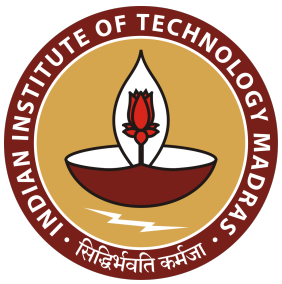
Solar Tower Heliostat Field Layout Optimization
A Black-Box Optimizer



A DAAD-IIT Master Sandwich

M. Tech. Thesis

Submitted by

Manish Krishan Lal
MA15M007

In partial fulfillment of the requirements
for the award of the degree of

Master of Technology
in
Industrial Mathematics & Scientific Computing

Department of Mathematics, IIT Madras
&
Lehrstuhl für Mathematik (MathCCES),
RWTH Aachen

May 2017

DECLARATION

This is to declare that the thesis entitled **"Solar Tower Heliostat Field Layout Optimization"** is the result of the investigations carried out by me at Indian Institute of Technology, Madras, and RWTH, Aachen, Germany under the supervision of Prof. Dr. Martin Frank and Prof. S. Sundar. In keeping with the general practice of reporting scientific observations, due acknowledgement has been made whenever the work described is based on the findings of other investigations. Any omission which might have occurred by oversight or error in judgment is regretted.

Chennai, May 2017

Manish Krishan Lal

ACKNOWLEDGEMENT

I cannot help mentioning those people who had great influence on me both intellectually and physically while pursuing my Masters degree Industrial Mathematics and Scientific Computing. I admit that without their great support and help, I would not finish this journey to obtain my Masters degree. First of all, I am very grateful to Prof. Dr. Martin Frank who accepted me as a DAAD exchange student and guided me during my research work. His brilliant ideas have always inspired me wherever I could not find a way to look at the problem in broader vision. I would also like to thank Prof. S. Sundar for his great inspirations and continous support which made me purely concentrate on the research problems as a graduate student.

I would also like to thank from the bottom of my heart to Pascal Richter for his guidance and brilliant ideas that helped me to complete my work within the limited span of time. I would like to thank my course supervisors and the members of Math CCES, RWTH Aachen along with International Office, for their support, and all special friends at IIT campus and at RWTH; who made my whole stay feel like ”home away from home”. I would like to acknowledge DAAD for providing the funds to carry out the Research work, RWTH-IT center for providing the computing resources. A very special thanks is reserved for GOD and my parents, elder brother Mohit, who have shaped and trained me to study in India's best engineering institution IIT Madras.

ABSTRACT

This thesis addresses the problems related to the design optimization of Solar-Tower field in the context of CSP (Concentrating Solar Power) systems. The design of the heliostat field is a challenging task of exceptional importance. It is the sub-system with the highest cost that amount to around 50% of the total cost and its optimal design highly depends on the project specific requirements. The placement of the mirrors may lead to individual mirrors being blocked and shaded and and major optical losses are associated with the field which all together affects the efficiency of the power plant.

To select an appropriate deterministic optimization procedure we need to parametrize the plant design as a function of design variables which will formulate the problem of finding optimal designs to the numerical problem of finding the optimum value of a function of several variables. This parametric optimization problem is solved with global optimization procedures to obtain the best design parameters. A hybrid method of Differential Evolution and Nelder-Mead is proposed which outperforms the Globalized Constrained Nelder-Mead method.

The optimal designed field can further be optimized with best-trial-neighbour heuristics as a post-processing step. Single-neighbour Selection algorithm is easier to implement than other two algorithms. Radius-dependent neighbour selection algorithm have less time-computation time; so better than Single and Multiple neighbour selection algorithms.

Contents

List of Figures

List of Tables

1. Introduction

As it is well said, there is no cheaper energy than that which is not produced nor consumed. Reasonable use of energy and energy conservation are extremely capacious and diverse areas of interest, encompassing both the supply and the demand sides. Energy conservation deserves the highest priority, but it cannot be the only energy story: one has to first care about energy generation before there is something to think about conservation. Also another predicament, the limited supply of fossil hydrocarbon resources and the negative impact of greenhouse gases emissions on the global environment petitions a rational solution. Inevitably, renewable sources of energy are promising options for the longer run. Among all renewable energies, Sun is the amaranthine source of renewable energy and harbinger of one of the efficient technologies of solar power: CSP or concentrating solar power systems; these plants harness solar thermal power usually by employing the concentration of solar radiation via mirrors to produce temperatures of 300°C to 500°C, depending on the type of plant, in a circulating fluid. The thermal energy transfered to this fluid is then used to produce electricity through conventional thermodynamic cycle.

CSP is one of the most cost-effective renewable electricity technologies since its supply is not restricted if the energy generated is transported from the world's solar belt to the population centres. These technologies have reached a certain maturity, as has been demonstrated in pilot projects in different countries Germany, Spain, South Africa and the USA. Today there are many large tower plants are already operating such as in the US (Ivanpah 1-3 and Crescent Dunes), three in Spain (PS10, PS20 and Gemasolar). Numerous small-scale plants exist around the world for demonstration and research purposes (e.g. the Solarturm Jülich in Germany, CESA-1 and SSTS-CRS in Spain, CSIRO solar tower at the Energy Centre in Australia, National Solar Thermal Power Plant, designed and commissioned at Gurgaon by IIT Bombay.

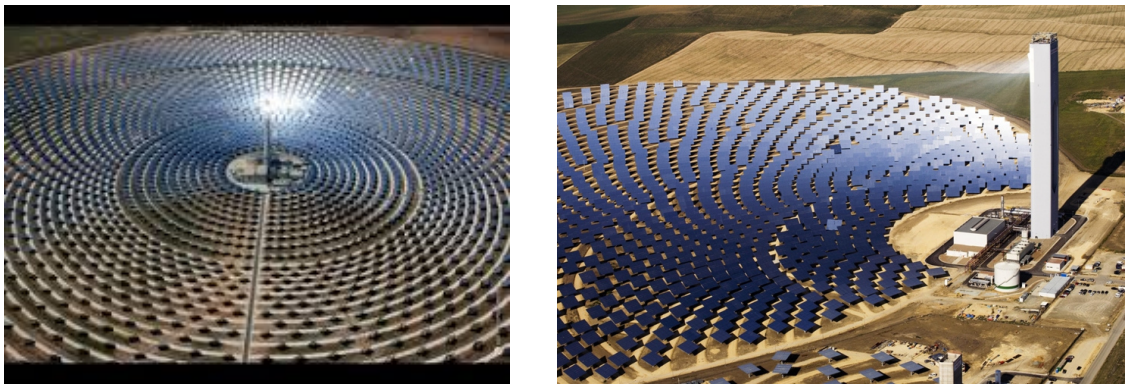


Figure 1: Gema Solar Power Tower plant in Sevilla, Spain (left) and Abengoa PS10 power plant (Seville, Spain) Soler 2010(right).

Sources: Wikipedia

The CSP plant typically has the following subsystems and components: heliostat field, tower and receiver, heat transport system, thermal energy storage (optional), supplementary fossil fuel firing system, power conversion system, plant control and auxiliary power supply and heat rejection. Because the heliostat field usually constitutes the largest fraction of the costs of a solar central receiver plant, the CSP development program has given particular attention to development of low cost designs.

The design of the heliostat field layout is a challenging task of exceptional importance. It is the sub-system with the highest cost that amount to around 50% of the total cost and its optimal design highly depends on the specifications for each project. The placement of the mirrors may lead to individual mirrors being blocked and shaded and and major optical losses are associated with the field which all together affects the efficiency of the power plant.

Therefore, optimization of a heliostat field is an essential task to make a solar central receiver system effective. The full heliostat field layout optimization is based on the calculation of five optical performance parameters: the mirror or the heliostat reflectivity, the cosine factor, the atmospheric attenuation factor, the shadowing and blocking factor, and the intercept factor. The main problems of full optimization of any heliostat field, with thousand of heliostat coordinates, would be the need to recalculate the shadows and blockings for all the heliostats in the field for every step of the process. In addition to the very strong computational load of these recalculations, this would also necessitate correct management of not only the changing position of any heliostat in the field, but also the location of its shading and blocking heliostats.

The performance of the heliostat field is defined in terms of the optical efficiency, defined as a ratio of the net power intercepted by the receiver to the power incident normally on the field. As heliostats are packed closer together, blocking and shadowing losses increase, but related costs for land and wiring decrease. Clearly, the heliostats should be carefully distributed in the field, so that maximum efficiency is obtained.

To select an appropriate deterministic optimization procedure we need to parametrize the plant design as a function of design variables which will formulate the problem of finding optimal designs to the numerical problem of finding the optimum value of a function of several variables. This optimization problem will be attacked with different algorithm procedures both local and global in nature to achieve an improvement in the efficiency of the field.

1.1. State-of-art

A remarkable number of studies have been done for optimizing the design of heliostat field. In this section, we details the existing codes and methods which improves the optical efficiency of heiostat field. One of the earliest studies was done by Lipps and Van't Hull, who concluded that the overall collection efficiency of staggered heliostat fields is usually higher than that of a cornfield layout. They used the ratio between total system cost to the total energy collected as figure of merit. In order to reduce the

number of calculations, individual heliostats were replaced by cells of representative heliostats. The commercial simulation tool based on this approach is UHC code also called as RCELL suite [9].

Sanchez and Romero [15] suggested the use of a growth based algorithm to design the heliostat field layout. The heliostats are added step by step on pre-defined points of the field. The algorithm terminates when the system requirements are met. The efficiency and the runtime of this algorithm highly depends on the number of the pre-defined points of the field. The first heliostat is placed in the position that yields the highest yearly energy collection, the second heliostat is placed in the next best position, considering blocking and shading by the first, and so forth until the entire field is populated [15].

A code based on the discretization of the heliostat surface into cells was suggested by Noone et al [12]. This code considered a non conventional biometric pattern for the heliostat field layout for computing the annual performance. Consequently, of using this new pattern, better annual efficiency was achieved; nonetheless, in a full optimization process, which includes a large number of heliostats, it can be time consuming due to the implementation of a discretization approach, especially if the intercept is calculated locally for each cell [12].

Collado and Guallar developed a code called Campo which considers all the heliostat positions in a full optimization process. This code was validated, using the available data on Gemasolar-a concentrated solar power plant. The validation was carried out on annual basis by varying the radial spacing manually in between the rows of the heliostats. Furthermore, the heliostats, which had low optical performance, were eliminated to obtain a desired number of heliostats and to increase the overall optical efficiency of the whole heliostat field [8].

A new technique was developed for the calculation of shadowing and blocking factor, using the Sassi procedure, which reduces the computation time in Mueen Code [7].

Wei et al [18] focused on the heliostat field for a cavity receiver, and concluded that there is a relationship between the effective field boundaries and cavity aperture geometry.

Lutchman et al [10] assigned a random location to each heliostat in a bounded field, and used a constrained classical gradient based optimization algorithm to force heliostats towards positions that maximize the field optical efficiency.The heliostat position vector for each heliostat is treated as a design variable.

Carizosa et al [3] used a greedy-based algorithm to optimize the electricity generated per unit cost for a heliostat field comprised off different sized heliostats in the same field.

Summing up there are multiple commercial field layout tools like UHC code, DELSOL/WinDELSOL, HFLCAL. Recent new tools like HFLD and Biometric have been better. The layouts generated by these codes are characterized by number and position of heliostats, the tower height and the position, size, orientation of the receiver. But, most of them optimizes specific objective function. The problem of field design is NP-hard, we can search for near-optimal solutions. In pursuit of finding optimal design; the task of this thesis would be the development of a black-box optimizer which is compatible with complex multi-objective, multi-modal and non-differentiable objective functions.

1.2. Optimization

Classical Approach:

- **Pattern based approach:** The heliostats are placed based on defined geometric patterns like rows, ellipses, radially staggered, spirals, nature inspired sunflower pattern; that can be described by a limited number of variables instead of optimizing the x-y coordinates of center of heliostats. The search domain is highly reduced from hundreds or thousands of x and y coordinates to a handful parameters. Thus, the pattern method essentially determines the best adaptation of the pattern for the problem and not necessarily the best x-y coordinates for optimal plant performance.

  A typical procedure for the field layout is to use one or multiple correlations with upto 11 or more variable parameters to define heliostat positions as well as tower height and receiver configuartions. Using these correlations, an oversized heliostat field is generated, and for each of the heliostats, the annual performance is calculated with a simplified simulation test function. From the oversized heliostat field, the best performing heliostats are selected, until a certain requirement such as design point thermal power is fulfilled. Best performance depends on the optimization goal and can mean, for example, the annual intercepted energy or the annual thermal receiver output, assuming that a receiver performance model is applied.

  A **superimposed parametric-optimization tool** controls the execution of multiple runs with different parameters, defining different configurations of the heliostat field, tower, and receiver, until an optimum is the goal to achieve. The optimization target is usually least cost of thermal energy or electricity. This procedure results in an optimized configuration with a heliostat field that is always characterized by some kind of regular pattern of the heliostat positions.

  There exists a large variety of optimization approaches which could be used for parametric optimization, such as non-linear mixed programming, gradient-based methods (which takes approximate value of gradient and hessian, doesn't actually evaluate the derivative) such as variants of Powell method [1] and Sequential Ap-

proximate Optimization(SAO) strategy [10] or the gradient free search methods like Nelder-Mead Method and coordinate-search methods.

- **Free-variable Method:** The nature-inspired genetic algorithm, differential evolution, simulated annealing and particle swarm intelligence, ant colony algorithms are global gradient free methods which follows a more classical approach to optimization by directly optimizing the x-y coordinates. Due to the complexity of the problem an appropriate heuristic is needed. This approach starts with a random layout which iteratively adjusts each x-y coordinate by following the gradient in the direction of a better function value until a certain objective is achieved. In this optimization the heliostats are not limited to a pattern, which means, that they can freely move through the field during the optimization.

Hybrid Methods Approach:

1. **Hybrid I: Pattern + Local Search Heuristics** The hybrid optimization strategy consists of a combination of two or more approaches defined above. The classical method finds a optimized pattern, The subordinated methods then should refine the solution locally, in context of that either a greedy heuristic or a linear programming algorithm might be a good choice. This strategy has shown to give better results when compared to each of the two algorithms alone. Buck [2] used a pattern based optimization method and refined the results with a greedy heuristic by perturbing each heliostat position locally. In this work, using classical methods Multi-dimensional Golden Section method and Globalized Nelder Mead method as **superimposed parametric-optimization tool** we obtain the best parameters for the field and then we perform local refinement of the pattern fields with a new approach **best-trial-neighbour heuristics**.

2. **Hybrid II: Metaheuristics + Nelder-Mead method** Another possiblity, at first a meta-heuristics can be used, which is able to search on a huge solution space and to move towards the global optimum. The subordinated methods then should refine the solution locally. In this work, a "hybrid method"(which can be extended into a multi-step optimization strategy) is developed to optimize the design parameters of field to reach to a optimal solution. This method is combination of Differential Evolution and Nelder-Mead method and is used to minimize the shading and blocking efficiency of the field which in turn maximizes the overall annual optical efficiency. Both the hybrid methods I and II are compared; IInd is better than the Ist in terms of time complexity and fast convergence.

Thus as post-processing step for Ist we use **best-trial-neighbour heuristics** and **Nelder-Mead method** for IInd and **best-trial-neighbour heuristics** can also further be used as next step in IInd hybrid method.

2. Mathematical Model

A solar field is given by N heliostats H_i, each with a diameter D_i and position coordinates (x_i, y_i) of center of heliostats. The ray-tracing model computes the received optical radiation over a year, while taking cosine effects η_{cos}, shading and blocking η_{sb}, heliostat reflectivity η_{ref}, atmospheric attenuation η_{aa} and spillage losses η_{spl} into account. For each heliostat H_i the time dependent received optical efficiency is defined by

$$\eta_{opt,i}(x, y, t, d) = \eta_{\text{cos},i} \cdot \eta_{\text{sb},i} \cdot \eta_{\text{ref},i} \cdot \eta_{\text{aa},i} \cdot \eta_{\text{spl},i} \tag{1}$$

at time t of the day d. In the next subsections a model is discussed which computes this annual received radiation of the full plant, which depends on the sunrise and sunset of every day in the year.

2.1. Heliostat Field Description

Solar Positioning Model In order to calculate the instantaneous optical efficiency of the heliostat field, it is necessary to include a solar positioning model. According to date, time and a geographical location on earth given by latitude ϕ and longitude θ, the solar position can be computed.

It is described by two angles, the solar zenith θ_s and the solar azimuth γ_s, both measured in radians [2]. In addition to the geographical location the time zones of summer and winter time, a year and of course the day and time need to be provided.

$$\delta = \frac{23.45\pi}{180} sin\left(2\pi \frac{284 + n_d}{365}\right)$$

$$\alpha_s = sin^{-1}(cos\phi\, cos\delta\, cos\omega_s + sin\phi\, sin\delta)$$

$$\gamma_s = sgn(\omega_s) cos^{-1}\left|\frac{sin\alpha_s\, sin\phi - sin\delta}{cos\alpha_s\, cos\phi}\right|$$

where δ is the solar declination angle, ω_s is the hour angle, n_d is the day of the year, ϕ is the latitude angle, α_s is the solar altitude also called elevation angle,which is complement of solar zenith angle θ_s. In the three-dimensional Cartesian coordinate system, the $\boldsymbol{x}$-axis is facing from west to east, the $\boldsymbol{y}$-axis from south to north, the $\boldsymbol{z}$-axis is vertically upwards [2].

$$\alpha_s = 90° - \theta_s \tag{2}$$

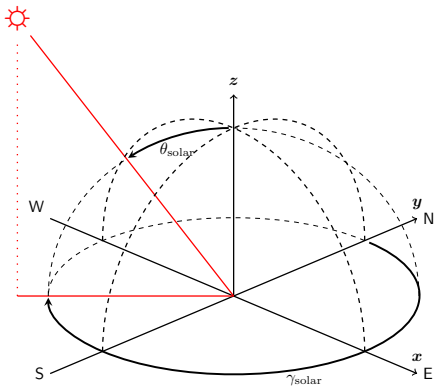
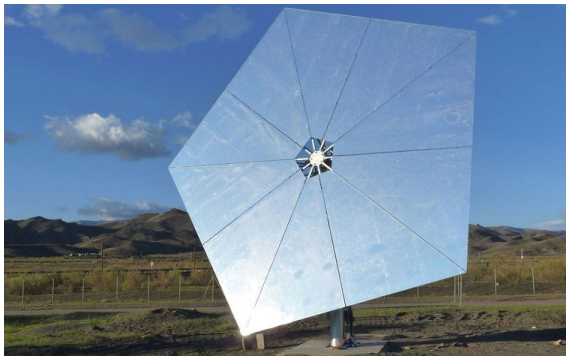


Figure 2: The solar position is given by solar zenith $\theta_s = \theta_{\text{solar}}$ and solar azimuth $\gamma_s = \gamma_{\text{solar}}$.

Heliostat geometry The heliostats are tracking the sun position, to concentrate the sun light on a central, tower-mounted receiver. Each heliostat H_i is raised on a pedestal. All heliostat are considered flat mirrors [3] of different shapes; example-rectangular with length l_i and width w_i. The heliostat's expansion D_i is the diameter of the minimum bounding sphere, $D_i = \sqrt{l_i^2 + w_i^2}$.



(a) Stellio 100

(b) Rectangular Heliostat

Figure 3: Different Geometric Heliostats

Minimal distance between heliostats For security reasons or to make sure every heliostat is accessible for cleaning, maintenance and cabling, it may be desired to have a minimal distance between two heliostats [4]. The distance is measured between the two bounding spheres of the heliostats.

Tower The tower is assumed to be a cuboid or cylinder, Each ray from a heliostat must hit the tower receiver's aim point. A subset of potentially tower-shaded heliostats can be computed by selecting just those heliostats, which are placed in a simplified

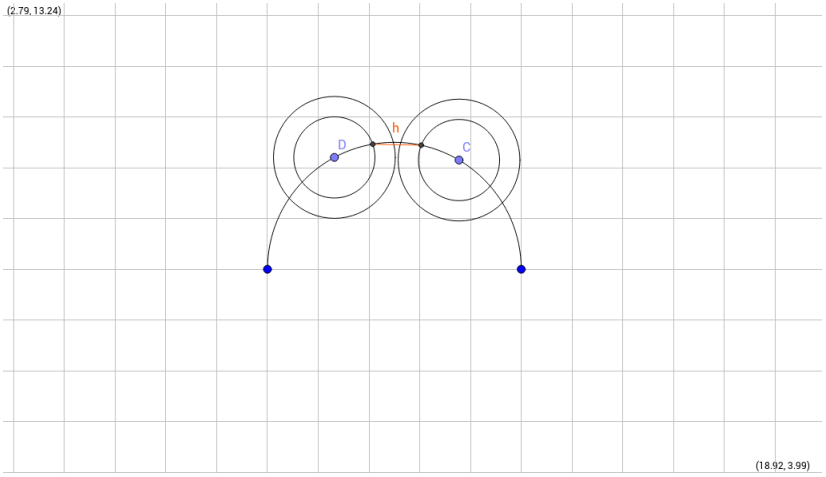


Figure 4: The safety distance(h) between two heliostats with centers D and C

shadow of the tower, called a feasible region with the width of the tower's expansion facing in opposite direction of the sun.

2.2. Optical losses and Efficiencies

- *Heliostat reflectivity:* The value of mirror reflectivity depends upon heliostat reflection rate. The mirror surface reflects the solar radiation in direction of the receiver and some radiation is scattered in a wrong direction due to slightly cleanliness or absorbance at the surface of the mirror. The reflectivity of a mirror depends on the incidence angle; not on the heliostat field configuartion or other factors related. Often a constant value of the heliostat reflectivity is considered for a particular layout.

$$\eta_{\text{ref},i} \equiv 0.88, \tag{3}$$

  This means that 88% of the energy is reflected and 12% is lost at the surface of the heliostat.

- *Cosine Efficiency:* The cosine efficiency is related to the cosine angle between the incident vectors of sunlight and normal vector of the heliostats.This is calculated using the law of specular reflection. The heliostats are tracking the sun in such a way that the rays are reflected on the surface to hit the receiver aim point. The surface of heliostats is assumed to be pefectly flat, so the normal vector of heliostat is definite. The incident rays can't be considered parallel because the sun is a disk rather than a point for any observation place on the earth. Thus it's of vital importance to generate the incident rays according to the energy distribution of the sun. Each incident ray has an angle subtended between the center of the sun to some point toward the edge called solar angle [9]. Due to the tilt of the heliostat surface, the projected area is reduced. This effect is

called cosine effect and is numbered as $\eta_{\cos}$. Cosine effects depend on the solar position and the location of the individual heliostat relative to the receiver. The heliostat surface normal bisects the angle between the solar rays and a line from the heliostat to the tower. The effective reflection area of the heliostat is reduced by the cosine of one-half of this angle. The scalar dot product of the solar vector pointing towards sun d_{sun} and the unit normal n_i to the heliostat surface is cosine efficiency of heliostat H_i.

$$\eta_{\cos,i} = d_{\text{sun}} \cdot n_i \tag{4}$$

- *Shading and blocking:* The shadowing and blocking factor is defined as the fraction of the area of the heliostat that is free of shadowing and blocking. In the algorithm explained by [16], the projections of the edge points of the affecting heliostats are used on the reference heliostat. For simplification, the heliostats can be considered as parallel planes. This approach implies that it is only necessary to project the center points of the affecting heliostats. The heliostat is then divided into n narrow vertical stripes and the height of the projection of the possible blocking or shadowing heliostat in each stripe is evaluated and so the mirror area that is free of blocking or shadowing is calculated based on these heights. During the process, the height is continuously updated for each stripe if a higher value has been found.

  Another procedure for calculation of shading and blocking was given by Collado and Turegano[5]. For each heliostat, a number of rays determine if a region of a heliostat is blocked or shaded by neighbouring heliostats and the tower. Therefore the neighbouring heliostats positions have to be selected to reduce shading and blocking effects. This is the most expensive part of a simulation.

- *Interception efficiency:* The intercept factor is defined as the fraction of the reflected rays intercepted by the receiver. The interception efficiency (or often called spillage losses) of a ray is described by a two-dimensional integral of the standard normal distribution,

$$\eta_{\text{spl},i} = \frac{1}{2\pi\sigma_{\text{total}}^2} \iint_{\Omega} \exp\left(-\frac{x^2+y^2}{2\sigma_{\text{total}}^2}\right) \, \mathrm{d}x \, \mathrm{d}y. \tag{5}$$

  where σ_{total} is the total standard deviation on the receiver plane, which is a result of convolution of four error functions, sun shape error σ_{sun}(standard deviation sun), beam quality error σ_{bq} , astigmatic error σ_{ast} , and tracking error σ_{track} . The total standard deviation was defined as

$$\sigma_{\text{total}} = \sqrt{d_i^2(\sigma_{\text{sun}}^2 + \sigma_{\text{bq}}^2 + \sigma_{\text{ast}}^2 + \sigma_{\text{track}}^2)}$$

- *Atmospheric attenuation efficiency:*

The atmospheric attenuation efficiency is defiend as the effect, that some of the energy of the reflected rays are scattered and absorbed by the the atmosphere different constituents which results in progressively less light. This radiation loss depends on the distance d_i between the heliostat H_i and the receiver aim point, in case of certain weather conditions, the farther the distance is, the smaller the value of $\eta_{\text{aa},i}$ [17].

$$d_i = |p_i - p_{\text{rec},i}|. \tag{6}$$

where p_i is the vector of center position of heliostat and $p_{\text{rec},i}$ is vector corresponding to receiver aim point of tower. With the goal to agree well with the model of Pitman and Vant-Hull for a visual range of about 40 km:

$$\eta_{\text{aa},i} = \begin{cases} 0.99321 - 1.176 \cdot 10^{-4}\ d_i + 1.97 \cdot 10^{-8}\ d_i^2 & ,\ d_i \leq 1000\,\text{m} \\ \exp(-1.106 \cdot 10^{-4}\ d_i) & ,\ d_i > 1000\,\text{m} \end{cases}. \tag{7}$$

2.3. Annual energy production

The annual received optical efficiency of the whole heliostat field is given by the sum of the annual received optical radiation of all heliostats H_i,

$$\eta_{\text{year}} = \sum_{i=1}^{N} \eta_{i,\text{year}} = \sum_{i=1}^{N} \sum_{d=1}^{365} \left(\int_{\text{sunrise}}^{\text{sunset}} \eta_{opt,i}\ \text{d}t \right), \tag{8}$$

with optical power $\eta_{opt,i}$ given in equation (1). The sunrise and the sunset depend on the day d. The value of the received optical radiation over a year η_{year}, is the basis for each objective function in the optimization process. For each different configuration of the solar field, this value has to be computed by a simulation. The following equation is used for calculating the daily averaged annual optical efficiency of the solar heliostat field.

$$\eta_{\text{average,year}} = \frac{\sum_{i=1}^{N} \sum_{d=1}^{365} \left(\int_{\text{sunrise}}^{\text{sunset}} \eta_{opt,i}\ \text{d}t \right)}{\sum_{d=1}^{365} \left(\int_{\text{sunrise}}^{\text{sunset}} \text{d}t \right)} \tag{9}$$

2.4. Parameters based Objective Function

To get an averaged annual optical efficiency η_{average} which depends on multiple parameters which are defined in optical power $\eta_{opt,i}$ given in equation (1); makes our problem a nonconvex nonlinear optimization problem. Each heliostat layout has its own defining variables and parameters corresponding to heliostats dimensions, tower, sun angles, terrain slope, area restrictions. It is difficult to optimize all these parameters together since the objective function may be complex. We write our objective function in general form

$$\eta_{average,year}(V, P)$$

where V is the set of optimization variables and P is the set of design parameters and we restrict our work to more simplified objective function assuming other variables like heliostat and tower dimensions, terrain lattitude and area as constant.

$$\eta_{average,year}(P)$$

which depends only on the solar heliostat field layout design parameters and heliostat positions. In the next section, we extract design parameters of different field layouts and the ranges for each paramters.

2.5. Black-Box Optimization

A function for which analytic form is not known is called a black-box(BB) function. Typically a BB function can be evaluated to obtain value, definiteness, (approximate) gradient. A BB function is not necessarily a complex one, it might be smooth and defined every where and even convex. The cost is the measure of the resources needed to evaluate the function.

- cheap function: it can be evaluated thousand of times
- costly function: it can be evaluated only few times(typically $<<$ 200)

For cheap black-box functions we can sample the feasible set: randomly or with experimental design or with deterministic procedure. We may want to sample around an available point (intensification) or just everywhere on the feasible set (exploration). Since we have non-linear problem to optimize we can use direct search methods or steepest descent, Quasi-Newton methods (BFGS) or implicit filtering methods, and thus large amount of function evaluations allows the gradient approximation. There are better methods like Nelder-Mead Simplex search, feasible set sequential partition, and metaheuristics which are easy to implement and to parallelize with almost no convergence theory and in general quite low performance. To balance the global/local phases,we use a two-phase approach:

- use a method algorithm to generate a new set of points (exploration).
- start local searches from some of them.

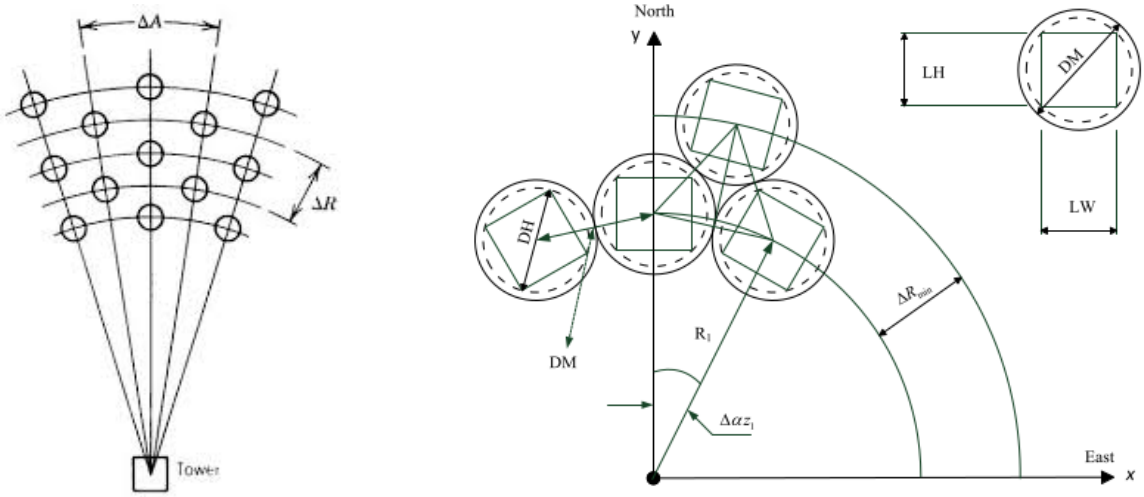
For costly functions we need to minimize the function evaluations to achieve the target. This can be done by carefully selecting the domain space. In this work, we defined a surrogate model to the practical problem and consiering the BB functions as cheap, we implement the two hybrid methods. The objective functions depend upon the coordinates of centers of heliostats, so we take simple simulations functions as objective function.

3. Heliostat Field Generation

Different field layout generation procedure were studied and for each of them optimization parameters with their ranges were determined which helps our objective function to be optimized in specific ranges of parameters.

3.1. Radial-Staggered Layout

The radial staggered configuration is a layout in which the heliostats are located around a tower in rings. The heliostats of a ring are placed with an azimuth angular spacing and no heliostat is in front of other heliostat of an adjacent row. Such a configuration allows for rays to pass in between heliostats located on neighboring rings. Figure [5a] demonstrates a caption of radially staggered heliostat configuration where each heliostat can be defined in $2D$ space by radial spacing ΔR and azimuth spacing $\Delta A(\Delta az$ in figure [5b]). The two different methods to optimize shading and blocking factor by Campo Code [8] and Mueen Code [7] are discussed here.



(a) Radial-Staggered Rows (b) DM-Characteristic Diameter

Figure 5: Radial-Staggered Configuration

Table 1: Common Paramters for all layouts

Input Parameters for Implementation	
Characteristic Heliostat Diameter	5m
Height of Tower	50m
Lattitude	36°

Dense Radial Staggered The most simplified version of this layout considers the spacing between the rings of a zone constant. The layout will be made of circular concentric rows of heliostats. The heliostats are placed as close as possible satisfying the minimum safety distance criteria. This layout theoretically signifies the hexagonal close dense packing of circles on 2D in a concentric circular fashion with appropriate .

In order to reduce the shadowing and blocking losses, and to keep the minimum distance for mechanical constraints, parameter d_{sep} is proposed to define the minimum distance between the heliostats. For this algorithm, the first step is the evaluation of the parameters that define each of the zones in the field which are azimuth spacing, the radius of the first row in the zone, number of heliostats per row and the number of rows in the zone. The following steps presents the implementation of dense radial staggered field which are followed from [8]. The characteristic diameter in Figure [22b] is DM the addition of distance between the center of the adjacent heliostats and seperation distance, the equations used to calculate it are

$$DH = (\sqrt{1+f^2}) \cdot LH \tag{10}$$

$$DM = (\sqrt{1+f^2} + ds) \cdot LH \tag{11}$$

The equation (11) can be re-written as

$$DM = DH + d_{\text{sep}} \tag{12}$$

$$d_{\text{sep}} = ds \times LH = (d_{\text{safety}} + dA)LH \tag{13}$$

where DM is the characteristic diameter(We will use notation D for the charateristic diameter DM), DH is the heliostat diagonal and d_{sep} is any additional security distance between the heliostats, which is equal to $ds \times LH$; d_{sep} is the safety distance and dA is the additional azimuthal seperation both are in unit of D. f is the ratio of the width to the height of the heliostat, LH is the height of the heliostat, LW is the width of the heliostat. In the equation (13); ds is ratio of heliostat separation distance to heliostat length. For no blocking, the minimum value of the ratio of separation distance to heliostat length for no blocking, is given by [5] is:

$$ds_{\min} = 2f - \sqrt{1+f^2} \tag{14}$$

for sqaure heliostats: $f = 1$, we get $ds_{\min} = 0.5858$

Keeping $\Delta R > 0$, it is necessary that

$$f_b > 1 - [\frac{2f - (\sqrt{1+f^2}ds)}{f}] \tag{15}$$

where f_b is the blockimg factor which is practically assumed to be annual avarage value, so constant in dense radial field. Here we also take dA annual azimuthal distance between two heliostats to be constant as well.

Campo Code The layout configuration proceeds from densest fields, with the worst shadowing and blocking factor, towards expanded fields with decrement in shading and blocking. The algorithm proposes a radial expansion of the dense radial staggered field. In first step calculation of the layout parameters that define each of the zones in the dense field is done and then for each heliostat a radial expansion is calculated using a blocking factor and that is used to calculate how much the radial distance increases. The expression for radial expansion is given by [16].

$$\Delta R = \left[\left(\frac{\cos \omega(x,y)}{\cos \varepsilon_T(x,y)} \right) \left(1 - \frac{(1 - f_b wr)}{2wr - (\sqrt{1 + wr^2} + ds)} \right) \right] D \tag{16}$$

which is dependent on blocking factor and heliostat postions

$$\Delta R = \Delta R(f_b, x_i, y_i) \tag{17}$$

The algorithm of Campo code is given by:

1. The angular azimuth spacing should be kept constant between contiguous heliostats in the same row throughout each zone in which the field is divided. This azimuth spacing is regularly decreased in passing to an outer zone. This distance is set equal to the minimum radial increment allowed by mechanical constraints $\Delta R_{\min}$. which is practically the height of an equilateral triangle formed by joining three centers of bounding circles of three heliostats in the configuartion as shown in Figure [5b].

2. The generation procedure for the radial staggered layout starts at radius R_1 with the first heliostat placed tangent to the right of the axis perpendicular to axis of tower; the second one with the same R_1 is placed clockwise at an azimuthal distance ΔA_1 from the first heliostat. Both R_1 and ΔA_1 will be a function of the number of heliostats of the first row N_{hel_1} .

$$N_{\text{hel}_1} = \frac{2\pi}{\Delta A_1} = \frac{2\pi R_1}{h} \tag{18}$$

$$R_1 = \frac{h N_{\text{hel}_1}}{2\pi} \tag{19}$$

3. Due to radial staggered configuration, the length of the azimuthal spacing between adjacent heliostats will progressively grow with the radius of the row. Any zone would then be complete when we could place an extra heliostat between two adjoining heliostats in the same row. Thus, the azimuth anular spacing of the second zone ΔA_2 between adjacent heliostats in its first row should be

$$\Delta A_2 = \frac{\Delta A_1}{2} = \frac{h}{R_2} \tag{20}$$

$$R_2 = 2(\frac{h}{\Delta A_1}) = 2R_1 \tag{21}$$

4. As the radial increment between the consecutive rows is kept constant throughout the field, the number of the rows in each zone N_{row} can be derived. For zone one, the number of rows are

$$N_{\text{row}_1} = \frac{(R_2 - R_1)}{\Delta R_{\min}} = \frac{R_1}{\Delta R_{\min}} \tag{22}$$

5. general formulae for each zone

$$\Delta A_n = \frac{\Delta A_1}{2^{n-1}} \tag{23}$$

$$R_n = 2^{n-1} R_1 \tag{24}$$

$$N_{\text{row}_n} = 2^{n-1} \frac{R_1}{\Delta R_{\min}} \tag{25}$$

6. For radial expansion, radius for next row is calculated by adding $\Delta R(i)_{\text{next}}$ to the coordinates of first row. d_{sep} is responsible for azimuthal expansion (13).

$$\Delta R(i)_{\text{next}} = max\big(R_{\min}(d_{\text{sep}}), \Delta R(f_b, x_i, y_i)\big) \tag{26}$$

Campo code presents a simplified version of this radial-staggered algorithm that considers the spacing between the rings of a zone constant. In order to reduce the shadowing and blocking losses, and to keep the minimum distance for mechanical constraints, this algorithm proposes parameter d_{sep} or d_{safety} and ΔR. The range of optimizing parmeter d_{sep} is between $[0, D]$. The range for radial expansion difference lies in interval $[0, 1]$.

The annual average density of mirror [5] will be given as

$$\delta = \frac{\text{reflecting surface}}{\text{level terrain covered}} = \frac{f f_a L H^2}{D \Delta R} \tag{27}$$

Putting the expression for ΔR in above equation, it is clear that maximum density will agree with the least value of blocking factor; for CESA-1,[5] it was 1.778 so that

$$\delta = 0.3286 \frac{m^2\text{mirror}}{m^2\text{level terrain}} \tag{28}$$

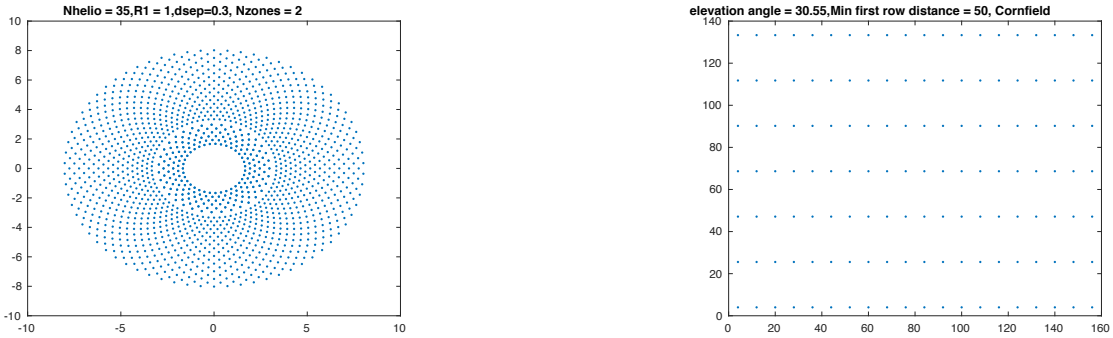
Thus, only 30% of the land area was used, with all parameters values taken as optimized.

The optimization problem formulation can be written as

$$\begin{aligned} &\text{maximize} \quad \eta_{year}(\Delta R_{\text{expansion-difference}}, d_{\text{sep}}, d_{\text{first row}}) \\ &\text{where} \\ &\qquad R_{\text{expansion-difference}} \in [0, 1] \\ &\qquad d_{\text{sep}} \in [0, D] \\ &\qquad d_{\text{first row}} \in [0.5 H_{tower}, 1.5 H_{tower}] \end{aligned}$$

Table 2: Campo Code

Input Parameters for Implementation	
Diameter of Heliostat	5m
Safety Distance	0.3m
Ist row radius	H_{tower}
Field Size	$\pi({N^{th}_{\text{zone-radius}}}^2 - H^2_{\text{tower}})$
Number of Zones	2
Number of Heliostats in First Zone	35



(a) Radial-Staggered Field (b) Row-Pattern Cornfield

Figure 6: Test- Fields generated with initial parameters

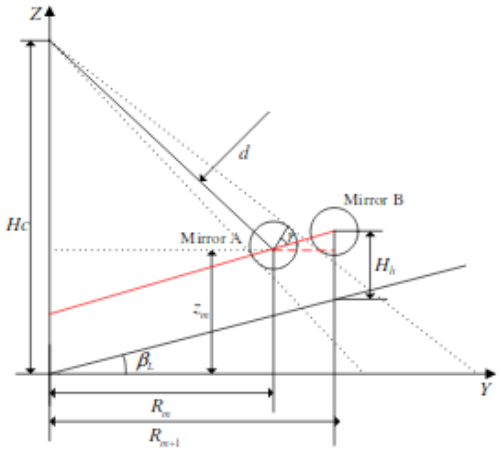


Figure 7: Scheme used in the graphical method to calculate new row radius:Resources:[7]

Mueen Code The algorithm here discusses a methodology to create a radial staggered configuration field using a graphical method that avoids blocking losses between heliostats. The field is also divided in zones in order to increase the land use efficiency. The methodology defines the radius for a new row selecting the radius that maximizes the land use. For this purpose, the algorithm has to decide whether to add a new row to the current zone or start a new zone when a new ring is needed. In order to add a new row in the zone the radius is trigonometrically calculated. To calculate this radius a line is drawn from the receiver centre point and tangential to the heliostat in the new row and the previous row as can be seen in the Figure [7]

While the method assures that neighboring mirrors block none of the energy reflected by a heliostat, it does not take into consideration the possibility of shading part, or all, of the reflecting surface. This is based on the experience that blocking has a more pronounced effect on the layout of the heliostat field.

This arrangement also ensures that no heliostat is placed directly in front of another heliostat in adjacent rings along a spoke to the tower. Here some mathematical expressions are given followed form [7].

- Essential Ring: The rings that have a heliostat on the north axis in the field. Staggered Ring: The rings that have no heliostat on the north axis in the field. Here, the radius of the first ring R_0, by definition an essential ring, is usually given in terms of the reciever aim point height on tower H_{tower}.

- Minimum radius is achieved with restriction of collisons between adjacent heliostats

$$\Delta R_{\min} = R_{m+1,\min} - R_{\min} = D \times cos30^\circ \times cos\beta_L \tag{29}$$

where β_L is the tilt angle of the field and L is set to 0. The maximum radius is determined according the principle of no blocking between the heliostats.

$$z_m = R_m tan\beta_L + H_h$$
$$d = \sqrt{{R_m}^2 + (H_{\text{tower}} - z_m)^2}$$
$$\gamma = sin^{-1}(\frac{D}{2d}) + sin^{-1}(\frac{R_m}{d}) - \beta_L$$
$$\Delta R_{\max} = R_{m+1,\max} - R_m = D\frac{cos\beta_L}{cos\gamma}$$

where H_h is the height of the heliostat mirror, and H_{tower} is the height of the tower. So the the radius of each ring can be expressed as:

$$R_{m+1} = R_m + \Delta R_{\min} + R_{\text{difference}}(\Delta R_{\max} - \Delta R_{\min}) \tag{30}$$

where $R_{\text{difference}}$ is the optimized parameter of the radius, $0 < R_{\text{difference}} < 1$.

- In the paper [7], the angular unit of each ring is assumed to be the same. The minimum value of the angular direction unit is given by:

$$\Delta A_{\min} = sin^{-1}(\frac{D/2}{R_m}) \tag{31}$$

The maximum value of the angular direction is to avoid blocking. So it is given by: For the first ring:

$$\Delta A_{\max} = sin^{-1}(\frac{D/2}{R_m}) + sin^{-1}(\frac{D/2}{R_{m+1}}) \tag{32}$$

For second ring:

$$\Delta A_{\max} = sin^{-1}(\frac{D/2}{R_m}) + sin^{-1}(\frac{D/2}{R_{m-1}}) \tag{33}$$

So the angular direction unit can be expressed as:

$$A_m = \Delta A_{\min} + A_{\text{difference}}(\Delta A_{\max} - \Delta A_{\min}) \tag{34}$$

where $A_{\text{difference}}$ is the optimized coefficient of the radius, $0 < A_{\text{difference}} < 1$. Thus the angle between the north axis and the distribution axes can be given by:

$$\phi_m = \pm nA_m \tag{35}$$

where $n = 0, 2, 4.....$ is for essential rings, $n = 1, 3, 5.....$ is for staggered rings. The optimal field layout with maximum value of the product of the ground coverage and annual optical efficiency can be obtained through the search of above decision variables. A heliostat is located in the field by defining the coordinates

of its center. These are known once the angular direction of the heliostat and the radius of the ring to which it belongs are fixed. The location of heliostats are given by:

$$\begin{cases} x = R_m sin\phi_m \\ y = -R_m cos\phi_m \\ z = z_m \end{cases}$$

The optimization problem formulation can be written as

$$\begin{aligned} &\text{maximize} \quad \eta_{year}(R_{\text{difference}}, A_{\text{difference}}, d_{\text{first row}}) \\ &\text{where} \\ &\qquad R_{\text{difference}} \in [0, 1] \\ &\qquad A_{\text{difference}} \in [0, 1] \\ &\qquad d_{\text{first row}} \in [0.5H_{tower}, 1.5H_{tower}] \end{aligned}$$

3.2. N-S Cornfield Layout

The approach is to utilize a tightly packed North-South cornfield configuration in an effort to pack as many heliostats as possible into the utilizable region of the tower for the primary aimpoint of the heliostat field. The utilizable region of a tower is defined as the region where heliostats can be placed so that tower can recieve reflected sun rays from heliostats. This heliostat layout ensures that a heliostat will not block a parallel vertical heliostat on a subsequent row if the heliostats, tower, and sun source are all co-linear. For any given row, heliostats are compacted side by side, as close as physically possible without obstructing the range of motion of an adjacent heliostat.

To allocate the row spacing for the field, the lowest position of the sun in the sky at solar noon for the geographical location over the course of the year is required. The elevation or solar altitude angle, α_s, is defined by the angle between the horizontal and a line to the sun. The elevation angle is a complement of the solar zenith angle θ_s, and therefore can be calculated according to the equation

$$\alpha_s = 90^\circ - \theta_s$$

At solar noon, in the Northern Hemisphere the elevation angle can be represented by the simplified equation

$$\alpha_s = 90^\circ - \theta_s + \delta_s \tag{36}$$

The solar declination angle δ_s ranges from 23.45° on the summer solstice, June 21, to -23.45° on the winter solstice, December 22. Using above equation 36 for test simulation model values(say 36°) on the winter solstice when the sun is at its lowest elevation yields the following minimum elevation angle, α_{min}, at solar noon.

$$\alpha_{\text{min}} = 90^\circ - 36^\circ - 23.45^\circ = 30.55^\circ$$

Table 3: Cornfield Code

Input Parameters for Implementation	
Field Size	H_{tower}^2
Minimum Elevation angle	$30.55°$

With the minimum elevation angle at solar noon specified, the row spacing, R_{spacing} ,is defined as ratio of characteristic diameter D of the heliostat and sine of the elevation angle, according to the following equation

$$R_{\text{spacing}} = \frac{D}{sin(\alpha_{\min})} \simeq \frac{D}{tan(\alpha_{\min})}$$

With the row spacing allocated, the heliostats are placed in the first row and for subsequent rows, having difference as defined by R_{spacing} , up to the distance of the maximum row. The width of each row is initially defined by laying down a square grid of heliostats, as determined by the dimensions of the region. The lower bound for the row spacing is equal to characteristic diameter D.

The column spacing between the mirrors is determined such that no two heliostats block each other and do not collide with each other. The lower bound of the search range of column space will be D and the upper bound is determined with no blocking condition [7] which can be taken as $2D$. Another input parameter is minimum feasible distance $d_{\text{first row}}$ from the tower to the first row of heliostats which depends on receiver point height on tower. The minimum elevation angle calculated through out the year is taken which depends on lattitude and longitude as well as on solar zenith angle.

$$\begin{aligned} &\text{maximize} \quad \eta_{year}(R_{\text{spacing}}, C_{\text{spacing}}, d_{\text{first row}}) \\ &\text{where} \\ &\qquad R_{\text{spacing}} \in [D, \frac{D}{tan(\alpha_{\min})}] \\ &\qquad C_{\text{spacing}} \in [D, 2D] \\ &\qquad d_{\text{first row}} \in [0.5H_{tower}, 1.5H_{tower}] \end{aligned}$$

3.3. Biometric Layout

Biometric spirals are based on the phyllotaxis pattern, as proposed in [12]. The characterizing parameters are (a, b). The following equations are used to calculate the position coordinates(x_k, y_k) of an individual heliostat k is calculated.

$$r_k = a \cdot k^b \tag{37}$$

$$\theta_k = k \cdot \left(2 \cdot \pi \cdot \phi^{-2}\right) \tag{38}$$

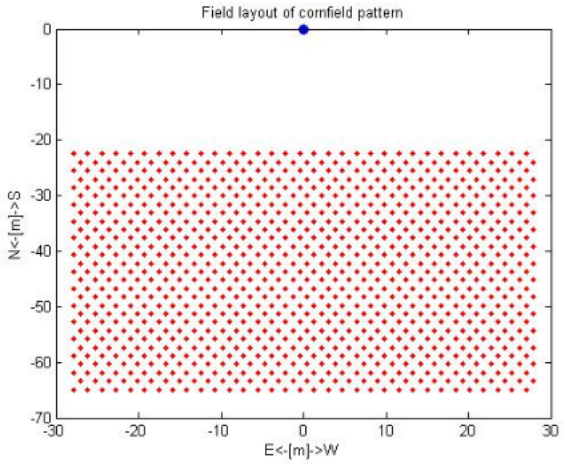


Figure 8: North-South Cornfield

$$x_k = r_k \cdot \cos(\theta_k) \tag{39}$$

$$y_k = r_k \cdot \sin(\theta_k) \tag{40}$$

where θ_k is linearly proportional to the k^{th} element of the sequence and r_k is the radial growth function. The angular component is related to golden ratio ϕ which equals $\frac{1+\sqrt{5}}{2}$. These equations determine the radial distance to the tower and the azimuth angle in the field. In this algorithm the density of heliostats between zones is more continuous than in radial staggered configurations. The value of ϕ is not optimized because small variations from the nominal produces dramatically different patterns which may be significanly suboptimal[12]. But to see how the change in irrational number affects the field layout and our objective function; we consider nearly same geometric irrational numbers silver ratio and plastic number. These numbers deviate the results ivariably with reference to golden ratio.

Table 4: Golden Ratio and its sister numbers

Golden Ratio	$\frac{1+\sqrt{5}}{2}$	1.61803398875
Silver Ratio	$1+\sqrt{2}$	2.4142135623730950488
Plastic Number	$\frac{\sqrt[3]{108+12\sqrt{69}}+\sqrt[3]{108-12\sqrt{69}}}{6}$	1.32471795724474602596

The essence of natural pattern is that it forgives the minimum bounding distance criteria between neighbouring heliostats. That's why here we don't need to optimize minimum bounding distance-security distance criteria which was given by d_{sep} or d_{safety} in radial-staggered layouts. The number of heliostats and dimensions of each heliostat will be fixed. So, we optimize two parameters azimuthal Expansion-Contraction control parameter a. related to radial component of spiral, b is the mean distance between neighboring heliostats. The heliostat pattern density as a function of distance from the receiver can be varied by varying b. The density fraction of this layout is given by

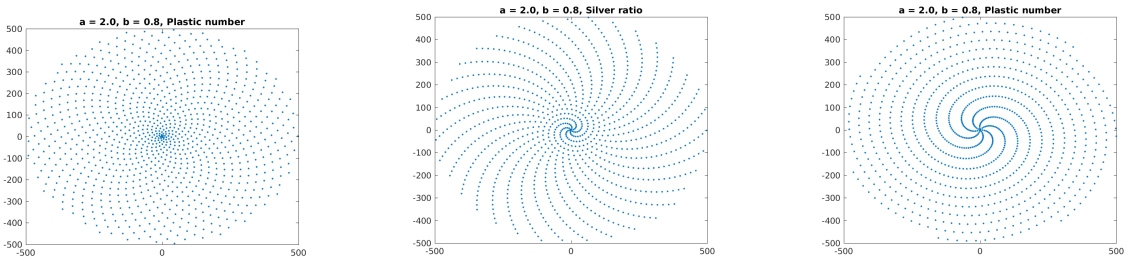
$$\delta = \frac{D \times D}{a \times b}$$

Table 5: Geometric Spirals

Input Paramteres for Implementation	
Irrational Numbers	Refer table[4]
Number of Heliostats	100,500,1000

where D is the characteristic diameter. Based on the observations in [12] the range for a and b are $[2, 8]$ and $[0.5, 0.7]$ respectively. Non-linear constrained optimization problem can be formulated as

$$\begin{aligned} &\text{minimize} \quad \eta_{year}(a, b, \phi) \\ &\text{where} \\ &\qquad 2 \leq a \leq 8 \\ &\qquad 0.5 \leq b \leq 0.7 \end{aligned}$$



(a) Golden Spiral (b) Silver Spiral (c) Plastic Spiral

Figure 9: Effect of changing irrational number with a = 2, b = 0.8

3.4. Basic Patterns Layout

Concentric regular patterns The two dimensional concentric regular patterns like row-pattern cornfield[6], circles, ellipses, squares[10], rectangles could be the first choice for placements of mirrors in a CRS plant. Let's take the case of rectangles, the user can limit the maximum number of the concentric rectangles as well as the spacing between rows of rectangles. The optimization parameters are the distance between two heliostats in a rectangle a and expansion factor b ,the distance between two consecutive rectangles. Similarly for the circles, ellipses and squares patterns, the heliostats are moved freely on the their regular boundaries to achieve optimal configuration. The equations below show the basic computation of a position on an ellipse.

$$\theta = 360 \cdot \frac{spacing}{circumference} \tag{41}$$

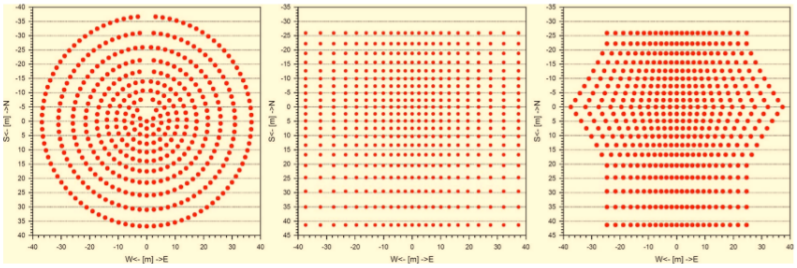


Figure 10: Basic Patterns:Resources:[14]

$$x = a \cdot \cos\left(\theta \cdot \frac{\pi}{180}\right) \tag{42}$$

$$y = b \cdot \sin\left(\theta \cdot \frac{\pi}{180}\right) \tag{43}$$

The problem formulation would be:

$$\begin{aligned} &\text{maximize} && \eta_{year}(a, b) \\ &\text{where} \\ & && a_{\min} \leq a \leq a_{\max} \\ & && b_{\min} \leq b \leq b_{\max} \end{aligned}$$

3.5. Comparison

Ground coverage or field density can be defined as the total area of heliostats divided by the field area of given configuartion represents the local density of the heliostats. The results of ground coverage for existing research of different patterns and theoretical packing density which is the maximum possible density of ground usage by mirror heliostats based on the circle-close packing configuaration are given in table [6].

In two dimensional Euclidean space,the highest-density lattice arrangement of circles is the hexagonal packing arrangement, in which the centres of the circles are arranged in a hexagonal lattice (staggered rows, like a honeycomb)[8], and each circle is surrounded by 6 other circles. The density fraction of this arrangement is

$$d = \frac{\pi}{2\sqrt{3}} = 0.9069$$

.

The square tiling can be used as a circle(locus of heliostat spinning on its axis) packing for North-South Cornfield and basic patterns, placing equal diameter circles at the center of every point. Every circle is in contact with 4 other circles in the packing which is known as kissing number. The packing density fraction is

$$d = \frac{\pi}{4} = 0.7854$$

Table 6: Ground Coverage of Fields

Layout	Practically Acheived	Maximum Possible
North-South & Basic Pattern	35-38%	78.54%
Radial-Staggered Configuration	30-35%	90.69%
Biometric Spiral Pattern	35-40%	$f(\frac{D^2}{ab}, \phi)$

Table 7: Objective Values of different layouts

Initial Field Configuartion		
Layout Type	Paramters	Practical Annual Efficiency)
Dense Radial-Staggered	$[0.5, \frac{D}{2}, H_{\text{tower}}]$	75%
NS- Cornfield	$[\frac{3D}{2}, \frac{3D}{2}, H_{\text{tower}}]$	86.3%
Row- Cornfield	Figure[6]	43.3%
Campo Code	Figure[6]	82.5%
Mueen Code	$[0.5, 0.5, H_{\text{tower}}]$	84.45%
Golden Spiral	Figure[9]	85.65%
Silver Spiral	Figure[9]	85.65%
Plastic Spiral	Figure[9]	85.65%

This is evident as the density of field increases the shading and blocking factor would increase and in result decease the annual total optical efficiency of the field. To achieve an overall improvement in annual performance, there should be trade-off between the ground coverage and the optical efficiency as depicted in Pareto Curve in Figure 7(b) of paper [12]. Thus, maximizing the product of annual optical efficency and the ground coverage would be the best choice for the objective function for optimization. To simplify the problem; we consider here simulation dummy function for the shading and blocking efficiency and rest of the other factors which contributes to the objective cost are not conidered.

$$f_1((x_1, y_1), (x_2, y_2), ...(x_n, y_n)) = \sum_{i=1}^{n} x_i^2 + y_i^2$$

Since the coordinates in the field design depends on the parameters, taking objective function as f_1 is justified. The simulation test-optimization problem can be written as:

$$\begin{aligned} &\text{minimize} \quad \eta_{sb}(a, b) \\ &\text{where} \\ &\qquad a_{\min} \leq a \leq a_{\max} \\ &\qquad b_{\min} \leq b \leq b_{\max} \end{aligned}$$

4. Post-Processing Step

There objective functions for optimization can be convex or non-convex, but there is no explicit relationship between the parameters and objective function. So, the idea to develop a black-box optimizer which do not depends on the charateristics of the objective function explicitly during optimization is followed strictly in this work.

Pattern + Classical Approach + Local Search Approach:

4.1. Classical Approach

The geometric configuration described in previous section are used by commercial softwares like UHC-RCELL Code[9], DELSOL3(winDELSOL), HFLCAL[17]. We discussed decision variables(used-defined) and paramters defining these field layouts. All the decision variables could also be considered as parameters to optimize. A typical field layout configuartion is optimized by optimizing different combinations of these parameters. We consider those paramters whose ranges have been determined and considering others as constant user-defined input.

There exists interpolation methods, line search methods like Newton method, Secant method, Taylor methods which involves derivatives. Here, the methods which do not involve Jacobian evaluation are considered. These are elimination methods which can be used to solve problems where the interval in which the optimum is known to lie is finite. Let x_s and x_f denote, respectively, the starting and final points of the range of parameters. We call this range interval as interval of uncertainty. Analysis of one-dimensional methods is performed and Golden-Section method is extended to mult-dimensional case. A gradient-free method downhill-simplex method is also used to get the optimal parameters in the prescribed ranges for the patterns. A meta-heuristics Differential Evolution is also used for the same situation.

4.1.1. Multidimensonal Golden Section Method

Exhaustive Search The exhaustive search method consists of evaluating the objective function at a predetermined number of equally spaced points in the interval (x_s, x_f) and reducing the interval of consideration using the assumption of unimodality of objective function. In general, if the function is evaluated at n equally spaced points in the original interval of uncertainty of length $L_0 = x_f - x_s$, and if the optimum value of the function (among the n function values) turns out to be at point x_j then the final interval of uncertainty is given by

$$L_n = x_{j+1} - x_{j-1} = \frac{2}{n+1} L_0$$

The final interval of uncertainty obtainable for different number of trials in the exhaustive search method is given below.

Number of Trials	2	3	4	5	.....	n-1	n
$\frac{L_n}{L_0}$	$^2/_3$	$^2/_4$	$^2/_5$	$^2/_6$	.....	$^2/_n$	$^2/_{n+1}$

Since the function is evaluated at all n points simultaneously, this method can be called a simultaneous search method. This method is relatively inefficient compared to the sequential search methods discussed next, where the information gained from the initial trials is used in placing the subsequent experiments [13].

Dichotomous Search The exhaustive search method is a simultaneous search method in which all the experiments are conducted before any judgment is made regarding the location of the optimum point. The dichotomous search method, as well as the Fibonacci and the golden section methods discussed later, are sequential search methods in which the result of any experiment influences the location of the subsequent experiment. In the dichotomous search, two experiments are placed as close as possible at the center of the interval of uncertainty. Based on the relative values of the objective function at the two points, almost half of the interval of uncertainty is eliminated. Let the positions of the two experiments be given by

$$x_1 = \frac{L_0}{2} - \frac{\delta}{2}$$

$$x_2 = \frac{L_0}{2} + \frac{\delta}{2}$$

where δ is a small positive number chosen so that the two experiments give significantly different results. Then the new interval of uncertainty is given by $\frac{L_0}{2} + \frac{\delta}{2}$. The building block of dichotomous search consists of conducting a pair of experiments at the center of the current interval of uncertainty. The next pair of experiments is, therefore, conducted at the center of the remaining interval of uncertainty. This results in the reduction of the interval of uncertainty by nearly a factor of 2. The intervals of uncertainty at the end of different pairs of experiments are given in the following table.

Number of Experiments	2	4	6
Final Interval of Uncertainty	$\frac{(L_0+\delta)}{2}$	$\frac{1}{2}\left(\frac{L_0+\delta}{2}\right)+\frac{\delta}{2}$	$\frac{1}{2}\left(\frac{L_0+\delta}{4}+\frac{\delta}{2}\right)+\frac{\delta}{2}$

In general, the final interval of uncertainty after conducting n experiments (n even) is given by

$$L_n = \frac{L_0}{2^{\frac{n}{2}}} + \delta\left(1 - \frac{1}{2^{\frac{n}{2}}}\right)$$

Interval Halving In the interval halving method, exactly one-half of the current interval of uncertainty is deleted in every stage. It requires three experiments in the first stage and two experiments in each subsequent stage [13]. The interval of uncertainty remaining at the end of n experiments (n > 3 and odd) is given by

$$L_n = L_0\left(\frac{1}{2}\right)^{\left(\frac{n-1}{2}\right)}$$

Fibonacci Method The Fibonacci method can be used to find the minimum of a function of one variable even if the function is not continuous [13]. The initial interval of uncertainty, in which the optimum lies, has to be known. The function being optimized has to be unimodal in the initial interval of uncertainty. The exact optimum cannot be located in this method. Only an interval known as the final interval of uncertainty will be known. The final interval of uncertainty can be made as small as desired by using more computations. The number of function evaluations to be used in the search or the resolution required has to be specified beforehand. This method makes use of the sequence of Fibonacci numbers, F_n for placing the experiments. These numbers are defined by this recursive relation

$$F_0 = F_1 = 1 \tag{44}$$

$$F_n = F_{n-1} + F_{n-2}, \qquad n = 2, 3, 4.. \tag{45}$$

The interval of uncertainty remaining at the end of n experiments

$$L_n = \frac{L_0}{F_n}$$

Golden Section Method The golden section method is same as the Fibonacci method except that in the Fibonacci method the total number of experiments to be conducted has to be specified before beginning the calculation, whereas this is not required in the golden section method. In the Fibonacci method, the location of the first two experiments is determined by the total number of experiments, n. In the golden section method we start with the assumption that we are going to conduct a large number of experiments. The total number of experiments can also be decided during the computation. The intervals of uncertainty remaining at the end of different number of experiments can be computed as follows:

$$L_2 = \lim_{N\to\infty} \frac{F_{N-1}}{F_N} L_0$$

$$L_3 = \lim_{N\to\infty} \frac{F_{N-2}}{F_N} L_0 = \lim_{N\to\infty} \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} L_0$$

$$\cong \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^2 L_0$$

This result can be generalized to obtain

$$L_k = \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^{k-1} L_0$$

Using the relation

$$F_N = F_{N-1} + F_{N-2}$$

we obtain, after dividing both sides

$$\frac{F_N}{F_{N-1}} = 1 + \frac{F_{N-2}}{F_{N-1}}$$

By defining a ratio γ as

$$\gamma = \lim_{N\to\infty} \frac{F_N}{F_{N-1}}$$

we get

$$\gamma \cong \frac{1}{\gamma} + 1$$

$$\gamma^2 - \gamma - 1 = 0$$

This gives the root $\gamma = 1.618$, and hence

$$L_k = \lim_{N\to\infty} \left(\frac{1}{\gamma}\right)^{k-1} L_0 = (0.618)^{k-1} L_0$$

the ratios $\frac{F_N}{F_{N-1}}$ and $\frac{F_{N-2}}{F_{N-1}}$ have been taken to be same for large values of N. The validity of this assumption can be seen from the following table:

Value of N	2	3	4	5	6	7	8	9	10	∞
$\frac{F_{N-1}}{F_n}$	0.5	0.667	0.6	0.625	0.6156	0.619	0.6177	0.6181	0.6184	0.618

The procedure is same as the Fibonacci method except that the location of the first two experiments is defined by

$$L_2' = \frac{F_{N-2}}{F_N} L_0 = \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} L_0 = \frac{L_0}{\gamma^2} = 0.382 L_0$$

The desired accuracy can be specified to stop the procedure.

Analysis The efficiency of an elimination method can be measured in terms of the ratio of the final and the initial intervals of uncertainty, $\frac{L_n}{L_0}$. The values of this ratio achieved in various methods for a specified number of experiments or for a specified ratio number of experiments are calculated. To achieve any specified accuracy, the Fibonacci method requires the least number of experiments, followed by the golden section method as well as in reducing the interval of uncertainty [13]. In all these methods function has to be unimodal; but the Golden Section(GS) method is fastest one except for Fibonacci Search; but in later method number of function evaluations should be known before experiment. GS method is not bonded by this restriction. Also, the main characteristic of one-dimensional GS is to save one function evaluation per iteration except for the first one. In one-dimensional case, approximately one half of computational cost will be saved.

2-D Golden Section Method-Extended Assuming that the objective function is denoted as $f(x, y)$, intervals of uncertainty are $[a, b]$ and $[c, d]$, respectively. Based on the same principle of one-dimesion, in 2-D case only three function evaluations are needed at each stage of iterations after the first iteration and one function evaluation will be saved per iteration. Simultaneously, the region of uncertainty or search space is reduced by a factor of γ^2. For 2-D GS, a saving of one fourth of computational cost of function evaluations is earned, not one half like the case of one-dimensional GS [4]. Under n-dimensional case, a saving of function evaluations is not one half,but $\frac{1}{2^n}$, where n is the dimension.Similarly, the Golden-Section Method can be generalized to multi-dimensional case; the limitations and benefits are discussed in [4].

Implementation One of the most important feature of GS method is, the number of iterations to achieve a prescribed tolerance can be established before the iterations start. The upper bound of the function is evaluated before and then passed to multi-dimensional GS method. A generic approach of search direction is also considered to initialize the iterations. So, both the functions need current position vector or design vector x and s perscribed direction vector. Although, the efficiency of 2D-GS is not better than in case of 1D-GS, yet this algorithm has potential for as gradient-free evaluations of the objective function which is unimodal in nature.

4.1.2. Globalized Bounded Nelder-Mead Algorithm

Nelder-Mead Method This method also called downhill-simplex method is a direct search method for minimizing unconstrained real functions. The basic idea in the simplex method is to compare the values of the objective function at the $n+1$ vertices of a general simplex and move the simplex gradually toward the optimum point during the iterative process. The following equations can be used to generate the vertices of a regular simplex function values at the $n+1$ vertices x_i of a simplex. A simplex of size a is initialized at x_0 based on the rule

$$x_i = x_0 + pe_i + \sum_{\substack{k=1 \\ k \neq i}}^{n} qe_k \tag{46}$$

where e_i are the unit base vectors and

$$p = \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} + n - 1 \right) \tag{47}$$

$$q = \frac{a}{n\sqrt{2}} \left(\sqrt{n-1} - 1 \right) \tag{48}$$

The simplex vertices are updated through reflection, expansion, and contraction operations in order to find an improving point. The algorithm terminates when the vertices function values become similar, which is measured with the inequality,

$$\sqrt{\sum_{i=1}^{n+1} \frac{(f_i - \bar{f})^2}{n}} < \epsilon, \bar{f} = \frac{1}{n+1} \sum_{i=1}^{n+1} f_i \tag{49}$$

where ϵ is a small positive scalar. The cumulative effect of the operations on the simplex is, very similar, to stretch the shape along the descent directions, and to zoom around local optima. There are two limitations of the algorithm. First, algorithm may fail to converge to a local optimum, which happens in particular when the simplex collapses into a subspace. Second,the method may escape a region that would be a area of attraction for a pointwise descent search if the simplex were large enough. Ultimately, as the size of the simplex decreases, the algorithm becomes local.

Improvements The Globalized Bounded Nelder–Mead algorithm, GBNM, is meant to be a black-box local-global approach to real constrained optimization problems. A restart procedure that uses an adaptive probability density keeps a mem- ory of past local searches. Constraints and limits on variables are taken into account through adaptive penalization and projection, respectively. Finally, GBNM can be applied to discontinuous (no gradient information needed), non-convex functions, since the local searches are based on a variant of the Nelder–Mead algorithm (Nelder and Mead 1965). Improvements to the Nelder– Mead algorithm consist of simplex degeneracy detection and handling through reinitialization [11].

- Local optimizers can make up a global search when repeatedly started from different points. The simplest restart methods initialize the search either from a regular grid of points, or from randomly chosen points. In the regular grid case, we need to know how many restarts will be performed to calculate the size of the mesh. In the probabilistic case, knowledge of past searches is not used, so that the same local optima may be found several times, costing vast unnecessary effort. Here we consider 2nd case, the number of restarts is unknown beforehand and the cost of each local search is unknown. A grid method cannot be applied here because discritizing the full land area would be compuatationally expensive. Also, a memory of previous local searches is kept by building a spacial probability density of starting a search [11].

- With bounded variables, the points cannot leave the domain after either the reflection or the expansion operation. These variable bounds are accounted by projection [11].

$$\text{if}(x_i < x^{\min}), x_i = x^{\min}; \tag{50}$$

$$\text{if}(x_i > x^{\max}), x_i = x^{\min}; \tag{51}$$

Implementation Minimization of simulation test function which is a scalar function is performed in the region determined by the column vectors $x^{\min}$ and $x^{\max}$ using this globalized Nelder-Mead direct search method based on geometric operations on an N-dimensional simplex. The code returns final global solution and function value at that optimal point. The algorithm parameters are: maximum probablistic or degenerated restarts 15; maximum function evaluations 2500; number of random points per restart 5; maximum iterations per restart 250; reflection coefficient $\alpha = 1$; contraction coefficient $\beta = 0.5$; expansion coefficient $\gamma = 2$; convergence tolerance $1e^{-9}$ and the gaussian

length parameter is set to 0.01, which means that one standard devaiation away from Gaussian means covers about 20% of the domain of parameters.

4.1.3. Differential Evolution

The main idea of DE is using vector differences for perturbing the vector population. This idea has been integrated in a novel recombination operator of two or more solutions and a self-referential mutation operator to direct the search toward good solutions.

- *Initialization of population:* The diversity of the initial population in DE is very important since the search is guided by the difference between vector elements. First, vectors of initial population are generated equivalent to the population size.Like any evolutionary algorithm, DE generates a randomly distributed initial population of size $k(k \geq 4)$. Each individual is a D-dimensional real vector $x_{i,j}$. Each individual is encoded as a vector of floating-point numbers. Values of various control parameters like the radial and the azimuthal spacing in radial-staggered layout, a and b in spiral case; which represent a candidate solution, are included in each of these vectors. This is accomplished by passing on random values for each parameter of solution, within the range $[x_{\min}, x_{\max}]$ defined for the corresponding parameters specified in second section. The vector x_{ij} would be:

  $$x_{i,j} = x_{\min,\mathrm{j}} + r(x_{\max,\mathrm{j}} - x_{\min,\mathrm{j}})$$

  where $r \in U[0,1]$ is uniformly distributed random number and $j \in [1, D]$ and $i = 1$ indicates initial population size.

- *Evaluating and finding the best solution:* After the formation of initial population, the objective like simulation test function of each vector is evaluated and compared. Therefore, an optimal solution is obtained, and its value is stored externally and updated by comparison with all the values in each generation.

- *Mutation operation:* For every solution x_i in the population in i^{th} population-generation, a mutant vector m_i is generated using:

  $$m_i^{(t+1)} = x_i + F \cdot (x_{\text{best}} - x_i) + F \cdot (x_{r1} - x_{r2})$$

  where x_{r1} and x_{r2} are randomly selected solution parameters from the i^{th} generation, which are different from each other. The variable $F \in [0,1]$ is the mutation factor, which is defined by the user, and it depends upon the type of problem to be optimized. This scaling factor controls the amplification of the difference between the individuals $r1$ and $r2$ and is used to avoid stagnation of the search process. x_{best} is the solution that attains the best value.

- *Crossover operation:* Instead of using the classical crossover operators of EAs where parts of the parents are recombined, the DE recombination operator is

Classical-Optimization Results-function f_1,$D = 5m$,$H = 50m$			
Layout	Parameters	N-dim GS	GBNM
NS-C	$[R_{\text{space}}, C_{\text{space}}, d_{\text{1st-row}}]$	$[\frac{3D}{2}, \frac{3D}{2}, H_{\text{tower}}]$	$[\frac{3D}{2}, \frac{3D}{2}, H_{\text{tower}}]$
R-S	$[R_{\text{diff}}, C_{\text{diff}}, d_{\text{1st-row}}]$	$[0.5, 0.5, H_{\text{tower}}]$	$[0.5, 0.5, H_{\text{tower}}]$
G-Spiral	$[a, b, \phi]$	$[2, 0.8, \phi]$	$[2, 0.8, \phi]$

Table 8: Results of Multi-GS & GBNM

based on a linear combination in which the distance concept plays an important role. The crossover operation is applied to further perturb the generated solutions and enhance the diversity. This is performed by copying the parameters of the generated mutant vector and its corresponding vector i in the original population according to a crossover factor denoted by $CR \in [0, 1]$. For each parameter, a random number in the range $[0, 1]$ is generated and compared with CR, and if its value is greater than or equal to CR, the parameter value is taken from the parent vector; otherwise, it will be taken from the mutant.

- *Selection procedure:* For the selection procedure, normally, the solutions of the old population are compared with trial solutions for the generation of a new population. For this purpose, the objective function corresponding to each trial solution is evaluated and compared with the value of the parent. If the new solution does not perform better, the old solution is retained; otherwise, the new solution replaces the parent solution. Thus an elitist replacement is considered, that is, the offspring will replace its parent if its objective value is better or equal to the parent one.

- *Stopping criteria:*

  A Matlab code is written, which effectively optimizes a preliminary generated heliostat field using Differential Evolution. After a new population is formed, the algorithm updates the global best solution. This code is run for 100 iterations which is considered as stopping criteria. The number of iterations is user defined parameter.

- DE has algorithmic complexity of $\mathcal{O}(n)$.

Implementation Maximum Number of Iterations is taken to be 100; Population Size 50; Lower bound of scaling factor $\beta_{\min} = 0.2$; and the upper bound of scaling factor is $\beta_{\max} = 0.8$ and crossover probability $pCR = 0.2$.

DE was performed for one parameter d_{sep} for RS-Configuration. The results of this work are presented in table [8].

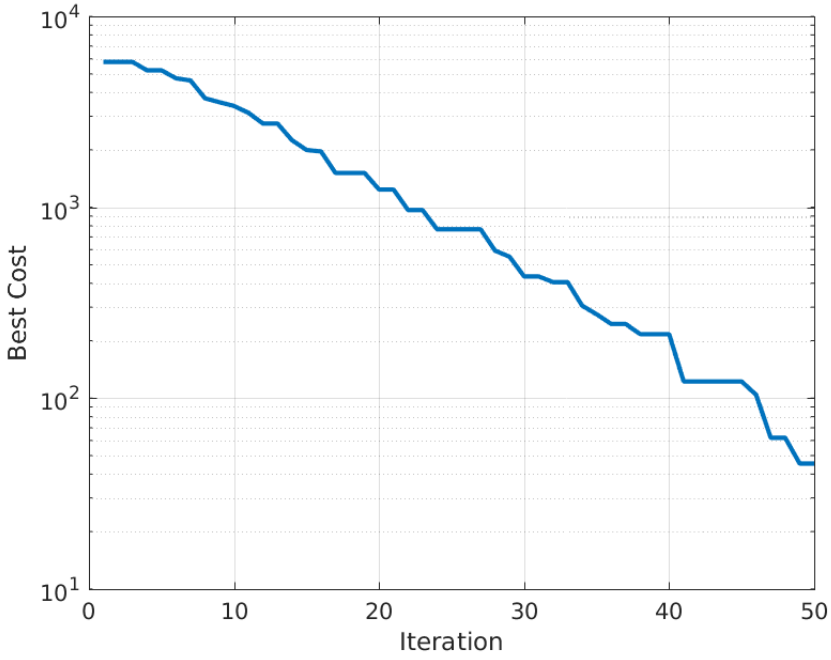


Figure 11: Results of Differential Evolution

4.2. Local Search Approach

The field configuartion at the end of previous section is taken as input for the local search algorithms. The basic idea of local search is to improve the existing layout's objective value until a certain requirement is fulfilled (such as molten salt flow feasibility in pipelines) by selecting the best performing heliostats. Here, we control the local search through number of iterations as input. In implementation of three approaches below, the cartesian coordiantes are converted to polar coordinates. If the polar coordinates of reference heliostat are (r, θ) and for neighbor are (r_1, θ_1). Then the new cartesian coordinate of a neighbor w.r.to reference point will be:

$$(a, b) = (rcos(\theta) - r_1 cos(\theta_1)), rsin(\theta) - r_1 sin(\theta_1))$$

and we change to polar coordiantes again:

$$((r_{new}, \theta_{new}) = \sqrt{(}a^2 + b^2), tan^{-1}(\frac{b}{a}))$$

.

4.2.1. Single Neighbor Selection

- Considering each coordinate as reference point, we calculate the Euclidean distances to all the other points, and sort the distances and select first neighbor of each point.

- Draw a circle of radius r equals the distance between reference point and its first nearest neighbor; then respecting the safety distance d_{safety}; update the radius[12]

  $$r' = r - d_{\text{safety}}$$

  We update the polar coordinates of neighbor w.r.to the reference point polar coordinates.

- We discretize this inner circle of radius r' along circumference in clockwise direction starting from the polar angle of neighbor with the fixed increment of angle $\frac{2\pi}{n}$ where n is user-defined input which is number of required trial points[13] and store these trials points.

- Evaluate a simmulation test function value at each of these trials positions and compare with the function value at reference position; The best value is chosen and the coordiante of reference point is updated by the new best trial position and each of the coordiantes must satisfy a restriction security constraint.

  $$d_{\text{safety}} \leq \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

  where d_{safety} is also a input parameter. If any of the new position violates the security constarints, we do not take that trial position.

- This procedure is repeated for all the reference heliostats and output for one iteration is given as input to the next iteration. We run this iteration for 200 times. The configuration corresponding to best objective function value is selected.

- Time complexity of this algorithm is $\mathcal{O}(n^2)$; where n is the number of heliostats.

This algorithm was tested for the three test cases with simulation function f_1 and N is the number of heliostats:

- PS10 and $N = 624$
- Golden Spiral and $N = 600$
- Dense Radial Staggered with Number of zones = 2.

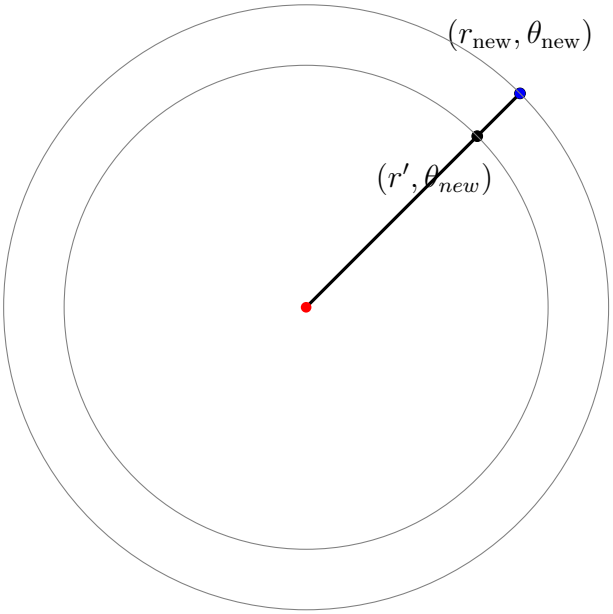


Figure 12: Selection of Ist neighbor

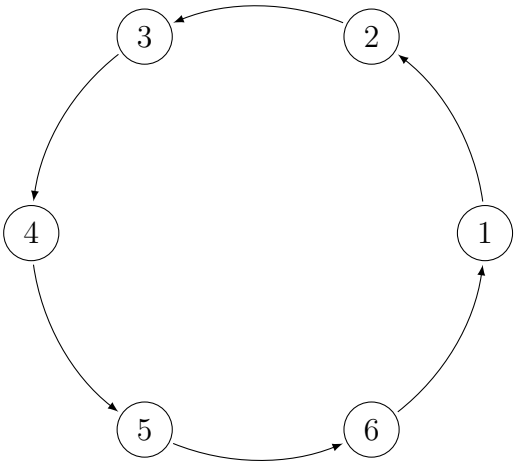
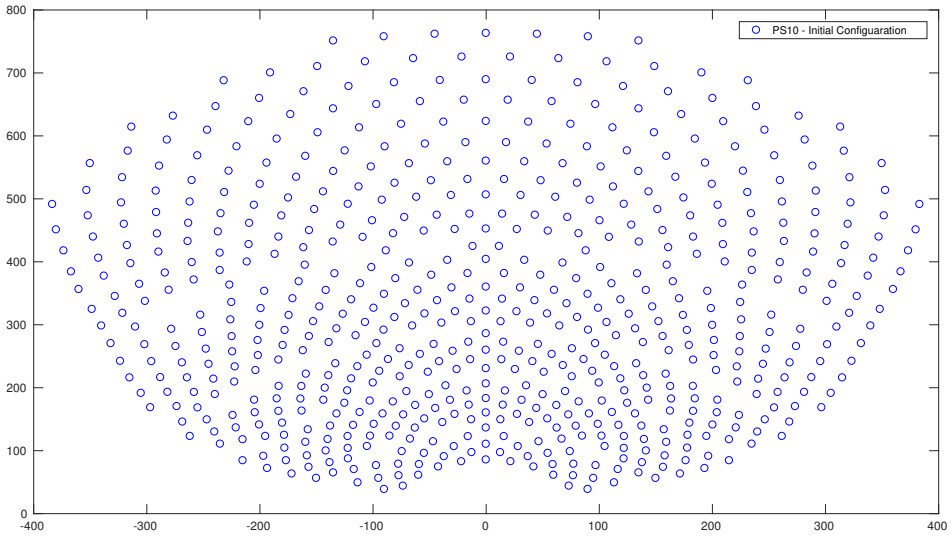
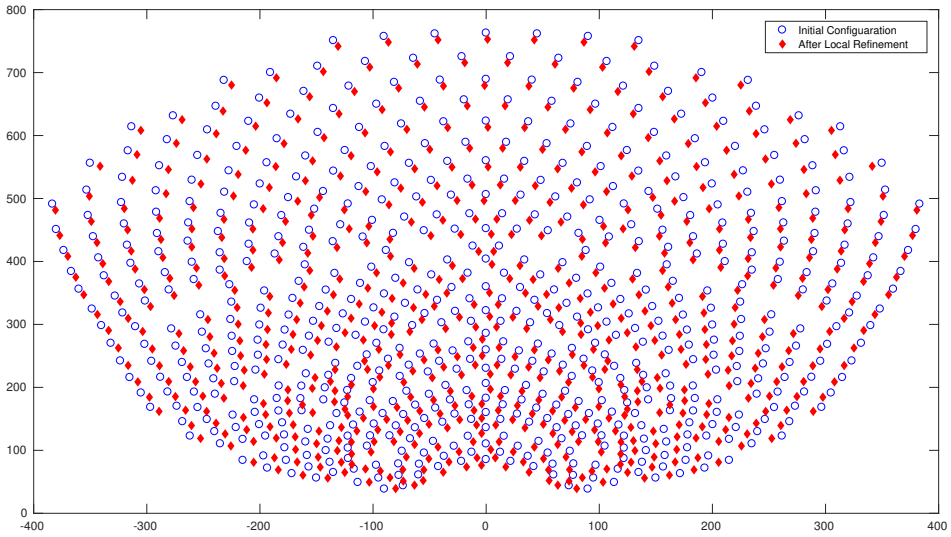


Figure 13: Discretization along circumference for 5 new trial points

Results for PS10



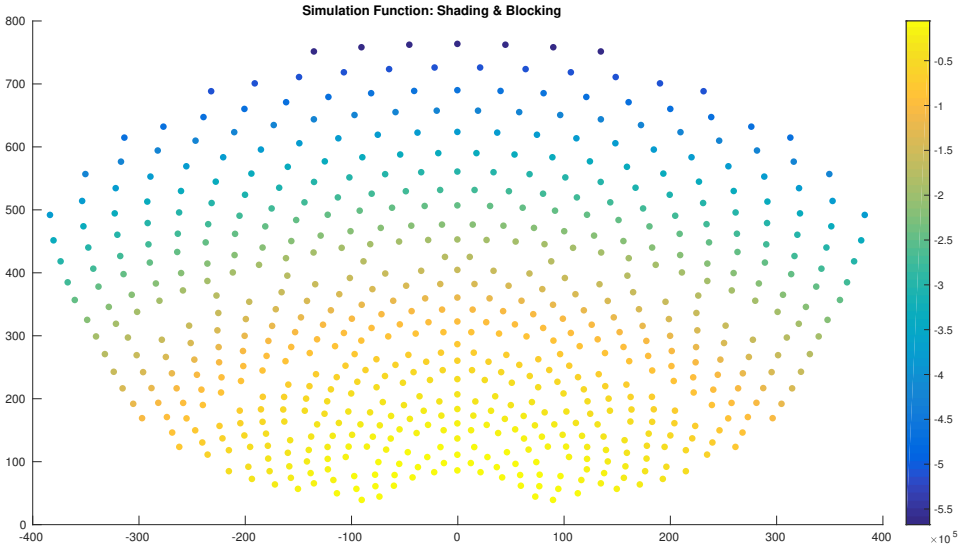
(a) PS10 Layout,Spain



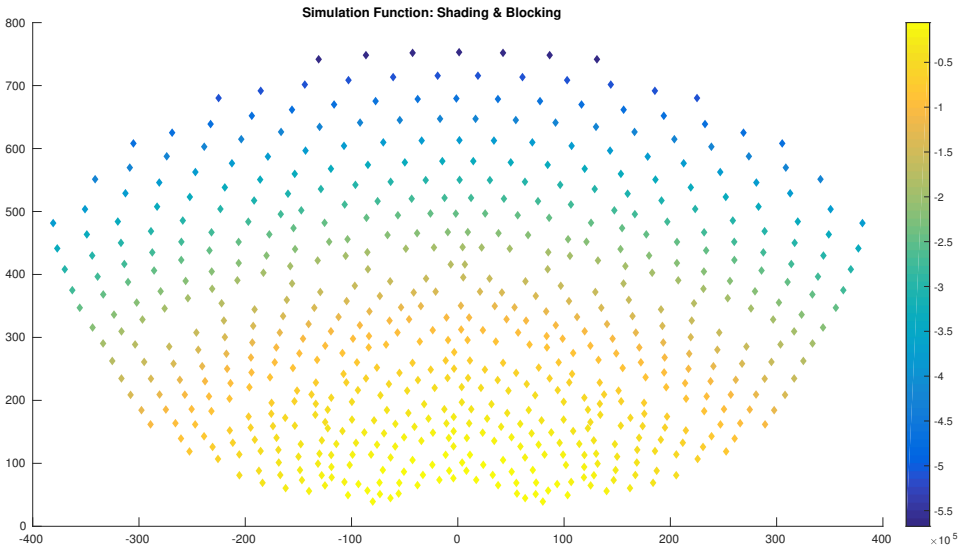
(b) After 10 Local Refinement iterations

Figure 14: PS10 Radial-Staggered Configuration

Results for PS10



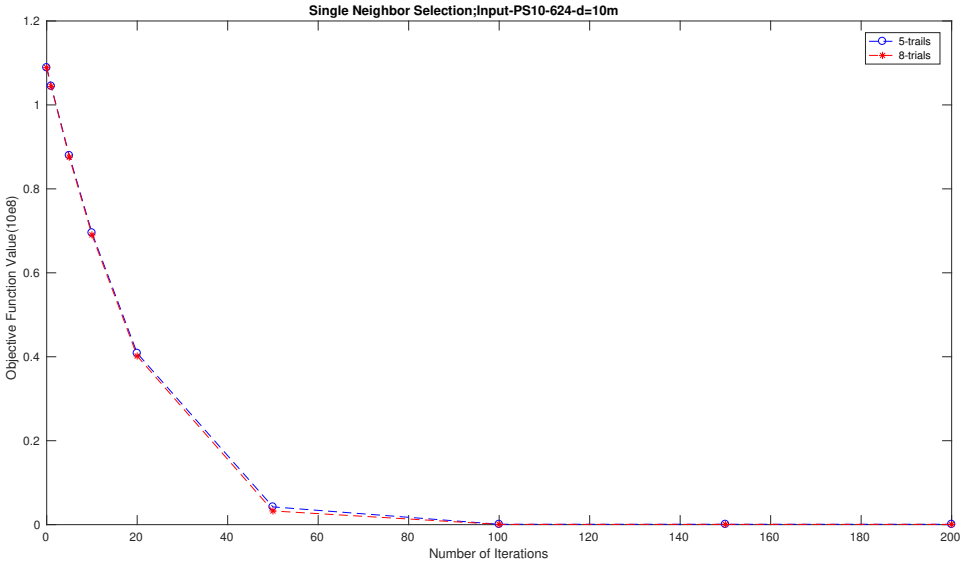
(a) PS10 Layout



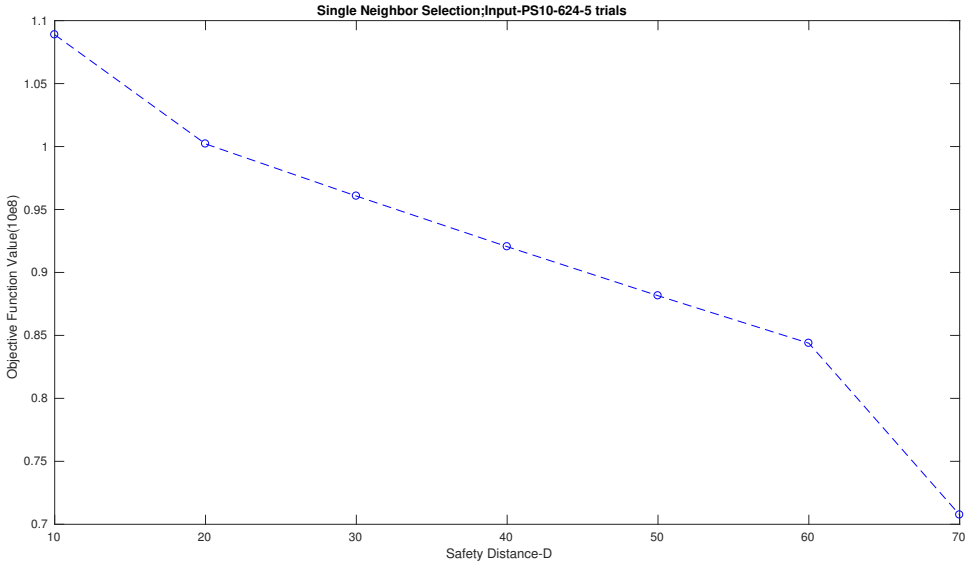
(b) After Local Refinement of PS10

Figure 15: Intensity Map - Shading and Blocking Efficiency

Results for PS10



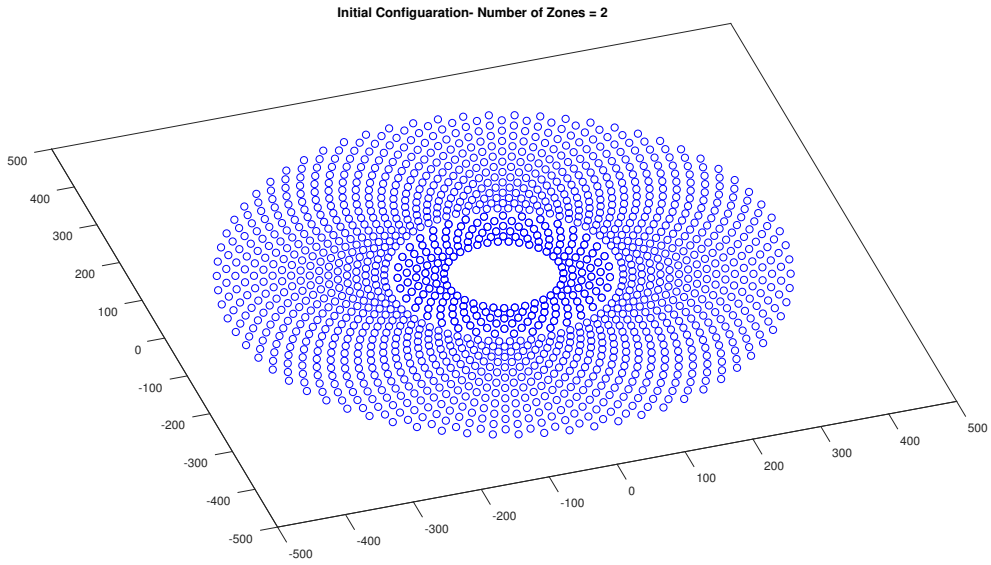
(a) Objective Value vs Number of Iterations



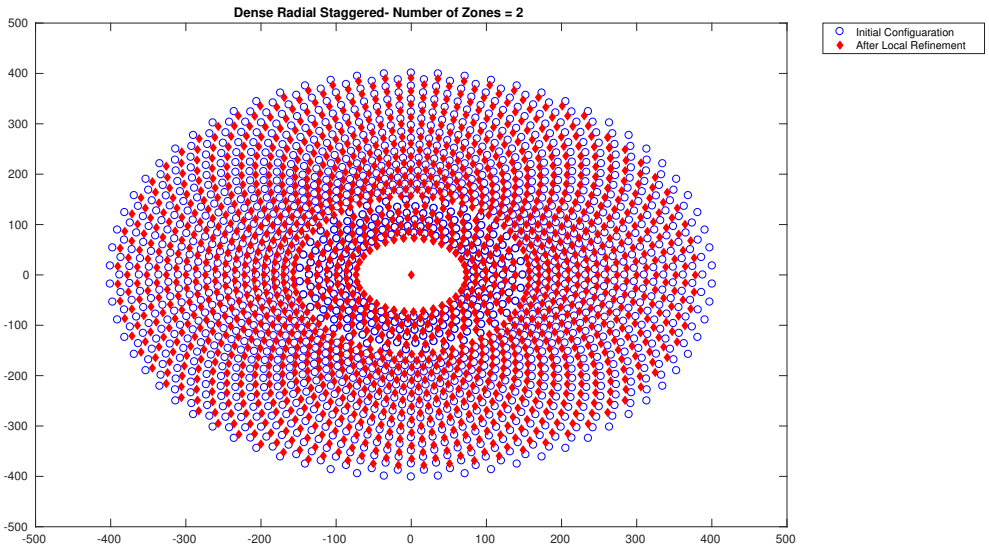
(b) Objective Function vs DM-Characteristic Diameter

Figure 16: Results of Single Neighbor best-trial heuristics

Results for Dense Radial Staggered-Campo Code



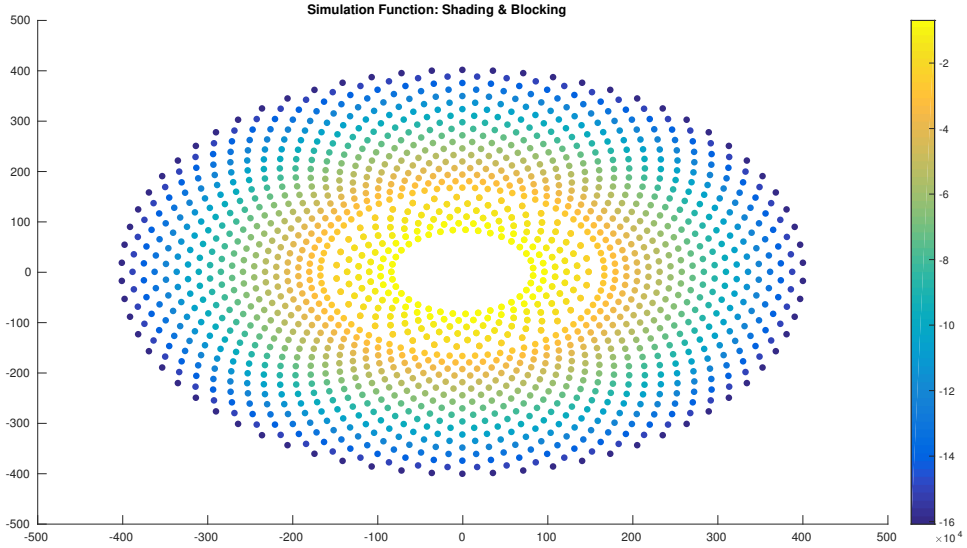
(a) Initial Configuarion



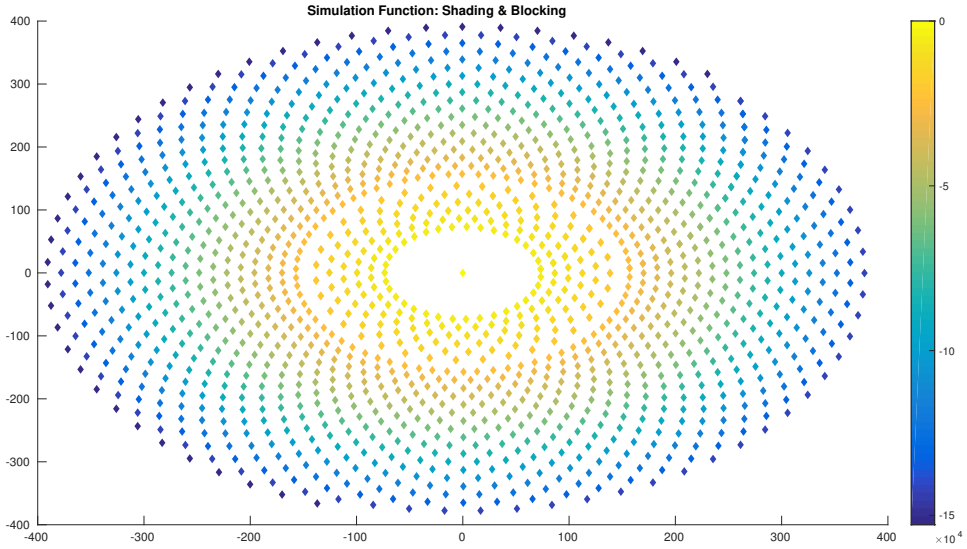
(b) After Local-Refinement

Figure 17: Dense Radial-Staggered Configuration

Results for Dense Radial Staggered-Campo Code



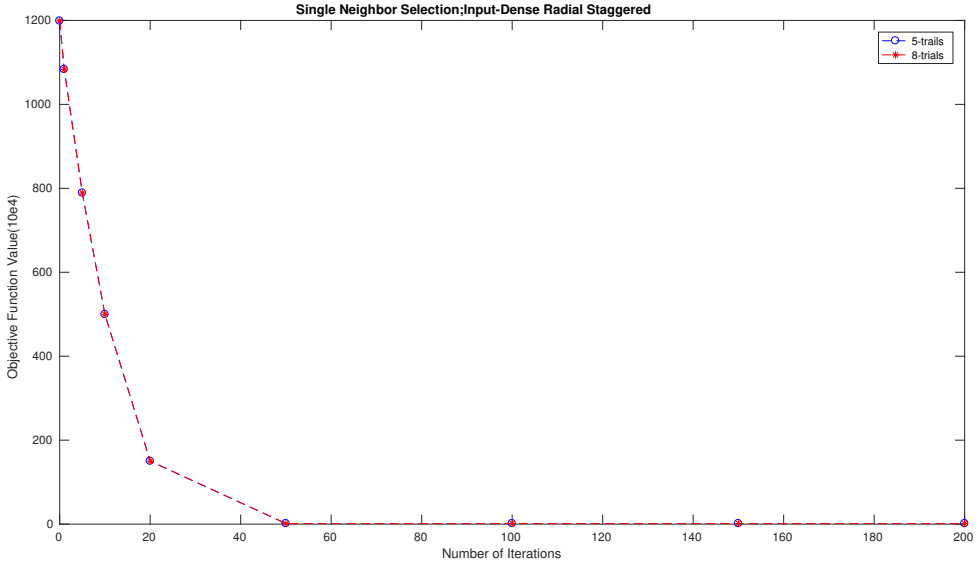
(a) Initial Dense RS Field



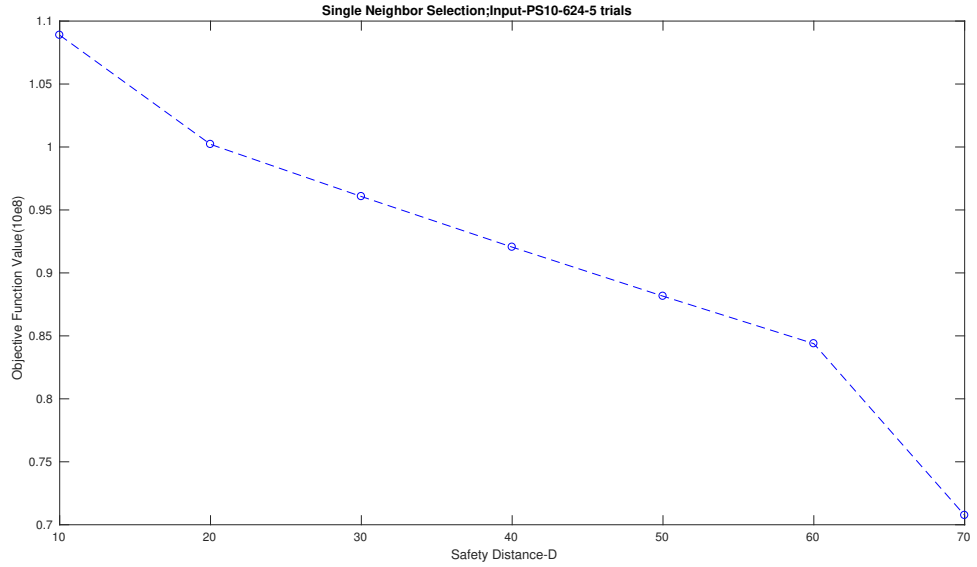
(b) After Local Refinement

Figure 18: Intensity Map - Shading and Blocking Efficiency

Results for Dense Radial Staggered-Campo Code



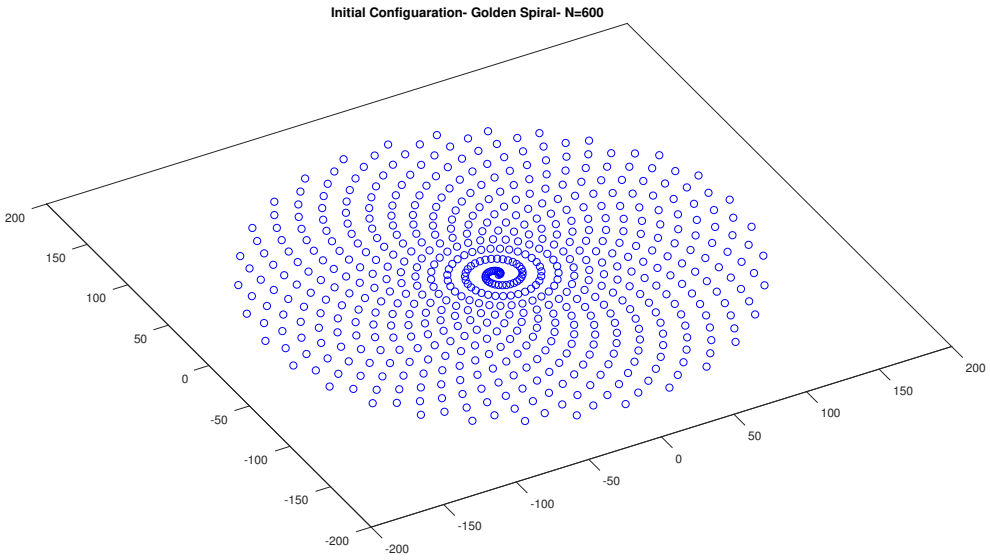
(a) Number of Iterations vs Objective Function



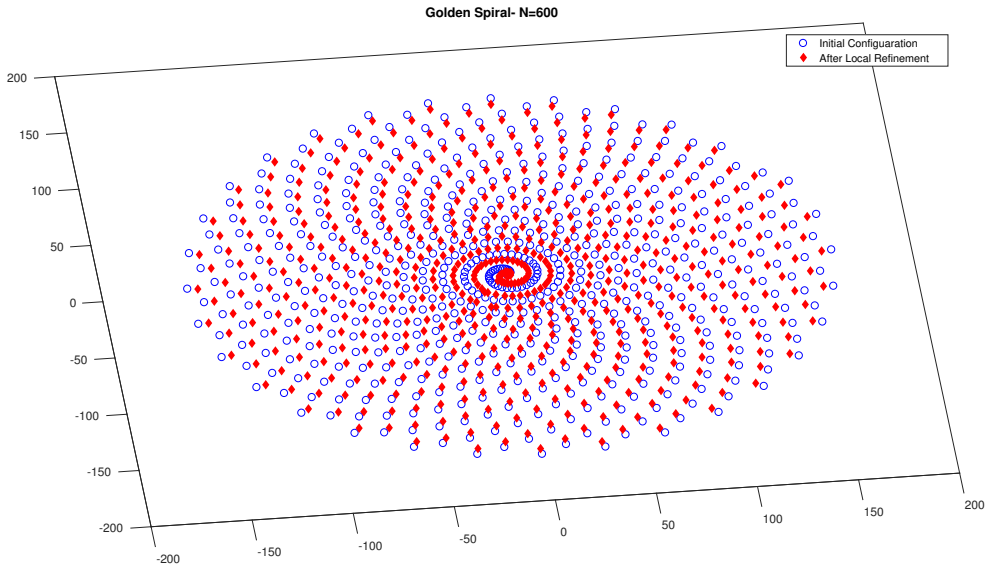
(b) Charateristic Diameter vs Objective Function

Figure 19: Results of Single Neighbor best-trial heuristics

Results for Biometric Spiral



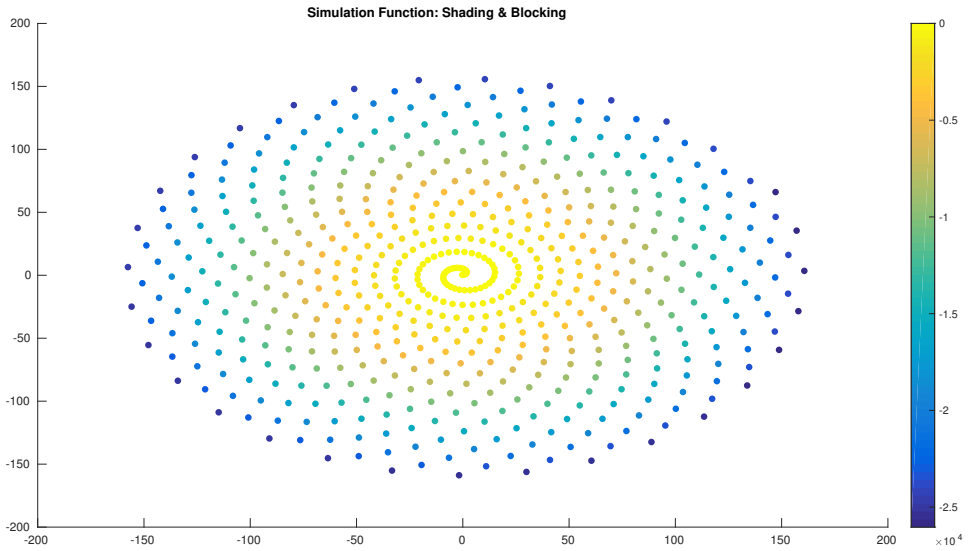
(a) Golden Spiral



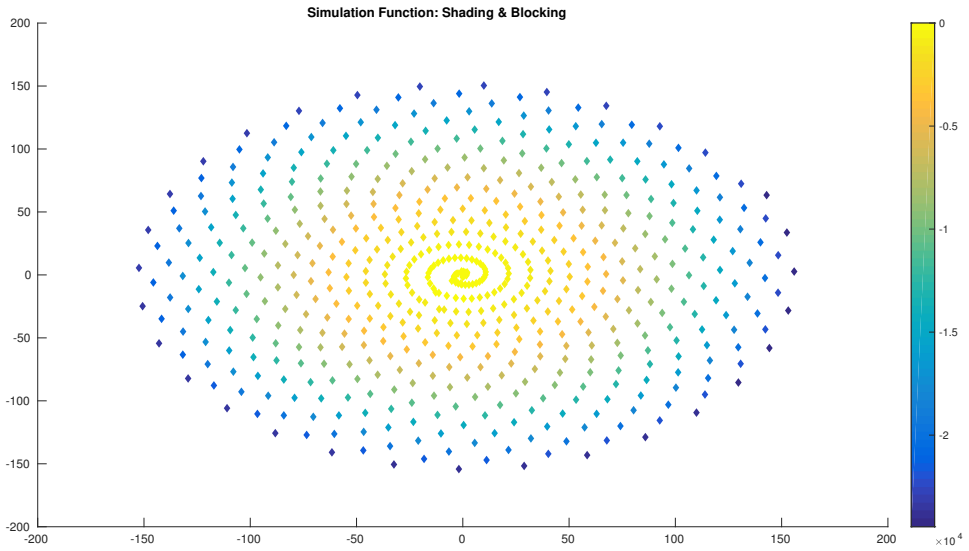
(b) After Local Refinement-Golden Spiral

Figure 20: Golden Spiral Configuration

Results for Biometric Spiral



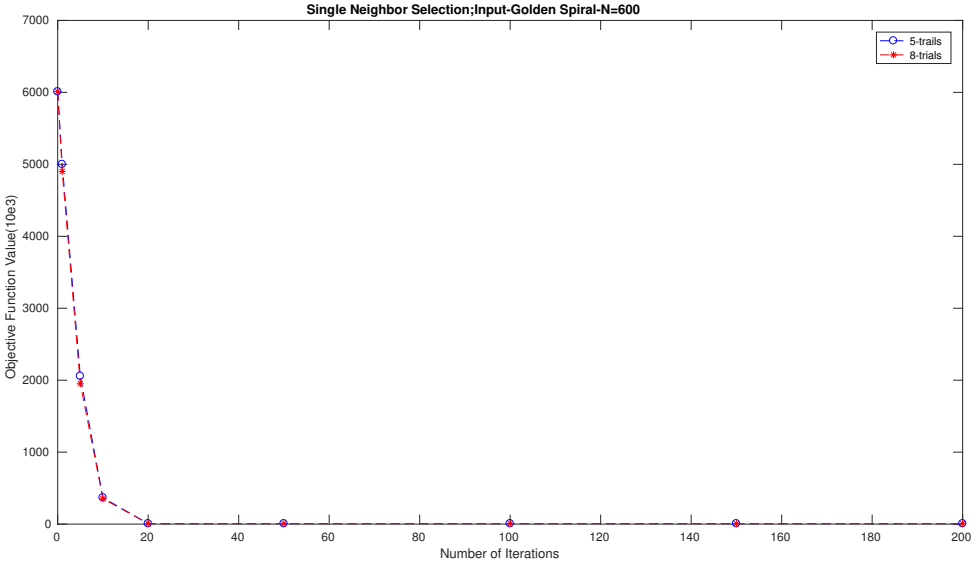
(a) Initial Configuration



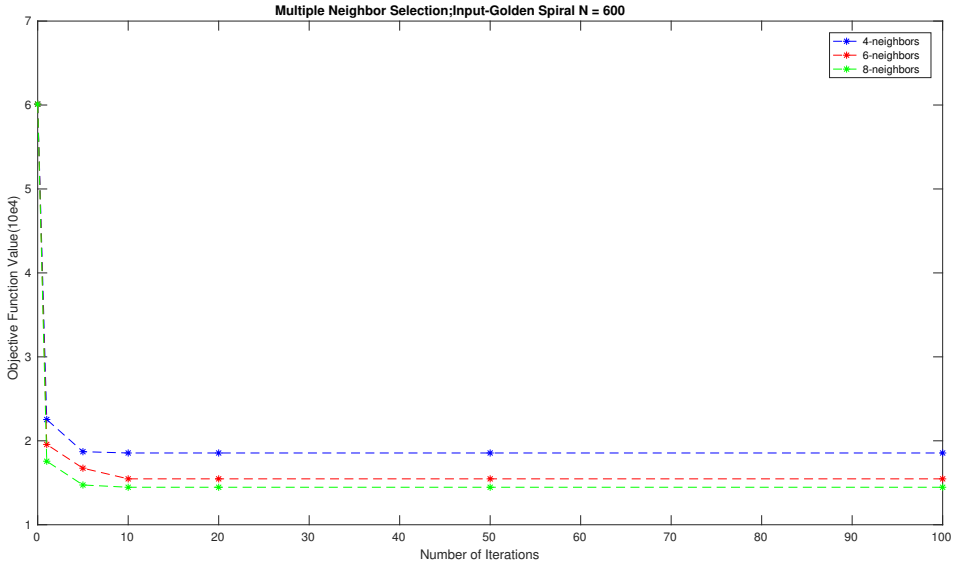
(b) After Local Refinement

Figure 21: Intensity Map - Shading and Blocking Efficiency

Results for Biometric Spiral



(a) Results of Single Neighbor best-trial heuristics



(b) Results of Multiple Neighbor best-trial heuristics

Figure 22: Results of best-trial heuristics

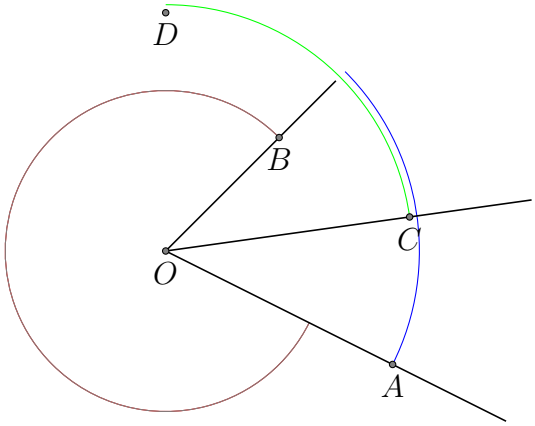


Figure 23: Selection of Multiple neighbors and triangular subregion

4.2.2. Multiple Neighbors Selection

- Considering each coordinate as reference point, we calculate the Euclidean distances to all the other points, and sort the distances and select nearest neighbors of a each point according to user defined input.
- For each reference point, the polar coordinates of its neighbors are updated w.r.to reference point polar coordinates.
- Then we sort these updated polar angles and for each adjacent pair of polar angles we define sweeping sector[23] of fix angular sweep $\frac{\theta_i+\theta_{i+1}}{2}$ in both the directions of the polar angle of each neighbor.
- Then we discretize[24]this sector using section formula and store these trials points for each sector; each sector have 8 new trial points. Thus if a reference point have 4 neighbors then total trial points for that reference point will be 32.
- we compare test function value at each of these trials positions with the function value at reference position; The best value is chosen and the coordiante of refernece point is updated by the new best trial position and each of the coordiantes must satisfy a restriction security constraint.

$$d_{\text{safety}} \leq \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

  where d_{safety} is also a input parameter.
- This is done for all the reference points and output for one iteration is given as input to the next iteration. We run this iteration for 100 times.
- Time complexity of this algorithm is $\mathcal{O}(n^2)$; where n is the number of heliostats.

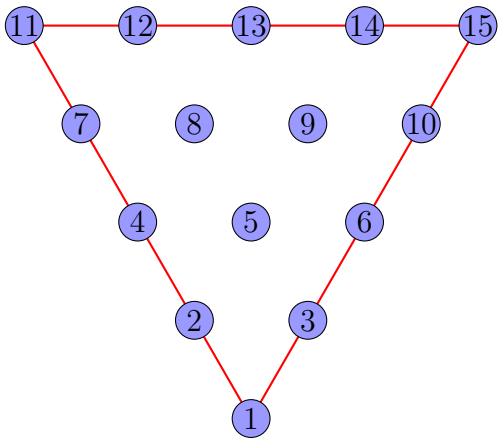


Figure 24: Discretization of a trianglur subregion

4.2.3. Radius-Dependent Neighbors Selection

- Given a predefined pattern, for each heliostat we select a neighbourhood search radius; let say 3D, where D is the heliostat characterstic diameter, we record the neighbours which lies in the circle with the radius 3D and center as the reference heliostat.
- Now, for each heliostat, we have specified number of neighbours; we find a convex hull of these coordinates in the plane [25]. Given a set of points in the plane. the convex hull of the set is the smallest convex polygon that contains all the points of it.
- The corner points are now used to divide the convex hull in convex triangular subregions.
- Each triangular subregions are then discretized [24] for trial positions to evaluate the objective function.
- Comparing all the trial positions with the reference heliostat position value; the position with minimum test function value is selected to update the position of reference heliostat.

Neighbourhood Search Based on different patterns; we will have different number of heliostats neighbouring a reference heliostat. So, this generic approach of neighbourhood search is compatible with all existing layouts; we simulate the neighbourhood search with different radius like 2D,3D,4D...etc; where D is the characteristic diameter of a heliostat which is same for all heliostats.

Convex Hull Algorithm Given a set of points in the plane. the convex hull of the set is the smallest convex polygon that contains all the points of it. In other words; a convex hull can be defined as:

- smallest perimeter fence enclosing the points.
- smallest area convex polygon enclosing the points.

There exists multiple algorithms for finding the convex hull of a given set of points. We use Graham's scan algorithm, from which we can find Convex Hull in $\mathcal{O}(n \log n)$ time where n is the number of neighbours. The Graham's algorithm is followed as: Let points $[0, 1.....n-1]$ be the input array.

1. Find the bottom-most point by comparing y coordinate of all points. If there are two points with same y value, then the point with smaller x coordinate value is considered. Let the bottom-most point be P_0. Put P_0 at first position in output hull.
2. Consider the remaining $n-1$ points and sort them by polar angle in counter-clockwise order around points[0]. If polar angle of two points is same, then put the nearest point first.
3. After sorting, check if two or more points have same angle. If two more points have same angle, then remove all same angle points except the point farthest from P_0. Let the size of new array be m.
4. If m is less than 3, then return Convex Hull not possible.
5. Create an empty stack S and push points[0], points[1] and points[2] to S.
6. Process remaining $m-3$ points one by one. Do following for every point points[i]
   - Keep removing points from stack while orientation of following 3 points is not counterclockwise or they don't make a left turn.
     a) Point next to top in stack
     b) Point at the top of stack
     c) points[i]
   - Push points[i] to S.
7. Print contents of S

The above algorithm can be divided in two phases.

Phase-1(Sort points): We first find the bottom-most point. The idea is to pre-process points by sorting them with respect to the bottom-most point. Once the points are sorted, they form a simple closed path which is shown in [25]. **Sorting criteria** The computation of actual angles would be inefficient since trigonometric functions are

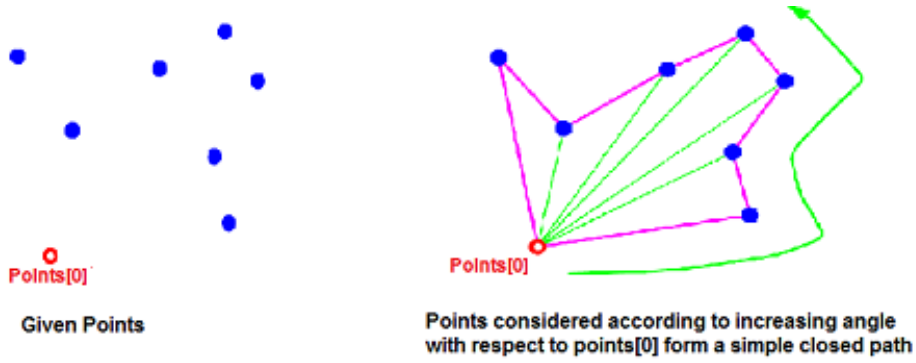


Figure 25: Sorting of points and Convex Hull Formation

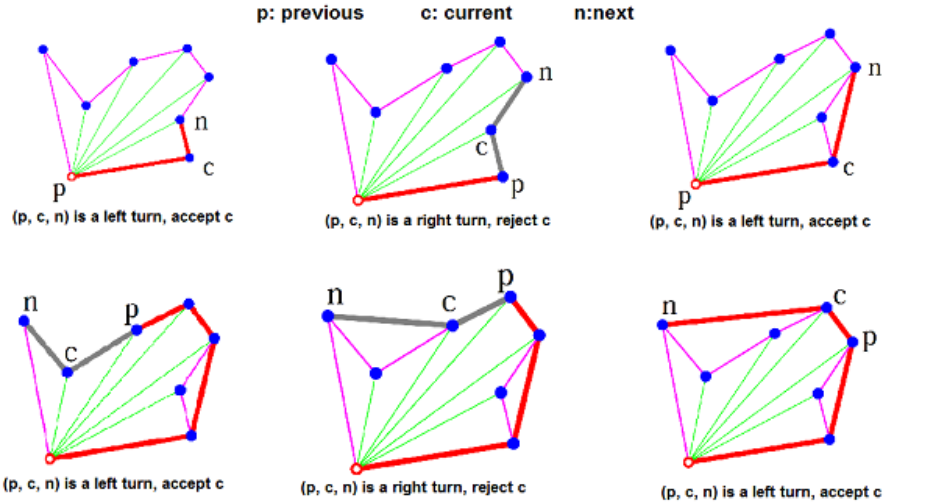


Figure 26: Accept and reject points with orientation function

not simple to evaluate. The idea is to use the orientation to compare angles without actually computing them.

Phase-2(Accept or Reject Points): Once we have the closed path, the next step is to traverse the path and remove concave points on this path. Again, The concept of orientation helps how to decide which point to remove and which to keep. The first two points in sorted array are always part of Convex Hull. For remaining points, we keep track of recent three points, and find the angle formed by them. Let the three points be prev(p), curr(c) and next(n). If orientation of these points; considering them in same order; is not counterclockwise, we discard c, otherwise we keep it. Diagram [26] shows step by step process of this phase.

Time Complexity: Let n be the number of input points. The algorithm takes $\mathcal{O}(n \log n)$ time if we use a $\mathcal{O}(n \log n)$ sorting algorithm. The first step (finding the bottom-most point) takes $\mathcal{O}(n)$ time. The second step; sorting points; takes $\mathcal{O}(n \log n)$

time. Third step takes $\mathcal{O}(n)$ time. In third step, every element is pushed and popped at most one time. So the sixth step to process points one by one takes $\mathcal{O}(n)$ time, assuming that the stack operations take $\mathcal{O}(1)$ time. Overall complexity is

$$\mathcal{O}(n) + \mathcal{O}(n \log n) + \mathcal{O}(n) + \mathcal{O}(n)$$

which is $\mathcal{O}(n \log n)$

Discretization Now we have a convex hull corresponding to each reference point. Now, we take all the corner points of convex hull and the reference point and we do traingulation of convex hull and then we discretize each triangle. Discretization[24] is one approach for getting trial positions, we can do it by random selection function. Here we create a regular mesh for each triangle.

Objective Evaluation In each triangle, we evaluate the function and compare with the reference heliostat function value and record the minimum function value position. There will be two type of triangles; one without any other heliostat and second type of traingle which have some previous selected neighbours; for which we already have the recorded function value. If the trial coordinates position violates the following constarints:

$$\delta \leq \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

where $\delta = d_{\text{safety}}$ is the safety distance. Then we reject that trial position and do not evaluate objective function at that position and compare rest of the others trials position function value and record the lowest value. From each triangle, we have minimum value which we refer as local minimum and then take the minimum of all these values at local minimum; we get a global minimum.

In summary, Methodologically, all three procedure are composed of a number of consecutive simulation cycles depending on the number of heliostats. Each simulation cycles(iteration) is a loop over all heliostats in the neighbourhood of the reference heliostat. In each loop one heliostat is selected as the one to be repositioned. S is the set of cartesian coordinates of all heliostats. $S = \{(x_i, y_i) \in \mathbb{R}^2) : i \in [1, N]\}$ For each reference heliostat H_i; $N(i) = \{1, 2, ...k\}$ is the index set of neighbouring heliostats and which satisfying following mechanical constarints:

$$\delta \leq \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \tau$$

where $\delta = d_{\text{safety}}$ is the safety distance and τ is the distance dependent on number of neighbours for a specified radius in a specified pattern.

In one simulation cycle; all the heliostats have been repositioned according to their convex hull region discretization. The best positions for each reference heliostat is recorded and will be used in futher simulations. Thus after each loop step, every reference heliostat will be at new position expectedly contribute for better performance of the system. If no better annual improvement is noticed, the heliostat position

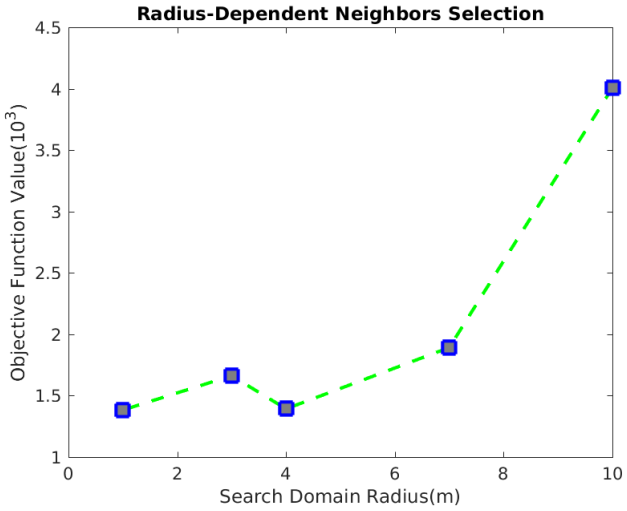


Figure 27: Results of Algorithm-3

remain unchanged. This position will be stored again and will be used in consequent simulation.

Since, relocating the heliostats will influence the adjacent heliostats, it offers additional improvement potential for those heliostats that were previously handled in the same loop. That's why, it is requird to perform consecutive simulation cycles to obtain best result.

To confirm the locality of algorithm, we can take the larger radius, which in result gives the larger convex hull with more neighbours and perform the same simulation.

We have thus tested all three algorithms and results of first algorithm, for the selection of best configuration are presented from Figure [14] to Figure [22]. A comparison of multiple neighbours selection is done in [22]. Figure [27] demonstrates the optimal radius for search of neighbours is 4 units for the test case of golden spiral.

5. Conclusion

Metaheuristics + Nelder-Mead method

5.1. Why Hybrid Meta-heuristics Strategy

The engineering optimization problems are characterized by the calculation-intensive system simulations, existence of designs constraints, and a multiplicity of local solutions. The high numerical cost of global optimizers has been at the heart of subsequent efforts to speed up the search either by adding problem-specific knowledge to the search, or by mixing efficient local algorithms with global algorithms. There are many ways in which local and global searches can cooperate.

The simplest strategy is to link the searches in series, meaning that firstly, a global optimization of limited cost is executed, the solution of which is refined by a local search. An example of the serial hybrid is given in Shang et al. (2001), in which simulated annealing, the global optimizer, is coupled with a sequential quadratic programming and a Nelder–Mead algorithm.

A hybrid method combining two algorithms is suitable for the global optimization of multiminima functions. To localize a "promising area", likely to contain a global minimum, it is necessary to well "explore" the whole search domain. When a promising area is detected, the appropriate tools must be used to "exploit" this area and obtain the optimum as accurately and quickly as possible. Both tasks are hardly performed through only one method.

We propose an algorithm using two processes, each one devoted to one task. Global metaheuristics, such as simulated annealing(SA), tabu search(TS), particle swarm optimization(PSO) and evolutionary algorithms like genetic algorithm (GA) or differential evolution(DE) are efficient to localize the "best" areas. On the other hand, local search methods are classically available: in particular the hill climbing (e.g. the quasi-Newton method, Powell method), and the Nelder–Mead simplex search. Therefore we worked out an hybrid method, performing the exploration with a DE, and the exploitation with a Nelder–Mead. Furthermore, DE will be replaced by a generalized metaheuristic framework, having other metaheuristics compatible with Nelder-Mead. For further improvement, the local refinement procedures can be used.

Multi-dimensional Golden Section works for unimodal functions only, GBNM is suitable for our field design optimization problem but to select the best initial population for the GBNM is itself an hard optimization problem. Thus the idea of using metaheuristics as first step is appropriate; for the generation of initial population under the given constraints and then use Nelder-Mead for the local search. The results of this work are presented in the table [9]. Here, also we present the profound pseudocode of DE, PSO hybridized with Nelder-Mead which outperforms the GBNM. TS can also be hybridized with Nelder-Mead. For future work, a comparison analysis would be performed to find the best hybrid method for the specific problem.

Classical-Optimization Results-function f_1,$D = 5m$,$H = 50m$			
Layout	Parameters	Hybrid DE-NM	GBNM
NS-C	$[R_{\text{space}}, C_{\text{space}}, d_{\text{1st-row}}]$	$[\frac{3D}{2}, \frac{3D}{2}, H_{\text{tower}}]$	$[\frac{3D}{2}, \frac{3D}{2}, H_{\text{tower}}]$
R-S	$[R_{\text{diff}}, C_{\text{diff}}, d_{\text{1st-row}}]$	$[0.5, 0.5, H_{\text{tower}}]$	$[0.5, 0.5, H_{\text{tower}}]$
G-Spiral	$[a, b, \phi]$	$[4.0, 0.65, \phi]$	$[4.0, 0.65, \phi]$

Table 9: Results of GBNM & Hybrid DE-NM

5.2. Pseudocodes

Velocity Update Equations:

$$V_i^{\text{New}}(t+1) = c_0 \cdot V_i^{\text{old}}(t) + c_1 \cdot \text{rand}() \cdot (P_i(t) - X_i^{\text{old}}(t)) + c_2 \cdot \text{rand}() \cdot (P_g(t) - X_i^{\text{old}}(t)) \quad (52)$$

$$X_i^{\text{New}}(t+1) = X_i^{\text{old}}(t) + V_i^{\text{New}}(t+1) \quad (53)$$

Algorithm 1 PSO

```
Initialize a population of particles with random values positions and velocities from D dimensions in the search space
while Termination condition not reached do
    for Each particle i do
        Adapt velocity of the particle using Equation 52
        Update the position of the particle using Equation 53
        Evaluate the fitness f(X_i)
        if f(X_i) < f(P_i) then
            P_i ← X_i
        end if
        if f(X_i) < f(P_g) then
            P_g ← X_i
        end if
    end for
end while
```

The pseudo-code of hybrid method is inspired from [6]. The pseudocodes of PSO and its NM-hybrid are taken from [6]. The main blocks of algorithm: initialization of parameters, generation of the initial population, production of the new population, intensification around the best point, and output of the best point found.

- Diversification: The algorithm starts with a large population uniformly dispersed within the whole solution space, then the new population is pro- duced through evoluationary steps. Once the offspring have been produced by selection, recombination, and mutation of individuals belonging to the old population, the objective function values of the offspring may be calculated, the new population

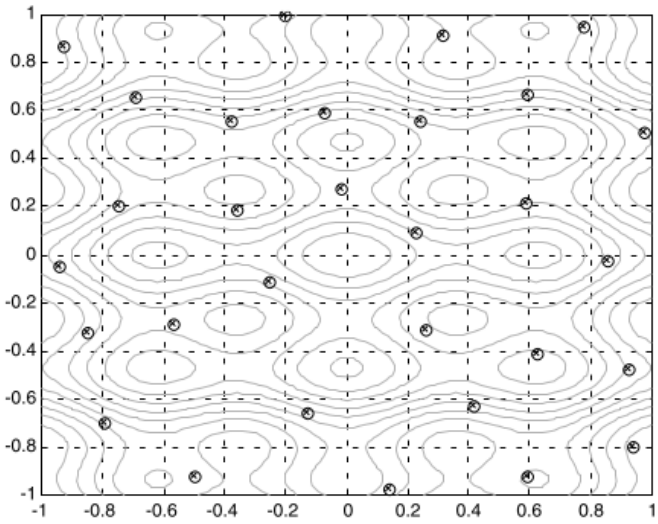


Figure 28: Dispersion of the initial population(30 individuals) in the solution space $[-1, 1]^2$, for f_2 function

is produced, and the process is reiterated. The exploration stops when one of the following conditions is reached:

1. given number of successive generations Max-Gen without detection of promising area is reached;
2. a given accuracy relating to the individuals coordinates is obtained.

The area where the most individuals lie is named as "promising area".

- Intensification inside the promising area: The best found point in the previous phase becomes the new initial solution x_0 in this intensification phase. We construct a new solution space centered around x_0. Each domain edge is reduced in a ratio given by the reduction parameter. The Nelder–Mead simplex algorithm performs the initial simplex. The first vertex is x_0 and other vertices x_i are chosen in such a way that they form a geometrical base, in generally an orthogonal one. The NM starts the search by using the various geometric moves.

- Stopping criteria: The algorithm performs the exploitation phase inside the promising area until the local stopping criteria are reached. There are two stopping criteria. The first criterion is related to the maximal number of iterations of the objective function MaxIter, the second one is a measure of how far the simplex s_i was moved from one iteration k to the following one $(k + 1)$.

The simulation test function is:

$$f(x_1, x_2) = x_1^2 + 2 + x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_1) + 0.7$$

with search domain:

$$-100 < x_j < 100; j = 1, 2$$

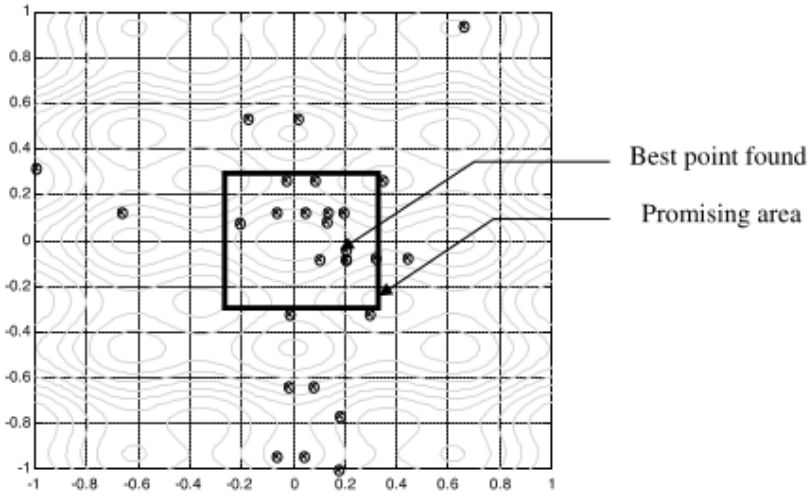


Figure 29: Detection of Promising area

Algorithm 2 Hybrid NM-PSO

1: Generate a population of size $k \cdot N + 1$
2: REPEAT
3: **while** Termination condition not reached **do**
4: Constraint Handling Methods

- Repair particles that violate the constraints by directing the infeasible solution toward the feasible region. Leave unrepairable solutions as they are.
- Evaluate the constraint fitness, the objective fitness of each particle and rank them.

5: Simplex Method: Apply NM operator to the top $N + 1$ particles and update the $(N + 1)^{th}$ particle.
6: PSO Method: Apply PSO operator for updating the remaining $(k - 1) \cdot N$ particles with worst fitness.

- Selection. From the population select the global best particle and the neighborhood best particles.
- Velocity Update. Apply velocity updates to the $(k - 1) \cdot N$ particles with worst fitness according to Equations 52 and 53.

7: **end while**

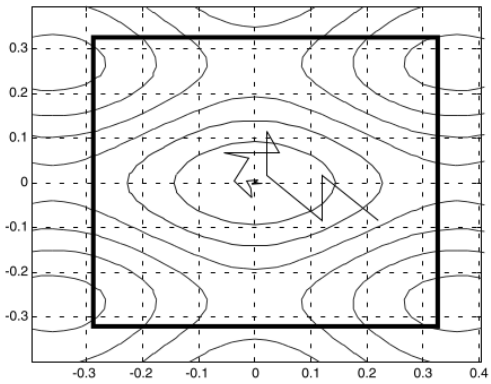


Figure 30: The path followed by the NM to find best solution in local space for f_2 simulation function

Algorithm 3 Hybrid NM-DE

G, G' Quality = table with size N;
Generation of initial population
$(f_{\min}, x_{\min})$ = best point among parent population
Diversification
Repeat
while G' was full **do**
    while detection of promising area **do**
        Quality = produce the most adapted population
        Repeat(phases of evolutionary reproduction)

- selection
- recombination
- mutation

    end while
    Replacement(G, G')
end while
Intensification
Repeat
reducation of search domain
construction of initial simplex
while stopping criteria is reached **do**
    Geometric modifications

- reflection
- exapnsion
- contarction

end while

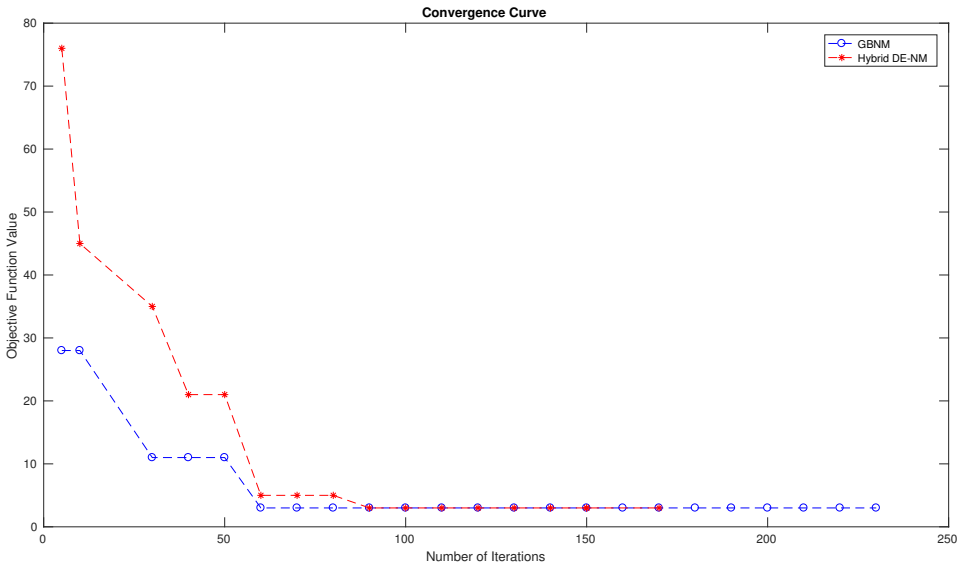


Figure 31: Convergence of GBNM and Hybrid DE-NM for f_2 simulation function

The DE-NM hybrid may converge more quickly than GBNM, though starting from a point far from the solution which can be observed from Figure [31].

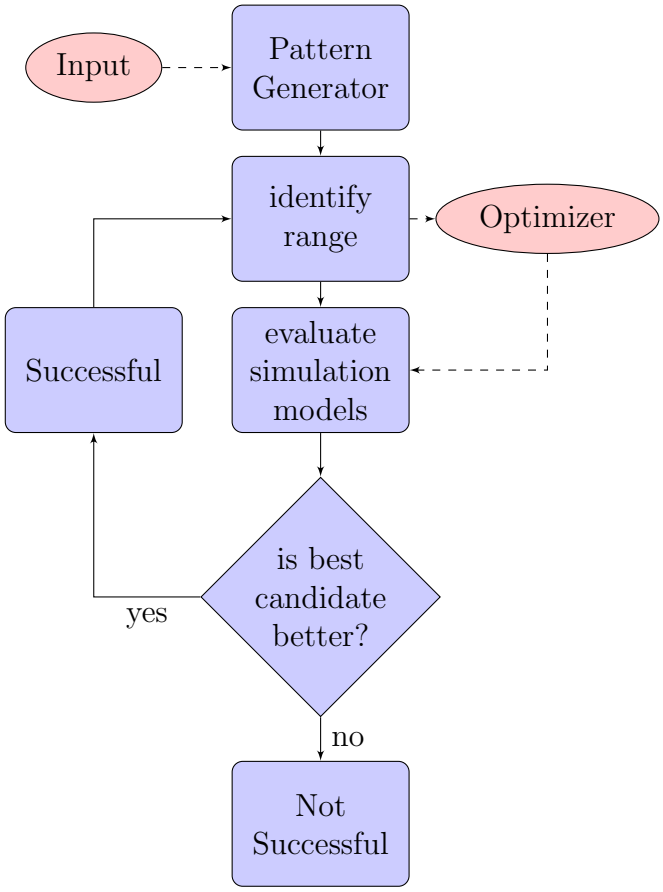
A. Appendix

A.1. Code Structure

- Pattern generator(PG) creates test-cases of different pattern with user-defined variables. PG also determines the range for the design parameters.
- Optimzer takes two types of input and work on two hybrid method approahes:
  1. Pattern + Classical Method + Local Refinement
  2. Differential Evolution + Nelder-Mead method

A.2. Outlook

- The Black-box optimizer might be verified by functions having global optimum. Some examples are Branin RCOS, Easom function, Goldstein and Price function, Shubert function, De Joung, Hartmann, Rosenbrock, Shekel, Zakharov function etc.
- The Hybrid method Differential Evolution and Nelder-Mead is a robust method can be used for general engineering design problems.
- The work in this thesis has been considered for only a simplified simulation tool and simplified objective function; but in practice the objective function might be complex, non-convex, multi-modal, non-differentiable and multi-objective;but still the Black-Box Optimizer is flexible for all type of objective functions.
- One of the future attempt to optimize heliostat field layout could be to introduce Atificial Intelligence and build a neural-network through deep-learning from data related to sun and atmospheric conditions throughout the year.
- There is significant decrement in objective value for simulation test function f_1, but in practice due to complexity of objective function, the improvement sometimes is very less significant around 1-5%.


Input
Pattern Generator
identify range
Optimizer
Successful
evaluate simulation models
is best candidate better?
yes
no
Not Successful

References

[1] F. Ramos A. Ramos. Strategies in tower solar power plant optimization. *arXiv:1205.3059 [math.OC]*, 2012.

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

[3] E. Carrizosa, C. Dominguez-Bravo, E. Fernandez-Cara, and M. Quero. An optimization approach to the design of multi-size heliostat fields. *Technical report IMUS*, 2014.

[4] Yen-Ching Chang. N-dimension golden section search: Its variants and limitations. In *Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on*, 2009.

[5] F.J. Collado and J.A. Turegano. Calculation of the annual thermal energy supplied by a defined heliostat field. *Solar energy*, 42(2):149–165, 1989.

[6] Yi-Tung Kao Erwie Zahara. Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications*, 2009.

[7] Siala F. and M. Elayeb. Mathematical formulation of a graphical method for a no-blocking heliostat field layout. *Renewable Energy*, 2011.

[8] Jesús Guallar Francisco J. Collado. Campo code: Generation of regular heliostat fields. *Renewable Energy*, 2011.

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

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

[11] R. Le Riche M.A. Luersen. Globalized nelder–mead method for engineering optimization. *Elsevier*, 2004.

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

[13] IIT Kharagpur NPTEL. Optimization. `http://nptel.ac.in/courses/111105039/#`, 2014. [Online; accessed 16-June-2014].

[14] David R Mills Philipp Schramek. Design of the heliostat field of the csiro solar tower. *Solar Energy Engineering*, 2009.

[15] 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.

[16] Sassi. Some notes on shadow and blockage effects. *Solar Energy*, 1983.

[17] M. Schmitz, P. Schwarzbözl, R. Buck, and R. Pitz-Paal. Assessment of the potential improvement due to multiple apertures in central receiver systems with secondary concentrators. *Solar energy*, 80(1):111–120, 2006.

[18] Z. Yao, Z. Wang, Z. Lu, and X. Wei. Modeling and simulation of the pioneer 1mw solar thermal central receiver system in china. *Renewable Energy*, 34(11): 2437–2446, 2009.