6th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2013)
27-29 August 2010, Gent, Belgium

MISTA 2013

A Grouping Hyper-heuristic for Graph Colouring

Anas Elhag · **Ender Özcan**

1 Introduction

Many well known combinatorial optimisation problems, such as data clustering, timetabling, and graph colouring are classified as *grouping problems* [6], requiring the same common task of partitioning a set of objects U into a collection of mutually disjoint subsets u_i of U such that each object is in exactly one subset, as follows:

$$\begin{aligned} \bigcup u_i &= U \quad \forall i \\ u_i \bigcap u_j &= \emptyset \quad \forall i, \forall j \quad where\ i \neq j \\ u_i &\neq \emptyset \quad \forall i \end{aligned} \tag{1}$$

Mostly, the overall fitness f of a grouping problem solution $U_g = \{u_1, ..., u_i, ..., u_k\}$ can be measured using an evaluation function which adds up the partial contribution from each group as in Equation 2.

$$f(U_g) = \sum_{i=1}^{k} f(u_i) \tag{2}$$

Assuming that each colour represents a group, then the graph colouring problem [5] can be formulated as a grouping problem in which the task is to assign each vertex of an undirected graph to a group, such that no two connected vertices are in the same group, with the goal of minimising the number of groups. Formally, given a graph $G = (V, E)$ with vertex set V and edge set E, and given an integer k, a *k-grouping* of G is a function $u_i : V \to 1, ..., k$, where u_i of a vertex x is the group of x. If two connected vertices x and y are in the same group u_i, x and y are conflicting vertices, and the edge $E_{x,y}$ is called a conflicting edge. If there are no conflicting edges, then the groups are all independent sets and the *k-grouping* is valid. The graph colouring problem is to determine the minimum integer k (the chromatic number of $G - \chi(G)$) such that there exists a legal *k-grouping* of G.

A. Elhag and E. Özcan
University of Nottingham, School of Computer Science
Jubilee Campus, Nottingham NG8 1BB UK
Tel.: +44 (115) 95 15544
Fax: +44 (115) 846 7877
E-mail: {axe, exo}@cs.nott.ac.uk

Clearly, not all groupings are feasible for all grouping problems, since different grouping problems impose different constraints and so they introduce different objectives. For example, in graph colouring, two connected nodes must not be placed into the same group.

Also, a common feature of the grouping problems as defined is that they require the number of groups k to be minimized as well as the combined fitness of the groups $f(U_g)$. Consequently, grouping problems can be considered to be multi-objective optimisation problems [10]; or more precisely, bi-objective optimisation problems. In the graph colouring problem, the minimum number of colours causing the least conflicts is searched. These objectives are conflicting objectives; i.e optimising one of them will cause the other to deteriorate. Solving a grouping search problem will, hopefully, yield a set of optimal solutions at the end of the search process. Each one of these solutions is better than the others in terms of one of the objectives, and worse in terms of the other. This set of optimal solutions is known as the *pareto-optimal front*. Traditionally, the pareto-optimal front is approximated by a set of what is called non-dominated solutions. A solution U_1 dominates another solution U_2 if (i) U_1 is no worse than U_2 in anyone of the objectives, and (ii) U_1 is better than U_2 in at least one objective.

Hyper-heuristics are methodologies that select or generate heuristics during the search process for solving computationally hard problems [4,8]. A software framework, known as HyFlex, has been recently developed to support the design of cross-domain heuristic search methods particularly hyper-heuristics [2]. The Java version of HyFlex implementing six different problem domains was recently used at the first Cross-Domain Heuristic Search Challenge (CHeSC 2011) [3], which then became a benchmark for selection hyper-heuristics. Each problem domain implementation came with its specific set of low level heuristics. In this study we are aiming to achieve a different level of generality by designing a selection hyper-heuristic framework that is provided with a fixed set of low level heuristics which can be used for solving all grouping problems. The low level heuristics are designed based on the representation suggested as part of the Genetic Grouping Algorithm (GGA)[6]. Our aim is not to beat the state of the art techniques which are designed and tuned for a particular problem domain. We aim to provide the same selection hyper-heuristic framework based on a single point based search approach for all grouping problems. The initial tests using our framework is performed on instances of the graph colouring problem.

2 Experimental Results

Instances from the DIMACS benchmark suite are used for the experiments. For each instance, a range for the suitable number of groups k is selected around the known best number. Then, an initial pareto front is created by building a random solution for each value of k. Each of these solutions is then iteratively improved using a selection hyper-heuristic which perturbs the current solution generating a new one using a chosen low level heuristic and then decides whether to accept or reject the new solution. At the end of the search, we use a method known as *the elbow criterion* to determine the best point in the pareto front. Seven different perturbative low level heuristics were implemented. The main objective of these low level heuristics is to make modifications on the current solutions, such as dividing a group, merging two groups or swapping items between groups. Different hyper-heuristics were tested during the experiments. The experiments were carried out on a 3.6GHz Intel Core i7-3820 Windows 7 machines with a memory of 16GB. Each experiment is repeated 31 times and a run is terminated after 600 seconds.

The results of the experiments show that our proposed framework successfully finds the best solutions for all the data sets. In the overall, a modified version of the (GIHH)[1] hyper-heuristic [7] was found to be the best among the tested hyper-heuristics in terms of improving the pareto fronts as well as finding the best solutions. Also, our framework pushes the initial pareto fronts further than a previously proposed framework that uses a different grouping representation known as linear linkage encoding (LLE) [1]. Yet, the same best-of-runs results are obtained. Figure 1 shows a comparison between the initial and the final pareto front obtained by GIHH for data set DSJC125.5. Figure 2 shows a comparison between the ratios of calls, acceptance and acceptance for best made to each low level heuristic by GIHH hyper-heuristic on DSJC125.5 data set. Table 1 shows a comparison between the best results obtained by our approach (GGA-GIHH) to those obtained by Greedy Partition Crossover Lowest Index (GPX-LI), Greedy Partition Crossover Cardinality Based (GPX-CB) and Lowest Index Max Crossover (LIMX) proposed in [9]. Initial experiments show that our best-of-runs results are competitive with the previously proposed approaches.

More details of the multi-objective framework hyper-heuristic approach as well as more results on some other grouping problem domains will be provided at the conference.

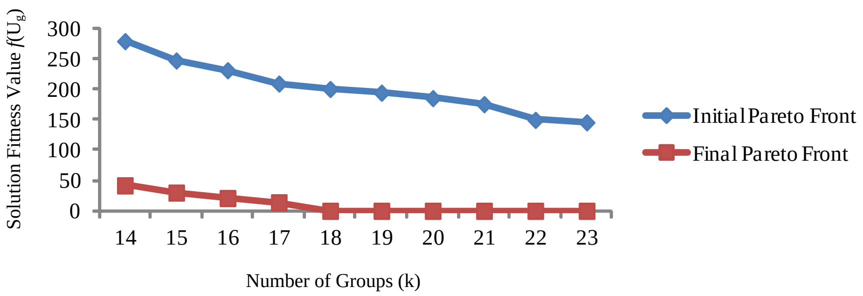

Fig. 1 Comparison between an initial and a final pareto fronts obtained by the AdapHH on DSJC125.5 data set.

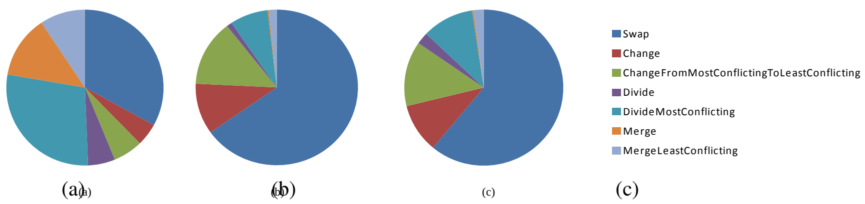

Fig. 2 The ratios of low level heuristics (a) selected at 2,465,530 decision points, (b) accepted considering 910,064 (out of 2,465,530) times, and (c) yielding improvement in the best solution considering 288,151 (out of 2,465,530)by AdapHH on DSJC125.5.

[1] http://www.code.google.com/p/generic-intelligent-hyper-heuristic/downloads/list

Table 1 Best results comparison

Instance	$\chi(G)$	GGA-GIHH	LIMX	GPX-LI	GPX-CB
DSJC125.5	?	**18**	**18**	**18**	**18**
DSJC125.9	?	**44**	**44**	**44**	**44**
DSJC250.1	?	**9**	**9**	**9**	**9**
DSJC250.5	?	**30**	31	31	31
DSJC250.9	?	**74**	75	75	**74**
DSJC500.1	?	**14**	**14**	**14**	**14**
le450_15a	15	**15**	16	16	16
le450_15b	15	**15**	16	16	16
le450_15c	15	**15**	23	23	23
le450_15d	15	**15**	23	23	23
le450_25a	25	**25**	**25**	**25**	**25**
le450_25b	25	**25**	**25**	**25**	**25**
le450_25c	25	**25**	28	28	28
le450_25d	25	**25**	28	28	28
Wins/Draws		**7/7**	0/6	0/6	0/7

References

1. Birben, M.: Solving grouping problems using a selection hyper-heuristic framework based on linear linkage encoding. Tech. rep., Yeditepe University, Computer Engineering (2011)
2. Burke, E., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.: Hyflex: A flexible framework for the design and analysis of hyper-heuristics. In: Proceedings of the Multidisciplinary International Scheduling Conference (MISTA09), pp. 790–797 (2009)
3. Burke, E., Gendreau, M., Hyde, M., Kendall, G., McCollum, B., Ochoa, G., Parkes, A., Petrovic, S.: The cross-domain heuristic search challenge - an international research competition. In: X. Yao, C.A.C. Coello (eds.) Proceedings of Learning and Intelligent Optmization (LION5), *Lecture Notes in Computer Science*, vol. 6683, pp. 631–634 (2011)
4. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: A survey of the state of the art. Tech. Rep. NOTTCS-TR-SUB-0906241418-2747, University of Nottingham, School of Computer Science (2010)
5. Chiarandini, M., Stützle, T.: An analysis of heuristics for vertex colouring. In: P. Festa (ed.) Experimental Algorithms, Proceedings of the 9th International Symposium, (SEA 2010), *Lecture Notes in Computer Science*, vol. 6049, pp. 326–337. Springer (2010)
6. Falkenauer, E.: Genetic Algorithms and Grouping Problems. John Wiley & Sons, Inc., New York, NY, USA (1998)
7. Misir, M., Verbeeck, K., Causmaecker, P.D., Berghe, G.V.: An intelligent hyper-heuristic framework for chesc 2011. In: Y. Hamadi, M. Schoenauer (eds.) LION, *Lecture Notes in Computer Science*, vol. 7219, pp. 461–466. Springer (2012)
8. Özcan, E., Bilgin, B., Korkmaz, E.E.: A comprehensive survey of hyperheuristics. Intelligent Data Analysis **12**(1), 3–23 (2008)
9. Ülker, O., Özcan, E., Korkmaz, E.E.: Linear linkage encoding in grouping problems: Applications on graph coloring and timetabling. In: International Conference on the Practice and Theory of Automated Timetabling (2006)
10. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput **8**(2), 173–195 (2000)