Geoscientific
Model Development

Locating trees to mitigate outdoor radiant load of humans in urban areas using a metaheuristic hill-climbing algorithm – introducing TreePlanter v1.0

Nils Wallenberg, Fredrik Lindberg, and David Rayner

Department of Earth Sciences, University of Gothenburg, Gothenburg, 413 20, Sweden

Correspondence: Nils Wallenberg (nils.wallenberg@gvc.gu.se)

Abstract. Mean radiant temperature (T_{mrt}) is a frequently used measure of outdoor radiant heat conditions. Excessive T_{mrt}, linked especially to clear and warm days, has a negative effect on human wellbeing. The highest T_{mrt} on such days is found in sunlit areas, whereas shaded areas have significantly lower values. One way of alleviating high T_{mrt} is by planting trees to provide shade in exposed areas. Achieving the most efficient mitigation of excessive T_{mrt} by tree shade with multiple trees requires optimized positioning of the trees, which is a computationally extensive procedure. By utilizing metaheuristics, the number of calculations can be reduced. Here, we present TreePlanter v1.0, which applies a metaheuristic hill-climbing algorithm on input raster data of T_{mrt} and shadow patterns to position trees in complex urban areas. The hill-climbing algorithm enables dynamic exploration of the input data to position trees, compared with very computationally demanding brute-force calculations. The hill-climbing algorithm has been evaluated with a static greedy algorithm that positions trees one at a time based on ranking and is expected to always find relevant locations for trees. The results show that the hill-climbing algorithm, in relatively low model runtime, can find positions for several trees simultaneously, which lowers T_{mrt} substantially. TreePlanter, with its two algorithms, can assist in optimization of tree planting in urban areas to decrease thermal discomfort.

1 Introduction

The increased risk of exposure to excessive heat in urban areas during extreme events as an effect of a modified urban climate can lead to excess mortality and morbidity (Dousset et al., 2010; Gabriel and Endlicher, 2011). The modified and generally warmer urban climate is a result of several factors, such as density of building stock, street orientation, color of materials, absence of permeable surfaces, anthropogenic heat and lack of vegetation (Arnfield, 2003). Mean radiant temperature (T_{mrt}, °C) is an important meteorological variable in the human energy balance and outdoor human thermal comfort, especially during clear and warm weather (Mayer and Höppe, 1987; Höppe, 1992; Mayer et al., 2008). T_{mrt} is the sum of all short- and longwave radiation fluxes (both direct and reflected) to which the human body is exposed, defined by ASHRAE (2001) as the "uniform temperature of an imaginary enclosure in which radiant heat transfer from the human body equals the radiant heat transfer in the actual nonuniform enclosure". High T_{mrt} has negative effects on human health (Mayer et al., 2008). Thorsson et al. (2014) showed that there is a higher correlation between T_{mrt} and mortality compared than between air temperature and mortality on hot days. High daytime T_{mrt} correlated with heat related mortality among people aged 80+, whereas high nighttime T_{mrt} correlated with heat related mortality among peopled aged 45–79.

Lateral longwave irradiance from surroundings is the largest component of T_{mrt} (Lindberg et al., 2014), and although this component does increase on clear days, high values of outdoor T_{mrt} only occur in sunlit areas exposed to direct shortwave irradiance from the Sun (Lindberg and Grim-

mond, 2011a). T_{mrt} peaks around noon and early afternoon when the Sun is at its highest position during the day. Areas in front of sunlit south-facing walls (Northern Hemisphere) are exposed to high radiant load (Lindberg et al., 2016; Wallenberg et al., 2020), due to high amount of shortwave irradiance from the Sun, but also reflected shortwave irradiance from adjacent walls, as well as emitted longwave irradiance from nearby warm surfaces (Lindberg et al., 2016; Thorsson et al., 2017). A proven method to tackle the issue of high daytime T_{mrt} on clear days is through shading by, e.g., buildings or vegetation (Lindberg and Grimmond, 2011b; Srivanit and Jareemit, 2020). Buildings are static and their geometries challenging to modify, whereas increasing the fraction of vegetation is a favored heat-mitigating strategy (Konarska et al., 2014; Norton et al., 2015; Lindberg et al., 2016).

Reducing solar radiation (e.g., Bajsanski et al., 2016; Stojakovic et al., 2020) and T_{mrt} (e.g., Konarska et al., 2014; Lindberg et al., 2016; Zhao et al., 2018; Abdi et al., 2020; Lee et al., 2020; Lee and Mayer, 2021; HosseiniHaghighi et al., 2020) with tree shade has been extensively studied. Studies of the effects of vegetation on thermal comfort often focus on either vegetation in general or more specifically on the positioning of trees. Zhao et al. (2018) studied the effects of trees on a neighborhood in Phoenix (Arizona, USA) and showed that two trees with equal distance had a higher effect on thermal comfort compared to two dispersed trees. Lee et al. (2020) analyzed the distance between trees in combination with tree canopy size and showed that the shading effect of trees increases with a lower aspect ratio (height / width ratio, H/W) of surrounding buildings. They also showed that a larger tree canopy in combination with an increased distance between the trees had a larger positive effect on thermal comfort on the northern sidewalk in an east–west-oriented street canyon, compared to the southern sidewalk. This effect was mainly attributable to a decrease in T_{mrt}. Trees had, on the other hand, little effect on air temperature regardless of whether the position was sunlit or shaded.

A common denominator within the topic of trees as a mitigating tool, regardless of whether the focus is on distance between trees or tree size, is that the tree or trees are located without knowledge of how they optimally should be placed with respect to reducing radiant load. The locations of the tree or trees have an effect on the thermal comfort (Abdi et al., 2020; Lee et al., 2020; Srivanit and Jareemit, 2020), and therefore optimized positioning could enhance the mitigating effect. One way of addressing this issue is to calculate the optimal positions of the trees using spatial information (insolation and shadow patterns) and the possible locations for the trees (i.e., where there are no buildings or other obstructing structures). However, brute-force calculations for optimal positions increase exponentially with an increase in number of trees and quickly become extremely computationally demanding. A different, less demanding method of finding optimal locations for trees is by exploiting metaheuristic algorithms. Examples of metaheuristic algorithms are genetic or evolutionary algorithms. Such algorithms inherit "genes", e.g., coordinates of positions of trees, from previous "populations", e.g., an iteration in an algorithm for positioning the trees. Stojakovic et al. (2020) utilized an evolutionary algorithm for positioning trees to mitigate insolation in a rectangular city block in Belgrade, Serbia, and clearly showed that locations differ depending on prerequisites like height of surrounding buildings. Chen et al. (2008) and Ooka et al. (2008) exploited genetic algorithms to study optimal positions for trees and tree arrangements using computational fluid dynamics simulations in a hypothetical urban block in Tokyo. Another example of a metaheuristic algorithm is the greedy algorithm, utilized by Zhao et al. (2017), to optimize tree locations to study shading effects and shade coverage on building facades. This algorithm finds the optimal position for one tree at a time, which means that once the position for the first tree is established it cannot be adjusted. A hill-climbing algorithm (Luke, 2013) is an additional example where neighboring positions of a tree are explored to identify a better position concerning, e.g., shading and reduction of radiant load.

Here, we present TreePlanter, a model for optimization of tree positions to mitigate heat stress by reducing outdoor radiant load in urban settings. The optimization of tree positions is achieved by utilizing a metaheuristic hill-climbing algorithm to reduce T_{mrt}. TreePlanter is incorporated in the Urban Multi-Scale Environmental Predictor climate service tool (UMEP; Lindberg et al., 2018) to facilitate usage by other researchers and practitioners such as urban planners and landscape architects (http://umep-docs.readthedocs.io/, last access: 1 February 2022).

2 Methods

TreePlanter make use of output data generated by the SOlar and LongWave Environmental Irradiance Geometry (SOLWEIG) model (Lindberg et al., 2008). SOLWEIG is a 2.5-D model able to estimate spatial variations of T_{mrt} using commonly available meteorological forcing data (incoming shortwave radiation, air temperature and relative humidity) and surface information such as a digital surface model (DSM) including the elevation of buildings and ground (e.g., Fig. 3b). Developments in SOLWEIG enable inclusion of 3-D vegetation data (Lindberg and Grimmond, 2011b) and variations in ground surface cover (Lindberg et al., 2016). SOLWEIG has been subject of evaluation in several studies (e.g., Lindberg and Grimmond, 2011b; Lindberg et al., 2016; Lau et al., 2016; Chen et al., 2016; Kántor et al., 2018; Gál and Kantor, 2020). Furthermore, SOLWEIG is a popular radiation model, utilized extensively in applied studies (e.g., Lindberg et al., 2014; Lau et al., 2015; Jänicke et al., 2016; Thom et al., 2016; HosseiniHaghighi et al., 2020).

TreePlanter uses the gridded T_{mrt} and shadow patterns output from SOLWEIG, as well as information on the locations

of buildings, and the meteorological forcing data used by SOLWEIG. Example outputs from SOLWEIG are shown in Fig. 3c and d. The figures show output of T_{mrt} and shadow patterns, respectively, for 17:00 LST on 22 June 1983. By comparing T_{mrt} with shadow patterns, it is evident that radiant load in shaded areas is substantially lower compared to sunlit areas. Output data from SOLWEIG are crucial for running the particular model described here. However, in theory, any raster data of T_{mrt} and shadow patterns could be used.

A planting area (study area) is defined inside the spatial extent of the output data from SOLWEIG. This planting area can cover either the entire spatial extent of the output data or be delimited to a smaller area to confine positioning of trees to areas within the model domain that are not occupied by buildings or existing trees. Furthermore, it is not possible to position trees within one radian of the tree canopy diameter from buildings or other trees. See Sect. 5.2 for further description.

2.1 Model pre-processing

A general and simplified flowchart of the model pre-processing is shown in Fig. 1a. First, a general 3-D tree form is designed from input tree morphology data: tree type (deciduous or conifer), tree height, trunk zone height (height from ground to canopy, i.e., bare trunk) and canopy diameter. The meteorological forcing data (same as for the SOLWEIG run) are used to estimate T_{mrt} for a point that is shaded by a tree with a transmissivity (τ) to shortwave radiation for specified time steps (specified by the meteorological data). Additionally, the specified time steps from the meteorological data are used to generate corresponding shadows in a fictitious flat environment (i.e., no buildings or trees), using the general 3-D tree shape produced from the input tree morphology data (hereafter referred to as shadows$_{tree}$).

With the generated shadows$_{tree}$ and calculated T_{mrt} for a point shaded by a tree for each time step, T_{mrt} in the tree shadows for each time step can be estimated (hereafter referred to as $T_{mrt.tree}$). By iterating over every position in the planting area where a tree can potentially be located and moving only one tree, it is possible to estimate the difference in T_{mrt} between $T_{mrt.tree}$ and sunlit conditions in the output from SOLWEIG (hereafter referred to as $T_{mrt.solweig}$) for each position for each time step. The comparison take into account shadows from surrounding buildings and existing vegetation (hereafter referred to as shadows$_{solweig}$) and their physical structures, i.e., pixels in $T_{mrt.tree}$ are removed where there are shadows from existing buildings or vegetation. In short, only sunlit pixels in $T_{mrt.solweig}$ that can be shaded by the generated tree for a given position are used in the comparison. The product from the comparison between $T_{mrt.solweig}$ and $T_{mrt.tree}$ is a new raster with an estimated difference in T_{mrt} between sunlit and tree shade (hereafter referred to as ΔT_{mrt}) for every position in the study area where it is possible to locate a tree. This raster gives a potential decrease in T_{mrt} and

is estimated from simply one tree, which means that the estimated potential decrease in T_{mrt} for each position is without any interference of tree shadows from any of the other trees, i.e., combination of several trees if the model is used to locate optimal positions for more than one tree. The output ΔT_{mrt} raster from the model initialization is used subsequently in the model to find optimal positions for several trees. To avoid possible spatial boundary effects, 5 % of the y extent and x extent are removed at all sides in the input data from SOLWEIG ($T_{mrt.solweig}$ and shadows$_{solweig}$).

2.2 Algorithms in the model

In the model initialization, described in Sect. 2.1, only one tree is used to calculate ΔT_{mrt} for every possible location. In theory, it would be possible to calculate ΔT_{mrt} for any number of trees for all possible combinations of locations. However, brute-force calculations with more than one tree would substantially increase the computational time as the number of combinations would increase exponentially. One way to avoid brute-force calculations, but still find suitable solutions within limited time or with limited computational power, is the use of metaheuristic algorithms (Luke, 2013). Metaheuristic algorithms are not guaranteed to find the best solution, but nevertheless, they are helpful when brute-force calculations are too extensive and, with a given number of iterations or amount of time, metaheuristic algorithms can lead to a satisfactory result to an optimization problem. The model described in this paper utilizes a hill-climbing algorithm (Luke, 2013) to find optimal positions, in combination with two ways of assigning starting positions for new trees: random and genetic (random and genetic starting positions are described in Sect. 2.2.2). The basic principle in TreePlanter, with a hill-climbing algorithm, is that the ΔT_{mrt} raster is explored stepwise for better positions, and if shadows$_{tree}$ of two or more trees overlap, adjustments are applied. This means that the ΔT_{mrt} raster can be explored freely, and the estimated difference in T_{mrt} at each position applies, only to be adjusted if there is an interference, i.e., overlapping tree shadows. In this way, the number of calculations is significantly reduced compared to the brute-force approach. The exploration of the ΔT_{mrt} raster is conducted until no better positions can be established for the trees, which means that the trees are in their local optimal positions based on their starting positions. After this, the model restarts a new iteration. The model is set to restart for i number of iterations. The iterations are necessary to initiate new starting positions for the trees. In this way, the trees can avoid finding the same local optimums. Here, a local optimum is defined as a position for a tree, where if it were in any of its neighboring positions, it would have a less mitigating effect on T_{mrt} (described in more detail in Sect. 2.2.1). Local optimums are, however, not necessarily the best positions in the ΔT_{mrt} raster, as there can be many local optimums within this raster. A large number of iterations will increase the number of unique starting positions
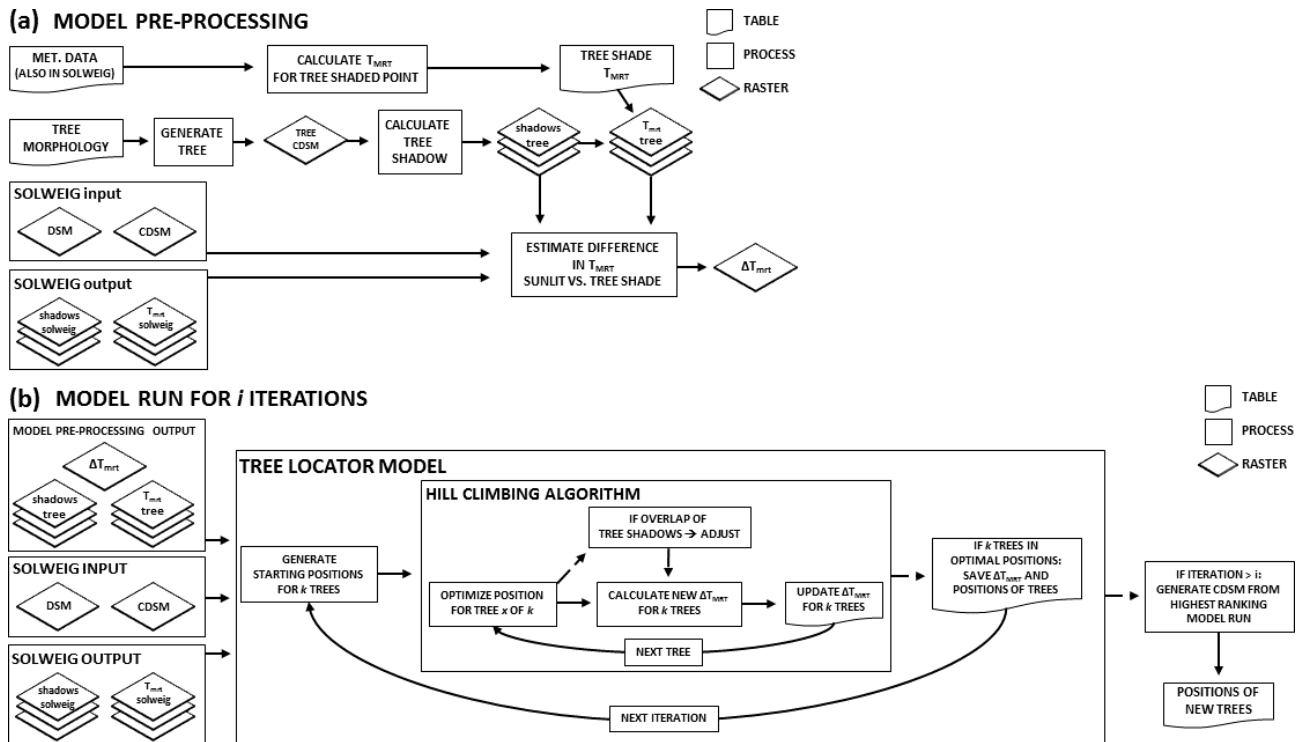
Figure 1. Flowchart of the model **(a)** showing the model pre-processing and **(b)** showing the optimization of tree locations. The flowchart consists of tables (e.g., tree morphology and meteorological data), processes (e.g., GENERATE TREE, which is an algorithm that generates a general 3-D tree form from input tree morphology table data) and raster grids. Some grids can have more than one layer, where each layer represents one time step (e.g., TREE SHADOW, where each layer represents shadow pattern for a specific time step). The end product is a table with the positions of the trees.

and combinations of starting positions by several trees. This in turn means that more of the ΔT_{mrt} raster and its local optimums will be explored.

The following is an example of how the hill-climbing algorithm can decrease number of calculations compared to brute-force calculations. An area with $n = 500$ possible locations for trees, where optimal positions for $k = 5$ trees are studied, would require $\frac{n!}{(n-k)!k!} = 2.5 \times 10^{11}$ brute-force calculations, considering all possible combinations. Using the hill-climbing algorithm, it is possible to run the model for a given number of iterations. Running it with, e.g., $i = 5000$ iterations, where in every iteration each tree would move imaginably $m = 100$ times, would estimate to approximately $k \cdot m \cdot i = 2.5 \times 10^6$ calculations, which is substantially faster in comparison to the 2.5×10^{11} required with brute-force calculations.

A flowchart of the second part of the model, the tree locator, is shown in Fig. 1b. Input data to the model are the $\Delta T_{\mathrm{mrt}} (\Delta T_{\mathrm{mrt}}$ in Fig. 1b), $T_{\mathrm{mrt.tree}}$ and shadows$_{\mathrm{tree}}$ rasters from the model initialization, as well as $T_{\mathrm{mrt.solweig}}$ and shadow$_{\mathrm{solweig}}$ rasters from SOLWEIG. Pseudocode for the algorithms in Sect. 2.2 is presented in Fig. 2. The model is run for i number of iterations as stated in line 4. Starting positions are determined in line 5 (see Sect. 2.2.2). The ΔT_{mrt}

raster is then explored until all trees are in their local optimums. In line 11, total potential decrease in T_{mrt} in the area shaded by the moving trees, integrated over all time steps used, is estimated. The tree locations will continue to change as long as T_{mrt} mitigation continues to increase in efficiency, as seen in lines 12–14. When there are no better locations for the trees and all are in local optimums the iteration stops, tree locations with corresponding decrease in T_{mrt} are saved (line 15–17), and a new iteration commences (line 4). When the model has finished all iterations, it returns the locations of the trees from the iterations with largest decrease in T_{mrt} (line 18).

2.2.1 An iteration of the hill-climbing algorithm

The fundamental procedure of the hill-climbing algorithm is that the model begins with k number of starting positions for k number of trees, and then cycles through these trees repeatedly to attempt to move trees to a better position. For each tree, the algorithm searches the adjacent eight pixels for a higher difference in T_{mrt} compared to the tree's current position and potentially moves the tree one pixel, and then performs the search for the next tree. When the search (and potential move) has been performed for all trees, the algorithm

```
1. trees = number of trees
2. iterations = number of iterations the model will be run
3. tmrt = decrease in Tmrt

4. for i in iterations:
5.    tree_positions = give trees starting positions (section 2.2.2)
6.    no_better_position = 0
7.    while no_better_position < 1:
8.       for moving_tree in trees:
9.          explore neighbouring positions of tree (section 2.2.1)

10.         check for possible overlap of tree shadows and make necessary adjustments (section 2.2.1)

11.         estimate sum of decrease in Tmrt of trees when moving_tree is in neighbouring positions (section 2.2.1)

12.         if decrease in Tmrt is larger when tree is positioned in any of the neighbouring positions (section 2.2.1):
13.            tree_positions = store new position for tree
14.            tmrt = store new larger decrease in Tmrt for trees

15.      if none of trees can find a better position:
16.         no_better_position = 1
17.         save tree_positions with corresponding decrease in Tmrt

18. return tree_positions for iteration with largest decrease in Tmrt (section 2.2.1)
```

Figure 2. Pseudocode of the tree locator algorithm in the model.

cycles over all trees again. In this way, the raster is explored until local optimums are determined for each tree in relation to the other moving trees. If two trees are in such a proximity to each other that their shadows overlap, the overlap is adjusted for; i.e., the decrease in T_{mrt} is counted only once. This means that shadows can overlap if this would provide a more favorable shading effect. When all trees have found their optimal positions for an iteration of the hill-climbing algorithm, it saves the positions and the corresponding decrease in T_{mrt}. A new iteration of the hill-climbing algorithm then commences with new starting positions (Sect. 2.2.2), and the model continues for i number of iterations. In the end of the model run, the iteration with greatest decrease in T_{mrt} is determined and the corresponding tree positions will be used as output.

If two or more tree shadows overlap, an adjustment of the decrease in T_{mrt} is necessary, or the decrease in the overlapping shadows would be accounted for more than once. Testing for potential overlaps is conducted in different ways in TreePlanter. A first test is executed by comparing distances between the trees, where large distances can rule out any possible overlap. Here, large distance is defined as the largest shadow created from a tree within the modeled time span. If distances, on the other hand, are close enough for potential overlap, additional calculations are executed to determine the exact number of overlapping pixels, from which adjustments of reduction in T_{mrt} are estimated.

Another functionality in the model, connected to adjustment, is nudging. Nudging is initiated if two or more shadows are overlapping or next to each other, creating a large continuous shadow. When initiated, it will try to move the

trees simultaneously in the same direction, to see if there are better positions in their vicinity. For example, if two trees have a combined continuous shadow, it will search their adjacent west pixels simultaneously, then search the northwest pixels, etc., until all eight adjacent directions have been explored. This is to prevent trees from occupying a position, when there are potentially better positions for them if one of them is relocated.

2.2.2 Starting positions for iterations

Two methods to derive starting positions are available in the model: random and genetic. Depending on the size of the model domain, number of starting positions can be extensive. For example, a 100×100 planting area would have $10\,000$ possible locations, and thus same amount of possible starting positions for the trees. Building shadows would largely influence the T_{mrt} for some pixels. Pixels where the decrease in T_{mrt} is zero, i.e., pixels that are already shaded, are excluded. Furthermore, trees cannot be within one tree canopy diameter of each other when they start, as this would mean that their canopies would be overlapping.

In the random algorithm for starting positions, each tree is assigned a random starting position in the ΔT_{mrt} raster at the beginning of each iteration. That is, if the model is executed with 3000 iterations, the trees will start with new random positions in each individual iteration. The algorithm for random starting positions was evaluated against a genetic algorithm for starting positions. Compared to the random starting algorithm, the genetic algorithm inherits starting positions from local optimums of the previous population. This means that

starting positions of the next model iteration will be based on the best positions of the previous iteration. In the first iteration, each tree is assigned a random starting position based on the ΔT_{mrt} raster. In the second iteration, the positions will be determined randomly from the y and x coordinates of the local optimal positions (local optimums) in the previous iterations. Here, a tree can have its y coordinate from one tree and x coordinate from another tree. Mutation of either the y or the x position will occur if trees have reached positions without improvement in decrease in T_{mrt} for three consecutive iterations. In this sense, mutation means that the y or the x position (randomly decided which one that will mutate) is set to a random position in the planting area. Mutation can also occur if the starting positions for the trees are too close to each other; i.e., the trees have converged to a very confined area and the starting position for a tree is within one diameter of the other trees 50 times in a row.

2.3 Summary of the model

This section gives a short summary of the model in chronological order:

1. SOLWEIG is executed for a given number of time steps based on meteorological data (incoming shortwave irradiance, air temperature and relative humidity) and other necessary gridded input data to simulate T_{mrt} ($T_{\mathrm{mrt.solweig}}$) and shadow (shadows$_{\mathrm{solweig}}$).

2. A planting area is determined within the spatial extent of the output data from SOLWEIG, e.g., a square in an urban area.

3. A tree form is generated based on tree height, canopy diameter and trunk zone height (height between ground and canopy).

4. The effect of the tree form in step 2 on radiant load is calculated ($T_{\mathrm{mrt.tree}}$ and shadows$_{\mathrm{tree}}$) based on the same meteorological data as in step 1. This is determined for a flat unobstructed area.

5. Based on the T_{mrt} and shadow patterns from step 1 ($T_{\mathrm{mrt.solweig}}$ and shadows$_{\mathrm{solweig}}$) and step 3 ($T_{\mathrm{mrt.tree}}$ and shadows$_{\mathrm{tree}}$) a difference in T_{mrt} is estimated (ΔT_{mrt}) for each position in the planting area.

6. The user decides the number of trees (k) to optimize locations for, with respect to T_{mrt} mitigation.

7. Give each tree a random starting position in the gridded planting area.

8. Execute hill-climbing algorithm. Adjust for potential overlap by any of the k trees.

9. Save optimal positions of trees and corresponding decrease in T_{mrt}.

10. Repeat from step 7 for i number of iterations, with either the random or genetic starting algorithm, where i number of iterations is set by the user.

11. Review the output of tree positions for iteration with greatest decrease in T_{mrt}.

2.4 Greedy algorithm

As explained in Sect. 2.2, the best solution cannot be known unless brute-force calculations are applied. Therefore, the hill-climbing algorithm has been evaluated against a "greedy" algorithm. Zhao et al. (2017) used a greedy algorithm for strategic tree placement for optimized tree shade coverage to decrease solar exposure on facades and in turn lower building energy use. The results by Zhao et al. (2017) showed that the greedy algorithm determined locations where tree shade coverage was optimized, while simultaneously had low negative effect of tree shade on rooftop solar panels. The greedy algorithm is elemental in that optimal positions for trees are determined one at a time, based on the ΔT_{mrt} raster described in Sect. 2.1. When a position is determined for a tree, this position and the pixels covering the canopy and a surrounding buffer of one radian of the tree canopy are occupied. Furthermore, spatial patterns of $T_{\mathrm{mrt.solweig}}$ and shadows$_{\mathrm{solweig}}$ are updated to include spatial T_{mrt} and shadow patterns of the newly added tree, which means that a tree shade of a subsequent tree will not have a mitigating effect in those areas. The greedy algorithm can be described as ranking positions, where the first tree will be located in the position where the highest potential decrease in T_{mrt} can be achieved, the second tree in the second ranking position (taking into account the shading from the first tree), etc. Tree locations determined by a greedy algorithm for mitigation of T_{mrt} are used here to evaluate tree locations determined by the hill-climbing algorithm, as the greedy algorithm is expected to always find tree locations with high potential decrease in T_{mrt}.

A short summary of the greedy algorithm is as follows:

1. Run steps 1–5 in Sect. 2.3.

2. Determine best position in ΔT_{mrt} for one tree.

3. Update $T_{\mathrm{mrt.solweig}}$ and shadows$_{\mathrm{solweig}}$ to include $T_{\mathrm{mrt.tree}}$ and shadows$_{\mathrm{tree}}$ based on the position in step 2. Remove all positions within one canopy diameter of the tree location as a future possible location for a tree.

4. Recalculate ΔT_{mrt} with the updated $T_{\mathrm{mrt.solweig}}$ and shadows$_{\mathrm{solweig}}$.

5. Repeat from step 2 for k trees.

Table 1. Table with different tree sizes used for model evaluation.

Tree size	Tree height (m)	Canopy diameter (m)	Trunk zone height (m)	Transmissivity (τ)
Small	5	3	2	0.03
Medium	8	5	2	0.03
Large	12	7	3	0.03

3 Model evaluation

3.1 Model domain and forcing data

The case study area is located in Gothenburg, Sweden, centered on the square Järntorget (Fig. 3), a hub for public transport (latitude 57.6997°, longitude 11.9530°). It is a hub for public transport with surrounding blocks occupied by restaurants and small shops, thus making it one of the busiest areas in Gothenburg. The southern part of the square was selected for the model evaluation. The square is intersected by tram tracks seen in the northern and western parts of Fig. 3a, with bike lanes in the eastern part. Within the potential planting area, there is a fountain, which is excluded as a possible location for new trees. The input geodata consist of a DSM, a DEM (digital elevation model) including ground heights and a CDSM (canopy digital surface model) including vegetation height. Only vegetation higher than 2 m is present in the CDSM (Fig. 3b). All gridded geodata have a pixel resolution of 1 m. The geodata originate from the Building and Planning Office in Gothenburg. Hourly meteorological data (shortwave radiation, air temperature and relative humidity) used were from the nearby Swedish Meteorological and Hydrological Institute's weather station no. 92513 (WMO 2513). The meteorological data represent a typical clear summer day in Sweden close to the summer solstice (22 June 1983).

3.2 Tree locations generated by TreePlanter

The model was evaluated for 09:00–16:00 LST and 13:00–16:00 LST for three different tree sizes (see Table 1 for description of tree sizes), as well as an evaluation with four, five and six trees with tree size large (Table 1). The first period (09:00–16:00 LST) is used to test the model over a longer time span with solar azimuth shifting from east to west. The second period (13:00–16:00 LST) covers a time interval when excessive radiant load and heat stress is in general most pronounced, when solar radiation potentially is high, and surrounding heated surfaces emit large amounts of longwave radiation.

The positions of the trees were determined using the genetic algorithm and 20 000 iterations. Running the model with 20 000 iterations ensures that tree locations are satisfactory (see Sect. 4). The rasters with mean T_{mrt} were produced by running SOLWEIG but with an updated CDSM containing the optimized tree locations as well as updated sky-view factors (SVFs). The difference maps were produced by comparing SOLWEIG outputs of T_{mrt} with and without the optimized trees, i.e., before (sunlit) and after (shaded). In addition, summary statistics have been calculated by running SOLWEIG with updated SVF and CDSM including the trees established by TreePlanter.

3.2.1 Optimal locations for trees with different sizes during 09:00–16:00 LST

Figure 4a shows the location of five small trees over the period 09:00–16:00 LST. TreePlanter locates the trees close to the west building. Since the south-facing facade of the west building is sunlit most of the time steps during the studied time period, this is where the model identifies optimal positions for the trees. The relatively small trees also cast relatively small shadows, which allow the trees to be located close to each other, as well as close to the building. This results in approximately evenly spaced trees aligned along the south- and east-facing facades, where their shade allows for a decrease in average T_{mrt} of up to 23 °C (Fig. 4b).

Locations for medium trees (Table 1) are shown in Fig. 4c. Here, the trees are more scattered, and optimal positions are established between the two buildings. The area in front of the south-facing facade of the west building is now shaded by only one tree, compared to the previous example, where this area was shaded by two to three trees. Continuing, the area in front of the east-facing facade is now less shaded than in the previous example. On the other hand, there is now an increased shaded area extending to the east building, because of the greater tree size with their corresponding increased tree shades (Fig. 4d).

When tree size is increased further (large, Table 1), the trees are dispersed even more (Fig. 4e). The area in front of the south-facing facade of the west building is now less shaded than in previous examples, and the area in front of the east-facing facade is barely shaded at all. Furthermore, two trees are at the northern border, shading areas outside the planting area. Another two trees are shading parts of the fountain. This results in a mitigating effect outside the planting area (Fig. 4f).

3.2.2 Optimal locations for trees with different sizes during 13:00–16:00 LST

The model was also run for 13:00–16:00 LST, with same tree sizes as in Fig. 4, to study the tree locations during this time period. The corresponding results are presented in Fig. 5.

The results from the model run with small trees (Table 1) are shown in Fig. 5a. One striking difference compared to Fig. 4a is that all trees except one are concentrated around the east building instead of the west building. One tree ended up shading the area in front of the south-facing facade of the
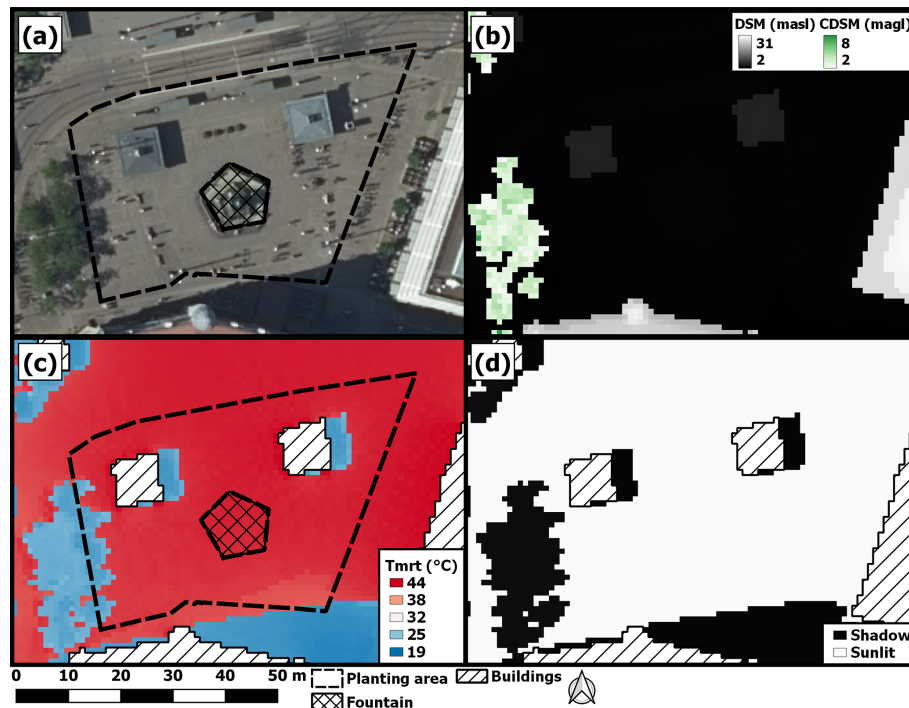
Figure 3. Map of case study area with **(a)** Orthophoto RGB 0.25 m © Lantmäteriet (2018), where the dashed black line determines the planting area for possible tree locations and **(b)** DSM and CDSM. Panels **(c)** and **(d)** show examples of output from SOLWEIG, where panel **(c)** indicates T_{mrt} and **(d)** indicates the shadow pattern for 17:00 LST on 22 June 1983.

west building. In Fig. 5b, it is possible to see a decrease in average T_{mrt} of up to 26 °C for almost all areas shaded by the trees.

When tree size is increased to medium, all trees end up in the eastern part of the study area (Fig. 5c). The locations established by the model provide shade around the entire previously sunlit area of the east building, and a large decrease in T_{mrt}, as seen in Fig. 5d.

In the last example, using large tree size (Fig. 5e), the trees are more or less in the same locations as in Fig. 5c, and similar assumptions can be made as in the previous example. However, they are now positioned slightly further south, as their shadows extend further because of the higher tree height. It is also visible that they are farther apart, as the diameter of the tree canopy is larger, and thus the tree shade increases in width as well. In this example, however, the area shaded by the trees is now continuous for all trees (Fig. 5f), compared to Fig. 5d. That is, the area where the tree shadows have an effect covers almost all areas along the southern and western sides of the east building.

3.2.3 Optimal locations depending on number of trees

One aim with TreePlanter and the hill-climbing algorithm is to see if the combined and continuous shadow of several trees could influence the positioning of trees. In other words, can, e.g., the combined and continuous shadow of two trees shade

an area equivalent to that shaded by one tree but from different positions? To investigate this, the model was executed with varying number of trees (four, five and six) with the large tree size (Table 1). The results from the model run with four trees (Fig. 6a) show that the tree shading the area in front of the southwest-facing corner in Fig. 5e is now missing. Furthermore, the westernmost tree is located slightly more south. Other than that, no large difference is visible. The missing tree, of course, has an effect on the amount of area with a decrease in T_{mrt}, as seen in Fig. 6b. When increasing the number of trees to six, it is possible to see that the three trees shading the south- and west-facing facades (Fig. 6c) are in same positions as in Fig. 5e. However, when comparing with Fig. 6a, only one tree is in the same position (the central one in Fig. 5a, shading the south-facing facade). All other trees are in different positions. The trees in the western part, along the border of the planting area, are now located further south and only the very north part of this border is still sunlit. For all model runs, it is possible to see a large decrease in T_{mrt} (Figs. 5f, 6b and d).

4 Model performance

The model performance and runtime are dependent on a combination of spatial extent and the pixel resolution of the study area, i.e., the number of model grid points within a
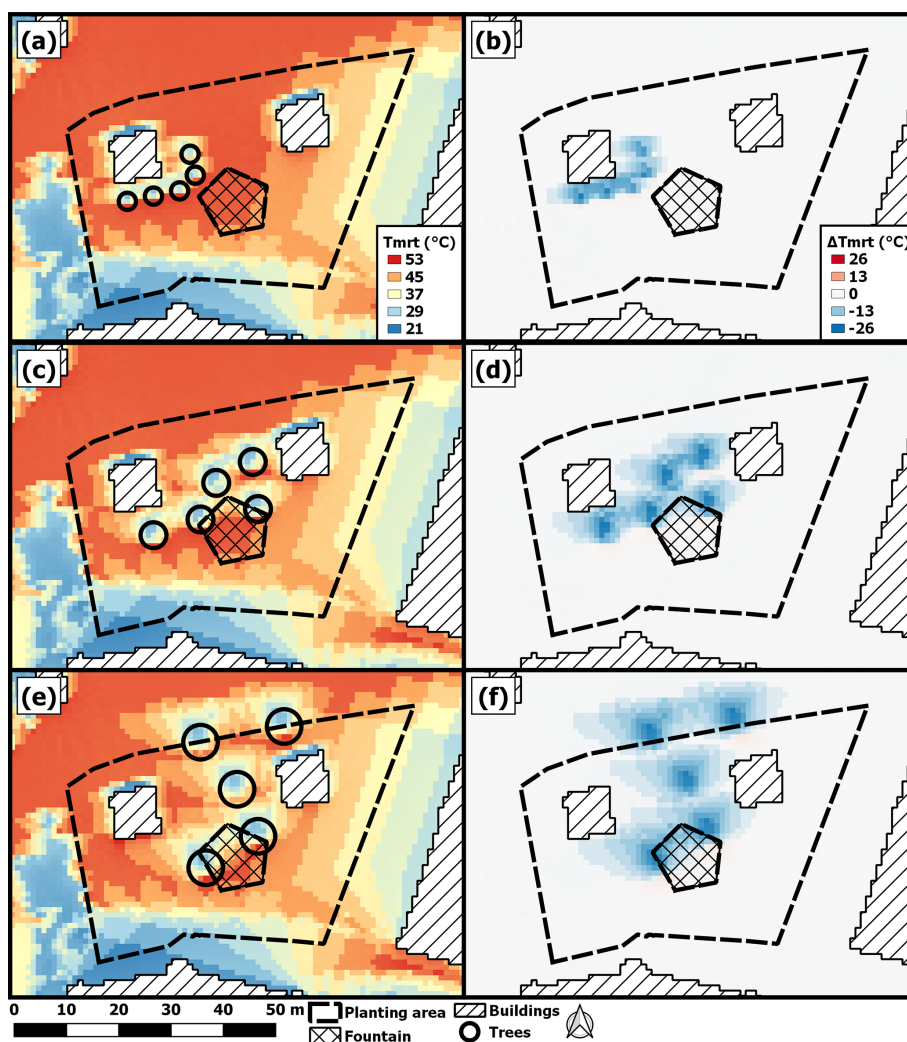
Figure 4. Mean T_{mrt} for 09:00–16:00 LST on 22 June 1983, with locations of **(a)** five small trees, **(c)** five medium trees and **(e)** five large trees in green (see Table 1). Panels **(b)**, **(d)** and **(f)** show corresponding differences in T_{mrt} between tree shade and sunlit areas for panels **(a)**, **(c)** and **(e)**, respectively. The positions are determined with the genetic starting algorithm and 20 000 iterations.

domain. Factors such as tree size, number of trees and time steps also affect model performance. The model was executed with the same model domain and forcing data as in Sect. 3.1, but with varying tree size, number of trees, time steps, model iterations and domain pixel resolution, to investigate these dependencies. In the performance analysis, a ratio of the potential decrease in T_{mrt} between the hill-climbing algorithm and the greedy algorithm was used to quantify the mitigation benefits of the hill-climbing algorithm. The model performance tests were executed on an Intel Core i7-7700 CPU at 3.60 GHz with 16 GB RAM at 2400 MHz. Figure captions include number of potential locations for trees, which differs depending on tree size due to the fact that a tall tree generates a more extensive ground shadow compared to a short tree. Mean model runtimes (s) for all model perfor-

mance tests are presented in Table 2. Initialization time is excluded and disk I/O is negligible.

4.1 Tree size

All three tree sizes in Table 1 were used for the performance analysis on tree size. The model was run with five trees for each tree size, and both starting algorithms (random and genetic) and 10, 100, 500, 1000, 2000, 3000, 10 000 and 20 000 iterations.

The results from the performance analysis on tree size for seven hourly time steps (09:00–16:00) LST are shown in Fig. 7. The results indicate that the hill-climbing algorithm had established positions with high mitigating potential already after five iterations. After 100 iterations, all trees are at positions similar to or better than the greedy algorithm, regardless of starting algorithm. However, some divergence
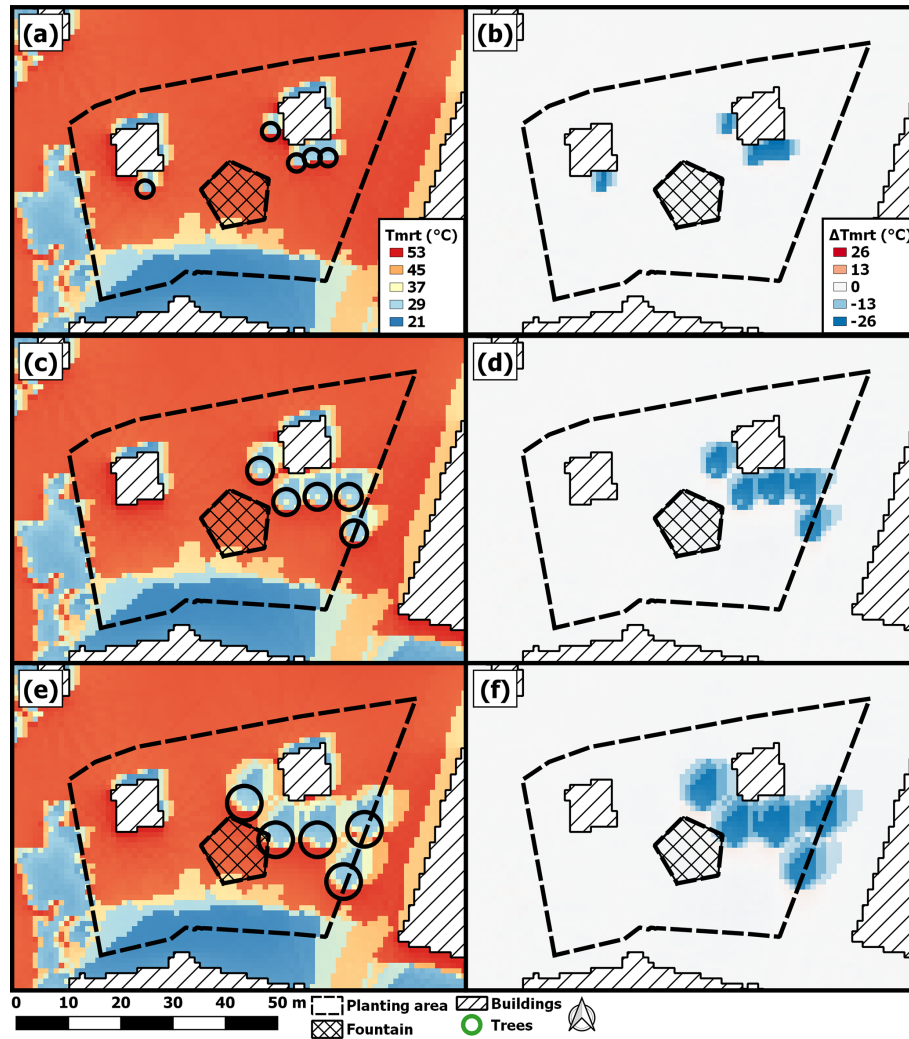
Figure 5. Mean T_{mrt} for 13:00–16:00 LST on 22 June 1983, with locations of **(a)** five small trees, **(c)** five medium trees and **(e)** five large trees in green (Table 1). Panels **(b)**, **(d)** and **(f)** show corresponding differences in T_{mrt} between tree shade and sunlit areas for panels **(a)**, **(c)** and **(e)**, respectively. The positions are determined with the genetic starting algorithm and 20 000 iterations.

can be seen, for example, for tree size small and 3000 iterations.

It is noticeable in Table 2 that a larger tree size decreases the speed of the model, but with five trees the genetic starting algorithm is faster compared to the random starting algorithm. The difference in model runtime between the two starting algorithms also increases with tree size, from -9.7% with small trees to -33.3% with large trees.

4.2 Number of trees

An increase in number of trees increases the complexity of the model and influences the performance and speed as the probability of overlapping tree shadows would increase, regardless of tree size. The model was executed with two, three, four, five, six, seven and eight trees with tree size large (Table 1), with three time steps (13:00–16:00 LST), with cor-

responding results presented in Fig. 8. As illustrated, the potential decrease in T_{mrt} is high after 5–10 iterations, and after 500 iterations mitigation benefits for position with the genetic starting algorithm always exceed those from the greedy algorithm. There is also a tendency for a higher ratio with a higher number of trees.

Model speed decreases with an increase in number of trees for both random and genetic conditions, as shown in Table 2. The speed performance of the genetic algorithm outcompetes the random algorithm for all runs except those with six, seven and eight trees. With six, seven and eight trees model runtime increases, for eight trees quite extensively with a difference of around 150 % between the random and genetic with 20 000 iterations.
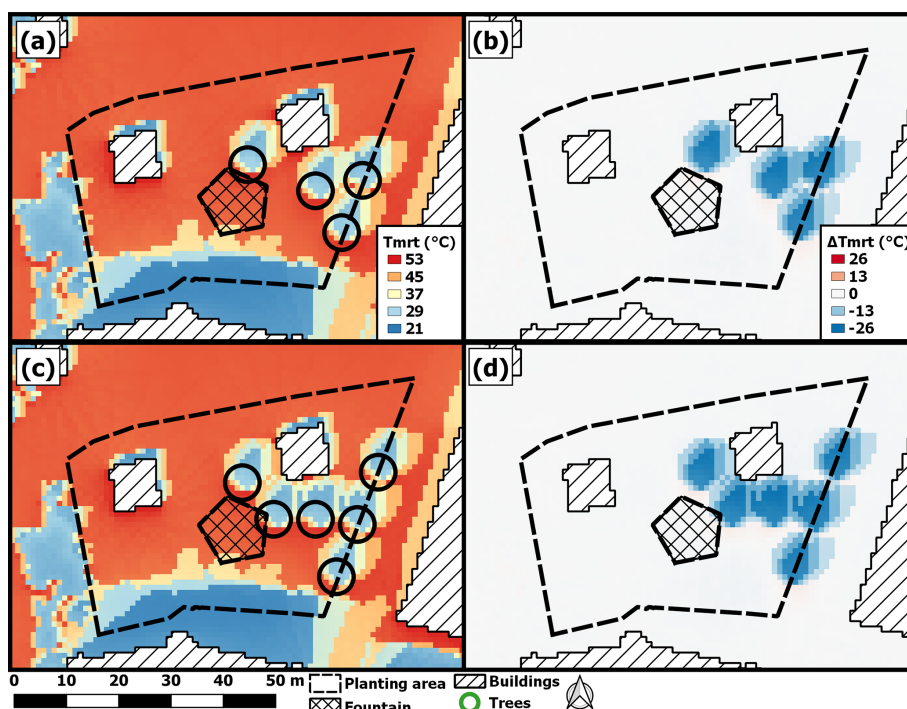
Figure 6. Mean T_{mrt} for 13:00–16:00 LST on 22 June 1983, with locations of **(a)** four large trees and **(c)** six large trees in green (see Table 1 for detailed tree sizes). Panels **(b)** and **(d)** show corresponding differences in T_{mrt} between tree shade and sunlit areas for panels **(a)** and **(c)**. The positions are determined with the genetic starting algorithm and 20 000 iterations.

Table 2. Table showing mean model runtime (s) for 100 iterations for two starting algorithms (random and genetic) with different time periods, number of trees and tree sizes. The difference in model runtime (%) corresponds to a change in model runtime with the genetic starting algorithm compared to the random algorithm. A negative (positive) value corresponds to a decrease (increase) in model runtime with the genetic starting algorithm.

Time period (LST)	Trees	Tree size	Random (s)	Genetic (s)	Difference (%)
09:00–10:00	5	Small	8.4	4.9	−42.1 %
09:00–10:00	5	Large	10.0	7.7	−23.3 %
13:00–16:00	5	Small	9.7	6.1	−36.9 %
13:00–16:00	5	Medium	11.6	8.3	−28.7 %
13:00–16:00	5	Large	11.8	9.6	−19.1 %
09:00–16:00	5	Small	14.1	12.7	−9.7 %
09:00–16:00	5	Medium	26.5	19.3	−27.0 %
09:00–16:00	5	Large	38.8	25.9	−33.3 %
13:00–16:00	2	Large	1.7	1.0	−40.5 %
13:00–16:00	3	Large	4.1	2.7	−34.1 %
13:00–16:00	4	Large	7.6	5.3	−30.2 %
13:00–16:00	5	Large	11.8	9.6	−19.1 %
13:00–16:00	6	Large	16.7	17.7	6.5 %
13:00–16:00	7	Large	22.0	36.0	63.9 %
13:00–16:00	8	Large	29.0	72.7	150.6 %

4.3 Time steps

The model was also tested for different time steps – one (09:00–10:00 LST), three (13:00–16:00 LST) and seven

(09:00–16:00 LST) – to analyze performance and speed, with two different tree sizes: small and large (Table 1).

The time-step performance analysis for small trees (Fig. 9a), similar to previous examples, found positions with high potential decrease in T_{mrt} relatively fast, after 100–500
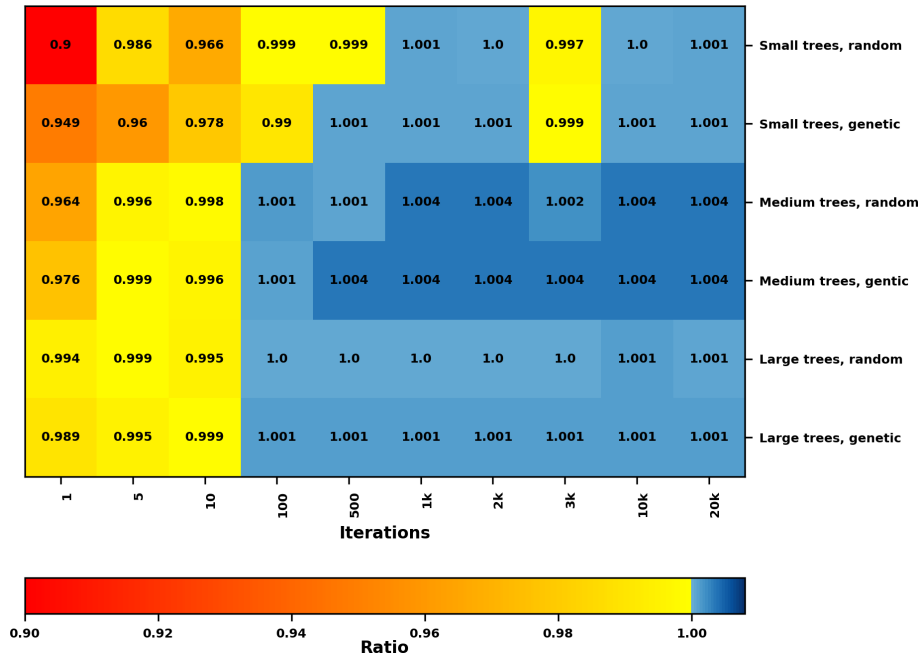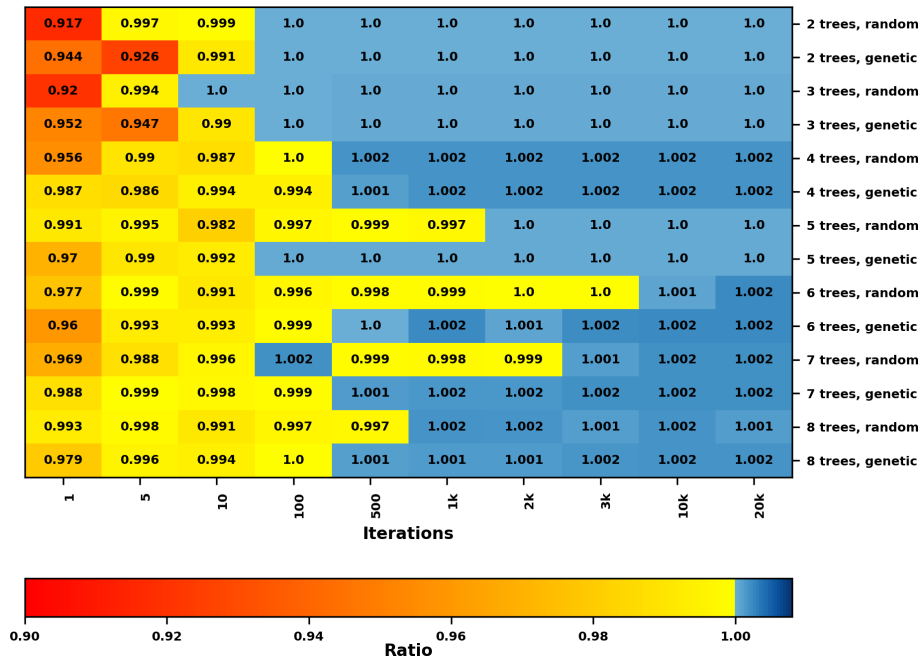
Figure 7. Heat map showing the ratio of potential decrease in T_{mrt} between the hill-climbing algorithm and the greedy algorithm for the three different tree sizes (small, medium and large; see Table 1) for different numbers of model iterations, for 09:00–16:00 LST on 22 June 1983. Two starting algorithms were used: random and genetic. Each model run was executed with five trees. There are 1709 potential locations for small trees, 1603 for medium trees and 1481 for large trees. A ratio > 1 indicates a larger T_{mrt} decrease with the hill-climbing algorithm. Note the cell color where ratios ∼ 1.0 is determined using an extended number of decimal places.

Figure 8. Heat map showing the ratio of potential decrease in T_{mrt} between the hill-climbing algorithm and the greedy algorithm for two, three, four, five, six, seven and eight large (Table 1) trees for different numbers of model iterations, for 13:00–16:00 LST on 22 June 1983. Two starting algorithms were used: random and genetic. There are 1481 potential locations for trees. A ratio > 1 indicates a larger potential decrease with the hill-climbing algorithm.

iterations, and with the genetic starting algorithm, positions are always better compared to the greedy algorithm after 500 iterations (ratio > 1.0), with the exception of 3000 iterations at 09:00–16:00 LST. For the random starting algorithm, however, some anomalies were found for 09:00–10:00 and 09:00–16:00 LST and 2000 and 3000 iterations.

Considering speed with the small tree size, there is a large increase in model runtime for the longer time period (09:00–16:00 LST) compared to the two other model runs as seen in Table 2 (09:00–10:00 and 13:00–16:00 LST). Furthermore, differences in speed between genetic and random algorithms are highest with one time step (−42.2 %).

The time-step performance analysis for large trees (Fig. 9b) found adequate positions after 100 iterations, but similarly to the small trees, there are some anomalies with the random starting positions, e.g., for 100 iterations at 09:00–10:00 LST and 100, 500 and 1000 iterations at 13:00–16:00 LST.

When analyzing model runtime, it is clear that this increases with number of time steps, similar to that of small trees. However, the largest difference between starting algorithms is with seven time steps (09:00–16:00 LST, −33.3 %).

4.4 Model domain size

Model performance analysis was conducted with special attention to model domain size, here represented by changing pixel resolution. Four different pixel resolutions were evaluated: 2, 1, 0.5 and 0.25 m. The pixel resolution was tested for three trees with tree height = 10 m, canopy diameter = 5 m and trunk zone height = 3 m. As expected with a 2-D modeling approach, the model runtime increased exponentially with higher pixel resolution, from 17 s with a 2 m pixel resolution to 44, 185 and 1290 s for 1, 0.5 and 0.25 m, respectively.

4.5 Tree locations – hill-climbing algorithm vs. greedy algorithm

As shown in the results for the performance analysis, the hill-climbing algorithm, with a high number of iterations, gives an equal or marginally higher potential decrease in T_{mrt} compared to the greedy algorithm. This suggests that the resulting locations for trees are different in the two algorithms. This is illustrated in Fig. 10. The examples from the hill-climbing algorithm are with 20 000 iterations, for which the hill-climbing algorithm always determined positions that had higher potential decrease in T_{mrt} than the greedy algorithm.

In all cases, some trees are in the same locations for both the hill-climbing and greedy algorithms, but the hill-climbing algorithm clusters the trees more closely together than the greedy algorithm. In Fig. 10a, the locations for the two western trees are the same, two central trees are only slightly different, but the greedy algorithm places a tree near the east building rather than near the western building. Similar observations can be made in Fig. 10b–d, with some locations the same, some similar, and one tree placed differently. In Fig. 10d, the most interesting difference is in the eastern part of the planting area. Here, the greedy algorithm finds an optimal position in between the two trees found by the hill-climbing algorithm and places a tree by the west building instead. Comparable results are visible in Fig. 10e, where one of the tree locations by the greedy algorithm is at the very northwest corner of the planting area.

A summary of statistics for the different model runs is presented in Table 3. Here, SVF has been recalculated using a CDSM including the new trees, after which the simulation of T_{mrt} has been repeated. This simulation is compared to the previous simulation where new trees are excluded, i.e., the original simulation. The results from the model runs (Figs. 7–9) indicated that the hill-climbing algorithm, with 20 000 iterations, had higher potential decrease in T_{mrt} compared to the greedy algorithm in all cases. Here, on the other hand, results are different. Comparing the results of the greedy algorithm with the results of the hill-climbing algorithm, the tree shade for locations based on the hill-climbing algorithm has a larger decrease in average T_{mrt} in its shaded area ($\Delta °C$ in shadow; °C) with small trees for 09:00–16:00 LST, small trees for 13:00–16:00 LST and large trees for 13:00–16:00 LST. However, locations found with the hill-climbing algorithm always have a larger average decrease per shaded area ($\Delta °C$ shadow area^{-1}; °C m^{-2}) as a result of fewer pixels shaded but with a similar decrease in its shaded area as that of the trees positioned with the greedy algorithm. Total change in average T_{mrt} for the entire raster (raster $\Delta °C$; °C) between simulations differs marginally when locations are based on the hill-climbing versus greedy algorithm. This difference, which is always larger compared to the difference calculated exclusively for the shaded area, is the result of changes in SVF in the vicinity of the new trees and has an effect on radiation. Examining the total change in average T_{mrt} for the entire raster per canopy area ($\Delta °C$ canopy area^{-1}; °C m^{-2}), it is evident that the decrease in T_{mrt} is very similar regardless of tree size.

4.6 Occurrences of tree positions

Occurrence maps, showing where trees were located after each iteration in a model run with the random starting algorithm and 20 000 iterations, are shown in Fig. 11. The preferred positions are relatively warm, where mitigation from tree shade is high. The general pattern is that preferred positions lie in arcs around one tree diameter from buildings, along the northern and eastern borders of the planting area, or close to a fountain. Within these regions, there are a few highly preferred pixels. Furthermore, south of the buildings, two rows of preferred positions can be seen in almost all figures. The highest occurrence is for small trees in front of the west building (72.6 %).
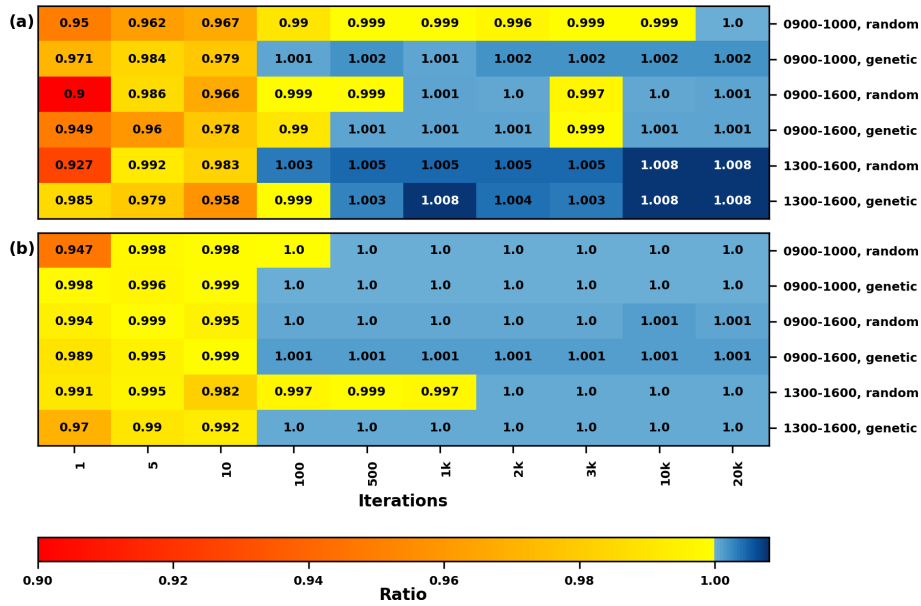
Figure 9. Heat maps showing the ratio of potential decrease in T_{mrt} between the hill-climbing algorithm and the greedy algorithm for **(a)** five small trees with different time steps and **(b)** five large trees with different time steps, on 22 June 1983 (see Table 1 for detailed tree sizes). Two starting algorithms were used: random and genetic. There are 1709 potential locations for small trees and 1481 for large trees. A ratio > 1 indicates a larger potential decrease with the hill-climbing algorithm.

Table 3. Table showing difference in average T_{mrt} (°C) between the SOLWEIG simulation without new trees and the SOLWEIG simulation with new trees and recalculated SVF for the two algorithms, different tree sizes, different time periods and number of trees. The differences in T_{mrt} are for the shaded area (m^2) of the new trees (°C) with corresponding shadow area (m^2) and decrease per shaded pixel, and decrease for the entire raster (°C) with corresponding canopy area (m^2) and decrease per canopy area (°C m^{-2}).

Algorithm	Time period	Trees	Tree size	Δ°C in shadow (°C)	Shadow area (m^2)	Δ°C shadow area^{-1} (°C m^{-2})	Raster Δ°C (°C)	Canopy area (m^2)	Δ°C canopy area^{-1} (°C m^{-2})
Greedy	09:00–16:00	5	Small	-1412.10	138	-10.23	-1745.94	45	-38.80
Hill-climbing	09:00–16:00	5	Small	-1534.11	132	-11.62	-1743.59	45	-38.75
Greedy	09:00–16:00	5	Medium	-4040.68	380	-10.63	-4427.87	105	-42.17
Hill-climbing	09:00–16:00	5	Medium	-3899.84	351	-11.11	-4457.12	105	-42.45
Greedy	09:00–16:00	5	Large	-7670.51	875	-8.77	-7933.23	185	-42.88
Hill-climbing	09:00–16:00	5	Large	-7388.42	794	-9.31	-7898.14	185	-42.69
Greedy	13:00–16:00	5	Small	-1358.92	81	-16.78	-1648.90	45	-36.64
Hill-climbing	13:00–16:00	5	Small	-1371.81	76	-18.05	-1643.54	45	-36.52
Greedy	13:00–16:00	5	Medium	-3788.00	236	-16.05	-4138.16	105	-39.41
Hill-climbing	13:00–16:00	5	Medium	-3767.56	225	-16.74	-4114.40	105	-39.18
Greedy	13:00–16:00	5	Large	-6892.36	453	-15.21	-7310.08	185	-39.51
Hill-climbing	13:00–16:00	5	Large	-6922.94	434	-15.95	-7271.21	185	-39.30
Hill-climbing	13:00–16:00	4	Large	-5492.64	361	-15.22	-5836.44	148	-39.44
Hill-climbing	13:00–16:00	6	Large	-8327.50	526	-15.83	-8748.15	222	-39.41

5 Discussion

The aim of the TreePlanter is to find locations for the trees where they would have optimal potential to reduce T_{mrt}. Trees end up, e.g., shading areas in front of sunlit walls, areas which are known to be exposed to high radiant load

(Thorsson et al., 2011; Lindberg et al., 2016; Wallenberg et al., 2020). During the hottest time of the day in the case study, T_{mrt} under the trees dropped by as much as 26 °C from optimized positioning of trees (13:00–16:00 LST, Fig. 5). Such a sharp drop can have a profound effect on thermal (dis)comfort, making an area available to pedestrians, with
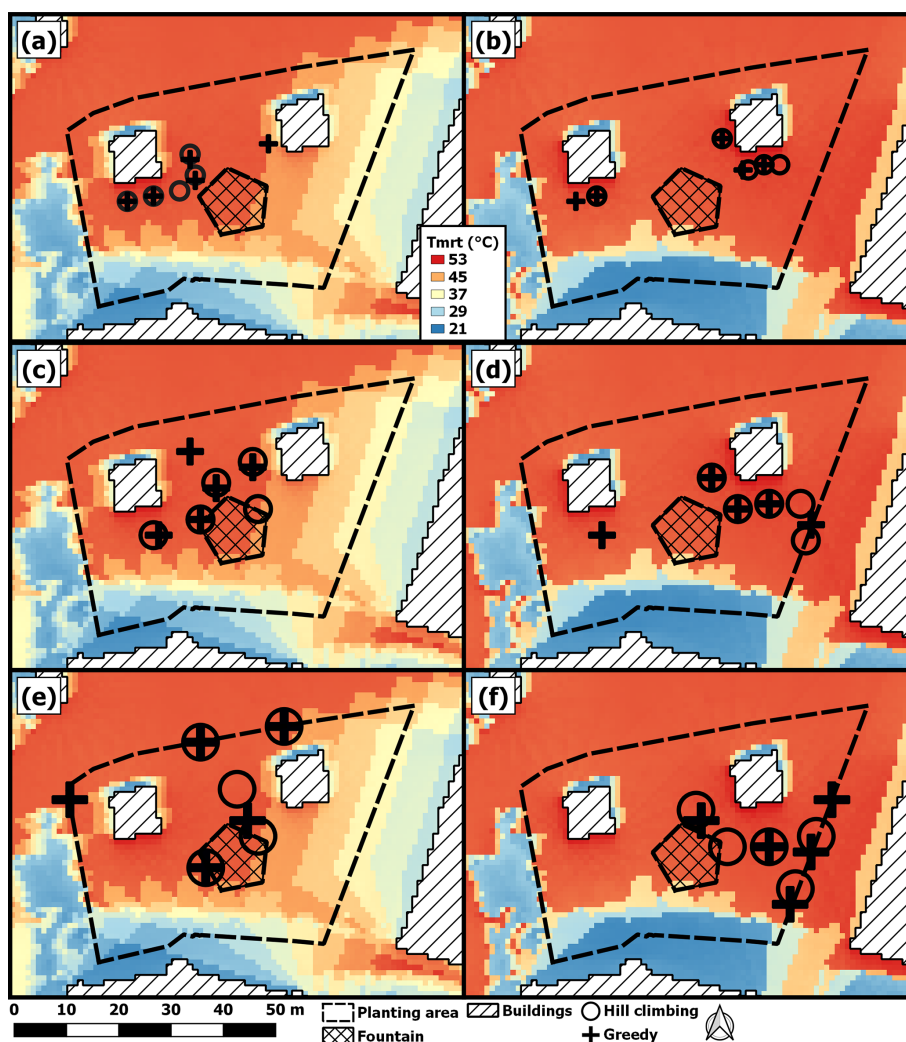
Figure 10. Locations for trees from the hill-climbing algorithm with the genetic starting algorithm and 20 000 iterations, and the greedy algorithm. The left column **(a, c, e)** and the right column **(b, d, f)** are for 09:00–16:00 and 13:00–16:00 LST, respectively, on 22 June 1983. Panels **(a)** and **(b)** are with small trees, **(c)** and **(d)** with medium trees and **(e)** and **(f)** with large trees. The underlying map shows mean T_{mrt} for the respective time periods from which the locations were determined.

fewer negative health effects on warm and clear summer days.

However, some differences in tree locations were found, depending on time of day, tree size and number of trees. Analyzing time of day, it possible to see that trees are, in general, located in the western part of the planting area during the longer time period (09:00–16:00 LST) in Fig. 4, whereas they end up in the eastern part during the shorter time period (13:00–16:00 LST) in Fig. 5. The differences in locations for trees are explained by spatial T_{mrt} patterns. The hours before noon would increase the amount of sunlit areas in front of west-facing facades integrated into TreePlanter. Thus, mitigation in these areas becomes more important. This shows the importance of timescale for planning of tree locations, and in this sense, season could also be an interesting aspect. Konarska et al. (2014), for example, discussed the impor-

tance of deciduous trees for mitigation of high radiant load in summer, as deciduous trees would allow higher transmissivity of solar irradiance in winter when leaves have dropped. Nevertheless, deciduous trees would block approximately 50 % of the incoming shortwave irradiance without leaves (Konarska et al., 2014). This means that optimized locations for trees in summer could have negative effect on thermal comfort in winter.

Tree size was also found to influence tree locations. The main finding is that, when tree size and hence shadow size were increased, the trees were dispersed as the model strives for the largest mitigating effect, which is mainly achieved by utilizing the entire tree shadows. This is most evident in Fig. 4 (09:00–16:00 LST), where small trees are aligned along the west building, but medium and large trees are dispersed, covering areas between the buildings. This leads to
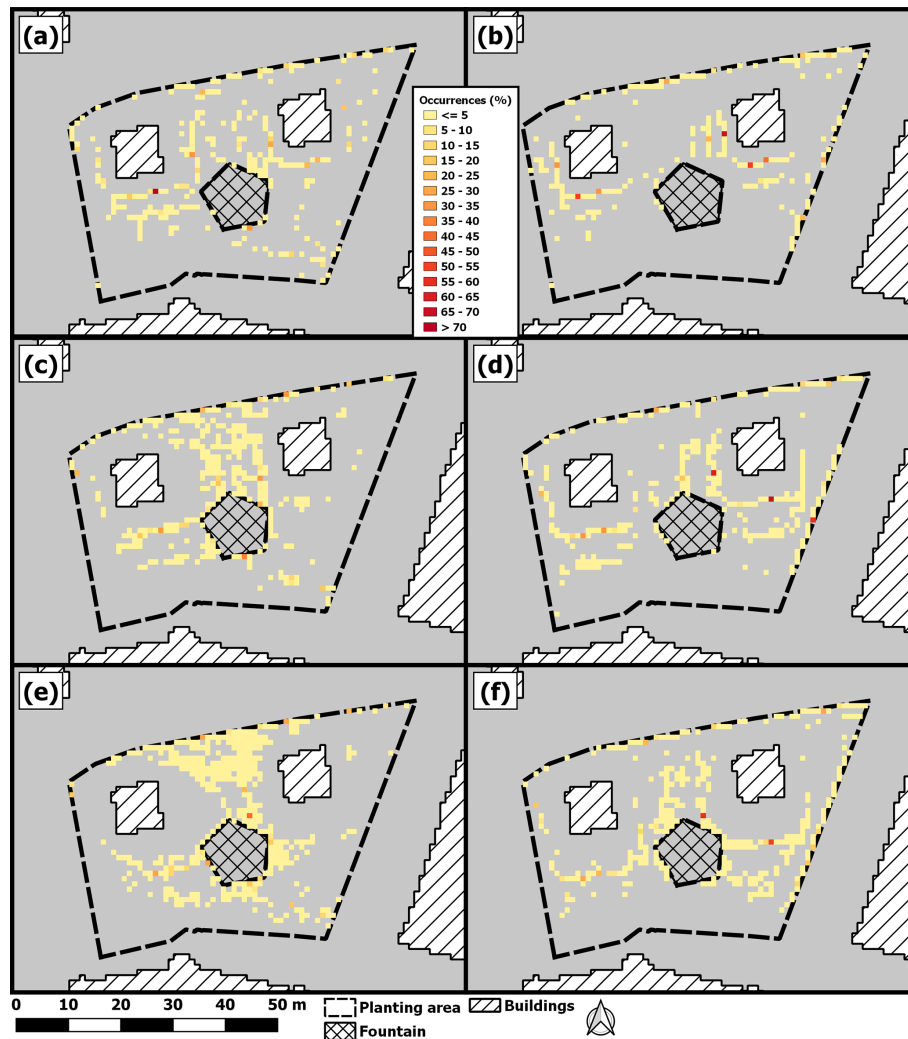
Figure 11. Occurrence maps showing percentage of times pixels were found to be an optimal tree position based on model runs with five trees, the random starting algorithm and 20 000 iterations. Gray indicates zero occurrences. Maps in panels **(a)** and **(b)** are for small trees, **(c)** and **(d)** are for medium trees and **(e)** and **(f)** are for large trees. The left column **(a, c, e)** is for 09:00–16:00 LST and the right column **(b, d, f)** is for 13:00–16:00 LST.

two trees, with tree size large, mitigating radiant load outside the planting area, where there in reality are tram tracks (Fig. 4e–f). The effects of tree size on tree locations is also visible in Fig. 5 (13:00–16:00 LST). In this case, however, it is rather that the shadow has a higher effect in a different location depending on tree size. One of the small trees was positioned in front of the west building. When tree size was increased to medium or large, tree shadows of all five trees have enhanced effect at the east building. This suggests that a larger tree shade, as in Fig. 5c, is more beneficial in different locations compared to smaller tree shades, and smaller exposed areas can remain exposed. Thus, tree size can be an important factor when optimized mitigation of high radiant load is desired. Tree size in this sense can also resemble stages in the tree's life, i.e., juvenile or mature. Thus, it can be important for planners and others to keep in mind that an

optimal position for a tree might differ depending on the age (size) of the tree.

Tree locations also changed depending on the number of trees. This is in clear contrast to the greedy algorithm, where tree positions are fixed when a new tree is added. TreePlanter enables tree locations to be influenced by each other, which means that none of the locations are fixed while the hill-climbing algorithm is still cycling through the trees, even if a single tree has reached an optimal position. This is realized with the nudging effect, which will enable two or more trees to explore if their combined and continuous shadow can find a more favorable and efficient mitigation of T_{mrt}. An example of this is apparent in Fig. 5e–f and Fig. 6, where the trees (at the east border of the planting area) have different locations with four (Fig. 6a and b) and five (Fig. 5e and f) trees compared to six trees (Fig. 6c and d). Another exam-

ple of how nudging and non-static locations are important when optimizing positions for trees is when comparing positions between the hill-climbing algorithm and the greedy algorithm, shown in Fig. 10. Here, it is evident that static locations, as with the greedy algorithm, can result in scattered tree locations. Even though locations established with the greedy algorithm are very efficient for mitigation of high T_{mrt}, they are inherently different compared to those established with the hill-climbing algorithm. The locations established with the hill-climbing algorithm had higher T_{mrt} mitigation (although the difference was small). However, when T_{mrt} was recalculated in SOLWEIG with updated SVF including the new trees, the tree locations established with the greedy algorithm in some few cases had higher mitigating effect compared to those established with the hill-climbing algorithm. This can to some extent be explained by changes in SVF near a tree, which influences radiation patterns in these areas in addition to the shaded area. Furthermore, shading patterns were different and tree shade from trees positioned with the greedy algorithm had larger tree shadows, which logically results in a larger total decrease in T_{mrt}. The average decrease in T_{mrt} in one pixel can be as large as 26 °C. Therefore, only a few extra shaded pixels can result in a larger total decrease in T_{mrt} with tree locations established with the greedy algorithm. On the other hand, the tree locations determined by the hill-climbing algorithm provided more efficient T_{mrt} mitigation per shaded area. Still, importantly, tree locations found by the hill-climbing algorithm are closer together. The occurrence map in Fig. 11 shows that the locations established with the greedy algorithm were also found by the hill-climbing algorithm. However, with the nudging function, allowing a continuous shadow of several trees to explore the ΔT_{mrt} raster, space is used more efficiently. In conclusion, from a planning perspective, it can be noteworthy that these two algorithms (hill-climbing and greedy) can result in considerably different tree locations. For this reason, they could potentially complement each other, serving as two alternatives.

In the model performance analysis, in Sect. 4, potential decrease in T_{mrt} with the hill-climbing algorithm and its two starting algorithms was evaluated against potential decrease in T_{mrt} with the greedy algorithm. The greedy algorithm was expected to always find tree locations that provided high (even if not optimal) T_{mrt} mitigation. This expectation was supported by the results presented here, for example, in Fig. 10, where tree locations determined by the greedy algorithm were always in areas exposed to high radiant load. Zhao et al. (2017), similarly, demonstrated optimized tree shade coverage with a greedy algorithm but used the algorithm to decrease solar exposure on facades. Thus, the ratio between the hill-climbing and greedy algorithms gives a good measure of the performance of the hill-climbing algorithm. From the results in Figs. 7–9, it is evident that the hill-climbing algorithm finds acceptable locations already after one iteration as none of the results show a ratio lower than

0.9. After 100–500 iterations, tree locations established by the hill-climbing algorithm show an equal or higher potential mitigating effect than the greedy algorithm, in almost all cases. There are, however, some exceptions, especially with the random starting algorithm. Running the hill-climbing algorithm with the genetic algorithm, on the other hand, results in tree locations with a potential mitigating effect equal to, or higher than, the greedy algorithm after 500 iterations for all cases except 09:00–16:00 LST with small trees with 3000 iterations. Running the model with 20 000 iterations, on the other hand, always resulted in tree locations with equal or higher potential mitigating effect compared to the greedy algorithm, regardless of starting algorithm. The exceptions, seen for example in Fig. 7 with small trees and 3000 iterations, are an example of the fact that, as explained in Sect. 2, metaheuristics are not guaranteed to find an absolutely optimal solution to a problem. Adding to this, it is not possible to prove that any of the locations established by the hill-climbing algorithm are absolutely optimal, unless extensive and computationally demanding brute-force calculations are performed.

The results from the comparison between the genetic and random starting algorithms are in line with those by Stojakovic et al. (2020). Although their approach was different from the one presented in this paper in that they did not use a hill-climbing algorithm, they did use a genetic algorithm. In their method, tree locations were determined from tree locations from a previous iteration. Their results showed that tree locations with the genetic algorithm had a higher mitigating effect compared to randomly positioned trees. This can be compared to the results found in this paper, where tree locations with a high potential mitigating effect are sometimes found with fewer iterations with the genetic algorithm compared to the random algorithm. Adding to this, Stojakovic et al. (2020) found convergence after approximately 3000 iterations. Here, some of the examples show convergence already after 500 iterations, indicating that exploration with the hill-climbing algorithm in combination with a genetic algorithm for starting positions could be a beneficial approach.

When model runtime was evaluated, it was quite clear that when time steps, tree size or number of trees increased, so did model runtime (Table 2). When time step was increased, the number of necessary calculations increased, mainly when tree shadows overlapped. This was further reinforced when tree size was increased. An increase in tree size increases the area shadowed by the tree, regardless of time of day, as long as the solar elevation is above the horizon. A larger shadow increases the possibility of tree shadows overlapping, and overlapping shadows need adjustment to find optimal locations. Increasing the number of trees also adds to the probability of overlapping shadows. Increasing model domain size (by changing pixel resolution) resulted in an exponential increase in model runtime, from very fast for low resolution (2 m) to relatively long model runtime with a high resolution (0.25 m). As model pixels become smaller, the number

of possible locations for trees increases substantially, leading to increased model runtime. The four factors described above all influence the model complexity and computational time. The largest influence comes from the number of time steps. The increase in computational time is considered to be an effect of overlapping tree shadows that need adjustment. Increasing tree size and the number of trees further increases the possibility of overlapping shadows. Additionally, an increase in the number of time steps would increase the time to adjust, as well as a larger shadow of integrated time steps. Moreover, larger model domains can substantially increase model runtime.

When evaluating and comparing the two starting algorithms, the genetic algorithm improved model runtime in all cases except when the number of trees was increased to six, seven and eight, for which model runtime increased. On the other hand, the genetic algorithm determined optimal positions at an earlier stage compared to the random algorithm. This, however, needs further testing, but the enhanced speed of the genetic algorithm can reasonably be explained by two factors. One factor is that warm areas, i.e., local optimums with high T_{mrt}, are in close proximity to each other, meaning that subsequent generations (iterations) potentially start closer to local optimums compared to random starting positions. The other factor could be the geometry of the urban setting studied here, where sunlit areas follow the x axis or the y axis. As an example, the two small buildings inside the study area are at approximately same y positions, with the right-hand one being located at a slightly higher position. The area in between the buildings is located at approximately the same x position. Thus, if a tree inherits the y position from a tree that in the prior run ended up in front of one these buildings, it would probably not have to explore too many pixels in order to a find a new local optimum. Likewise, if a tree inherits its x position from a tree with a local optimum between the two buildings, it would not have to move far to find an optimal position. The random starting algorithm, on the other hand, could have trees starting in positions where exploration of the ΔT_{mrt} raster would take longer in order to find a local optimum. These factors can possibly also explain the decrease in model runtime with the genetic starting algorithm when number of trees was set to seven or eight. The trees would start closer to each other, and the chance of overlapping shadows increases.

As described above, model runtime differs depending on tree size, the numbers of trees and time steps, and the size of the model domain. With the genetic starting algorithm, for five small trees giving 1709 possible locations, and seven time steps (09:00–16:00 LST), each iteration takes approximately 0.13 s. Using same setup, but with the large tree size and 1481 possible locations, each iteration takes approximately 0.26 s. For comparison, Stojakovic et al. (2020) utilized Rhinoceros 3D (CAD modeling software), Grasshopper (add-on to Rhinoceros 3D), and an evolutionary algorithm (inheritance) from Galapagos (a plugin for Grasshopper with

generic solvers; Rutten, 2013). With their setup, for five trees (two sizes simultaneously, three with a canopy diameter of 8 m and two with 17 m), 10 time steps and 625 possible locations for trees, they had a mean iteration runtime of around 9 s. This suggests that the 0.26 s for five large trees and seven time steps in the model presented in this paper are highly efficient.

Genetic algorithms have been used successfully in previous research on optimization problems, e.g., locations for hospitals in Hong Kong (Li and Yeh, 2005), locations for train stations in Leicester, UK (Ahmed et al., 2020), spatial land use allocation planning in Guitiriz, Spain (Porta et al., 2013), and tree locations (Stojakovic et al., 2020). The results presented in this paper show how random starting positions and a genetic algorithm to determine starting positions can be used together with a hill-climbing algorithm for optimizing tree locations. The benefit of using a hill-climbing algorithm is that it enables thorough exploration of potential tree locations and at the same time simplifies any potential adjustment for potentially overlapping tree shadows. Using the genetic algorithm to determine starting positions, in most cases, improved model runtime and convergence compared to randomly determined starting positions.

Even if model runtime is relatively fast, it needs improvement for the hill-climbing algorithm to become a generic tree-location-finding tool. Real-world applications will likely have more trees and larger areas of interest than this study. This will require an increase in the number of iterations, as the number of possible locations would increase. Running TreePlanter with many iterations, several trees and in a large area is time consuming. The examples shown here are relatively fast. However, as seen with the genetic algorithm and six, seven or eight trees, model runtime increases quite extensively. A greedy algorithm, as an alternative to the hill-climbing algorithm, is included in TreePlanter to support larger studies (see Sect. 5.2). A thorough evaluation of the algorithms included in the model as well as their pros and cons are aspects for future studies. How do their results compare? Under what circumstances are the different algorithms applicable?

5.1 Model limitation and potential

In SOLWEIG, as well as TreePlanter presented in this paper, there is no option for individual tree parameterization. For example, transmissivity is set to one general value for all trees. The default in SOLWEIG is 3 % for summer (Konarska et al., 2014). Continuing, the model and the examples presented in this paper only include radiant load. However, there are other factors that can be affected from adding more trees, e.g., wind and evapotranspiration, which are not included. While evapotranspiration from trees has been shown to have a negligible effect on the thermal comfort, trees affecting pedestrian-level wind can have a large effect on outdoor thermal comfort (Lee and Mayer, 2021). There are also other fac-
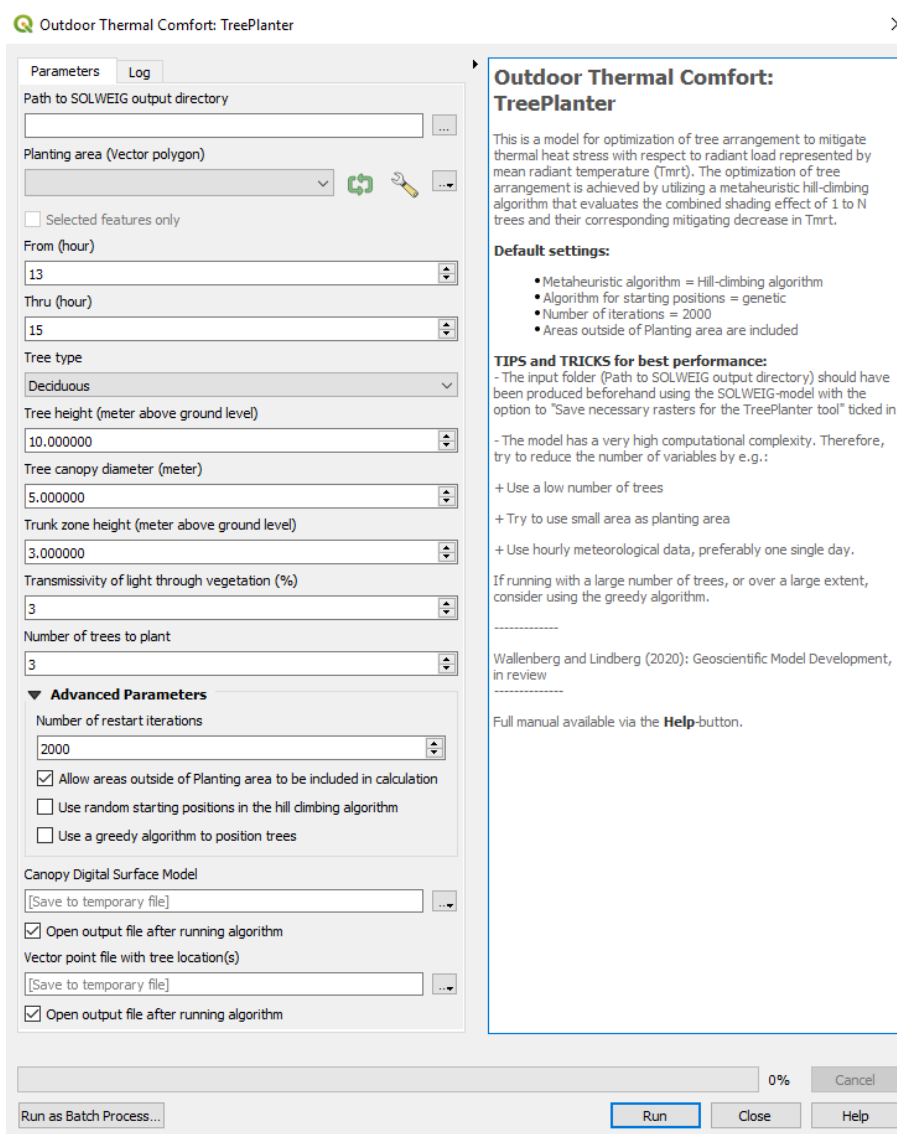
Figure 12. Graphical user interface of the TreePlanter 1.0 in QGIS.

tors, connected to plant physiology, that need consideration when planting a tree in an urban area, such as root spacing, soil conditions, climatic growing conditions and water availability (Vogt et al., 2017), which are not examined here.

Not included in the current version of TreePlanter is a recalculation of SVF between iterations. Omitting recalculation of SVFs has a minor effect on the total radiant conditions under the tree as well as the influence on the radiant conditions in the surrounding environment, when the moving trees block parts of the sky. View factors influence, e.g., the amount of diffuse shortwave irradiance reaching an area as well as influencing the longwave irradiance under and around the trees. However, these effects are relatively small, whereas including recalculations in the model would increase runtime considerably. We thus do not foresee including SVF recalcu-

lations in the model. See Lindberg and Grimmond (2011a) for a more detailed discussion.

Speak et al. (2020) studied the shading effect of single trees in Bolzano, Italy. They found that leaf area density and canopy diameter are key in mitigating surface temperatures and concluded that strategic planting of urban trees with cautious selection of species can help mitigate surface temperatures. Antoniadis et al. (2020), similarly, discussed strategic planting and positioning of trees, to alleviate heat stress in urban schoolyards. The model presented in this paper can aid in the strategic planting of trees based on T_{mrt}, as it enables the possibility of positioning trees and changing canopy diameters, tree height and trunk zone height. The possibility of changing transmissivity of shortwave irradiance through the canopy can be used as an analogy for leaf area density. Fur-

thermore, it allows for analysis of days with different meteorological data, and even a possibility of combining days from different seasons. Moreover, with modifications, the methodology described in this paper could be used for other optimization purposes, e.g., mitigation of incoming solar irradiance with respect to harmful UV.

Future possible developments include the possibility to position trees of different sizes simultaneously. Furthermore, studies on the effects of locations during different weather conditions and seasons are expected, as well as evaluating if locations are feasible with respect to root spacing, pipes, etc.

5.2   Tool accessibility

To facilitate use and accessibility to other researchers and service providers, TreePlanter is available as a tool as part of the UMEP climate service tool in the open-source geographical information system QGIS (https://qgis.org, last access: 1 February 2022). The graphical user interface (GUI) from the Processing Toolbox in QGIS is presented in Fig. 12. For simplification, the pre-processed input data required from SOLWEIG for TreePlanter will be provided through a tick box in the SOLWEIG GUI.

Moreover, a vector polygon layer that determines a planting area where it is possible to plant trees within the extent of the SOLWEIG output is required (see first paragraph in Sect. 2). This option can be used to avoid roads, statues, water bodies or other obstructing objects that are not shown in the building raster and would prevent trees from being planted in such locations. The extent of the planting area can be set to the same size as the extent of the output data from SOLWEIG. In this case, a buffer zone will be enforced to avoid edge effects. Another alternative in the tool will be to disable shading outside of the study area.

Optional settings can be set under "Advanced Parameters". These include options to use either the random or the genetic starting algorithms, as well as number of iterations. Furthermore, an option to use a greedy algorithm, instead of the hill-climbing algorithm, is included. The greedy algorithm is faster, which can be useful when requesting positions for a larger number of trees in extensive model domains. However, as shown in the results and mentioned in the discussion, tree locations can differ considerably between the hill-climbing algorithm and the greedy algorithm.

The output from the TreePlanter is an updated raster CDSM and a vector point file with the positions of the new trees.

6   Concluding remarks

The TreePlanter model presented in this paper has several advantages for future studies of mitigation of T_{mrt} and analysis of shadow patterns in urban areas. Conclusions from the

model performance analysis and the case study are as follows:

- Modeling and optimization of positioning of trees with respect to mitigation of T_{mrt} are very complex and computationally extensive. TreePlanter and its metaheuristic hill-climbing and greedy algorithms can give guidance in this issue.

- Both algorithms investigated (hill-climbing and greedy) find tree locations that result in substantial decrease in T_{mrt} and thus increase thermal comfort in exposed areas on clear, hot days. The locations, however, can differ considerably between the two algorithms.

- Tree locations depend primarily on tree size and time of day. Tree size indicates that juvenile and mature trees have different optimal positions, which can be important in, e.g., urban planning.

- The hill-climbing algorithm incorporates the combined shading effects of several trees simultaneously, which are not addressed in the greedy algorithm.

- The model presented in this paper can give advice to urban planners and others in mitigating and improving thermal comfort in outdoor urban settings.

References

Abdi, B., Hami, A., and Zarehaghi, D.: Impact of small-scale tree planting patterns on outdoor cooling and thermal comfort, Sustain. Cities Soc., 56, 102085, https://doi.org/10.1016/j.scs.2020.102085, 2020.

Ahmed, C., Nur, K., and Ochieng, W.: GIS and genetic algorithm based integrated optimization for rail transit system planning, J. Rail Transport Plann. Manage., 16, 100222, https://doi.org/10.1016/j.jrtpm.2020.100222, 2020.

Antoniadis, D., Katsoulas, N., and Papanastasiou, D. K.: Thermal environment of urban schoolyards: current and future design with respect to children's thermal comfort, Atmosphere, 11, 1144, https://doi.org/10.3390/atmos11111144, 2020.

Arnfield, A. J.: Two decades of urban climate research: a review of turbulence, exchanges of energy and water, and the urban heat island, Int. J. Climatol., 23, 1–26, https://doi.org/10.1002/joc.859, 2003.

ASHRAE: ASHRAE Fundamentals Handbook 2001, SI Edition, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, ISBN 1883413885, 2001.

Bajsanski, I., Stojakovic, V., and Jovanovic, M.: Effect of tree location on mitigating parking lot insolation, Comput. Environ. Urban Syst., 56, 59–67, https://doi.org/10.1016/j.compenvurbsys.2015.11.006, 2016.

Chen, H., Ooka, R., and Kato, S.: Study on optimum design method for pleasant outdoor thermal environment using genetic algorithms (GA) and coupled simulation of convection, radiation and conduction, Build. Environ., 43, 18–30, https://doi.org/10.1016/j.buildenv.2006.11.039, 2008.

Chen, L., Yu, B., Yang, F., and Mayer, H.: Intra-urban differences of mean radiant temperature in different urban settings in Shanghai and implications for heat stress under heat waves: A GIS-based approach, Energ. Build., 130, 829–842, https://doi.org/10.1016/j.enbuild.2016.09.014, 2016.

Dousset, B., Gourmelon, F., Laaidi, K., Zeghnoun, A., Giraudet, E., Bretin, P., Mauri, E., and Vandentorren, S.: Satellite monitoring of summer heat waves in the Paris metropolitan area, Int. J. Climatol., 31, 313–323, https://doi.org/10.1002/joc.2222, 2010.

Gabriel, K. M. A. and Endlicher, W. R.: Urban and rural mortality rates during heat waves in Berlin and Brandenburg, Germany, Environ. Pollut., 159, 2044–2050, https://doi.org/10.1016/j.envpol.2011.01.016, 2011.

Gál, C. V. and Kántor, N.: Modeling mean radiant temperature in outdoor spaces, A comparative numerical simulation and validation study, Urban Climate, 32, 100571, https://doi.org/10.1016/j.uclim.2019.100571, 2020.

Höppe, P.: A new procedure to determine the mean radiant temperature outdoors, Wetter und Leben, 44, 147–151, 1992.

HosseiniHaghighi, S., Izadi, F., Padsala, R., and Eicker, U.: Using Climate-Sensitive 3D City Modeling to Analyze Outdoor Thermal Comfort in Urban Areas, Int. J. Geo-Inf., 9, 688, https://doi.org/10.3390/ijgi9110688, 2020.

Jänicke, B., Meier, F., Lindberg, F., Schubert, S., and Scherer, D.: Towards city-wide, building-resolving analysis of mean radiant temperature, Urban Climate, 15, 83–98, https://doi.org/10.1016/j.uclim.2015.11.003, 2016.

Kántor, N., Gál, C. V., Gulyás, Á., and Unger, J.: The Impact of Façade Orientation and Woody Vegetation on Summertime Heat Stress Patterns in a Central European Square: Comparison of Radiation Measurements and Simulations, Adv. Meteorol., 2018, 2650642, https://doi.org/10.1155/2018/2650642, 2018.

Konarska, J., Lindberg, F., Larsson, A., Thorsson, S., and Holmer, B.: Transmissivity of solar radiation through crowns of single urban trees – application for outdoor thermal comfort modelling, Theor. Appl. Climatol., 117, 363–376, https://doi.org/10.1007/s00704-013-1000-3, 2014.

Lantmäteriet: Gothenburg, Sweden, SWEREF 99 TM EPSG:3006, Orthophoto RGB 0.25 m, Map, https://maps.slu.se (last access: 24 August 2020), 2018.

Lau, K.-L., Lindberg, F., Rayner, D., and Thorsson, S.: The effect of urban geometry on mean radiant temperature under future climate change: a study of three European cities, Int. J. Biometeorol., 59, 799–814, https://doi.org/10.1007/s00484-014-0898-1, 2015.

Lau, K.-L., Ren, C., Ho, J., and Ng, E.: Numerical modelling of mean radiant temperature in high-density subtropical urban environment, Energ. Build., 114, 80–86, https://doi.org/10.1016/j.enbuild.2015.06.035, 2016.

Lee, H. and Mayer, H.: Solar elevation impact on the heat stress mitigation of pedestrians on tree-lined sidewalks of E-W street canyons – Analysis under Central European heat wave conditions, Urban Forest. Urban Green, 58, 126905, https://doi.org/10.1016/j.ufug.2020.126905, 2021.

Lee, H., Mayer, H., and Kuttler, W.: Impact of spacing between tree crowns on the mitigation of daytime heat stress for pedestrians inside E–W urban street canyons under Central European conditions, Urban Forest. Urban Green., 48, 126558, https://doi.org/10.1016/j.ufug.2019.126558, 2020.

Li, X. and Yeh, A. G-O.: Integration of genetic algorithms and GIS for optimal location search, Int. J. Geogr. Inform. Sci., 19, 581–601, https://doi.org/10.1080/13658810500032388, 2005.

Lindberg, F. and Grimmond, C. S. B.: Nature of vegetation and building morphology characteristics across a city: influence on shadow patterns and mean radiant temperature in London, Urban Ecosyst., 14, 617–634, https://doi.org/10.1007/s11252-011-0184-5, 2011a.

Lindberg, F. and Grimmond, C. S. B.: The influence of vegetation and building morphology on shadow patterns and mean radiant temperature in urban areas: Model development and evaluation, Theor. Appl. Climatol., 105, 311–323, https://doi.org/10.1007/s00704-010-0382-8, 2011b.

Lindberg, F., Holmer, B., and Thorsson, S.: SOLWEIG 1.0 – modelling spatial variations of 3D radiation fluxes and mean radiant temperature in complex urban settings, Int. J. Biometeorol., 52, 697–713, https://doi.org/10.1007/s00484-008-0162-7, 2008.

Lindberg, F., Holmer, B., Thorsson, S., and Rayner, D.: Characteristics of the mean radiant temperature in high latitude cities –

implications for sensitive climate planning applications, Int. J. Biometeorol., 58, 613–627, https://doi.org/10.1007/s00484-013-0638-y, 2014.

Lindberg, F., Thorsson, S., Rayner, D., and Lau, K.: The impact of urban planning strategies on heat stress in a climate-change perspective, Sustain. Cities Soc., 25, 1–12, https://doi.org/10.1016/j.scs.2016.04.004, 2016.

Lindberg, F., Grimmond, C. S. B., Gabey, A., Huang, B., Kent, C. W., Sun, T., Theeuwes, N., Järvi, L., Ward, H., Capel-Timms, I., Chang, Y. V., Jonsson, P., Krave, N., Liu, D., Meyer, D., Olofson, F., Tan, J. G., Wästberg, D., Xue, L., and Zhang, Z.: Urban Multi-Scale Environmental Predictor (UMEP): An integrated tool for city-based climate services, Environ. Modell. Softw., 99, 70–87, https://doi.org/10.1016/j.envsoft.2017.09.020, 2018.

Luke, S.: Essentials of Metaheuristics, Lulu, 2nd edn., http://cs.gmu.edu/~sean/book/metaheuristics/ (last access: 1 February 2022), 2013.

Mayer, H. and Höppe, P.: Thermal comfort of man in different urban environments, Theor. Appl. Climatol., 38, 43–49, https://doi.org/10.1007/BF00866252, 1987.

Mayer, H., Holst, J., Dostal, P., Imbery, F., and Schindler, D.: Human thermal comfort in summer within an urban street canyon in Central Europe, Meteorol. Z., 17, 241–250, https://doi.org/10.1127/0941-2948/2008/0285, 2008.

Norton, B. A., Coutts, A. M., Livesley, S. J., Harris, R. J., Hunter, A. M., and Williams, N. S. G.: Planning for cooler cities: A framework to prioritise green infrastructure to mitigate high temperatures in urban landscapes, Landscape Urban Plan., 134, 127–138, https://doi.org/10.1016/j.landurbplan.2014.10.018, 2015.

Ooka, R., Chen, H., and Kato, S.: Study on optimum arrangement of trees for design of pleasant outdoor environment using multi-objective genetic algorithm and coupled simulation of convection, radiation and conduction, J. Wind Eng. Ind. Aerod., 96, 1733–1748, https://doi.org/10.1016/j.jweia.2008.02.039, 2008.

Porta, J., Parapar, J., Doallo, R., Rivera, F. F., Santé, I., and Crecente, R.: High performance genetic algorithm for land use planning, Comput. Environ. Urban Syst., 37, 45–58, https://doi.org/10.1016/j.compenvurbsys.2012.05.003, 2013.

Rutten, D.: Galapagos: on the logic and limitations of generic solvers, Archit. Des., 83, 132–135, https://doi.org/10.1002/ad.1568, 2013.

Speak, A., Montagnani, L., Wellstein, C., and Zerbe, S.: The influence of tree traits on urban ground surface shade cooling, Landscape Urban Plan., 197, 103748, https://doi.org/10.1016/j.landurbplan.2020.103748, 2020.

Srivanit, M. and Jareemit, D.: Modeling the influences of layouts of residential townhouses and tree-planting patterns on outdoor thermal comfort in Bangkok suburb, J. Build. Eng., 30, 101262, https://doi.org/10.1016/j.jobe.2020.101262, 2020.

Stojakovic, V., Bajsanski, I., Savic, S., Milosevic, D., and Tepavcevic, B.: The influence of changing location of trees in urban green spaces on insolation mitigation, Urban Forest. Urban Green., 126721, https://doi.org/10.1016/j.ufug.2020.126721, 2020.

Thom, J., Coutts, A. M., Broadbent, A. M., and Tapper, N. J.: The influence of increasing tree cover on mean radiant temperature across a mixed development suburb in Adelaide, Australia, Urban Forest. Urban Green., 20, 233–242, https://doi.org/10.1016/j.ufug.2016.08.016, 2016.

Thorsson, S., Lindberg, F., Björklund, J., Holmer, B., and Rayner, D.: Potential changes in outdoor thermal comfort conditions in Gothenburg, Sweden due to climate change: the influence of urban geometry, Int. J. Climatol., 31, 324–335, https://doi.org/10.1002/joc.2231, 2011.

Thorsson, S., Rocklöv, J., Konarska, J., Lindberg, F., Holmer, B., Dousset, B., and Rayner, D.: Mean radiant temperature – A predictor of heat related mortality, Urban Climate, 10, 332–345, https://doi.org/10.1016/j.uclim.2014.01.004, 2014.

Thorsson, S., Rayner, D., Lindberg, F., Monteiro, A., Katzschner, L., Lau, K. K.-L., Campe, S., Katschner, A., Konarska, J., Onomura, S., Velho, S., and Holmer, B.: Present and projected future mean radiant temperature for three European cities, Int. J. Biometeorol., 61, 1531–1543, https://doi.org/10.1007/s00484-017-1332-2, 2017.

Vogt, J., Gillner, S., Hofmann, M., Tharang, A., Dettmann, S., Gerstenberg, T., Schmidt, C., Gebauer, H., Van de Riet, K., Berger, U., and Roloff, A.: Citree: A database supporting tree selection for urban areas in temperate climate, Landscape Urban Plan., 157, 14–25, https://doi.org/10.1016/j.landurbplan.2016.06.005, 2017.

Wallenberg, N. and Lindberg, F.: TreePlanter v1.0 as part of UMEP-processing, Zenodo [code], https://doi.org/10.5281/zenodo.4616761, 2021a.

Wallenberg, N. and Lindberg, F.: TreePlanter v1.0 test dataset, Zenodo [data set], https://doi.org/10.5281/zenodo.4616770, 2021b.

Wallenberg, N. and Lindberg, F.: TreePlanter v1.0 tutorial and tool description, Zenodo [data set], https://doi.org/10.5281/zenodo.4767387, 2021c.

Wallenberg, N., Lindberg, F., Holmer, B., and Thorsson, S.: The influence of anisotropic diffuse shortwave radiation on mean radiant temperature in outdoor urban environments, Urban Climate, 31, 100589, https://doi.org/10.1016/j.uclim.2020.100589, 2020.

Zhao, Q., Wentz, E. A., and Murray, A. T.: Tree shade coverage optimization in an urban residential environment, Build. Environ., 115, 269–280, https://doi.org/10.1016/j.buildenv.2017.01.036, 2017.

Zhao, Q., Sailor, D. J., and Wentz, E. A.: Impact of tree locations and arrangements on outdoor microclimates and human thermal comfort in an urban residential environment, Urban Forest. Urban Green., 32, 81–91, https://doi.org/10.1016/j.ufug.2018.03.022, 2018.