Using Division to Classify Low-level Heuristics for the Quadratic Assignment Problem

Fan Xue, C.Y. Chan, W.H. Ip, and C.F. Cheung

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong
{mffxue,mfcychan,mfwhip,mfbenny}@inet.polyu.edu.hk

1 Division: a source of feedbacks for hyper-heuristics

In hyper-heuristics, online learning collects feedbacks from one instance, and off-line learning collects from many instances of the underlying problem for classifying low-level heuristics (LLHs) [1]. Division can also provide feedbacks for learning. Given a problem, a division is defined on a subset of the variables with the corresponding parts of the objective function and the constraints.

Take the Quadratic Assignment Problem (QAP) as an example. A QAP P is described as follows: Given a cost coefficient matrix $C = ||c_{ijpq}||$ ($c_{ijpq} = d_{ij}w_{pq}$, $i, j, p, q = 1, \ldots, n$), determine an $n \times n$ solution matrix $X = ||x_{ij}||$ so as to

$$\begin{aligned}
\min \ & \sum_{ij} \sum_{pq} c_{ijpq} x_{ij} x_{pq} \\
\text{s.t.} \ & \sum_{j} x_{ij} = 1 \\
& \sum_{i} x_{ij} = 1 \\
& x_{ij} = 0 \text{ or } 1.
\end{aligned}$$

A division D_P is defined on a new cost coefficient matrix $C' = ||c'_{ijpq}||$ ($i, j, p, q = 1, \ldots, n'$ and $n' \leq n$), its goal is to find an $n' \times n'$ solution matrix $X' = ||x_{ij}||$, where there exist an injective function $g(y) : \{1, \ldots, n'\} \mapsto \{1, \ldots, n\}$ so that $c'_{ijpq} = c_{g(i)g(j)g(p)g(q)} = d_{g(i)g(j)} w_{g(p)g(q)}$ for each entry c'_{ijpq}. The equations of objective function and constraints formally remain the same. The inherited similarity makes it possible to reuse the feedbacks of LLHs from D_P for P.

In this paper, the target hyper-heuristic is the Pearl Hunter (PHunter) [2]. PHunter employs an off-line classification to decide a mode consisting of an LLH portfolio and a way of local search. The off-line learning is replaced with an elementary learning to lessen errors: Test each mode in the same division and select the mode with the best result. In fact, advanced learning methods could also be carried out on divisions, such as the Class Association Rules [3].

HyFlex[1] (Hyper-heuristics Flexible framework) [4] is a Java cross-domain platform with a standard hyper-heuristic interface for definitions and calls of

[1] See http://www.asap.cs.nott.ac.uk/chesc2011/hyflex_description.html

Table 1. 20 LLHs implemented for QAP

Group	Method	Num of LLHs: {RoE}	Reference
Crossover	Partially Matched Crossover	3:$\{0,1,2\}$	[5]
	Order Crossover	3:$\{0,1,2\}$	[5]
	Voting recombination crossover	3:$\{0,1,2\}$	—
Mutation	Shifting mutation	1: $\{0\}$	[6]
	Spiral reassignment	1: $\{0\}$	[5]
	Random mutation	3:$\{0,1,2\}$	—
Ruin-recreate	GRASP	3:$\{0,1,2\}$	[7]
	Chan et al.'s	1: $\{0\}$	[8]
Local search	Variable Depth Search (VDS)	1: —	[9]
	Robust tabu search	1: —	[10]

LLHs. In this research, 20 LLHs were implemented in 4 groups as shown in Table 1: crossover, mutation, ruin-recreate and local search. A "range of effect" (RoE) of diversification was introduced. During the perturbation, when a row i in solution matrix subjects a change, the "nearby" row j or rows ($d_{ij} \leq$ RoE) needs a change too.

2 Selecting a presentative division for hyper-heuristics

The selection criteria of division are dependent on n' and g. n' can be easily determined by the estimated complexity of the algorithm and the expected computation time on the division. For an $\mathrm{O}(n^{\mathrm{k}})$ algorithm, $n' = \sqrt[\mathrm{k}]{\frac{t_{D_P}}{t_P}}n$, where t_{D_P} is the expected time on the division and t_P is the time of solving the given problem. Function g is measured by the mean $M_d^{'}$ and the standard deviation $D_d^{'}$ of $\{d_{ij}\}$ and the mean $M_w^{'}$ and the standard deviation $D_w^{'}$ of $\{w_{pq}\}$.

In the sensitivity sampling, the expected time $t_{D_P} = 0.05t_P$, hence $n' = \sqrt[4]{0.05}n = 0.47n$. Factors $M_d^{'}$, $D_d^{'}$, $M_w^{'}$ and $D_w^{'}$ of D_P by random selection and nearest area selection of g seem not significantly different after thousands times of each selection: All factors ranged from about 0.94 to 1.06 times of those of P. The sensitivity samples were tested as Fig. 1. Each factor was from 0.95 to 1.05 while others were locked to 1, ten trials for each value. Results show the first three factors should be almost the same as those of P for slightly better results.

3 Implementation and Experiments

Tests were carried out on 10 largest benchmark instances from QAPLIB[2]. The average results on $n = 100$ instances were shown in Fig. 2, some details were shown in Table 2. NIFLS, ILS and PMA-SLS were two Iterated Local Search [11,12] and a Parallel Memetic Algorithm with Selective Local Search (10 islands) [13]. PH was PHunter trained by off-line learning on other domains. PH_{D30}, PH_{D40}, PH_{D47}, PH_{D47} and $\mathrm{PH}_{D_{\mathrm{root}}}$ were PHunters learned from divisions with $n' = 30$,

[2] See `http://www.seas.upenn.edu/qaplib/`.

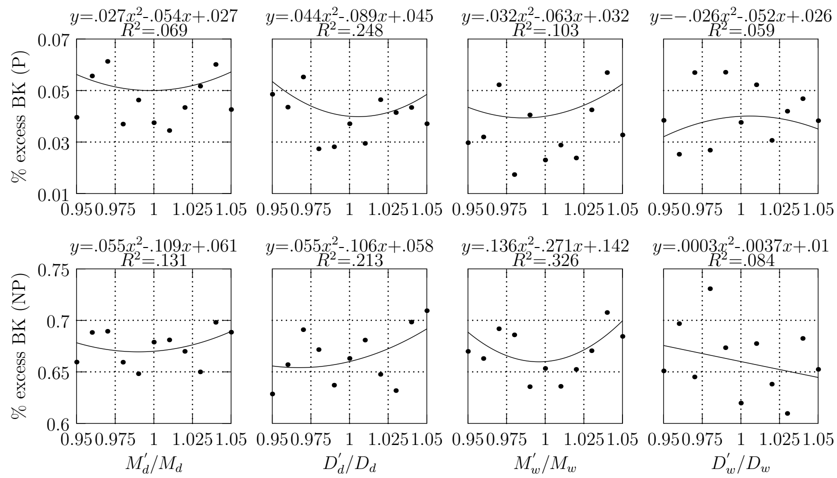

Fig. 1. Sensitivity samples of the four factors and quadratic regressions on planar (P) and nonplanar (NP) QAPs

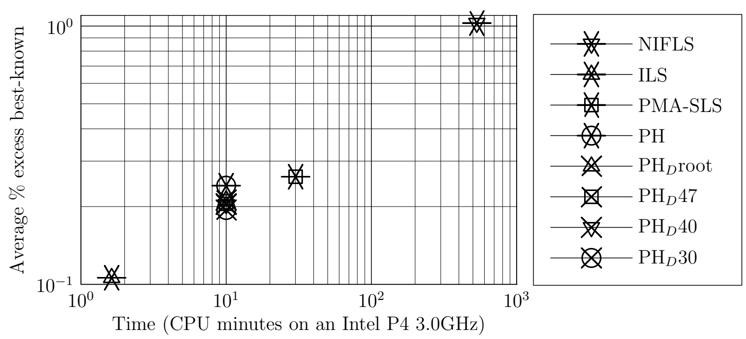

Fig. 2. Comparisons of algorithms on $n = 100$ benchmark instances

$n' = 40$,$n' = 47$ and $n' = 0.47n$ respectively, then solve P in the rest of time. PHunter showed strong portability and effectiveness even the off-line learning and algorithm code were conducted out of QAP domain. The division demonstrated the capacity and stability to provide feedbacks to guide hyper-heuristics. What is more interesting is the fast and effective learning on the division could possibly enable some "pseudo-online" style off-line learning hyper-heuristics, which can collect feedbacks "online" from divisions of the given instance.

Acknowledgements

The work described in this paper was supported by a grant from the Department of Industrial and Systems Engineering of The Hong Kong Polytechnic University (No. RP1Z).

Table 2. Details of average objective functions found by the algorithms

Instance	Best-known	Percentage excess best-known							
		NIFLS	ILS	PMA-SLS	PH	PH_{D30}	PH_{D40}	PH_{D47}	$\mathrm{PH}_{D\mathrm{root}}$
Sko100a	152002	0.32	**0.012**	0.0663	0.0918	0.0684	0.0766	0.0680	0.0586
Sko100b	153890	0.49	**0.007**	0.0636	0.0412	0.0101	0.0217	0.0415	0.0402
Sko100c	147862	0.34	**0.002**	0.0226	0.0454	0.0093	0.0068	0.0077	0.0093
Sko100d	149576	0.59	0.021	0.0706	**0.0013**	0.0511	0.0298	0.0475	0.0517
Tai100a	21052466	1.83	0.693	1.5684	0.7237	0.5714	0.6205	0.6212	**0.5585**
Tai100b	1185996137	3.36	**Opt**	0.0048	0.7197	0.5992	0.6017	0.5829	0.6981
Tai150b	498896643	—	**0.095**	—	0.9917	1.3362	1.3364	1.2579	1.2418
Tai256c	44759294	0.34	—	—	**0.2682**	0.2914	0.2915	0.3058	0.3088
Tho150	8133398	—	**0.068**	0.1418	0.1448	0.1873	0.1897	0.1926	0.1919
Wil100	273038	0.26	**0.004**	0.0332	0.0671	0.0473	0.0708	0.0693	0.0642
Average of n=100		1.027	**0.106**	0.261	0.241	0.194	0.204	0.205	0.212

References

1. Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., and Qu, R. Hyper-heuristics: A Survey of the State of the Art, School of Computer Science and Information Technology, University of Nottingham, Computer Science. Technical Report No. NOTTCS-TR-SUB-0906241418-2747. (2010)
2. Chan, C. Y., Xue, F., Ip, W. H., Cheung, C. F. A Hyper-heuristic Inspired by Pearl Hunting. (Submitted to LION6)
3. Xue F., Chan, C.Y., Ip, W.H., Cheung, C.F. Towards a learning-based heuristic searching reform scheme, XXIV European Conference on Operational Research (EURO), Lisbon, Portugal. (2010)
4. Burke, E., Curtois, T., Hyde, M., Ochoa, G., Vazquez-Rodriguez, J. A. HyFlex: A Benchmark Framework for Cross-domain Heuristic Search, ArXiv e-prints, arXiv:1107.5462v1. (2011) Available at `http://arxiv.org/abs/1107.5462v1`
5. Yaman, R., Gethin, D. T., Clarke, M. J. An effective sorting method for facility layout construction, Int. J. Prod. Res., Vol. 31, No. 2, pp. 413–427. (1993)
6. Production Scheduling and Soft-Computing Lab. The Genetic Algorithm for Solving the Quadratic Assignment Problem and Continuous Problem. (2005) Available at: `http://peterchenweb.appspot.com/publications/sourceCodes/OpenGA/OpenGA.pdf`
7. Feo T.A., Resende M.G.C. Greedy randomized adaptive search procedures, J. of Global Optimization, Vol 6, pp 109–133. (1995)
8. Chan, W. M., Chan, C. Y., Ip, W. H. A heuristic algorithm for machine assignment in cellular layout, Comput. Ind. Eng., Vol. 44, No.1, pp 49-73 (2003)
9. Burke E.K., Curtois T., Qu R., Vanden Berghe G. A Time Pre-defined Variable Depth Search for Nurse Rostering. TR. NOTTCS-TR-2007-6. University of Nottingham. (2007)
10. Taillard E. Robust taboo search for the quadratic assignment problem, Parallel Computing, Vol. 17, pp. 443–455. (1991)
11. Ramkumar, A. S., Ponnambalam, S. G., Jawahar N. A new iterated fast local search heuristic for solving QAP formulation in facility layout design, Robotics and Computer-Integrated Manufacturing, 25(3):620-629. (2009)
12. Stützle, T. Iterated local search for the quadratic assignment problem, European Journal of Operational Research, Vol 174, No.3, pp. 1519–1539. (2006)
13. Tang, J., Lim, M. H., Ong Y. S., Er M. J. Parallel memetic algorithm with selective local search for large scale quadratic assignment problems, International Journal of Innovative Computing, Information and Control, Vol. 2 No. 6, pp. 1399–1416 (2006)