

# LOCAL SEARCH BASED OPTIMIZATION OF A SPATIAL LIGHT DISTRIBUTION MODEL

---

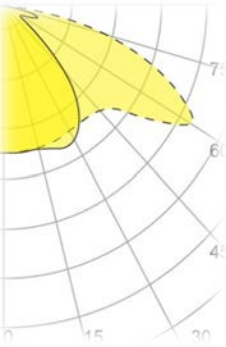
David Kaljun , Janez Žerovnik

*FS, University of Ljubljana, Aškerceva 6, 1000 Ljubljana, Slovenia*

BIOMA 2014

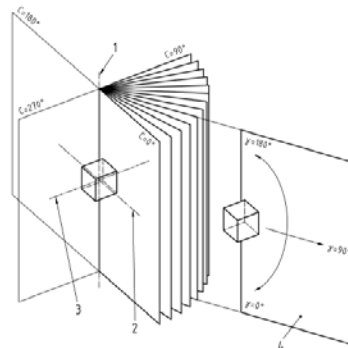
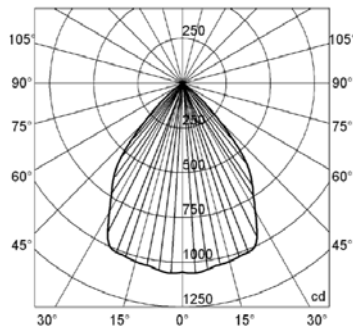
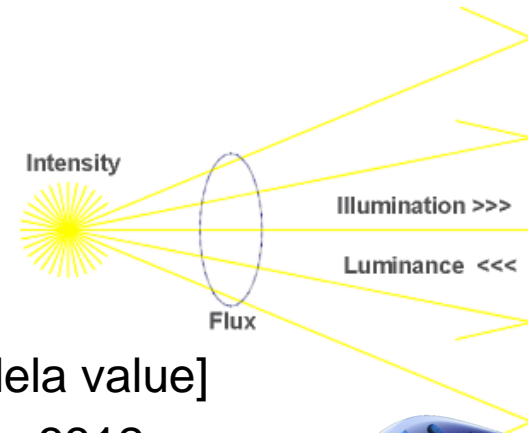
*13 September 2014, Ljubljana, Slovenia*

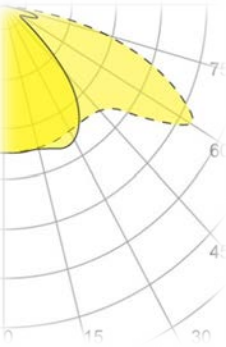




# Introduction to photometry

- Photometry is the science of the measurement of light, in terms of its perceived brightness to the human eye.
- We distribute photometric data with standard file types .ies and .ldt.
- The files contain general information about the measured source and a set of vectors written in spherical coordinates [horizontal angle, polar angle, candela value]  
Typical number of vectors for an asymmetric distribution is 3312.





# Problem definition

- **Global research objective :**

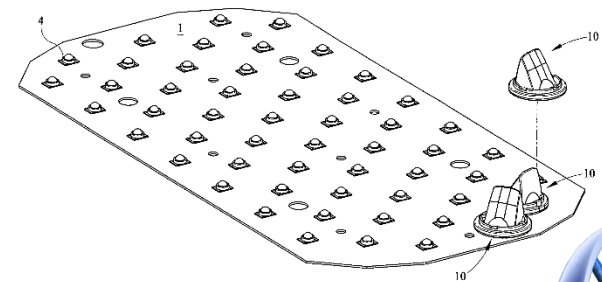
- To define a method for goal driven optimization of the luminaire photometry. The goal of the method is to define the combination and position of secondary optical elements on a LED array in a way that satisfies the user demands on the arrays end photometry.

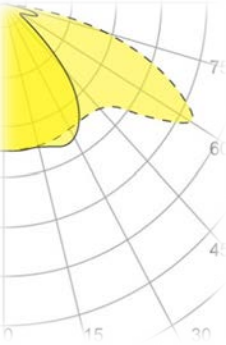
- **Prerequisite for an efficient method:**

- Low number of parameters to be optimized
- Fast and adaptive algorithms

- **Problem at hand:**

- To drastically reduce the number of parameters needed to describe the spatial light distribution (photometry) by fitting a function to the measured data.





# Analytical model

Ivan Moreno and Ching-Cheng Sun, *Modeling the radiation pattern of leds*, OPTICS EXPRESS 1808, Vol. 16, No. 3, Februar 2008

- Proposed by Moreno and Sun in 2008 for describing the spatial light distribution of a LED without mounted secondary optical elements.

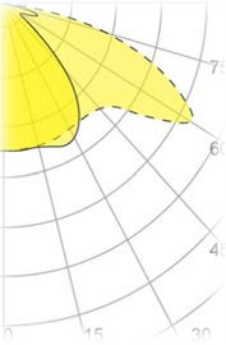
$$f(\varphi) = a * \cos(|\varphi| - b)^c \quad (\text{basic model})$$

$$f(\varphi) = \sum_1^i L_{max} * a_i * \cos(|\varphi| - b_i)^{c_i} \quad (\text{enhanced model})$$

$L_{max}$	...	Max. luminous intensity (cd)
$a_i, b_i, c_i$	...	function parameters
$\varphi$	...	polar angle

- One function with 10 parameters per c-plane is enough to appropriately describe the spatial distribution of a source. This in fact reduces the parameter count up to 80% (3312 vectors apposed to 720 function parameters)





# Analytical model

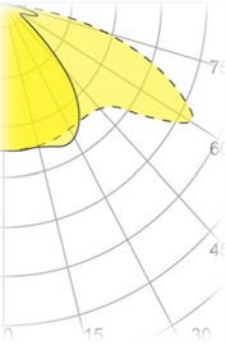
- Good fit definition

- Good fit is defined by the value of the RMS error. The RMS error for a sufficiently accurate fit must be less than 5% on every c-plane, because the best measuring tools are methods known to allow up to 2% noise in data but most of the measured data is measured at a tolerance of  $\pm 7\%$ . Therefore, the target results of the fitting algorithms are at less than 5% RMS error, but at the same time there is no practical need for less than 1% or 2% RMS error.

- The RMS evaluation function

$$RMS = \sqrt{\frac{1}{M} \sum_{i=1}^M [L(\varphi_i) - f(\varphi_i)]^2}$$

- $M$  ... number measurements taken at different polar angles on a c-plane
- $L(\varphi)$  ... measured luminous intensity at the polar angle  $\varphi$
- $f(\varphi)$  ... calculated luminous intensity at the polar angle  $\varphi$  with the current parameter set

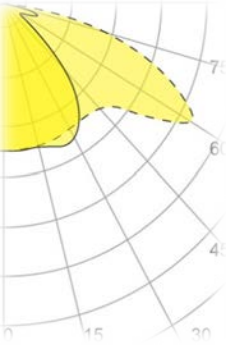


# Solution to the given problem

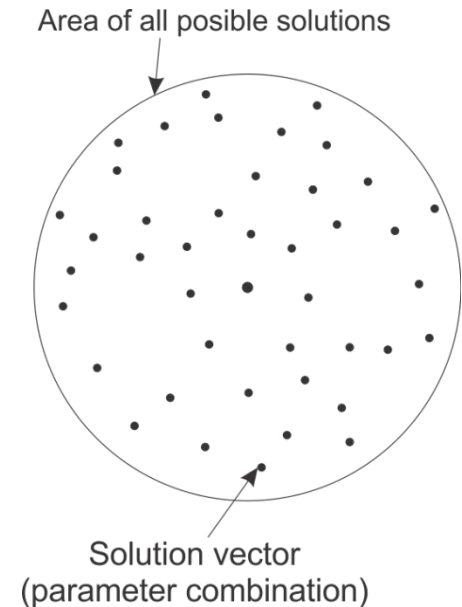
- Provide a set of function parameters that represent an accurate fit of the model function presented before to the measured data of the spatial light distribution of a LED light source with mounted secondary optical element.
- The above can be achieved with a variety of optimization algorithms. The trick here is to choose the most appropriate algorithms.
- To determine the appropriateness of the algorithms we have set-up an experiment that shows the advantages or disadvantages of the compared algorithms.



# The experiment



- We compare 6 different algorithms
- With the same pool of possible solutions
  - $a_i = \{0, 0.001, 0.002, \dots, 1\}$
  - $b_i = \{-90, -89.9, -89.8, \dots, 90\}$
  - $c_i = \{0, 1, 2, \dots, 100\}$



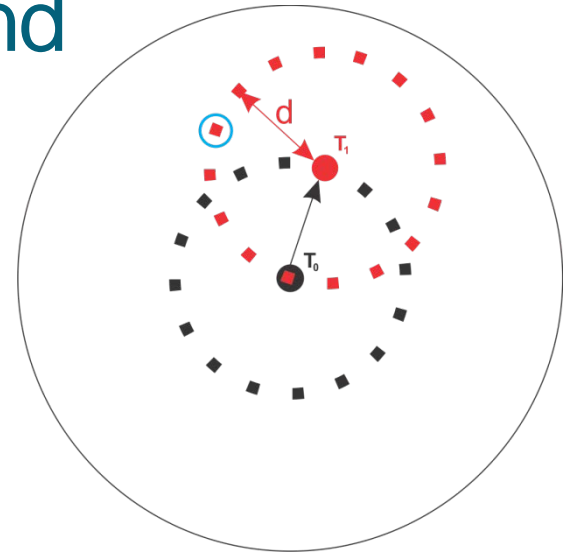
- All algorithms run for four million calculating iterations  
(one calculating iteration is when the algorithm assesses the RMS error, because 95% of the execution time is spent on estimating the error and 5% are spent on other functions)
- Algorithms save a log entry at every 100-th iteration
- The code is written in c++ (not optimized)
- Execution time for one approximation is 30 minutes  
(measured on a Intel CORE-I3 4130 @ 3,6 Ghz)



# Algorithms

## Steepest descend

1. Defines a fixed neighborhood with step  $+d$  &  $-d$
2. Checks all 512 possible solution with this step
3. Moves to the best one and starts from [1.]
4. If no better solution than the current one is found it manipulates the neighborhood with a factor  $g$  ( $g*d$ ) and starts from [1.]
5. It runs for four million iterations.

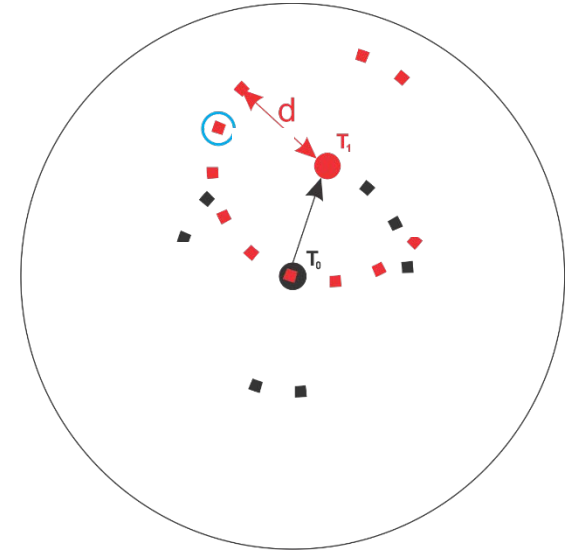




# Algorithms

## Iterative improvement with fixed neighborhood

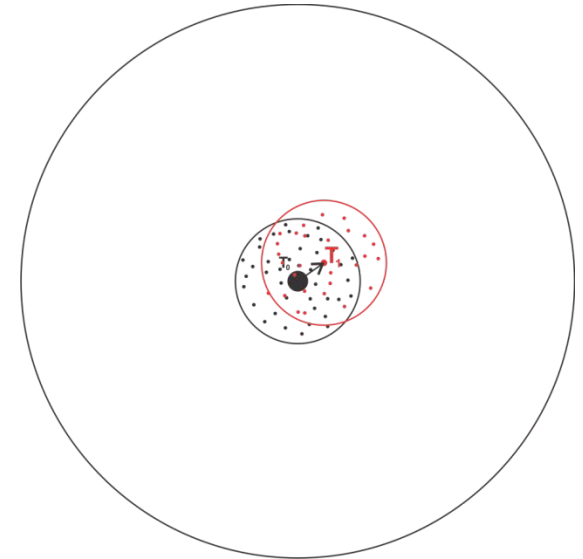
1. Defines a fixed neighborhood with step  $+d$  &  $-d$ .
2. Starts checking possible solutions with this step and as soon it finds a better solution it breaks and moves to that solution.
3. Next it starts from [1.] at the new solution.
4. If no better solution than the current one is found it manipulates the neighborhood with a factor  $g$  ( $g*d$ ) and starts from [1.].
5. It runs for four million iterations.



# Algorithms

## Iterative improvement with variable neighborhood

1. Defines a variable neighborhood with  $+d$  &  $-d$ .
2. Starts checking possible solutions with a random step that is inside the variable neighborhood and as soon it finds a better solution it breaks and moves to that solution.
3. Next it starts from [1.] at the new solution.
4. If no better solution than the current one is found within 1000 iterations it manipulates the neighborhood with a factor  $g$  ( $g*d$ ) and starts from [1.].
5. It runs for four million iterations.

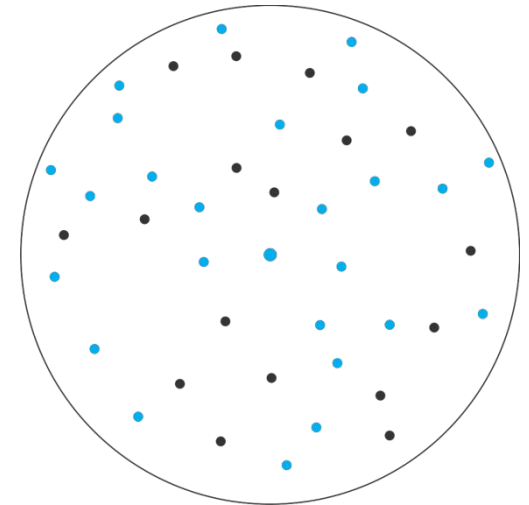


# Algorithms

## Standard genetic algorithm

John McCall, Genetic algorithms for modeling and optimization, Journal of Computational and Applied Mathematics 184, 205-222, 2005

1. Generates the initial population in size  $P$  and calculates the RMS errors for each entity.
2. Sorts the current generation from the best to the worst.
3. Cross-Breeds the entities in the current generation to generate the next generation in size of  $P$  in a way that every pair of the parent entities generates two children that inherit the genes from both parents according to the cross point. Better parents are more likely to be chosen as bad ones.
4. Randomly mutates a random number of entities of the new generation.
5. Calculates the RMS errors for the new generation. If the generation limit is not achieved it continues from [2.] otherwise it stops.
6. It runs for four million iterations. The number of generations is calculated according to the number of population  $P$ .

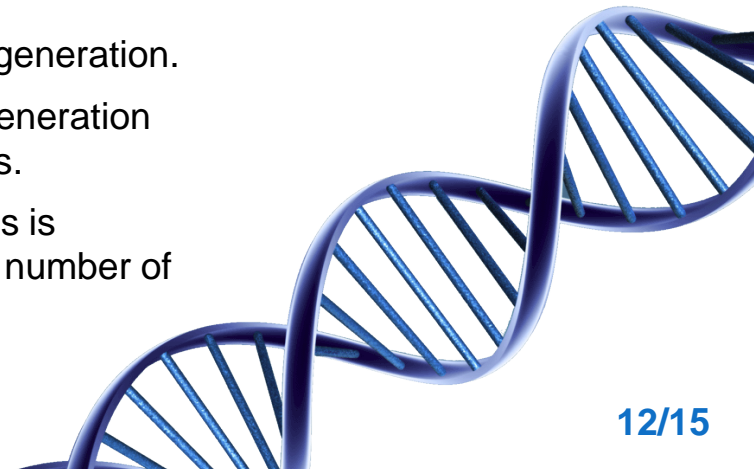
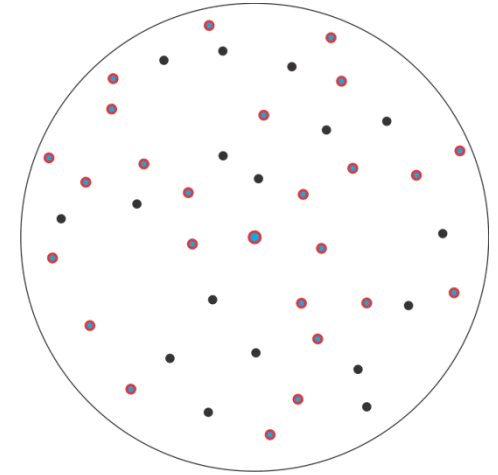


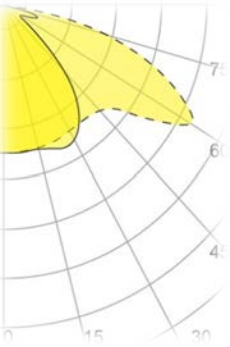


# Algorithms

## Hybrid genetic algorithm

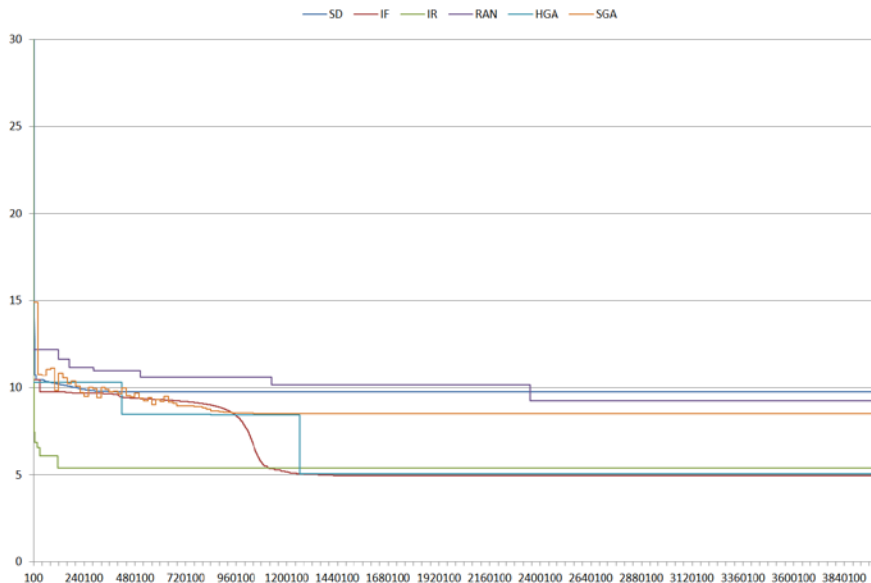
1. Generates the initial population in size  $P$  and calculates the RMS errors for each entity.
2. Sorts the current generation from the best to the worst.
3. Locally optimizes 10 best entities from the current solution with  $x$  number of iterations.
4. Cross-Breeds the optimized entities in the current generation to generate the next generation in size of  $P$  in a way that every pair of the parent entities generates children that inherit the genes from both parents according to a random cross point.
5. Randomly mutates a random number of entities of the new generation.
6. Calculates the RMS errors for the new generation. If the generation limit is not achieved it continues from [2.] otherwise it stops.
7. It runs for four million iterations. The number of generations is calculated according to the number of population  $P$  and the number of optimization iterations  $x$ .





# Results

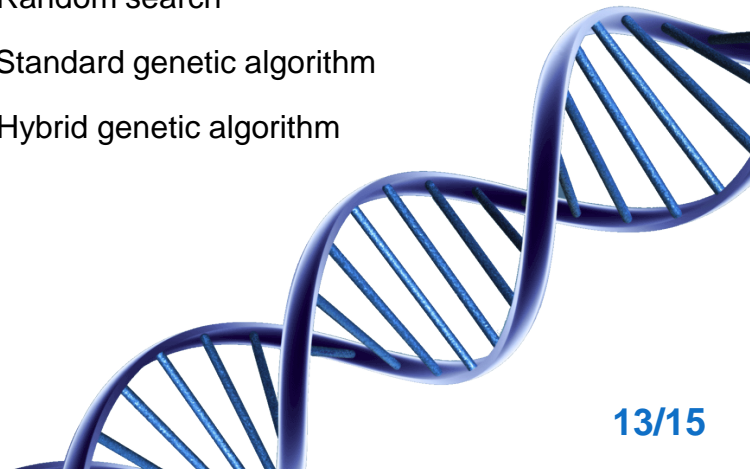
1. Almost all algorithms achieve appropriate results.
2. The winner in quantity of best results is IF followed by HGA.
3. IF also provided solutions with the best quality.
4. As expected RAN is not competitive.
5. All have a very steep convergence curve.

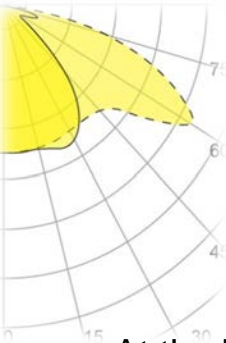


Lens/Alg.	SD	IF	RAN	IR	HGA	SGA
C13353	9,757	<b>4,942</b>	9,243	5,389	<b>5,076</b>	8,531
CA11265	2,775	<b>2,372</b>	4,936	4,798	<b>2,729</b>	4,259
CA11268	<b>2,227</b>	<b>2,229</b>	4,1	2,471	2,578	2,742
CA11483	<b>3,1</b>	<b>3,066</b>	4,13	3,387	3,141	3,867
CA11525	3,15	<b>1,108</b>	3,217	1,907	<b>1,087</b>	2,175
CA11934	3,94	<b>2,514</b>	4,196	3,543	<b>2,909</b>	3,346
CA12392	<b>1,636</b>	<b>1,641</b>	3,424	2,445	2,277	2,395
CA13013	1,202	<b>0,695</b>	2,136	2,241	<b>0,916</b>	0,932
CP12632	5,537	5,493	4,918	4,974	<b>4,362</b>	<b>4,681</b>
CP12633	2,431	<b>2,415</b>	4,063	3,708	<b>2,347</b>	2,496
CP12636	<b>2,348</b>	<b>2,107</b>	4,571	4,217	2,479	4,299
FP13030	<b>2,267</b>	<b>2,257</b>	3,762	3,659	2,414	2,749

RMS error after four million iterations.

- SD – Steepest descend
- IF – Iterative improvement fixed neighborhood
- IR – Iterative improvement random neighborhood
- RAN – Random search
- SGA – Standard genetic algorithm
- HGA – Hybrid genetic algorithm





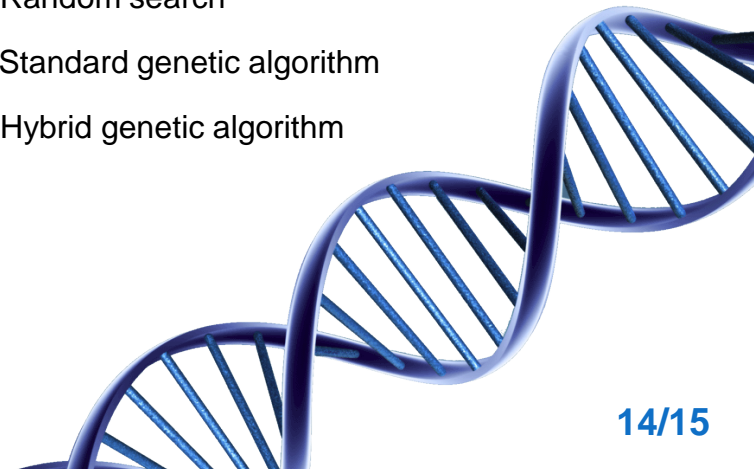
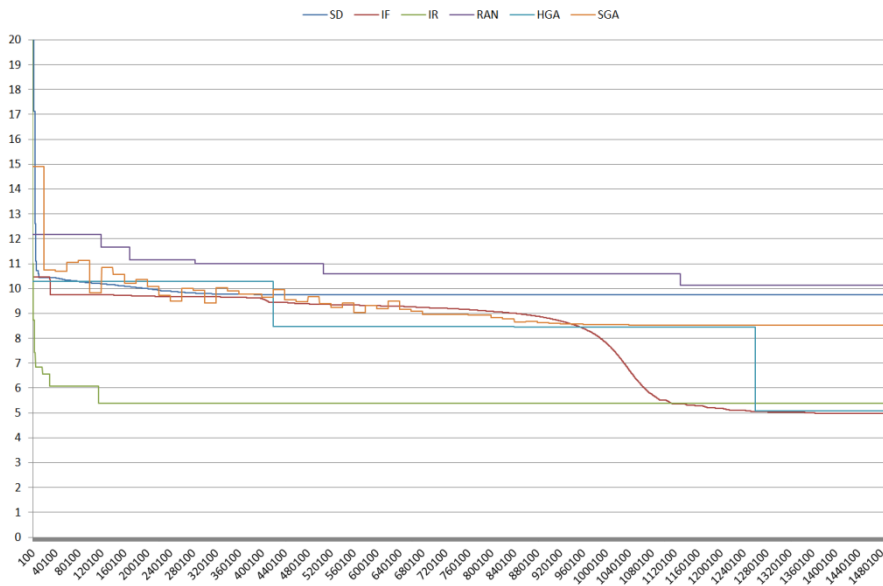
# Results

1. At the lower number of iterations the HGA is the clear winner in both quality and quantity.
2. IF struggles in second together with SD.
3. As expected RAN is not competitive, but it does find a best solution in one case.
4. As before the convergence curve is steep.

Lens/Alg.	SD	IF	RAN	IR	HGA	SGA
C13353	9,757	9,167	10,950	<b>5,389</b>	<b>8,477</b>	8,966
CA11265	<b>3,477</b>	<b>2,7</b>	7,282	5,073	4,183	5,883
CA11268	<b>2,376</b>	2,62	5,893	<b>2,471</b>	2,932	2,996
CA11483	4,181	<b>3,4</b>	4,13	3,784	<b>3,641</b>	4,027
CA11525	3,813	<b>3,395</b>	4,811	3,789	<b>1,601</b>	2,175
CA11934	4,032	<b>1,662</b>	4,988	3,543	<b>3,789</b>	4,473
CA12392	<b>1,814</b>	<b>1,661</b>	3,597	2,717	2,577	3,867
CA13013	2,804	3,115	<b>2,136</b>	2,241	<b>1,331</b>	3,558
CP12632	9,501	9,839	8,474	<b>5,054</b>	<b>4,703</b>	5,474
CP12633	<b>2,465</b>	4,511	4,757	4,296	<b>2,613</b>	3,918
CP12636	5	6,297	5,506	<b>4,217</b>	<b>3,803</b>	4,590
FP13030	2,8	5,679	6,611	3,659	<b>3,233</b>	5,363

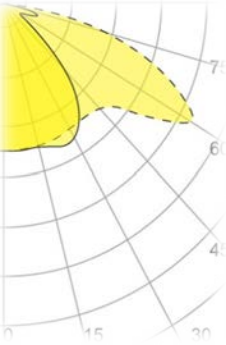
RMS error after 750.000 iterations.

- SD – Steepest descend
- IF – Iterative improvement fixed neighborhood
- IR – Iterative improvement random neighborhood
- RAN – Random search
- SGA – Standard genetic algorithm
- HGA – Hybrid genetic algorithm

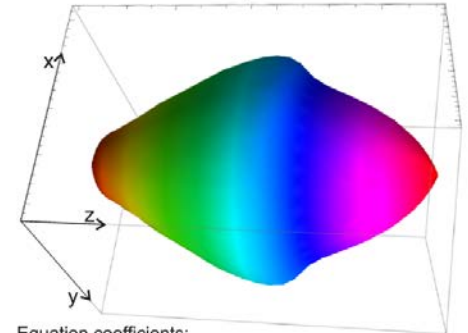
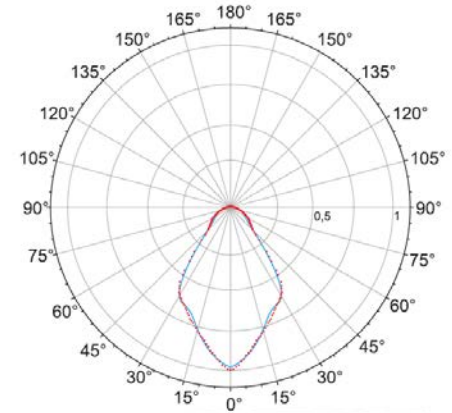




# Conclusion

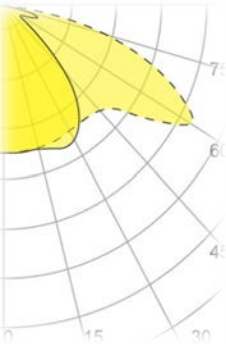


- We designed several algorithms and tested them on real lens data.
- The results show that all except the random search algorithms produce acceptable solutions.
- The genetic algorithms were very competitive, but we have to note that the one with infused local optimization performed better.
- The experiment presented here gave important information about the number of complexity of solving the general problem in the case of instances with symmetric spatial light distribution.
- Future work includes adaptation of the model to lenses with asymmetric spatial light distribution, and definition of a general model.
- The general model will presumably include a larger number of parameters which in turn most probably means larger search spaces and more challenging optimization problems.



Equation coefficients:  
 $a_1$  1,00973    $b_1$  -6,5697°    $c_1$  3,5295  
 $a_2$  0,07883    $b_2$  64,1715°    $c_2$  27,7455  
 $a_3$  0,17075    $b_3$  30,8478°    $c_3$  102,3540





# References

- Randy L. Haupt and Sue Ellen Haupt, . *PRACTICAL GENETIC ALGORITHMS*, John Wiley & Sons, Inc., 2 edition, 2004. ISBN 0-471-45565-2
- Hongming Yang, JanW. M. Bergmans, Tim C. W. Schenk, Jean-Paul M. G. Linnartz, Ronald Rietman, *An analytical model for the illuminance distribution of a power LED*, Optical Society of America, 2008
- Ivan Moreno, *LED intensity distribution*, In *The International Optical Design Conference 2006*, Optical Society of America , paper TuD6, 2006
- Ivan Moreno and Ching-Cherng Sun, *Modeling the radiation pattern of leds*, OPTICS EXPRESS 1808, Vol. 16, No. 3, Februar 2008
- John McCall, Genetic algorithms for modeling and optimization, *Journal of Computational and Applied Mathematics* 184, 205-222, 2005
- Ledil Oy. Salorankatu 10, -24240 salo, Finland, <http://www.ledil.com/>, 2014
- Randy L. Haupt and Sue Ellen Haupt, . *PRACTICAL GENETIC ALGORITHMS*, John Wiley & Sons, Inc., 2 edition, 2004. ISBN 0-471-45565-2

