Clustering of Hyper-heuristic Selections using the Smith-Waterman Algorithm for Offline Learning

W. B. Yates
University of Exeter
Computer Science, College of Engineering, Mathematics and Physical Sciences
Exeter EX4 4QF, UK

E. C. Keedwell
University of Exeter
Computer Science, College of Engineering, Mathematics and Physical Sciences
Exeter EX4 4QF, UK

KEYWORDS

Hyper-heuristic, Offline Learning, Smith-Waterman

ACM Reference format:
W. B. Yates and E. C. Keedwell. 2017. Clustering of Hyper-heuristic Selections using the Smith-Waterman Algorithm for Offline Learning. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076025

1 INTRODUCTION

Selection hyper-heuristics are methods that are typically used to solve computationally hard optimisation problems (see [1]). A selection hyper-heuristic selects heuristics from a given set of low level heuristics, deciding which heuristic to apply at a given point during the optimisation process. The sequences of low level heuristic selections and objective function values that result from the application of a simple selection hyper-heuristic to the HyFlex problem set (see [3]) are used to construct an offline learning database. The intention is to select effective subsequences of heuristics from this database and use them as inputs to machine learning algorithms in order to improve optimisation.

The purpose of this study is to algorithmically identify and analyse the similarities and dissimilarities that occur between the sequences of the database. By employing a suitable measure of similarity, the sequences of the offline database can be grouped or clustered according to the view of the similarity algorithm. It can be shown that by using a well-known algorithm from bioinformatics more commonly used to explore the conserved regions of DNA sequences, the Smith-Waterman algorithm (see [4]), it is possible to characterise problems using only the sequence of heuristic choices made by the hyper-heuristic. The Smith-Waterman algorithm is able to provide a measure of the level of similarity between two strings operating over any alphabet, and is used here to define a distance function between sequences of heuristics which is then used to perform a cluster analysis. The results presented here show that the Smith-Waterman algorithm is able separate the offline database into distinct problem domains.

The automatic separation and identification of problem domains from sequences of heuristics is important because it demonstrates that there are subsequences of heuristics that are common to each problem domain, and that these subsequences vary between domains. This strengthens the thesis that subsequences of heuristics play an important role in the optimisation process. In addition, the identification of a (similar) problem domain can improve the choice of learning algorithm, learning algorithm parameterisation, and training data. For example, in [5] a k-nearest neighbour classifier is used to identify problems in an offline database that are similar to a target problem based on a set of measurable problem characteristics. This *metaknowledge* is then used to retrieve further problem specific information which is used to optimise the performance of a planning algorithm. The method described here differs from conventional metalearning approaches to algorithm selection in that, as only sequences of low level heuristic classes are employed, no problem specific information is required, preserving the domain barrier.

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3076025

2 HYFLEX AND THE OFFLINE LEARNING DATABASE

The Hyper-heuristics Flexible framework (or HyFlex, see [3]) is an implementation of 4 computationally hard benchmark problem domains:

(1) 1D bin packing (BP),
(2) permutation flow shop (PFS),
(3) boolean satisfiability (SAT), and
(4) personnel scheduling (PS).

Each problem domain contains 10 distinct problems of varying complexity. HyFlex hides all problem specific information such as the solution representations, the solution constructions, and the low level heuristic implementations. Each HyFlex domain has four general classes of low level heuristic:

(1) mutation (`M`) which perturbs a solution randomly,
(2) crossover (`C`) which constructs a new solution from two or more existing solutions,
(3) ruin and recreate (`R`) which destroys a given solution partially and then rebuilds the deleted parts, and
(4) local search (`L`) that incorporates an iterative improvement process and returns a non-worsening solution.

The actual number and implementation of the low level heuristics differs between problem domains.

A simple hyper-heuristic is executed for 150 selections, 40 times on each of the 10 HyFlex problems in each domain. The resulting 1600 sequences of heuristic selections and objective function values are used to construct the offline database. The number of 40 trials was chosen because for a sufficiently large number (say

$n > 30$) the central limit theorem ensures that the arithmetic mean of any results will be approximately normally distributed, regardless of the underlying distribution. This allows robust statistics to be calculated for each problem. The number of 150 selections was chosen after experimental observations indicated that no major improvements in objective function occurred beyond this point.

3 THE SMITH-WATERMAN ALGORITHM

The goal is to compare sequences of heuristic classes to obtain an understanding of the problem space from an algorithmic perspective. However, the comparison of sequences is not straightforward. For example, using the Hamming distance, two otherwise identical binary strings will appear dissimilar, that is score a high Hamming distance, if one string is shifted by one character in either direction. The Smith-Waterman algorithm (see [4]) is intended to overcome this because it attempts to identify similar regions of any given pair of strings. In bioinformatics, the Smith-Waterman algorithm is used to analyse the arrangement of DNA/RNA or protein sequences. The algorithm performs a *local sequence alignment* by use of dynamic programming; instead of looking at the whole sequence, the Smith-Waterman algorithm compares subsequences of all possible lengths and optimises a similarity measure. A large similarity score produced by the algorithm implies that the strings are very similar. A similarity score of 0 implies that the two strings have no symbols in common. The similarity measure is defined by a *similarity matrix* and a set of *gap penalties*. The similarity matrix defines the positive score for matching two symbols or the cost of mismatching two symbols. The gap penalties specify the score or cost of opening up a gap in a string and extending that gap in order to improve the fit with another string. Although the similarity matrix and gap penalties can be adjusted to alter the behaviour of the algorithm, in general it is not known which values are best suited for optimisation problems. In this study the similarity matrix is

	L	C	R	M
L	3	−2	−2	−2
C	−2	3	−2	−2
R	−2	−2	3	−2
M	−2	−2	−2	3

while the gap open and gap extend penalties are −3 and −1 respectively.

In this paper two distance functions, defined on sequences of heuristic selections, are considered: a distance function based on the Smith-Waterman algorithm and for comparison purposes, the Hamming distance.

The Smith-Waterman algorithm can be used to construct a simple notion of distance d between sequences. In symbols

$$d(s_1, s_2) = \max_{SW} - sw(s_1, s_2)$$

where $\max_{SW}$ is the maximum value that can be attained by the Smith-Waterman function sw on the subsequences under consideration. A low d value indicate that two sequences are similar or close. In this study the maximum Smith-Waterman score over the 1600 sequences of the database is 357. This function should only be loosely interpreted as a distance function as it is not a metric in the formal sense.

4 CLUSTER ANALYSIS

A k-medoid clustering algorithm employing the Smith-Waterman and Hamming distances is used to separate the entire offline learning database of 1600 sequences into 4 clusters corresponding to the 4 HyFlex domains. For clustering purposes, only the sequence selections up to and including the minimum objective function value are used, as these are the selections that are used as learning algorithm inputs. The accuracy of the resulting clusters are evaluated using the four commonly used measures: *purity*, *normalised mutual information (NMI)*, *Rand index*, and the F_5 (see [2]). For each measure, the worst clusterings have values close to 0 while a perfect clustering has a value of 1. The results shown in table 1 demonstrate that the Smith-Waterman distance is superior in each measure.

Table 1: A comparison of clustering accuracy.

Distance	Purity	NMI	Rand	F_5
S-W	0.8269	0.5954	0.7951	0.8001
Hamming	0.5350	0.2955	0.6185	0.7320

5 CONCLUSIONS

This experiment demonstrates that the sequences of heuristic selections produced by a simple hyper-heuristic on the HyFlex problems contain subsequences that are common to each problem domain and that differ significantly between problem domains. These similarities and differences can be identified automatically using the Smith-Waterman algorithm. Specifically, the clusters produced by a k-medoid cluster algorithm using Smith-Waterman are more accurate than those produced using the Hamming distance across 4 standard accuracy measures. The existence of discernible subsequences of heuristics in the database lends weight to the argument that the ordering of a subsequence is crucial to search efficacy and this ordering varies with problem domain. The ability to identify (similar) problem domains from a sample of heuristic selections can also be used to guide the choice of learning algorithm and learning algorithm parameters for unseen problems or those with novel heuristic sets without requiring problem specific information. These results demonstrate the suitability of the Smith-Waterman algorithm as a measure of sequence similarity for offline learning applications.

REFERENCES

[1] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. Woodward. 2010. *A Classification of Hyper-heuristic Approaches.* Springer US. 449–468 pages.
[2] C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press.
[3] G. Ochoa, M. Hyde, T. Curtois, J. A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. J. Parkes, S. Petrovic, and E. K. Burke. 2012. HyFlex: A Benchmark Framework for Cross-Domain Heuristic Search. In *Evolutionary Computation in Combinatorial Optimization*, J. K. Hao and M. Middendorf (Eds.). Springer Berlin Heidelberg, 136–147.
[4] T. F. Smith and M. S. Waterman. 1981. Identification of common molecular subsequences. *Journal of molecular biology* 147, 1 (1981), 195–197.
[5] G. Tsoumakas, D. Vrakas, N. Bassiliades, and I. Vlahavas. 2004. Using the k Nearest Problems for Adaptive Multicriteria Planning. In *in Proceedings of the 3rd Hellenic Conference on Artificial Intelligence, SETN04.* Springer, 132–141.