Multi-objective Optimisation of a Water Distribution Network with a Sequence-based Selection Hyper-heuristic

David J. Walker[1], **Ed Keedwell**[1], **and Dragan Savić**[1]

[1]Centre for Water Systems, University of Exeter, UK

Correspondence to: David Walker (D.J.Walker@exeter.ac.uk)

Abstract. Multi-objective hyper-heuristics are fast becoming an efficient way of optimising complex problems. The water distribution network design problem is an example of such a problem, and this work employs a recent hyper-heuristic that generates sequences of low-level heuristics to solve the multi-objective water distribution design problem. The results presented are comparable to those generated by state-of-the-art metaheuristics, as well as a single-objective version of the algorithm from the literature. The information revealed from analysing the sequences generated to solve the problem reveal important information about the nature of the problem space that is not available from the metaheuristics, and the entire Pareto front can be explored in a single run as opposed to the multiple runs needed with the original single-objective algorithm.

1 Introduction

Hyper-heuristics are search and optimisation approaches for optimising complex problems. Whereas metaheuristics use a pre-defined strategy for applying low-level heuristics to generate new solutions to an optimisation, hyper-heuristics optimise the set of heuristics used to optimise a problem. They operate above the "domain barrier", requiring no detailed information about the optimisation problem they are employed to solve. They have been used extensively to optimise single-objective combinatoric problems, making them a natural candidate for optimising water distribution network (WDN) design problems, and they have been used successfully in this area (e.g. McClymont et al., 2013; Kheiri et al., 2015). Recent work has considered the solution of multi-objective problems with hyper-heuristics. Multi-objective optimisation hyper-heuristics are also emerging Guizzo et al. (2015); Maashi et al. (2015, 2014); Walker and Keedwell (2016a), and as the WDN problem can be cast in terms of multiple objectives it is sensible to consider the use of multi-objecitve hyper-heuristics to solve it.

The multi-objective algorithm employed herein generates sequences of heuristics that lead to good solutions. A single-objective version of the algorithm has been shown to produce good results (Kheiri et al., 2015) when optimising the same problem; that work cast the problem as a single-objective one by aggregating the objective functions with a weighted sum. This work instead considers the optimisation of the two objective functions independently, using Pareto dominance to compare the solutions. The *multi-objective sequence-based selection hyper-heuristic* (MOSSHH) has been demonstrated on continuous multi-objective test problems (Walker and Keedwell, 2016a), but this is the first time it has been used to optimise either a combinatoric or real world problem.

The remainder of this paper is organised as follows. Some relevant background material is presented in Section 2 before the algorithm is introduced in Section 3. Section 4 presents the experimental framework employed, before results are presented in Section 5. Concluding remarks are made in Section 6.

2 Background

2.1 Multi-objective Hyper-heuristics

A multi-objective problem is characterised by two or more objectives, which are usually in conflict. The quality $\mathbf{y}$ of a solution $\mathbf{x}$ to a multi-objective problem is defined by a set of M objective functions as follows:

$$\mathbf{y} = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})). \quad (1)$$

Metaheuristics, such as evolutionary algorithms (EAs), have long been used to optimise multi-objective problems, and operate by generating an approximation to the true *Pareto front*, the best possible trade-off surface between the problem objectives. It is not possible to generate feasible solutions that offer better quality beyond the Pareto front. Metaheuristics operate by using a predefined set of one or more low-level heuristics (such as a mutation or crossover operator) by which a randomly initialised solution, or population of solutions, is improved over time. Though these heuristics can be parametrised to allow them to adapt during the course of an optimisation, their operation is largely pre-determined. Significant time can be expended by an expert in designing a good low-level heuristic with which to optimise a specific problem. In contrast, *hyper-heuristics* optimise a problem by determining a good set of heuristics for that problem. Hyper-heuristics are either *generative*, in which a set of novel heuristics is generated, or *selection*-based, whereby a good set of existing heuristics is identified. This work considers selection hyper-heuristics. A hyper-heuristic operates above the domain barrier; no information about the specific problem is included in the algorithm, and the only information it receives about solutions is their objective values. In this sense, they operate as black-box optimisers.

2.2 Water Distribution Network Design

The water distribution network design problem is NP hard. Solutions to the problem are represented as a vector of pipe diameters d_k, and the problem is solved by identifying the best combination of pipe diameters. The operation of the network is simulated using EPANET (Rossman, 2000) so that the hydraulic performance of the network can be quantified. The first objective function represents the cost of the network:

$$f_1 = \sum_{k=1}^{K} \left(1.1 d_k^{1.24} \times l_k\right), \tag{2}$$

where the length of the k-th pipe is denoted by l_k. The second objective function quantifies the head deficit of the network:

$$f_2 = \sum_{n=1}^{N} \left(\left(\hat{h}_n - h_n\right) > 0\right), \tag{3}$$

where h_n represents the hydraulic head of the n-th node and $\hat{h}_n$ is the target head at that node. Though the target head is often the same throughout the network, some of the nodes in the problem used in this work have different targets. Only head deficit is considered.

The approach taken by (Kheiri et al., 2015) was to aggregate these two objective functions in order to optimise them with the single-objective version of the algorithm used herein. Since this work considers a multi-objective algorithm, the two problems are considered independently and an approximation to the true Pareto front between the two is sought.

Algorithm 1 Multi-objective Sequence-based Selection Hyper-heuristic (MOSSHH)

```
x, h_c, A, B = initialise()
repeat
    h_p = h_c
    h_c = select(A, h_p)
    AS = select(B, h_c)
    record(h_p, h_c, AS)
    x' = apply(x, h_c)
    if AS == 1 then
        archived = update_archive(x)
        if f(x) ≰ f(x') then
            x = x'
            if archived == true then
                x* = x'
                update()
            end if
        end if
        clear_records()
    end if
until termination criterion met
```

3 Multi-objective Sequence-based Hyper-heuristic

The sequence-based hyper-heuristic (SSHH) algorithm is a hyper-heuristic first proposed by Kheiri and Keedwell (2015) and more recently extended to the multi-objective domain by (Walker and Keedwell, 2016a) with the definition of the multi-objective SSHH (MOSSHH). Whereas selection hyper-heuristics generally operate by applying low-level heuristics to a solution in order to generate a hopefully improved solution, MOSSHH uses sequences of low-level heuristics that can be applied. The algorithm is based on a hidden Markov model; it employs a transition matrix to determine the next low-level heuristic to be added to a sequence, and a hidden state that determines whether or not the current sequence is complete, known as the acceptance strategy. The transition probabilities are updated if the acceptance strategy is met and the child solution resulting from the application of the sequence of low-level heuristics is not inferior to the original parent from which it was generated. In the single-objective algorithm probabilities are updated when the fitness of the child is superior to that of the parent. In order to optimise multi-objective problems MOSSHH uses the dominance relation instead. MOSSHH employs an elite archive of solutions to store the current approximation to the Pareto front, and the transition probabilities are updated if the child solution is added to the archive.

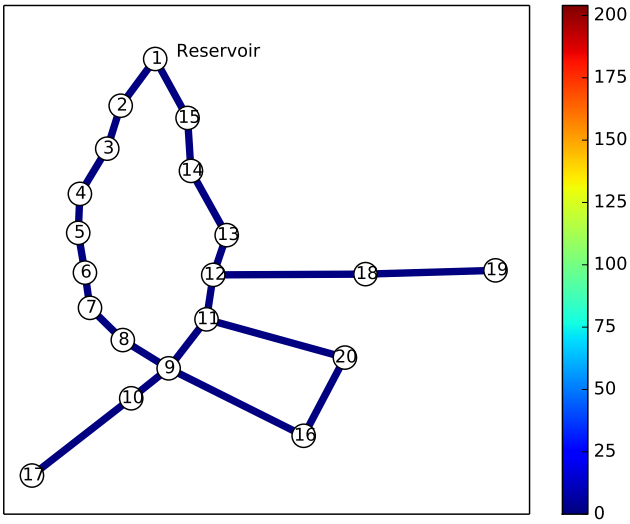

Figure 1. Schematic showing the New York Tunnels network.

The MOSSHH algorithm is described in Algorithm 1. The first step (Line 1) is to initialise a random solution, as well as the transition probability matrix (A) and the acceptance strategy probabilities (B). All transition probabilities are initialised to $1/H$ (for H low-level heuristics) and the acceptance strategy for each heuristic is initialised to $1/2$. A current heuristic h_c is chosen as a starting position. The iterative phase of the algorithm then commences, and runs until a stopping condition is met (we use a fixed budget of function evaluations). It begins by selecting the next heuristic, based on the current heuristic using B (Line 3-4). A record is made of the heuristic (Line 6) and it is applied to the current solution $\mathbf{x}$ to generate a child $\mathbf{x}'$. If the acceptance strategy is met (Line 8) then the solution is evaluated under the objective functions and the archive is updated (Line 9). If the child is not dominated by the parent (Line 10) then it replaces the parent as the current solution in the next generation. If the solution was added to the archive then the sequence is completed and a new one begins in the next generation.

4 Experiments

The MOSSHH algorithm is demonstrated on the well-known New York Tunnels benchmark problem, illustrated in Fig. 1 (Schaake and Lai, 1969). The problem requires the optimisation of 21 pipe diameters linking 20 nodes. There are 16 potential pipe diameters, and as such the problem is combinatoric. The low-level heuristics employed are a combination of those used in Kheiri et al. (2015) to optimise a single-objective variant of the same problem, and those introduced to optimise multi-objective problems in Walker and Keedwell (2016a). The heuristics are as follows:

- h_1**: change pipe** – replace a single pipe with one of the 16 possible diameters at random (similar to the parameter-wise ruin and recreate parameter used in Walker and Keedwell (2016a)).
- h_2**: swap:** – select parameters at random and swap their pipe diameters.
- h_3**: change by one size** – increase the pipe diameter to the next biggest diameter, or decrease it to the next smallest.
- h_4**: shuffle** – select a group of between 1 and 5 parameters and shuffle their pipe diameters.
- h_5**: ruin and recreate** – replace the entire chromosome with one generated at random.
- h_6**: archive parameter replacement** – replace a randomly chosen parameter and replace it with the corresponding diameter of an archived solution.

The algorithm was run for 40,000 function evaluations, the experiment was repeated 30 times. Results are compared using the hypervolume indicator (Fleischer, 2003), which evaluates the results of an optimiser both in terms of its convergence to the true Pareto front, and its ability to cover the Pareto front. In order to demonstrate the efficacy of MOSSHH's online learning component, comparative runs are also shown for the individual low-level heuristics. Each comparative run is executed for 40,000 function evaluations, and the optimiser is run 30 times for each heuristic. These experiments employ a simple (1+1)–evolution strategy that at each generation creates a child from the current parent solution by mutating it (with whichever of the low-level heuristics is being tested); the parent solution in the next generation is the child, unless it is dominated by the parent in which case the parent is retained.

5 Results and Discussion

5.1 Optimisation Results

Summary attainment surfaces are used to show the extent to which an optimiser has converged over repeated executions. Figure 2 illustrates such an attainment surface for the MOSSHH executions. The distribution of colour indicates the frequency with which a region is dominated by a member of one of the estimated Pareto fronts; most of the range is close to the median Pareto set (shown in black) indicating that the optimiser frequently converges close to this point in the thirty runs of MOSSHH. There is more spread in the second objective, as seen by the height of the medium front in the second objective.

Comparative hypervolume (Fleischer, 2003) results for MOSSHH and the individual low-level heuristics are shown in Figure 3. As can be seen, MOSSHH has overtaken all of

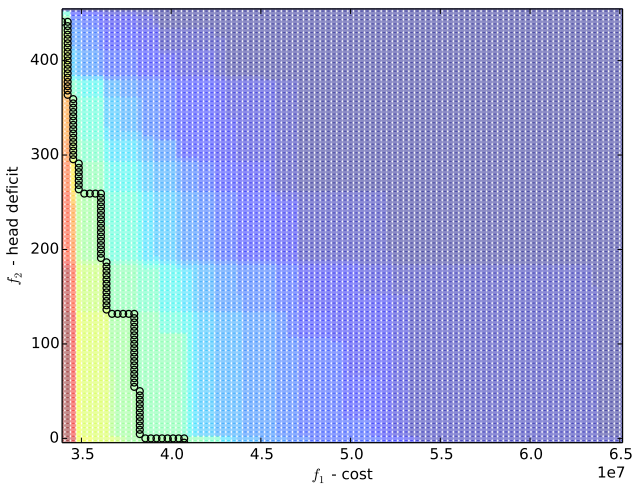

Figure 2. Summary attainment surfaces for MOSSHH. The colour indicates how frequently a region of the objective space was dominated by a member of one of the approximated Pareto fronts generated by MOSSHH (blue indicates frequently dominated, red indicates infrequently dominated). The black line shows the median attainment surface for the 30 runs.

the low-level heuristics 10,000 function evaluations in. In fact, it is superior to the swap, shuffle and pipe-wise ruin and recreate operators from a very early stage in the optimisation. The remaining operators (change pipe and change pipe by one size) are known to be effective heuristics for optimising problems of this type. The results show that the algorithm has converged by 20,000 function evaluations, as have the algorithms using the individual heuristics. These algorithms converge much faster – albeit to a worse set of solutions – due to the online learning mechanism used within MOSSHH, which allows for the generation of superior solutions but takes longer to accomplish it.

Fig. 4 illustrates three of the networks optimised by the algorithm, taken from the Pareto front approximation formed by aggregating the estimated Pareto fronts generated by MOSSHH in all 30 runs. The leftmost network represents the solution that best optimises the cost objective. The rightmost network is the corresponding solution for the head deficit objective. The centre network is the median solution. Pipes are coloured according to their diameter. As the results show, low-cost solutions are characterised by smaller pipes while networks minimising head deficit tend to comprise pipes with larger diameters.

5.2 Sequence Analysis

Fig. 5 shows the transition probabilities at the end of the algorithm's execution for one of the 30 runs. The row-wise and column-wise ordering of the heuristics is the same, and a region indicates the probability of transitioning from the heuristic represented by a given row to that region's heuristic. The size of the region conveys the probability, with a large region indicating a high probability of transition. The lower plot shows acceptance strategy probabilities. The ordering of heuristics is as in the top plot, but this time the regions indicate the likelihood that a heuristic will be accepted. As the heuristic can either be accepted or not, there are only two regions; black indicates that a heuristic will be accepted, white that it will not. As can be seen, the probability of transitioning to the pipe-wise ruin and recreate operator is considerably lower than for any of the other heuristics. The largest transition probabilities are to the change by one size, and archive parameter heuristics, which is supported by acceptance strategies – these heuristics have the largest acceptance strategies.

This conclusion is supported by Figure 6, which shows cumulative averages for the low-level heuristics, presenting the frequency with which a heuristic appears in a sequence as the optimisation process executes. This visualisation shows the use of the change by one size heuristic increasing over time, while the pipe-wise ruin and recreate heuristic is used significantly less over time. The remaining four heuristics do not change considerably during the course of the optimisation process, though we note that the slight upward trend visible in the archive parameter replacement heuristic indicates the algorithm entering an exploitative phase, instead of exploring. This is because the archive is becoming populated with strong solutions, and including their parameters in new child solutions in turn creates strong solutions. Examining the results in this way is a useful way of understanding which heuristics are of use when optimising a problem such as this. This understanding is an important emergent property of the online learning mechanism that is not provided by standard metaheuristics.

6 Conclusions

This paper has demonstrated the use of a multi-objective sequence-based hyper-heuristic for optimising the water distribution network problem. The problem has been optimised in the past using a single-objective sequence-based hyper-heuristic (SSHH), which aggregated the two objective functions to compute a single fitness function. This work is the first to use a sequence-based hyper-heuristic that works with Pareto dominance in order to consider the two objectives independently. The MOSSHH algorithm has previously been demonstrated on continuous multi-objective test problems, and this work represents its first application to real-world discrete problems.

Having used MOSSHH to optimise the New York Tunnels problem, the results presented herein show that it is capable of generating the best approximation to the problem's true Pareto front compared to using any one of the low-level heuristics it relies on. In addition, the algorithm's use of online learning enables it to provide additional informa-

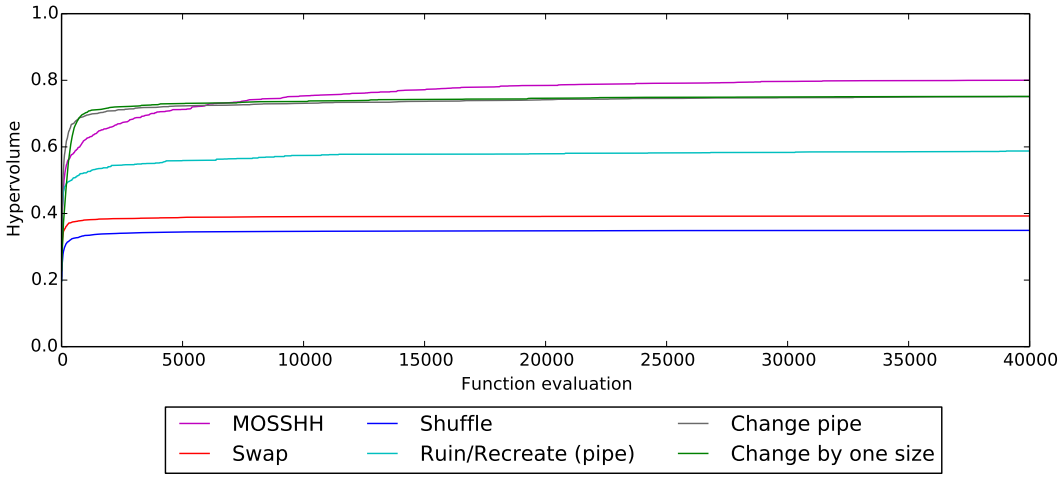

Figure 3. Normalised hypervolume for MOSSHH and the five of the low-level heuristics used individually showing their convergence over time as the optimisation process executes. The results are averaged over the thirty runs for each optimiser.

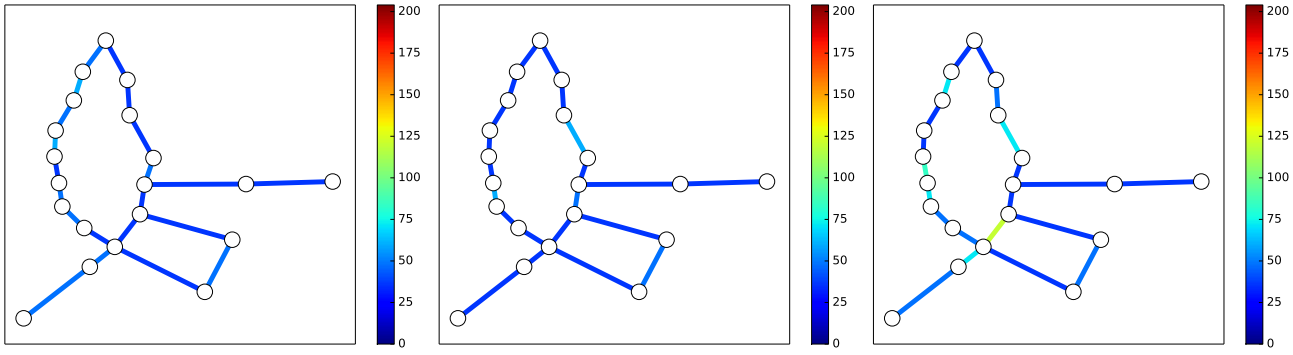

Figure 4. Three networks from the estimated Pareto front generated by the algorithm. The left minimises cost, the right minimises head deficit, and the middle network is the median solution. Colour indicates pipe diameter.

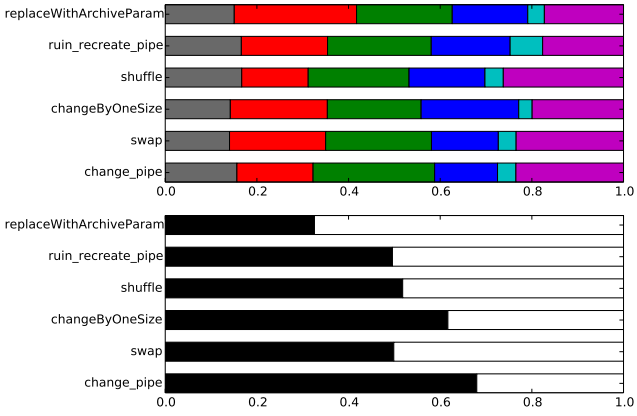

Figure 5. Top: An exemplar transition probability matrix at the end of the optimisation procedure. Bottom: The corresponding acceptance strategies. Black indicates that a sequence will be accepted.

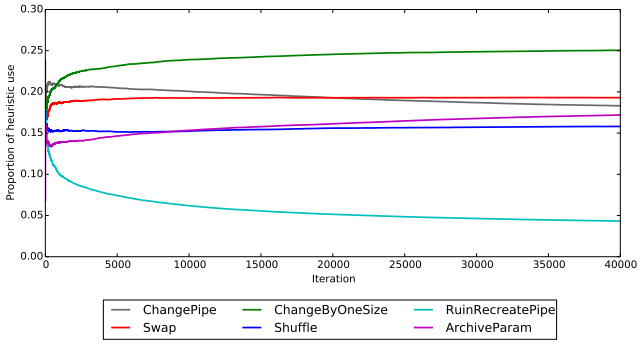

Figure 6. Cumulative averages of the frequency with which each heuristic appears in a sequence over the time of the optimisation. The upward trend of the change by one size heuristic indicates that heuristic is being used increasingly as the optimisation proceeds, while the downward trend of the pipe-wise ruin and recreate operator indicates that heuristic has been tuned out by the algorithm.

tion about how these heuristics are used. This is a valuable source of information for heuristic designers, who can use it to understand exactly which heuristic features are of use for specific types of problems. In this work, the change by one size heuristic is a strong candidate heuristic, while replacing a pipe with another chosen at random is shown to provide little useful contribution to the optimisation process.

This work has demonstrated the efficacy of the MOSSHH algorithm on a simple benchmark problem with a small set of low-level heuristics. While the results shown herein are extremely promising, there exist multiple areas of future work. Two avenues of ongoing investigation involve testing the process on a range of more complex test problems, to better characterise instances that MOSSHH struggles with, and investigating and analysing its use with a larger pool of heuristics. In addition to providing a means for generating superior solutions to the problems themselves, we expect to be able to use the emergent information from the online learning aspect of the algorithm to better characterise the problems themselves. A recent paper proposed the use of a modified MOSSHH for optimising *many*-objective problems (comprising four or more objectives) Walker and Keedwell (2016b). Such problems are difficult to optimise with dominance-based optimisers, and that work used alternative solution comparison methods to generate approximations to a many-objective Pareto front. A further aspect of ongoing work is to consider the many-objective water distribution network design problem.

References

Fleischer, M.: The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics, in: Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, pp. 519–533, Springer, 2003.

Guizzo, G., Vergilio, S. R., and Pozo, A. T. R.: Evaluating a Multi-objective Hyper-Heuristic for the Integration and Test Order Problem, in: 2015 Brazilian Conference on Intelligent Systems (BRACIS), pp. 1–6, 2015.

Kheiri, A. and Keedwell, E.: A Sequence-based Selection Hyper-heuristic Utilising a Hidden Markov Model, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015), 2015.

Kheiri, A., Keedwell, E., Gibson, M., and Savić, D.: Sequence analysis-based hyper-heuristics for water distribution network optimisation, in: Proceedings of Computing and Control for the Water Industry (CCWI 2015), 2015.

Maashi, M., Ozcan, E., and Kendall, G.: A multi-objective hyper-heuristic based on choice function, Expert Systems with Applications, 41, 4475 – 4493, 2014.

Maashi, M., Kendall, G., and Ozcan, E.: Choice function based hyper-heuristics for multi-objective optimization, Applied Soft Computing, 28, 312 – 326, 2015.

McClymont, K., Keedwell, E., Savić, D., and Randall-Smith, M.: A general multi-objective hyper-heuristic for water distribution network design with discolouration risk, Journal of Hydroinformatics, 15, 700–716, 2013.

Rossman, L. A.: EPANET 2 Users Manual, 2000.

Schaake, J. and Lai, D.: Linear Programming and Dynamic Programming Application to Water Distribution Network Design, Tech. rep., MIT, 1969.

Walker, D. J. and Keedwell, E.: Multi-objective Optimisation with a Sequence-based Selection Hyper-heuristic, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016), 2016a.

Walker, D. J. and Keedwell, E.: Towards Many-objective Optimisation with Hyper-heuristics: Identifying Good Heuristics with Indicators, in: Proceedings of the Parallel Problem Solving from Nature (PPSN 2016), 2016b.