

Diese Arbeit wurde vorgelegt am
Lehr- und Forschungsgebiet Theorie der hybriden Systeme

**Mehrstufige Layoutoptimierung von Heliostatfeldern in
solarthermischen Kraftwerken**
**Multi-Step Layout-Optimization of Heliostat Fields in
Central Receiver Systems**

Bachelorarbeit
Informatik

Februar 2021

Vorgelegt von Presented by	Lars Fischer Rosstraße 7 52064 Aachen Matrikelnummer: 368030 lars.fischer@rwth-aachen.de
Erstprüfer First examiner	Prof. Dr. rer. nat. Erika Ábrahám Lehr- und Forschungsgebiet: Theorie der hybriden Systeme RWTH Aachen University
Zweitprüfer Second examiner	Prof. Dr. rer. nat. Thomas Noll Lehr- und Forschungsgebiet: Software Modellierung und Verifikation RWTH Aachen University
Externer Betreuer External supervisor	Dr. rer. nat. Pascal Richter Steinbuch Centre for Computing Karlsruhe Institute of Technology

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen und Abbildungen. Diese Arbeit hat in dieser oder einer ähnlichen Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Aachen, Februar 2021

Lars Fischer

Contents

1. Introduction	1
1.1. Related work	1
1.2. Outline	2
2. Concentrated solar power plant model	3
2.1. Optical model	3
2.2. Thermal model	9
2.3. Storage model	9
2.4. Thermal-hydraulic model	12
2.5. Economic model	12
2.6. Modeling existing power plants	15
2.7. Validation	18
2.7.1. Basic test cases	18
2.7.2. Ray resolution study	24
2.7.3. Existing power plants	25
3. Heliostat layout-optimization	29
3.1. Pattern optimization	29
3.1.1. Staggered Cornfield	29
3.1.2. Hexagon	31
3.1.3. Radial staggered	31
3.1.4. Spiral	34
3.1.5. Pattern optimization algorithms	35
3.2. Local Search optimization	41
4. Case study	50
4.1. Layout optimization for PS10	51
4.2. Layout optimization for PS20	53
4.3. Layout optimization for Gemasolar	55
4.4. Discussion	57
5. Conclusion	58
References	60
A. Optimized heliostat layouts	63

1. Introduction

Conventional energy production is based on the burning of fossil resources such as coal, natural gas or oil. This form of energy production has several problems. Firstly, fossil resources are finite on earth, which means that conventional energy production is not renewable and cannot function in the long run. On the other hand, burning the carbon-rich fossil fuel always emits CO_2 , resulting in the energy production industry being responsible for the majority of global greenhouse gas emissions [3]. Because of the devastating consequences of greenhouse gas emissions and the associated climate change, renewable energy production is an important component in meeting the goals of the Paris Agreement [4].

The production of renewable energy is based on renewable resources such as sunlight or wind and has become increasingly popular in recent decades. With the growth of renewable energy production, the levelized cost of energy (LCOE) for all renewable energy technologies is steadily decreasing. Moreover, the LCOE of renewable energy technologies is expected to continue to decrease in the future. This will enable renewable energy production technologies to compete even more successfully with conventional energy production from fossil fuels [8].

Concentrated solar power (CSP) is a promising renewable energy technology best suited for regions with high solar radiation. CSP is becoming more and more popular as its global capacity has reached 6.275 GW in 2019, an increase of almost 500% over 2010 [7]. The energy production of CSP plants is based on the accumulation of solar radiation in a receiver mounted on a tower. The tower is surrounded by a large number of mirrors, called heliostats, mounted on tracking systems. Each heliostat aligns its mirrors with the position of the sun so that solar radiation is reflected into the receiver. Inside the receiver, a heat transferring fluid is heated by solar radiation, which is then used to evaporate water that drives a turbine, producing electrical energy.

Heliostats are influenced by multiple position dependant effects that lessen the solar radiation reflected onto the receiver and thus their efficiency. Therefore, the goal of heliostat layout optimization is to maximize the heliostat efficiencies, which maximizes the performance of the CSP plant.

1.1. Related work

CSP plants have been optimized since the 1980s. For the optimization of heliostat layouts, a distinction can be made between the class of *pattern-based* and *pattern-free* optimization algorithms. The prevalent methods are the *pattern-based* optimization algorithms, which place heliostats on geometric patterns to reduce the complexity of the optimization problem. The most common pattern, which can be found in many commercial CSP plants, is the radial staggered pattern. It was originally proposed along with the staggered cornfield pattern by the University of Houston for their *RCELL* code [25, 24]. Other optimization tools such as *DELSOL* [23], *MUEEN* [33] and *Campo* [14] are also based on the radial staggered pattern. Other patterns include the more recent spiral pattern [28] or the hexagon pattern [29] as well as hybrid patterns, which com-

bine multiple patterns, for example the spiral with the radial staggered pattern [36]. A detailed comparison of these state-of-the-art heliostat layout algorithms was done by Barberena et al. [10].

The pattern-free optimization algorithms do not use patterns for heliostat positions, since patterns drastically limit the search space and optima are unlikely to be found. Among these are methods which fill the heliostat field step-by-step such as the *heliostat growth method* [32] or the *expansion-contraction algorithm* [13], which can also optimize for multiple heliostat sizes. Other pattern-free methods start from an existing or random initial heliostat layout and redesign it until an optimal layout is found. There are gradient-based methods such as the *free variable method* presented by Lutchman et al. [26] or the *physics-based gradient method* of Yang et al. [35]. Another class of pattern-free optimization algorithms are local search algorithms, which relocate heliostats individually. These include the *nonrestricted refinement method* of Buck [12], as well as the local search methods of Reinholz et al. [30] and Kim et al. [22].

While the advantage of *pattern-based* optimization algorithms is the reduced search space of the optimization problem, their disadvantage is that the optimized heliostat layout generated from a pattern is unlikely to be an optimum. Therefore, this work increases the search space for the *pattern-based* heliostat layouts by introducing two new parameters, which scale the pattern along an arbitrary axis. Furthermore, the optimized heliostat pattern layout is refined by a local search to find local optima, which can not be found inside the reduced search space of the pattern optimization. Overall, this work presents a novel multi-step layout optimization pipeline, which combines a *pattern-based* optimization algorithm with a *pattern-free* local search algorithm. Multi-step optimizations combining global and local optimization have, to our knowledge, already been applied to offshore wind farms [16], but not yet to heliostat layouts.

1.2. Outline

This work uses an existing model for CSP plants, which is implemented in the **SunFlower** project, written in C++. With **SunFlower** it is possible to quickly simulate a CSP plant to calculate the annual energy production for a given heliostat layout, while taking weather data into account. The multi-step optimization pipeline introduced in this work was implemented for the **SunFlower** project.

The underlying model is described and validated in Section 2. The multi-step optimization algorithm is introduced in Section 3 and consists of a heliostat pattern optimization and a local search algorithm. Section 4 presents and discusses a case study for the developed optimization algorithm. Finally, Section 5 concludes on the presented multi-step optimization pipeline and provides limitations as well as an outlook for future research.

2. Concentrated solar power plant model

To properly optimize a CSP plant, a fast and accurate model must be developed, which is used to simulate a CSP plant and evaluate the performance of the heliostat field. Using the simulation, it is then possible to make changes to the heliostat layout to further improve its performance.

This section presents a CSP model which is capable of calculating both economic values as well as the energy production over a year. An overview of this model can be seen in Figure 1. It is divided into five sub-models, which are described individually in the following sections.

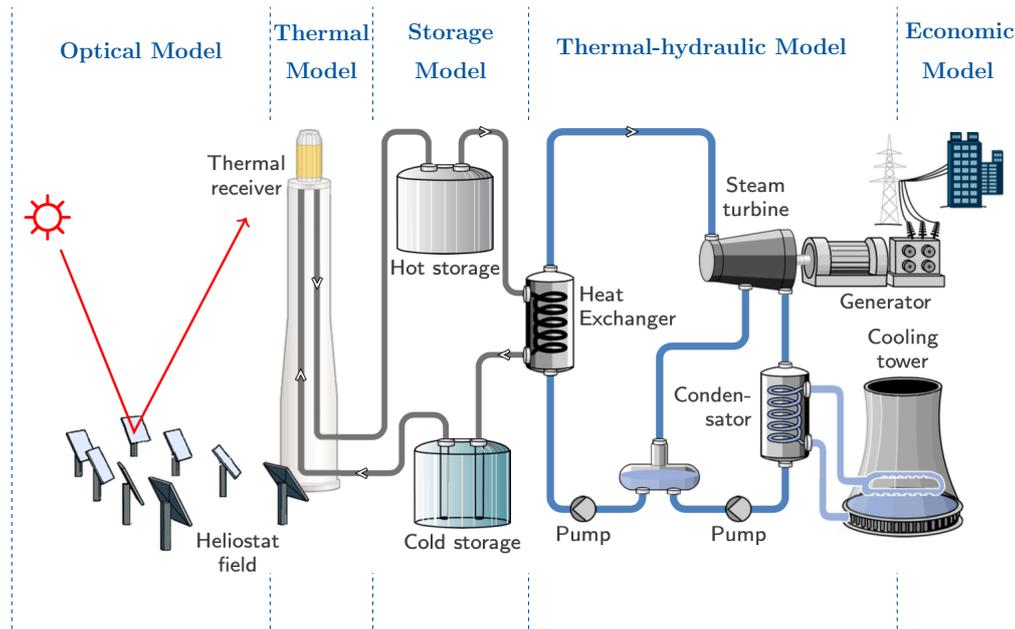


Figure 1: An overview of the complete CSP model, which consists of an optical model, a thermal model and storage model, as well as a thermal-hydraulic and economic model [17].

2.1. Optical model

The optical model is adopted from the models of Richter [31] and Franke [17]. At a given time, it considers the set of heliostats, the sun position and the weather for the computation of the accumulated optical radiation on the receiver. Various effects that attenuate the radiation or even block it completely are also taken into account.

Site The power plant site is given as a rectangle with four geographic coordinates for the corners and one geographic coordinate for the tower position. Every geographic coordinate is a pair of latitude and longitude values. Although the site is defined by

geographic coordinates, the local coordinate system uses cartesian coordinates with the tower position as its origin. The x-axis points towards east, while the y-axis points towards north and the z-axis points vertically upwards. The z-position for any position (x, y) is fixed, as the terrain for any power plant site implies the elevation for (x, y) . Furthermore, there may be restricted areas, which must not contain any heliostat. Each restricted area is given as a polygon.

Sun The position of the sun is given by its azimuth γ_{solar} and altitude θ_{solar} angles. These angles are used to compute the solar vector

$$\boldsymbol{\tau}_{\text{solar}} = \begin{pmatrix} \sin(-\gamma_{\text{solar}}) \cdot (-\cos(\theta_{\text{solar}})) \\ \cos(-\gamma_{\text{solar}}) \cdot \cos(\theta_{\text{solar}}) \\ \sin(\theta_{\text{solar}}) \end{pmatrix}. \quad (1)$$

With the solar vector $\boldsymbol{\tau}_{\text{solar}}$, illustrated in Figure 2, and the parameter for direct normal irradiation I_{DNI} in $[Wm^{-2}]$, the direction and intensity of the radiated sunlight can be computed.

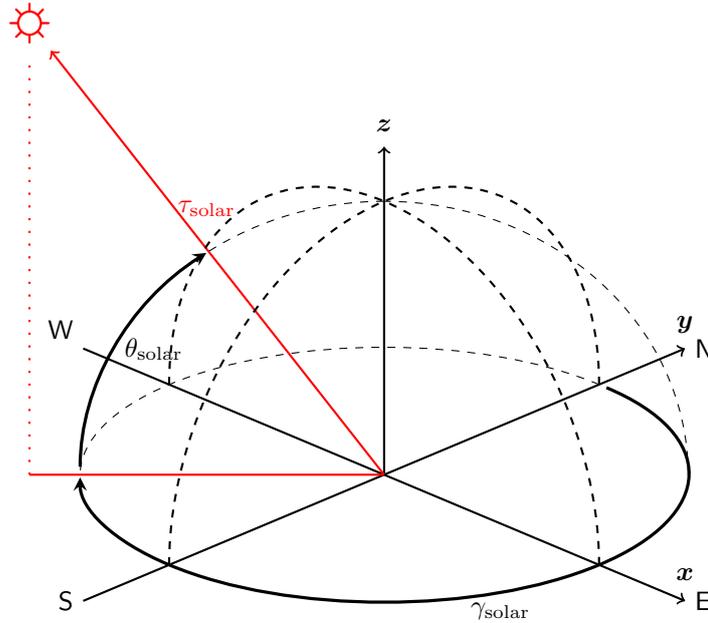


Figure 2: Illustration of the solar vector $\boldsymbol{\tau}_{\text{solar}}$ for a given solar altitude θ_{solar} and solar azimuth γ_{solar} , which is measured clockwise starting from north [17].

Heliostats A solar field is given by N_{hel} heliostats, such that for $i \in \{1, \dots, N_{\text{hel}}\}$, H_i denotes the i -th heliostat. Each heliostat consists of a pillar on which a frame is mounted that holds a layout of several mirrors called facets. The central position of

the facets is called \mathbf{p}_i and the sum of the facet areas is denoted by A_i . A heliostat has width w_i , length ℓ_i and a minimum bounding sphere with diameter $d_i = \sqrt{\ell_i^2 + w_i^2}$. For a heliostat H_i to be valid, its bounding sphere must be completely inside the site polygon and must not intersect with a restricted area or collide with another heliostats bounding sphere H_j , while considering an additional safety and maintenance distance. For practical reasons each heliostat H_i has its own local coordinate system, shown in Figure 3, which uses its center position \mathbf{p}_i as the origin. There exists a translation from points on the local coordinate system to the global coordinate system that uses the alignment of the heliostat. Each heliostat is aligned to reflect the sun rays to a target point $\mathbf{p}_{\text{aim},i}$ on the receiver. As illustrated in Figure 4, the solar entrance vector $\boldsymbol{\tau}_{\text{solar}}$ is reflected along the surface normal to the reflection vector \mathbf{r}_i , which aims at $\mathbf{p}_{\text{aim},i}$. The normalized reflection vector and the resulting surface normal of the heliostat can be computed as

$$\mathbf{r}_i = \frac{\mathbf{p}_{\text{aim},i} - \mathbf{p}_i}{|\mathbf{p}_{\text{aim},i} - \mathbf{p}_i|}, \quad (2)$$

$$\mathbf{n}_i = \frac{\mathbf{r}_i + \boldsymbol{\tau}_{\text{solar}}}{|\mathbf{r}_i + \boldsymbol{\tau}_{\text{solar}}|}. \quad (3)$$

In addition to the heliostat alignment, each individual facet can also be aligned, which is called canting. While the alignment of the heliostat changes depending on the position of the sun, the alignment of each facet is static. Besides no aligning, where all facets remain flat, there are two methods of aligning the facets.

The first one is called on-axis canting. It considers the case where the sun, receiver and heliostat are all on the same axis. The facets are then adjusted in a paraboloid centered in the center point \mathbf{p}_i , such that each facet's focus point is on the receiver.

The second one, called off-axis canting, works the same as on-axis canting but considers the sun not to be on a common axis, resulting in an off-centered parabolic shape of the facets.

Additionally, the optical model considers flat and curved facets. Curved facets have a parabolic shape, and their focal length is either fixed to a certain value from a range of focal lengths, or it is set to a focal length that comes closest to the ideal focal length $f_i = |\mathbf{p}_{\text{aim},i} - \mathbf{p}_i|$.

Tower The tower is located at the origin of the global coordinate system and has the shape of a cylinder with the radius r_{tower} and the height h_{tower} . These dimensions are relevant for the shadow cast on the heliostat layout. The diameter of the minimal bounding circuit is given by $d_{\text{tower}} = 2 \cdot r_{\text{tower}}$. Each heliostat's bounding sphere must not intersect with the bounding circuit of the tower.

Receiver The receiver is mounted on the tower and heats the heat transferring fluid in its interior using the solar radiation reflected by the heliostats.

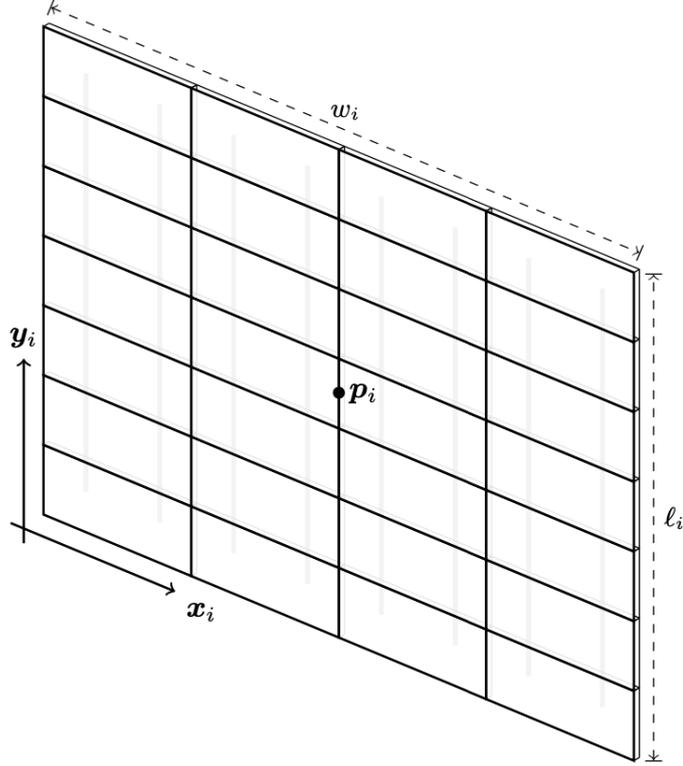


Figure 3: Heliostat with four horizontal facets and seven vertical facets. The horizontal axis x_i , vertical axis y_i , length l_i , width w_i and center point p_i are indicated [17].

There are several receiver concepts, the most common being the tilted flat receiver, cylindric cavity receiver and cylindric external receiver as illustrated in Figure 5. Further information about these receiver concepts can be found in [31, 17].

It is also possible to have a multi-receiver system where heliostats target a different receiver depending on their position in the field. However, his type of receiver is not considered in this work.

For each receiver concept the aiming points $p_{\text{aim},i}$ for the heliostats are calculated differently. On a tilted flat receiver the target point $p_{\text{aim},i}$ is the center of the receiver for all heliostats. For cylindrical cavity receivers, $p_{\text{aim},i}$ is the point on the receiver which is furthest away from the heliostat H_i . In case of cylindrical external receivers $p_{\text{aim},i}$ is calculated as the closest point in the center of the aperture to the heliostat H_i .

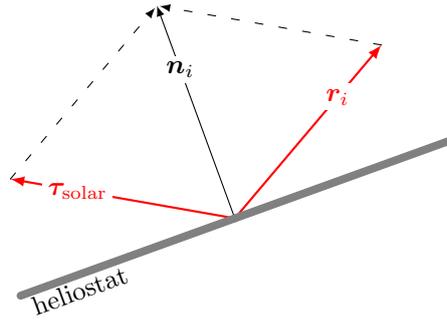
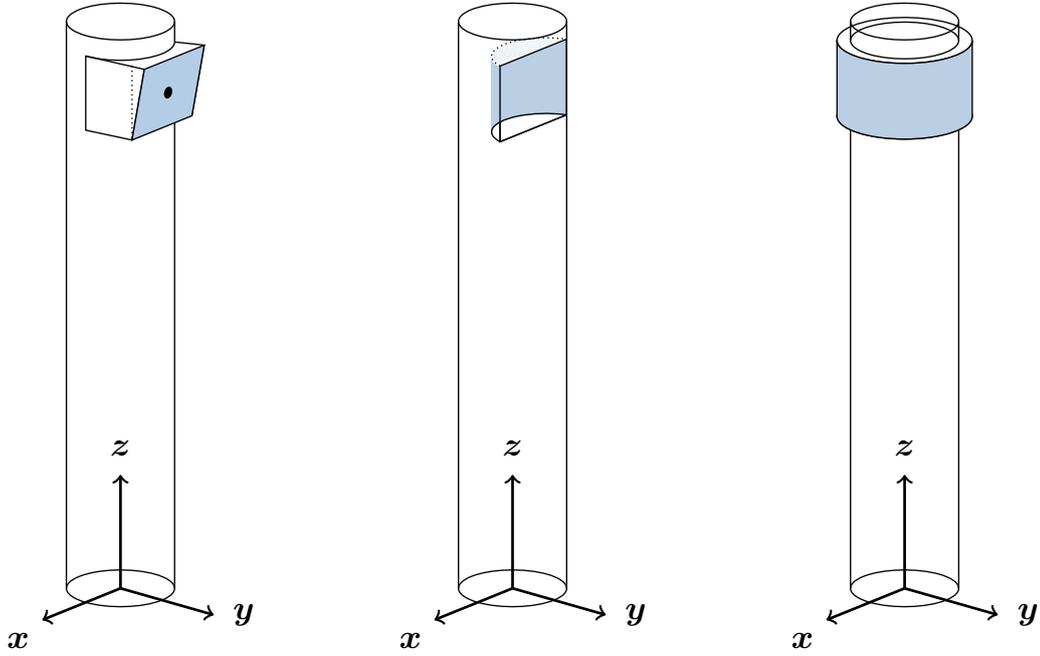


Figure 4: Reflection of a solar ray τ_{solar} on heliostat H_i with normal vector \mathbf{n}_i such that the reflection \mathbf{r}_i aims at the aiming point $\mathbf{p}_{\text{aim},i}$ on the receiver [17].



- (a) The aiming point $\mathbf{p}_{\text{aim},i}$ for every heliostat H_i is the center, which is marked with a black dot. (b) The aiming point $\mathbf{p}_{\text{aim},i}$ for every heliostat H_i is the point on the receiver which is furthest away. (c) The aiming point $\mathbf{p}_{\text{aim},i}$ is the closest point in the center of the aperture to the heliostat H_i .

Figure 5: Illustration of the three different types of receivers and towers: tilted flat receiver (5a), cylindric cavity receiver (5b) and cylindric external receiver (5c). Figures are derived from [31].

Raytracer When tracking the rays from the sun via the heliostats to the receiver, various effects and losses are considered which negatively influence the power transmission. The model considers the following effects:

- The **cosine effect** η_{\cos} accounts for the reduced mirror area due to the tilted alignment of the heliostat as seen in Figure 4. The cosine effect depends on the position of the sun, the position of the receiver and the position of the heliostat. The effect increases with a more acute angle of incidence of the solar rays $\boldsymbol{\tau}_{\text{solar}}$. It is calculated as

$$\eta_{\cos,i} = \langle \boldsymbol{\tau}_{\text{solar}}, \mathbf{n}_i \rangle, \quad (4)$$

where \mathbf{n}_i is the surface normal of heliostat H_i , calculated according to Equation 3.

- **Shading and blocking effects** η_{sb} occur when the solar rays are blocked by the tower, adjacent heliostats or hilly terrain on their way from the sun via the heliostat to the receiver. The raytracer needs to check every traced ray for such collisions with the environment, so it uses optimized clipping methods to speed up the computing time. However, since the raytracer processes a large number of rays, this part of the simulation is the most time-consuming.
- **Heliostat reflectivity** η_{ref} accounts for the radiation lost at the mirror surface due to surface impurities and the energy absorbency of the surface material. The model assumes η_{ref} to be a constant value.
- **Atmospheric attenuation efficiency** η_{aa} considers the power loss of light traveling through the atmosphere. This loss depends on the travel distance $d_i = |\mathbf{p}_i - \mathbf{p}_{\text{aim},i}|$ between center point \mathbf{p}_i of the heliostat H_i and the aiming point on the receiver $\mathbf{p}_{\text{aim},i}$. The Mirval transmissivity model [21] is used, which calculates $\eta_{\text{aa},i}$ as

$$\eta_{\text{aa},i} = \begin{cases} 0.99321 - 1.176 \cdot 10^{-4} \cdot d_i + 1.97 \cdot 10^{-8} d_i^2 & d_i \leq 1000m \\ \exp(-1.106 \cdot 10^{-4} \cdot d_i) & d_i > 1000m \end{cases}. \quad (5)$$

- **Optical errors** combine three effects which can be described by means of Gaussian distributions with standard derivation of σ in [mrad]. First off is the sun error σ_{sun} , which occurs because the sun is modeled as a plane and not a sphere. The second error is called tracking error σ_{tracking} and is caused by the precision of the motors that align the heliostat. The third error, called slope error σ_{slope} takes into account the imperfection of the mirror surface, which leads to an imperfect reflection. All three errors combined result in a beam error that can be computed as

$$\sigma_{\text{beam}} = \sqrt{\sigma_{\text{sun}}^2 + \sigma_{\text{tracking}}^2 + \sigma_{\text{slope}}^2}. \quad (6)$$

Because of these errors, a solar ray does not hit the aiming point with perfect accuracy, but rather with a Gaussian distributed error cone. The resulting losses are considered in the interception efficiency, also called spillage losses η_{sp} .

Optical power The optical model calculates the received power by solar radiation at the receiver for each heliostat H_i with mirror surface area A_i and direct normal irradiation $I_{\text{DNI}}(t)$ at time t as follows

$$P_i(t) = A_i \cdot I_{\text{DNI}}(t) \cdot \eta_{\text{cos},i}(t) \cdot \eta_{\text{sb},i}(t) \cdot \eta_{\text{ref},i}(t) \cdot \eta_{\text{aa},i}(t) \cdot \eta_{\text{spl},i}(t), \quad (7)$$

while taking the cosine effects $\eta_{\text{cos},i}(t)$, shading and blocking losses $\eta_{\text{sb},i}(t)$, heliostat reflectivity $\eta_{\text{ref},i}(t)$, atmospheric attenuation losses $\eta_{\text{aa},i}(t)$ and spillage losses $\eta_{\text{spl},i}(t)$ into account. The combined received power from all N_{hel} heliostats at time t is defined by

$$P(t) = \sum_{i=1}^{N_{\text{hel}}} P_i(t). \quad (8)$$

2.2. Thermal model

Inside of the receiver radiant power which is collected by the heliostats on the receiver gets converted into thermal power by heating up a heat transfer fluid (HTF). The receiver consists of several plates with a set of tubes through which the HTF flows. The HTF used in the this model is a molten salt. The thermal model considers three types of losses that affects the thermal power of the HTF \dot{Q}_{HTF} . First there are the reflection losses \dot{Q}_{ref} , which are caused by the fact that every material reflects some radiation. Secondly, the temperature of the receiver causes radiation to be emitted into the environment, because it is above absolute zero. These losses are called radiation losses \dot{Q}_{rad} . At last, the convection losses \dot{Q}_{conv} denote the heat dissipation losses of the HTF. The thermal power of the HTF \dot{Q}_{HTF} is defined as the difference between the incident radiation \dot{Q}_{inc} and the above mentioned losses as

$$\dot{Q}_{\text{HTF}} = \dot{Q}_{\text{inc}} - (\dot{Q}_{\text{ref}} + \dot{Q}_{\text{rad}} + \dot{Q}_{\text{conv}}), \quad (9)$$

where the incident radiation \dot{Q}_{inc} is obtained from the optical model. Detailed information on the thermal model can be found in Heiming [19] and Franke [17].

2.3. Storage model

The storage system consists of a hot storage tank and a cold storage tank. Both store the HTF, in this case a molten salt, to avoid an additional heat exchanging step, which would reduce the efficiency. HTF that comes from the receiver tubes fills the hot storage and increases the amount of stored energy. The HTF can be stored over a long period of time with low losses.

When needed, the hot storage is discharged and the HTF enters a heat exchanger, which is connected to the thermal-hydraulic model. The cooled HTF is then stored in the cooling tank to be reused later in the receiver.

For the storage tank model it is sufficient to model only the hot storage tank, since the cold storage tank does not store thermal energy. The power flows in the system are illustrated in Figure 6. The thermal power coming from the solar block $P_{\text{th}}^{\text{sb}}(t)$ is split up in the power going directly into the power block $P_{\text{th}}^{\text{sb,pb}}(t)$, the power charging the storage $P_{\text{th}}^{\text{sb,st}}(t)$ and the excess power $P_{\text{th}}^{\text{excess}}(t)$.

The excess power node is needed in this case to model situations in which the thermal input power surpasses the capacity of the power and storage block.

$P_{\text{th}}^{\text{pb}}(t)$ denotes the thermal input power of the power block and it is the sum of the power coming directly from the receiver $P_{\text{th}}^{\text{sb,pb}}(t)$ and the power drawn from the hot storage $P_{\text{th}}^{\text{st,pb}}(t)$.

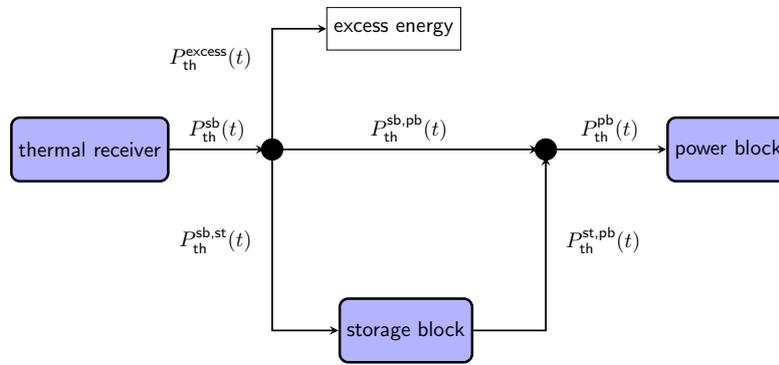


Figure 6: Overview on the power flows in between the thermal receiver, power conversion unit and the storage system. Power is flowing from the thermal receiver either directly into the power block or is fed into the storage. If neither is possible, it is modeled as excess energy. Furthermore, the storage block can be discharged to feed the power block. Figure derived from Coumbassa [15].

To maximize power block efficiency, in each time step a simple buffer strategy maximizing the power input is applied to the power flows in the storage system. The strategy is derived from Coumbassa [15]. The buffer strategy consists of two operation modes and works as follows:

- *Generate mode* [$P_{\text{th}}^{\text{sb}} \leq P_{\text{th}}^{\text{pb,max.in}}$]: The thermal power from the receiver does not exceed the maximal input power of the turbine. As a result, there is no power directed to the storage and no excess power, i.e.

$$P_{\text{th}}^{\text{sb,st}} := 0, \quad (10)$$

$$P_{\text{th}}^{\text{excess}} := 0, \quad (11)$$

because all the available thermal power is directed to the power block

$$P_{\text{th}}^{\text{sb,pb}} := P_{\text{th}}^{\text{sb}}. \quad (12)$$

Furthermore, any additional power that can be processed by the power block is drawn from the storage

$$P_{\text{th}}^{\text{st,pb}} := \min \left(P_{\text{th}}^{\text{pb,max,in}} - P_{\text{th}}^{\text{sb,pb}}, P_{\text{th}}^{\text{st,max,out}} \right). \quad (13)$$

If, however, the power flow to the power block does not exceed its minimum input power ($P_{\text{th}}^{\text{pb}} < P_{\text{th}}^{\text{pb,min,in}}$), the entire thermal power is instead fed into the storage and not into the power block

$$P_{\text{th}}^{\text{sb,pb}} := 0, \quad (14)$$

$$P_{\text{th}}^{\text{st,pb}} := 0, \quad (15)$$

$$P_{\text{th}}^{\text{sb,st}} := \min \left(P_{\text{th}}^{\text{sb}}, P_{\text{th}}^{\text{st,max,in}} \right), \quad (16)$$

$$P_{\text{th}}^{\text{excess}} := \max \left(0, P_{\text{th}}^{\text{sb}} - P_{\text{th}}^{\text{sb,st}} \right). \quad (17)$$

- *Surplus mode* [$P_{\text{th}}^{\text{sb}} > P_{\text{th}}^{\text{pb,max,in}}$]: The thermal power from the receiver exceeds the maximum input power of the turbine. The power directed from the receiver to the power block is limited by its maximum input power

$$P_{\text{th}}^{\text{sb,pb}} := P_{\text{th}}^{\text{pb,max,in}}, \quad (18)$$

As much excess power as possible is stored

$$P_{\text{th}}^{\text{sb,st}} := \min \left(P_{\text{th}}^{\text{sb}} - P_{\text{th}}^{\text{sb,pb}}, P_{\text{th}}^{\text{st,max,in}} \right), \quad (19)$$

Remaining power is discarded and no thermal power is drawn from the storage

$$P_{\text{th}}^{\text{excess}} := \max \left(0, P_{\text{th}}^{\text{sb}} - P_{\text{th}}^{\text{sb,pb}} - P_{\text{th}}^{\text{sb,st}} \right), \quad (20)$$

$$P_{\text{th}}^{\text{st,pb}} := 0. \quad (21)$$

$P_{\text{th}}^{\text{st,max,in}}$ and $P_{\text{th}}^{\text{st,max,out}}$ denote the maximum charging and discharging power of the storage system and $P_{\text{th}}^{\text{pb,max,in}}$ and $P_{\text{th}}^{\text{pb,min,in}}$ denote the maximum and minimum input thermal power of the power block. The thermal energy stored in the hot storage is denoted by $Q_{\text{th}}^{\text{st}}$ and after applying the buffer strategy at time step t it is updated as follows

$$Q_{\text{th}}^{\text{st}}(t + \Delta t) = Q_{\text{th}}^{\text{st}}(t) + (\eta_{\text{st,in}} P_{\text{th}}^{\text{sb,st}}(t) - \eta_{\text{st,out}}^{-1} P_{\text{th}}^{\text{st,pb}}(t) - P_{\text{th}}^{\text{loss}}(t)) \cdot \Delta t, \quad (22)$$

where $\eta_{\text{st_in}}$ is the charging efficiency and $\eta_{\text{st_out}}$ is the discharging efficiency of the storage system. Additionally, the lost thermal power depends on the amount of stored energy and the heat loss factor $\xi_{\text{st_loss}}$

$$P_{\text{th}}^{\text{loss}}(t) = Q_{\text{th}}^{\text{st}}(t) \cdot \xi_{\text{st_loss}}. \quad (23)$$

As the storage capacity is limited by the maximum capacity $Q_{\text{th}}^{\text{st_max}}$, at any time t it must hold, that

$$Q_{\text{th}}^{\text{st}}(t) \leq Q_{\text{th}}^{\text{st_max}}. \quad (24)$$

Finally, the total thermal power which is sent to the power block is given by the sum

$$P_{\text{th}}^{\text{pb}}(t) = P_{\text{th}}^{\text{sb,pb}} + P_{\text{th}}^{\text{st,pb}}. \quad (25)$$

2.4. Thermal-hydraulic model

The thermal-hydraulic model describes the conversion of thermal energy into electrical energy in the power block of the CSP plant. The power block consists of a heat exchanger, a steam turbine, a generator and a cooling system. The molten salt comes either directly from the receiver or from the thermal storage and enters the heat exchanger where the thermal power $P_{\text{th}}^{\text{pb}}$ of the HTF causes the water to evaporate. The steam then drives the turbine, which generates electricity, and eventually condenses back into water. Instead of modeling each component individually, the thermal-hydraulic model from Franke [17] uses a look-up table to determine the efficiency of the power block η_{pb} as a function of ambient temperature T_{ambient} and turbine load ℓ . Therefore, the electrical power output $P_{\text{el}}^{\text{pb}}$ at time t is given by

$$P_{\text{el}}^{\text{pb}}(t) = \eta_{\text{pb}}(T_{\text{ambient}}, \ell) \cdot P_{\text{th}}^{\text{pb}}(t), \quad (26)$$

where η_{pb} may look similar to the power block efficiency shown in Figure 7. When calculating η_{pb} for a given time, a bilinear interpolation is applied to the temperature and load levels to fill in missing data.

An important indicator is the annual energy production (AEP) defined as

$$E_{\text{AEP}} = \sum_{d=1}^{365} \int_0^{24} P_{\text{el}}^{\text{pb}}(t) dt. \quad (27)$$

Since the exact calculation of E_{AEP} is a time-consuming task, quadrature methods are applied to approximate E_{AEP} , which are described in detail in Franke [17].

2.5. Economic model

The economic model evaluates the performance and profitability of a CSP plant. Three different cost factors have to be considered and hence modeled. First of all, at the beginning of the lifetime of a CSP plant it has to be planned and built. The one-time investment costs for the plant, called C_{invest} , are made up of various costs, e.g. for

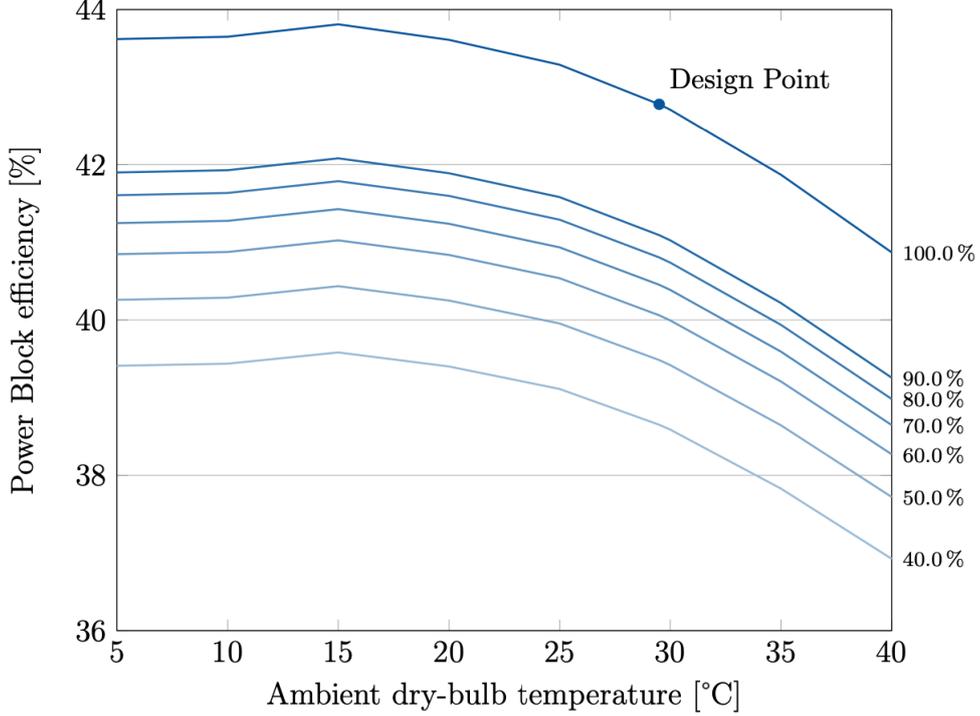


Figure 7: Diagram of temperature-dependent power block efficiencies for different turbine load levels are shown as lines for a $100 \text{ MW}_{\text{th}}$ power conversion unit from TSK Flagsol. A higher turbine load and a lower ambient temperature result in a higher efficiency of the power conversion. The power block is designed to run at full load at around 30° [19].

land, materials and labor. Running the plant then incurs annual costs for operation and maintenance, which are denoted by $C_{\text{O\&M}}$. The annual income from the production and sale of electricity is referred to as C_{revenue} . These costs are modeled as follows:

- C_{invest} sums up various one-time investment costs as suggested by Augsburger [9] and Heimig [19],

$$C_{\text{invest}} = C_{\text{invest}}^{\text{land}} + C_{\text{invest}}^{\text{hel}} + C_{\text{invest}}^{\text{tower}} + C_{\text{invest}}^{\text{rec}} + C_{\text{invest}}^{\text{st}} + C_{\text{invest}}^{\text{pb}}, \quad (28)$$

where each sub-cost is described as follows.

- $C_{\text{invest}}^{\text{land}}$ is the cost of purchasing and improving the terrain for the plant site.
- $C_{\text{invest}}^{\text{hel}}$ is the investment cost of the heliostat layout. Including material, labor, manufacturing, management and tooling costs. It depends on the number of heliostats and the unit cost as well as the cost of cabling the heliostats,

$$C_{\text{invest}}^{\text{hel}} = c_{\text{invest}}^{\text{hel}} \cdot N_{\text{hel}} + C_{\text{cable}}. \quad (29)$$

Where C_{cable} are the costs for cable laying, calculated from the length of the minimum spanning tree connecting all heliostats and the cable costs per meter,

$$C_{\text{cable}} = c_{\text{invest}}^{\text{cable}} \cdot \ell_{\text{MST}}. \quad (30)$$

- $C_{\text{invest}}^{\text{tower}}$ is the cost of the tower.
- $C_{\text{invest}}^{\text{rec}}$ is the cost of the receivers, which depends on the unit cost and the number of receivers,

$$C_{\text{invest}}^{\text{rec}} = c_{\text{invest}}^{\text{rec}} \cdot N_{\text{rec}}. \quad (31)$$

- $C_{\text{invest}}^{\text{st}}$ is the cost of the storage system, which depends on the storage capacity and the cost per storage unit,

$$C_{\text{invest}}^{\text{st}} = c_{\text{invest}}^{\text{st}} \cdot Q_{\text{th}}^{\text{st-max}}. \quad (32)$$

- $C_{\text{invest}}^{\text{pb}}$ is the cost of the power block, considering its installed capacity,

$$C_{\text{invest}}^{\text{pb}} = c_{\text{invest}}^{\text{pb}} \cdot P_{\text{el}}^{\text{pb-max-out}}. \quad (33)$$

- The operating and maintenance costs $C_{\text{O\&M}}$ are defined as a fraction $\xi_{\text{O\&M}}$ of the investment costs

$$C_{\text{O\&M}} = \xi_{\text{O\&M}} \cdot C_{\text{invest}}. \quad (34)$$

- The annual revenue (AR) C_{revenue} depends on the annual energy production and the hourly electricity tariff π_{tariff}

$$C_{\text{revenue}} = \sum_{d=1}^{365} \int_0^{24} \pi_{\text{tariff}}(t) \cdot P_{\text{el}}^{\text{pb}}(t) dt. \quad (35)$$

With C_{invest} , $C_{\text{O\&M}}$ and C_{revenue} calculated as described, the economic model considers the following economic values, where N_{lifetime} is the project duration in years and r_{rate} is the nominal interest rate for the investment loan.

- The **Levelized Cost of Energy (LCOE)** is a measure of the production cost per MWh. It is a common value used to compare power plants and is defined as

$$\pi_{\text{LCOE}} = \frac{C_{\text{invest}} \cdot f_{\text{annuity}} + C_{\text{O\&M}}}{E_{\text{AEP}}}, \quad (36)$$

where the annuity factor f_{annuity} is defined as

$$f_{\text{annuity}} = \frac{(1 + r_{\text{rate}})^{N_{\text{lifetime}}} \cdot r_{\text{rate}}}{(1 + r_{\text{rate}})^{N_{\text{lifetime}}} - 1}. \quad (37)$$

- The **Net Present Value (NPV)** measures the total profit over the lifetime of the project. Hence it should be positive. It is defined as the difference between the present values of all incomes and expenses over the project lifetime considering the interest rate r_{rate}

$$C_{\text{NPV}} = \sum_{\ell=1}^{N_{\text{lifetime}}} \frac{C_{\text{revenue}} - C_{\text{O\&M}}}{(1 + r_{\text{rate}})^\ell} - C_{\text{invest}}. \quad (38)$$

- The **Internal Rate of Return (IRR)** is a measure of the profitability of the project. It is the interest rate r_{irr} with which the NPV equation (38) reaches zero

$$\sum_{\ell=1}^{N_{\text{lifetime}}} \frac{C_{\text{revenue}} - C_{\text{O\&M}}}{(1 + r_{\text{irr}})^\ell} - C_{\text{invest}} \stackrel{!}{=} 0. \quad (39)$$

A Newton iteration is applied to solve for r_{irr} . For the project to be profitable, r_{irr} should be larger than the sum of the interest rate r_{rate} and other risk deficits.

- The **Payback Period (PP)** denotes the number of years N_{payback} after which the plant begins to generate profits and is calculated as

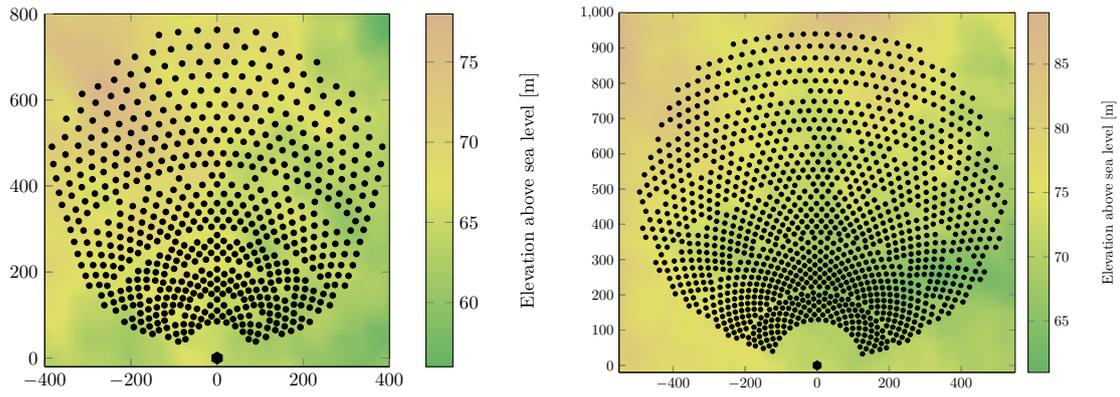
$$N_{\text{payback}} = \frac{\log\left(\frac{C_{\text{revenue}} - C_{\text{O\&M}}}{C_{\text{revenue}} - C_{\text{O\&M}} - (C_{\text{invest}} \cdot r_{\text{rate}})}\right)}{\log(1 + r_{\text{rate}})}. \quad (40)$$

2.6. Modeling existing power plants

This work models existing CSP plants, namely Planta Solar 10 (PS10) [5], Planta Solar 20 (PS20) [6] and Gemasolar [2], all located in the Spanish province of Seville. Table 1 summarizes the settings used for each power plant in the model.

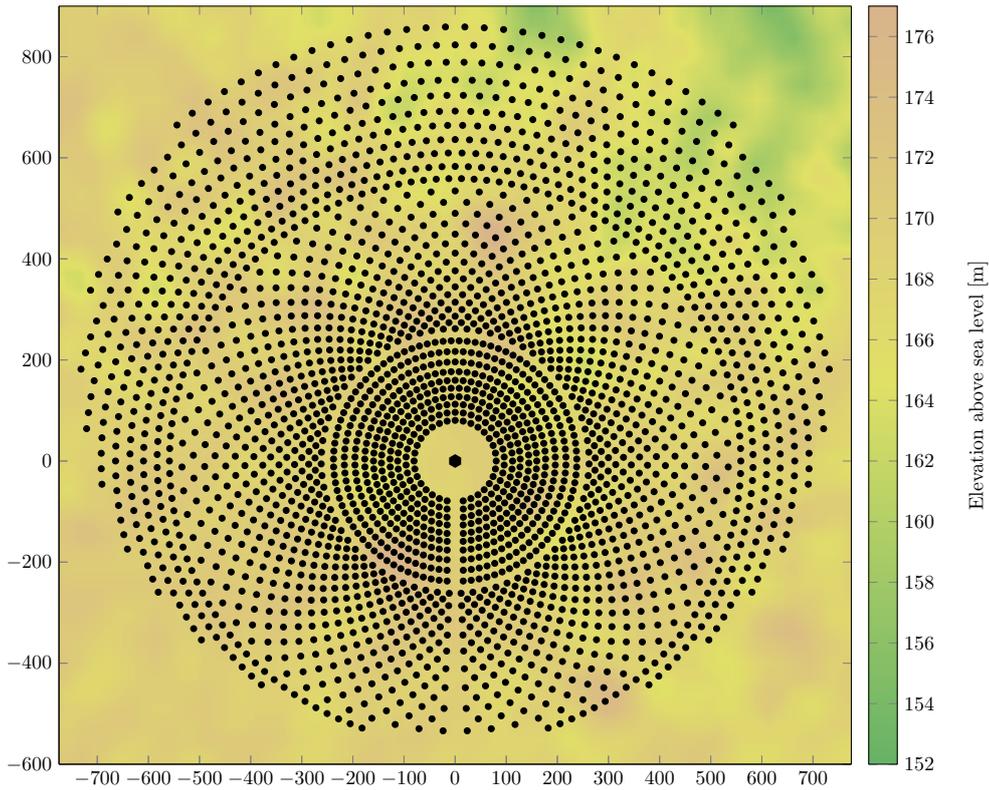
Parameter		PS10	PS20	Gemasolar
Location		37.44°N 6.25°W	37.44°N 6.25°W	37.56°N 5.33°W
Optical model				
Number of heliostats	N_{hel}	624	1254	2650
Heliostat layout		see Figure 8a	see Figure 8b	see Figure 8c
Heliostat type		rectangular	rectangular	rectangular
Pedestal height		5.17 m	5.17 m	5.675 m
Canting		on-axis	on-axis	on-axis
Facet surface		flat	flat	flat
Heliostat mirror area		121.3 m ²	121.3 m ²	115.7 m ²
Tower type		see Figure 5b	see Figure 5b	see Figure 5c
Tower height		120 m	170 m	140 m
Gaussian sun sigma	σ_{sun}	2.35 mrad	2.35 mrad	2.35 mrad
Tracking sigma	σ_{tracking}	1 mrad	1 mrad	1 mrad
Slope sigma	σ_{slope}	1 mrad	1 mrad	1 mrad
Heliostat reflectivity	η_{ref}	88 %	88 %	93 %
Thermal model				
HTF material		Air	Air	Salt
Storage model				
Storage capacity	$Q_{\text{th}}^{\text{st,max}}$	20 MWh	20 MWh	299 MWh
Charging efficiency	$\eta_{\text{st,in}}$	97 %	97 %	97 %
Discharging efficiency	$\eta_{\text{st,out}}$	97 %	97 %	97 %
Heat loss factor	$\xi_{\text{st,loss}}$	0.05 %/h	0.05 %/h	0.05 %/h
Thermal-hydraulic model				
Power block capacity	$P_{\text{el}}^{\text{pb,max,out}}$	11 MW	20 MW	19.9 MW
Power block efficiency	η_{pb}	see Figure 7	see Figure 7	see Figure 7
Economic model				
Project lifetime	N_{lifetime}	30 years	30 years	30 years
Interest rate	r_{rate}	4.5 %	4.5 %	4.5 %
Energy tariff	π_{tariff}	271.188 €/MWh	271.188 €/MWh	271.188 €/MWh
Operation cost factor	$\xi_{\text{O\&M}}$	6 %	6 %	6 %
Land cost	C_{land}	1 M€	3 M€	8 M€
Heliostat cost	C_{hel}	25 k€/unit	25 k€/unit	25 k€/unit
Heliostat cabling cost	C_{cable}	30 €/m	30 €/m	30 €/m
Tower cost	C_{tower}	2 M€	5 M€	15 M€
Receiver cost	C_{sb}	5 M€/unit	8 M€/unit	30 M€/unit
Storage cost	C_{st}	100 k€/MWh	100 k€/MWh	100 k€/MWh
Power block cost	C_{pb}	1 M€/MW	1 M€/MW	1 M€/MW
Total investment cost	C_{invest}	≈ 37 M€	≈ 70 M€	≈ 171 M€

Table 1: Overview on the optical, thermal, storage, electrical and economical model settings for PS10, PS20 and Gemasolar. Note that the economic values shown are estimates, since accurate information on investment costs is usually not available.



(a) Original PS10 heliostat field layout.

(b) Original PS20 heliostat field layout.



(c) Original Gemasolar heliostat field layout.

Figure 8: Original heliostat layouts for the three CSP plants PS10 (8a), PS20 (8b) and Gemasolar (8c). Black dots indicate heliostat positions and the black hexagon indicate the tower positions. The y-axis points north and the x-axis points east with units in meters. The background colors indicate the topography i.e. the elevation above sea level of the plant location.

2.7. Validation

Since this work aims at optimizing the heliostat layout, which is part of the optical model, the optical model must be verified. This section validates the optical model, implemented in **SunFlower**, against the existing Monte Carlo raytracing softwares **SolTrace** [34] and **Tonatiuh** [11].

First, various simple test cases with one or two heliostats were performed to validate individual functionalities of the raytracer. Subsequently, simulations of entire CSP plants with hundreds of heliostats were validated.

Each test case was executed multiple times for each tool to obtain an average simulation result for the optical power at the receiver. This is necessary because the ray tracing algorithms use the Monte Carlo method, which is based on randomness, thus repeating the same simulation can lead to slightly different results.

2.7.1. Basic test cases

The following basic test cases are based on the **base** test case, whose settings are summarized in Table 2. It considers a setup of a single heliostat with a single flat facet which is placed 100 meters away from the tower. On the tower, which has a height and a diameter of 1.2m, a flat receiver of the same size is mounted. The receiver, the heliostat and the position of the sun are arranged in such a way that the heliostat reflects all sunlight coming from the south parallel to the ground back into the receiver. Optical errors, atmospheric attenuation and tower shading are disabled in this case.

- **base** There are no modifications to the base settings.
- **reflectivity** Reduces the heliostat reflectivity to 80%.
- **small receiver** Reduces the receiver area to 0.25 m².
- **multiple facets** Replaces the single heliostat facet with four 0.25 m² facets.
- **curved** Use an ideal focused facet whose focus point is on the receiver. Also uses the on-axis canting method and the small 0.25 m² receiver.
- **curved focal length** Same setup as the **curved** test but instead of the ideal focus, it uses a fixed focal length of 180 m, which is behind the receiver.
- **atmospheric attenuation** Enabled atmospheric attenuation effect. Note that **SolTrace** does not support this functionality.
- **polygon** Instead of a rectangular heliostat, a flat pentagonal shaped heliostat with a side length of 2 m and a larger receiver are used.
- **sun sigma** Gaussian sun sigma set to 2.35 mrad.
- **tracking error** Tracking error set to 1 mrad.

Parameter	Value
Sun azimuth	180°
Sun altitude	0°
DNI	1000 W/m ²
Raytracer	
Reflectivity	100%
Atmospheric attenuation	off
Tower shading	off
Gaussian sun sigma	0 mrad
Tracking sigma	0 mrad
Slope sigma	0 mrad
Heliostats	
Positions	(0, 100)
Cluster pattern	single
Pedestal height	0.6 m
Canting	none
Facet surface form	flat
Heliostat shape	single facet
Facet width	1 m
Facet height	1 m
Horizontal facets count	1
Vertical facets count	1
Tower	
Tower height	1.2 m
Tower diameter	1.2 m
Height above receiver	0.0 m
Receiver	
Type	flat
Count	1
Tilt angle	0.0°
Orientation angle	0.0°
Horizontal panels	1
Panel width	1.2 m
Panel height	1.2 m
Horizontal pieces per panel	10
Vertical pieces per panel	10

Table 2: Settings for the base test case.

- **slope error** Slope error set to 1 mrad.
- **tracking slope error** Slope and tracking error set to 1 mrad.
- **combined sigma errors** Slope and tracking error set to 1 mrad and Gaussian sun sigma set to 2.35 mrad.
- **none flat** A single PS10 heliostat (see Table 1) with a tilted flat receiver on a 95 m tall tower without canting.
- **on axis flat** Same as **none flat** but with on-axis canting.
- **off axis angles flat** Same as **none flat** but with off-axis canting.

- **on axis curved** Same as **on axis flat** but with curved facets.
- **off axis angles curved** Same as **off axis angles flat** but with curved facets.
- **none flat polygon** Same as **none flat** but with a pentagon shaped heliostat with a side length of 2 m.
- **on axis flat polygon** Same as **none flat polygon** but with on-axis canting.
- **on axis curved polygon** Same as **none flat polygon** but with curved facets. Note that Tonatiuh does not support curved triangles.
- **blocking** One heliostat is at (0,20) and a second one is behind it at (0,40). Both heliostats are identical to the heliostats from PS10 (see Table 1). The receiver is placed on a 15m tower. The sun's position is set so that the light is incident at an angle of 90° . This configuration tests if the first heliostat blocks the light reflected by the second heliostat.
- **shading** One heliostat is at (0,80) and a second one is behind it at (0,100). Both heliostats are identical to the heliostats from PS10 (see Table 1). The receiver is placed on a 95m tower and is tilted downwards slightly by 11.5° . The sun's position is set so that the light is incident from the south at an angle of 0° , i.e. parallel to the ground. This configuration tests if the first heliostat casts a shadow on the second heliostat.
- **multiple receivers** Instead of a single receiver, three flat receivers are mounted on a 115 m high tower, which are facing in separate directions. One heliostat is at (-100,50) and a one second is at (100,50), each aiming at a different receiver. Both heliostats are identical to the heliostats from PS10 (see Table 1). The sun's position is set so that the light is incident from the south with an angle of 80° .
- **external** Instead of a flat receiver, an external receiver is mounted on a 95 m tall tower. The PS10 heliostat is placed at (0,100). The sun is positioned at an altitude of 90° so that it does not shine directly on the receiver. This is important because Tonatiuh, unlike SolTrace and SunFlower, takes into account the direct radiation onto the receiver.
- **full shadow** The PS10 heliostat is placed at (0,50) and the sun is positioned in the south with an altitude of 45° , such that the heliostat is completely shaded by the 95m tower. Since SolTrace does not support the tower shading functionality, only Tonatiuh is considered for this test case.
- **partial shadow** The PS10 heliostat is placed at (10,50) and the sun is positioned in the south with an altitude of 45° , so that the heliostat is only partially shaded by the 95m tower. Since SolTrace does not support the tower shading functionality, only Tonatiuh is considered for this test case.

- **cavity raise** Instead of a flat receiver, a cavity receiver is mounted on a 115 m tall tower. The PS10 heliostat is placed at (15,100) and the sun is positioned in the south with an altitude of 0° , so that the heliostat is not shaded by the tower. Since **SolTrace** does not support the tower shading functionality, only **Tonatiuh** is considered for this test case.

Test summary Test results are shown in Table 3. Despite the large number of rays traced, there are still noticeable deviations.

This is caused by the Monte Carlo method and the different implementation in **SunFlower**, compared to **Tonatiuh** and **SolTrace**. In general, Monte Carlo raytracers need a very large number of traced rays to achieve a high accuracy. **Tonatiuh** and **SolTrace** are forward raytracer tools, that trace every ray through the entire geometry in multiple stages starting from the light source. **SunFlower** on the other hand, traces every ray starting from the facet surface on the heliostats. That is why the number of rays for **SunFlower** is measured per square meter of heliostat surface area, while **Tonatiuh** and **SolTrace** receive a total number of rays. Therefore, for **Tonatiuh** and **SolTrace** the number of rays must be raised with increasing geometry size to maintain accuracy.

For example, the simplest test case **base** shows an absolute average deviation of $2.6 \times 10^{-10}\%$, $1.3 \times 10^{-4}\%$ and $7.5 \times 10^{-3}\%$ for **SunFlower**, **Tonatiuh** and **SolTrace**, respectively. Figure 9 illustrates the simulation accuracy of all tools for the **base** test. It can be seen that the simulation accuracy of **SunFlower** is accurate, even at a lower number of rays per square meter, while the simulation results of **Tonatiuh** and **SolTrace** start to converge when more than 1×10^4 rays are traced.

Thus, it is assumed that with an increasing number of rays, even larger than the one chosen for these tests, an even smaller deviation between the tools could be achieved. Overall, for each test case in Table 3, **SunFlower**, **Tonatiuh** and **SolTrace** agree on the simulation result with little deviations and a low variance. This indicates that they would converge to the same result if even more rays were traced.

Further testing of the Gaussian sigma has revealed slight inaccuracies in the implementation of **SolTrace** and **Tonatiuh**. For instance, when increasing the sigma errors, **Tonatiuh** and **SolTrace** increasingly deviate from the theoretical result, as shown in Figure 10. They behave similarly when the distance of the heliostat to the receiver is increased. The simulation results of **SunFlower** for these tests on the other hand show little deviations to the theoretical results, which are caused by the randomness of the Monte Carlo method. The sigma error implementation of **SunFlower** is thus considered to be correct.

Ultimately, the test cases show the correctness of the tested functionalities of the optical model of **SunFlower**.

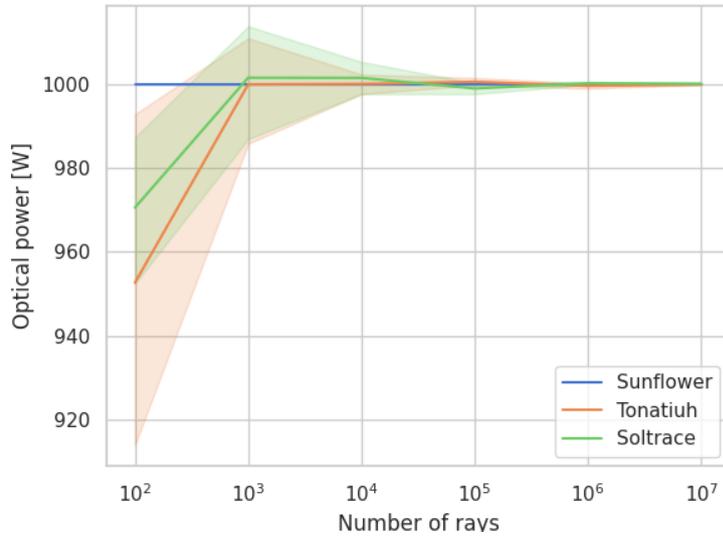


Figure 9: Simulation results versus number of traced rays for the **base** test case. Since **SunFlower** traces the given number of rays per square meter of facet area, its simulation results are accurate even for a small number of rays. **SolTrace** and **Tonatiuh** trace the total number of rays through the entire geometry starting at the light source, resulting in inaccuracies with a smaller number of rays.

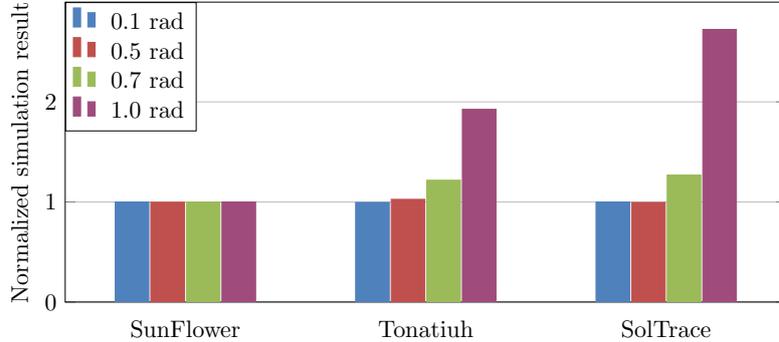


Figure 10: Test cases revealing an inaccuracy in the implementation of the Gaussian error sigmas in **Tonatiuh** and **SolTrace**. For different sigmas, a single heliostat was simulated in front of a single flat receiver and the simulation results were normalized by the theoretical result. **SunFlower** show little deviation to the theoretical result, while both **Tonatiuh** and **SolTrace** deviate with increasing sigmas.

		SunFlower		Tonatiuh		SolTrace	
# rays		$1 \times 10^6 \text{ m}^{-2}$		$1 \times 10^8 \text{ total}$		$1 \times 10^7 \text{ total}$	
# simulations		10		10		100	
Test	Reference [W]	deviation [%]	variance [%]	deviation [%]	variance [%]	deviation [%]	variance [%]
base	1000	2.6×10^{-10}	1.4×10^{-29}	1.3×10^{-4}	2.6×10^{-6}	7.5×10^{-3}	5.7×10^{-5}
reflectivity	800	2.2×10^{-11}	2.9×10^{-30}	2.6×10^{-3}	4.1×10^{-6}	1.1×10^{-2}	5.9×10^{-5}
small receiver	250	1.5×10^{-4}	7.5×10^{-31}	6.9×10^{-3}	7.6×10^{-6}	3.7×10^{-3}	7.0×10^{-5}
multiple facets	1000	1.6×10^{-13}	0	2.7×10^{-3}	4.1×10^{-6}	6.1×10^{-3}	6.1×10^{-5}
curved	1000	3.7×10^{-4}	0	2.5×10^{-4}	9.9×10^{-7}	5.8×10^{-3}	5.4×10^{-5}
curved focal length	1000	2.2×10^{-4}	3.0×10^{-30}	4.1×10^{-4}	9.1×10^{-7}	2.1×10^{-3}	5.8×10^{-5}
atmospheric attenuation	982	4.4×10^{-2}	6.0×10^{-30}	4.6×10^{-2}	1.9×10^{-5}	-	-
polygon	27525	9.6×10^{-3}	0	1.1×10^{-2}	3.2×10^{-4}	1.4×10^{-3}	10.0×10^{-4}
sun sigma	804	3.2×10^{-2}	2.4×10^{-5}	3.0×10^{-2}	8.3×10^{-6}	3.0×10^{-2}	6.8×10^{-5}
tracking error	968	4.9×10^{-2}	2.5×10^{-6}	5.0×10^{-2}	5.8×10^{-6}	5.1×10^{-2}	4.5×10^{-5}
slope error	967	5.8×10^{-2}	3.1×10^{-6}	5.4×10^{-2}	5.6×10^{-6}	4.9×10^{-2}	7.0×10^{-5}
tracking slope errors	923	2.9×10^{-2}	9.3×10^{-6}	2.7×10^{-2}	7.1×10^{-6}	2.9×10^{-2}	5.3×10^{-5}
combined sigma errors	753	5.5×10^{-2}	1.3×10^{-5}	5.6×10^{-2}	9.2×10^{-6}	5.2×10^{-2}	5.7×10^{-5}
none flat	119878	2.5×10^{-3}	0	1.6×10^{-3}	1.9×10^{-3}	1.7×10^{-4}	3.5×10^{-3}
on axis flat	121308	2.1×10^{-3}	0	2.8×10^{-3}	3.0×10^{-4}	4.6×10^{-3}	1.5×10^{-3}
off axis angles flat	118007	4.1×10^{-3}	0	1.3×10^{-3}	3.3×10^{-4}	1.7×10^{-3}	3.0×10^{-3}
on axis curved	121315	7.4×10^{-4}	0	2.7×10^{-3}	3.4×10^{-4}	2.6×10^{-3}	1.9×10^{-3}
off axis angles curved	118100	3.0×10^{-3}	0	7.8×10^{-4}	1.5×10^{-4}	1.0×10^{-3}	1.9×10^{-3}
none flat polygon	27221	1.8×10^{-3}	0	1.1×10^{-3}	9.5×10^{-4}	1.3×10^{-3}	7.4×10^{-4}
on axis flat polygon	27519	1.5×10^{-3}	0	1.5×10^{-3}	7.7×10^{-4}	3.0×10^{-4}	7.5×10^{-4}
on axis curved polygon	27515	1.6×10^{-2}	0	-	-	7.0×10^{-4}	8.5×10^{-4}
blocking	114460	1.4×10^{-2}	1.9×10^{-19}	2.2×10^{-2}	8.2×10^{-3}	3.2×10^{-3}	9.3×10^{-3}
shading	113925	6.9×10^{-3}	6.1×10^{-23}	1.2×10^{-2}	6.3×10^{-3}	1.3×10^{-3}	4.6×10^{-3}
multiple receivers	226637	2.0×10^{-3}	0	5.5×10^{-3}	10.0×10^{-3}	9.0×10^{-4}	1.6×10^{-2}
full shadow	0	0	0	0	0	-	-
partial shadow	80393	3.8×10^{-3}	0	3.5×10^{-3}	1.2×10^{-3}	-	-
cavity raise	91854	5.3×10^{-3}	0	5.2×10^{-3}	4.0×10^{-2}	-	-
external	72074	1.1×10^{-2}	0	1.1×10^{-2}	1.8×10^{-3}	2.5×10^{-3}	4.9×10^{-3}

Table 3: Simulation results of SunFlower, Tonatiuh and SolTrace for the basic test cases. Using the reference values, simulation results were normalized and absolute average deviations and variances were calculated. Since Tonatiuh does not support curved triangles and SolTrace does not support the atmospheric attenuation effect and tower shading, results for **on axis curved polygon** as well as **atmospheric attenuation**, **full shadow**, **partial shadow** and **cavity raise** are omitted.

2.7.2. Ray resolution study

Since the dimensions of existing CSP plants are much larger compared to the simple test cases, a fast but accurate optical simulation for **SunFlower** is necessary, especially for the optimization later in this work.

The most important factor influencing the runtime and accuracy of the Monte Carlo raytracer is the number of rays traced. Increasing the number of rays traced increases the accuracy of the simulation result, since the Monte Carlo method is based on the law of large numbers. However, increasing the number of rays traced increases the computation time, which limits the optimization algorithms that require a large number of simulation results.

Therefore it is of interest to find out how many traced rays are needed to guarantee a simulation accuracy of 99.9%. This is an important compromise between low runtime and high accuracy.

In this test, the optimal number of rays traced per square meter for **SunFlower** is evaluated for each presented CSP plant.

For this purpose, each CSP plant was simulated multiple times at three different days of the year. These days are March 21st, June 21st and December 21st.

On each of these days, thirteen evenly spaced simulations were run between 8:00 and 19:00. This test setup covers a large part of the possible sun positions, as December 21st is the shortest and June 21st is the longest day of the year, thus covering the lowest and highest sun altitude in a year. Additionally, March 21st covers sun positions between the lowest and highest altitude.

Furthermore, realistic DNI data [1] from the region of Seville in Spain was interpolated and used in the simulations.

Figure 11 illustrates the 36 different solar positions for which each CSP plant was simulated.

The whole test setup was then run for various numbers of rays per square meter ranging from one to 265. PS10 was run with up to 265 rays per square meter, PS20 with up to 128 rays per square meter and Gemasolar with up to 64 rays per square meter.

Since the resulting optical power seems to converge at an increasing number of traced rays, it is assumed that the simulation results with the highest number of traced rays per square meter are the most accurate. Thus every simulation result is normalized by the average result of the same simulation using the highest number of traced rays.

Test summary Figure 12 shows the result of the ray resolution study. It can be seen that for every CSP plant, the simulation accuracy of **SunFlower** increases as the number of rays per square meter increases.

For the largest CSP Gemasolar however, the simulation accuracy converges faster compared to PS10 and PS20. This effect is caused by the law of large numbers, which causes inaccuracies in the raytracing simulation to cancel each other out, which increases the simulation accuracy even for a lower number of rays per square meter.

Overall, **SunFlower** achieves a simulation accuracy of 99.9% or higher for all three CSP systems with only five rays per square meter of heliostat facet area. For the

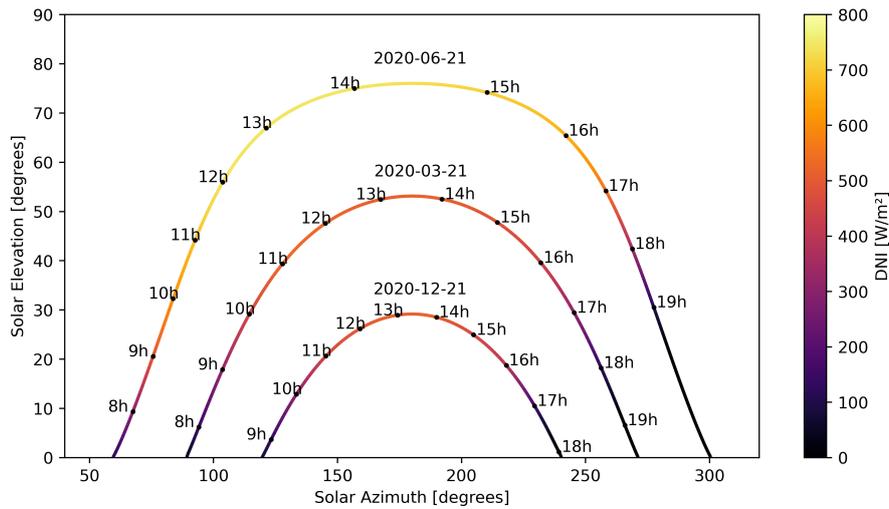


Figure 11: Hourly solar positions on December 21st (bottom), March 21st (middle) and June 21st (top). The DNI at each solar position is indicated by the line color. The sun positions are calculated for the same location in Seville from which the DNI data was measured i.e. 37.41667 (latitude), -5.9 (longitude).

optimization part of this work, the raytracer of **SunFlower** is therefore configured with five rays per square meter of facet area.

2.7.3. Existing power plants

Only Tonatiuh was used to validate the simulation results of **SunFlower** for entire CSP plants, since SolTrace does not support the important raytracing features for tower shading and atmospheric attenuation. However, the Gaussian sigma errors for slope, tracking and sun shape were turned off due to their inaccurate implementation in Tonatiuh.

PS10, PS20 and Gemasolar is tested on March 21, June 21 and December 21 between 9:00 and 19:00 with the respective DNI, illustrated by Figure 11. These tests cover a wide range of possible solar positions over the course of a year.

For each test case the optical power on the receiver of the CSP was simulated with the optical model settings from Table 1. The resulting optical power was then normalized by the average simulation result of **SunFlower** to obtain a percentage deviation and variance.

Test summary The simulation results are summarized in Table 4, Table 5 and Table 6 for PS10, PS20 and Gemasolar.

For every test, the mean deviation of **SunFlower** is 0% as the reference values are the mean simulation results of **SunFlower**. Tonatiuh shows a deviation of less than 0.1% for

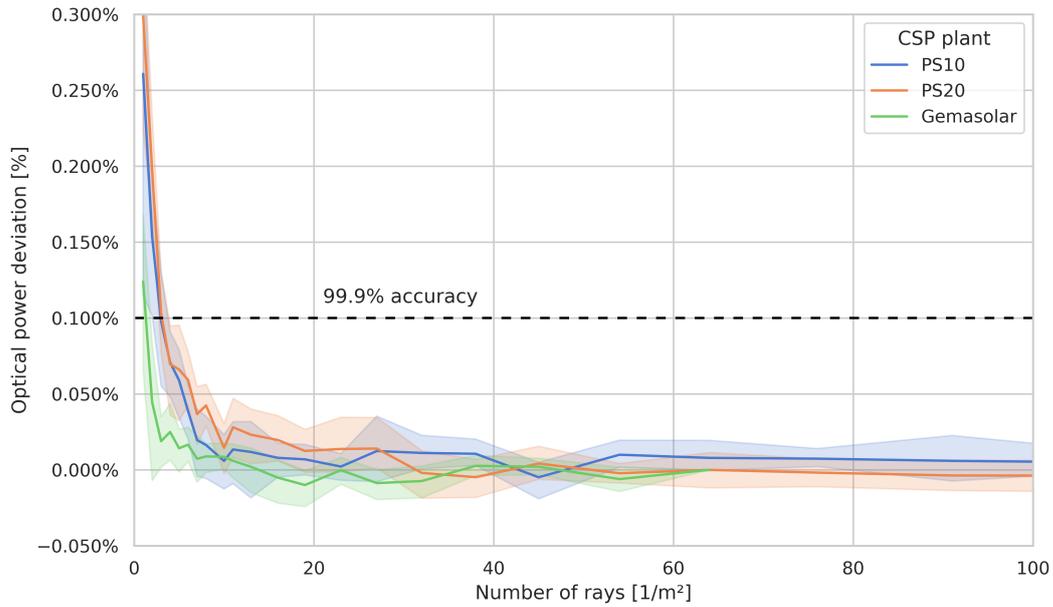


Figure 12: Results of the ray resolution study, which measured the number of traced rays per square meter against the simulation accuracy for each CSP plant. Deviations from the mean value are indicated by the colored area around the darker tinted line that represents the mean value. The dashed horizontal line indicates a simulation accuracy of 99.9%, which is achievable with five rays per square meter. The simulation accuracy of the largest CSP plant Gemasolar converges faster compared to PS10 and PS20 due to the law of large numbers, since with an increasing plant size inaccuracies in the ray tracing cancel each other out.

PS10 and PS20, as well as less than 0.2% for Gemasolar. Since **Tonatiuh** was run only once per test case, there is no variance value. Also, both tools agree on sun positions with a DNI of zero.

Since for each test case the deviation of **Tonatiuh** is close to zero, it is assumed that with an increased number of rays the simulation results of both tools would converge.

PS10		SunFlower		Tonatiuh	
# rays		1.51×10^7 total		5×10^7 total	
# simulations		10		1	
Test	Reference [W]	deviation [%]	variance [%]	deviation [%]	variance [%]
21. Mar. 09h	$\approx 1.40 \times 10^7$	0	9.72×10^{-19}	3.18×10^{-2}	-
21. Mar. 11h	$\approx 2.83 \times 10^7$	0	3.80×10^{-18}	6.91×10^{-2}	-
21. Mar. 13h	$\approx 2.93 \times 10^7$	0	4.72×10^{-18}	2.27×10^{-2}	-
21. Mar. 15h	$\approx 2.69 \times 10^7$	0	1.38×10^{-18}	6.72×10^{-2}	-
21. Mar. 17h	$\approx 1.52 \times 10^7$	0	7.71×10^{-19}	9.93×10^{-2}	-
21. Mar. 19h	0	0	0	0	-
21. Jun. 09h	$\approx 1.87 \times 10^7$	0	9.18×10^{-20}	1.08×10^{-2}	-
21. Jun. 11h	$\approx 3.26 \times 10^7$	0	2.36×10^{-18}	1.90×10^{-3}	-
21. Jun. 13h	$\approx 3.86 \times 10^7$	0	2.95×10^{-18}	1.12×10^{-2}	-
21. Jun. 15h	$\approx 3.73 \times 10^7$	0	2.88×10^{-18}	5.80×10^{-3}	-
21. Jun. 17h	$\approx 2.73 \times 10^7$	0	2.15×10^{-18}	2.12×10^{-2}	-
21. Jun. 19h	$\approx 4.36 \times 10^6$	0	3.09×10^{-19}	2.46×10^{-2}	-
21. Dec. 09h	$\approx 2.86 \times 10^6$	0	3.12×10^{-19}	9.47×10^{-2}	-
21. Dec. 11h	$\approx 2.38 \times 10^7$	0	1.21×10^{-18}	1.71×10^{-2}	-
21. Dec. 13h	$\approx 2.97 \times 10^7$	0	6.03×10^{-18}	3.68×10^{-2}	-
21. Dec. 15h	$\approx 2.54 \times 10^7$	0	2.13×10^{-18}	4.28×10^{-2}	-
21. Dec. 17h	$\approx 6.32 \times 10^6$	0	3.42×10^{-20}	3.79×10^{-2}	-
21. Dec. 19h	0	0	0	0	-

Table 4: Simulation results of SunFlower and Tonatiuh for PS10 for 18 different solar positions and DNI values, which are illustrated in Figure 11. Simulation results are normalized by the average simulation result of SunFlower. Since only a single simulation of Tonatiuh was performed, there is no variance value.

PS20		SunFlower		Tonatiuh	
# rays		3.04×10^7 total		3×10^7 total	
# simulations		10		1	
Test	Reference [W]	deviation [%]	variance [%]	deviation [%]	variance [%]
21. Mar. 09h	$\approx 2.80 \times 10^7$	0	7.87×10^{-19}	5.81×10^{-2}	-
21. Mar. 11h	$\approx 5.69 \times 10^7$	0	4.10×10^{-18}	7.18×10^{-2}	-
21. Mar. 13h	$\approx 5.90 \times 10^7$	0	3.33×10^{-18}	3.66×10^{-2}	-
21. Mar. 15h	$\approx 5.43 \times 10^7$	0	2.34×10^{-18}	3.80×10^{-2}	-
21. Mar. 17h	$\approx 3.05 \times 10^7$	0	1.04×10^{-18}	6.85×10^{-3}	-
21. Mar. 19h	0	0	0	0	-
21. Jun. 09h	$\approx 3.77 \times 10^7$	0	6.03×10^{-19}	1.12×10^{-2}	-
21. Jun. 11h	$\approx 6.57 \times 10^7$	0	4.13×10^{-19}	3.55×10^{-2}	-
21. Jun. 13h	$\approx 7.78 \times 10^7$	0	9.25×10^{-19}	3.68×10^{-2}	-
21. Jun. 15h	$\approx 7.54 \times 10^7$	0	2.33×10^{-18}	9.53×10^{-3}	-
21. Jun. 17h	$\approx 5.53 \times 10^7$	0	6.08×10^{-19}	3.17×10^{-2}	-
21. Jun. 19h	$\approx 8.84 \times 10^6$	0	4.34×10^{-19}	4.45×10^{-2}	-
21. Dec. 09h	$\approx 6.05 \times 10^6$	0	2.33×10^{-20}	1.39×10^{-3}	-
21. Dec. 11h	$\approx 4.67 \times 10^7$	0	2.65×10^{-18}	6.89×10^{-2}	-
21. Dec. 13h	$\approx 5.90 \times 10^7$	0	3.54×10^{-18}	2.82×10^{-2}	-
21. Dec. 15h	$\approx 4.99 \times 10^7$	0	6.19×10^{-19}	9.15×10^{-2}	-
21. Dec. 17h	$\approx 1.11 \times 10^7$	0	6.60×10^{-20}	4.38×10^{-3}	-
21. Dec. 19h	0	0	0	0	-

Table 5: Simulation results of SunFlower and Tonatiuh for PS20 for 18 different solar positions and DNI values, which are illustrated in Figure 11.

Gemasolar		SunFlower		Tonatiuh	
# rays		1.53×10^7 total		1×10^7 total	
# simulations		10		1	
Test	Reference [W]	deviation [%]	variance [%]	deviation [%]	variance [%]
21. Mar. 09h	$\approx 4.98 \times 10^7$	0	6.88×10^{-21}	3.79×10^{-2}	-
21. Mar. 11h	$\approx 9.95 \times 10^7$	0	4.67×10^{-21}	1.27×10^{-1}	-
21. Mar. 13h	$\approx 1.03 \times 10^8$	0	1.27×10^{-20}	3.57×10^{-2}	-
21. Mar. 15h	$\approx 9.44 \times 10^7$	0	2.96×10^{-21}	1.04×10^{-2}	-
21. Mar. 17h	$\approx 5.30 \times 10^7$	0	2.70×10^{-21}	2.18×10^{-2}	-
21. Mar. 19h	0	0	0	0	-
21. Jun. 09h	$\approx 7.65 \times 10^7$	0	4.79×10^{-21}	1.42×10^{-1}	-
21. Jun. 11h	$\approx 1.29 \times 10^8$	0	2.27×10^{-20}	2.80×10^{-2}	-
21. Jun. 13h	$\approx 1.52 \times 10^8$	0	4.66×10^{-20}	9.74×10^{-2}	-
21. Jun. 15h	$\approx 1.47 \times 10^8$	0	1.02×10^{-20}	1.70×10^{-1}	-
21. Jun. 17h	$\approx 1.07 \times 10^8$	0	2.08×10^{-21}	5.43×10^{-2}	-
21. Jun. 19h	$\approx 1.73 \times 10^7$	0	1.72×10^{-21}	7.35×10^{-3}	-
21. Dec. 09h	$\approx 1.07 \times 10^7$	0	5.50×10^{-22}	4.47×10^{-2}	-
21. Dec. 11h	$\approx 7.42 \times 10^7$	0	2.56×10^{-21}	1.19×10^{-2}	-
21. Dec. 13h	$\approx 9.30 \times 10^7$	0	7.12×10^{-21}	1.36×10^{-1}	-
21. Dec. 15h	$\approx 7.89 \times 10^7$	0	1.07×10^{-20}	5.78×10^{-2}	-
21. Dec. 17h	$\approx 1.89 \times 10^7$	0	1.50×10^{-20}	1.60×10^{-2}	-
21. Dec. 19h	0	0	0	0	-

Table 6: Simulation results of SunFlower and Tonatiuh for Gemasolar for 18 different solar positions and DNI values, which are illustrated in Figure 11.

3. Heliostat layout-optimization

The goal of the heliostat layout-optimization is to find positions for all heliostats in the field, such that an objective function is maximized. This optimization problem is difficult to solve, since the search space is large as there are innumerable possible heliostat positions.

To reduce the search space, a common approach is to utilize patterns for the heliostat layouts. The heliostat layouts of the presented CSP plants are also based on patterns as shown in Figure 8. Such patterns have the advantage that there are only a few parameters describing the whole heliostat layout, which reduces the search space.

The decreased search space also limits the pattern approach, since it is not guaranteed that the optimal solution for the heliostat layout problem can be described as a pattern. However, as with many difficult optimization problems, a good solution calculated in a reasonable time is preferred over the optimal solution, whose calculation can take exponentially longer.

Another optimization approach is local search, a variable neighborhood descent algorithm that looks at only one heliostat at a time and tries to move it to a local maximum of the objective function, which results in an increased performance of the entire heliostat layout. However, this approach also only finds local optima, which can be much worse than the global optimum.

This work combines both the pattern and local search approach in a multi-step optimization pipeline, which may achieve better results than each algorithm on its own. This multi-step approach first optimizes a pattern layout, which is then fed into the local search algorithm, which tries to further improve the solution by moving the heliostats to a local optimum individually.

In the following sections, four patterns and two algorithms for solving the pattern optimization problem are presented, then the local search algorithm is introduced.

3.1. Pattern optimization

The following sections introduce different geometrical patterns, which are used to parameterize the heliostat field layout. Section 3.1.5 presents a simple combinatorial solver and the downhill simplex algorithm, both of which can be used to find optimal parameters for the patterns.

3.1.1. Staggered Cornfield

The staggered cornfield pattern, derived from Lipps and Vant-Hull [24], places the heliostats on a north-south aligned grid in which every odd row is staggered to allow for a close packing while avoiding blocking effects. The linear factors l_x , l_y and the exponential factors s_x , s_y are used to control the spacing between columns and rows such that the heliostat density decreases with increasing distance from the tower. Note that for $s_x = 1$ and $s_y = 1$, the cornfield pattern is regular i.e. the heliostat density is constant. Due to the axis symmetric shape of the pattern, it is only necessary to

generate heliostat positions in the first quadrant. The heliostat position on the i -th row and j -th column with $i, j \in \mathbb{N}_0$ is computed by

$$x_j = \begin{cases} \ell_x \cdot D \cdot 0.5 \cdot (j^{s_x} + (j+1)^{s_x}) & \text{if } i = 1 \pmod{2} \\ \ell_x \cdot D \cdot j^{s_x} & \text{otherwise,} \end{cases} \quad (41)$$

$$y_i = \ell_y \cdot D \cdot i^{s_y}, \quad (42)$$

where D is the heliostat diameter. These positions are then mirrored to the other quadrants, as visualized in Figure 13. The parameters and their ranges that control the shape of the pattern are listed in Table 7. The upper limits were determined empirically, while the lower limits must be equal to one, to ensure that the heliostat density decreases at greater distances from the tower and no heliostat collide.

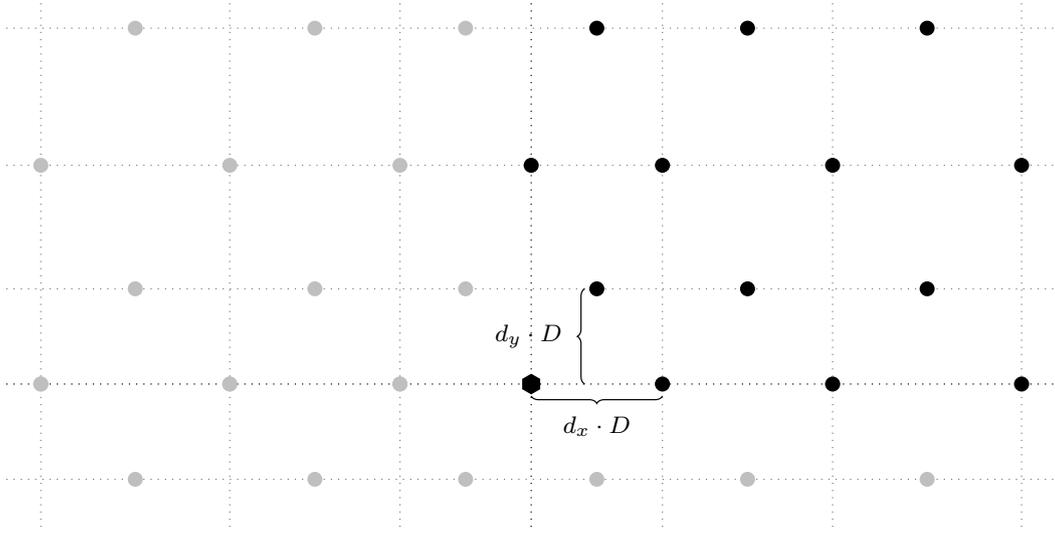


Figure 13: Illustration of the staggered cornfield pattern, which is generated in the first quadrant (black dots) and mirrored to the other quadrants (gray dots). The tower is indicated by a hexagon, ℓ_x and ℓ_y are linear scaling factors and D is the heliostat diameter.

Parameter	Range	Description
ℓ_x	[1, 1.5]	x-axis linear factor [D]
ℓ_y	[1, 1.5]	y-axis linear factor [D]
s_x	[1, 1.5]	x-axis exponential factor
s_y	[1, 1.5]	y-axis exponential factor

Table 7: Parameters of the staggered cornfield pattern. To avoid heliostat collisions ℓ_x and ℓ_y must not be less than one. Furthermore, to allow for a decreased heliostat density further away from the tower, s_x and s_y are greater or equal to one. The upper limits were empirically determined.

3.1.2. Hexagon

The hexagon pattern, based on a patent by Pham et al. [29], places the heliostats on the edges of concentric hexagons around the tower. Each hexagon has at least one more heliostat per edge than the previous one, which automatically results in a staggered layout. The density of the layout is controlled by the density factor d , which influences the distance between each heliostat ℓ_{sep} as

$$\ell_{\text{sep}} = d \cdot D, \quad (43)$$

where D is the heliostat diameter.

Furthermore, d controls the distance between adjacent hexagons. The side length ℓ_i for the i -th hexagon is computed inductively as

$$\ell_i = \begin{cases} \ell_{\text{sep}} & i = 1 \\ \ell_{i-1} + \max\left(\ell_{\text{sep}}, \frac{d\ell_{i-1} \cdot h_{\text{hel}}}{h_{\text{tower}} - h_{\text{hel}}}\right) & i > 1, \end{cases} \quad (44)$$

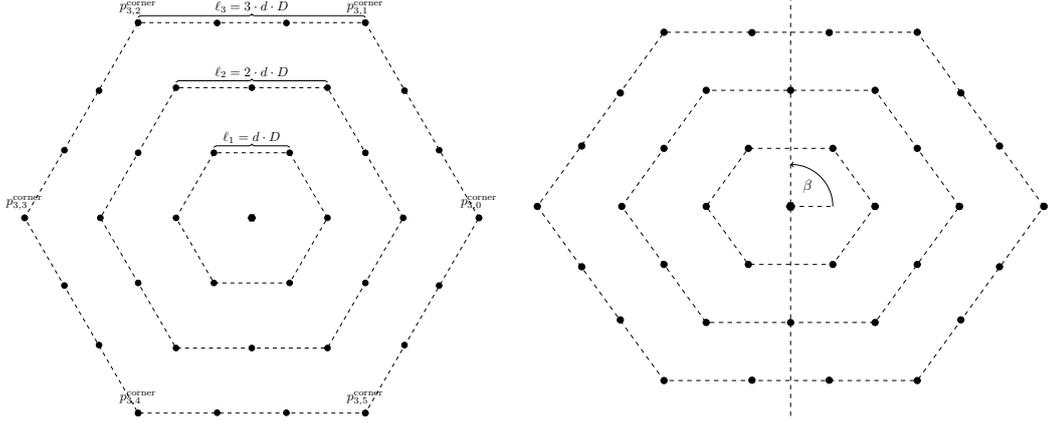
where h_{hel} is the height of the heliostat, which is approximated by $\frac{D}{2}$ and h_{tower} is the height of the tower. For hexagons close to the tower, the distances between hexagons is equal to the distance between heliostats ℓ_{sep} . With increasing distance to the tower, the blocking distance of the heliostats of the previous hexagon $\frac{\ell_{i-1} \cdot h_{\text{hel}}}{h_{\text{tower}} - h_{\text{hel}}}$ is considered, which is computed by the intercept theorem. It is also scaled by the density factor d . To compute the heliostat positions on the edges of the i -th hexagon, first the j -th corner position for $j \in \{0, \dots, 5\}$ is calculated as

$$p_{i,j}^{\text{corner}} = \begin{pmatrix} \ell_i \cdot \cos(j \cdot \frac{\pi}{3}) \\ \ell_i \cdot \sin(j \cdot \frac{\pi}{3}) \end{pmatrix}. \quad (45)$$

Since for the i -th hexagon holds $\ell_i \geq i \cdot \ell_{\text{sep}}$, $\lfloor \frac{\ell_i}{\ell_{\text{sep}}} \rfloor$ heliostats are distributed between each neighboring corner points, as visualized in Figure 14. In addition to the density parameter d , scaling is implemented according to Equation 56 to 58, which is controlled by the angle β and the scaling factor s . All parameters of the hexagon pattern and their feasible value ranges are listed in Table 14. The lower limit for the density factor d ensures that heliostats do not collide, while the upper limit was chosen to be as close to one to keep the layout as dense as possible. The range for s was chosen to be close to one, while the range for β allows for scaling along every possible axis.

3.1.3. Radial staggered

The radial staggered pattern is based on the work of Collado and Guallar [14] and is recognizable in many commercially operated CSP plants. Instead of straight lines, heliostats are positioned on concentric circles around the tower. Similarly to the cornfield pattern, the heliostats on each successive circle are staggered to achieve a close packing while avoiding blocking effects. As illustrated in Figure 15a, the circles are



(a) First three hexagons and their side length ℓ_i . (b) Scaled hexagon pattern, $\beta = 90$ and $s = 0.8$.

Figure 14: Illustration of the first three hexagons of the hexagon pattern (14a) and its scaling (14b) along the y-axis by 80%. The tower is indicated by the filled hexagon.

Parameter	Range	Description
d	[1.0, 2.0]	density factor [D]
s	[0.5, 1.5]	elliptical scaling factor
β	[0.0, 180.0]	angle of scaling axis [$^\circ$]

Table 8: Parameters of the hexagon pattern. To avoid heliostat collisions, d has to be greater than one. The ranges for s were chosen to be close to one, while β has to be between zero and 180 degrees, which allows for scaling along every possible axis.

grouped in different zones Z_i , in which the number of heliostats per circle N_i is equal to avoid heliostat collisions. N_i is calculated by

$$N_i = \left\lfloor \frac{\pi}{\arcsin\left(\frac{\ell_{\text{sep}}}{2r_{i,1}}\right)} \right\rfloor, \quad (46)$$

where $r_{i,j}$ is the radius of the j -th circle within the i -th zone and ℓ_{sep} is the distance between two heliostats on a circle. This safety distance depends on the heliostat diameter D and the density scaling factor d as

$$\ell_{\text{sep}} = d \cdot D. \quad (47)$$

A distinction is made between the radius delta between two circles in the same zone and those between successive zones. The radius delta between $r_{i,j}$ and $r_{i,j+1}$ in the i -th zone is calculated as

$$\Delta r_{i,j}^{\text{circle}} = \max \left(\frac{\sqrt{3}D}{2}, \frac{r_{i,j} \cdot h_{\text{hel}}}{h_{\text{tower}} - h_{\text{hel}}} \right), \quad (48)$$

while the radius delta to the first circle of a potential new zone Z_{i+1} is calculated as

$$\Delta r_{i,j}^{\text{zone}} = \max \left(D, \frac{r_{i,j} \cdot h_{\text{hel}}}{h_{\text{tower}} - h_{\text{hel}}} \right). \quad (49)$$

Within a zone, the densest possible layout is allowed, which is achieved by $\Delta r_{i,j}^{\text{circle}} = \frac{\sqrt{3}D}{2}$ [14]. However, as the number of heliostats between zones changes, staggered heliostat positions are not guaranteed. $\Delta r_{i,j}^{\text{zone}}$ thus has to be at least as large as the heliostat diameter D .

Furthermore, for both radius deltas a simple heuristic is used to avoid blocking effects for heliostats on large circles. This is done by estimating the blocking distance of the heliostats on the current circle by solving the intercept theorem $\frac{r_{i,j} \cdot h_{\text{hel}}}{h_{\text{tower}} - h_{\text{hel}}}$, where h_{hel} is the height of the heliostats, which is estimated by $\frac{D}{2}$.

Based on these rules the pattern is created, illustrated in Figure 15a, starting from the first circle inside the first zones, whose radius is set to the height of the tower

$$r_{1,1} = h_{\text{tower}}. \quad (50)$$

After each circle, it is necessary to decide whether to create a new circle in the same group or a whole new zone. For this decision, the heuristic proposed by Collado and Guallar [14] is used, which creates a new zone if at least twice as many heliostats fit in the initial circle of the potential new zone compared to the current zone. If Equation 51 holds, the new zone Z_{i+1} is created with Equation 52, otherwise a new circle within the current zone Z_i is created with Equation 53.

$$\left\lfloor \frac{\pi}{\arcsin\left(\frac{\ell_{\text{sep}}}{2(r_{i,j} + \Delta r_{i,j}^{\text{circle}})}\right)} \right\rfloor \geq 2 \cdot N_i. \quad (51)$$

$$r_{i+1,1} = r_{i,j} + \Delta r_{i,j}^{\text{zone}} \quad (52)$$

$$r_{i,j+1} = r_{i,j} + g \cdot \Delta r_{i,j}^{\text{circle}}. \quad (53)$$

In Equation 53, the radius delta is scaled by the growth factor g . Furthermore, the pattern is extended with two parameters s and β . These parameters control the elliptical scaling of the circles along an arbitrary axis, as described by Equation 56 to 58 and illustrated in Figure 15b.

All parameters and their value ranges are listed in Table 9. The density factor d controls the density of heliostats on a single circle and has to be close to one, but must not be less than one so that the heliostats are closely packed without colliding. However, the growth factor g , which controls the density of the circles, should be close to one but may be even less than one to allow for an even denser packing. The scaling

factor s should be close to one to allow a slight stretching or shrinking of the overall pattern. Finally β has a range between zero and 180 degrees to allow for scaling along every possible axis.

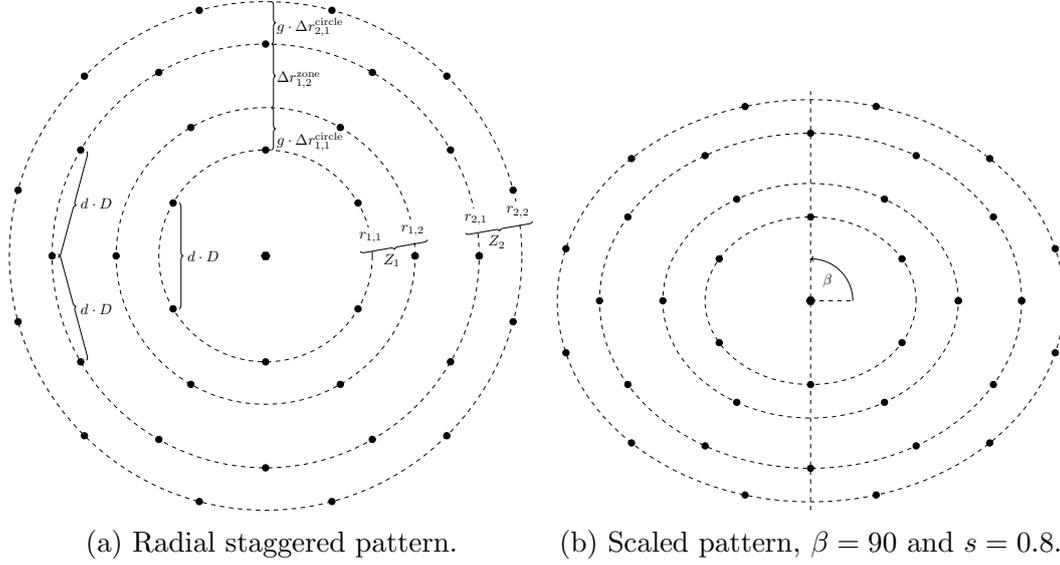


Figure 15: Circles of the first two zones of the radial staggered pattern (15a) and its elliptical scaling (15b) along the y-axis by 80%.

Parameter	Range	Description
d	[1, 1.5]	heliostat density factor [D]
g	[0.5, 1.5]	radius growth factor
s	[0.5, 1.5]	elliptical scaling factor
β	[0.0, 180.0]	angle of scaling axis [$^\circ$]

Table 9: Parameters of the radial staggered pattern. The range for d has to be close to one but not less than one, while the range of g has to be close to one. The limits for s were chosen to be close to one, while a value of β between zero and 180 degrees enables the pattern to be scaled along every possible axis.

3.1.4. Spiral

The spiral pattern is based on the biomimetic pattern of Noone et al. [28]. It is inspired by the sunflower, whose seeds are arranged in a spiral pattern. Its advantage is the continuously decreasing density of heliostats with increasing distance to the tower. This allows for dense packing of efficient heliostats close to the tower and less dense packing further away from the tower, reducing shadowing and blocking effects. The k -th position for this pattern is calculated as polar coordinates (α_k, r_k) . The original spiral pattern has two parameters a and b , which control the growth of the radial component, which is defined as,

$$r_k = ak^b. \quad (54)$$

The angular component is linearly proportional to k and is related to the golden ratio $\frac{1+\sqrt{5}}{2}$,

$$\alpha_k = 2\pi k \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{-2}. \quad (55)$$

Figure 16a illustrates the polar coordinates for the first heliostat in the original spiral pattern.

The original spiral pattern is extended by two new parameters s and β , which control the scaling s of the spiral along an axis defined by the angle β .

To scale the original spiral, first the polar coordinates (α_k, r_k) are converted to Cartesian coordinates (x_k'', y_k'') ,

$$x_k'' = r_k \cdot \cos(\alpha_k) \quad (56)$$

$$y_k'' = r_k \cdot \sin(\alpha_k). \quad (57)$$

The position (x_k'', y_k'') is then rotated clockwise around the origin by β degrees,

$$x_k' = x_k'' \cdot \cos(-\beta) - y_k'' \cdot \sin(-\beta) \quad (58)$$

$$y_k' = x_k'' \cdot \sin(-\beta) + y_k'' \cdot \cos(-\beta). \quad (59)$$

After scaling the x-axis by the factor s , the rotation is reversed by rotating counter-clockwise by β degrees,

$$x_k = s \cdot x_k' \cdot \cos(\beta) - y_k' \cdot \sin(\beta) \quad (60)$$

$$y_k = s \cdot x_k' \cdot \sin(\beta) + y_k' \cdot \cos(\beta). \quad (61)$$

The final heliostat position is (x_k, y_k) . This scaling along an axis results in an elliptical spiral, which is illustrated by Figure 16b. If $s = 1$, the original spiral pattern is obtained. Each parameter and its feasible value range is listed in Table 10. The value ranges for a and b were empirically determined with the combinatorial solver. As β is between zero and 180 degrees, the pattern can be scaled along every possible axis. The limits for the scaling factor s were chosen to be close to one.

3.1.5. Pattern optimization algorithms

Parameterized heliostat layouts are optimized by finding combinations of parameters which maximize the objective function. This section introduces the pattern evaluation process as well as a simple combinatorial search and the more complex downhill simplex algorithm.

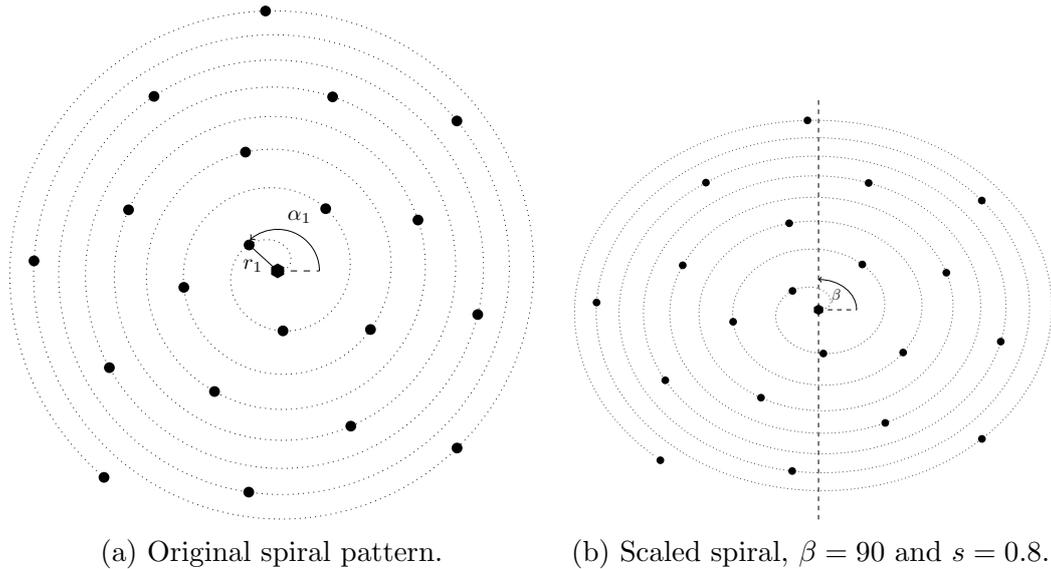


Figure 16: Polar coordinates of the first heliostat in the original spiral pattern (16a) and its elliptical scaling (16b) along the y -axis by 80%.

Parameter	Range	Description
a	[0, 5]	linear density factor
b	[0.5, 1]	exponential density factor
s	[0.5, 1.5]	elliptical scaling factor
β	[0.0, 180.0]	angle of scaling axis [$^\circ$]

Table 10: Parameters of the spiral pattern. The ranges for a and b were empirically determined using the combinatorial solver. The limits for s were chosen to be close to one, while a value of β between zero and 180 degrees enables the pattern to be scaled along every possible axis.

Pattern evaluation The process of evaluating a pattern is split up in different steps. In the first step, a pattern with up to eight times as many heliostats as desired is generated for the given parameters. On one hand, this is necessary for heliostat fields such as PS10 or PS20 that extend only to the north; on the other hand, this is useful for later selecting the most efficient positions from the superset. The next step is to remove invalid heliostat positions, which include colliding heliostats or heliostats outside the field boundaries. If the number of heliostat position is less or equal the number of the desired heliostats afterwards, the layout is evaluated for the given objective function by an annual simulation of the CSP.

Otherwise, the optical efficiencies of the heliostats in the superset is evaluated with an annual simulation, selecting the most efficient ones. After that, a second annual simulation is necessary to compute the score of the given objective function. This method was also used by Noone et al. [28].

Combinatorial search The combinatorial search iterates through combinations of the parameters of the pattern, evaluating the objective function for each and selecting the parameters that maximize the objective function.

For a pattern with N parameters value ranges $p_i^{\min}, p_i^{\max} \in \mathbb{R}$, $p_i^{\min} \leq p_i^{\max}$ and resolutions $r_i \in \mathbb{N}_{>0}$ for each parameter $p_i, i \in \{1, \dots, N\}$ are given. The parameter steps considering the parameter resolution are computed as,

$$p_i^k = p_i^{\min} + \frac{k}{r_i} \cdot (p_i^{\max} - p_i^{\min}). \quad (62)$$

The set containing all parameter combinations is then given as,

$$P = \{(p_1^{k_1}, \dots, p_N^{k_N}) \mid p_i^{k_i} \in \mathbb{R}, 0 \leq k_i < r_i\}. \quad (63)$$

The combinatorial search terminates after each of the parameter combinations in P is evaluated. Thus, the runtime depends on the size of P , which depends on the resolutions of the parameters as,

$$|P| = \prod_{i=1}^N r_i. \quad (64)$$

This algorithm is therefore quite inefficient, as no optimal parameter combination can be found at low resolutions. At higher resolutions on the other hand, the runtime of the combinatorial search grows quickly. Therefore, parameter configurations that produce a pattern with fewer than the desired number of heliostat positions are not evaluated to speed up the algorithm. Additionally, the evaluation of a heliostat layout, which performs an annual simulation of the entire CSP plant, has been parallelized so that multiple parameter combinations can be simulated simultaneously.

Donwhill Simplex The downhill simplex, also known as the Nelder-Mead method [27], is a numerical method originally used to find the minimum of a function. For this application, the algorithm was adapted to search for the maximum of an objective function f .

For a pattern with N parameters, the algorithm operates on $N+1$ points $x_1, \dots, x_{N+1} \in \mathbb{R}^N$ that form a so called simplex, which is the simplest possible polytope in \mathbb{R}^N . For instance, a simplex in two dimensions is a triangle. The algorithm iteratively applies a sequence of operations on the initial simplex to replace the worst of its points, thus moving the entire simplex to a local maximum. The initial simplex is computed by randomly generating a single point $x_{N+1} \in \mathbb{R}^N$ from the pattern's parameter bounds, while the remaining points $x_i, i \in \{1, \dots, N\}$ are equal to x_{N+1} except for the i -th component, which is slightly changed. The algorithm then proceeds as follows (illustrated for $N = 2$):

1. Terminate if the maximum number of iterations i_{\max} is reached. Restart with a new initial simplex if one of two conditions is met. First: if the standard

deviation of the performance of the simplex points is below a certain threshold $\varepsilon > 0$

$$\sqrt{\frac{\sum_{i=1}^{N+1} (f(x_i) - \hat{f})^2}{N}} < \varepsilon, \quad (65)$$

where \hat{f} is the mean performance of of the simplex points

$$\hat{f} = \frac{1}{N} \sum_{i=1}^{N+1} f(x_i). \quad (66)$$

Second: as the results of the simulation vary due to randomness, an additional termination condition is the decreasing size of the simplex. Therefore, the same root-mean-square deviation is computed for each dimension of the simplex points. If the maximum standard deviation is below the threshold ε the iteration is restarted.

2. Order the simplex points according to the performance of the corresponding parameterized heliostat layout considering the objective function f

$$f(x_1) \geq f(x_2) \geq \dots \geq f(x_{N+1}), \quad (67)$$

such that x_{N+1} generates the worst performing pattern.

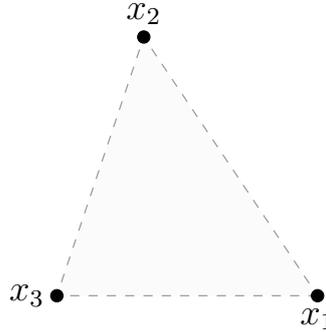


Figure 17: Ordered simplex points.

3. Compute the centroid point x_m , excluding the worst point x_{N+1}

$$x_m = \frac{1}{N} \sum_{i=1}^N x_i. \quad (68)$$

4. Reflect the worst point x_{N+1} on the centroid x_m

$$x_r = x_m + \rho(x_m - x_{N+1}), \quad (69)$$

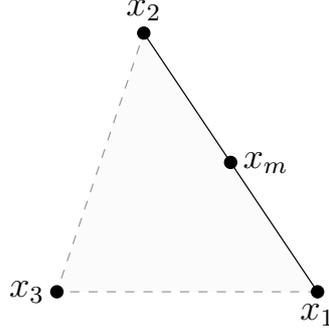


Figure 18: Centroid point x_m .

where $\rho > 0$ is the reflection coefficient. If the reflection point performs better than the best point $f(x_1) < f(x_r)$, go to step 5. Otherwise, if the reflection point still performs better than the second worst point $f(x_1) \geq f(x_r) \geq f(x_N)$, replace the worst point $x_{N+1} = x_r$ and go to step 1. Otherwise, the reflection point performs worse than the second worst point $f(x_r) < f(x_N)$, thus go to step 6.

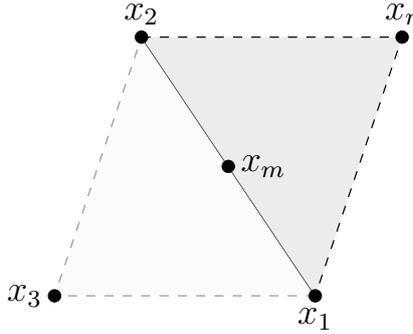


Figure 19: Reflection operation computing x_r .

5. Expand the reflection point

$$x_e = x_m + \gamma(x_r - x_m), \quad (70)$$

where $\gamma > 1$ is the expansion coefficient. Replace the worst point by either the reflection or expansion point and go to step 1.

$$x_{N+1} = \begin{cases} x_e & \text{if } f(x_e) > f(x_r), \\ x_r & \text{otherwise.} \end{cases} \quad (71)$$

6. Contract either x_{N+1} or x_r , depending on which point performs better

$$x_c = \begin{cases} x_m + \beta(x_{N+1} - x_m) & \text{if } f(x_{N+1}) > f(x_r) \\ x_m + \beta(x_r - x_m) & \text{otherwise,} \end{cases} \quad (72)$$

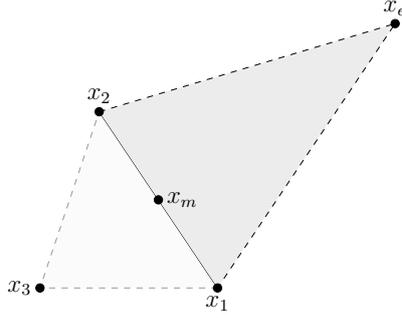


Figure 20: Expansion operation computing x_e .

where $0 < \beta < 1$ is the contraction coefficient. If the contracted point is better or equal to the worst point, replace it with the contracted point $x_{N+1} = x_c$ and go to step 1.

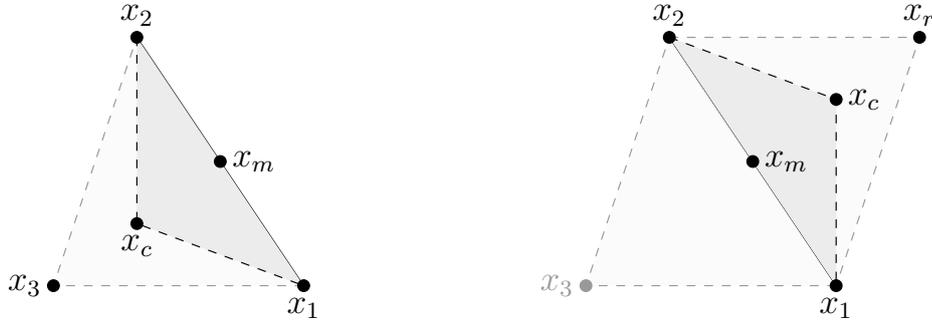


Figure 21: Contraction operation on x_3 (left) and x_r (right), computing x_c .

7. Shrink the entire simplex, while keeping the best point x_1 in place. Thus replace every point

$$x_i = x_1 + \sigma(x_i - x_1), \quad (73)$$

where $0 < \sigma < 1$ is the shrinking coefficient.

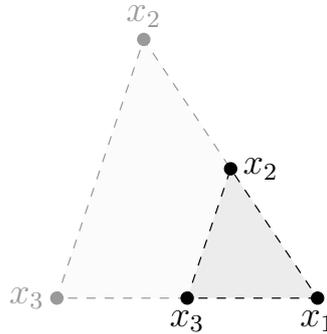


Figure 22: Shrinking operation.

The standard values for the four coefficients listed in Table 11 are used. Although the downhill simplex method does not converge very fast, it is robust and simple to implement as it does not rely on derivatives. To validate the feasibility of the listed ε , PS10 was optimized for AEP with a spiral pattern where $i_{\max} = 100$ was the only termination criterion. As shown in Figure 23 the size of the simplex converges towards zero, meaning that the points of the simplex move closer and closer together, which indicates that the algorithm has found a local maximum. With a convergence threshold of $\varepsilon = 0.01$, as indicated in Figure 23b, the algorithm would restart the current iteration with a new randomly generated simplex if it fell below this threshold.

Parameter	Value	Description
ρ	1	Reflection coefficient
γ	2	Expansion coefficient
β	0.5	Contraction coefficient
σ	0.5	Shrinking coefficient
ε	0.01	Termination threshold
i_{\max}	100	Max. number of iterations

Table 11: Configuration of the downhill simplex method, consisting of the default values for the coefficients and the termination parameters.

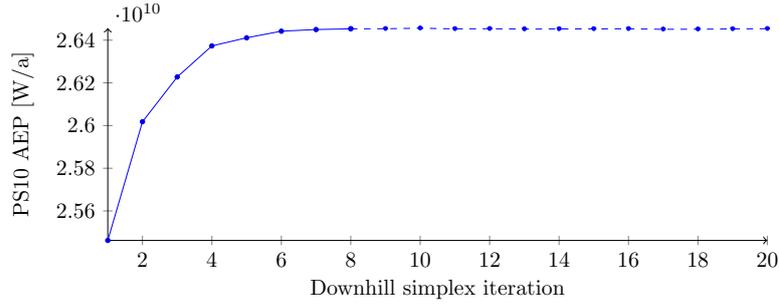
Verification To verify the results of the downhill simplex algorithm, it was run with the spiral pattern on PS10 and the neighborhood of the best found parameter combination was inspected using the combinatorial method. The inspected optimization result is $a = 0.685$, $b = 0.901$, $s = 0.872$ and $\beta = 141.48$. Although the spiral pattern consists of four parameters, only the neighborhood of the solution in the a, b and s, β planes are shown in Figure 24. Nevertheless, it can be seen that the optimization result is indeed a local maximum regarding the AEP, as the patterns close to the solution do not yield better AEP results.

3.2. Local Search optimization

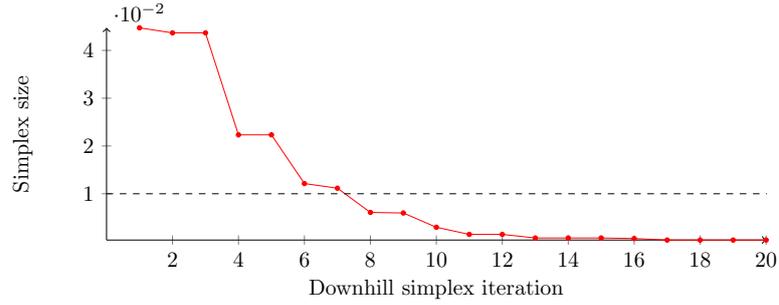
The local search algorithm is the second part of the multi-step heliostat layout optimization pipeline. The algorithm is used to refine the given heliostat layout, which was generated by one of the patterns. While the pattern optimization optimizes every heliostat position at once, the key feature of this algorithm is the local optimization of each heliostat, which takes local features such as neighboring heliostats and local topography into account.

The local search optimization is a variable neighborhood descent algorithm [18]. The algorithm is given an initial solution, a heliostat layout consisting of the two-dimensional positions of each heliostat H_i on the field

$$\mathbf{L}^{\text{init}} = \{H_i \in \mathbb{R}^2 \mid i \in \{1, \dots, N_{\text{hel}}\}\} \quad (74)$$



(a) The dashed part indicates the iterations after which the convergence threshold was undercut.



(b) The dashed grey line indicates the convergence threshold $\varepsilon = 0.01$ after which the simplex iteration would restart.

Figure 23: A single downhill simplex run for the spiral pattern on PS10. For each iteration the AEP improvement (23a) and the simplex size (23b) is shown.

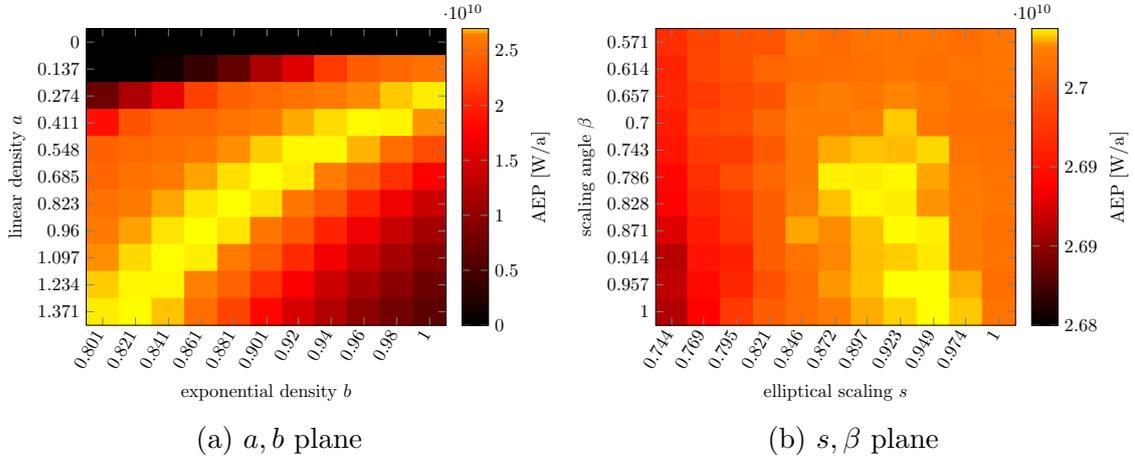


Figure 24: Neighborhood of the downhill simplex solution $a = 0.685, b = 0.901, s = 0.872, \beta = 141.48$ for the spiral pattern for PS10, computed by the combinatorial method. The a, b plane (left) and s, β plane (right) show that there are no close solution with a higher AEP. The centered solution thus depicts a local maximum for the AEP.

and an objective function f , which evaluates a solution and that is to be maximized. The algorithm then iterates over different neighborhoods of the solution and moves to a local maximum in each one with respect to the objective function.

The core idea behind the local search algorithm is that the local maximum for one neighborhood is not necessarily a local maximum for a different neighborhood. Additionally, a global maximum is a local maximum for all possible neighborhood [18], thus iteratively inspecting different neighborhood structures prevents the algorithm from converging to a local maximum.

Neighborhood structure The i -th neighborhood $N_i(\mathbf{L})$ of the heliostat layout \mathbf{L} is defined as the set of layouts where only the position of the i -th heliostat is changed. Instead of repositioning the heliostat on a rectangular grid, as done by Buck [12] and Reinholz et al. [30], the heliostats are moved on a circular grid consisting of n_{circle} equidistant circles centered around the original position, on which there are n_{position} equidistant positions each. Formally, it is defined as

$$N_i(\mathbf{L}) = \{\mathbf{L} \setminus \{H_i\} \cup \{H_i + \text{move}_i^{c,p}\} \mid c \in \{0, \dots, n_{\text{circle}}\}, p \in \{0, \dots, n_{\text{position}} - 1\}\}, \quad (75)$$

where the move term $\text{move}_i^{c,p}$ for the i -th heliostat and the p -th position on the c -th circle is defined as,

$$\text{move}_i^{c,p} = \begin{pmatrix} c \cdot r_i \cdot \cos(2\pi \frac{p}{n_{\text{position}}}) \\ c \cdot r_i \cdot \sin(2\pi \frac{p}{n_{\text{position}}}) \end{pmatrix}. \quad (76)$$

$p = n_{\text{position}}$ is not considered, as $\text{move}_i^{c,0}$ and $\text{move}_i^{c,n_{\text{position}}}$ are equal.

The radius r_i is not only depending on the diameter D of the heliostat but also proportionally on the distance of the heliostat H_i to the tower. This was done similarly for rectangular grids by Buck [12]. This allows heliostats which are far away to perform larger movements, while heliostats closer to the tower, which are typically packed close together, can make small adjustments without colliding with each other. For the heliostat H_i with diameter D and euclidean distance d_i^{tower} to the tower, the radius for the circular grid is defined as

$$r_i = \ell_{\text{circle}} \cdot \frac{d_i^{\text{tower}}}{10^4} \cdot D, \quad (77)$$

where the parameter ℓ_{circle} controls the proportional increase of the radius with increasing distance of the heliostat to the tower. The term is scaled by 10^4 because tests have shown that reasonable values for ℓ_{circle} are between one and ten.

Overall, Table 12 lists the parameters which define the i -th neighborhood $N_i(\mathbf{L})$ of a solution \mathbf{L} . Additionally, Figure 25 shows the 17 possible moves of heliostat H_i on the circular grid for parameters $n_{\text{circle}} = 2$ and $n_{\text{position}} = 8$, which define the i -th neighborhood $N_i(\mathbf{L})$.

Parameter	Description
ℓ_{circle}	Scaling factor of circle radius
n_{circle}	Number of equidistant circles
n_{position}	Number of equidistant positions on a circle

Table 12: Parameters that define the i -th neighborhood $N_i(\mathbf{L})$ of a solution \mathbf{L} .

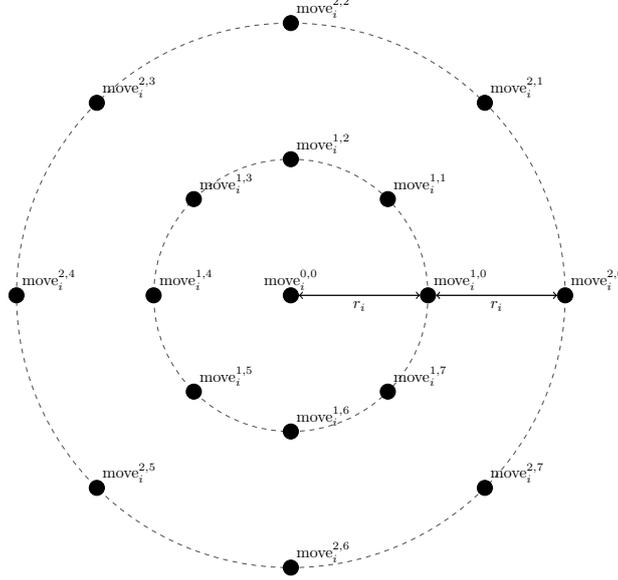


Figure 25: Illustration of the 17 possible moves in circular grid of the neighborhood $N_i(\mathbf{L})$ of heliostat H_i for $n_{\text{circle}} = 2$ and $n_{\text{position}} = 8$. The radius r_i depends on the diameter of H_i and its distance to the tower as defined in Equation 77.

Algorithm With the definition of a neighborhood structure, the local search algorithm, shown in Algorithm 1, is implemented as follows.

It is initialized with a heliostat layout \mathbf{L}^{init} . The algorithm repeatedly iterates over every heliostat. An order of heliostats in ascending efficiency is chosen, starting with the least efficient. For every heliostat H_j it then evaluates every solution in its neighborhood of the current solution $N_j(\mathbf{L}_i)$ in iteration i . Any solution in the neighborhood is discarded if the move of H_j is invalid, e.g. if H_j collides with another heliostat. From the remaining heliostat layouts, the one which maximizes f is selected, i.e. it finds the best possible valid move of the i -th heliostat on the circular grid. The algorithm terminates if the maximum number of iterations i_{max} is reached or the new heliostat layout shows less than ε improvement over the last iteration.

Heliostat subgrouping The local search algorithm needs to evaluate the objective function at least $N_{\text{hel}} \cdot (n_{\text{circle}} \cdot n_{\text{position}} + 1)$ times per iteration.

Even for PS10 with the neighborhood defined by $n_{\text{circle}} = 2$ and $n_{\text{position}} = 8$ as shown in Figure 25, this results in 10608 evaluations of f per iteration.

Algorithm 1 LocalSearch

```
1: function LOCALSEARCH( $\mathbf{L}^{\text{init}}$ )
2:    $\mathbf{L}_0 \leftarrow \mathbf{L}^{\text{init}}$ 
3:    $i \leftarrow 0$ 
4:   repeat
5:      $i \leftarrow i + 1$ 
6:      $\mathbf{L}_i \leftarrow \mathbf{L}_{i-1}$ 
7:     for  $j \in \{1, \dots, N_{\text{hel}}\}$  do
8:        $\mathbf{L}_i \leftarrow \arg \max_{\mathbf{L} \in N_j(\mathbf{L}_i)} f(\mathbf{L})$ 
9:   until  $i \geq i_{\text{max}}$  or  $\frac{f(\mathbf{L}_i) - f(\mathbf{L}_{i-1})}{f(\mathbf{L}_{i-1})} \leq \varepsilon$ 
```

As the objective function f performs an annual simulation of the entire heliostat field, its evaluation time is the most time consuming part of the local search algorithm. As the optical model was already optimized in terms of speed versus accuracy in Section 2.7.2, the local search algorithm itself must be accelerated.

An intuitive approach for optimizing the runtime of the local search is to limit the evaluation of the objective function to a subgroup around the moved heliostat. This is reasonable, since the performance of a heliostat is mainly influenced by shading and blocking of sunlight by its neighboring heliostats and less by heliostats further away. Therefore, a simulation of the entire heliostat layout is not necessary to decide which of the $n_{\text{circle}} \cdot n_{\text{position}} + 1$ moves maximizes f for given a heliostat.

A similar approach was taken by Buck [12], who only considered a subgroup of the eleven nearest heliostats around the one who is being optimized.

Instead of taking a fixed size subgroup around a heliostat, a circular subgroup whose radius depends on the heliostat diameter D is considered. The subgroup set S_j around heliostat H_j , which includes every heliostat that has an euclidean distance of $\ell_{\text{subgroup}} \cdot D$ or less to H_j is defined as

$$S_j = \{H_k \mid |H_j - H_k| \leq \ell_{\text{subgroup}} \cdot D\}. \quad (78)$$

Figure 26 shows the average number of heliostats in the subgroup set for different subgroup scalings. For $\ell_{\text{subgroup}} = 5$, the subgroup sets for PS10, PS20 and Gemasolar contain around 30 heliostats. Restricting the evaluation of the objective function f to the subgroup set S_j when optimizing heliostat H_j , compared to the evaluation of the entire heliostat layout, lead to a massive runtime reduction of the local search algorithm. Algorithm 2 indicates this change over Algorithm 1 in blue. A summary over all parameters which control this algorithm is given in Table 13.

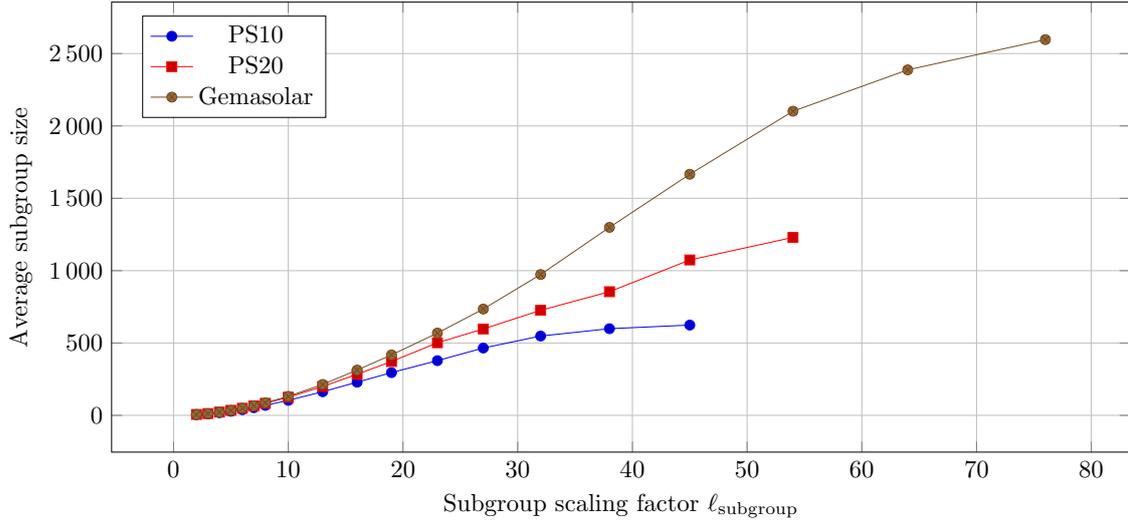


Figure 26: Average subgroup size for PS10, PS20 and Gemasolar for different subgroup scalings ℓ_{subgroup} . With increasing ℓ_{subgroup} the subgroup size approaches the number of heliostats N_{hel} in the entire layout.

Algorithm 2 LocalSearch with Subgrouping

```

1: function LOCALSEARCH( $\mathbf{L}^{\text{init}}$ )
2:    $\mathbf{L}_0 \leftarrow \mathbf{L}^{\text{init}}$ 
3:    $i \leftarrow 0$ 
4:   repeat
5:      $i \leftarrow i + 1$ 
6:      $\mathbf{L}_i \leftarrow \mathbf{L}_{i-1}$ 
7:     for  $j \in \{1, \dots, N_{\text{hel}}\}$  do
8:        $\mathbf{L}_i \leftarrow \arg \max_{\mathbf{L} \in N_j(\mathbf{L}_i)} f|_{S_j}(\mathbf{L})$ 
9:   until  $i \geq i_{\text{max}}$  OR  $\frac{f(\mathbf{L}_i) - f(\mathbf{L}_{i-1})}{f(\mathbf{L}_{i-1})} \leq \varepsilon$ 

```

Subgroup study As the size of the subgroup set S_i heavily influences the runtime of Algorithm 2, it is necessary to find values for ℓ_{subgroup} that yield good optimization results in a feasible time. For this test, the original layout of PS10 was optimized for AEP with different values for the subgroup scaling factor. A neighborhood structure with eight positions on two circles with a radius scaling factor of one was chosen. Furthermore, the maximum number of iterations was set to ten and the convergence criterion to 0.01%. Table 14 summarizes the settings for this test.

The results of each test run are shown in Figure 27. It can be seen that the runtime per iteration increases with an increasing subgroup radius, as the subgroup size and therefore the total number of simulated heliostats increases. However, the maximum improvement does not seem to scale with the subgroup size. The AEP improvement

Parameter	Description
ℓ_{circle}	Scaling factor of circle radius
n_{circle}	Number of equidistant circles
n_{position}	Number of equidistant positions on a circle
i_{max}	Maximum number of iterations
ε	Minimum improvement
ℓ_{subgroup}	Scaling factor of the subgroup set
f	Objective function

Table 13: Summary of all parameters, which control Algorithm 2. The parameters of the first block define the neighborhood structure, those of the second block control the termination behavior, and those of the last block control the optimization objective.

for values of ℓ_{subgroup} greater than five, compared to $\ell_{\text{subgroup}} = 5$ is not significant, while the runtime gets much worse. Therefore, $\ell_{\text{subgroup}} = 5$ was chosen to be constant, since this configuration provided the best AEP improvement in the shortest time.

Parameter	Value
ℓ_{circle}	1
n_{circle}	2
n_{position}	8
i_{max}	10
ε	0.01 %
ℓ_{subgroup}	{3, 4, 5, 6, 7, 8}
f	AEP

Table 14: Settings of the local search algorithm with subgrouping when run on the original PS10 heliostat layout with different values for ℓ_{subgroup} for the subgroup tests.

Parameter study With the subgroup radius factor ℓ_{subgroup} set to five, it is of interest to further determine good parameter combinations of the neighborhood structure. For this purpose, Algorithm 2 was run on the original layout of PS10 with different combinations for the number of circles n_{circle} , number of positions per circle n_{position} and the circle radius factor ℓ_{circle} .

The test results are shown in Figure 28. It can be clearly seen that the algorithm performs significantly worse for ℓ_{circle} less than three than with a value greater or equal to three, regardless of the other two parameters. Moreover, three positions per circle and a single circle performs relatively badly.

The top five parameter combinations resulting in the highest AEP improvement over the original heliostat layout of PS10 are listed in Table 15. The best performing combination $\ell_{\text{circle}} = 8$, $n_{\text{circle}} = 2$ and $n_{\text{position}} = 6$ was chosen. Since the neighborhood set contains only 13 positions to be evaluated, this also has a positive effect on the runtime.

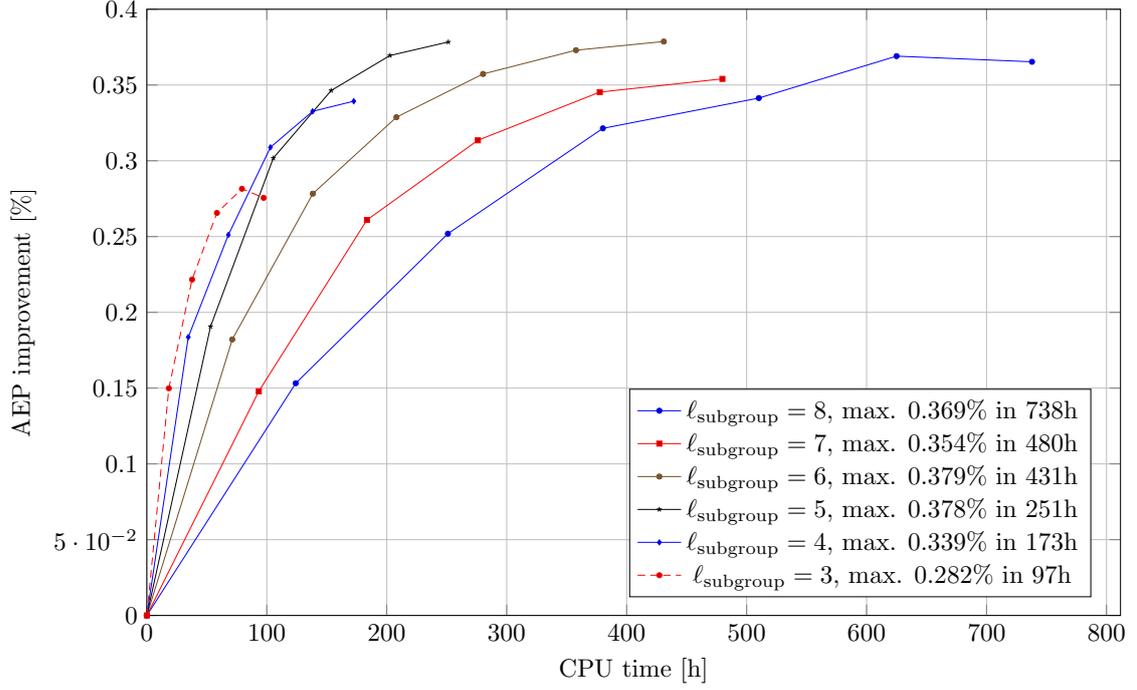


Figure 27: Results of the local search algorithm with subgrouping on the original PS10 layout for different values for the subgroup scaling l_{subgroup} . All settings for this test are listed in Table 14.

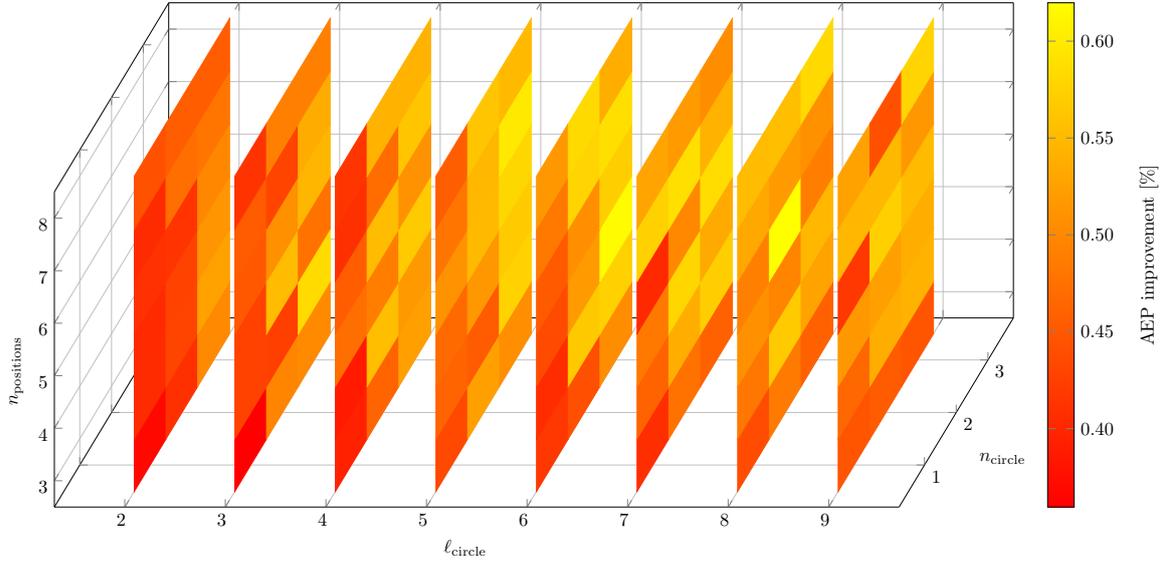


Figure 28: PS10 AEP optimization results of Algorithm 2 for different combinations for the parameters n_{circle} , n_{position} and l_{circle} of the neighborhood structure.

Convergence study To determine a feasible termination condition, Algorithm 2 was run with the five best performing parameter combinations from Table 15 as long as

Parameter					
ℓ_{circle}	8	6	5	7	6
n_{circle}	2	3	3	2	3
n_{position}	6	5	7	7	7
max. AEP improvement	0.620 %	0.616 %	0.597 %	0.587 %	0.587%

Table 15: Five best performing parameter combinations of the neighborhood structure, yielding the highest AEP improvement over the original heliostat layout of PS10.

there was a positive improvement over the previous iteration, i.e., a positive relative improvement. Figure 29 shows the relative improvement of these five combinations per iteration. As time progressed, the relative improvement over the last iteration quickly decreased until eventually a negative relative improvement was made. This is likely caused by inaccuracies due to the tradeoffs in the number of rays in ray tracing and the subgrouping approach.

As the highest relative improvements are made in the first two iterations of the local search, a minimum relative improvement of 0.03% was chosen as a convergence criterion as a tradeoff between runtime and good optimization results. The chosen parameter configuration would thus terminate after at most four iterations for the original heliostat layout of PS10, although the relative improvement rises above this threshold in the sixth iteration. The maximum number of local search iterations i_{max} was also set to five, to further restrict the runtime of the algorithm. An overview on the final parameter configuration for the local search algorithm is given in Table 16.

Parameter	Value
ℓ_{circle}	8
n_{circle}	2
n_{position}	6
i_{max}	5
ε	0.03 %
ℓ_{subgroup}	5
f	AEP

Table 16: Final parameter configuration of the local search algorithm with subgrouping.

Verification To verify that the local search algorithm for heliostat layout optimization works as intended, it was run on the original heliostat layouts of PS10, PS20, and Gemasolar. The resulting AEP improvements are listed in Table 17. The local search was able to improve the original layouts of PS10 by 0.466%, PS20 by 0.106% and Gemasolar by 1.932%. Additionally, the optimized layouts, overlaid with the original heliostat layouts can be found in Figures 33, 38 and 43. It is noticeable that the terrain influences the decisions of the local search algorithm, as, for example, heliostats are moved uphill to reduce shadowing and blocking effects.

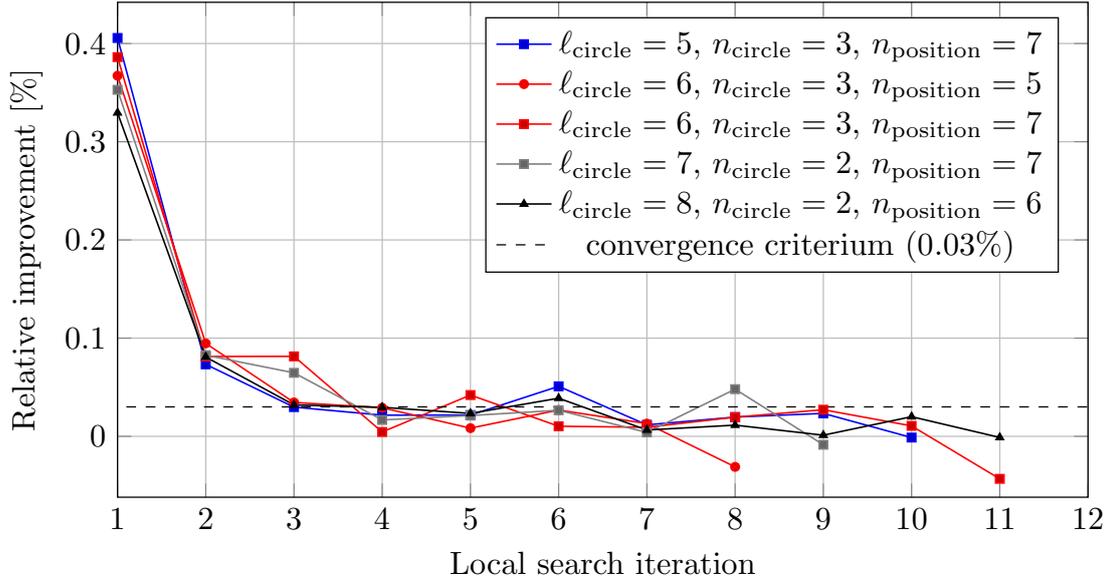


Figure 29: Relative improvements for each iterations of the local search algorithm for the top five parameter combinations. As time progresses, the relative improvements quickly decrease until eventually a negative relative improvement is made. The chosen convergence criterion marks the boundary after which the algorithm would terminate.

	PS10	PS20	Gemasolar
AEP original layout	26.819 GWh	51.112 GWh	88.639 GWh
AEP optimized layout	26.944 GWh	51.167 GWh	90.352 GWh
AEP improvement	0.466%	0.106%	1.932%
Local search iterations	3	2	5
Figure	33	38	43

Table 17: Results of AEP optimization of the local search algorithm on the original layouts of PS10, PS20 and Gemasolar.

4. Case study

For the case study, the AEP of the heliostat layouts were optimized for PS10, PS20 and Gemasolar using the presented multi-stage optimization pipeline. First, for each CSP plant and pattern, the pattern parameters were optimized using the downhill simplex method. The optimized pattern parameters for PS10, PS20 and Gemasolar are listed in Table 18, 20 and 22. These optimized patterns were then fed into the local search algorithm (LS). The resulting heliostat layouts of the pattern optimization and the multi-step optimization were compared in terms of their AEP in Table 19, 21 and 23, using the original layouts as a reference. Finally, the effects of AEP optimization on the annual revenue (AR), internal rate of return (IRR), net present value (NPV), payback period (PP) and levelized cost of energy (LCOE) were compared between the pattern optimization and the multi-step optimization in Figure 30, 31 and 32.

4.1. Layout optimization for PS10

Pattern	Parameters				Section
Cornfield	$\ell_x = 1.18907$	$\ell_y = 1.00924$	$s_x = 1.01895$	$s_y = 1.28248$	3.1.1
Hexagon	$d = 1.54698$		$s = 0.73040$	$\beta = 0.0$	3.1.2
Radial staggered	$d = 1.35155$	$g = 0.85277$	$s = 0.97925$	$\beta = 83.813$	3.1.3
Spiral	$a = 0.59600$	$b = 0.92237$	$s = 0.84737$	$\beta = 138.134$	3.1.4

Table 18: PS10 optimized parameters for the Cornfield, Hexagon, Radial staggered and Spiral patterns.

Heliostat layout	AEP	Improvement	Figure
Original (real-world)	23.4 GWh [5]	-	8a
Original	26.819 GWh	-	8a
Cornfield	26.502 GWh	-1.183 %	
Cornfield + LS	26.775 GWh	-0.165 %	34
Hexagon	26.933 GWh	0.424 %	
Hexagon + LS	27.040 GWh	0.825 %	35
Radial staggered	27.008 GWh	0.705 %	
Radial staggered + LS	27.067 GWh	0.927 %	36
Spiral	26.953 GWh	0.501 %	
Spiral + LS	27.060 GWh	0.898 %	37

Table 19: PS10 AEP of the original layout, compared to the optimized patterns from Table 18 as well as the presented multi-step optimization that additionally performs a local search. The estimated annual energy production of the real PS10 plant is about 23.4 GWh.

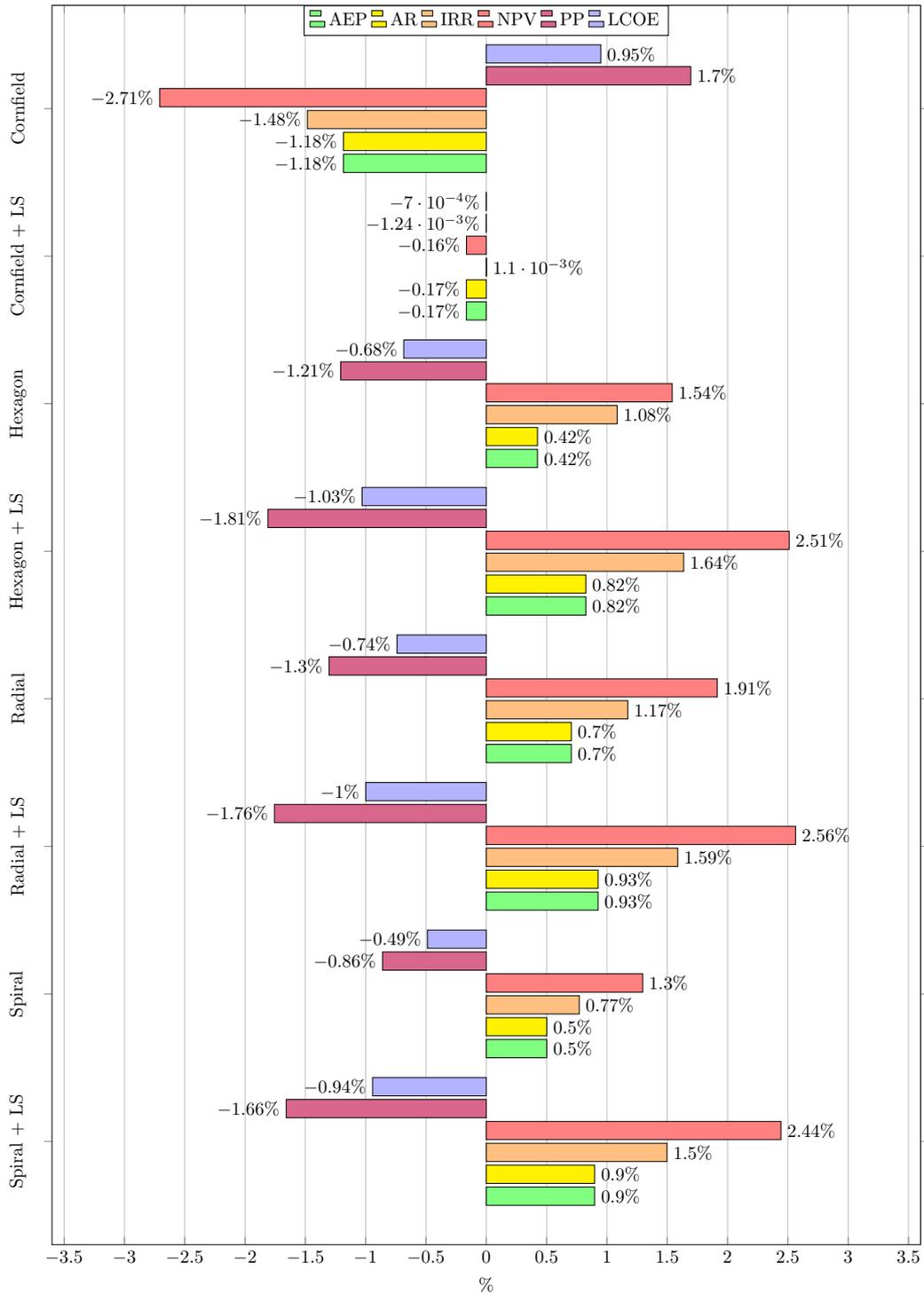


Figure 30: Impact of AEP optimization of PS10 with pattern optimization and multi-step optimization on the economic values, which are normalized to the original layout.

4.2. Layout optimization for PS20

Pattern	Parameters				Section
Cornfield	$\ell_x = 1.00482$	$\ell_y = 1.02122$	$s_x = 1.04334$	$s_y = 1.21462$	3.1.1
Hexagon	$d = 1.40661$		$s = 0.79272$	$\beta = 0.0$	3.1.2
Radial staggered	$d = 1.42959$	$g = 0.75772$	$s = 0.88296$	$\beta = 92.416$	3.1.3
Spiral	$a = 1.24243$	$b = 0.78726$	$s = 0.93436$	$\beta = 0.0$	3.1.4

Table 20: PS20 optimized parameters for the Cornfield, Hexagon, Radial staggered and Spiral patterns.

Heliostat layout	AEP	Improvement	Figure
Original (real-world)	48 GWh [6]	-	8b
Original	51.112 GWh	-	8b
Cornfield	50.633 GWh	-0.937 %	
Cornfield + LS	50.889 GWh	-0.436 %	39
Hexagon	51.273 GWh	0.316 %	
Hexagon + LS	51.311 GWh	0.391 %	40
Radial staggered	51.297 GWh	0.363 %	
Radial staggered + LS	51.402 GWh	0.567 %	41
Spiral	51.225 GWh	0.221 %	
Spiral + LS	51.313 GWh	0.393 %	42

Table 21: PS20 AEP of the original layout, compared to the optimized patterns from Table 20 as well as the presented multi-step optimization that additionally performs a local search. The estimated annual energy production of the real PS20 plant is about 48 GWh.

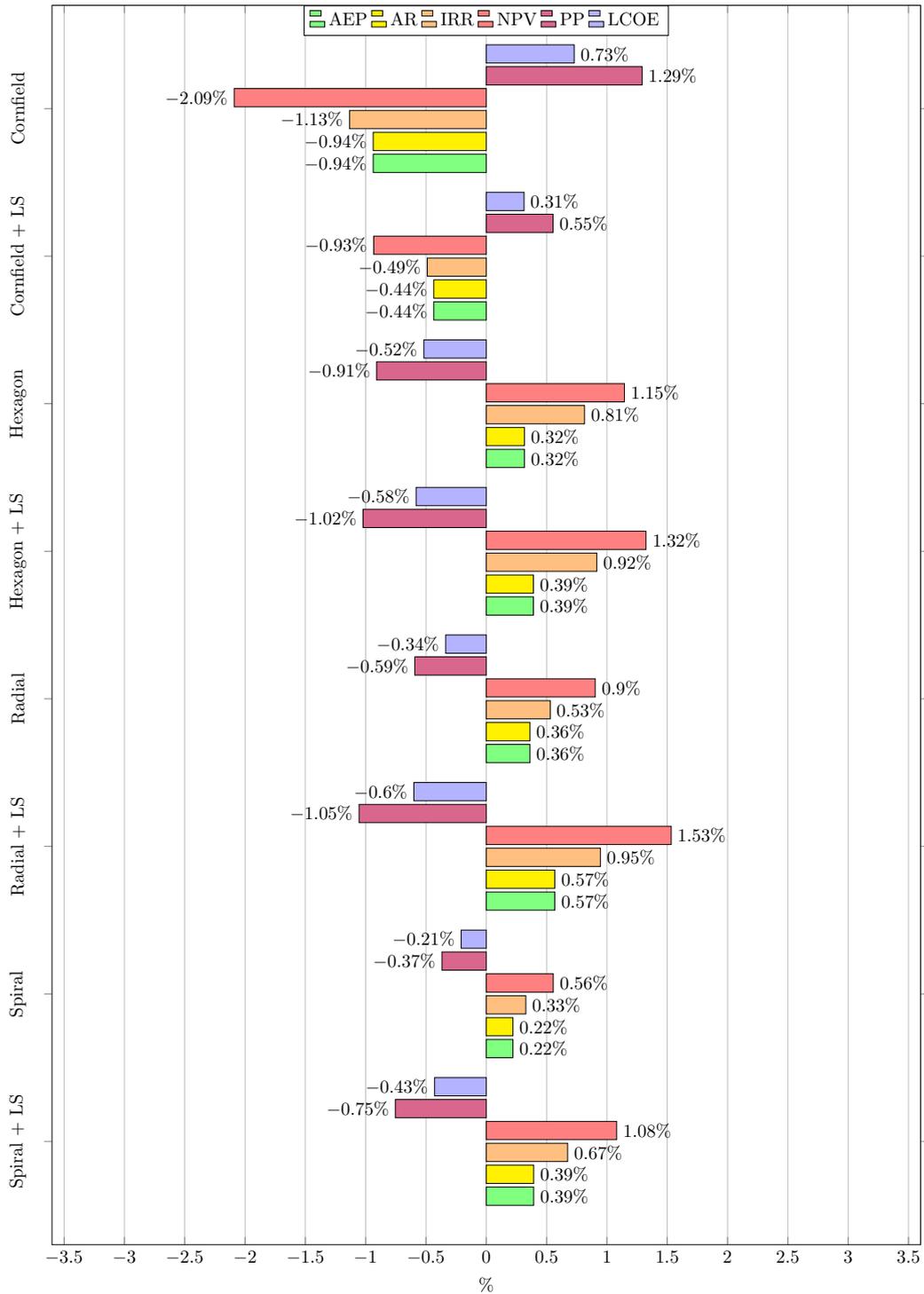


Figure 31: Impact of AEP optimization of PS20 with pattern optimization and multi-step optimization on the economic values, which are normalized to the original layout.

4.3. Layout optimization for Gemasolar

Pattern	Parameters				Section
Cornfield	$\ell_x = 1.11822$	$\ell_y = 1.00008$	$s_x = 1.08718$	$s_y = 1.18231$	3.1.1
Hexagon	$d = 1.6$		$s = 0.62243$	$\beta = 5.059$	3.1.2
Radial staggered	$d = 1.13883$	$g = 1.07047$	$s = 0.90689$	$\beta = 152.230$	3.1.3
Spiral	$a = 1.02115$	$b = 0.83696$	$s = 0.80007$	$\beta = 159.977$	3.1.4

Table 22: Gemasolar optimized parameters for the Cornfield, Hexagon, Radial staggered and Spiral patterns.

Heliostat layout	AEP	Improvement	Figure
Original (real-world)	80 GWh [2]	-	8c
Original	88.639 GWh	-	8c
Cornfield	88.690 GWh	0.058 %	
Cornfield + LS	89.870 GWh	1.389 %	44
Hexagon	89.652 GWh	1.144 %	
Hexagon + LS	90.368 GWh	1.951 %	45
Radial staggered	90.369 GWh	1.952 %	
Radial staggered + LS	90.541 GWh	2.146 %	46
Spiral	90.370 GWh	1.954 %	
Spiral + LS	90.516 GWh	2.118 %	47

Table 23: Gemasolar AEP of the original layout, compared to the optimized patterns from Table 22 as well as the presented multi-step optimization that additionally performs a local search. The estimated annual energy production of the real Gemasolar plant is about 80 GWh.

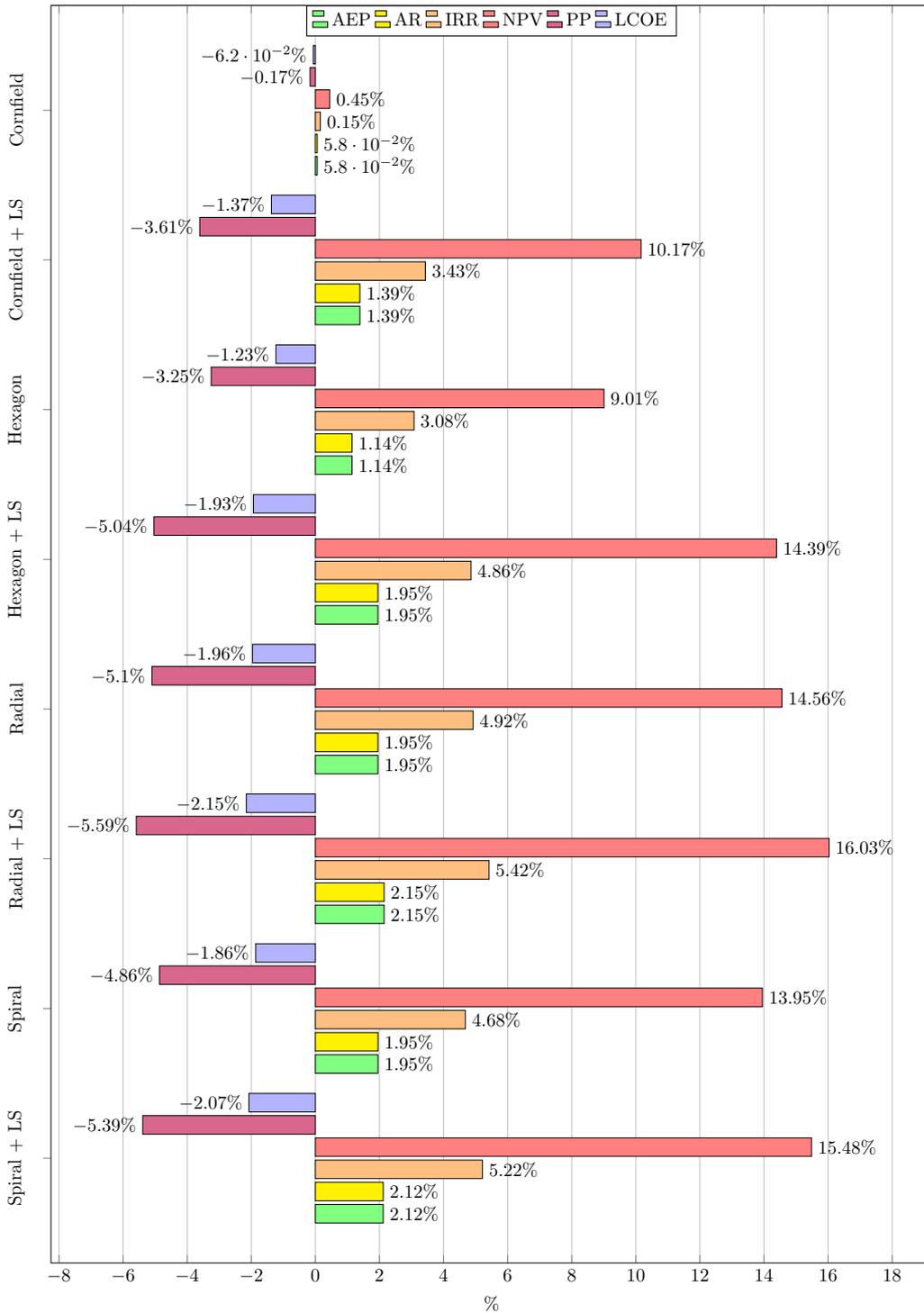


Figure 32: Impact of AEP optimization of Gemasolar with pattern optimization and multi-step optimization on the economic values, which are normalized to the original layout.

4.4. Discussion

The first thing to note is that the AEP of the modeled CSP plants significantly differs from the estimated AEP of the real CSP plants. The AEP yielded by the simulation differed by 14.6% for PS10, 6.5% for PS20 and 10.8% for Gemasolar. These deviations may occur for several possible reasons. On the one hand, the plant's cited real-world AEP values are estimated or planned values that fluctuate in reality simply due to weather conditions. On the other hand, the CSP model does not perfectly reflect reality, e.g. energy losses due to maintenance work or due to overheating of the receiver during phases of strong solar radiation are not taken into account. Furthermore, the model simplifies the energy storage and power generation processes and assumes the same generator with different capacity for all three CSP plants. Finally, the model requires accurate technical information about the power plants, but this is difficult to obtain, due to the incompleteness and inaccuracy of publicly available information on CSP plants.

The results of the case-study show that the four presented patterns perform differently. Considering only the pattern optimization, the hexagon pattern, radial staggered pattern and the spiral pattern outperform the original layouts. The staggered cornfield pattern performs the worst and is even significantly worse than the original layout of the PS10 and PS20 plants.

Compared to the pattern optimization, the multi-step optimization manages to further improve the heliostat layouts through the use of the local search. In general, the radial staggered pattern performs slightly better than the spiral pattern, closely followed by the hexagon pattern. Despite the local search, the staggered cornfield pattern performs the worst, even worse than the original layout for the PS10 and PS20 plants. The multi-step optimization generated the best performing heliostat layouts for all three CSP plants using the radial staggered pattern. The improvement of the AEP compared to the original heliostat layout was 0.927% for PS10, 0.567% for PS20, and 2.146% for Gemasolar (Table 19, 21, 23).

An AEP improvement of around one percent may not sound like much, but it should be noted that this means a similarly high AR improvement of tens to hundreds of thousands of euros per year. This results in an increased profitability of the CSP plant without incurring additional operating and maintenance costs.

The best performing layout for PS10 was able to increase the IRR by 1.59%, NPV by 2.56% and reduce the PP by 1.76% (Figure 30). For PS20 the best performing heliostat layout increased the IRR by 0.95%, NPV by 1.53% while the PP was reduced by 1.05% (Figure 31). Finally, the best performing layout for Gemasolar was able to increase the IRR by 5.42%, NPV by 16.03%, while decreasing the PP by 5.59% (Figure 32). Overall, the LCOE compared to the original layout was reduced by 1% for PS10, 0.6% for PS20 and 2.15% for Gemasolar, which enables renewable electricity from these CSP plants to be more competitive on the electricity market.

5. Conclusion

In this work, a novel multi-step optimization pipeline for heliostat layouts in CSP plants was developed. The underlying CSP model implemented in the **SunFlower** project was presented and the optical sub-model was successfully validated against the existing raytracing softwares **Tonatiuh** and **SolTrace**.

For the first step in the multi-step optimization pipeline, a *pattern-based* optimization using the downhill simplex method was implemented and applied to one of four different parametrized patterns. The search spaces of the three most promising patterns were extended by scaling along an arbitrary axis. For the second step of the pipeline, a *pattern-free* local search optimization algorithm was developed which refined the optimized patterns.

In a case-study, three Spanish CSP plants, PS10, PS20 and Gemasolar, were modeled and their heliostat layout were optimized using the multi-step optimization pipeline. The results of the case study show that multi-step optimization not only provides better results than pattern optimization alone, but can also improve the original layouts and thus increase the profitability and competitiveness of these CSP plants.

Limitations Despite good results, there are some limitations to the multi-step optimization algorithm.

- **Long runtime**

The runtime depends linearly on the number of desired heliostats in the field. Due to ever larger CSP plants with more and more heliostats, this optimization pipeline is becoming increasingly slower. Especially the pattern evaluation (Section 3.1.5) slows down the downhill simplex algorithm since most of the time an additional annual simulation of a substantially larger pattern is needed to select the most efficient heliostat positions.

- **Local search**

After a few iterations, the local search algorithm worsen the AEP of the heliostat layout as seen in Figure 29, indicating that the algorithm moves heliostats to more inefficient positions. This is probably due to the subgrouping approach, which ignores the influence of non-adjacent heliostats. Additionally, it may also be due to inaccuracies caused by the subgrouping and a small number of rays traced per square meter.

Another disadvantage of the local search is that very regular patterns can become quite chaotic due to local optimization as seen in the optimized heliostat fields in the appendix, which is a problem for maintenance roads through the heliostat field.

Future work To further improve the performance of the multi-step heliostat layout optimization, it may be of interest to investigate the following ideas:

- **Improve simulation runtime**

The optimization heavily benefits from an improved simulation runtime as more possible layouts can be tested in the same amount of time. The existing model may be analyzed to find ways to parallelize computations. Especially the ray-tracer of the optical model, which is the most time-consuming part of the entire model simulation may benefit from parallelization. It may even be possible to utilize the GPU to further accelerate the raytracing.

- **Pattern evaluation**

The current pattern evaluation (Section 3.1.5) needs to evaluate a substantially larger pattern to be able to select the most efficient heliostat positions. A more runtime efficient approach is to select the desired heliostat positions based on their position in the field using an approach similar to the precomputed Yearly Normalized Energy Surface map as proposed by Sanchez and Romero [32].

- **Pattern parameters**

Additional parameters, which increase the search space for the pattern optimization may be investigated.

- **Pattern optimization**

Other derivative-free optimization algorithms such as the Multilevel Coordinate Search [20] may be implemented and their number of evaluations and convergence behavior compared with the downhill simplex method.

- **Pattern-free optimization**

Instead of the local search, other *pattern-free* optimization algorithms such as Simulated Annealing, which can be initialized with a given heliostat layout, may be implemented and their performance in the multi-step optimization pipeline compared with local search.

References

- [1] Sevilla weather data, 2020.
https://energyplus.net/weather-location/europe_wmo_region_6/ESP/ESP_Sevilla.083910_IWEC.
- [2] Gemasolar, 2020.
<https://solarpaces.nrel.gov/gemasolar-thermosolar-plant>.
- [3] International emissions, 2020.
<http://www.c2es.org/content/international-emissions>.
- [4] Paris agreement, 2020.
https://ec.europa.eu/clima/policies/international/negotiations/paris_en.
- [5] Planta solar 10, 2020.
<https://solarpaces.nrel.gov/planta-solar-10>.
- [6] Planta solar 20, 2020.
<https://solarpaces.nrel.gov/planta-solar-20>.
- [7] IRENA (2020). *Renewable capacity statistics 2020*. International Renewable Energy Agency, Abu Dhabi, 2020.
- [8] IRENA (2020). *Renewable Power Generation Costs in 2019*. International Renewable Energy Agency, Abu Dhabi, 2020.
- [9] Germain Augsburguer. *Thermo-economic optimisation of large solar tower power plants*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2013.
- [10] JG Barberena, A Mutuberria Larrayoz, Marcelino Sánchez, and Ana Bernardos. State-of-the-art of heliostat field layout algorithms and their comparison. *Energy Procedia*, 93:31–38, 2016.
- [11] M. J. Blanco, J. M. Amieva, and A. Mancillas. The tonatiuh software development project: An open source approach to the simulation of solar concentrating systems. In *In ASME 2005 International Mechanical Engineering Congress and Exposition*, pages 157–164. American Society of Mechanical Engineers, 2005.
- [12] Reiner Buck. Heliostat field layout improvement by nonrestricted refinement. *Journal of Solar Energy Engineering*, 136(2), 2014.
- [13] E Carrizosa, C Domínguez-Bravo, E Fernández-Cara, and M Quero. An optimization approach to the design of multi-size heliostat fields. *Tech. rep., Institute of Mathematics of University of Seville (IMUS)*, 2014.
- [14] Francisco J Collado and Jesús Guallar. Campo: Generation of regular heliostat fields. *Renewable energy*, 46:49–59, 2012.

- [15] Stella Coumbassa. Optimal storage strategy for hybrid concentrated solar thermal - photovoltaic plants. 2019.
- [16] Christopher N Elkinton, James F Manwell, and Jon G McGowan. Algorithms for offshore wind farm layout optimization. *Wind Engineering*, 32(1):67–84, 2008.
- [17] Linus Franke. Modelling and optimization of large scale solar tower power plants. 2018.
- [18] Pierre Hansen, Nenad Mladenović, and José A Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1): 367–407, 2010.
- [19] Gregor Heiming. *Development of a techno-economic model for solar tower power plants*. PhD thesis, Master’s thesis, MathCCES-RWTH Aachen, gregor.heiming@rwthachen.de, 11 2017.
- [20] Waltraud Huyer and Arnold Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.
- [21] Leary P.L. Hankins J.D. A user’s guide for mirval- a computer code for programming designs of heliostat-receiver optics for central receiver solar power plants. page pp. 14. American Society of Mechanical Engineers, 1979.
- [22] Sihoon Kim, Ikjin Lee, and Bong Jae Lee. Development of performance analysis model for central receiver system and its application to pattern-free heliostat layout optimization. *Solar energy*, 153:499–507, 2017.
- [23] Bruce L Kistler. A user’s manual for delsol3: a computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants. Technical report, Sandia National Labs., Livermore, CA (USA), 1986.
- [24] FW Lipps and Lorin L Vant-Hull. A cellwise method for the optimization of large central receiver systems. *Solar Energy*, 20(6):505–516, 1978.
- [25] FW Lipps and Lorin L Vant-Hull. Programmer’s manual for the university of houston computer code rcell: cellwise optimization for the solar central receiver project. Technical report, Houston Univ., TX (USA). Energy Lab., 1980.
- [26] Singh L Lutchman, Albert A Groenwold, Paul Gauché, and S Bode. On using a gradient-based method for heliostat field layout optimization. *Energy Procedia*, 49:1429–1438, 2014.
- [27] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [28] Corey J Noone, Manuel Torrilhon, and Alexander Mitsos. Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy*, 86(2):792–803, 2012.

- [29] Quoc Pham, Christian Gregory, Michael Slack, Bill Gross, Dan Reznik, and Porter Arbogast. Heliostat array layouts for multi-tower central receiver solar power plants, February 25 2014. US Patent 8,656,907.
- [30] Andreas Reinholz, Christof Husenbeth, Peter Schwarzbözl, and Reiner Buck. Optimizing heliostat positions with local search metaheuristics using a ray tracing optical model. In *AIP Conference Proceedings*, volume 1850, page 160023. AIP Publishing LLC, 2017.
- [31] Pascal Richter. *Simulation and optimization of solar thermal power plants*. Dissertation, RWTH Aachen University, Aachen, 2017. URL <http://publications.rwth-aachen.de/record/690762>. Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2017.
- [32] Marcelino Sanchez and Manuel 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.
- [33] FMF Siala and ME Elayeb. Mathematical formulation of a graphical method for a no-blocking heliostat field layout. *Renewable energy*, 23(1):77–92, 2001.
- [34] Tim Wendelin. Soltrace: a new optical modeling tool for concentrating solar optics. In *ASME 2003 International Solar Energy Conference*, pages 253–260. American Society of Mechanical Engineers, 2003.
- [35] Seonghyeok Yang, Kyungeun Lee, and Ikjin Lee. Pattern-free heliostat field layout optimization using physics-based gradient. *Solar Energy*, 206:722–731, 2020.
- [36] Maolong Zhang, Lijun Yang, Chao Xu, and Xiaoze Du. An efficient code to optimize the heliostat field and comparisons between the biomimetic spiral and staggered layout. *Renewable Energy*, 87:720–730, 2016.

A. Optimized heliostat layouts

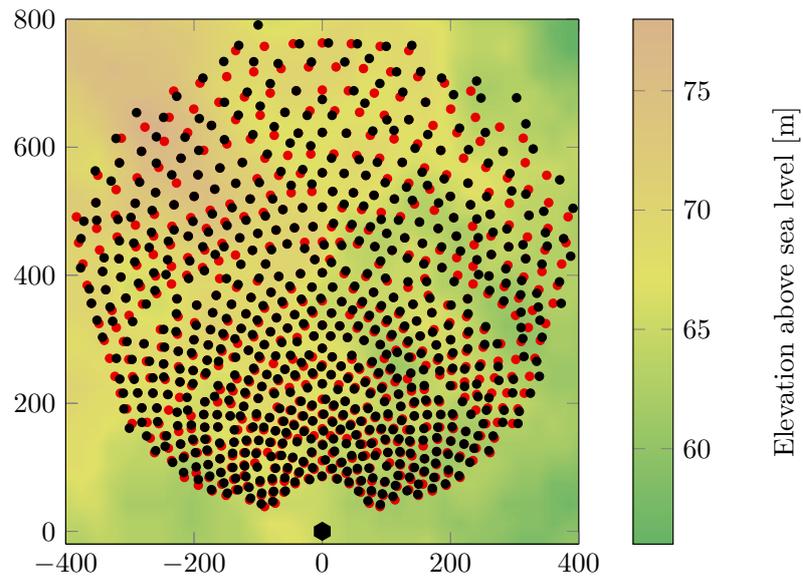


Figure 33: Original PS10 layout (red) and optimized layout (black) from the local search algorithm.

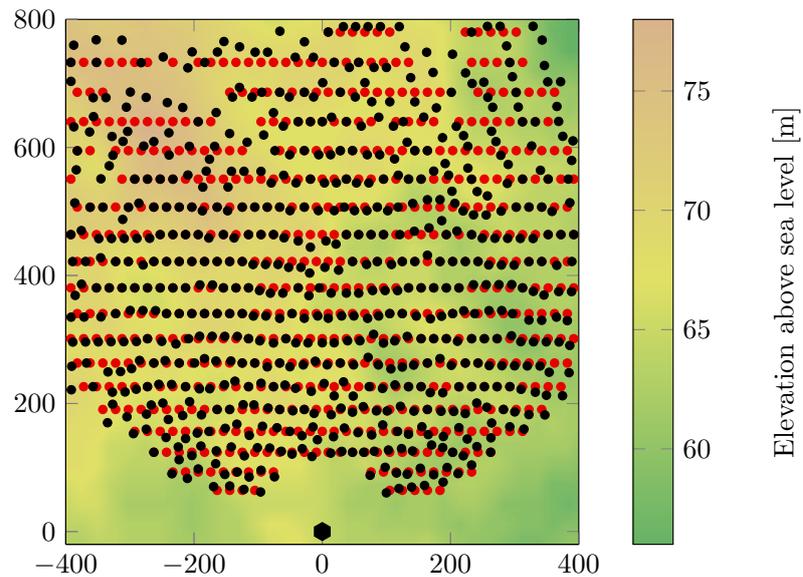


Figure 34: Staggered cornfield pattern for PS10 before (red) and after local search optimization (black).

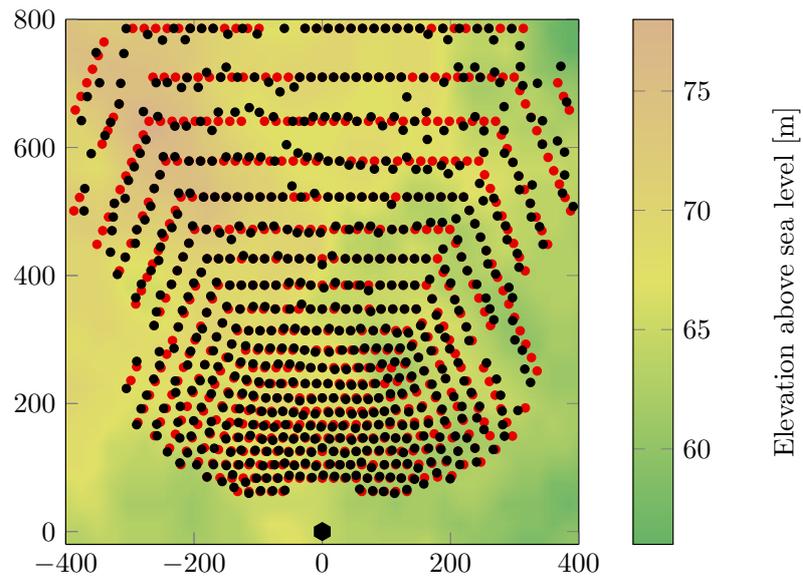


Figure 35: Hexagon pattern for PS10 before (red) and after local search optimization (black) .

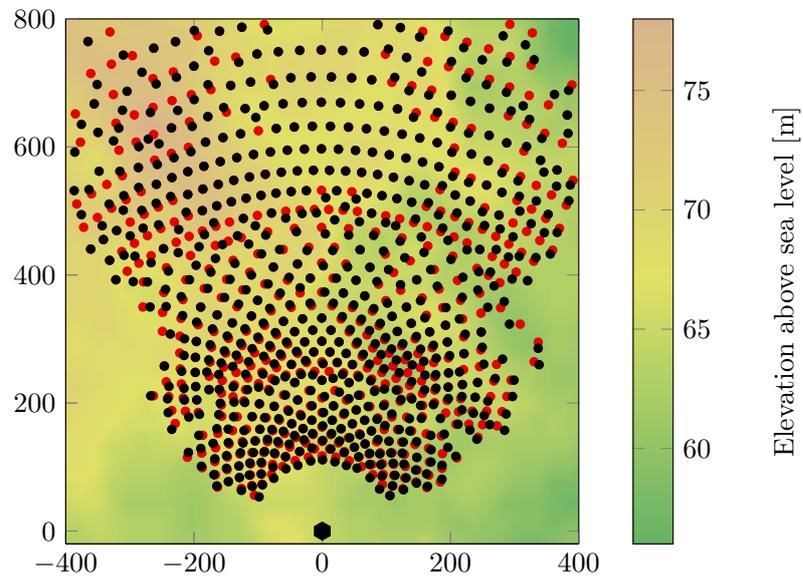


Figure 36: Radial staggered pattern for PS10 before (red) and after local search optimization (black) .

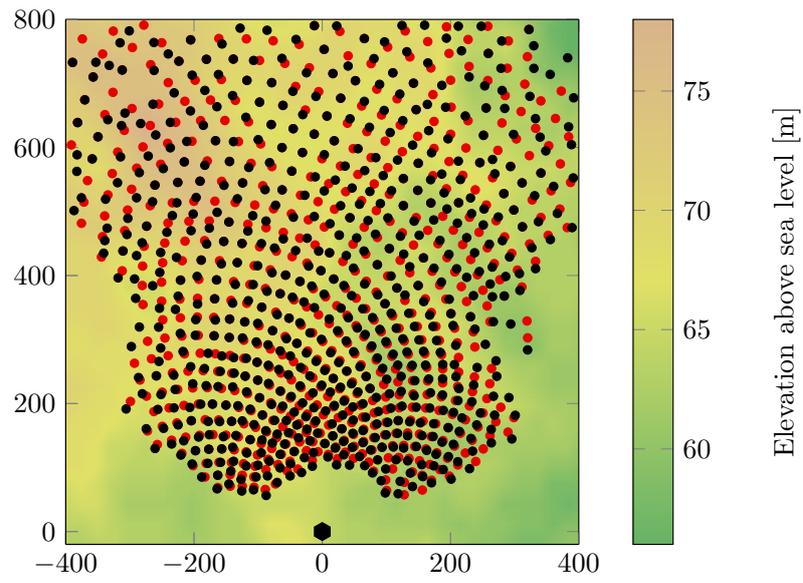


Figure 37: Spiral pattern for PS10 before (red) and after local search optimization (black) .

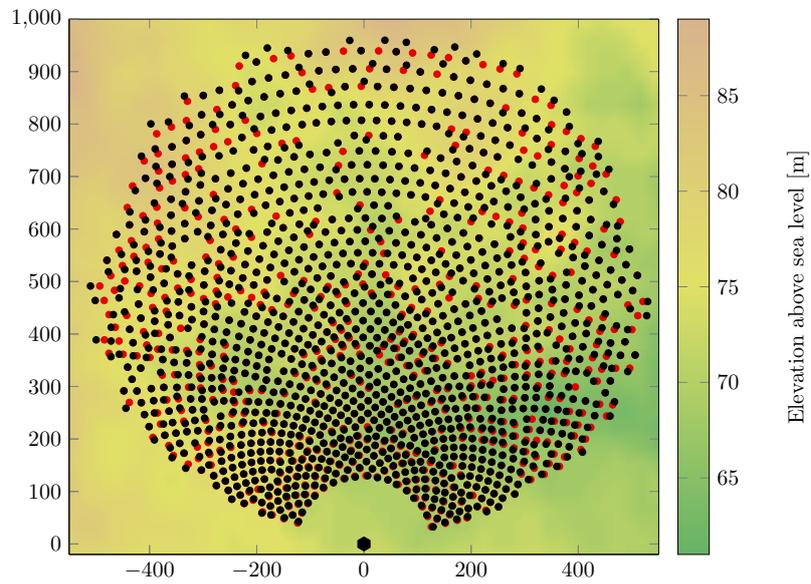


Figure 38: Original PS20 layout (red) and optimized layout (black) from the local search algorithm.

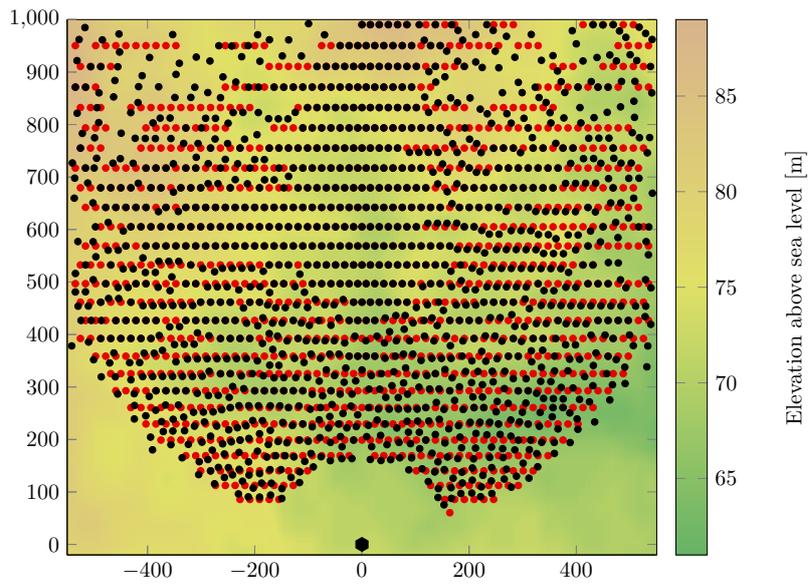


Figure 39: Staggered cornfield pattern for PS20 before (red) and after local search optimization (black).

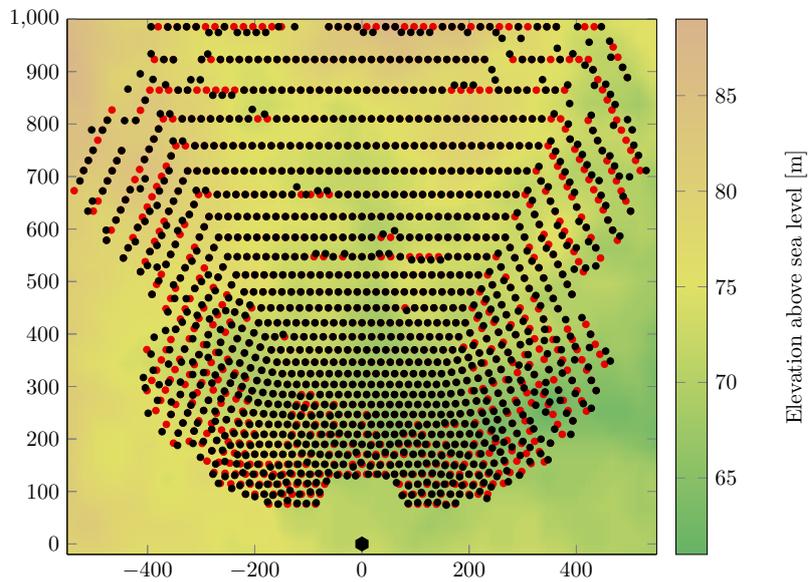


Figure 40: Hexagon pattern for PS20 before (red) and after local search optimization (black).

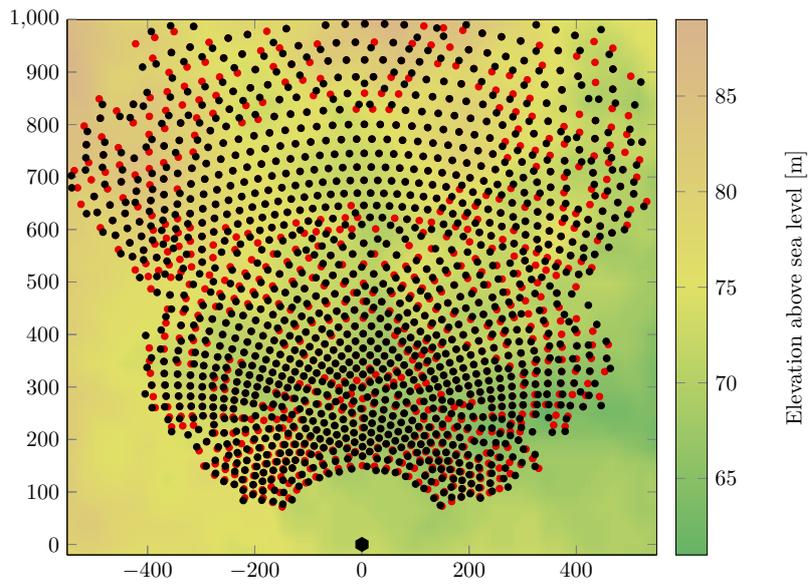


Figure 41: Radial staggered pattern for PS20 before (red) and after local search optimization (black).

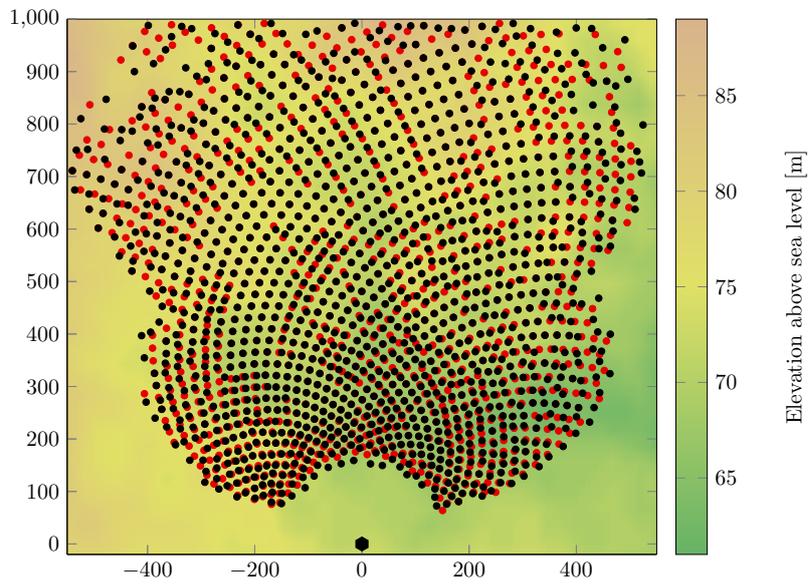


Figure 42: Spiral pattern for PS20 before (red) and after local search optimization (black).

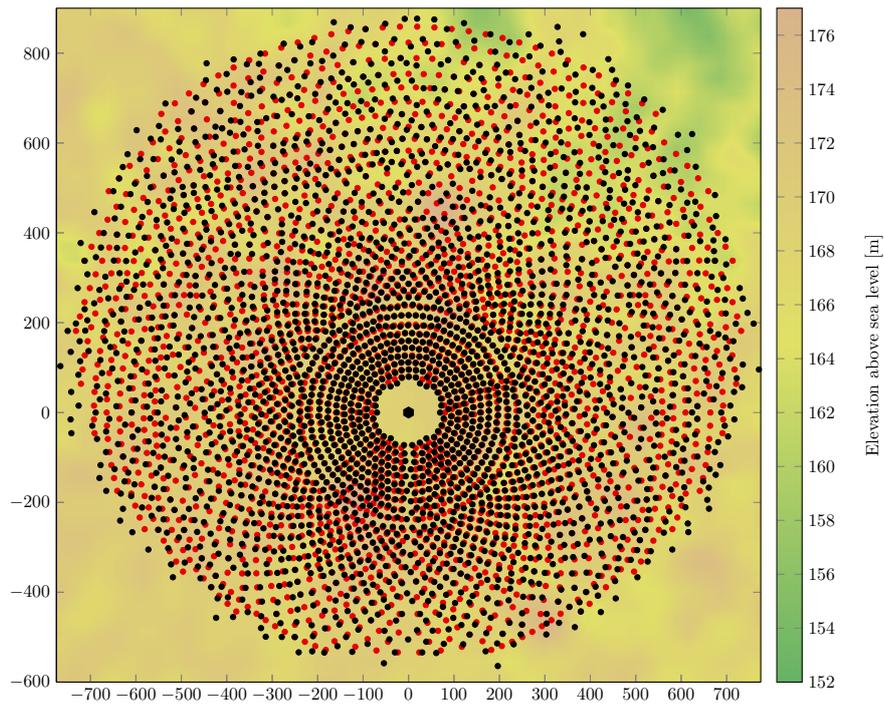


Figure 43: Original Gemasolar layout (red) and optimized layout (black) from the local search algorithm.

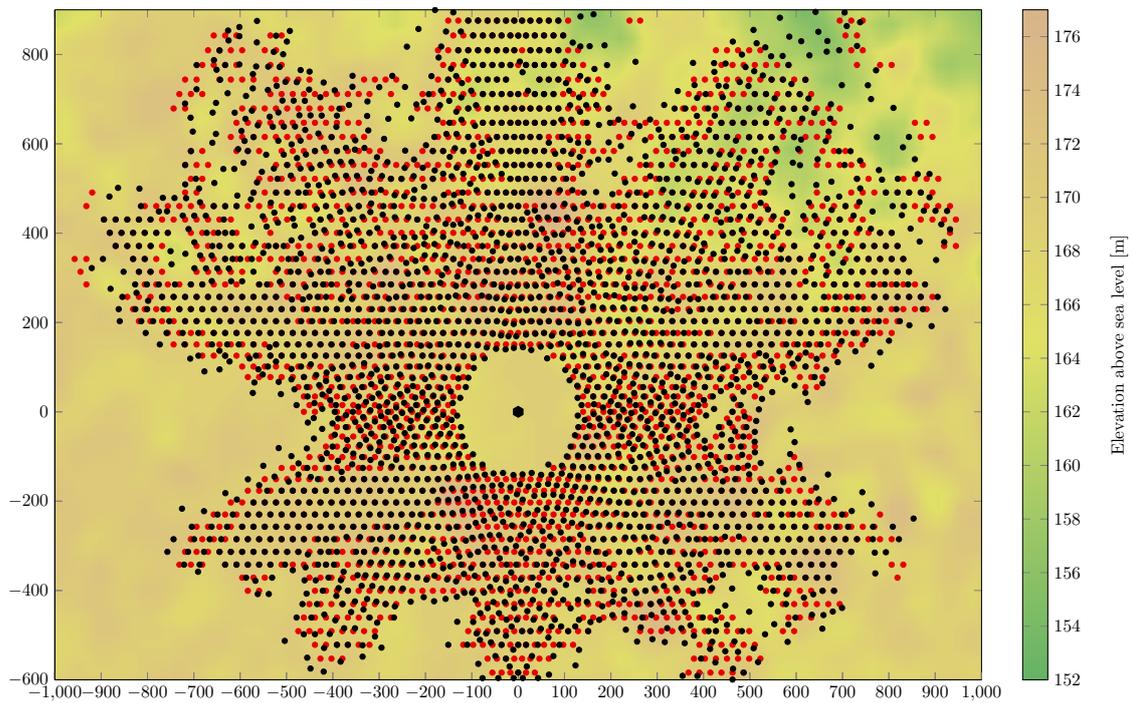


Figure 44: Staggered cornfield pattern for Gemasolar before (red) and after local search optimization (black).

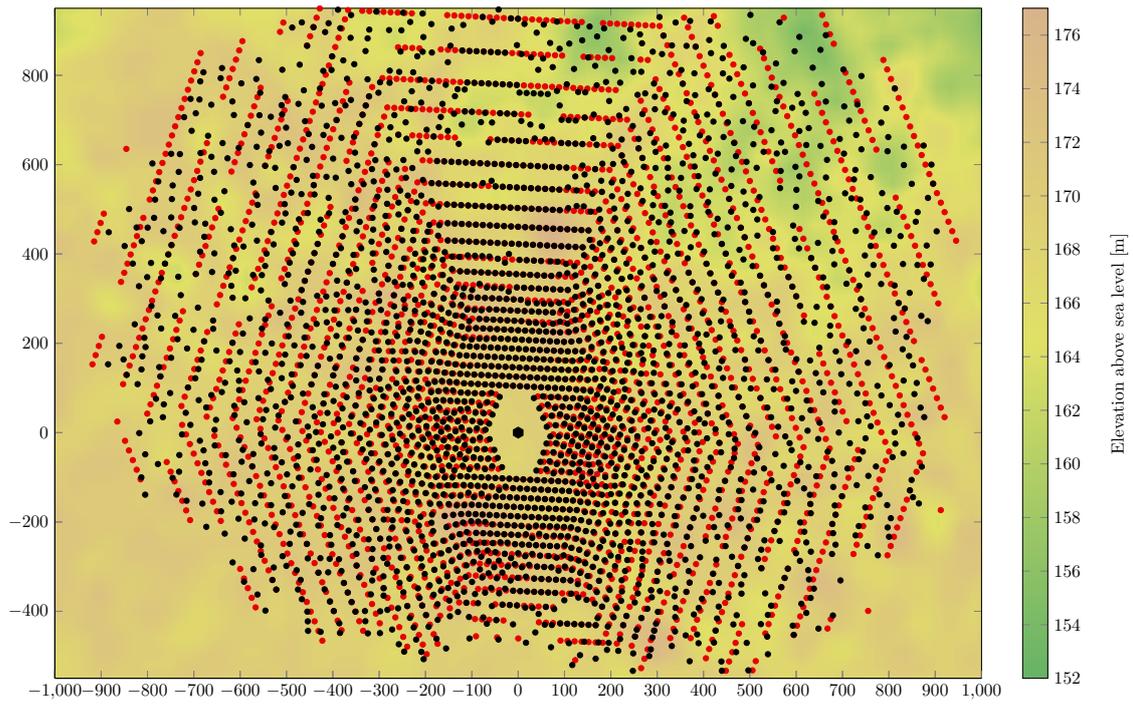


Figure 45: Hexagon pattern for Gemasolar before (red) and after local search optimization (black).

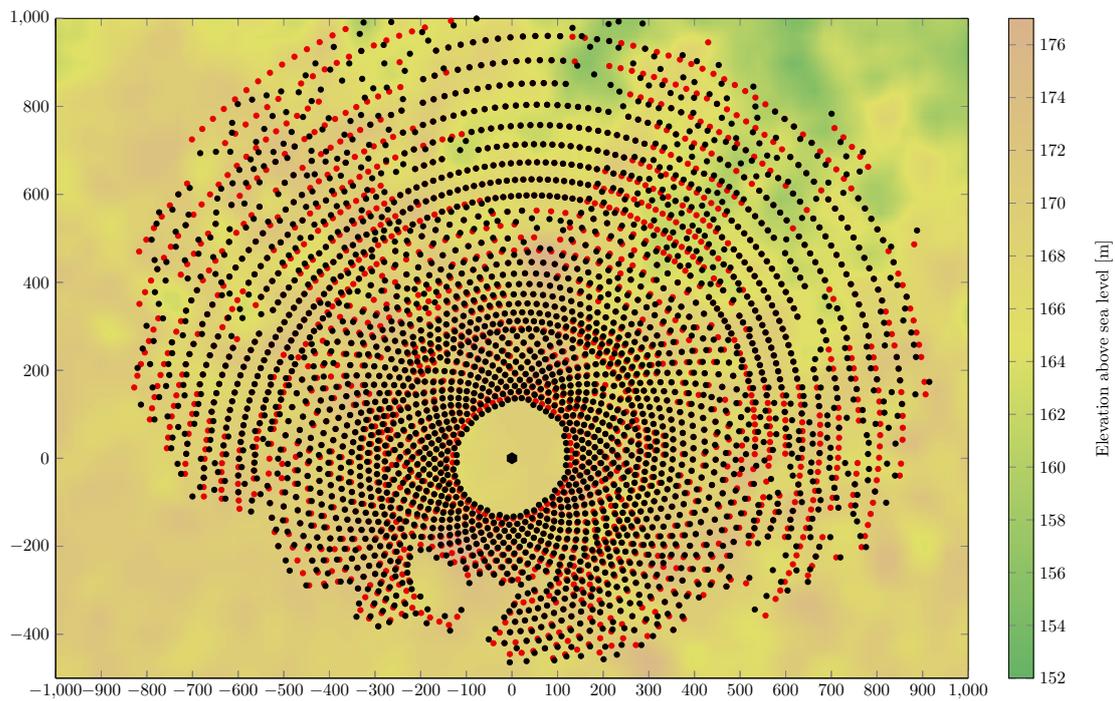


Figure 46: Radial staggered pattern for Gemasolar before (red) and after local search optimization (black).

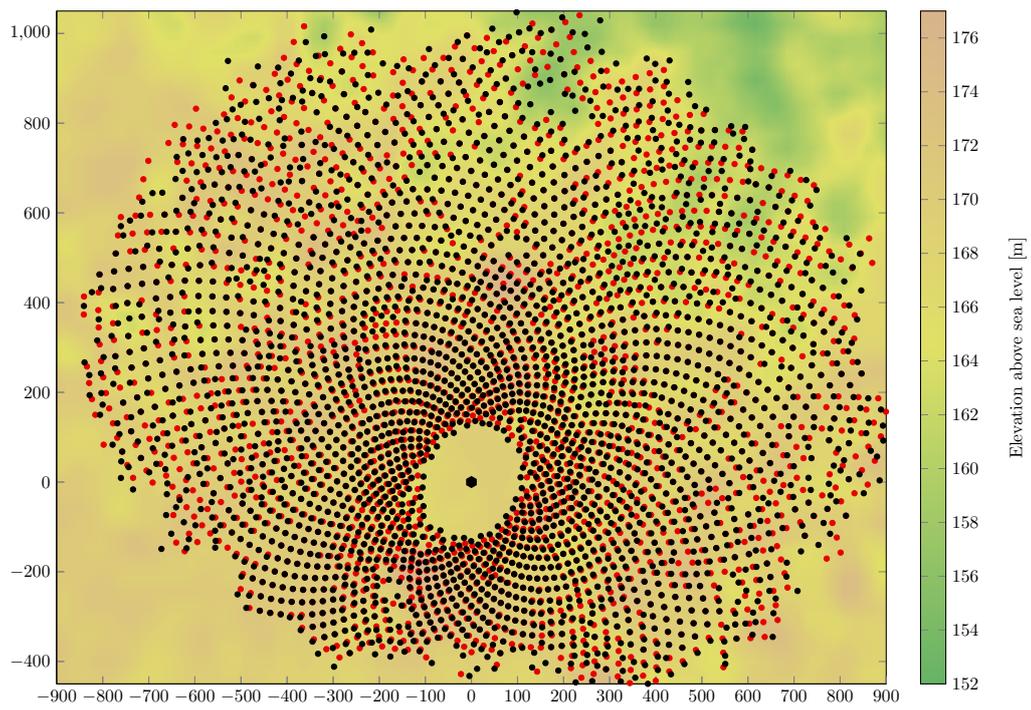


Figure 47: Spiral pattern for Gemasolar before (red) and after local search optimization (black).