Applied Soft Computing Journal 99 (2021) 106931

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

An algorithm for automatic dormant tree pruning

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

[a] University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia
[b] Purdue University, West Lafayette, IN 47906, USA

ARTICLE INFO

Article history:
Received 2 October 2019
Received in revised form 16 October 2020
Accepted 18 November 2020
Available online 27 November 2020

Keywords:
Virtual tree
Tree pruning
Pruning automation
Discrete differential evolution
Tree height control

ABSTRACT

Tree pruning is a labor and cost-intensive task. Still, it is a necessary activity that ensures a high yield of good quality products in horticulture and increases the overall health of trees in general. Extensive research has been done attempting to automate this labor-intensive procedure, lower the cost, and demand a skilled workforce. We introduce a new algorithm based on discrete differential evolution that simulates the pruning of virtual trees. Although pruning driven by differential evolution alone optimizes the overall tree light intake, it cannot maintain the distance between individual trees, nor can it shape trees into any of the growing forms. In the article, we show that adding additional steps into the pruning process, which is an initial trimming of the tree into a desired shape, can be improved significantly. We demonstrate our method by simulating the pruning of virtual trees and show that it provides results comparable to the results obtained by a human expert. By simulating the tree pruning over a few consecutive years, We show that our method is also capable of autonomous tree training toward the desired growing form.

1. Introduction

The tree canopy pruning is the process of removing branches that has many objectives. Among them, improving the overall light intake is one of the most important ones. Living branches are removed and dead ones to balance the reproductive and vegetative growth in the wood and fruit production. Pruning must be done carefully not to damage the tree and to prevent fungus infections on the cuts. This makes pruning one of the most expensive and labor-intensive tasks responsible for approximately 20% of the annual pre-harvest cost for crops like apples, cherries, and pears [1]. A large number of trained seasonal workers is needed to accomplish this task, following a set of predefined rules [1] or experience and intuition.

Extensive research has been done on how to automate the tree pruning process to reduce costs and demands for a skilled workforce [1] resulting in the development of early mechanical systems for mass pruning. The fully automated results were not satisfactory, as evidenced by the reduced quality and yield of fruit [1]. Another previous work used a mobile platform for automated pruning of grape vines [2]. Both approaches create a 3D model of the plant by using computer vision-based reconstruction, and a decision system determines which parts of the plant should be removed. Actual pruning is carried out by a six-degrees-of-freedom robotic arm. Both the apple tree and grapevine pruning are carried out while the plants are in a dormant state, and in both cases, a set of predefined pruning rules controls the pruning.

In those approaches, the pruning is carried out in two phases. First, the 3D model of a tree is generated upon which the pruning rules are applied. Plant reconstruction is a hard problem by itself, and many approaches have been introduced [1,3]. By applying the tree pruning rules on the generated tree 3D model the surplus branches are identified that have to be removed [1,4]. The pruning rules aim to increase the canopy's irradiance intake to improve the tree health and, in effect, fruit quality [5].

Strnad and Kohek [6] used discrete differential evolution (DDE) to determine which branches should be removed. The method does not allow the tree to form into a specific growing form and has significant variations in the pruning results. We solved these problems by trimming the trees into a predefined template to maintain a desired tree height and distance to its neighbors. In the second step, we employ DDE to determine which branches should be removed to optimize the light intake.

We have implemented our method in the software apple tree plant simulator EduAPPLE [7]. We ran several experiments using two different pruning templates in the first step of pruning: a cylinder and a cone. We compared the light distribution inside the tree canopy after pruning trees using the newly-developed method with those pruned by the expert. Our results show comparable levels of irradiance within the canopy. Additionally, by

* Corresponding author.
E-mail address: simon.kolmanic@um.si (S. Kolmanič).

https://doi.org/10.1016/j.asoc.2020.106931
1568-4946/

using the proposed pruning method for several consecutive years, the space between trees was kept free of competing branches, along with preserving their height.

2. Materials and methods

The problem of tree pruning equals the reduction of the internodes in the tree in order to achieve a balance between vegetative and reproductive growth. Thus the pruning can be expressed as a function of a tree structure and tree crown light distribution. This function is currently described only by more or less strict pruning rules that are individually followed by human pruners regarding their working experience. The pruning rules are defined better by the pruning automation approach, but the set of rules, in this case, is reduced to a few rules only, adjusted to a particular tree growing form [1,8].

To overcome these limitations, we defined the pruning problem as the optimization of the tree crown light distribution by a limited removal of the branches and, thus, as little interference in the tree structure as possible, to avoid adverse effects the pruning might have on the tree. We evaluated the pruning intensity by a ratio between the number of internodes before and after the pruning. The changes in the light distribution are calculated using the concept of self-organizing trees of [9]. The tree crown light distribution is represented by the histogram with the ten bins of equal size. We incorporated both the pruning intensity and the tree crown light distribution into the objective function, which is optimized by using the discrete differential evolution method [7]. By shaping a tree into a conical or cylindrical form, we prevented the competition among trees for space.

2.1. Tree growth model

The tree is modeled as a hierarchy of modules (Fig. 1a) which is a common approach used in many plant simulators such as the developmental models of de Reffye et al. [10], recently introduce IMapple [11], self-organized tree models that compete for space and light [9], plastic trees that automatically learn and adapt to the environment [12], or developmental models of [13,14].

One or more leaves are attached to the stem at called a node, and the part of the stem between two nodes is an internode. The growth is controlled by apical meristem that is a region of dividing cells that responds by growth against gravity (gravitropism) [15] and toward the incoming light (phototropism) [16]. Depending on the plant species and environmental factors (light, temperature, nutrients, etc.), the plant produces lateral buds that are either dormant or active and produces leaves with specific orientation (phyllotaxis). An internode with attached leaves and a lateral bud is called a metamer, and a sequence of metamers grown at a single spurt forms a shoot. The shoot axis is produced by the terminal bud, located at the end of the shoot (apex).

Our approach uses the previously developed framework EduAPPLE [7], where a tree growth is driven by buds' illuminations [16–18] and a competitive process for growth resources [9, 19–21], which leads to the self-organizing structure of the tree [9]. The tree attempts to maximize its branch mass by growth and light intake. Buds with higher irradiance produce new shoots that compete for space. Buds with lower light intake produce quickly growing shoots that attempt to get from shade (light seeking stage). Poorly illuminated buds do not create new shoots, become dormant, and form new shoots later, when the conditions are more favorable. The key factor of this simulation is the calculation of the illumination of leaves that feed buds. Although inner reflection can be considered [22], the algorithms are usually time-consuming, and the indirect light does not contribute significantly to growth, because the direct lighting gives major light intake. A faster way is to use only the direct illumination [12,16–18].

The tree casts a shadow on itself and thus forming a shadow space of the tree, as shown in Fig. 1b. The shadow density is highest at the trunk; therefore, leaves/shoots rarely grow from the dormant buds positioned there. Trees produce shoots on their trunks rarely (for example, when they are sick), because of the low lighting, thick bark, etc. This is in tune with observations in nature [23].

2.2. Illumination

The goal of pruning is to improve the illumination of the inner parts of the tree. When the lighting conditions around the stem improve after removing some branches, dormant buds can reactivate and start to produce new shoots. We calculated the inner shading by using the algorithm from [9] that was later extended in [6,12,14].

The light distribution inside the tree crown is calculated from the illuminated buds that are sources of a conical shadow volume. We calculate the bud illumination. Assume a bud inside one of the shadow volumes of one of the neighbors. Let d_y be the bud's vertical distance from the volume apex and $d_{x,z}$ its horizontal distance from the volume axis, then the received shadow, originating from a given shadow volume δs is calculated [6]. Total irradiance Q of a given bud is then:

$$Q = max\{1 - s, 0\}, \tag{1}$$

where s is the cumulative contribution of all shadow volumes captured by the given bud.

2.3. Differential evolution and pruning

The Differential Evolution (DE), developed by Storn and Price [24], aims to optimize certain properties of a system pertinently choosing the system parameters, usually represented as a vector. It is widely used to solve various problems from different fields (*e.g.*, optimization [25,26], path planning [27,28], and agriculture [29–31]).

The objective function models the goals, while incorporating constraints. For the objective function:

$$f : X \subseteq R^n \to R, \tag{2}$$

where $X \neq \emptyset$, the minimization problem is to find solution vector $\mathbf{s} \in X$, such that $f(\mathbf{s}) \leq f(\mathbf{p}), \forall \mathbf{p} \in X$. In case of the tree pruning $\mathbf{s}$ is a prune specification and fitness function $f(\mathbf{s})$ is the light intake after completion of prune $\mathbf{s}$. DE employs evolutionary operators (mutation, crossover, and selection) to minimize the objective function [32]. The three operators are sequentially (in a fixed order) carried out in a loop, until an adequate fitness or number of iterations is reached.

The DE produces a population of individuals and evaluates their fitness function (light intake in our case). The **mutation operator** produces a trial vector for each individual of the current population by mutating a target vector with a weighted differential. For each parent $\mathbf{x}_i \in X$ three distinct individuals $\mathbf{x}_{i_1}$, $\mathbf{x}_{i_2}$, and $\mathbf{x}_{i_3}$ are selected randomly from the population, $i \neq i_1 \neq i_2 \neq i_3$ and used to calculate the trial vector $\mathbf{u}_i$ as follows:

$$\mathbf{u}_i = \mathbf{x}_{i_1} + \beta \times (\mathbf{x}_{i_2} - \mathbf{x}_{i_3}) \tag{3}$$

where $\beta \in (0, \infty)$ is called differential weight. Vector x_{i_1} is a target vector, while the difference $x_{i_2} - x_{i_3}$ refers to a differential vector. The best found solution so far is used as a target vector. Together with the population β can greatly impact the optimization performance. Both parameters, the differential weight and population size are selected by the user.

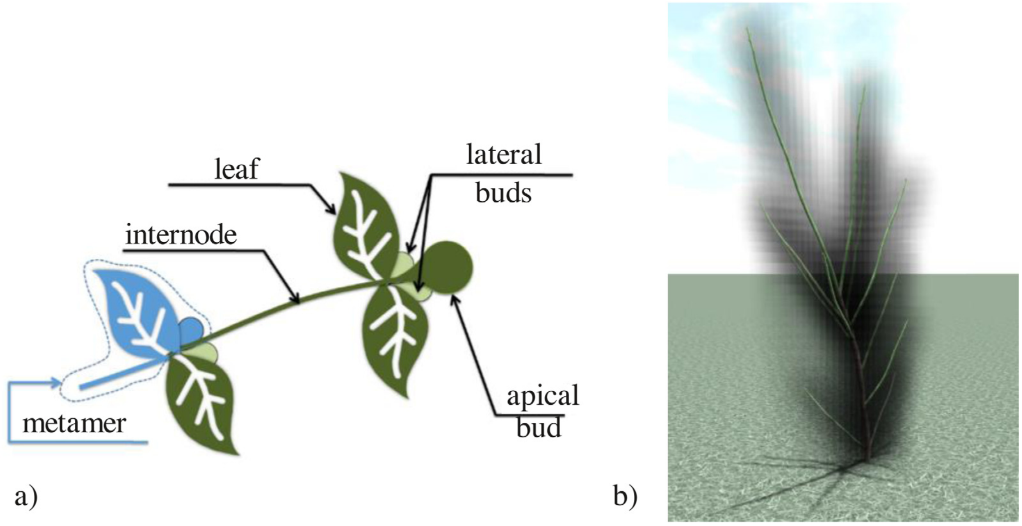


Fig. 1. (a) Plant modules of a self-organizing tree growth model and (b) the shadow space of a tree in used tree growth model from [7].

The DE **crossover operator** implements a discrete recombination of trial $\mathbf{u}_i$ and the parent $\mathbf{x}_i$ vectors, to produce offspring $\mathbf{x}_i^{'}$. The crossover operator is defined by [6,33]:

$$x_{i,j}^{'} = \begin{cases} u_{i,j}, & \text{if } r \leq C \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{4}$$

where $x_{i,j}^{'}$ refers to the jth element of the vector $\mathbf{x}_i$, $r \sim U(0, 1)$ is a random number generated with uniform distribution. The parameter $C \in (0, 1)$ determines the probability of inheritance from the mutant. The base vectors in our case are randomly chosen. In order to form a mutant population, only one vector difference is used and uniform crossover was employed during the formation of the trial population. The values of β and C had to be tuned in regard of the tree age and even selected pre-pruning shape to generate acceptable pruning results parameters. As shown in [6], this tuning is a time demanding process, therefore it is beneficial to calculate these parameters in advance.

The **selection operator** is then applied to construct the population for the next generation. The offspring replaces the parent if the offspring's fitness surpasses that of the parent; otherwise, the parent survives to the next generation. This ensures that the average fitness of the population does not deteriorate.

2.3.1. Discrete differential evolution

Discrete DE (DDE) variants have been presented concerning specific combinatorial optimization problems, e.g., [34–36]. The search space of possible pruning options grows exponentially, which makes a brute force search unfeasible. Various heuristics could be used, and in this paper, we adapt the DDE algorithm [6] that attempts to optimize the light conditions within the canopy by pruning branches by using genetic algorithms. Through optimization of pruning locations, a combination of cuts is obtained that maximizes the amount of light received by the remaining buds of the tree crown. For that purpose, for all buds in the tree crown, the irradiance is calculated by the use of Eq. (1). A bud's irradiance, Q corresponds to the percentage of available light intercepted by the bud. For the sake of faster detection of ancestor/successor relationship between internodes, each internode is associated with a unique variable-length binary string as shown in Fig. 2.

Tree crown light distribution is calculated next, where the irradiance of each bud in the crown is assigned to one of ten quantization classes of equal width of 0.1 on the interval [0, 1].

The objective function $f(\mathbf{x})$ is:

$$f(\mathbf{x}) = \frac{\sqrt{S_{\text{tree}}}}{H} \sum_{i=1}^{10} i h_i, \tag{5}$$

where S_{tree} is the number of remaining internodes after pruning, H is the total number of buds, and h_i is the number of buds in the i-th class of light distribution.

The solution vector is denoted by $\mathbf{x} = \{x_1, \ldots, x_s\}$ and it contains the encoded sequence of cut positions labeled by the corresponding internodes. The root is denoted by a bit-string "0". A '0' or '1' is appended to the parent's string for each main or lateral child internode, respectively (Fig. 2). Each cut position x_i identifies the internode, at which the branch is removed. Variable s in this case is a population size and is a value between $s_{\min}$ and $s_{\max}$, which are custom set parameters, representing the minimum and the maximum number of allowed cuts. The objective function $f(\mathbf{x})$ favors solutions that provide a maximal improvement of light distribution with a minimal amount of removed biomass. Many possible cuts are redundant e.g., sub-branches of already removed branches. To avoid such operation, four methods have been proposed in [6]: BinDE, IndexDE, PathDE, and SetDE. These methods perform similarly in optimizing the light conditions within the canopy, and they also perform similarly to the state of the art methods. Although PathDE provided slightly better results than the other algorithms, we use BinDE, because it is more straightforward and thus easier to understand (see Fig. 2). If, for example, a cut should be made at internode "00", the cuts "001", "000", or "0010", would be redundant and thus unnecessary. Used labeling allows for quick detection of the redundant cuts. In a given example, the later three internodes share the same prefix "00", which is the label of their predecessor, and with removing it, all of its successors are removed as well.

The encoding scheme uses the real-valued vectors to represent solution genotypes as shown in Fig. 2. For the mapping of genotypes to phenotypes (i.e., pruning instances), the intermediate cut list is used. It is produced by traveling the tree model in a depth-first manner and recording the labels of all internodes encountered in the process into a list. The length of the cut list defines the dimensions of solution vectors. Each component $x_{i,j} \in [0, 1]$ of the solution vector $\mathbf{x}_i$ determines whether j-th internode from the cut list will be selected as the cutting point or not. For that purpose the binary vector $\mathbf{z}_i$ is constructed by thresholding:

$$z_{i,j} = \begin{cases} 0, & \text{if } x_{i,j} < 0.5 \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

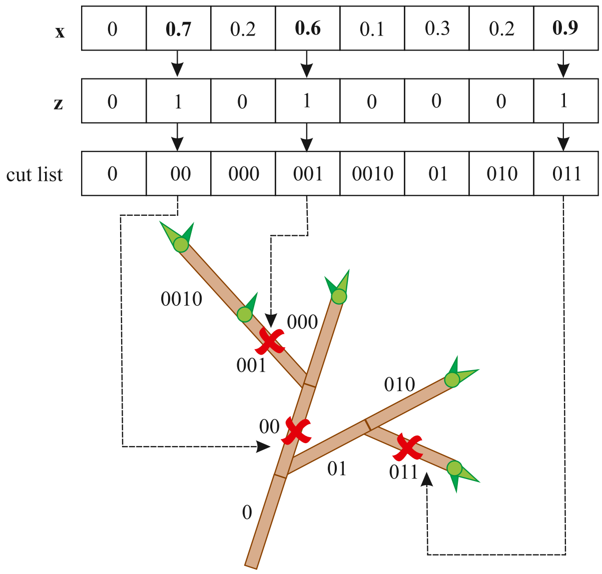


Fig. 2. Mapping of the solution vertex into the tree cutting sequence — genotype to phenotype mapping. Solution vector components that exceed the threshold are converted into the cuts with the help of the cut list.

If the number of cuts proposed by the vector $\mathbf{z}_i$ violates the solution size constraints, the threshold is adjusted up or down so that the number of cuts fall into valid range. The value $x_{i,j}$ is the stability of cut inclusion in the pruning, because cuts with higher value of $x_{i,j}$ are less likely to be disturbed by small perturbations of the solution vector.

2.4. Tree height and neighboring distance control

Although the DDE method improves the light conditions inside the tree crown, it offers no control over the tree size or the distance to neighbor trees. Since the anti-hail nets mostly protect the modern orchards, it is crucial to keep apple trees below a certain height. Similarly, the neighboring distance must be preserved during the entire lifetime of a farm so that the tree branches do not overlap and do not compete for the same space.

A straightforward method to get control over the tree height and neighboring distance would be the extension of the objective function $f(\mathbf{x})$ with additional constraints regarding the tree size. It turned out, however, that while the integration of the tree height into the tree model and objective function is possible, the determination of the extent of the tree crown after the pruning was not efficient, since the buds' locations inside the tree have to be included into the growth model for that purpose. This would lead to higher memory requirements and increased processing time.

Our inspiration for the solution to this problem comes from observing a human during manual pruning. Instead of just following the pruning rules, the human pruner strives to shape the tree into one of the distinct growing forms. The experts developed those growing forms over the years and gave the best fruiting results under certain growing conditions. In high-density orchards, the most common tree forms are the Slender Spindle [37], and more recent, the Tall Spindle [38]. While the Slender Spindle is conically shaped (Fig. 3a), the Tall Spindle resembles a cylinder (Fig. 3b). Tall spindle is popular because it is suitable for mechanical pruning and formation of the Fruiting Wall [38].

Instead of changing the objective function, we propose adding a preprocessing level to the DDE method. We call this step *shaping*, and by adding this to the desired form, the tree height and neighboring distance can be controlled easily. We prune the trees to either a conical or cylindrical shape with adjustable height and the base radius. After the branches outside the chosen shape are trimmed off, the DDE method is used to optimize the light condition inside the tree crown. The entire tree pruning process is depicted in Fig. 4.

The results of the final pruning depend on the maximum allowed number of cuts $s_{\max}$, which has to be adjusted to the tree age and thus complexity of the tree crown. In our implementation, we set $s_{\max} = 20$ which provided good results for young trees.

While we used circular proxies in our work, other shapes could be used: for example, orchards with Fruiting Wall planting system could use rectangular blocks.

Table 1
The percentage of buds receiving more than 70% of light in an unpruned tree and after manual pruning by using HeP and HeF methods and our automatic pruning by using DDECn and DDECy. The improvement in all cases is in average of 168%.

Method	% of buds receiving more than 70% of light	Δ to unpruned [%]
Unpruned	14.89	0
Manual Expert (HeP)	26.80	180
Manual Expert (HeF)	22.95	154
Automatic (DDECn)	25.93	174
Automatic (DDECy)	24.18	162

3. Results

We pruned a four years old virtual untrained tree (Fig. 5) by using the proposed method. A horticulture expert also pruned the same tree, and the results were compared. The tree training teaching environment EduAPPLE [7] was used for this purpose. The expert shaped the tree into two primary forms, a pyramidal growing form (denoted as HeP — Human expert - Pyramidal), which is similar to a Slender Spindle, and a Flatt growing form (denoted as HeF — Human expert - Flatt), suitable for the Fruiting Wall planting system. For the automated pruning, we used a cone with the height of 3m and the opening angle of 45° (denoted as DDECn — DDE method, Conical initial shape), and a cylinder with a height of 3 m and radius of 0.75 m (denoted as DDECy — DDE method, Cylindrical initial shape). In both cases, we set $s_{\max} = 50$. The resulting tree geometry is the basis for the evaluation of the correctness of a growing form. By showing the trees pruned by the DDE method to those pruned by humans side-by-side in Fig. 5, it can be observed that the shapes are comparable, but an automatic metrics should be developed to provide an exact comparison.

Light Distribution: In all four cases, the light distribution inside the tree crown has improved (see Table 1) as compared to the unpruned tree (Fig. 6) with the average increase of 168% of buds. In particular, only 14.89% of buds receive more than 70% of available light in the unpruned trees, while in the human-pruned tree, the amount of such buds increases to 26.80% (HeP) and the 22.95% (HeF). The light distributions of the automated tree pruning are comparable with that of the human expert (25,93% for DDECn and 24.18% for DDECy). When comparing HeP to DDECn the result in the cases of human pruning is slightly better but in the case of HeF and DDECy the automated pruning achieved better result, although the difference in both cases is less than 1.3%.

Pruned Internodes: The difference between the DDE and human pruning is also visible in the category of the internodes left in the tree after the pruning as shown in Table 2. The unpruned

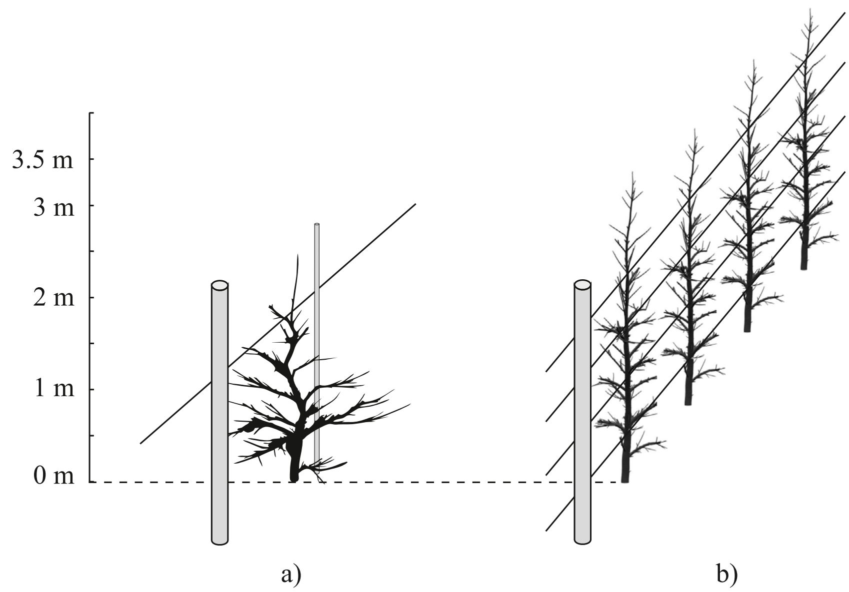


Fig. 3. High density apple systems (a) Slender Spindle is a growing form used in older orchards, (b) Tall Spindle is a recent growing form with higher yields.

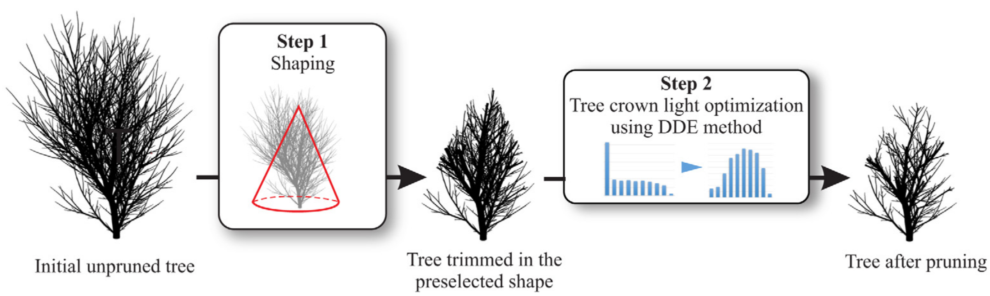


Fig. 4. Two-step virtual pruning process enables the control over tree height and neighboring distance. First, the tree is shaped into a cone or cylinder shape with adjustable size. In the second step the DDE method selectively removes branches to improve the light conditions inside the tree crown.

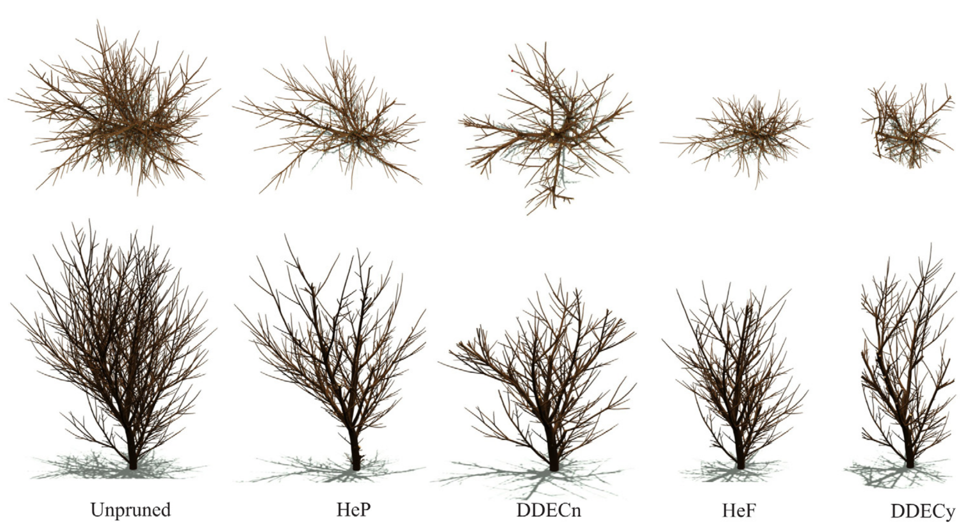


Fig. 5. Comparison of pruning with the initial unpruned tree, tree pruned by a human expert in a pyramidal shape (HeP), automated pruning with initial cone shape (DDECn), pruning by human expert pruning in a plat plane (HeF), automated pruning with the use of cylindrical initial shape (DDECy).

tree has 18,618 internodes. After manual pruning by using HeP the number decreased to 7810 and manual pruning by using HeF decreased the number to 5715. Automatic pruning provided smaller numbers. The DDECn algorithm pruned the tree to 5714 and the DDECy to 4867 nodes. The average number for manual pruning was 6776 and for automatic 5291.

By combining both results from Tables 1 and 2, it can be concluded that the human expert achieved similar bud light

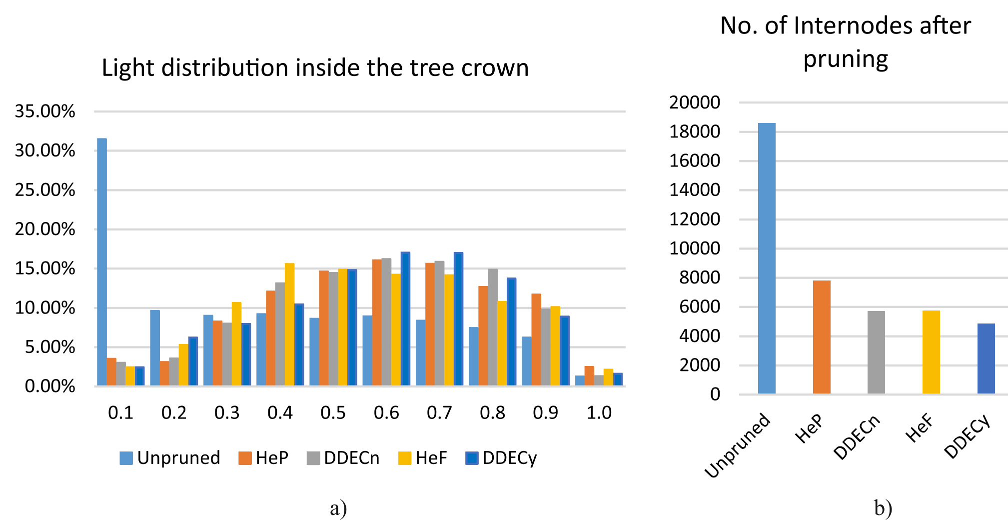


Fig. 6. Evaluation of tree pruning results, (a) Comparison of light distribution after the pruning by a human expert (HeP and HeF) and automated pruning (DDECn and DDECy), (b) Number of internodes left after the pruning. A higher number of internodes, combined with higher light exposure of buds signify better results.

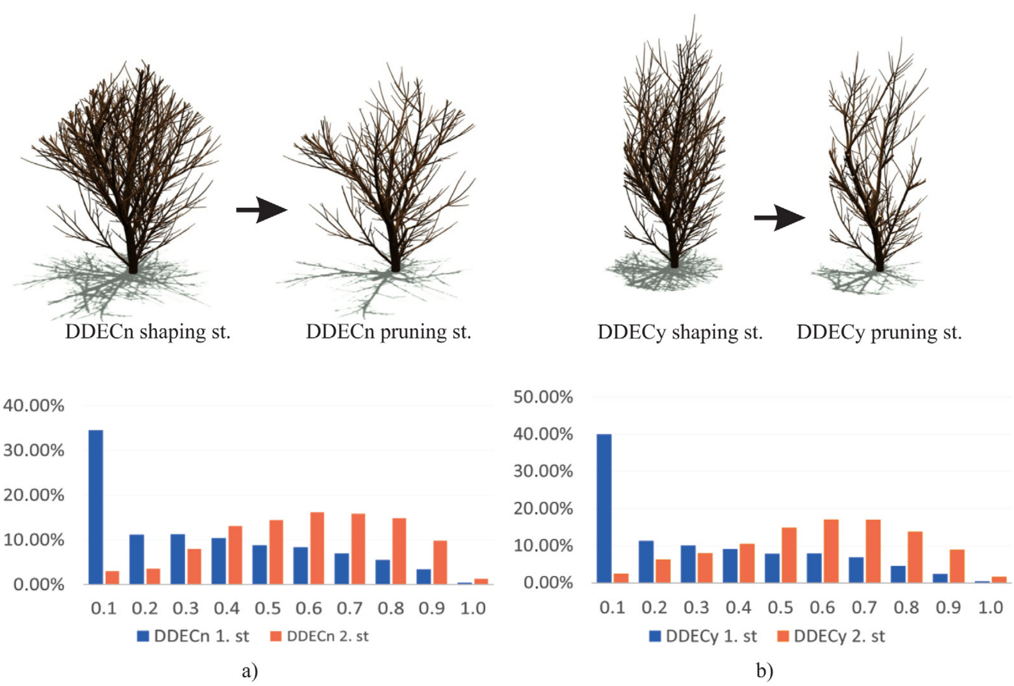


Fig. 7. Tree crown light distribution after the shaping and after the pruning step, (a) DDECn method, (b) DDECy method.

Table 2
The number of internodes on the tree in an unpruned tree and the comparison to the number of internodes after manual pruning by using HeP and HeF methods and our automatic pruning by using DDECn and DDECy. The average number of internodes is 6033 that corresponds to 32% of nodes of the unpruned tree.

Method	Number of internodes on the tree	Δ to unpruned [%]
Unpruned	18,618	0
Manual Expert (HeP)	7,810	42
Manual Expert (HeF)	5,715	31
Automatic (DDECn)	5,741	31
Automatic (DDECy)	4,867	26

exposure with less biomass removed, which is desirable since the higher number of internodes with well-illuminated buds represent the higher potential for the yield of both increased quality and quantity.

The DDECn and DDECy methods represent selective pruning, where the results of the first step represent the result of pruning by the currently used automatic pruning systems. The difference between the first and second steps of DDECn and DDECy can be seen in Fig. 7. Visually, the trees after the second step are less dense as the number of branches is drastically diminished, while the overall height of the tree remains the same.

Pruning stability: the quality of the pruning solution depends on the number of the iteration in the main DDE loop. To exclude the possibility that this can impact the pruning results, we conducted two experiments in which the pruning of a test tree was repeated 100 times. In the first experiment, the main loop was repeated for 1000 times, and in the next experiment, 10,000 iterations was run. The average bud's illumination by the DDECn was 0.649 (SD = 0.024) in the first experiment and 0.664 (SD = 0.022) in the second, where 1.0 is unobstructed illumination. By the DDECy, the average bud illumination was 0.650 (SD = 0.023) in the first

Table 3
Comparison of the tree pruning results using BinDE, Simulated Annealing (SA), PathDE, and SetDE methods while using Cone and Cylinder as the initial shaping form. Parameters E and SD represent the average values and standard deviations for 10 runs, respectively. In the methods BinDE, PathDE, and SetDE for the parameters β and C the values 0.3, 0.5, and 0.9 were used. The number of fitness evaluation was set to 1000, and the tree initially consisted from 13,055 internodes.

Method		FE value		Remaining internodes after pruning		Duration [s]	
		E	*SD*	*E*	*SD*	*E*	*SD*
Cone	SA	80.74	5.68	11,092.50	2,273.71	33.87	4.67
	BinDE - 0.3	87.70	1.55	9,282.50	12,25.98	88.08	5.27
	BinDE - 0.5	87.07	1.83	9,211.10	16,86.49	79.55	0.97
	BinDE - 0.9	84.48	2.08	8,690.20	34,25.67	83.20	4.10
	PathDE - 0.3	99.81	1.46	6,307.90	675.00	61.87	7.63
	PathDE - 0.5	99.77	1.83	5,917.90	577.81	50.58	3.80
	PathDE - 0.9	96.53	2.17	6,449.60	1,064.83	51.24	3.14
	SetDE - 0.3	93.41	4.25	6,684.10	2,175.61	50.76	2.02
	SetDE - 0.5	93.27	4.52	5,618.10	2,300.41	50.13	3.62
	SetDE - 0.9	94.87	4.97	4,961.50	1,822.18	52.82	1.85
Cylinder	SA	76.51	5.19	11,156.09	1,366.55	24.02	0.75
	BinDE - 0.3	83.10	1.83	9,156.70	378.93	82.07	1.49
	BinDE - 0.5	82.88	1.43	9,244.40	399.65	81.46	0.59
	BinDE - 0.9	79.01	1.29	9,558.40	1,593.50	77.33	3.41
	PathDE - 0.3	96.33	1.52	6,255.90	630.67	46.57	2.42
	PathDE - 0.5	94.67	1.90	6,069.30	661.25	44.09	3.01
	PathDE - 0.9	92.39	2.00	6,686.40	492.33	40.54	2.31
	SetDE - 0.3	88.78	4.32	7,982.90	1,222.62	44.81	3.73
	SetDE - 0.5	91.33	3.70	7,427.10	1,101.61	46.41	2.20
	SetDE - 0.9	92.11	2.06	6,765.50	731.43	47.54	2.10

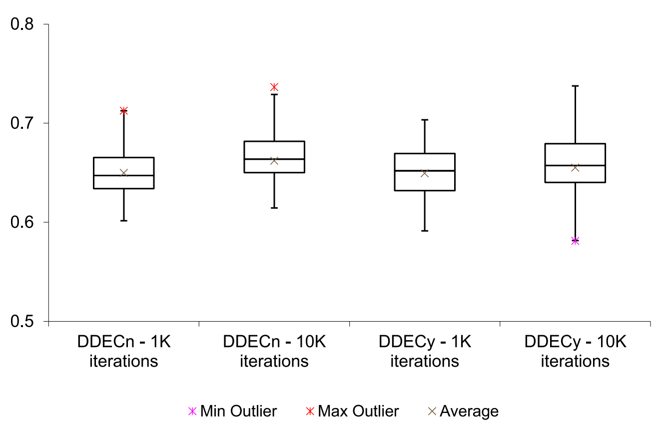


Fig. 8. Comparison of the average bud's illumination in DDECn and DDECy methods after repeating the main DDE loop for 1000 and 10,000 times, respectively. Maximum illumination value is 1.0.

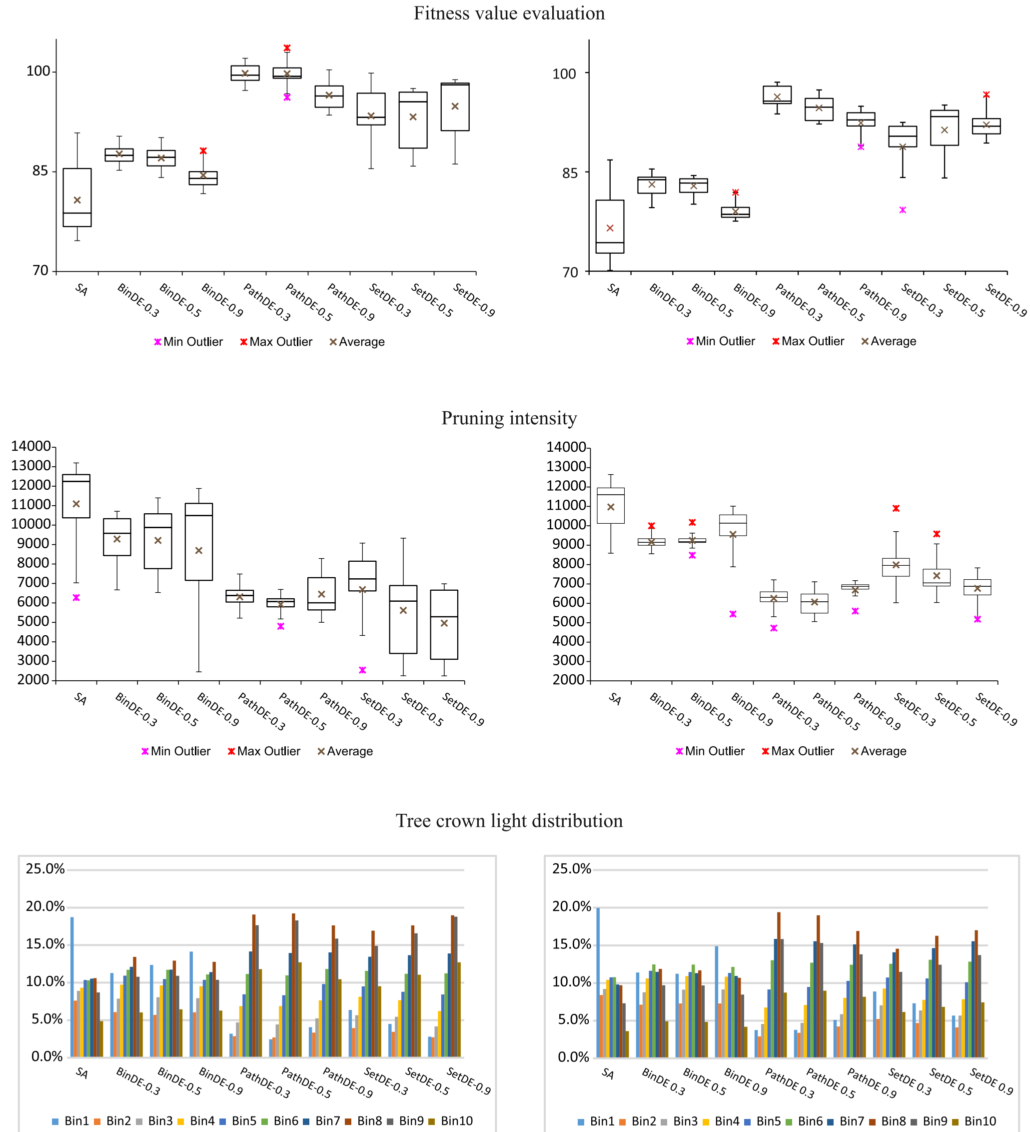
experiment and 0.658 (SD = 0.028) in the second one. The test showed that the results were similar, as depicted in Fig. 8.

Although the bud illumination in the second experiment was slightly higher, the difference is not high enough to significantly influence the pruning results. The analysis also showed that although the solutions differ, the small standard deviation indicates that the solutions are very close together, and the differences are only minor.

We also compared BinDE with other Differential Evolution methods. The closest methods to BinDE were Simulated Annealing algorithm (SA) and methods PathDE and SetDE [6]. During the tests, the population size by BinDE, PathDE, and SetDE were set to 20, while the stopping condition was the number of fitness evaluations (FE). In the first series of tests, FE was set to 1000 and later increased to 10,000 in the second series. In this series the most noticeable change was the increase of the performance of the SA method. However, due to the constant execution time of FE, the total computational time increased. For all three methods BinDE, PathDE, and SetDE, the parameters β and C were set to values $\beta = C = 0.3$, $\beta = C = 0.5$, and $\beta = C = 0.9$, respectively. Because the methods with 10,000 FE were too slow, in Table 3 only the results of the first test series are shown.

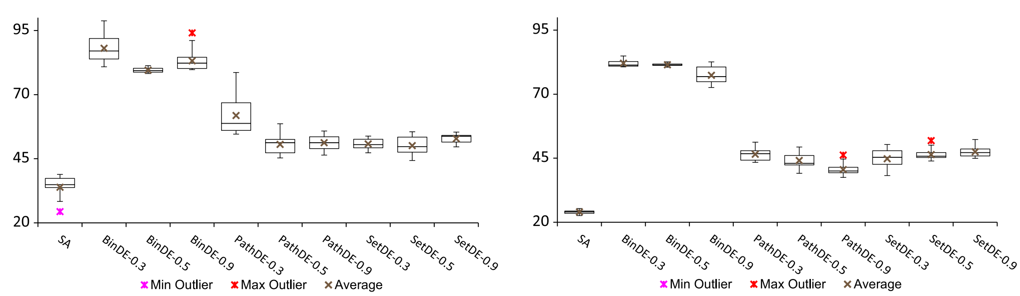
The methods were compared by using four criteria. First, we compared the obtained fitness values, see Fig. 9. Rather to compare the fitness values directly, we compared the heights of their Box plots. The methods with longer charts lack the stability and tend to damage the tree. For example, lower values of the Box plot by the pruning intensity reaching under value 3000 indicates that the central leader with the majority of the branches was removed and the tree was damaged. Such is the case by the SetDE-0.5 and SetDE-0.9 methods, with the first cone as a basic shape. Since this is not the case by the cylinder basic shape, we conclude that SetDE method by 1000 FE is sensitive to the selected basic shape. PathDE ($\beta = C = 0.5$) was the most efficient method, followed by BinDE with the same values of parameters, β, and C. In the pruning intensity category, we compared the number of internodes left after the pruning, where the pruning should improve the lighting conditions in the tree crown, but at the same time, it should not be too strong. Fig. 9 shows, that the weakest pruning is performed by the SA method, while the most intense pruning is obtained by the SetDE method ($\beta = C = 0.9$). None of these pruning can be used in orchards, as either the lighting conditions after pruning do not improve much or pruning is too intense. As the best methods again proved BinDE and PathDE, both with parameters β and C set to 0.5. The third criterion for comparison was the average light distribution intercepted by the tree crown. The most efficient method was the SetDE ($\beta = C = 0.9$), closely followed by PathDE method ($\beta = C = 0.5$). The last criterion for comparison was the computational time (see Fig. 10). The fastest algorithm was the implementation of the SA algorithm with an average running time of 24.02 [sec] by the cylinder as an initial pruning form and 33.9 [sec] by the cone. The slowest was BinDE by all the parameter settings and by both initial pruning forms. The average computational time ranged between 77.3 [sec] to 88 [sec].

Figs. 9 and 10 show that observed relations between the methods hold in all categories without regard to the initial pruning form we used. If we change the FE to 10,000, the pruning results



a) b)

Fig. 9. Comparison of tree pruning using algorithms SA, BinDE, PathDE, and SetDE with FE set to 1000, population size to 20, parameters β and C set to 0.3, 0.5, and 0.9, respectively in 10 runs for (a) cone and (b) cylinder basic shapes.



a) b)

Fig. 10. Comparison of calculation time for tree pruning by algorithms SA, BinDE, PathDE, and SetDE for (a) cone and (b) cylinder basic shapes.

of AS and SetDE improve significantly at the expense of increasing the computation time by factor 10. Concerning these tests, the findings by all four methods are comparable, but the advantage of methods BinDE and PathDE are the better pruning results by lower FE values.

Long-term Pruning: In another experiment, we evaluated a long-term exposure to pruning. We were curious if the proposed automated tree pruning method is capable of tree training (i.e., getting the tree into a desired growing form) without human intervention. We have simulated a row of five trees for six consecutive

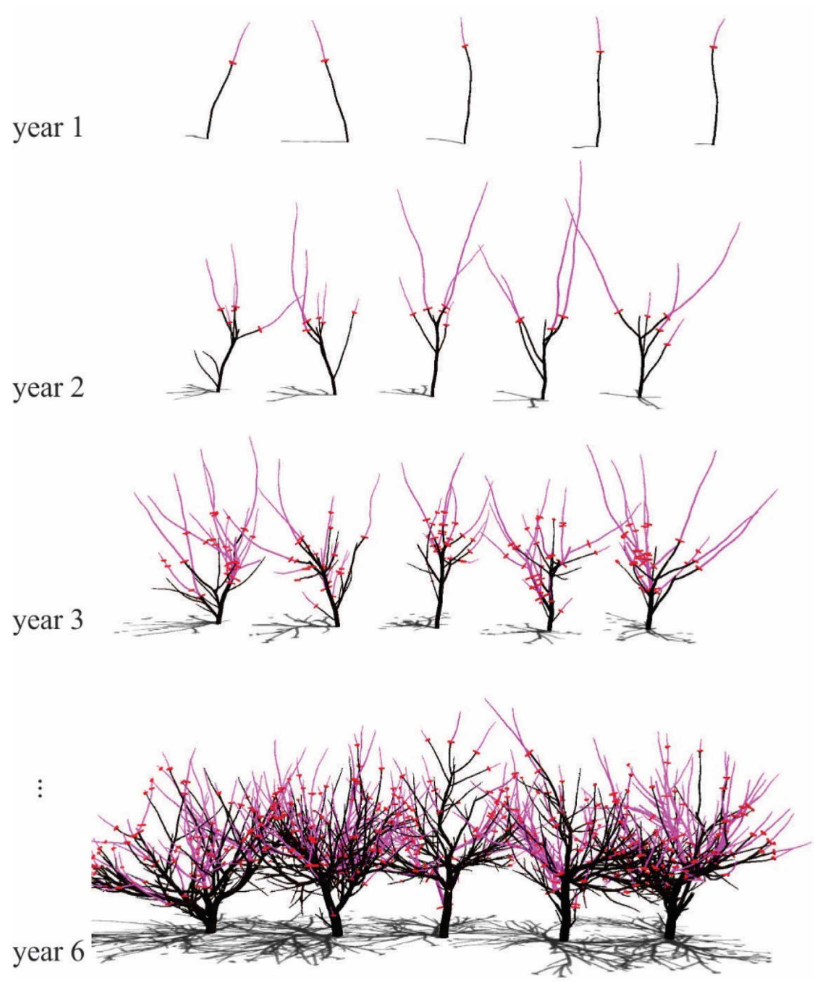


Fig. 11. Tree training of five apple trees into a Slender Spindle growing form for six consecutive years with the DDECn method. The purple color denotes the branches removed at the pruning.

years. At the beginning of each year, the trees were pruned by the DDECn method to shape them into the Slender Spindle growing form. The starting cone height was 1 m and was linearly increased to 2.5 m, which was the trees' target height in the following three years. The opening angle was constant at 45° for the experiment duration. The initial value of the parameter $s_{\max} = 20$ and was linearly increased to $s_{\max} = 70$ in the sixth year. The result of the experiment can be seen in Fig. 11.

As the tree structure becomes more complex in time, the value of $s_{\max}$ should be increased. In our experiments we observed that $s_{\max} \leq 150$ even for older trees provides reasonable results.

4. Conclusions and future work

We have introduced an automated method for simulation of pruning of trees and tree colonies. The objective was to propose pruning that maximizes light exposure of buds within the crown, and we used a two-step method, where the first step prunes the tree to a desired shape, and the second step maximizes the bud irradiance. Our results show that our algorithm achieves results comparable to the human pruning regarding the light distribution inside the tree crown. The pruning simulation of a group of trees for several consecutive years showed that the method could also be used for the tree training toward desirable growing form.

The possible limitation of the proposed method is its strong dependency on the tree growth model topological structure. This is important because the challenging step in the automated tree pruning is constructing the appropriate tree model from tree images. Successful algorithms of this kind are presented e.g, in [8–10,18]. However, they are not directly compatible with the EduAPPLE tree growth model. It would be essential to bring these algorithms to a common ground, for example, by generating the same tree representation. The construction of such an image would be even more desirable since the proposed pruning method shows pruning results comparable to human experts. Some work in this direction has already been done [39], but it is too early to assess its efficacy.

The main contribution of our work is to show that good pruning results can be obtained automatically and without a fixed set of pruning rules. This is not surprising since the pruning rules were developed by the long-term experiments whose goal was to improve yield quality and quantity. Once the main parameters that influence the yield were determined, the pruning techniques have been developed to maximize the influence of those parameters. That is what the DDE objective function is trying to mimic. Since light exposure is a critical factor, the proposed method searched for the combination of cuts that maximize light exposure. This resulted in the higher light distribution inside the tree crown. This distribution was also used to evaluate tree pruning since the obtained tree forms cannot be compared directly. While the shapes of the trees in Fig. 5 differ, their light distributions are comparable. An additional pruning benchmark is the number of internodes that corresponds to the tree volume. The amount of removed internodes represents the pruning intensity, and Fig. 6b shows that the human expert preferred slightly less aggressive pruning than the proposed method. The pruning intensity by the proposed automated pruning is controlled by the relation between tree volume and the value of the $s_{\max}$ parameter. To preserve the tree pruning intensity, as the tree is growing, the value of $s_{\max}$ should increase with the increasing number of internodes in the tree. In our experiments, $s_{\max} = 20$ turned out to be a good initial value.

There are many possible avenues for future work. As mentioned above, coupling our method with other plant growth algorithms would be an essential task to do. Another future work would be to use different optimization methods than the DDE. The long-term effect of growth should also be evaluated. It may be possible that bud illumination's immediate value is not the most indicative factor, and long-term plant health and development should be tested. Last but not least, our algorithm is a first step in an exciting direction of automated pruning. It would be interesting to automatically prune thousands of tree models and calculate statistics of the values we reported in Tables 1 and 2. Although, we paid attention to report representative models, there may be a substantial variation and statistics would provide a better representative value. Another extension of our work considers other properties of trees during the pruning process, such as overall tree stability, physical load of branches, proximity to other trees, etc. To speed up the convergence of the DDE method, it may be beneficial to introduce some implicit constraints to the objective function, for example, favoring the cuts on the branches' first internode. Our future research will also aim in this direction.

CRediT authorship contribution statement

Simon Kolmanič: Conceptualization, Methodology, Formal analysis, Writing - original draft, Visualization. **Damjan Strnad:** Methodology, Investigation, Validation. **Štefan Kohek:** Software, Data curation. **Bedrich Benes:** Writing - review & editing, Visualization. **Peter Hirst:** Writing - review & editing. **Borut Žalik:** Supervision.

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

The authors acknowledge the Project BI-US/17-18-012, and Programme P2-0041 were supported financially by the Slovenian Research Agency. This research was funded in part by the National Science Foundation, USA grant #10001387, *Functional Proceduralization of 3D Geometric Models*. This research was supported by 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

[1] M. Karkee, B. Adhikari, S. Amatya, Q. Zhang, Identification of pruning branches in tall spindle apple trees for automated pruning, Comput. Electron. Agric. 103 (2014) 127–135.

[2] T. Botterill, S. Paulin, R. Green, S. Williams, J. Lin, V. Saxton, S. Mills, X. Chen, S. Corbett-Davies, A robot system for pruning grape vines, J. Field Robotics 34 (6) (2017) 1100–1122.

[3] X. Zhang, H. Li, M. Dai, W. Ma, L. Quan, Data-driven synthetic modeling of trees, IEEE Trans. Vis. Comput. Graph. 20 (9) (2014) 1214–1226.

[4] H. Medeiros, D. Kim, J. Sun, H. Seshadri, S.A. Akbar, N.M. Elfiky, J. Park, Modeling dormant fruit trees for agricultural automation, J. Field Robotics 34 (7) (2017) 1203–1224.

[5] R.M. Bastias, L. Corelli-Grappadelli, Light quality management in fruit orchards: physiological and technological aspects, Chil. J. Agric. Res. 72 (4) (2012) 574.

[6] D. Strnad, S. Kohek, Novel discrete differential evolution methods for virtual tree pruning optimization, Soft Comput. 21 (4) (2017) 981–993.

[7] Š. Kohek, N. Guid, S. Tojnko, T. Unuk, S. Kolmanič, EduAPPLE: Interactive teaching tool for apple tree crown formation, HortTechnology 25 (2) (2015) 238–246.

[8] S.A. Akbar, N.M. Elfiky, A. Kak, A novel framework for modeling dormant apple trees using single depth image for robotic pruning application, in: Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, 2016, pp. 5136–5142.

[9] W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Měch, P. Prusinkiewicz, Self-organizing tree models for image synthesis, ACM Trans. Graph. 28 (2009) 58.

[10] P. De Reffye, C. Edelin, J. Francon, M. Jaeger, C. Puech, Plant models faithful to botanical structure and development, ACM SIGGRAPH Comput. Graph. 22 (1988) 151–158.

[11] H. Kang, M. Fiser, B. Shi, F. Sheibani, P. Hirst, B. Benes, Imapple—functional structural model of apple trees, in: Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA), International Conference on, IEEE, 2016, pp. 90–97.

[12] S. Pirk, O. Stava, J. Kratt, M.A. Massih Said, B. Neubert, R. Měch, B. Benes, O. Deussen, Plastic trees: interactive self-adapting botanical tree models, ACM Trans. Graph. 31 (4) (2012) 1–10.

[13] P. Prusinkiewicz, A. Lindenmayer, J. Hanan, Development models of herbaceous plants for computer imagery purposes, ACM SIGGRAPH Comput. Graph. 22 (1988) 141–150.

[14] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, B. Benes, Inverse procedural modelling of trees, Comput. Graph. Forum 33 (2014) 118–131.

[15] A. Bérut, H. Chauvet, V. Legué, B. Moulia, O. Pouliquen, Y. Forterre, Gravisensors in plant cells behave like an active granular liquid, Proc. Natl. Acad. Sci. (2018).

[16] R. Měch, P. Prusinkiewicz, Visual models of plants interacting with their environment, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1996, pp. 397–410.

[17] B. Benes, An efficient estimation of light in simulation of plant development, in: Computer Animation and Simulation, Springer, 1996, pp. 153–165.

[18] B. Benes, Visual model of plant development with respect to influence of light, in: Computer Animation and Simulation '97, Springer, 1997, pp. 125–136.

[19] M. Alsweis, O. Deussen, Modeling and visualization of symmetric and asymmetric plant competition, in: Eurographics, 2005, pp. 83–88.

[20] J. Arvo, D. Kirk, Modeling plants with environment-sensitive automata, in: Proceedings of Ausgraph'88, 1988, pp. 27–33.

[21] A. Runions, B. Lane, P. Prusinkiewicz, Modeling trees with a space colonization algorithm, in: NPH, Vol. 7, 2007, pp. 63–70.

[22] C. Soler, F.X. Sillion, F. Blaise, P. DeRreffye, An efficient instantiation algorithm for simulating radiant energy transfer in plant models, ACM Trans. Graph. 22 (2) (2003) 204–233.

[23] P. Wohlleben, The Hidden Life of Trees: What They Feel, How They Communicate—Discoveries From a Secret World, Greystone Books, 2016.

[24] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[25] C. Fu, Y. Liu, Z. Xiao, Interval differential evolution with dimension-reduction interval analysis method for uncertain optimization problems, Appl. Math. Model. 69 (2019) 441–452.

[26] J. Liang, W. Xu, C. Yue, K. Yu, H. Song, O.D. Crisalle, B. Qu, Multimodal multiobjective optimization with differential evolution, Swarm Evol. Comput. 44 (2019) 1028–1059.

[27] P. Sharma, H. Sharma, S. Kumar, K. Sharma, Black-hole gbest differential evolution algorithm for solving robot path planning problem, in: Harmony Search and Nature Inspired Optimization Algorithms, Springer, 2019, pp. 1009–1022.

[28] A. Zamuda, J.D.H. Sosa, Success history applied to expert system for underwater glider path planning using differential evolution, Expert Syst. Appl. 119 (2019) 155–170.

[29] A.A. Agrawal, A.P. Hastings, Plant defense by latex: Ecological genetics of inducibility in the milkweeds and a general review of mechanisms, evolution, and implications for agriculture, J. Chem. Ecol. 45 (11) (2019) 1004–1018.

[30] Z.A. Al-Sudani, S.Q. Salih, Z.M. Yaseen, et al., Development of multivariate adaptive regression spline integrated with differential evolution model for streamflow simulation, J. Hydrol, 573 (2019) 1–12.

[31] J.-W. Han, Q.-X. Li, H.-R. Wu, H.-J. Zhu, Y.-L. Song, Prediction of cooling efficiency of forced-air precooling systems based on optimized differential evolution and improved BP neural network, Appl. Soft Comput. 84 (2019) 105733.

[32] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[33] A.P. Engelbrecht, Computational Intelligence An Introduction, Jonh Wiley & Sons, Ltd., Chichester, England, 2007.

[34] D. Davendra, I. Zelinka, G.C. Onwubolu, R. Hill, Flow shop scheduling using clustered differential evolution, in: ECMS, 2009, pp. 70–76.

[35] Q.-K. Pan, M.F. Tasgetiren, Y.-C. Liang, A discrete differential evolution algorithm for the permutation flowshop scheduling problem, Comput. Ind. Eng. 55 (4) (2008) 795–816.

[36] L. Wang, Q.-K. Pan, P.N. Suganthan, W.-H. Wang, Y.-M. Wang, A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems, Comput. Oper. Res. 37 (3) (2010) 509–520.

[37] M. Weber, Optimizing the tree density in apple orchards on dwarf rootstocks, in: VII International Symposium on Orchard and Plantation Systems 557, 2000, pp. 229–234.

[38] T. Robinson, S. Hoying, M.M. Sazo, A. DeMarree, L. Dominguez, A vision for apple orchard systems of the future, N.Y. Fruit Q. 21 (3) (2013) 11–16.

[39] Š. Kohek, D. Strnad, B. Žalik, S. Kolmanič, Estimation of projection matrices from a sparse set of feature points for 3D tree reconstruction from multiple images, Period. Eng. Nat. Sci. 5 (3) (2017).