Diese Arbeit wurde vorgelegt am
Lehrstuhl für Mathematik (MathCCES)

Gradientenbasierten Heliostatfeld Layout-Optimierung
Gradient Based Heliostat Field Layout Optimization

Seminar
Simulation Sciences

Juli 2016

Vorgelegt von Presented by	Rohit Lad, 359892 rohit.lad@rwth-aachen.de
Erstprüfer First examiner	Prof. Dr. Martin Frank Lehrstuhl für Mathematik (MathCCES) RWTH Aachen
Zweitprüfer Second examiner	Pascal Richter Doktorand Lehrstuhl für Mathematik (MathCCES) RWTH Aachen

Contents

List of Figures

1 Introduction

The project deals with optimization of solar tower power plants. A solar tower power plant has an arrangement of large mirrors which are used to concentrate rays of sunlight on a reciever where fluid is heated up. The heat of fluid is used to generate steam which powers a turbine to generate electricity. The placement of mirrors may lead to individual mirrors being blocked and shaded which affects the efficiency of power plant. Thus, the output of the powerplant is a function of the individual positions of the heliostats. The objective thus, is to optimize the positions of the heliostats such that the output of the powerplant is maximum.
In this project, a mathametical model has been created to calculate the effect of shading, blocking and cosine effect on efficiency of the power plant. The 'Sequential approximate optimization' or 'SAOi', a gradient based method is used to improve the placement of heliostats to increase the efficiency of the power plant. The optimization is a post processing step as an already setup heliostat field $PS10$ is considered and the positions are optimized to maximize the output for a specific sun position. Several testcases are considered in the project to check the validity of the generated model and the optimization algorithm.

2 The optimization problem

In heliostat field optimization, the objective is to maximize the intercepted energy, I,which is given as follows:

$$I = E \times A_{eff} \tag{1}$$

The effective area of each heliostat is reduced due to optical losses and thus, the effective area $A_{i,eff}$ for every i^{th}heliostat is given as follows:

$$A_{i,eff} = A_i \cdot (\eta_{c_i}\eta_{s_i}\eta_{b_i}\eta_{sp_i}\eta_{a_i}) \tag{2}$$

The cosine, blocking and shading efficiencies of each heliostat depends upon its location in the field and the hour of the day. Thus the total intercepted energy by the powerplant throughout the year is given as follows:

$$I = A\sum_{h=1}^{8760} E_h \left(\sum_{i=1}^{m} \eta_{c_i}\eta_{s_i}\eta_{b_i}\eta_{sp_i}\eta_{a_i}\right) \tag{3}$$

2.1 Objective function

In this project, only cosine, shading and blocking effects have been considered. Also, the intercepted energy is calculated only for a fixed sun position. Thus, the objective function for such a configuration becomes:

$$f_0(x) = -I = -AE_h\left(\sum_{i=1}^{m} \eta_{c_i}\eta_{s_i}\eta_{b_i}\right) \tag{4}$$

The position of each heliostat is determined by its x and y coordinate in the field thus leading to $2m$ design variables. These design variables lead to the construction of a vector **x** of size $2m$ for a field of m heliostats.

2.2 Constraints

There are two types of constraints that need to be taken care of:

- Each heliostat should have a certain mimimum distance between them to avoid collision
- Each heliostat should have a certain minimum distance from the Tower to avoid collision with it

In this project, as a previously constructed heliostat is being optimized, the position of heliostat is bounded by a upper and lower x and y limit. Thus, aleady care is taken that the heliostats do not hit the tower which is why, only first type of constraint is considered. Thus, the number of constraint functions arising from first type is:

$$\left(m^2 - m\right)/2$$

The minimum distance between two heliostats is given by the distance between their centers which should atleast be equal to the length of diagonal of the mirror. Thus, the constraints appear as follows:

$$-\|\left(x_i, x_{m+i}\right) - \left(x_j, x_{m+j}\right)\| + diag \leq 0$$

$$i = 1,2,3....m-1 \; and \; j = i+1, i+2, i+3....m$$

3 Modeling

The efficiency model models the cosine efficiencyη_{c_i}, shading efficiency $\eta_{s_i}$and blocking efficiency $\eta_{b_i}$for each heliostat $i = 1, 2...m$.

3.1 Efficiency modeling

The sunrays are considered incident at same angle on every heliostat, thus the incident vector is as follows:

$$V_{inc} = [-S_x, -S_y, -S_z]$$

The reflected vectors from each i^{th}heliostat depends upon the position of heliostat and the position of the receiver. Thus the reflected vector is as follows:

$$V_{ref,i} = [T_x - x_i, T_y - y_i, T_z - z_i] \; i = 1, 2...m$$

All the heliostats are considered to be on the reference ground and thus, $z_i = 0$

3.1.1 Cosine efficiency η_{c_i}

The cosine efficiency of a heliostat depends upon the sun's position $[S_x, S_y, S_z]$ and the position of the heliostat $[x_i, y_i]$. The heliostat surfact is ideally oriented such that the normal to it's surface bisects the angle between the sun's rays and the reflected ray to the tower. The effective reflection area is reduced to cosine of half of the angle of incidence.

$$\eta_{c_i} = \cos\left(\frac{1}{2}\arccos\left(\frac{-V_{ref,i}\cdot V_{inc}}{\| V_{ref,i} \|\| V_{inc} \|}\right)\right) \; ,i = 1, 2...m \tag{5}$$

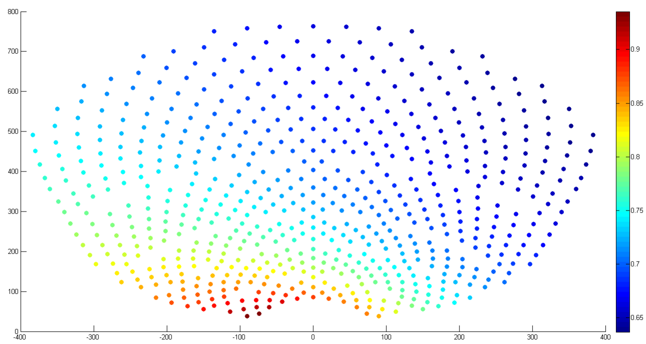


Figure 1: Cosine Efficiency plot for PS10

3.1.2 Shading efficiency η_{s_i}

The shading effects occur due to the projection of shadow of one heliostat on the reflecting surface of another heliostat. This results in the overall reduction of reflection area which therefore reduces the reflected intensity of light. The shading efficiency calculation is carried out as follows:

- In this model, an approximate but efficient shadow plotting algorithm has been used.
- Every i^{th} heliostat is checked if it lies in the domain of any of the plotted shadows for $i = 1, 2..i - 1, i + 1.m$
- If i^{th} heliostat lies fully inside a shadow, $\eta_{s_i} = 0$
- If i^{th} heliostat lies partially inside a shadow, $\eta_{s_i} = 1 - \frac{Area\,inside\,shadow}{Area\,of\,heliostat}$
- If i^{th}heliostat does not lie inside any shadow, $\eta_{s_i} = 1$

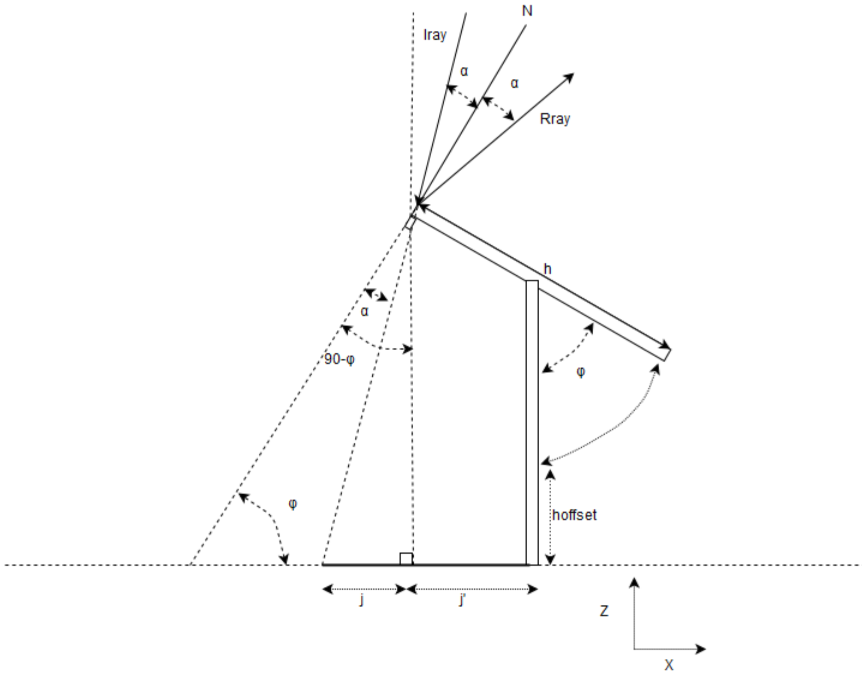


Figure 2: Shadow Calculation

$$j + j' = \tan(90 - \phi - \alpha)\left(h_{offset} + 0.5h\left(1 + \cos(\phi)\right)\right) + 0.5h\sin(\phi) \tag{6}$$

The length $j + j'$is the shadow length in the $y$coordinate. The shadow width in $x$coordinate can be calculated in a similar manner to obtain a plot of shadow domains for every heliostat

3.1.3 Blocking efficiency η_{b_i}

Blocking effect occurs when the sunlight reflected by a heliostat is blocked by another heliostat, which reduces the overall magnitude of reflected energy. The blocking efficiency is calculated as follows:

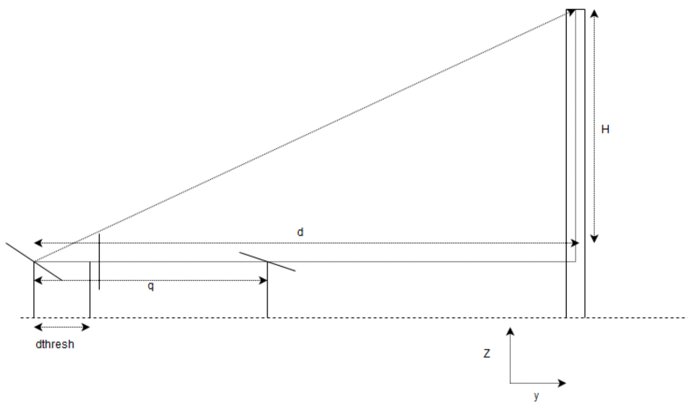


Figure 3: Blocking Calculation

- In this model, The number of heliostats lying between every i^{th} heliostat and the reciever tower are counted and checked if those heliostats lie within the threshold distance $dthresh$
- If $q > dthresh$, $\eta_{b_i} = 1$
- If $q \leq dthresh$, $\eta_{b_i} = \frac{q}{dthresh}$

4 Optimization method

4.1 A gradient based method

The problem presented above can be casted as an inequality constrained optimization problem:

$$\text{minimize } f_0(x)$$

$$\text{subject to } f_j(x) \leq 0, \quad j = 1, 2, 3...w \tag{7}$$

$$x_{i,l} \leq x_i \leq x_{i,u} \quad i = 1, 2, 3...n$$

Here, the objective is to minimize the real valued scalar objective function $f_0(x)$ while considering the constraint functions $f_j(x)$ $j = 1, 2, ...w$. These functions are dependent upon the $n = 2m$ design variables $\mathbf{x} = \{x_1, x_2...x_{2m}\}^T \in \boldsymbol{X} \subset \mathrm{R}^{\mathrm{n}}$while lying under the lower limit $x_{i,l}$ and upper limit $x_{i,u}$. Here, $x_1...x_m$ are the $x$coordinates and $x_{m+1}....x_{2m}$are the y coordinates of heliostats. The objective function is minimized in order to maximize the intercepted energy I.

4.2 SAOi Algorithm

The SAOi algorithm constructs successive approximate sub-problems $P[k], k = 1, 2, 3...$at successive iteration points $\mathbf{x}^{[k]}$which are cheaper to solve. The mimimizer solution to $P[k]$is $\mathbf{x}^{[k*]}$which is used as the initial condition for the next approximate sub-problem $P[k+1]$. The approximations and sub-problems in SAOi are as follows:

4.2.1 Diagonal quadratic approximations

SAOi constructs approximations $\underline{f}(x)$to the objective function $f_0(x)$ and the constraint functions $f_j(x)$as follows:

$$\underline{f_j}(x) = f_j^{[k]} + \sum_{i=1}^{n} \left(\frac{\partial f_j}{\partial x_i}\right)^{[k]} \left(x_i - x_i^{[k]}\right) + \frac{1}{2} \sum_{i=1}^{n} c_{2i,j}^{[k]} \left(x_i - x_i^{[k]}\right)^2 \tag{8}$$

$$j = 0 \text{ for Objective function}$$

$$j = 1, 2..w \text{ for constraints}$$

$$i = 1, 2....2m = n \text{ for design variables}$$

$f_j^{[k]}$=$f_j\left(x^{[k*]}\right)$and $c_{2i,j}^{[k]}$is the approximate second order Hessian terms. To force $\underline{f_j}(x)$to be convex, we enforce:

$$c_{2i,0}^{[k]} = max\left(\varepsilon_0 > 0, c_{2i,0}^{[k]}\right)$$

$$c_{2i,j}^{[k]} = max\left(\varepsilon_j \geq 0, c_{2i,j}^{[k]}\right) \quad j = 1, 2, ...w$$

$\varepsilon_j$is a small tolerance. Thus, the objective function $\underline{f_j}(x)$is strictly convex, while the constraint approximations $\underline{f_j}(x)$ $j = 1, 2...w$ are convex or strictly convex.

4.2.2 Estimating the partial derivatives and Hessian terms

The partial derivatives $\frac{\partial f_j}{\partial x_i}$are calculated by central difference scheme as follows:

$$\left(\frac{\partial f_j}{\partial x_i}\right)^{[k]} \approx \frac{f_j\left(x^{[k*]} + \triangle x\right) - f_j\left(x^{[k*]} - \triangle x\right)}{2\triangle x} \tag{9}$$

This is obtained by moving each heliostat by a distance of $\triangle x$ while keeping other heliostats constant and calculating $f_j\left(x^{[k*]}+\triangle x\right)$ and $f_j\left(x^{[k*]}-\triangle x\right)$. The Hessian terms $c_{2i,j}^{[k]}$are calculated by using the quadratic Taylor series expansion to the reciprocal approximation :

$$c_{2i,j}^{[k]}=\frac{\partial^2 f_j}{\partial x_i^2}\left(x^{[k*]}\right)\approx\frac{-2\left(\frac{\partial f_j}{\partial x_i}\right)^{[k]}}{x_i^{[k*]}} \tag{10}$$

4.2.3 SAOi Algorithm

With an initial point $x^{[0]}$to start with, the SAOi algorithm proceeds as follows:

1. **Initialization:** Select the constants $\epsilon_1,\epsilon_2,\epsilon_x,k_{stop},\chi_1>1,\chi_2>1$;Set the outer iterator $k:=0$ and inner iterator $l:=0$
2. **Simulation and sensitivity analysis:** Compute $f_j\left(x^{[0]}\right)$,$\nabla f_j\left(x^{[0]}\right)$, $j=0,1,2...w$
3. **Construct approximate Hessian terms:** Calculate the Hessian terms $c_{2i,0}^{[k]}>0$ and $c_{2i,j}^{[k]}\geq 0,\ j=1,2...w$
4. **Approximate optimization:** Construct local approximate sub-problem $P\left[k\right]$at $x^{[k]}$and solve this to arrive at $\left(x^{[k*]},\lambda^{[k*]}\right)$
5. **Simulation analysis:** Calculate $f_j\left(x^{[k*]}\right)$, $j=0,1,2...w$
6. **Test if $x^{[k*]}$is acceptable:** If satisfied, GOTO step 8; else CONTINUE
7. **Initiate an inner loop:**
   a) Set $l:=l+1$
   b) If $\underline{f_0}\left(x^{[k*]}\right)<\left(f_0\left(x^{[k*]}\right)+\epsilon_1\right)$,set $c_{2i,0}^{[k]}:=\chi_1 c_{2i,0}^{[k]}$
   c) If $\underline{f_j}\left(x^{[k*]}\right)<\left(f_j\left(x^{[k*]}\right)+\epsilon_2\right)$,set $c_{2i,j}^{[k]}:=\chi_2 c_{2i,j}^{[k]}$, $j=0,1...w$
   d) GOTO step 4
8. **Move to new iterate:** Set $x^{[k+1]}:=x^{[k*]}$
9. **Convergence test:** If $\|x^{[k+1]}-x^{[k]}\|\leq\epsilon_x$, OR $k=k_{stop}$, STOP
10. **Simulation outer loop:** Calculate $\nabla f_j\left(x^{[k+1]}\right)$, $j=0,1...w$
11. **Initiate an additional outer loop:** Set $k:=k+1$, GOTO step 3

4.2.4 Optimization Solver (BFGS)

Broyden-Fletcher-Goldfarb-Shanno or the BFGS method is a hill-climbing optimization method that seeks a stationary point of a function. Other line search methods like the Conjugate gradient requires the evaluation of a Hessian matrix which is computationally a heavy load task. BFGS uses approximations for Hessian matrix and updates it after every iteration.

1. Choose initial guess for the optimization point $x^{[0]}$and tolerances $\varepsilon_1,\varepsilon_2,\varepsilon_3$
2. Set $G^{[0]}=I$, where $G^{[0]}$is the initial approximation to the Hessian matrix
3. Iterate over $k=1,2,3...p$:
   a) Set $x^{[k]}=x^{[k-1]}+\lambda^{[k]}u^{[k]}$, where $u^{[k]}=-G^{[k-1]}\nabla f\left(x^{[k-1]}\right)$and $\lambda^{[k]}$is such that $f\left(x^{[k-1]}+\lambda^{[k]}u^{[k]}\right)=min\, f\left(x^{[k-1]}+\lambda^{[k]}u^{[k]}\right),\lambda^{[k]}\geq 0$ (line search)
   b) Test for the convergence criteria:
   if $\|x^{[k]}-x^{[k-1]}\|<\varepsilon_1$or $\|\nabla f\left(x^{[k]}\right)\|<\varepsilon_2$or $|f\left(x^{[k]}\right)-f\left(x^{[k-1]}\right)|<\varepsilon_3$then STOP and $x^*\backsimeq x^{[k]}$else GOTO step 3c

c) Set $v^{[k]} = \lambda^{[k]} u^{[k]}$ and set $y^{[k]} = \nabla f\left(x^{[k]}\right) - \nabla f\left(x^{[k-1]}\right)$

d) Compute $G^{[k]}$used for new descent direction:

$$G^{[k-1]} + \left[1 + \frac{y^{[k]T} G^{[k-1]} y^{[k]}}{v^{[k]T} y^{[k]}}\right] \left[\frac{v^{[k]} y^{[k]T}}{v^{[k]T} y^{[k]}}\right] - \left[\frac{v^{[k]} y^{[k]T} G^{[k-1]} + G^{[k-1]} y^{[k]} v^{[k]T}}{v^{[k]T} y^{[k]}}\right]$$

4. Set $x^{[0]} = x^{[p]}$ and $G^{[0]} = I$and GOTO step 3

5 Test Cases

5.1 Optimization test cases

One mathematical test case was considered to check the optimization algorithm before using it on the efficiency model.

5.1.1 Svanberg's 5-variate cantilever beam

A weight minimization problem proposed by Svanberg with five design variables and subject to single constraint expressed as follows:

$$min\, f_0\left(x\right) = c_1 \sum_{i=1}^{5} x_i$$

$$subject\,to\, f_1\left(x\right) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - c_2 \leq 0,$$

$$0 < x_i,\, i = 1, 2..5$$

$$c_1 = 0.0624 \text{ and } c_2 = 1.0$$

In this problem, the initial and final data after optimization were as follows:

$$x^{[0]} = [5, 5, 5, 5, 5]\, x_l = [1, 1, 1, 1, 1]\, x_u = [10, 10, 10, 10, 10]$$

$$f_0\left(x^{[0]}\right) = 1.5600$$

$$f_0\left(x^{[k*]}\right) = 1.3396$$

$$x^{[k*]} = [6.0144, 5.3078, 4.4932, 3.5006, 2.1521]$$

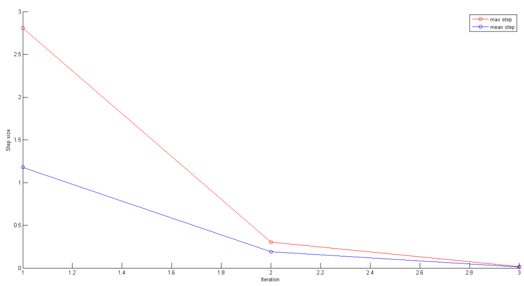


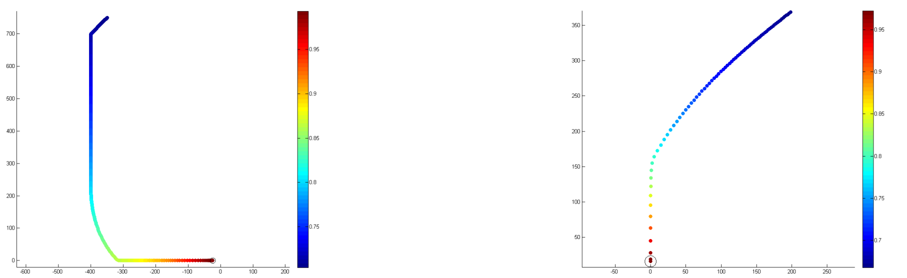
Figure 4: Svanberg $(stepsize\, vs.\, Iteration)$

5.2 Modeling test cases

Six test cases were considered to test the optimization algorithm together with the efficiency model. For these test cases, the sun position vector was set $[S_x, S_y, S_z] := [1, 1, 4]$, The tower vector $[T_x, T_y, T_z] := [0, 0, 100]$ and the width , height and ground offset was considered to be 12.84m, 9.45m and 2m respectively.

5.2.1 One Heliostat

In this test case, Only one heliostat is placed at two random locations in the field. The optimization algorithm is run with no local position constraint so as to get the optimum position for maximum energy output



$I_{rec} := 85.4717 \rightarrow 120.4801$
423$iterations$ 270.066 s

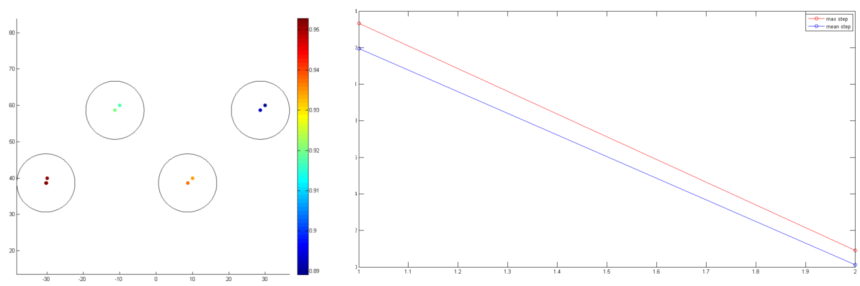
$I_{rec} := 81.0459 \rightarrow 117.9422$
79$iterations$ 19.365s

Figure 5: One Heliostat

As seen from the figures, The heliostat position iterates so as to increase it's cosine efficiency η_c (Because a single heliostat would not have shading or blocking effects).

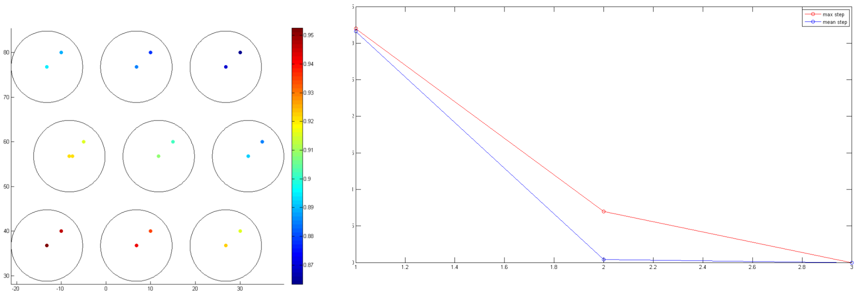
5.2.2 n×n Heliostat layout

In this test case, an $n \times n$ combination of Heliostats has been considered. The optimization has been run with a local position constraint for local optimization of the heliostat field



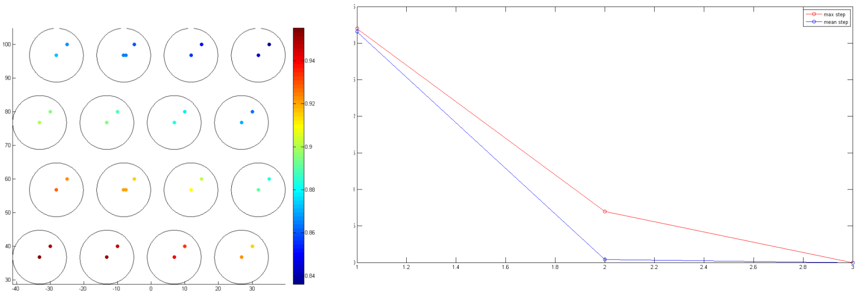
$I_{rec} := 448.0295 \rightarrow 448.2035$

Figure 6: $2 \times 2\, Heliostat\, Layout$



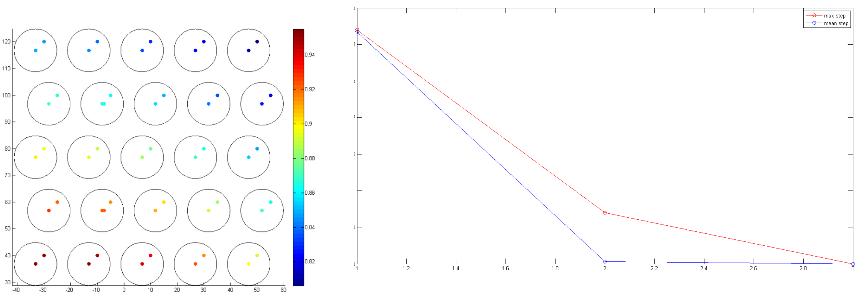
$I_{rec} := 986.2209 \rightarrow 993.4609$

Figure 7: 3×3 $Heliostat\ Layout$



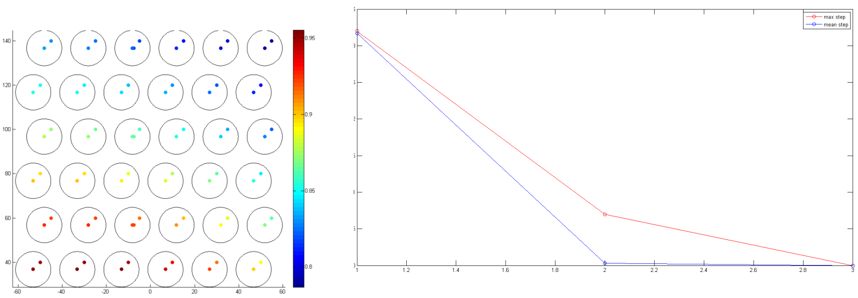
$I_{rec} := 1.7253e+03 \rightarrow 1.7367e+03$

Figure 8: 4×4 $Heliostat\ Layout$



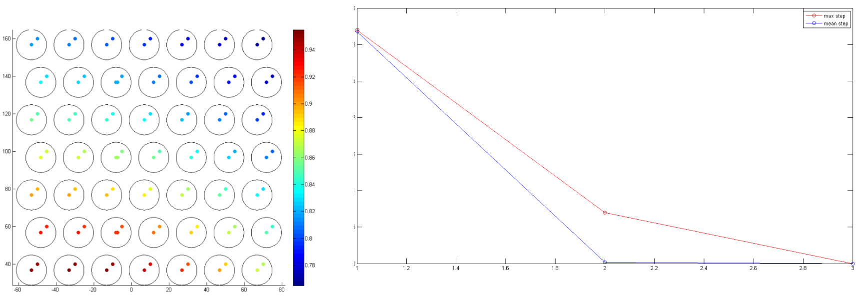
$I_{rec} := 2.6536e+03 \rightarrow 2.6714e+03$

Figure 9: 5×5 $Heliostat\ Layout$



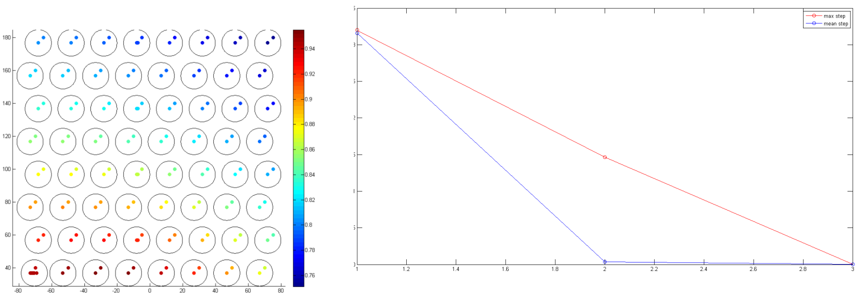
$I_{rec} := 3.7884e+03 \rightarrow 3.8116e+03$

Figure 10: 6×6 $Heliostat\ Layout$



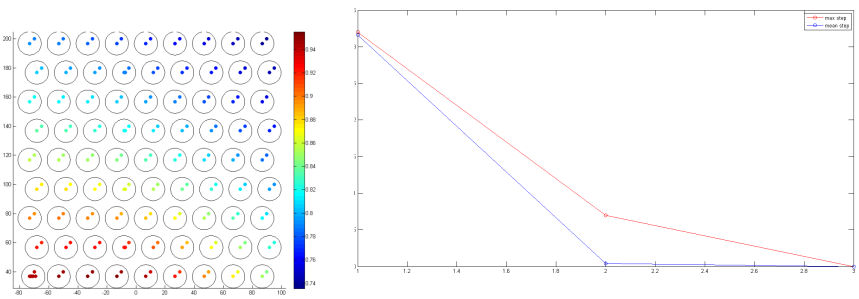
$$I_{rec} := 5.0566e + 03 \rightarrow 5.0870e + 03$$

Figure 11: 7×7 *Heliostat Layout*



$$I_{rec} := 6.5570e + 03 \rightarrow 6.5932e + 03$$

Figure 12: 8×8 *Heliostat Layout*



$$I_{rec} := 8.1577e + 03 \rightarrow 8.2015e + 03$$

Figure 13: 9×9 *Heliostat Layout*

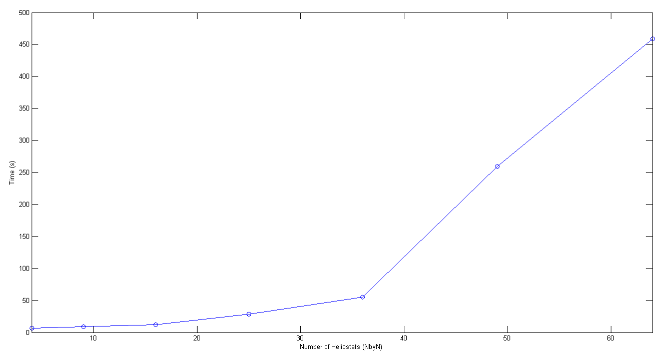
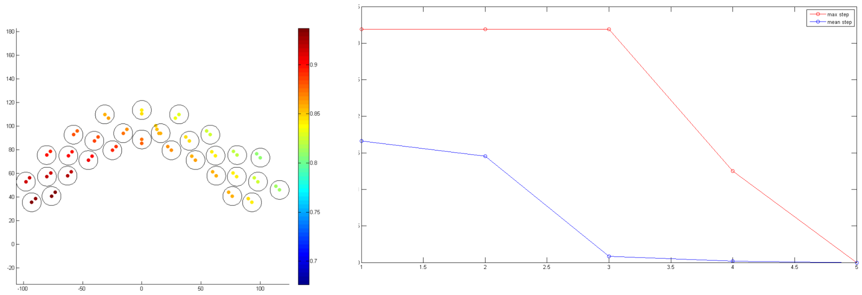


Figure 14: Runtime vs. Heliostats

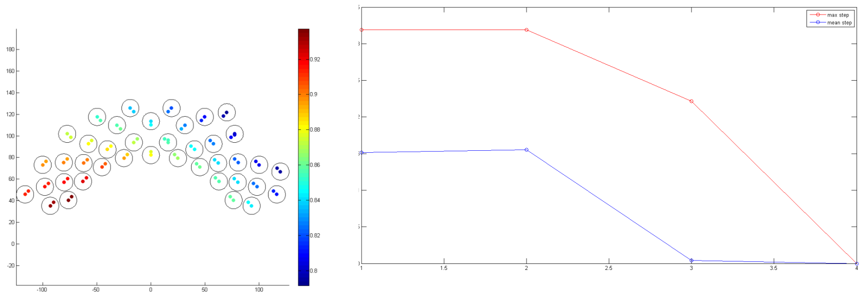
5.2.3 PS-10

In this test case, the heliostat field of PS-10 thermal power plant has been used. The number of heliostats have been varied from 30 to 70 and the power output was maximized



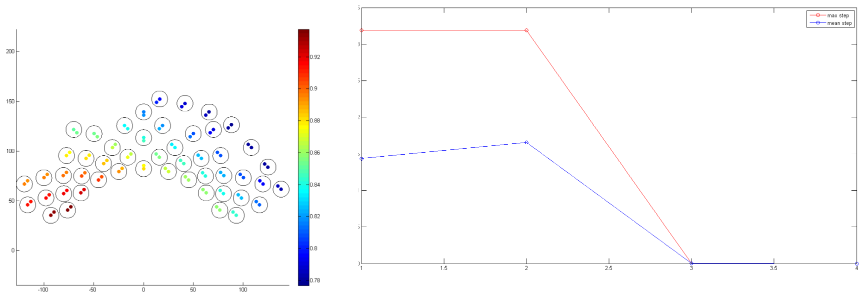
$I_{rec} := 3.1560e + 03 \rightarrow 3.1591e + 03$

Figure 15: 30 $Heliostats$



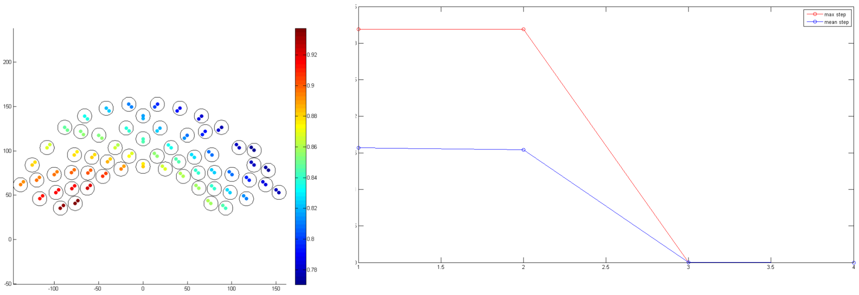
$I_{rec} := 4.1768e + 03 \rightarrow 4.1778e + 03$

Figure 16: 40 $Heliostats$



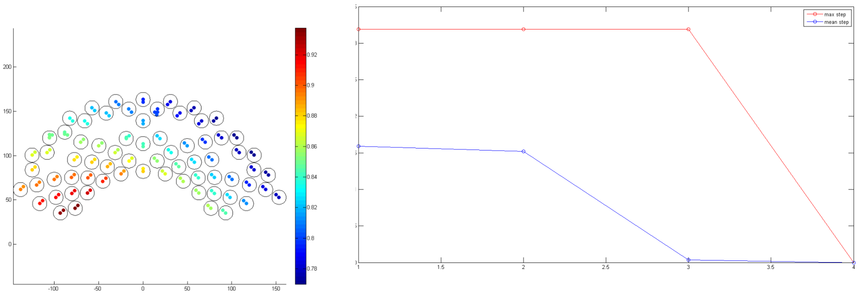
$I_{rec} := 5.1559e + 03 \rightarrow 5.1576e + 03$

Figure 17: 50 $Heliostats$



$$I_{rec} := 6.1607e+03 \rightarrow 6.1618e+03$$

Figure 18: 60 $Heliostats$

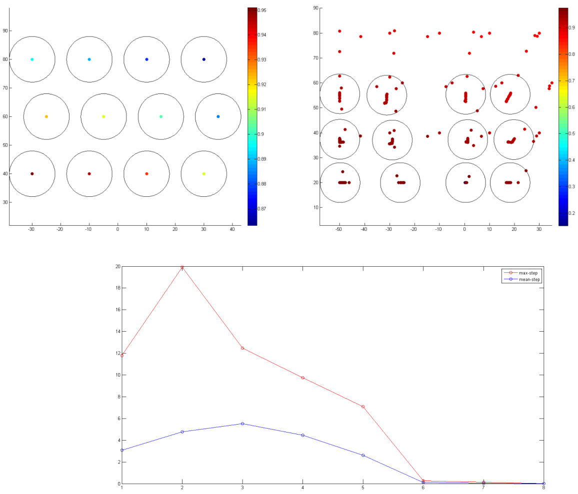


$$I_{rec} := 7.1424e+03 \rightarrow 7.1467e+03$$

Figure 19: 70 $Heliostats$

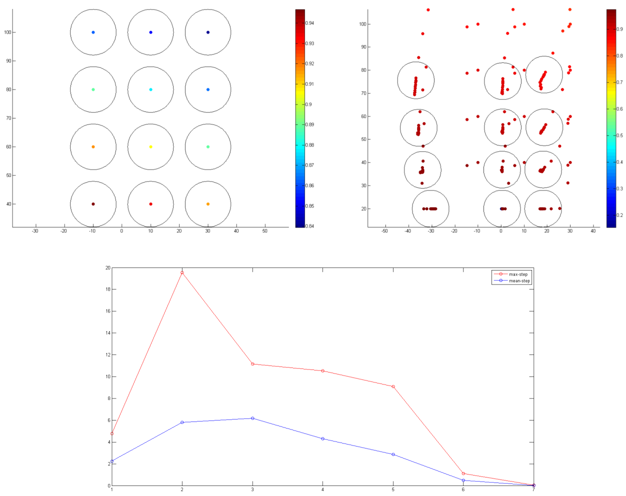
5.2.4 Local Optimization

To prove that the above algorithm is a locally optimim algorithm, sets of 3×4 and 4×3 heliostat matrix is considered. For a globally optimum algorithm, the final positions of heliostats is expected to be same, but that does not happen as seen in the figures which confirms that the problem is locally optimized.



$$I_{rec} := 1.3221e + 03 \to 1.3746e + 03$$

Figure 20: $3 \times 4\, Matrix$



$$I_{rec} := 1.2974e + 03 \to 1.3528e + 03$$

Figure 21: $4 \times 3\, Matrix$

6 Conclusion

- A sufficiently easy and fast model has been developed to calculate shadowing efficiency, blocking efficiency and cosine efficiency
- Although some rough approximations are considered in the model, it works perfectly for different sun positions
- A gradient based algorithm SAOi has been implemented and validated with optimization test cases
- The model is validated with several hypothetical test cases like single heliostat, $N \times N$ heliostat matrix, excerpts from PS10 layout
- The SAOi method is seen to be locally optimum using a test case
- BFGS algorithm is preferred over Conjugate gradient as it requires less computational efforts for calculating the Hessian matrix and thus has less runtime
- The scope of the project is a post processing step and thus an already setup field layout PS10 is considered and the heliostats are optimized within a small range of $x_i \pm \frac{heliostat\, diag}{4}$
- An exponential rise in runtime is seen with increase in the number of heliostats, this could be redued by parallelizing the efficiency model

References

[1] LFP Etman, Albert A Groenwold, and JE Rooda. First-order sequential convex programming using approximate diagonal qp subproblems. *Structural and Multidisciplinary Optimization*, 45(4):479–488, 2012.

[2] Albert A Groenwold and LFP Etman. Saoi: an algorithm for very large scale optimal design. In *Proc. ninth world congress on structural and multidisciplinary optimization, paper*, volume 35, 2011.

[3] Narayanan Krishnamurthy. Optimization methods the unconstrained extremization problem.

[4] 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.

[5] Corey J Noone, Manuel Torrilhon, and Alexander Mitsos. Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy*, 86(2):792–803, 2012.

[6] Surika Vosloo. *Analytical modelling and optimization of a thermal convective microfluidic gyroscope*. PhD thesis, University of Stellenbosch, 2010.