Analysing Heuristic Subsequences for Offline Hyper-heuristic Learning

WILLIAM B. YATES, University of Exeter
EDWARD C. KEEDWELL, University of Exeter

Abstract

A selection hyper-heuristic is used to optimise a number of well-known benchmark problems. The resulting sequences of heuristics and objective function values are used to create a database. The sequences in the database are broken down into subsequences and the concept of a *logarithmic return* is used to discriminate between "effective" subsequences, which tend to decrease the objective value, and "disruptive" subsequences, which tend to increase the objective value. These subsequences are then employed in a sequenced based hyper-heuristic and evaluated on unseen benchmark problems. The results demonstrate that the "effective" subsequences perform better than the "disruptive" subsequences across a number of problem domains with 99% confidence. The identification of subsequences of heuristic that can be shown to be effective across a number of problems or problem domains could have important implications for the design of hyper-heuristics.

CCS Concepts: • **Mathematics of computing → Combinatorial optimization**; *Optimization with randomized search heuristics*;

Additional Key Words and Phrases: Hyper-heuristics, Offline learning

ACM Reference Format:
William B. Yates and Edward C. Keedwell. 2019. Analysing Heuristic Subsequences for Offline Hyper-heuristic Learning. 1, 1 (June 2019), 2 pages. https://doi.org/10.1145/3319619.3326760

1 INTRODUCTION

The paper [5] explores the impact of sequences of search operations on the performance of an optimiser through the use of log returns and a database of sequences. The study demonstrates that although the performance of individual perturbation operators is important, understanding their performance in sequence provides greater opportunity for performance improvements within and across operations research domains.

2 HYPER-HEURISTICS

Hyper-heuristics are general purpose heuristic methods that are employed to solve computationally hard problems. Typically such problems are presented as optimisation problems where the goal is to minimise an *objective function* defined on a space of solutions.

Hyper-heuristics either *generate* or *select* low level heuristics. A generation hyper-heuristic generates new heuristics by discovery, or by modifying or combining existing low level heuristics. Selection hyper-heuristics, such as those presented in this study, must select and apply a heuristic chosen from a set of low level heuristics.

Hyper-heuristics are intended to be "re-useable" and applicable to many different problem domains with minimal changes, rather than being specialised to a particular problem domain or problem instance. Such methods have proved effective when applied to a number of real world problems [1].

3 OFFLINE LEARNING

Many selection hyper-heuristics employ learning algorithms to improve optimisation performance. Such learning can be categorised as either *online* or *offline*. Online learning is based on the low level heuristic selections and resulting objective function values computed during the execution of a hyper-heuristic. The objective is to improve optimisation performance on the problem at hand. In contrast, offline learning is performed on a database of low level heuristic selections and objective function values computed by a hyper-heuristic on a fixed number of benchmark problems. The objective is to *generalise* across the benchmark training problems leading to improvements in optimisation performance on unseen test problems. This study is concerned with offline learning.

4 AN OFFLINE LEARNING DATABASE

A single selection hyper-heuristic is run on a number of benchmark problems in order to generate a database of low level heuristic selections and objective function values. The benchmark problems are drawn from four problem domains from the Hyper-heuristics Flexible framework (or HyFlex, [4]). Specifically, 1D bin packing (BP), permutation flowshop (PFS), boolean satisfiability (SAT), and personnel scheduling (PS).

The domains contains a number of distinct problem instances of varying complexity. This study uses the first 10 problem instances in each domain.

Each domain has four general *classes of heuristic*: parameterised mutation (M) which randomly perturbs a solution, crossover (C) which constructs a new solution from two or more existing solutions, parameterised ruin and recreate (R) which destroys a given solution partially and then rebuilds the deleted parts, and parameterised hill climbing or local search (L) that incorporates an iterative improvement process and returns a non-worsening solution.

The number and implementation of the *low level heuristics* in each class differs between problem domains (see table 1).

Table 1. The low level heuristics for each domain.

Dom.	Heuristics
BP	$M_0, R_1, R_2, M_3, L_4, M_5, L_6, C_7$
PFS	$M_0, M_1, M_2, M_3, M_4, R_5, R_6, L_7, L_8, L_9, L_{10}, C_{11}, C_{12}, C_{13}, C_{14}$
SAT	$M_0, M_1, M_2, M_3, M_4, M_5, R_6, L_7, L_8, C_9, C_{10}$
PS	$L_0, L_1, L_2, L_3, L_4, R_5, R_6, R_7, C_8, C_9, C_{10}, M_{11}$

Authors' addresses: William B. Yates, University of Exeter, Computer Science, College of Engineering, Mathematics and Physical Sciences, Exeter, UK, EX4 4QF, wy254@exeter.ac.uk; Edward C. Keedwell, University of Exeter, Computer Science, College of Engineering, Mathematics and Physical Sciences, Exeter, UK, EX4 4QF, E.C.Keedwell@exeter.ac.uk.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326760

5 HEURISTIC SUBSEQUENCES

Most offline (and online) learning research aims to improve the selection of *single* heuristics (or heuristic pairs). However, recent research argues that heuristic selections should be understood as part of a *sequence* (or subsequence) of selections [5]. This study presents a novel statistical framework for the analysis of subsequences of heuristics based on the concept of *logarithmic returns*. The sequences of heuristic selections and objective function values in the database are broken down into subsequences, and logarithmic returns are used to categorise and select subsequences of heuristics based on their associated objective function values. Log returns are used to discriminate between "effective" subsequences, which tend to decrease the objective value, and "disruptive" subsequences, which tend to increase the objective value. Log returns are also be applied to the measurement and analysis of hyper-heuristic performance. Specifically, the unit log return function β is used to compare subsequences that have different lengths or have objective values that have different ranges. The mean final log return $\overline{\alpha}_f$ is used to compare hyper-heuristic performance over a number of problems and problem domains, and provides a better measure of performance than the arithmetic mean. Lastly, the γ-ratio is used to select sets of subsequences of heuristic selections from the offline database. These functions are used to demonstrate that

(1) the expected exploration-exploitation behaviour in sequences is seen in some, but not all domains,
(2) offline learning can outperform online learning of heuristic subsequences,
(3) the combination of heuristics into subsequences outperforms individual heuristic selections, and
(4) generalisation across domains is possible for 3 out of the 4 domains tested.

6 EMPIRICAL RESULTS

Subsequences chosen using the γ-ratio are employed in a simple, sequence based hyper-heuristic, denoted EvalHH and evaluated on unseen examples of the benchmark problems. Results are presented for subsequences of low level heuristics, chosen for each domain, and subsequences of heuristic classes, which are used to investigate cross-domain offline learning. The results are compared with those produced by the SSHH hyper-heuristic (see [3]) which is known to perform well on the HyFlex problems. The SSHH hyper-heuristic is a sequence based selection hyper-heuristic which employs online learning. The results for SSHH are included to provide a comparison between an online learning hyper-heuristic and the offline methodology presented here. The empirical results are analysed using a one tailed Wilcoxon signed-rank test [2], and demonstrate that the effective subsequences perform significantly better than the disruptive subsequences. For example, for each problem instance in a domain, the top 10 subsequences with the largest γ-ratio are selected. Table (2) shows the results of the EvalHH hyper-heuristic using this subsequence set (with a leave-one-out methodology) and the SSHH hyper-heuristic, averaged over 40 runs of each HyFlex problem; a total of 1600 runs. In this case, the EvalHH hyper-heuristic outperforms the SSHH hyper-heuristic overall, and on 3 domains out of 4, with 99% confidence.

Table 2. The mean final log return $\overline{\alpha}_f$ for EvalHH and SSHH. The domain statistics are calculated over 400 runs. Winning scores are shown in bold.

Dom.	EvalHH	SSHH
BP	**-0.3565**	-0.3009
PFS	**-0.0074**	-0.0049
SAT	**-0.9509**	-0.6908
PS	-1.7708	**-1.7770**
All	**-0.7714**	-0.6934

The result that a simple hyper-heuristic such as EvalHH using a fixed set of subsequences is able to outperform SSHH, a published hyper-heuristic which employs online learning, demonstrates the utility of the subsequence based approach and the statistical framework proposed in this study. The result also shows that there is scope to improve the performance of the SSHH hyper-heuristic using offline learning techniques.

7 CONCLUSIONS

The identification of subsequences of heuristic selections that can be shown to be effective across a number of problems has important implications for the design of sequence based hyper-heuristics. Effective subsequences of low level heuristics can be used to improve hyper-heuristic performance either directly, by embedding them in a suitable hyper-heuristic, or indirectly as the inputs to an appropriate offline learning algorithm. Furthermore, by comparing effective subsequences across different problem domains it is possible to investigate the potential for cross-domain learning by, for example, attempting to identify a set of effective cross-domain subsequences. Effective subsequences of heuristic classes could be useful in constructing optimisers for problems from novel domains.

This work has demonstrated that subsequences can be reliably extracted in an offline manner from a database of heuristic operations that have both effective and disruptive characteristics. It has also been shown that when those subsequences are well-chosen, they can provide significant improvements in performance. The approach has also shown that differences between problem domains and the interface between domain and heuristic class can be quantified by using statistics which is an important consideration for generalist algorithms such as these. For example, as 3 of the 4 domains behaved similarly, an effective subsequence of heuristic classes on one of these would transfer to the other domains. By quantifying the differences between domains, outlier domains (such as BP) can be identified and bespoke optimisers can be created for them.

REFERENCES

[1] Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Ŏzcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society **64**(12), 1695–1724 (2013)
[2] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006)
[3] Kheiri, A., Keedwell, E.C.: A sequence-based selection hyper-heuristic utilising a hidden Markov model. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 417–424. ACM (2015)
[4] Ochoa, G., Hyde, M., Curtois, T., Vazquez-Rodriguez, J.A., Walker, J., Gendreau, M., Kendall, G., McCollum, B., Parkes, A.J., Petrovic, S., Burke, E.K.: HyFlex: A benchmark framework for cross-domain heuristic search. In: J.K. Hao, M. Middendorf (eds.) Evolutionary Computation in Combinatorial Optimization, pp. 136–147. Springer Berlin Heidelberg (2012)
[5] Yates, W.B., Keedwell, E.C.: An analysis of heuristic subsequences for offline hyper-heuristic learning. Journal of Heuristics **25**(3), 399–430 (2019)