EPiC Series in Computing

Volume 41, 2016, Pages 240–252

GCAI 2016. 2nd Global Conference on Artificial Intelligence

# Heliostat Field Layout Optimization with Evolutionary Algorithms

Pascal Richter[1], David Laukamp[1], Levin Gerdes[2], Martin Frank[1], and Erika Ábrahám[2]

[1] RWTH Aachen University, MathCCES, Aachen, Germany
[2] RWTH Aachen University, Computer Science Department, Aachen, Germany

**Abstract**

The exploitation of solar power for energy supply is of increasing importance. While technical development mainly takes place in the engineering disciplines, computer science offers adequate techniques for optimization. This work addresses the problem of finding an optimal heliostat field arrangement for a solar tower power plant. We propose a solution to this global, non-convex optimization problem by using an evolutionary algorithm. We show that the convergence rate of a conventional evolutionary algorithm is too slow, such that modifications of the recombination and mutation need to be tailored to the problem. This is achieved with a new genotype representation of the individuals. Experimental results show the applicability of our approach.

## 1 Introduction

The contribution of renewable energy to our global energy use has significantly increased over the past ten years. Completely new industry branches have developed in the field of solar, wind, and biomass energy. Among the renewable energy technologies, *concentrated solar power* (*CSP*) *plants* are a promising option for power generation in regions with high direct solar irradiation. In a CSP plant, a field of hundreds or thousands of large mirrors (*heliostats*) are used to reflect rays of sunlight onto a *receiver* placed at the top of a tower (see Figure 1). At the receiver, the concentrated sunlight is absorbed and the resulting thermal energy is used to heat a transfer fluid, which in turn can either be used directly to produce electricity through a conventional thermodynamic cycle or be stored. Today four large tower plants are already operating in the US (Ivanpah 1-3 and Crescent Dunes), three in Spain (PS10, PS20 and Gemasolar) and one is under construction in South Africa (Khi Solar One). Additionally, numerous small-scale plants exist around the world for

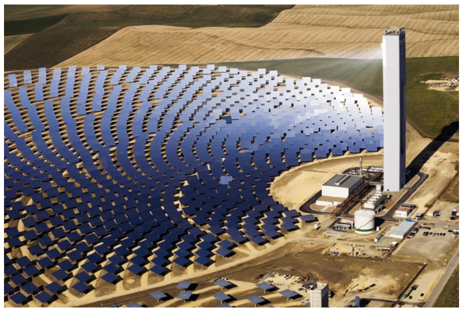


Figure 1: Solar tower plant PS10 in Andalusia, Spain (source: [4]).

demonstration and research purposes, for example the Solarturm Jülich in Germany, and the facilities CESA-1 and SSTS-CRS in Spain.

The *layout* of the heliostat field, i.e., the arrangement of the heliostats around the receiver, affects not only the construction costs (the heliostat field is the most expensive sub-system of a CPS plant) but also the efficiency of the plant, due to several reasons. Firstly, the irradiation energy that a single heliostat can contribute to energy generation depends on its position. Secondly, a heliostat can cast a shadow over other heliostats, reducing the amount of sunlight reaching them. Last but not least, a first heliostat can be blocked by a second heliostat if the second one is positioned between the first one and the receiver.

Thus the layout of a CSP plant should be optimized carefully before construction. This *layout optimization problem* is a highly complex global, non-convex optimization problem [2]. As constraints on the placement we need to assure, e.g., that all heliostats are placed inside an allowed area and that there is a minimum distance between neighboring heliostats such that they cannot touch each other. As objective function usually the annual performance, received irradiation energy, thermal energy, or levelized costs of efficiency in Euro/kWh (LCOE) should be maximized.

Most layout optimization approaches use simulation tools to compute the objective function value for a fixed layout [3]. Such simulations use (1) a sun irradiation model based on meteorological data, (2) a heliostat model for light concentration and reflection, and (3) models for the interaction (shading/blocking) of heliostats. Using these models, the irradiation energy at the receiver is computed for, e.g., each hour of a year to estimate the annual energy generated by the plant. For further details, the interested reader is referred to [7, 9].

Several approaches were proposed to solve the layout optimization problem, using different concepts [6]:

- The *field growth method* is a concept where the heliostats are added step by step on predefined points of the field. The algorithm terminates when the system requirements (e.g. minimum power output) are met. The efficiency and the runtime of this algorithm highly depends on the number of the predefined points of the field. Additionally, due to the successive approach each heliostat allocation depends on the preceding allocations, such that the optimization can hardly be parallelized.

  Sánchez and Romero [10] employed this concept by using a greedy heuristic. The algorithm starts with an empty field. The whole field is discretized in a set of possible points for placing heliostats. Each point in the field is evaluated, such that the points can be rated by their energy contribution. The best point is chosen as position for placing the next heliostat. Due to shading and blocking effects of the new heliostat, all free points in the field have to be evaluated again. Sánchez and Romero called this algorithm YNES, an abbreviation for *yearly normalized energy surfaces*.

- Much research has been done in the field of *pattern-based method*, where all heliostats are arranged in geometric patterns which can be described by certain adjustable parameters. With this approach the search domain is highly reduced from hundreds or thousands of $x$ and $y$ coordinates to a handful parameters. So, instead of optimizing the $x$-$y$ coordinates, here now the pattern parameters are optimized which though influence the $x$-$y$ coordinates. Thus, the pattern method essentially determines the best adaptation of the pattern for the problem and not necessarily the best $x$-$y$ coordinates for optimal plant performance [6]. In literature, several different patterns have been used: rows [11], radial staggered [5], and biomimetic patterns [7]. The disadvantage of these optimizers is the small search space by construction.

- The *free variable method* follows a more classical optimization approach by directly optimizing the $x$-$y$ coordinates. Due to the complexity of the problem an appropriate heuristic is needed. There exists a large variety of optimization approaches which could be used, such as non-linear programming, general gradient-based methods, to nature-inspired genetic, evolutionary, viral, simulated annealing, and particle swarm algorithms.

  So far, we just know from a gradient-based method [6] which was developed for the heliostat layout optimization problem. This approach starts with a random layout which iteratively adjusts each $x$-$y$ coordinate by following the gradient in the direction of a better function value until a certain objective is achieved. The gradient of the simulation may be obtained by finding the partial derivatives of the ray tracer function with respect to each variable. In this optimization concept the heliostats are not limited to a pattern, which means, that they can freely move through the field during the optimization process.
- The *multi-step optimization strategy* consists of a combination of two or more optimization methods. First a meta-heuristic is used, able to search on a huge solution space and to move towards the global maximum. Afterwards, subordinated methods like a greedy heuristic or a linear programming refine the solution locally. The work [1] uses a pattern based optimization method and refined the results with a greedy heuristic by perturbing each heliostat position locally. This strategy has shown to give better results when compared to each of the two algorithms alone.

Despite this wide spectrum of achieved results, there is still a strong need for new approaches to solve the layout optimization problem to further improve the solutions. Because this problem belongs to the NP-hard class of combinatorial optimization problems, methods from artificial intelligence may help to successfully find a good solution. In this work, we propose a classical optimization approach by using an *evolutionary algorithm* (which belongs to the above introduced class of free variable methods). We show that it is necessary to modify the crossover and mutation step, to increase the slow convergence rate. The new genetic operators are tailored to the underlying problem of two-dimensional genes (the $x$ and $y$ positions of the heliostats). We give a comparison of a pattern-based algorithm with an evolutionary algorithm (with classical and modified genetic operators) to show the efficiency of our approach.

In [8] we already used a genetic algorithm and combined it with neural networks for parameter synthesis in a solar thermal power plant. We could reduce the calculation time of the optimization procedure by around 90% compared to an approach without neural networks. In [9] we extended the optimizer for a multi-objective optimization: We used smoothing functionals to disperse the local optima. Within this work we now present the results of using modified genetic operators which are tailored to the underlying problem of finding an optimal heliostat field layout.

The rest of the paper is structured as follows. In Section 2 we describe our optimization algorithm using the evolutionary approach. We show the importance of adapting the classical operators used in evolutionary algorithms by providing some experimental results in Section 3. We conclude the paper in Section 4 with an outlook on future work.

## 2 Layout Optimization using Evolutionary Algorithms

For a fixed tower position $(x_t, y_t) \in \mathbb{R}^2$, a number $N$ of heliostats and a given set $\Omega \subseteq \mathbb{R}^2$ of possible two-dimensional[1] heliostat positions, a *layout configuration* is defined as a collection

[1]The third dimension (height) is also relevant, however, for a given parcel of land on which the plant should be built the coordinate in the third dimension is determined by the coordinates in the first two dimensions.

of positions $\mathcal{I} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq \Omega$ for the heliostat centers. To avoid collisions with neighboring heliostats, valid layout configurations must satisfy the constraints $\|(x_i, y_i) - (x_j, y_j)\| \geq d$ for each $1 \leq i < j \leq N$, where $d$ is the expansion size (incl. safety distance) of all heliostats and $\| \cdot \|$ is the Euclidean distance. The goodness of a layout configuration $\mathcal{I}$ can be measured by some objective function $\mathcal{F}(\mathcal{I})$ (e.g. annual performance) which can be computed by an annual simulation of the sun irradiation. The *layout optimization problem* can be specified as follows:

$$\begin{array}{ll} \max\limits_{\mathcal{I}} & \mathcal{F}(\mathcal{I}) \\ \textit{such that} & \mathcal{I} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq \Omega \textit{ and} \\ & \|(x_i, y_i) - (x_j, y_j)\| \geq d \textit{ for all } 1 \leq i < j \leq N \ . \end{array}$$

To solve the layout optimization problem, we use an *evolutionary algorithm* as a free variable method. That means, our optimizer is not based on any fixed pattern but offers the possibility to freely position heliostats inside a given area, as long as they have a sufficient distance to each other. The advantage is that a larger search space might contain more efficient solutions. If for any reasons a pattern is requested, our approach could be applied to any fixed topology in a quite straightforward manner.

The functionality of an evolutionary algorithm is inspired by the nature: In our setting, each layout configuration (*individual*) is specified by its properties (*genotype*), each property (*gene*) being either the $x$ or the $y$ coordinate value of a heliostat position. A *population* is a set of individuals. To measure the goodness of a population, we simulate all of its layout configurations to determine their *fitness values* (e.g. efficiency or received irradiation). The fitness of a population is its highest individual fitness value, which serves as the objective function for the optimization. The optimization algorithm starts from an initial population and iteratively derives a new population from the previous one until some *termination criterion* is fulfilled. This could either be a maximum number of iterations, the convergence of the last rounds, or just a time limit. Upon termination, the best individual that was generated during the whole optimization process is returned.

To not loose the best solutions at the transition from one population to the next and thus to assure monotonicity of the population fitness, we initialize a new population to contain a certain number of fittest individuals from the previous population (*elitism*). Additionally, in order to avoid settling in a local optimum, it is also possible to introduce a certain number of new *random individuals* to each population. Afterwards these steps, we iteratively derive new individuals from the previous population and add them to the new population if they satisfy the minimal distance requirements (otherwise they are discarded). This procedure is repeated until the new population has the same size as the previous population.

To derive a new individual, three major operations are used:

1. *Selection:* Two or more individuals are randomly selected from the previous population according to their fitness values.
2. *Crossover:* The properties of the selected individuals are recombined according to their fitness values.
3. *Mutation:* Some genes of the new recombined individual might be modified before adding it to the new population.

To ensure to find a good solution in appropriate time, some modifications to the classical operators are needed. In the following we describe the selection, crossover and mutation algorithms along with the termination criterion that we use in our layout optimization approach.

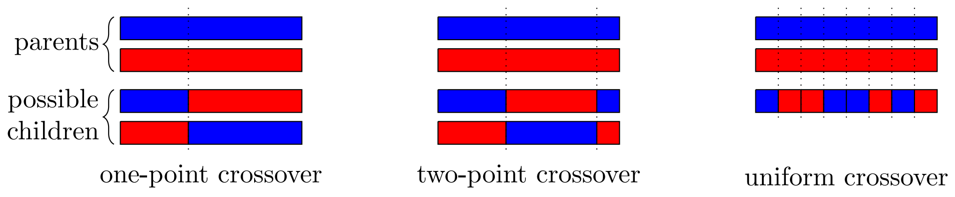


Figure 2: Classical crossover operators.

## 2.1 Selection

Different techniques were proposed in literature for the selection of individuals that should be recombined. In our work, we utilize one of the most common methods, called the *roulette wheel method*. The objective of the roulette wheel method is to select potentially useful individuals to contribute to a new population with improved fitness. For that purpose, the fitness values are used in order to associate a probability of selection with each individual. This means that from a population $\mathcal{P}$ an individual $\mathcal{I} \in \mathcal{P}$ with fitness value $F(\mathcal{I})$ is selected with probability

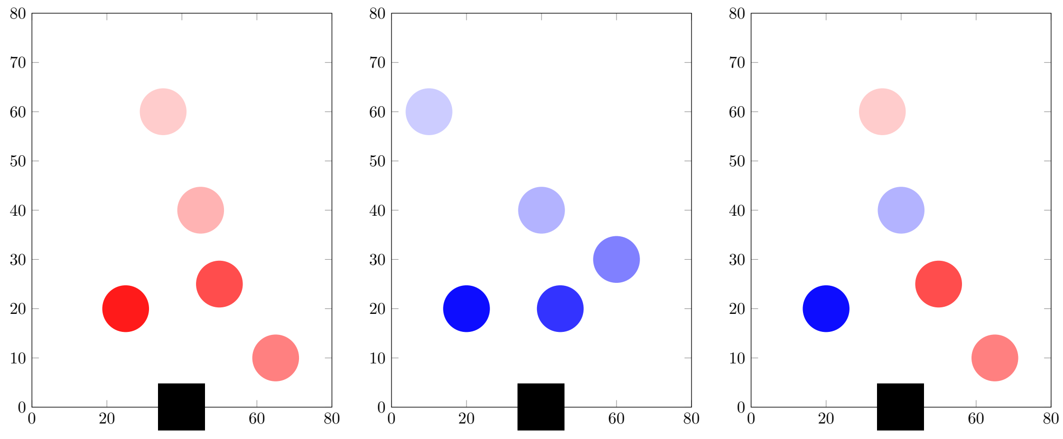
$$p(\mathcal{I}) = \frac{F(\mathcal{I})}{\sum\limits_{\mathcal{I}' \in \mathcal{P}} F(\mathcal{I}')} \in [0,1]. \tag{1}$$

## 2.2 Crossover

Using the above selection technique, we choose two *parent* individuals to be recombined into a new configuration (*child*). The classical crossover operators are *one-point crossover*, *two-point crossover* or *uniform crossover*. All three operators, illustrated on Figure 2, assume that the genes are stored as an ordered sequence of values. One-point crossover determines a sequence index, and generates children having genes from one of the parents up to the given index and from the other parents for larger indices. Two-point crossover works similarly but cuts the gene sequences at two points; children inherit genes from one of the parents for the indices between the two points and from the other parent for the remaining parts. Finally, uniform crossover determines for each gene in the sequence a parent, from which the gene is inherited, randomly using a uniform distribution (probability 50% for both parents).

The drawback of these crossover approaches is that many generated children violate the constraint of minimum distance and need to be sorted out. Additionally, these approaches are highly sensitive to the order of the heliostats in their genotype representation. Finally, when applying these approaches for layout optimization, it is meaningful to encapsulate heliostat positions, represented by two genes (one for the $x$ and one for the $y$ coordinate value). Therefore we need to adapt both the genotype representation as well as the crossover operators to the layout optimization problem.

**Genotype representation** Note that in our setting each gene is either an $x$ or an $y$ coordinate value. The classical genotype representation is an *ordered sequence of genes*. Instead of sequences, in our approach, the genotype representation is a *set of genes*, where a gene is a position $(x, y)$. Additionally, the set is combined with an order relation: The value of the objective function of a configuration is determined by an annual simulation of the sun irradiation, based on meteorological data. Besides the objective function value, the simulation also provides



(a) A mother individual (b) A father individual (c) A new valid child individual

Figure 3: Recombination of two parent individuals in crossover. The saturation of the color (red or blue) is a simplified measure for the goodness of the single heliostat.

information for each single heliostat, e.g., its power contribution to the overall received power. Based on this information, we order the heliostats of a configuration by their goodness. Using this genotype representation, we define three different crossover operators, which are adapted to the layout optimization problem.

**Zero-step crossover** First the genes (the heliostat positions) from the genotype representations of both parents are sorted in descending order according to their fitness. From this sorted base list, the heliostats with the highest goodness values are stepwise inserted into the child individual (and popped from the base list). If an inserted heliostat causes a conflict, it is skipped and the next heliostat is chosen. If there are no more heliostats left in the base list, the child configuration is completed with randomly generated heliostats. This way we generate only valid individuals. Figure 3 illustrates the zero-step crossover, where each of the parent genotypes contains five heliostat positions.

**One-step crossover** One weakness of the zero-step crossover operator is that the heliostats are weighted with their goodness in the parent individual, which does not guarantee to be a good choice in the child configuration due to new upcoming neighboring effects like blocking and shading. So, the heliostats' goodness for the generated child may not correlate to the one in the parent configuration.

In the one-step algorithm, we tackle this problem by placing all parent heliostats in decreasing goodness order into one field (skipping those which would affect collisions) and compute a new goodness value for each single heliostat. Based on these values, we select the desired number of heliostats for the child individual by choosing the best heliostats, similarly to the zero-step crossover (but now based on more appropriate goodness values).

**Multi-step crossover** The one-step crossover is based on improved goodness values, however, as not all the parent heliostats will be contained in the child configuration, these values still do not fully reflect the unique heliostat contributions to the final fitness value of the child.

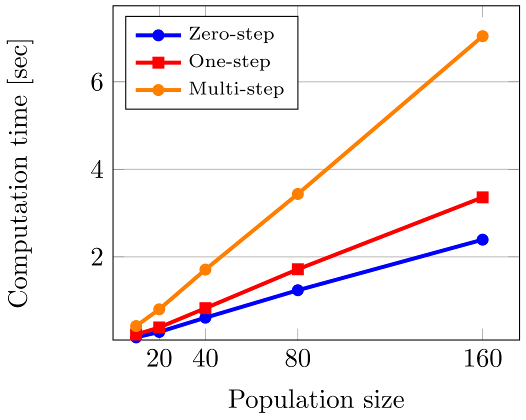


Figure 4: Crossover speed comparison.

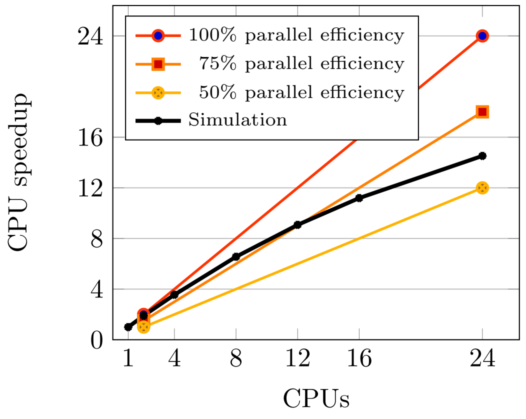


Figure 5: CPU speedup of an optimization with our evolutionary algorithm using a population size of 100 individuals.

Consequently, with this approach heliostats in densely placed groups could be completely sorted out due to neighboring effects.

Therefore, we define a third operator called multi-step crossover, which uses several evaluation steps. First the goodness value of each single heliostat is computed as if it would be the only one on the field, i.e., without considering neighboring effects. Based on this ranking, the best heliostat is chosen and added to the (initially empty) child configuration. Now for each remaining heliostat that does not collide with the already added one we re-compute its contribution to the child fitness if it would be added to the current (incomplete) child configuration, and insert the best one into the child's genotype. This process is repeated iteratively until the required number of heliostats is reached. Again, if there are no more parent genes, we complete the child genotype with random genes.

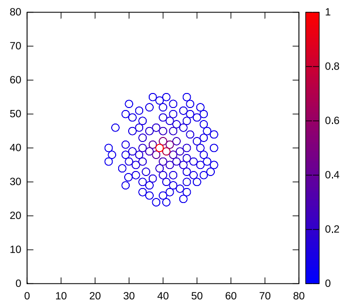
It is obvious that the number of "steps" reflects the computational effort for the different crossover operators; the multi-step crossover is far more expensive to compute than the zero- and one-step versions. The differences in the computation time are illustrated in Figure 4.

## 2.3 Mutation

After the recombination of two parent individuals to one child individual by applying a crossover operator, some child genes might be modified by random mutation. Mutation leads to additional diversity of individuals in the new population. The classical mutation would affect single genes in isolation, which would lead to a modification of either the $x$- or the $y$-position of single heliostats. In our application, we encapsulate the $(x, y)$-position as one gene, such that in the case of mutation the whole position is shifted with random distance in a random direction. If a conflict appears, the mutation is repeated again.

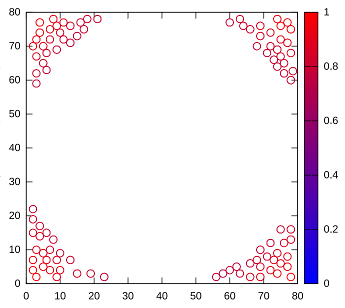
# 3 Experimental Results

We implemented our evolutionary algorithms with adapted genotype representation and crossover and mutation operators. For testing the quality of the different crossover opera-



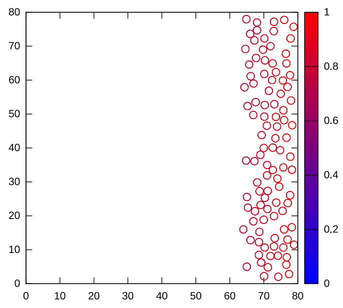
$f(x, y) = \frac{1}{\sqrt{50+(40-x)^2+(40-y)^2}}$

(a) Center example



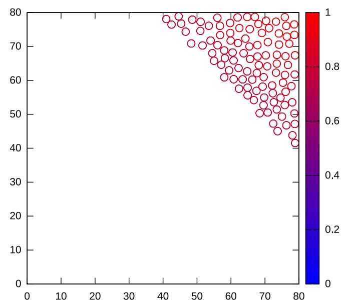
$f(x, y) = \sqrt{(40-x)^2 + (40-y)^2}$

(b) Corner example



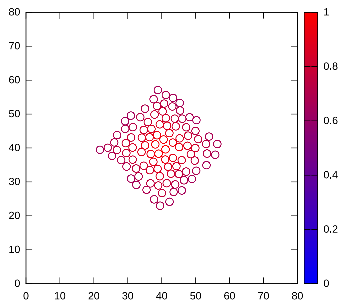
$f(x, y) = |x|$

(c) Max $x$ example



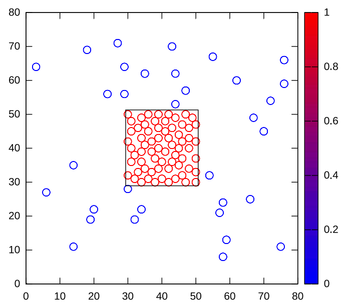
$f(x, y) = x + y$

(d) Triangle example



$f(x, y) = 50 - |40 - x| - |40 - y|$

(e) Rhombus example



Within rectangle

(f) Sphere packing example

Figure 6: Overview of the different test function results.

tors, we first replaced the complex simulation-based fitness evaluation by using more simple objective functions: For a given individual $\mathcal{I}$ with its genotype representation $\mathcal{I} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq \Omega$ we need to compute the objective function value $\mathcal{F}(\mathcal{I})$. Instead of using the simulation model we replace it by summing up function values $f : \mathbb{R}^2 \to \mathbb{R}$ for each heliostat:

$$\mathcal{F}(\mathcal{I}) = \sum_{i=1}^{N} f(x_i, y_i). \tag{2}$$

This test was successfully applied to all mentioned crossover operators. Figure 6 shows the results for the one-step crossover operator.

As the simulation-based fitness value computation is very time-consuming, we parallelized the optimization algorithm using OpenMP. Because the main workload of our optimization process is the calculation of the objective function values for every individual, we can apply the parallelization to this step. Due to the fact that each individual can be processed independently, the introduced parallelization overhead due to blocking is negligible. After these parallelized computations, an additional single-threaded pass over all individuals is performed to compute global values describing the whole population (e.g. min/max energy). The achieved speed-up by using parallelization is depicted in Figure 5. For reference, a program with 100% parallel efficiency scales linearly, i.e., the number of workers (here: CPUs) equals the speed-up factor, or more intuitively, it runs $n$ times as fast when $n$ times the number of CPUs are available. As depicted in the aforementioned figure, a typical optimization run with a population size of 100 individuals achieves more than 75% parallel efficiency when using 8 CPUs and still more than 50% when using 24 CPUs.

To show the applicability of our approach, we applied our algorithms to optimize two real solar tower power plants. As objective function the annual performance is used, which is defined as the fraction of irradiation energy received at the tower and the total energy reaching the mirrors without shading,

$$\mathcal{F}(\mathcal{I}) = \frac{\int_0^{8760} A \cdot \mathrm{DNI}(t) \cdot \eta(t)\ \mathrm{d}t}{\int_0^{8760} A \cdot \mathrm{DNI}(t)\ \mathrm{d}t}, \tag{3}$$

whereas 8760 is the number of hours in a year, $A$ is the mirror area, $\mathrm{DNI}(t)$ the time-dependent direct normal irradiation, and $\eta(t)$ the time-dependent efficiency of the field, considering cosine effects, blocking & shading of neighboring heliostats, interception efficiency and atmospheric attenuation [7, 9].

**Planta Solar 10 (PS10)** The PS10 solar tower power plant is placed near Seville, in Andalusia, Spain. Since 2007, the 11 megawatt (MW) solar power tower produces electricity with 624 large heliostats. Each heliostat has a mirror surface of 120 square meters, whereas the receiver is placed on top of a tower at 115 meters height. More details about the configuration can be found in [7].

We used our evolutionary algorithms to optimize the PS10 power plant, using a random initial population. As shown in Figure 7, using different crossover operators leads to different convergence rates for the optimization. Whereas the zero-step crossover shows a convergence behavior similar to the standard crossover operators, the one-step crossover shows a very fast convergence rate. The multi-step crossover converges, against our expectations, much slower. We suspect that this phenomenon is due to the fact that the multi-step crossover prefers highly efficient heliostat positions in the context of the current incomplete child genotype. Such heliostats are usually free-standing without other heliostats in their neighborhood. Whereas it pays off for the first few heliostats, this heuristics possibly reduces the possibilities for adding further efficient heliostat positions. According to the convergence rate of the multi-step crossover, we expect that problems with smaller fields show better convergence rates in optimization time.

We also compared the results of our approach to results using two different pattern-based optimization approaches (Figure 8). The first one is the original PS10 field layout, which was optimized using the radially staggered grid approach. Additionally we compare our results with the biomimetic approach that was applied to optimize the PS10 plant in [7]. The original field layout collects less energy than the biomimetic approach, and even less than our evolutionary algorithms. Our best approach using one-step crossover collects 0.2927% more energy than the original layout, and 1.7794% more energy than the biometric “sun flower” approach.

**Helio100** The Helio100 solar tower facility is placed near Stellenbosch, South Africa. 120 heliostats are used, each with a mirror surface of 2.2 square meters, whereas the receiver is

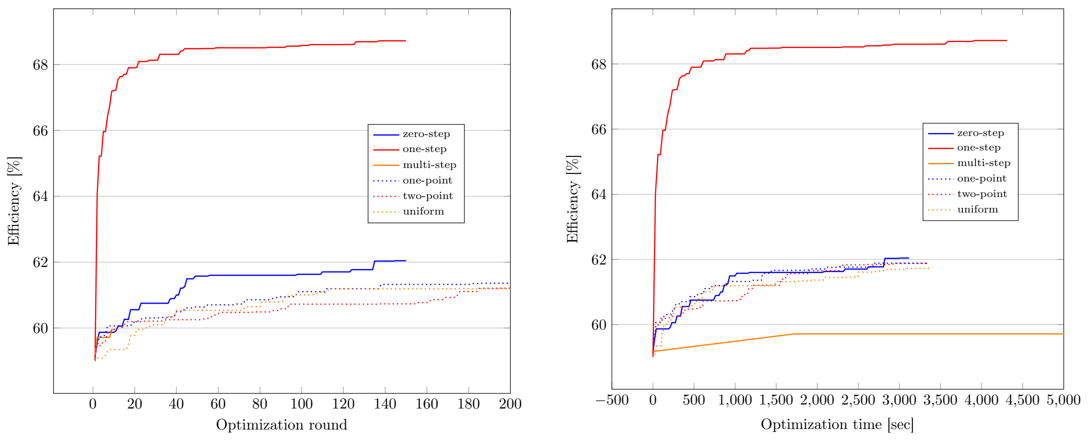
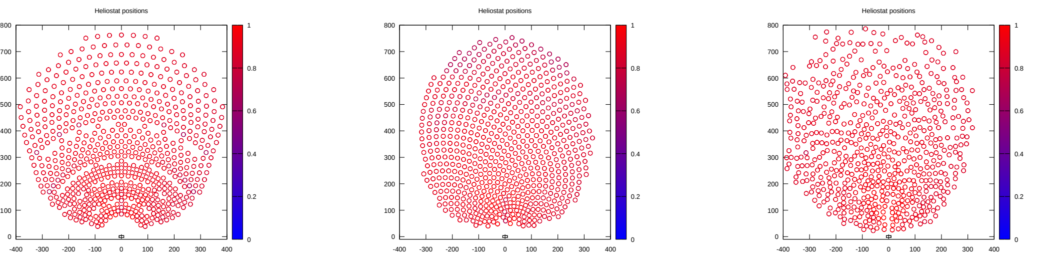


Figure 7: Comparison of crossover operators on PS10, plotting optimization progress vs. optimization steps (left) and vs. optimization time (right).



(a) Original PS10 layout as radially staggered grid: 66.5189% efficiency.

(b) Optimized layout using a biomimetic approach: 67.518% efficiency.

(c) Optimized layout using evolutionary algorithm with one-step crossover: 68.7194% efficiency.

Figure 8: Comparison of optimization algorithms on PS10.

placed on top of a tower at 12 meters height. The plant is in operation since 2015.

The behavior of different standard and adapted crossover operators is investigated in Figure 9. All three adapted crossover operators perform better than the classical ones. The approach using the one-step crossover operator reaches the fastest convergence rate. On this Helio100 application the multi-step crossover shows somewhat better performance than on PS10, possibly due to the smaller number of heliostats.

We compare the layout results of our optimization approaches to pattern-based optimization approaches and to the classical crossover operators in Figure 10. Our best result using the one-step crossover collects 7.5335% more energy than the original layout, and 0.0595% more energy than the biometric "sun flower" approach.

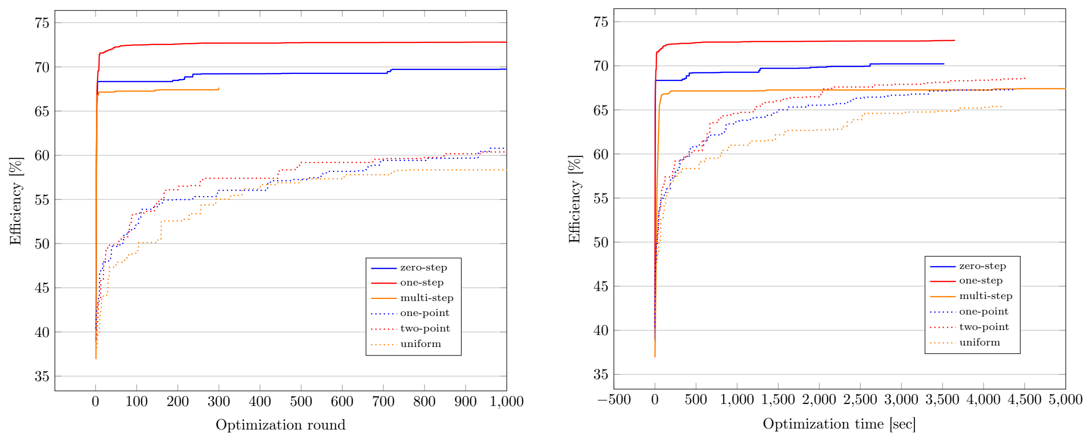


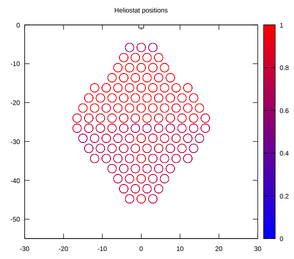
Figure 9: Comparison of crossover algorithms on Helio100, plotting optimization progress vs. optimization steps (left) and vs. optimization time (right).

## 4 Conclusion and Outlook

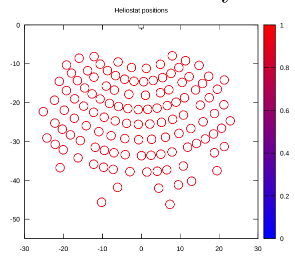
The heliostat layout optimization problem of solar tower power plants is a global, non-convex optimization problem with constraints. Usually this kind of problem is solved using pattern-based optimizers. The drawback of these methods is the small search space by construction, as the solution is always a regular heliostat field. Within this work we used an evolutionary algorithm to improve the solution of the heliostat layout problem. Because the classical crossover operators lead to invalid layouts, and additionally are highly sensitive to the order of the heliostats, we introduced three new crossover operators. All operators are successfully tested and applied to two benchmarks, showing the applicability of our approach.

The achieved heliostat layouts still offer space for further improvements. One idea can be to improve a multi-step strategy, where the result from the evolutionary algorithm is improved by applying a local optimization strategy as a post-processing step. The latter can either be a gradient descent method by using algorithmic differentiation, or just a simple greedy heuristic, which randomly selects heliostats, moves them in random directions for random distances, and if the new heliostat position yields a higher fitness then leave it there, and undo the modification otherwise.

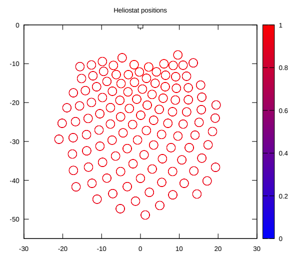
Pattern-based layouts



(a) Row-pattern-based original field layout 65.3512% efficiency

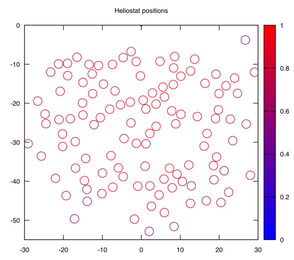


(b) Radially staggered grid approach 72.127% efficiency

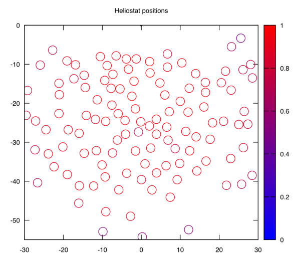


(c) Biomimetic approach 72.8252% efficiency

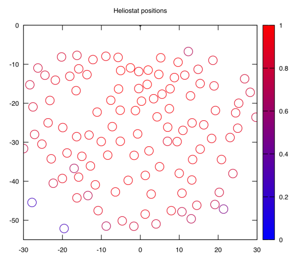
Evolutionary methods: classical crossover operators



(d) One-point crossover 67.2945 % efficiency

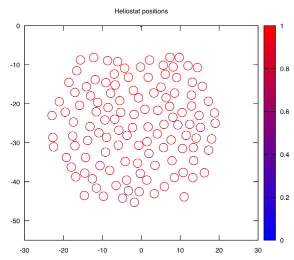


(e) Two-point crossover 68.6612 % efficiency

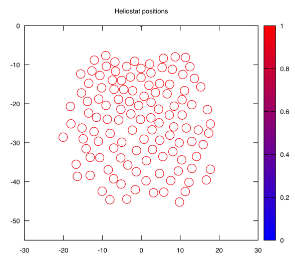


(f) Uniform crossover 65.5853 % efficiency

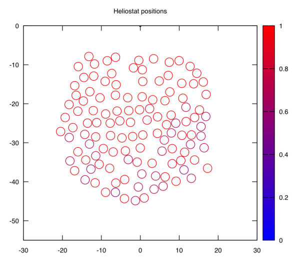
Evolutionary methods: adapted crossover operators



(g) Zero-step crossover 70.2242% efficiency



(i) One-step crossover 72.8847% efficiency



(h) Multi-step crossover 67.6485 % efficiency

Figure 10: Results for different layout optimization algorithms on Helio100.

## References

[1] R. Buck. Heliostat field layout using non-restricted optimization. In *SolarPACES Conference*, 2012.

[2] E Carrizosa, C Domínguez-Bravo, E Fernández-Cara, and M Quero. An optimization approach to the design of multi-size heliostat fields. Technical report, Technical report IMUS. 2014 URL:{http://www. optimization-online. org/DB_HTML/2014/05/4372. html}, 2014.

[3] P. Garcia, A. Ferriere, and J.-J. Bezian. Codes for solar flux calculation dedicated to central receiver system applications: a comparative review. *Solar Energy*, 82(3):189–197, 2008.

[4] Greens MPs. PS10 concentrating solar thermal power plant. https://www.flickr.com/photos/greensmps/6497633977, Last visited June 2016. Licence: https://creativecommons.org/licenses/by-nc-nd/2.0/.

[5] F.W. Lipps and L.L. Vant-Hull. A cellwise method for the optimization of large central receiver systems. *Solar Energy*, 20(6):505–516, 1978.

[6] S.L. Lutchman, A.A. Groenwold, P. Gauché, and S. Bode. On using a gradient-based method for heliostat field layout optimization. *Energy Procedia*, 49:1429–1438, 2014.

[7] C.J. Noone, M. Torrilhon, and A. Mitsos. Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy*, 2011.

[8] P. Richter, E. Ábrahám, and G. Morin. Optimisation of concentrating solar thermal power plants with neural networks. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 190–199. Springer, 2011.

[9] P. Richter, M. Frank, and E. Abrahám. Multi-objective optimization of solar tower heliostat fields. In *Progress in Industrial Mathematics at ECMI 2014*, page 357. Springer, 2014.

[10] M. Sanchez, M.and Romero. Methodology for generation of heliostat field layout in central receiver systems based on yearly normalized energy surfaces. *Solar Energy*, 80(7):861–874, 2006.

[11] P. Schramek, D.R. Mills, W. Stein, and P. Le Lièvre. Design of the heliostat field of the CSIRO solar tower. *Journal of Solar Energy Engineering*, 131(2):024505, 2009.