The ‘Blackboard’ Pattern for Metaheuristics

Kevin Graham
Computing Science and Mathematics
University of Stirling
FK9 4LA Scotland UK
kgr@cs.stir.ac.uk

Jerry Swan
Department of Computer Science
University of York
YO10 5GH England UK
jerry.swan@york.ac.uk

Simon Martin
Computing Science and Mathematics
University of Stirling
FK9 4LA Scotland UK
spm@cs.stir.ac.uk

ABSTRACT

We describe the ‘Blackboard’ design pattern for metaheuristics which allows multiple agents to combine their expertise opportunistically to contribute towards a solution. Features of the Blackboard pattern may include heterogeneity of solution representations (e.g. both graph and permutation for the TSP) and asynchronous processing, the latter rendering the traditionally hard distinction between ‘online’ and ‘offline’ activity less significant.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

Blackboard Architecture; Design Patterns; Metaheuristics

1. INTENT

Allow multiple metaheuristics (potentially operating on different types of solution representation) to co-operate to solve a problem.

2. FORCES ACTING

NP-complete optimization problems do not admit of a singular *a priori* solution strategy which is “good enough, fast enough”. This suggests that multiple metaheuristics, potentially operating at different process granularities, might usefully be combined. This is particularly the case when:

- **One wishes to incorporate rich and varying forms of domain knowledge**. It is well known that domain knowledge is key in making combinatorially-hard problems tractable [10]. However, it has proved challenging to design metaheuristics that allow the incorporation of arbitrary domain knowledge (i.e., ‘white-box’ information) into the search process.
While attempts have been made in the past to increase the level of domain knowledge used, (e.g., Tabu Search [6, 7] and Squeaky Wheel Optimization [8]), there is no general and domain-agnostic method for making domain knowledge available.

- **A good solution strategy is opportunistically data-driven**.
According to the ‘No-Free-Lunch Theorem’ [14], when considered uniformly across all ‘black box’ problems, there is no single best metaheuristic. This has lead to the development of so-called ‘hyper-heuristic’ approaches [2] that dynamically select from (and/or generate) collections of operators. It is traditional to consider the selection/generation process to be driven by the search trajectory, but the addition of richer domain information to this trajectory (i.e., making it more overtly data-driven) lacks systematic support across metaheuristics in general.

- **Solution states and operators are heterogeneous**
Metaheuristic approaches traditionally employ a singular solution representation (e.g., either permutations *or* graphs for the TSP, but not both simultaneously). In many cases, search progress made with one representation could be used as a starting point for an alternative representation [13]. With respect to heterogeneity of operators, it is well-known that multi-level approaches can offer benefits such as clear separation of concerns and increased modularity. Since the increasing hierarchical levels of operator application are likely to involve increasing computational cost, this may be well-served by asynchronous processing of computationally expensive computations.

3. THE SOLUTION

Introduced in [5], the structural and control patterns of the *blackboard architecture* have been used successfully to solve a variety of complex and under-specified problems [4].

A blackboard architecture maintains a collection of individual agents or ‘Knowledge Sources’ (KSs) within a globally visible and accessible ‘blackboard’ or ‘workspace’ data structure that contains both state of the search process and any further annotations that might be helpful in directing the search. A Knowledge Source can change the workspace with the side-effect of guiding subsequent knowledge sources to further add to the evolving solution(s). This opportunism inherent in the blackboard model lends itself well to concurrency. Although inter-agent communication (as mediated

GECCO ’15, July 11 - 15, 2015, Madrid, Spain

ISBN 978-1-4503-3488-4/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739482.2768500

via the workspace) is inherent within the model, a KS can be entirely self-contained and requires no permission or assistance from other KSs in the system to contribute to solution state construction [4].

In the blackboard model, the integration of distinct sources of domain-knowledge can be addressed by: a) interoperability between collaborating agents via the shared workspace b) the ability to asynchronously invoke operators at different temporal scales. Individual KSs communicate via the shared workspace in a manner which may be meaningful to other KSs, which may then exploit this information at some subsequent stage. In this scheme, each KS is considered to be a black-box, in which the exact implementation of a particular unit of collaboration is unimportant: irrespective of whether the implementation approach is an artificial neural network, integer programming, genetic algorithm etc., each KS can contribute to the solution at hand. Multi-level search is particularly facilitated in blackboard systems in that higher-level searches can proceed concurrently with lower-level activity. This further simplifies the communication between complete and partial solution representations, e.g. constructive KSs operating on partial solution state have the option of making any resulting complete solution state available to perturbative KSs.

An explicit control mechanism is used to guide the search process by facilitating opportunistic changes to the blackboard data structure by online scheduling of Knowledge Sources. The control mechanism selects a course of action based on (i) the current state of the blackboard and (ii) the historical and potential contribution of a 'triggered' KS which claims to be pertinent to the solution at hand.

4. CONSEQUENCES

- **Modularity**
  Modularity of Knowledge Sources provides a clear separation between control and the domain knowledge. The orthogonalisation of both control and domain knowledge means that different control strategies can be 'plugged in' without necessitating a change in the manner in which domain knowledge is represented.

- **Concurrency**
  Blackboard architectures allow different problem-solving trajectories (paths through the search space) to be followed concurrently [3, 13]. Within a blackboard model, concurrent but individual knowledge sources share knowledge of search trajectories via the workspace. However, because of the bottleneck necessitated by the mutual exclusion access to blackboard state, these benefits are most applicable if the 'useful work' a KS does in searching is large relative to the amount of access it requires to the blackboard.

- **Opportunistic Problem Solving**
  A blackboard architecture exemplifies opportunistic problem solving: the knowledge source to be applied can be strongly guided by workspace state, resulting in a method for solution generation that is highly domain-driven [12].

- **Accommodates dynamic introduction of information**
  During run-time, when new relevant sources of information become available (e.g., via landscape metrics which are expensive to compute, or other machine learning activity on the search trajectory), this can activate KSs that know how to take advantage of this information.

- **Interoperability of operators and solution representations**
  The blackboard pattern builds upon the notion of *syntactic interoperability*, defined as the capability of multiple systems/subsystems to communicate and exchange data using a pre-specified format. Blackboard systems provide syntactic interoperability through the use of standardized representations of the elements of workspace state. In this way, KSs are able to collaborate in building a a shared representation, while nonetheless remaining agnostic as to the internal mechanisms of other KSs.

- **Direct support for Multi-level search.**
  Multi-level search is readily supported by the blackboard model. For example, Booch et al. [1] describe the application of a blackboard architecture to a crypt-analysis problem, where KSs have specific expertise at various hierarchical levels of (structural and temporal) granularity, viz. at the level of letter substitution, the level of word substitution and the level of sentence structure. Successive levels might reasonably be assumed to require greater levels of computational effort, but the implied concurrency mechanism can be used to render such issues transparent if so desired.

5. EXAMPLES

- **MAGMA: Multi-Agent Architecture for Metaheuristics**
  MAGMA [11] was designed as both a practical and a research framework for metaheuristic techniques, wherein metaheuristics can be seen as arising from the interactions of multiple worker agents of various types, working at different hierarchical levels and granularities. In this scheme, agents can be categorized based on the 3 primary architectural levels. Agents in the first level (Level 0) are concerned with the construction of solutions and Level 1 agents improve already constructed solutions (i.e., act perturbatively). Level 2 agents are responsible for providing high-level search strategies. It is claimed by Milano & Roli [11] that this scheme can easily accommodate existing metaheuristics approaches and also to provide a means to further extend them. A number of communication mechanisms between these 'heuristic agents' are provided, including a global blackboard and message-passing.

- **Cooperative Multi-Blackboard Search**
  In work by Martin et al. [9], a blackboard architecture is described in which each agent has a blackboard of good edges it has identified and maintains. The agents then share these good edges with each other.

The system therefore essentially takes the form of a multiple blackboard system where each agent's blackboard can be considered as being its "own" view of the search space.

- **Hyper-heuristic Design Space**
A concise example of the utility of the blackboard pattern can be found in [13], where it is applied to boolean satisfiability problems. Compared to token-ring and proportional selection hyper-heuristics, the blackboard variant was shown to produce significantly better results in all test case instances.

6. REFERENCES

[1] Grady Booch. *Object Oriented Analysis & Design with Application*. Pearson Education India, 2006.

[2] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. R. Woodward. *A Classification of Hyper-heuristic Approaches*, chapter Handbook of Meta-Heuristics, pages 449–468. Kluwer, 2010.

[3] Norman Carver and Victor Lesser. Evolution of Blackboard Control Architectures. *Expert Systems with Applications*, 7(1):1–30, 1994.

[4] Daniel D Corkill. Blackboard systems. *AI expert*, 6(9):40–47, 1991.

[5] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys (CSUR)*, 12(2):213–253, 1980.

[6] Fred Glover. Tabu Search-Part I. *ORSA Journal on computing*, 1(3):190–206, 1989.

[7] Fred Glover. Tabu Search-Part II. *ORSA Journal on computing*, 2(1):4–32, 1990.

[8] David E Joslin and David P Clements. "Squeaky Wheel" Optimization. *Journal of Artificial Intelligence Research*, 10(1):353–373.

[9] Simon Martin, Djamila Ouelhadj, Pieter Smet, Greet Vanden Berghe, and Ender Özcan. Cooperative search for fair nurse rosters. *Expert Systems with Applications*, 40(16):6674–6683, 2013.

[10] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, 2nd edition, 2004.

[11] Michela Milano and Andrea Roli. MAGMA: a multiagent architecture for metaheuristics. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 34(2):925–41, April 2004.

[12] H. P Nii. Blackboard systems. Technical report, Stanford, CA, USA, 1986.

[13] Jerry Swan, John R Woodward, Ender Özcan, Graham Kendall, and Edmund K Burke. Searching the Hyper-heuristic Design Space. *Cognitive Computation*, 6(1):66–73, 2013.

[14] David H Wolpert and William G Macready. No Free Lunch Theorems for Optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.