Noname manuscript No.
(will be inserted by the editor)

HyperILS: An Effective Iterated Local Search Hyper-heuristic for Combinatorial Optimisation

Gabriela Ochoa · Edmund K. Burke

Received: date / Accepted: date

Keywords Hyper-heuristics · Iterated local search · Timetabling · Combinatorial optimisation

1 Introduction

Two powerful ideas from search methodologies, iterated local search and hyper-heuristics, are combined into a simple and effective framework to solve combinatorial optimisation problems (HyperILS). Iterated local search is a simple but successful algorithm. It operates by iteratively alternating between applying a move operator to the incumbent solution and restarting local search from the perturbed solution. This search principle has been rediscovered multiple times, within different research communities and with different names [2,12]. The term *iterated local search* (ILS) was proposed in [11]. Hyper-heuristics [4, 6,7] are a recent trend in search methodologies motivated (at least in part) by the goal of automating the design of heuristic methods to solve computational search problems. The aim is to develop more generally applicable methodologies. Metaheuristics are often used as the search methodology in a hyper-heuristic approach (i.e. a metaheuristic is used to search a space of heuristics). Machine learning approaches can and have also been used as the high-level strategy in hyper-heuristics such as reinforcement learning, case based reasoning, and learning classifier systems [4]. The ILS hyper-heuristic discussed here uses a form of reinforcement learning to adaptively select the best operator/heuristic to apply at each iteration (in either or both the perturbation and improvement stages) from an available pool of operators with different features. It differs from a standard ILS implementation which uses a

G. Ochoa and E. K. Burke
Department of Computing Science and Mathematics
University of Stirling
Stirling FK9 4LA, Scotland
E-mail: {gabriela.ochoa, e.k.burke}@cs.stir.ac.uk

single variation operator for each stage. The proposed approach has similarities with variable neighborhood search [13], and adaptive large neighborhood search [17]. Approaches combining local search and perturbation heuristics can be found in both memetic algorithms [10,15] and hyper-heuristics [3,5,16,22]. A previous approach specifically incorporating adaptive operator selection to iterated local search can be found in [21].

The next section gives more details of the proposed algorithmic framework. Section 3 overviews recent successful applications of HyperILS, while Section 4 present some concluding remarks and suggestions for future work.

2 HyperILS

Algorithm 1 shows the high-level pseudo-code of the proposed iterated local search hyper-heuristic (HyperILS). It differs from traditional ILS implementations in the design of the perturbation and improvement stages. Indeed, Algorithm 1 is a template or framework rather than a specific algorithm as there are alternative ways of designing the stages that we have renamed *Hyper-Improvement* and *HyperPerturbation*. These alternatives, however, share two key components. First, multiple heuristics or neighbourhood structures are considered, instead of a single standard one, in either or both stages. These operators need to be of different types and if possible incorporate some problem domain information in the form of ruin-recreate or large neighbourhood heuristics [17,18]. Second, these multiple heuristics are not selected uniformly at random or in a pre-determined order. Instead, they incorporate state-of-the-art adaptive operator selection and reinforcement learning mechanisms. The next section describes in more detail the adaptive mechanisms that have been used within this algorithmic framework.

Algorithm 1 *HyperILS: Iterated Local Search Hyper-heuristic.*

s_0 = GenerateInitialSolution
s^* = HyperImprovement(s_0)
repeat
  s' = HyperPerturbation (s^*)
  $s^{*'}$ = HyperImprovement(s')
  if $f(s^{*'}) \leq f(s*)$ **then**
    $s^* = s^{*'}$
  end if
until time limit is reached

3 Case studies

The first implementation of HyperILS was presented in [3] using the HyFlex framework as a benchmark for cross domain heuristic search [14]. In this implementation, a perturbation operator is selected uniformly at random (from the available pool of mutation and ruin-recreate heuristics) and applied to the incumbent solution; followed by a greedy improvement stage (using all the local search heuristics). The approach is extended in [5] by substituting the uniform random selection of neighbourhoods in the perturbation stage, by online learning strategies. Two strategies were implemented: *choice function* (from the hyper-heuristics literature), and *extreme value based adaptive operator selection* [9] (from the evolutionary computation literature), with the latter producing better overall results. A subsequent implementation, successfully tested on the vehicle routing problem, incorporated a mechanism for adaptively reordering the improvement heuristics [22]. Finally, HyperILS has been recently applied to solve real-world instances of the Course Timetabling [19]. The approach was found to be both general (solving different types of instances), and effective (producing and even improving some state-of-the-art results).

We describe below in more detail the learning mechanisms used in the previous mentioned HyperILS implementations. Adaptive operator techniques comprise two main stages, *credit assignment* and *operator selection*. Credit assignment involves assigning credit or reward to an operator, based upon potentially a number of factors, including their current performance, past performance and the amount of time since it has last been called. There are a variety of credit assignment methods, all attempting to ensure that the strongest performing operator will have the largest amount of credit apportioned to them. Once the operator merits are estimated an operator selection mechanism is used to define the next operator which will be applied. This cycle of credit estimation and operator selection is repeated through the search process.

Our current implementations of HyperILS [5,19,22] use *extreme* values for assigning credits [8], which is based on the principle that large (but possibly infrequent) credit improvements are more effective than small frequent improvements. It rewards operators which have had a recent large positive impact, while consistent operators yielding only small improvements receive less reward. Rewards are updated as follows, when an operator *op* is selected, it is applied to the current solution. The quality value of this new solution is computed and the change in quality is added to a list of size W. Thereafter, the operator reward is updated to the maximal value in the list.

Operator selection probabilities are calculated from their quality estimates following a selection rule. These rules maintain a probability vector $(p_i, t)_{i=1,\ldots,K}$ (where K denotes the number of operators), and use the operator's raw credit estimate to calculate probabilities. Our studies use two recent and well performing selection rules, namely, *Adaptive Pursuit* and *Dynamic Multi-Armed Bandit*. Adaptive pursuit was originally proposed for learning automata and was adapted to operator selection in [20]. It follows a winner-takes-all strategy, selecting at each step the operator with maximal reward, increasing its selection probability, while all other operators get their probability reduced. This method has two parameters: p_{min} that indicates the minimal probability of selection for each operator, and β, the *learning rate* taken from $(0, 1]$. The multi-armed bandit framework is commonly used in game theory for studying the *exploration vs. exploitation* dilemma. It involves N arms and a decision making-algorithm for selecting one arm at each time step with the goal of maximising the cumulative reward gathered along time. The exploration vs. exploitation balance is also relevant for heuristic search. Indeed, adaptive operator selection can be formulated using multi-armed bandits with arms corresponding to search operators [8]. Specifically, the *upper confidence bound* multi-armed bandit [1] was used as it provides optimal maximisation of cumulative rewards. Two considerations were required to use this framework for adaptive operator selection. First, a scaling factor C is needed, in order to properly balance the tradeoff between exploration and exploitation. Second, the original setting is static, while adaptive operator selection is dynamic, i.e., the quality of the operators is likely to change along the different stages of the search. The multi-armed bandit framework is thus combined with the *Page-Hinkley* statistical test for detecting changes in the reward distribution, and, upon such a detection, restarting the process [8]

4 Conclusions

HyperILS is a simple yet effective framework combining iterated local search and selective hyper-heuristics. It allows the incorporation of state-of-the-art ideas from reinforcement learning and adaptive operator selection. HyperILS has been successfully applied to both cross-domain search, and solving complex optimisation problems such as Vehicle Routing and Course Timetabling. Applications to other complex optimisation problems will be the subject of future research. For the sake of simplicity, the current framework considers a simple acceptance criterion (accepting all non-worsening solutions). Future work will consider a third adaptive stage corresponding to the acceptance mechanism.

Acknowledgements This research is partly funded by the *Engineering and Physical Sciences Research Council* (EPSRC, grant number EP/J017515).

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning **47**(2-3), 235–256 (2002)
2. Baxter, J.: Local optima avoidance in depot location. Journal of the Operational Research Society **32**, 815–819 (1981)
3. Burke, E.K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.A., Gendreau, M.: Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2010), pp. 3073–3080. Barcelona, Spain (2010)
4. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society (JORS) **64**(12), 1695–1724 (2013)
5. Burke, E.K., Gendreau, M., Ochoa, G., Walker, J.D.: Adaptive iterated local search for cross-domain optimisation. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO '11, pp. 1987–1994. ACM, New York, NY, USA (2011)

6. Burke, E.K., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-heuristics: An emerging direction in modern search technology. In: F. Glover, G. Kochenberger (eds.) Handbook of Metaheuristics, pp. 457–474. Kluwer (2003)
7. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.: Handbook of Metaheuristics, *International Series in Operations Research & Management Science*, vol. 146, chap. A Classification of Hyper-heuristic Approaches, pp. 449–468. Springer (2010). Chapter 15
8. Fialho, A., Costa, L., Schoenauer, M., Sebag, M.: Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms. In: Learning and Intelligent Optimization, *Lecture Notes in Computer Science*, vol. 5851, pp. 176–190. Springer Berlin Heidelberg (2009)
9. Fialho, A., Costa, L.D., Schoenauer, M., Sebag, M.: Extreme value based adaptive operator selection. In: Parallel Problem Solving from Nature PPSN X, *LNCS*, vol. 5199, pp. 175–184. Springe (2008)
10. Krasnogor, N., Smith, J.E.: A memetic algorithm with self-adaptive local search: TSP as a case study. In: Genetic and Evolutionary Computation Conference (GECCO 2000). Morgan Kaufmann (2000)
11. Lourenco, H.R., Martin, O., Stutzle, T.: Iterated Local Search, pp. 321–353. Kluwer Academic Publishers,, Norwell, MA (2002)
12. Martin, O., Otto, S.W., Felten, E.W.: Large-step Markov chains for the TSP incorporating local search heuristics. Operations Research Letters **11**(4), 219–224 (1992)
13. Mladenovic, N., Hansen, P.: Variable neighborhood search. Computers and Operations Research **24**(11), 1097–1100 (1997)
14. Ochoa, G., Hyde, M., Curtois, T., Vazquez-Rodriguez, J.A., Walker, J., Gendreau, M., Kendall, G., Parkes, A.J., Petrovic, S., Burke, E.K.: Hyflex: a benchmark framework for cross-domain heuristic search. In: Proceedings of the 12th European conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP'12, *Lecture Notes in Computer Science*, vol. 7245, pp. 136–147. Springer-Verlag (2012)
15. Ong, Y.S., Lim, M.H., Zhu, N., Wong, K.W.: Classification of adaptive memetic algorithms: a comparative study. IEEE Transactions on Systems, Man, and Cybernetics, Part B **36**(1), 141–152 (2006)
16. Özcan, E., Bilgin, B., Korkmaz, E.E.: Hill climbers and mutational heuristics in hyperheuristics. In: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN 2006), *Lecture Notes in Computer Science*, vol. 4193, pp. 202–211. Reykjavik, Iceland (2006)
17. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Computers and Operations Research **34**, 2403– 2435 (2007)
18. Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G.: Record breaking optimization results using the ruin and recreate principle. Journal of Computational Physics **159**(2), 139 – 171 (2000)
19. Soria-Alcaraz, J.A., Ochoa, G., Swan, J., Carpio, M., Puga, H., Burke, E.K.: Effective learning hyper-heuristics for the course timetabling problem. European Journal of Operational Research **238**(1), 77 – 86 (2014)
20. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05, pp. 1539–1546. ACM, New York, NY, USA (2005)
21. Thierens, D.: Adaptive operator selection for iterated local search. In: Second International Workshop on Engineering Stochastic Local Search Algorithms (SLS 2009), *Lecture Notes in Computer Science*, vol. 5752, pp. 140–144. Springer (2009)
22. Walker, J., Ochoa, G., Gendreau, M., Burke, E.: Vehicle routing and adaptive iterated local search within the HyFlex hyper-heuristic framework. In: Y. Hamadi, M. Schoenauer (eds.) Learning and Intelligent Optimization, Lecture Notes in Computer Science, pp. 265–276. Springer Berlin Heidelberg (2012)