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
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.
• **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

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

\[
f(\varphi) = a \cdot \cos(|\varphi| - b)^c \] (basic model)

\[
f(\varphi) = \sum_{i=1}^i L_{max} \cdot a_i \cdot \cos(|\varphi| - b_i)^{c_i} \] (enhanced model)

- 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 and methods known allow up to 2% noise in data but most of the measured data is measured at a tolerance of ±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

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

- \(M\) … number measurements taken at different polar angles on a c-plane
- \(L(\phi)\) … measured luminous intensity at the polar angle \(\phi\)
- \(f(\phi)\) … calculated luminous intensity at the polar angle \(\phi\) 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 chose the most appropriate algorithms.

• To determine the appropriates of the algorithms we have set-up an experiment that show the advantages or disadvantages of the compared algorithms.
The experiment

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

- 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

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-Breads 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 an 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-Breads 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. Random 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 an 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.

### RMS error after four million iterations.

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

- 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.

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

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.
References

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