Multi-objective optimization of solar tower heliostat fields

Pascal Richter, Martin Frank and Erika Ábrahám

Abstract We introduce a model to compute the annual performance of a heliostat field. We take into account topography, tracking errors, and the position and intensity of the sun. An approach is introduced, which improves on the otherwise expensive pairwise comparison to calculate shading and blocking. Because the computational time is reduced significantly, the presented implementation is sufficiently fast to allow for heliostat field layout optimization within a couple of hours. The optimization is executed via a genetic algorithm, which optimizes the heliostat positioning parameters as well as other design parameters, e.g. receiver tilt angle. A novel approach is used to reduce the search domain. Because the search domain delivers several local optima with comparable values of the objective function, the objective function is augmented. We use smoothing functionals to disperse the local optima. A field layout is optimized on a hilly ground in South Africa, with additional constraints on the heliostat positions.

1 Introduction

Solar tower plants generate electric power from sunlight by focusing concentrated solar radiation on a tower-mounted receiver, see Fig. 1. The collector system uses hundreds or thousands of sun-tracking mirrors called heliostats, to reflect the inci-

Pascal Richter
RWTH Aachen University, Center for Computational Engineering Sciences, Schinkelstrasse 2, 52062 Aachen, Germany, e-mail: richter@mathcces.rwth-aachen.de

Martin Frank
RWTH Aachen University, Center for Computational Engineering Sciences, Schinkelstrasse 2, 52062 Aachen, Germany, e-mail: frank@mathcces.rwth-aachen.de

Erika Ábrahám
RWTH Aachen University, Chair of Computer Science 2, Ahornstrasse 55, 52074 Aachen, Germany e-mail: abraham@informatik.rwth-aachen.de

dent sunlight onto the receiver where a fluid is being heated up. Today's receiver types use water/steam, air or molten salt to transport the heat. Usually, the heat of the fluid is exchanged into steam which powers a turbine to generate electricity.

Fig. 1 Solar tower plant PS10, 11 MW in Andalusia, Spain. [Source: flickr]

Solar tower plants are not yet cost-competitive [6]. Therefore, concentrating solar thermal power plant markets and projects today only evolve where a political framework ensures financial incentives. For commercial solar tower project developments, a conceptual plant design has to be determined in an early planning stage. Designing commercial power plants aims always at finding the most economic plant design under a given set of constraints.

In this paper a model and optimisation algorithm for heliostat field layout is introduced. The underlying solar tower model is presented in section 2. Because the model is used in an optimisation process, a computationally efficient calculation of insolation with enough accuracy is needed.

2 Ray-tracing model

A solar field is given by N heliostats H_i, each with an area A_i. For the time-dependent solar angles θ_{solar} and γ_{solar}, and the direct normal irradiation I_{DNI}, the ray-tracing model computes the received optical radiation over a year, while taking cosine effects $\eta_{\cos}$, shading and blocking η_{sb}, heliostat reflectivity η_{ref}, atmospheric attenuation η_{aa} and spillage losses η_{spl} into account. For each heliostat H_i the time dependent received optical radiation is defined by

$$P^i(t,d) = A_i \cdot I_{\text{DNI}}(t,d) \cdot \eta_{\cos,i}(t,d) \cdot \eta_{\text{sb},i}(t,d) \cdot \eta_{\text{ref},i}(t,d) \cdot \eta_{\text{aa},i}(t,d) \cdot \eta_{\text{spl},i}(t,d), \tag{1}$$

at time t of the day d. At the end of this section the annual received radiation of the full plant is computed, which depends on the sunrise and sunset of every day in the year. We essentially use the same model as [7], the main difference being that we use a hierarchical ray-tracing method, and speed-up the computation of shading and blocking effects.

Hierarchical ray-tracing method
The rays have their origin in the sun, hit the surface of a heliostat and are reflected in direction of the receiver. We are interested in the reflected power of a heliostat, which is hitting the receiver. To detect the optical flux over the heliostat's surface we are using a hierarchical approach of ray-tracing methods [1, 7], where the complete flux is computed by numerical integration with the use of Gauss-Legendre quadrature rule. Thus, the surface is partitioned in a number of regions, each with a representative ray. The influence on the reflection by shading, blocking and ray interception at the receiver is determined just for this single ray as representative for the whole region. Each area is weighted by the irradiance of its representative ray. Finally all values are summed to get the power of the heliostat. The number of representative rays per heliostat is given by the selected order of the Gaussian quadrature rule.

Shading and blocking
For each representative ray of a heliostat, shading and blocking effects by neighbouring heliostats or the tower must be detected. This is the most expensive part of a simulation. The brute-force approach of a pairwise comparison of each ray with all heliostats is computationally expensive. The computational complexity can be reduced by only considering a subset of heliostats that can potentially shade or block a heliostat [1]. To determine this subset, a data structure is needed which is fast in nearest-neighbour search and in range-search.

Therefore, for better performance, a two-dimensional bitboard index structure is used. The idea is to cover the two-dimensional *x*-*y* space with an equidistant grid such that the space is sub-divided in distinct quadratic cells. Inside those cells the information is stored if nearby is a heliostat.

For a nearest-neighbour search, just the surrounding cells around a cell have to be checked, instead of all heliostats. The same holds for a range-search, where e.g. all heliostats in one direction are wanted. Just the containing cells of the range have to be checked. In some test cases we could accelerate the simulation by factor 100.

Annual received optical radiation
The annual received optical radiation of the whole power plant is given by the sum of the annual received optical radiation of all heliostats H_i,

$$E_{\text{year}} = \sum_{i=1}^{N} E^i_{\text{year}} = \sum_{i=1}^{N} \sum_{d=1}^{365} \left(\int_{\text{sunrise}}^{\text{sunset}} P_i(t,d)\ \mathrm{d}t \right), \tag{2}$$

with power P_i given in equation (1). The sunrise and the sunset depend on the day d. The value of the received optical radiation over a year E_{year}, is the basis for each objective function in the optimisation process, see section 3. For each different configuration of the solar field, this value has to be computed by a simulation.

The time integral from sunrise to sunset in (2) is solved numerically. In common practice, an iteration with constant time step [3, 5, 10] is used. Noone et al. [7] propose an iteration with constant solar angle step, which allows the same accuracy but needs fewer iterations. Both approaches approximate the time integral with midpoint rule. For higher accuracy other numerical quadrature rules are recommended. The herein proposed Gauss-Legendre quadrature rule uses non-constant time (or solar angle) steps:

$$\int_{a:=\text{sunrise}}^{b:=\text{sunset}} P_i(t,d)\,\mathrm{d}t \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i \cdot f\Big(\frac{b-a}{2}t_i + \frac{a+b}{2}\Big), \tag{3}$$

with n Gaussian time-abscissas t_i and Gaussian weights w_i. Additionally the sum of the days can be approximated by using a sort of trapezoidal rule with just $m \in \{1,2,\ldots,365\}$ days.

3 Optimisation

Various effects – cosine effects, shading and blocking of heliostats (presented in section 2) – reduce the efficiency of the solar tower. An objective of an optimisation is to discover an optimal positioning of the heliostats in the field. In the literature, the general structure of the heliostat arrangement is predefined by assumptions, e.g. radial staggered, circles or spirals [7–10]. In these cases, an optimisation means to find an assignment of about two to four parameters which define the structure, e.g. radius or angle of a spiral. However, the assumption of the structure leads to many comparable local optima [9]. In addition, these optimisations generate a regular or symmetric structure which could be suitable for nearly flat areas but not for a hilly topology.

In this work, we introduce an approach where the heliostats' alignment does not depend on any structure. Namely, the heliostats obtain the highest amount of freedom in order to find their optimal position. For that purpose, we use a genetic algorithm [2, 4] with a novel genotype-representation which reduces the search domain of the algorithm. The only restriction of the approach is that neighbouring heliostats must be separated by a minimum distance in order to prevent a collision.

Genetic algorithm
The functionality of a genetic algorithm is inspired by the biological evolution. A population of candidate solutions, called individuals, evolves in order to provide better solutions for an optimisation problem. Each individual has a set of properties, called genotypes or genomes. Usually, a genotype is represented as an array of several genes such that a unique assignment of gene and property exists. Two or more individuals are combined by mixing the genotypes gene by gene in order to generate a new population. Using this approach for the position of heliostats, the sets of heliostats from different heliostats are "ordered" by an artificial identifier. This identifier is defined by the position of the corresponding gene in the array. To generate a new population of individuals from a set of evaluated individuals, four main operations are used by the genetic algorithm: First, two or more individuals are randomly selected by roulette-wheel method from the old population according

to their fitness values. The properties of the selected individuals are combined according to the fitness value of their heliostats. Therefore the heliostats of all parent individuals are sorted in descending order according to this value. Successively the best heliostats are picked for the new individual. If any selected heliostat causes a conflict, it is neglected and the next best heliostat is picked. In case that there are no more heliostats, the remaining heliostats are generated by random. Afterwards, the heliostats are mutated by locally change their position. The whole population is simulated to get the fitness values for the new individuals. The algorithm terminates if a stop criterion is satisfied, e.g. maximum simulation time or maximum number of generations.

Objective functions

The purpose of our optimisation is to find an individual which has a high efficiency. But additionally we aim to reward a solution which looks "nice", which means, that the distribution of the heliostat positions are somehow smooth. We scalarise our objective function, i.e. we look for a solution

$$\max_{\mathscr{I}\in\mathscr{P}} F(\mathscr{I}) = \max_{\mathscr{I}\in\mathscr{P}} \left(\sum_{j=1}^{n} w_j \cdot \frac{f_j(\mathscr{I}) - \min_{\mathscr{I}\in\mathscr{P}} f_j(\mathscr{I})}{\max_{\mathscr{I}\in\mathscr{P}} f_j(\mathscr{I}) - \min_{\mathscr{I}\in\mathscr{P}} f_j(\mathscr{I})} \right), \tag{4}$$

All objective functions f_j are normalised by the minimum and maximum value of the whole population $\mathscr{P}$, so that the normalised value lies in the range between 0 and 1. The weights of the objectives $w_j > 0$, with $\sum_{j=1}^{n} w_j = 1$, are the parameters of the scalarisation. Every objective function f_j has to be maximised. If there is an objective function $\hat{f}_j$ that should be minimised, we set the corresponding objective function as $f_j := -\hat{f}_j$ which is maximised.

The model described in section 2 delivers in equation (2) the annual received optical radiation, which is used as objective funtion

$$f_1(\mathscr{I}) := E_{\text{year}}(\mathscr{I}). \tag{5}$$

To reward solutions, which are looking "nice", additional smoothing functionals are created. The variance of the k-nearest-neighbour distance is given by,

$$f_2(\mathscr{I}) = -\iint |\nabla \mathsf{KNN}|^2 \, \mathrm{d}x\,\mathrm{d}y \approx -\sum_{T\in\mathscr{T}} A_T \cdot \left(\frac{\partial \mathsf{KNN}(T)}{\partial x} + \frac{\partial \mathsf{KNN}(T)}{\partial y} \right)^2, \tag{6}$$

$\mathscr{T}$ denotes the triangulation of the heliostats H_i in the x-y plane. A_T is the area of a triangle $T \in \mathscr{T}$. For each heliostat H_i the k-nearest-neighbour distance is given by

$$\mathsf{KNN}_i = \sum_{H_\ell \in \mathscr{N}_k(H_i)} |p_i - p_\ell| \tag{7}$$

where $\mathscr{N}_k(H_i)$ is the set of the k nearest neighbours of H_i and p_i and p_ℓ are the positions of the heliostats. By linear interpolation the k-nearest-neighbour distance is piecewise defined for each triangle $T \in \mathscr{T}$ which is denoted by $\mathsf{KNN}(T)$.

Another smoothing functional is the density distribution, which is given by

$$f_3(\mathscr{I}) = -\iint |\nabla\rho|^2 \,\mathrm{d}x\,\mathrm{d}y \approx -\sum_{T\in\mathscr{T}} A_T \cdot \left(\frac{\partial\rho^r(T)}{\partial x} + \frac{\partial\rho^r(T)}{\partial y}\right)^2. \tag{8}$$

For each heliostat H_i the density is given by

$$\rho_i^r = \left|\{H_\ell \in \mathscr{I} \;\mid\; |p_i - p_\ell| \le r\}\right| \tag{9}$$

where ρ_i^r is the number of H_i–neighbouring heliostats inside a defined radius r. Again by linear interpolation the density is piecewise defined for each triangle $T \in \mathscr{T}$ which is denoted by $\rho^r(T)$.

The variance of the kNN distance and the density distribution functionals aim to create a field of equally distributed or dense heliostats. The importance of the smoothing functionals can be adjusted by using the weights described in equation (4).

Testing the genetic algorithm

By combining the three functionals of energy, kNN and density, different results are reached, see Fig. 2. The produced energy is high for all combinations. By taking kNN and density into account, the optimisations yields a field in which the heliostats stand closer together and are evenly distributed. The single outliers that occur due to mutation could be eliminated during a post-processing step.

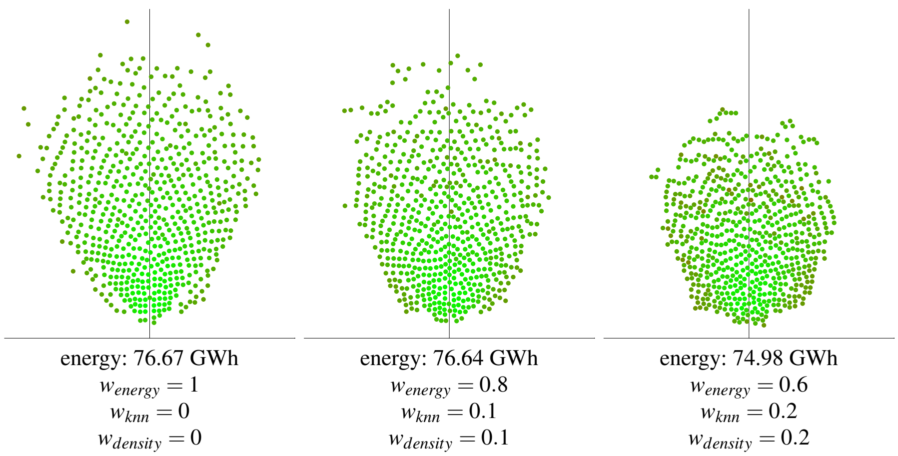
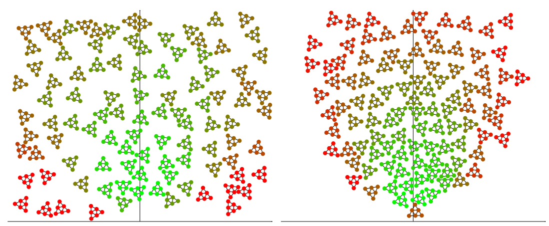


Fig. 2 Comparison of the best power plant configurations after optimisation with different weights for the density and knn functional. The color gradient from green to red shows the annual received optical radiation for each heliostat.

Using the smoothing functionals, it is possible to create “nicer looking” solutions that result in comparable energy production. For further fine-tuning one could also try to gradually adjust the weight of the functional during the course of the optimisation.

4 Application

With the above introduced optimisation algorithm a solar power plant can be optimized. To optimize the heliostats alignment of a planned pilot plant in South Africa we had to extend the model in such a way, that the heliostats can be grouped by a joint pod system, where they are positioned on an arbitrary truss construction. So, instead of positioning single heliostats, groups of heliostats with fixed relative positions are placed on the field. The pod systems are not allowed to touch each other, this includes all heliostats and the truss construction. Fig. 3 shows the distribution of the heliostats before and after the optimisation.



Initial configuration After 100 optimization steps

Fig. 3 Optimisation of a solar power plant with pod systems.

References

1. Belhomme, B., Pitz-Paal, R., Schwarzbözl, P., Ulmer, S.: A fast ray tracing tool for high - precision simulation of heliostat fields. Solar Energy Engineering-Transactions of the ASME 131 (3) (2009)
2. De Jong, K.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan (1975)
3. Elsayed, M., Allah, H., Al-Rabghi, O.: Yearly-averaged daily usefulness efficiency of heliostat surfaces. Solar energy **49**(2), 111–121 (1992)
4. Holland, J.: Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press (1975)
5. Kistler, B.: A user's manual for DELSOL3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants. Tech. rep., Sandia National Labs., Livermore, CA (USA) (1986)
6. Morin, G.: Techno-economic design optimization of solar thermal power plants. Ph.D. thesis, Technische Universität Braunschweig (2010)
7. Noone, C., Torrilhon, M., Mitsos, A.: Heliostat field optimization: A new computationally efficient model and biomimetic layout. Solar Energy (2011)
8. Noone, C.J., Ghobeity, A., Slocum, A.H., Tzamtzis, G., Mitsos, A.: Site selection for hillside central receiver solar thermal plants. Solar Energy **85**(5), 839–848 (2011)
9. Pitz-Paal, R., Botero, N., Steinfeld, A.: Heliostat field layout optimization for high-temperature solar thermochemical processing. Solar Energy **85**(2), 334–343 (2011)
10. Yao, Z., Wang, Z., Lu, Z., Wei, X.: Modeling and simulation of the pioneer 1mw solar thermal central receiver system in china. Renewable Energy **34**(11), 2437–2446 (2009)