Simultaneous Tuning of Metaheuristic Parameters for Various Computing Budgets

Juergen Branke
Warwick Business School
University of Warwick,
Coventry, CV4 7AL, UK
+44 (0)24 7652 4306
Juergen.Branke@wbs.ac.uk

Jawad Elomari
Warwick Business School
University of Warwick,
Coventry, CV4 7AL, UK
+44 (0)24 7652 4306
J.Elomari@warwick.ac.uk

ABSTRACT
Many heuristics require a number of parameters to be tuned. One way to do this is meta-optimization: a higher level heuristic searches for the best parameter settings of a lower level heuristic which solves the optimization problem. However, the optimal parameter settings depend on the computational budget or running time available to the lower level heuristic. In this paper, we present a new meta-optimization approach to identify the best parameter settings simultaneously for various computational budgets.

Categories and Subject Descriptors
I.2.8 [Computing Methodologies]: ARTIFICIAL INTELLIGENCE – Problem Solving, Control Methods, and Search

General Terms
Algorithms, Experimentation.

Keywords
Offline parameter tuning, meta-optimization.

1. INTRODUCTION
In meta-optimization, a meta-level algorithm is used to tune the parameters of a lower-level algorithm which solves an optimization problem. For example, a meta-level Evolutionary Algorithm (meta-EA) can be used to tune the parameters of a Lower-Level Evolutionary Algorithm (LL-EA). Each individual in meta-EA is a particular parameter setting. An individual's fitness is the solution quality obtained by running it on the LL-EA, for a pre-specified number of generations, to solve the actual optimization problem. See [1] for an overview.

Optimal parameter settings usually depend on the computational budget; for a very short running time, exploitation is more important than exploration, while a longer running time allows for more exploration. The above described approach requires specifying a computational budget, or running time, available to the LL-EA. Changing the budget will require re-running of the meta-optimization algorithm. Here we present a new approach which, in a single run, can identify the best parameter settings for all computational budgets up to a specified maximum.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. METHODOLOGY
The key idea is to use a population to maintain a variety of parameter settings useful for different computational budgets. This is achieved by modifying the selection scheme of the meta-EA to prefer solutions that are best for any computational budget, rather than for a pre-defined computational budget. To do so, compare the whole convergence curves, rather than just the final result obtained for different parameter settings.

Specifically, the method works as follows:

- Start with a random population of parameter settings θ at the meta-level.
- Evaluate each of the parameter settings by running it for a certain number of generations (maximal computation budget of interest n_{max}) at the lower level. Replicate k times.
- Maintain for each parameter setting a record of the best objective found so far at each generation, averaged over the k runs (average convergence curve).
- For each generation at the lower level, identify the parameter setting that achieves the best performance. These parameter settings are classified as rank 1.
- Remove these parameter settings from the population and repeat as before, only classify these new parameter settings as rank 2. Continue in this fashion until all parameter settings have been ranked.

Example: Figure 1 shows three mutation rates used by an evolutionary algorithm to minimize Schwefel's F7 function. For generations 0 – 1, mutation rate 0.8360 (solid line) has the best objective value, so it's ranked 1. For generations 2 – 3, mutation rate 0.3895 (dashed line) has the best objective function value, it's also ranked 1. Finally mutation rate 0.8360 is again the best. Taking first-rank parameters out, only one line remains (the dotted line, 0.5755), it's ranked 2.

- Within each rank, sort individuals according to the number of lower level generations this parameter setting was in that rank.

Example: In Figure 1, mutation rate 0.8360 (solid line) is in the first rank for 14 generations, mutation rate 0.3895 (dashed line) is in the first rank for 2 generations, and mutation rate 0.5755 (dotted line) is in the second rank during all 16 generations.

3. EXPERIMENTATION AND RESULT
A preliminary evaluation of the Flexible-Budget Meta-Optimization method was conducted as follows:

- A simple evolutionary algorithm with mutation only is used for the meta-EA, to tune the parameters of the LL-EA. Here, the LL-EA only has one parameter: mutation rate.
- The LL-EA is used to find the minimum on Schwefel's F7 function, with 2 dimensions only.

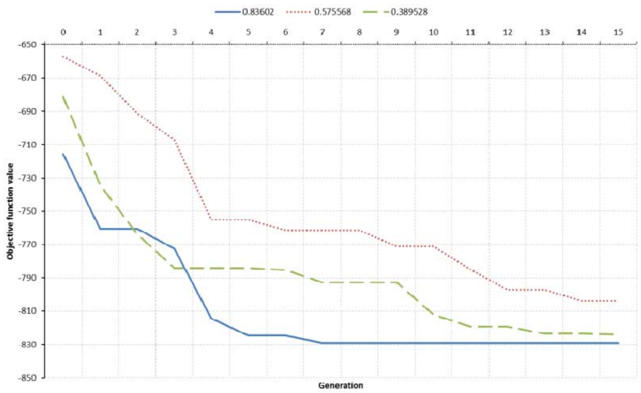

Figure 1: Ranking parameter settings based on convergence curves of the LL-EA.

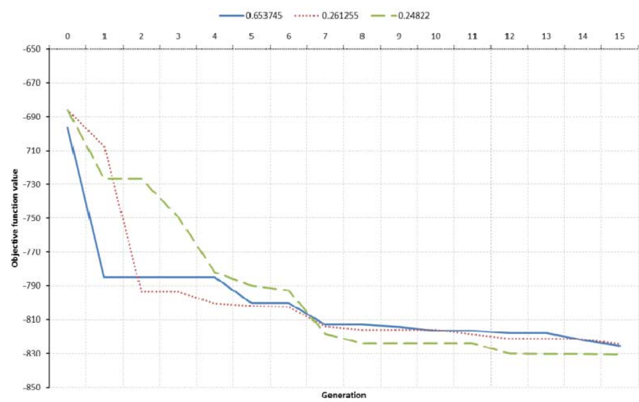

Figure 2: Convergence curves of Rank-1 parameters at the final generation of the meta-EA.

- Both algorithms are run for 40 generations (computational budget), and both use tournament selection (size = 2).
- The meta-EA bases its selection on the Flexible-Budget method, while the LL-EA selection is based on the objective function value.
- Each mutation rate generated by the meta-EA is replicated $k = 5$ times at the LL-EA and the average best-so-far objective value is kept. Replications are made with different seeds, but common random numbers are used for all individuals in the population.
- The meta-EA is replicated 20 times each with a different-random seed.
- Population size at the meta-EA was 20, and 30 at the LL-EA.
- Reproduction at the meta-EA is as follows: generate an offspring population with the same size as the parent population and pool them together, evaluate the entire pool according to the Flexible-Budget ranking method, and retain the best half for the next generation.

The Flexible-Budget method was compared to the standard Fixed-Budget method of selecting parents at the meta-level (i.e. fitness value).The number of generations allowed for the LL-EA were set to: 10, 15, 20, 25, 30, and 35. The expectation is that the Flexible-Budget's parameters will produce the same results as the Fixed-Budget's, for any computational budget smaller than the maximum tested.

A paired t-test on the mean difference in objective function values, obtained by each method, was conducted for each computational budget, see Table 1. For example, when the LL-EA is allowed to run for only 10 generations, the Flexible-Budget method was able to obtain a parameter setting that produced, on average, a slightly better solution (less by 0.6522), but the difference is statistically not significant.

Table 1: Summary statistics for the difference in objective function values between the two methods.

Gen.	Min.	Max.	Avg.	Std. Error	p-value
10	-3.518	1.613	-0.6522	0.4553	0.1682
15	-1.695	1.925	-0.2056	0.2072	0.3337
20	-0.27	1.505	0.197	0.0983	0.0595
25	-0.14	0.302	0.0178	0.0183	0.3416
30	-0.163	0.163	0.005	0.0123	0.6892
35	-0.163	0.05	-0.0684	0.0199	0.0027

Figure 3 shows the objective function values obtained by the Flexible-Budget method along with multiple runs of the Fixed-Budget method at the last generation of the meta-EA. The results are for one of the replications at the meta-level. The longer (solid) line is that of the Flexible-Budget method. It is composed of several line segments, each from a different parameter setting that is ranked 1 at that generation. The shorter (dashed) lines are those of the Fixed-Budget method, each representing a certain parameter setting identified as best after solving the problem for several (shorter) given computational budgets. Clearly, the Flexible-Budget method is able to identify best parameter settings for any computational budget in a single run.

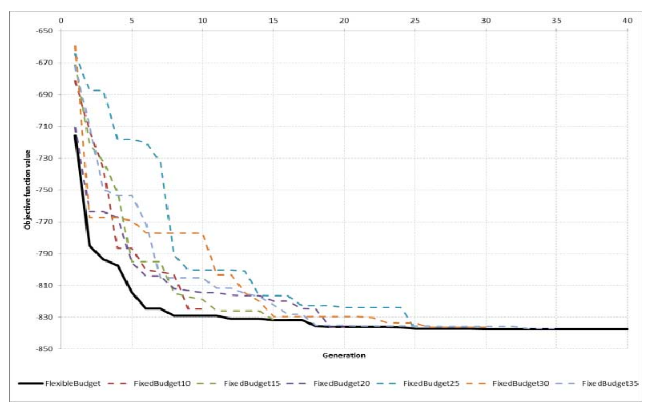

Figure 3: Comparison of the two methods at different computational budgets.

Acknowledgement: The authors would like to thank Thomas Stützle for his valuable input.

REFERENCES:

[1] A. Fialho, "Adaptive Operator Selection for Optimization," PhD, Ecole Doctorale d'Informatique, Universit´e Paris-Sud, Paris, 2010.