'}} \left\{ \|\mathbf{f}(\mathcal{T}, \mathbf{z}) - \mathbf{f}(\mathcal{T}, \mathbf{x}')\|_1 \right\}. \tag{10}$$

Moving the search into the neighborhood of dominated solutions in this way maintains the necessary level of exploration and helps in preventing the premature stagnation of the search.

According to Eq. (9), the probability of a worse solution replacing a better one is affected by the current temperature and the amount of solution value loss $\Delta\mathbf{f}$. Lower temperatures and higher objective losses result in smaller values of p. Setting the proper value of initial temperature T_0 allows the SA to escape from local maxima in the early exploration phase, but start converging to a stable solution in the later exploitation stage. A common extension, which is also implemented in Algorithm 1, is to restart from a random candidate solution if there is no improvement of the non-dominated set for a given number M of iterations.

When modifying the active solution to produce the next candidate, the following types of "local changes" are possible:

- displacing a randomly selected cut with probability P_{move},
- adding a random cut location with probability P_{add}, and
- removing a randomly selected cut with probability P_{remove}.

All changes to the solution vectors are restricted to the list $\mathcal{I}^{\mathcal{T}}_{\text{cut}}$ of allowed cut locations, which is constructed in the initialization phase of the optimization (Section 4.2). The type of performed change is determined according to the user-defined probability distribution $\mathbf{P}$ =

$\langle P_{\text{move}}, P_{\text{add}}, P_{\text{remove}}\rangle$. By using the constraint $P_{\text{add}} = P_{\text{remove}} = (1 - P_{\text{move}})/2$, the probability distribution is defined uniquely by choosing P_{move} alone. Cut removal and addition are performed only if the vector size remains within the prescribed range $[d_{\min}, d_{\max}]$. If this is not the case, the corresponding probability is set to zero and the other two are re-normalized. After performing one of the above modifications on a pruning vector, all other cuts are replaced independently by random alternatives with probability (i.e., mutation rate) Mr, which, together with **P**, determines the exploration intensity of the search.

It should be noted that local changes in the space of the solution vectors do not necessarily correspond to local modifications of the corresponding pruned tree models. Even a small increment or decrement of a cut index can result in a wide displacement of the matching cut location, potentially followed by deactivation of previously active, or activation of previously inactive cuts. From the perspective of the search, such behavior increases the level of exploration and facilitates escaping from local optima.

4.4.2. NSGA-II

Nondominated sorting genetic algorithm (NSGA-II) is a widely used and efficient MO method (Deb et al., 2002). It integrates a mechanism for preserving the diversity of solution population and elitist selection of solutions according to their rank in a sorted list of non-dominated subsets. The particular implementation of NSGA-II, used in this study, is shown in Algorithm 2.

Algorithm 2 NSGA-II

Input: number of objective function evaluations N, population size Ps, mutation rate Mr, modification probability distribution **P**
Output: non-dominated set of solutions $\mathcal{X}$

procedure NSGA-II($N, Ps, Mr, \mathbf{P}$)
  $\mathcal{P} \leftarrow$ initial population of random vectors $\mathbf{x}_i, i = 1, \ldots, Ps$
  compute the objective values $\mathbf{f}_i(\mathbf{x}_i)$ for $\forall \mathbf{x}_i \in \mathcal{P}$
  compute the rank $rank_i$ and crowding distance d_i for $\forall \mathbf{x}_i \in \mathcal{P}$
  $\mathcal{X} \leftarrow \emptyset$
  while $N > 0$ **do**
    $\mathcal{Q} \leftarrow \emptyset$
    while $|\mathcal{Q}| < Ps$ **do**
      $\mathbf{u}_1, \mathbf{u}_2 \leftarrow$ binary tournament selection from $\mathcal{P}$
      $\mathbf{v}_1, \mathbf{v}_2 \leftarrow$ crossover of $\mathbf{u}_1$ and $\mathbf{u}_2$
      $\mathbf{w}_1, \mathbf{w}_2 \leftarrow$ mutation of $\mathbf{v}_1$ and $\mathbf{v}_2$ using Mr and **P**
      if $|\mathcal{Q}| < Ps - 1$ **then**
        compute the objective values of $\mathbf{w}_1$ and $\mathbf{w}_2$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathbf{w}_1, \mathbf{w}_2\}$
      else
        compute the objective value of $\mathbf{w}_1$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathbf{w}_1\}$
      end if
    end while
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{Q}$
    compute the rank $rank_i$ and crowding distance d_i for $\forall \mathbf{x}_i \in \mathcal{P}$
    sort $\mathcal{P}$ by increasing $rank_i$ first, then by decreasing d_i
    $\mathcal{P} \leftarrow$ first Ps vectors from sorted $\mathcal{P}$
    update $\mathcal{X}$ using $\mathcal{P}$
    $N \leftarrow N - Ps$
  end while
  return $\mathcal{X}$
end procedure

The implementation uses three adjustable hyperparameters: the population size Ps, the mutation rate Mr, and the probability distribution $\mathbf{P} = \langle P_{\text{move}}, P_{\text{add}}, P_{\text{remove}}\rangle$. The mutation of a pruning vector in NSGA-II is performed in the same way as in SA, using **P** to select one of the three modifications that are applied, and displacing each of the remaining cuts randomly with probability Mr.

Table 1
Properties of tree models from Fig. 5 and the corresponding pruning scenarios.

Tree model	# of internodes	# of potential cut locations	# of cuts	Search space size
Fig. 5(a)	2325	376	15–35	2.85×10^{49}
Fig. 5(b)	3739	564	15–35	6.96×10^{55}
Fig. 5(c)	5273	581	15–35	2.03×10^{56}
Fig. 5(d)	2895	499	15–35	8.37×10^{53}
Fig. 5(e)	2348	289	15–35	1.77×10^{45}
Fig. 5(f)	1082	74	5–25	6.69×10^{19}
Fig. 5(g)	690	57	5–15	3.31×10^{13}
Fig. 5(h)	1824	78	5–25	3.07×10^{20}

5. Results and discussion

The experiments were carried out to answer the following research questions: (1) What are the characteristics of the non-dominated sets constructed by the proposed pruning optimization methods? (2) What is the relation of "no-pruning solution" and rule-based pruning to the solutions in the non-dominated set? and (3) How do different values of optimization meta-parameters affect the optimization?

5.1. Experimental setup

The experiment was performed on apple tree models of varying complexities. The test tree models were generated using EduApple by running from 5 to 10 growth simulation cycles until the onset of seasons with alternating low and high values of objective values, indicating the imbalance of reproductive and vegetative growth. Manual pruning of varying intensity was performed in the initial or intermediate seasons for some models to obtain trees with diverse structure and distribution of young vs. old wood. The tree models used in the experiments are shown in Fig. 5.

We have used the configuration $C_1 = 150, C_2 = 40, C_3 = 80$, and $C_4 = 3$ for the growth model parameters in the experiments. The same pruning heuristic was used for all trees, with bounds $a_{\min} = 1$ and $a_{\max} = 4$ for the age of the pruned wood, minimum pruning mass $m_{\min} = 10$ per cut, and enabled pruning only after a fork. The minimum and maximum number of cuts were determined empirically, and were based roughly on individual tree sizes. Table 1 summarizes the main properties of tree models from Fig. 5, the corresponding range $[d_{\min}, d_{\max}]$, and the estimated size of the solution space.

The values of optimization meta-parameters for both methods were tuned experimentally. To make the tuning process manageable computationally, the tuning optimization runs were limited to 3,000 objective function evaluations (FE). Additionally, the constraint $P_{\text{add}} = P_{\text{remove}} = (1 - P_{\text{move}})/2$ was applied to allow optimizing along P_{move} alone. The hypervolume indicator I_H (Zitzler, Thiele, Laumanns, Fonseca, & Da Fonseca, 2003) was used for configuration ordering with respect to the median result of multiple optimization runs. A short preliminary phase with three optimization runs per configuration was performed to establish useful ranges of meta-parameter values. Based on these results, a grid search for optimal parameter values was performed next, where the median result out of 11 optimization runs was used to select the best configuration. The overall best configuration $T_0/M/Mr/P_{\text{move}} = 3/100/0.2/0.5$ was selected for SA, while for NSGA-II the configuration $Ps/Mr/P_{\text{move}} = 50/0.05/0.3$ came out on top.

Fig. 6 shows an example of I_H variation when moving away from the selected SA configuration, while Fig. 7 shows the same for NSGA-II. It can be concluded that T_0 and M have a stronger influence on optimization performance of SA than Mr and P_{move}, although the values of the latter two should not be too low to give the SA enough exploratory power. In the case of NSGA-II, population size is important, while the acceptable range of values for P_{move} is similar to that in SA. On the other hand, Mr should be smaller in NSGA-II, but extra mutation still seems to benefit the search.

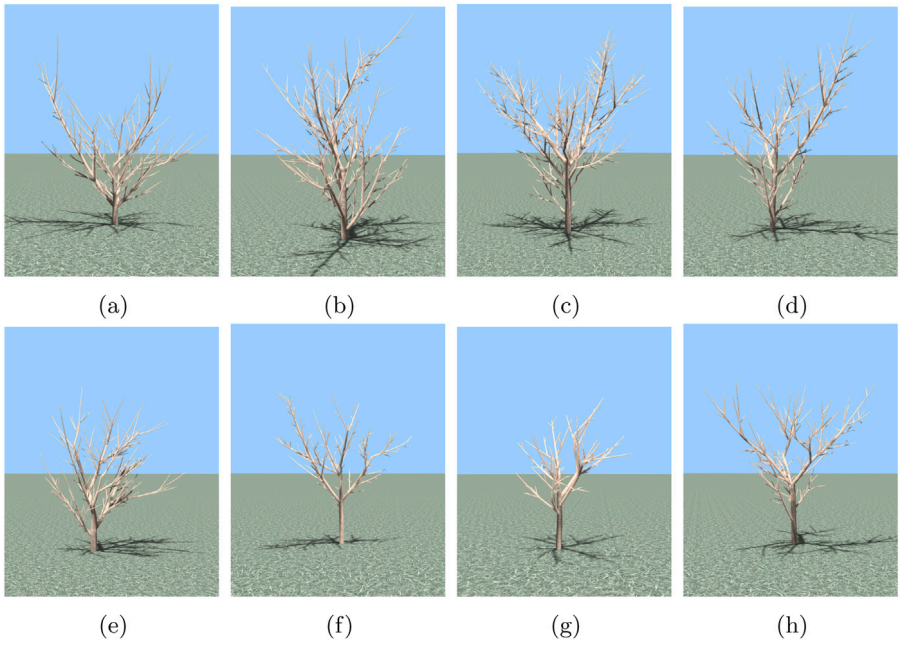

Fig. 5. Tree models used in the experiments.

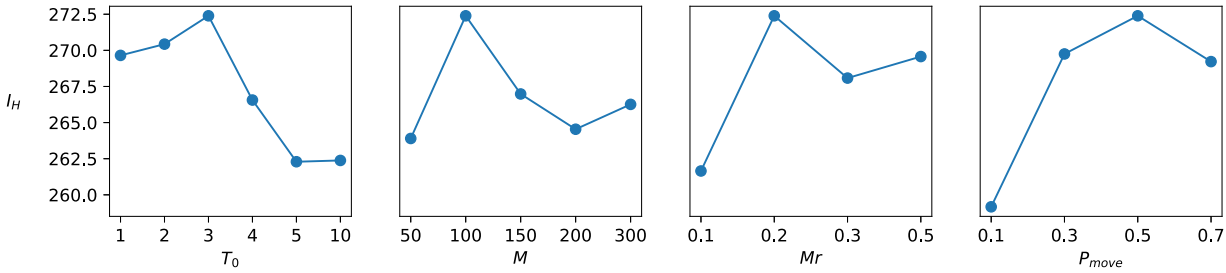

Fig. 6. The influence of changing the SA meta-parameter values on the median I_H, obtained from 11 optimization runs on the tree in Fig. 5(a). The influence of varying each parameter is shown with respect to the best configuration $T/M/Mr/P_{\text{move}} = 3/100/0.2/0.5$.

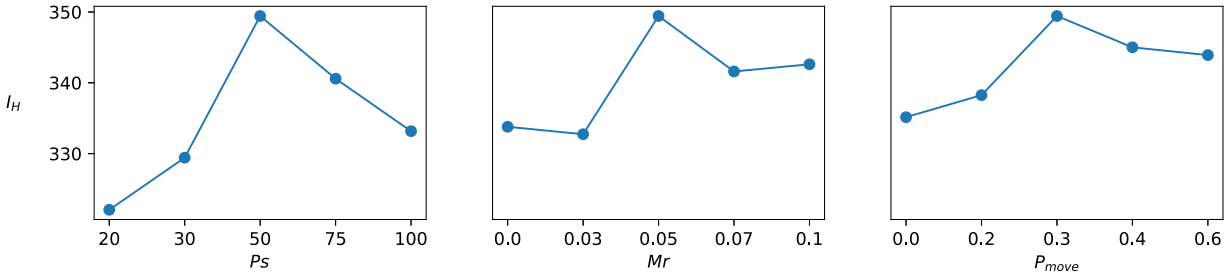

Fig. 7. The influence of changing the NSGA-II meta-parameter values on the median I_H, obtained from 11 optimization runs for the tree in Fig. 5(a). The influence of varying each parameter is shown with respect to the best configuration $Ps/Mr/P_{\text{move}} = 50/0.05/0.3$.

In the final experiments, the maximum number of objective function evaluations was set to $N = 10{,}000$, which amounts to the same number of iterations for SA, and 200 iterations of NSGA-II with population size 50. The initial SA temperature was scaled to $T_0 = 10$ to achieve the same cooling regime as in the shorter tuning runs. The number of growth simulations for estimating the f_2 objective value was $s = 20$.

The experiments were run on a desktop computer with Intel i7 CPU, Nvidia Geforce GTX 1060 graphics card, and 16 GB of RAM. The operating system was GNU Linux, kernel version 5.2.8, and the Nvidia graphics driver version was 430.40. The implementation of the framework in C++ was compiled using the `gcc` version 9.1.

5.2. Results

In this section, we compare the non-dominated sets constructed by SA and NSGA-II. We also include the results for the no-pruning case and two types of pruning. The first one simulates mechanical pruning to conical form. The second one is based on the distance rule (Karkee et al., 2014), where the necessary minimal spacing between branches is maintained by removing the more shadowed one of two branches if their distance is below some threshold.

Fig. 8 shows the non-dominated fronts constructed by the SA and NSGA-II methods for the trees from Fig. 5. Several conclusions can be

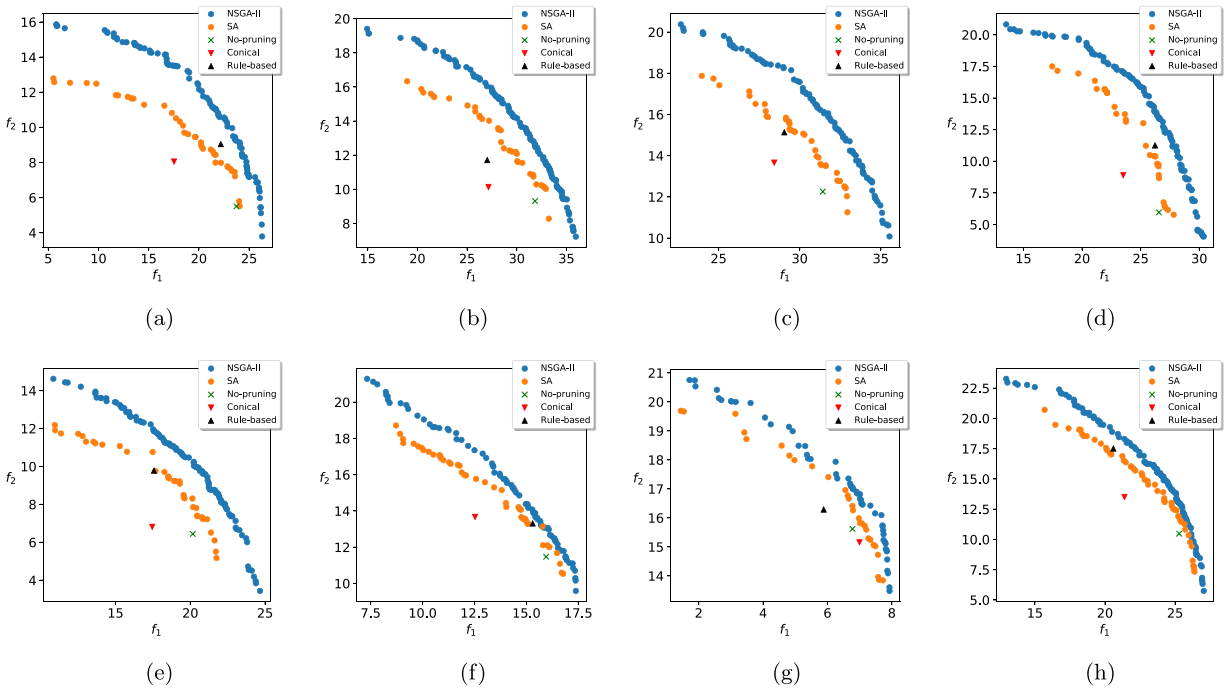

Fig. 8. Non-dominated fronts constructed by SA and NSGA-II for the corresponding trees from Fig. 5. The objective vectors for the no-pruning case, pruning to conical form, and pruning based on branch distance are also marked.

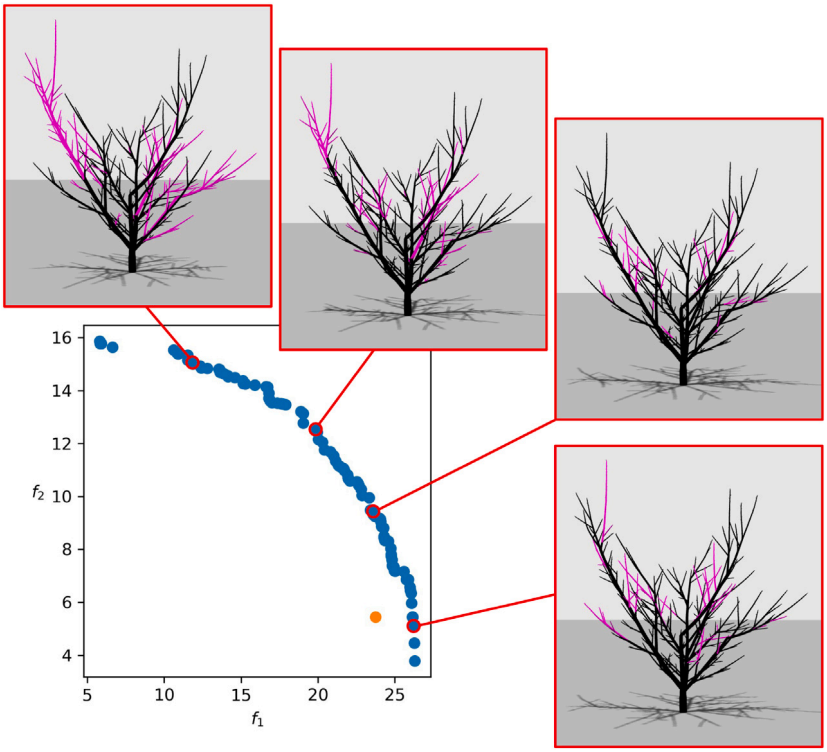

Fig. 9. Pruning examples corresponding to different non-dominated NSGA-II solutions for the tree from Fig. 5(a). The pruned parts of the tree are colored pink.

made based on these results. The main one is that the global exploration capabilities of NSGA-II make it evidently superior to the local search-based SA, both in terms of the achieved I_H and the spread of the front. The SA is relatively close only with respect to improving the immediate objective value on smaller problem instances, which is consistent with its comparative performance in single-objective pruning optimization. The fronts are approximately convex, and cover a spectrum of solutions that balance between improving either of the two objectives. However, it is possible to improve both objective values with respect to the no-pruning solutions simultaneously. This is evidenced by the positions of the latter, which is in the dominated region behind the frontier. The same is true for pruning to conical shape, while the pruning based on the distance rule is closer to the fronts, and, in multiple cases, dominated only by the NSGA-II solutions. Fig. 9 shows examples

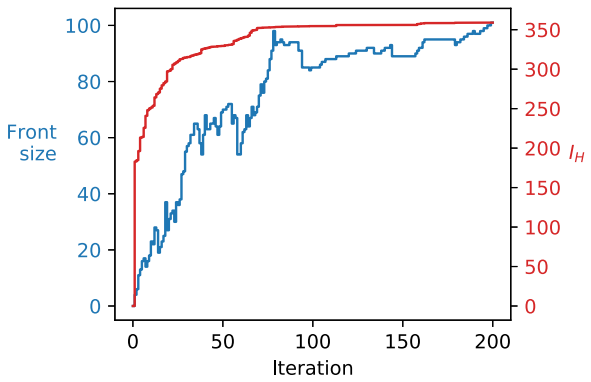

Fig. 10. The development of the NSGA-II non-dominated front for the tree from Fig. 5(a) during the optimization run.

Table 2
Experimental run-times (in minutes) of both methods for the trees from Fig. 5, where t_{sim} is the time spent in growth simulation, and t_{opt} is the time spent in optimization (which includes tree model pruning).

Tree model	SA		NSGA-II	
	t_{sim}	t_{opt}	t_{sim}	t_{opt}
Fig. 5(a)	20.4	1.6	25.1	1.7
Fig. 5(b)	37.7	2.5	40.8	2.8
Fig. 5(c)	56.9	3.6	57.8	3.7
Fig. 5(d)	28.1	1.9	31.2	2.1
Fig. 5(e)	24.7	1.6	26.7	1.7
Fig. 5(f)	15.4	0.9	15.3	0.9
Fig. 5(g)	12.8	0.7	12.8	0.7
Fig. 5(h)	21.6	1.3	21.7	1.3

of applied pruning solutions corresponding to objective vectors from different sections of the NSGA-II front.

It is informative to observe the development of the front throughout the optimization run. Fig. 10 shows the number of non-dominated objective vectors in the NSGA-II front from Fig. 9 as the optimization progresses, as well as the changing value of the hypervolume indicator I_H (i.e., the area of the first quadrant bounded by the front). The initial steep increase in the size of the front is followed by smaller increments interleaved with frequent reductions, which are due to newfound pruning vectors that dominate clusters of similar existing solutions and push the front outwards in larger steps. In later stages, the optimization focuses on the refinement of existing solutions, which leads to the extension of the front in the final exploitation stage of the search. The value of I_H, on the other hand, shows a rapid initial increase followed by steady growth, with occasional jumps at the points of significant front breakthroughs.

The running times of the experiments are presented in Table 2, where the execution time spent in optimization and growth simulation is reported separately. Because 90–95% of execution time is due to the simulations, the run-time can be reduced greatly by using fewer simulations per evaluation. The differences in optimization run-time between the two tested methods are, in most cases, negligible. For large tree models, the difference in simulation times is due to the varying complexity of generated pruning solutions, which affects the resource allocation and distribution computations.

5.3. Discussion

The presented concept of multi-objective pruning optimization with growth simulation in the loop indicates potential new educational and analytical uses of existing tools. To this end, there are several major directions for possible future work:

- extensions of the underlying simulation model to include more complex interactions of factors affecting tree development, such as those incorporated in some specialized models for particular genera (Lang & Lang, 2007; Lopez et al., 2010) or cultivars (Fišer et al., 2015; Kang et al., 2016),
- definition of new compound quantitative objectives, both short-term and long-term, that reflect particular goals of optimization, e.g., adherence of the tree structure to a certain form,
- implementation of more powerful MO methods, which can cope with the increased complexity of objective landscapes.

The complexity introduced by the above extensions would increase both simulation and optimization run-times, which could be counter-balanced by more aggressive pruning constraints and parallelization.

Besides being used in computer-aided education, pruning optimization is potentially interesting for tasks involving machine pruning (He, & Schupp, 2018; Zahid, He, & Zeng, 2019), where steady progress is being made towards automation. Recent advances in computer vision and deep learning improved the reconstruction of 3D tree models (Akbar, Elfiky, & Kak, 2016; Liu, Yao, Li, Qiu, & Liu, 2019; Zhang, He, Karkee, Zhang, Zhang et al., 2018), which could serve as inputs to the pruning decision module. Multi-objective pruning optimization could in such settings be used to supplement rule-based pruning.

6. Conclusion

A conceptual framework for multi-objective assessment of fruit tree pruning within the virtual tree simulation tool EduApple was presented. The optimization methodology allows the incorporation of different short-term, as well as long-term pruning objectives into simulated algorithmic pruning. The methodology was demonstrated for a bi-objective case study, where the maximization of estimated pre-growth and post-growth light intake of a pruned tree was performed. It was shown that NSGA-II outperforms local search-based optimization with respect to the spread and hypervolume of the obtained Pareto front approximations. The results indicate the potential application of multi-objective pruning optimization for educational and analytical purposes, while the framework can be extended in terms of included objectives and underlying simulation models.

CRediT authorship contribution statement

Damjan Strnad: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Štefan Kohek:** Conceptualization, Software, Visualization. **Bedrich Benes:** Writing - review & editing, Funding acquisition. **Simon Kolmanič:** Methodology, Visualization, Writing - review & editing. **Borut Žalik:** Supervision, Funding acquisition.

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.

Acknowledgments

This work was supported by the Slovenian Research Agency, Slovenia (research program P2-0041 and project J2-8176), the National Science Foundation, USA grant #10001387, *Functional Proceduralization of 3D Geometric Models*, and the Foundation for Food and Agriculture Research, USA under award number — Grant ID: 602757. The content of this publication is solely the responsibility of the authors, and does not necessarily represent the official views of the Foundation for Food and Agriculture Research.

References

Akbar, S. A., Elfiky, N. M., & Kak, A. (2016). A novel framework for modeling dormant apple trees using single depth image for robotic pruning application. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 5136–5142). IEEE.

Allen, M., Prusinkiewicz, P., & DeJong, T. M. (2005). Using L-systems for modeling source–sink interactions, architecture and physiology of growing trees: The L-PEACH model. *New Phytologist*, *166*, 869–880.

Balandier, P., Lacointe, A., Le Roux, X., Sinoquet, H., Cruiziat, P., & Le Dizès, S. (2000). SIMWAL: a structural-functional model simulating single walnut tree growth in response to climate and pruning. *Annals of Forest Science*, *57*, 571–585.

Bazzaz, F. A., Ackerly, D. D., & Reekie, E. G. (2000). Reproductive allocation in plants. In M. Fenner (Ed.), *Seeds: The ecology of regeneration in plant communities* (2nd ed.). (pp. 1–30). UK: CAB International Wallingford.

Benes, B. (1996). An efficient estimation of light in simulation of plant development. In R. Boulic, & G. Hégron (Eds.), *Springer computer science*, *Computer animation and simulation'96* (pp. 153–165). Springer.

Cokelaer, T., Fumey, D., Guédon, E., & Godin, C. (2010). Competition-based model of pruning: Applications to apple trees. In *6th International workshop on functional-structural plant models* (pp. 87–89).

Collette, Y., & Siarry (2003). *Multiobjective optimization: Principles and case studies*. Berlin, Germany: Springer.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.

Ersin, A. (2017). A new insight into pruning strategy in the biennial cycle of fruiting: Vegetative growth at shoot and whole-tree level, yield and fruit quality of apple. *Notulae Botanicae Horti Agrobotanici Cluj-Napoca*, *45*, 232–237.

Fišer, M., Ravi, J., Benes, B., Shi, B., & Hirst, P. (2015). IMapple: a source–sink developmental model for 'Golden Delicious' apple trees. In *X International symposium on modelling in fruit research and orchard management, vol. 1160* (pp. 51–60).

Giulivo, C. (2011). Basic considerations about pruning deciduous fruit trees. *Advances in Horticultural Science*, *25*, 129–142.

Harris, R. W. (1994). Clarifying certain pruning terminology: thinning, heading, pollarding. *Journal of Arboriculture*, *20*, 50–54.

He, L., & Schupp, J. (2018). Sensing and automation in pruning of apple trees: A review. *Agronomy*, *8*, 211.

Ikinci, A., Kuden, A., & Ak, B. E. (2014). Effects of summer and dormant pruning time on the vegetative growth, yield, fruit quality and carbohydrate contents of two peach cultivars. *African Journal of Biotechnology*, *13*, 84–90.

Kang, H., Fišer, M., Shi, B., Sheibani, F., Hirst, P., & Benes, B. (2016). IMapple — functional structural model of apple trees. In *2016 IEEE international conference on functional-structural plant growth modeling, simulation, visualization and applications (FSPMA)* (pp. 90–97).

Karkee, M., Adhikari, B., Amatya, S., & Zhang, Q. (2014). Identification of pruning branches in tall spindle apple trees for automated pruning. *Computers and Electronics in Agriculture*, *103*, 127–135.

Kirkpatrick, S., Gelatt, D. C., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Kohek, Š., Guid, N., Tojnko, S., Unuk, T., & Kolmanič, S. (2015). EduAPPLE: Interactive teaching tool for apple tree crown formation. *HortTechnology*, *25*, 238–246.

Kolmanič, S., Tojnko, S., Unuk, T., & Kohek, Š. (2017). The computer-aided teaching of apple tree pruning and training. *Computer Applications in Engineering Education*, *25*, 568–577.

Lang, R. J., & Lang, G. A. (2007). VCHERRY–an interactive growth, training and fruiting model to simulate sweet cherry tree development, yield and fruit size. In *VIII international symposium on modelling in fruit research and orchard management, vol. 803* (pp. 235–242).

Lescourret, F., Moitrier, N., Valsesia, P., & Génard, M. (2011). Qualitree, a virtual fruit tree to study the management of fruit quality. I. model development. *Trees*, *25*, 519–530.

Liu, S., Yao, J., Li, H., Qiu, C., & Liu, R. (2019). Research on 3D skeletal model extraction algorithm of branch based on SR4000. *Journal of Physics: Conference Series*, *1237*, Article 022059.

Lopez, G., Favreau, R. R., Smith, C., & DeJong, T. M. (2010). L-PEACH: a computer-based model to understand how peach trees grow. *HortTechnology*, *20*, 983–990.

Maggs, D. J. (1963). The reduction in growth of apple trees brought about by fruiting. *Journal of Horticultural Science*, *38*, 119–128.

Mohammadi, A., Mahmoudi, M. J., & Rezaee, R. (2013). Vegetative and reproductive responses of some apple cultivars (Malus domestica Borkh.) to heading back pruning. *International Journal of AgriScience*, *3*, 628–635.

Měch, R., & Prusinkiewicz, P. (1996). Visual models of plants interacting with their environment. In *Proceedings of the 23rd annual conference on computer graphics and interactive techniques* (pp. 397–410). New York, NY, USA: ACM.

Olesen, T., Menzel, C. M., McConchie, C. A., & Wiltshire, N. (2013). Pruning to control tree size, flowering and production of litchi. *Scientia Horticulturae*, *156*, 93–98.

Pałubicki, W., Horel, K., Longay, S., Runions, A., Lane, B., Měch, R., & Prusinkiewicz, P. (2009). Self-organizing tree models for image synthesis. *ACM Transactions on Graphics*, *28*(58).

Strnad, D., & Kohek, Š. (2017). Novel discrete differential evolution methods for virtual tree pruning optimization. *Soft Computing*, *21*, 981–993.

Villasante, M., Godoy, S., Zoffoli, J. P., & Ayala, M. (2012). Pruning effects on vegetative growth and fruit quality of 'Bing'/'Gisela®5' and 'Bing'/'Gisela® 6' sweet cherry trees (Prunus avium). *Ciencia e Investigación Agraria*, *39*, 117–126.

Xia, N., Li, A.-S., & Huang, D.-F. (2009). Virtual apple tree pruning in horticultural education. In *International conference on technologies for E-learning and digital entertainment* (pp. 26–37). Springer.

Yang, L., Chen, J., Hua, J., Kang, M., & Dong, Q. (2015). Interactive pruning simulation of apple tree. In *International conference on computer and computing technologies in agriculture* (pp. 604–611). Springer.

Zahid, A., He, L., & Zeng, L. (2019). Development of a robotic end effector for apple tree pruning. In *2019 ASABE annual international meeting*. American Society of Agricultural and Biological Engineers.

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., & Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (R-CNN). *Computers and Electronics in Agriculture*, *155*, 386–393.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*, 117–132.