Performance of Optimization Algorithms

Design Optimization
TMKT48

LINKÖPING UNIVERSITY

Performance of optimizations

- Different optimization algorithms are best for different problems
- We want to compare the algorithms
  - Can help us to choose optimization algorithm
  - Can be used to optimize the parameters of the optimization algorithm – Meta-optimization

Benchmarking Optimization Algorithms

- You want to choose the method that you think is best for your problem

1. Try to optimize simple mathematical functions that you think are similar to the problem that you really want to optimize
   - Linear/non-linear, uni-/multimodal, … ?

2. Optimize the mathematical functions numerous times with each candidate optimization algorithm

3. Order the optimization algorithms according to accuracy, robustness, efficiency, user friendliness etc.

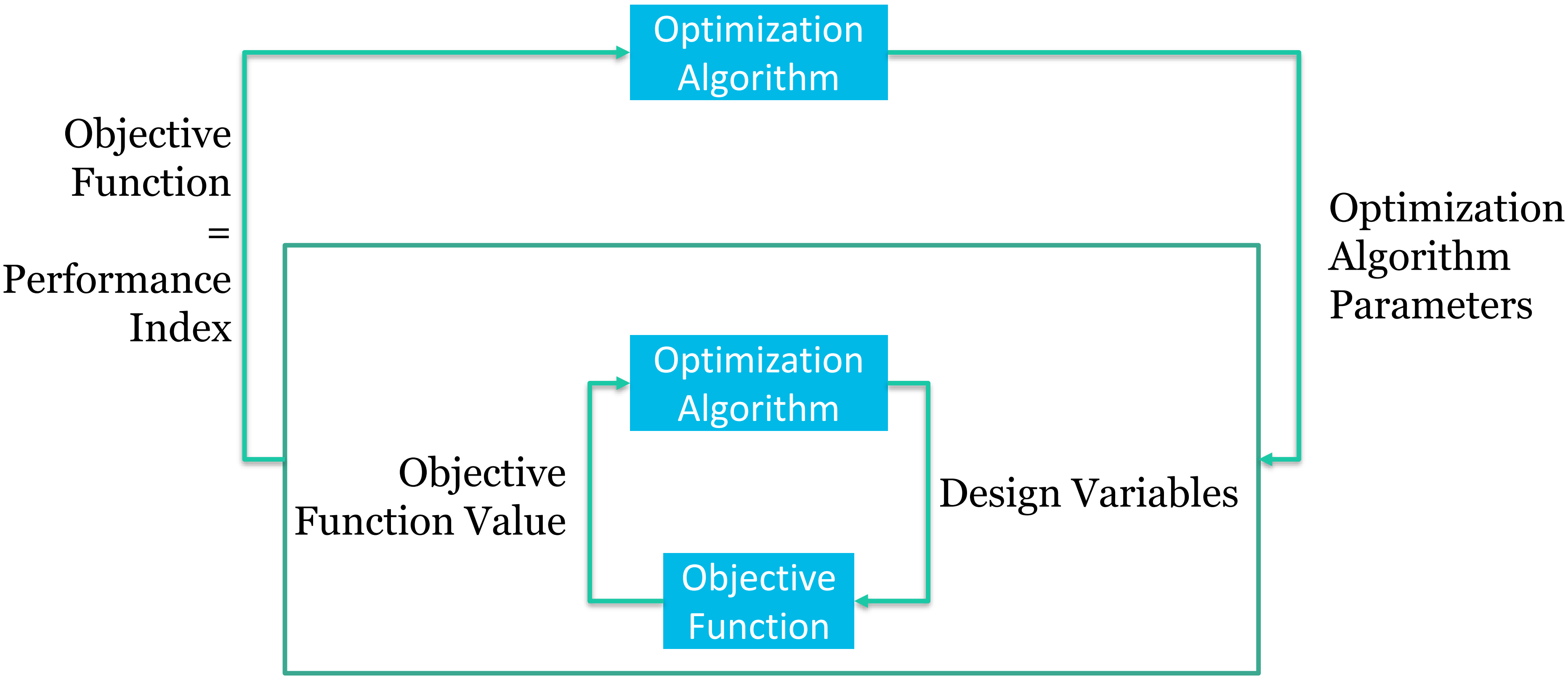
4. Choose the most suitable optimization algorithms

Meta-Optimization

LINKÖPING
UNIVERSITY

Meta-optimization

- Optimization of the parameters of the algorithm



Performance Index

A Performance Index

- We want a performance index that combines
  - Accuracy
  - Required number of function evaluations

- This will be a measure of the efficiency of the optimization algorithm

A Performance Index

- P_{opt} is the probability of finding the optimum
- N_{eval} is the required number of function evaluations

- The probability of not finding the optimum is

$$1 - P_{opt}$$

- The probability of NOT finding the optimum during m optimizations is

$$(1 - P_{opt})^m$$

A Performance Index

- P_{opt} is the probability of finding the optimum
- N_{eval} is the required number of function evaluations

- The probability of finding the optimum during m optimizations is

$$1 - (1 - P_{opt})^m$$

A Performance Index

- P_{opt} is the probability of finding the optimum
- N_{eval} is the required number of function evaluations

$$1 - (1 - P_{opt})^m$$

- The number of optimizations, m, that we can perform depends on the available number of function evaluations and N_{eval}

$$m = \frac{N_{available}}{N_{eval}}$$

- The value of $N_{available}$ does not depend on the optimization algorithm and can therefore be set to any number. It is here set to 100.

$$m = \frac{100}{N_{eval}}$$

A Performance Index

- P_{opt} is the probability of finding the optimum
- N_{eval} is the required number of function evaluations

- The performance index is therefore

$$1-(1-P_{opt})^{100/N_{eval}}$$

- It can be seen as the probability of finding the optimum if we can afford 100 function evaluations.

A Performance Index

- P_{opt} is the probability of finding the optimum
- N_{eval} is the required number of function evaluations
- The performance index is

$$1 - (1 - P_{opt})^{100/N_{eval}}$$

P_{opt}	N_{eval}	Performance Index
0.5	100	0.5
0.25	100	0.25
0.25	50	0.44

- High P_{opt} and low N_{eval} leads to high performance index

Question

- What happens if two optimization algorithms both have an accuracy of 100%

- Both get Performance Index = 1
- Choose the one that requires the least number of function evaluations

The Suggested Performance Index

- Advantages
  – Enables comparison of optimization algorithm performances

- Disadvantages
  – Will return 1 regardless of the required number of evaluations if 100% accuracy.
    - The one with the fewest required number of evaluations is the best
  – The criterion for a succesful optimization affects the result

LINKÖPING UNIVERSITY

Questions?