

# Multi-objective Optimisation with a Sequence-based Selection Hyper-heuristic

David J. Walker  
Computer Science  
University of Exeter, UK  
D.J.Walker@exeter.ac.uk

Ed Keedwell  
Computer Science  
University of Exeter, UK  
E.C.Keedwell@exeter.ac.uk

## ABSTRACT

Hyper-heuristics have been used widely to solve optimisation problems, often single-objective and discrete in nature. Herein, we extend a recently-proposed selection hyper-heuristic to the multi-objective domain and with it optimise continuous problems. The MOSSHH algorithm operates as a hidden Markov model, using transition probabilities to determine which low-level heuristic or sequence of heuristics should be applied next. By incorporating dominance into the transition probability update rule, and an elite archive of solutions, MOSSHH generates solutions to multi-objective problems that are competitive with bespoke multi-objective algorithms. When applied to test problems, it is able to find good approximations to the true Pareto front, and yields information about the type of low-level heuristics that it uses to solve the problem.

## CCS Concepts

•Mathematics of computing → Optimization with randomized search heuristics; •Applied computing → Multi-criterion optimization and decision-making;

## Keywords

Multi-objective optimisation, hyper-heuristic, heuristic sequences.

## 1. INTRODUCTION

Hyper-heuristics are well suited to solving optimisation problems and are used to solve (often combinatoric) problems by optimising the low level heuristics with which an optimisation problem will be solved. They are generally classified as either *generative* (creating novel low-level heuristics) or *selection-based* (identifying good low level heuristics from a pool of existing heuristics). This paper is concerned with the latter type.

A recent selection hyper-heuristic, SSHH, identifies sequences of low level heuristics using selection probabilities that are learned with a process based on a hidden Markov model [3]. That algorithm has been demonstrated on single-objective combinatoric problems, showing significant ability to both select suitable sequences of low-

level heuristics to solve the problem at hand, as well as producing solutions that are competitive with the current state of the art.

In this work we extend the original algorithm so that it is capable of optimising multi-objective problems. This is done by modifying the hidden Markov process so that the selection of low level heuristics that cause the generation of dominating solutions is rewarded. We evaluate the resulting algorithm on a range of multi-objective test problems drawn from the DTLZ problem suite [2]. An additional novelty of this work is that the single-objective version of the algorithm has so far not been applied to continuous problems, and we show that the algorithm is able to select low-level heuristics that are well suited to solving problems with such a representation.

## 2. ALGORITHM

SSHH generates sequences of low-level heuristics that generate good solutions to a given problem. It is a single-point algorithm, generating a single child solution from the current parent. Each low-level heuristic has an associated *acceptance strategy*. Each iteration of the algorithm begins with the application of a low-level heuristic (chosen at random based on the probability of selecting each heuristic). Depending on whether the acceptance strategy for the current heuristic is met the solution is either accepted, in which case its fitness is evaluated, or it is rejected, in which case another heuristic is applied and a sequence of heuristics is created. Transition probabilities between low-level heuristics are used to determine which heuristic is applied next. Once the acceptance strategy is met the solution is evaluated.

The basic operation of the multi-objective SSHH (MOSSHH) is unchanged; sequences of low-level heuristics are still generated, and their acceptance still depends on the acceptance strategy of each heuristic. MOSSHH uses an elite archive; as each acceptance strategy is met, and the objective values are evaluated, the archive is updated with the new solution. The use of an archive presents the possibility for incorporating genetic material from other solutions, in lieu of a crossover operator. We use two heuristics that draw genetic material from archived solutions. If the child solution was archived, transition probabilities are updated and the child replaces the parent in the next generation.

## 3. EXPERIMENTS

This paper is concerned with continuous problems, employing test problems from the DTLZ toolkit [2] and apply the MOSSHH algorithm to the solution of 2- and 3-objective instances of DTLZ1, DTLZ2 and DTLZ3, with recommended parametrisations [2]. Algorithms were run for 50,000 iterations in the case of DTLZ1 and DTLZ3, and 25,000 iterations for DTLZ2.

The following low-level heuristics are employed in this work:

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO'16 Companion July 20-24, 2016, Denver, CO, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: <http://dx.doi.org/10.1145/2908961.2909016>

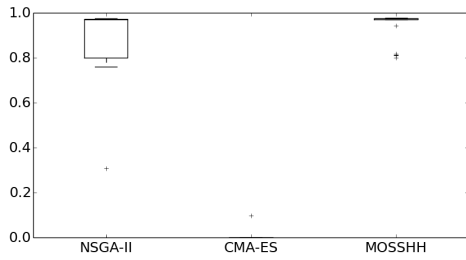


Figure 1: Comparative hypervolume results for a 3-objective DTLZ1. (CMA-ES results did not converge to within the reference point, bar one runx.)

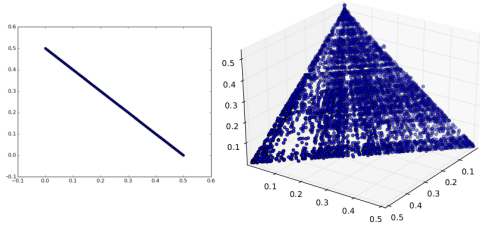


Figure 2: Sample Pareto front approximations generated by running MOSSHH on DTLZ1 in 2- and 3-objectives.

**Ruin and recreate** replaces the entire chromosome ( $h_1$ ) or a single parameter ( $h_2$ ) with an entirely new one.

**Additive mutation** from a uniform distribution between  $(-0.05, 0.05)$  ( $h_3$ ), a Gaussian distribution  $\mathcal{N}(0, 0.1)$  ( $h_4$ ), and a bimodal beta distribution between  $(-0.05, 0.05)$  ( $h_5$ ).

**Archive replacement** replaces the current solution with a complete solution ( $h_6$ ) or parameter ( $h_7$ ) from the archive.

We compare the MOSSHH results against those generated by optimising the problems with NSGA-II [1] and a MO-CMA-ES [4]; experiments were repeated 30 times. The hypervolume of the final estimated Pareto set was used to compare the results.

## 4. RESULTS

Figure 1 shows representative results for the 3-objective instance of DTLZ1. The results are based on the sampled hypervolume between the origin  $(0,0)$  and a reference point of  $(1,1)$ . In these results, and in the those conducted for DTLZ3, MOSSHH shows competitive or improved results when compared with the three benchmark algorithms. In some cases, MOSSHH has benefited from the ability to select different low-level heuristics to suit the specific characteristics of that problem, and has generated a close approximation of the true Pareto front. Results for DTLZ2 were less competitive; the online learning process is less effective for this problem, which, by design, is simpler for an EA to solve.

Example estimated Pareto fronts are shown for DTLZ1 in Figure 2. In both cases, we can see that the algorithm has both converged to and covered the full extent of the true Pareto front.

The operation of the hyper-heuristic is shown in the transition probabilities between and acceptance strategies of each low-level heuristic (again for 3-objective DTLZ1, Figure 3). The heuristics are grouped into three types. The ruin and recreate heuristics ( $h_1$  and  $h_6$ ) are rarely used; the parameter regeneration ( $h_2$ ) and mutation heuristics are used with relatively equal weight; and the archive parameter replacement heuristic ( $h_7$ ) is clearly used more than any

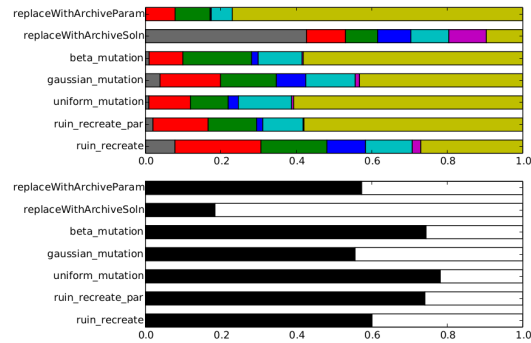


Figure 3: Top: transition probabilities for the low-level heuristics. The region occupied by each heuristic in the transition probability matrix shows the probability that a transition will be made between two heuristics, with a larger region indicating a greater probability. Heuristics are coloured as follows: grey –  $h_1$ ; red –  $h_2$ ; green –  $h_3$ ; blue –  $h_4$ ; cyan –  $h_5$ ; pink –  $h_6$ ; and yellow –  $h_7$ . Bottom: acceptance strategies; the larger the amount of black a heuristic is represented by, the greater the probability of a sequence being accepted following its use.

other. On inspection, this heuristic is used more frequently as the optimisation progresses and strong solutions enter the archive.

By inspecting the use of individuals over time, archive-based and ruin and recreate heuristics are used less frequently as optimisation proceeds, indicating that the algorithm has entered an exploitative mode and is mutating known good solutions.

## 5. CONCLUSION

We have presented the MOSSHH algorithm, demonstrating the use of a hyper-heuristic to optimise a multi-objective problem by selecting sequences of low-level heuristics. We have revealed useful information about the type of heuristics that can be applied to solve the multi-objective test problems used herein. Given that the problems are generally well solved with mutation heuristics, with some help from more disruptive heuristics on problems with challenging features (such as the deceptive fronts found in DTLZ1 and DTLZ3) expert time can be better spent developing heuristics that are known to be effective for those problems.

We intend to continue investigating the MOSSHH algorithm, and one aspect of this is basing transition probabilities on  $\epsilon$ -dominance. Rather than considering the dominance relationship between the current parent and child solutions, we intend to consider thresholding the probability updated based on the number of archived solutions dominated by the child, and the value of  $\epsilon$  needed to cause the child to dominate the parent. This will bring the multi-objective algorithm closer to the operation of the original SSHH algorithm.

## 6. REFERENCES

- [1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comp.*, 6(2):182–197, 2002.
- [2] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Proc. IEEE CEC 2010*, volume 1, pages 825–830, May 2002.
- [3] A. Kheiri and E. Keedwell. A sequence-based selection hyper-heuristic utilising a hidden markov model. In *Proc. GECCO 2015*, 2015.
- [4] T. Voss, N. Hansen, and C. Igel. Improved step size adaptation for the MO-CMA-ES. In *Proc. GECCO 2010*, pages 487–494, 2010.