Low Knowledge Algorithm Control for Constraint-Based Scheduling

TOM CARCHRAE [1]

1 Introduction

The central thesis of this dissertation is that *low knowledge* control methods for optimization algorithms allow non-experts to achieve high quality results from optimization technology. The primary motivation for our research is to extend the reach of optimization technology by making it more accessible. To this end, we are interested in methods that not only provide high quality solutions, but do so without the requirement for significant effort and expertise on the part of a practitioner who wants to use off-the-shelf methods to solve a problem.

Traditional *high knowledge* approaches [13, 8, 17, 9, 10, 5, 1, 11] to the algorithm selection and control problem have focused on building models of problem structure and algorithm performance. While high knowledge approaches have had success on specific classes of problems, they have not succeeded in making optimization technology easier to use in general. The use of such models has shifted the expertise requirement from algorithm engineering to feature engineering and predictive model building. In contrast, our methods make control decisions based on very general features that are independent of the type of problem being solved and the algorithms that are being employed. The *low knowledge* feature explored in this dissertation is the improvement of solution quality over time, which we believe is useful in many applications. Our empirical studies show that these control methods achieve strong performance on a range of problems, including problems that are far larger than typical academic benchmarks and more like those seen in industry. The strong performance is not the primary result; it is that these results are possible using simple general control methods that do not require significant effort or expertise.

In particular, in this dissertation:

- We create and investigate mechanisms for algorithm control that do not require detailed knowledge of the problem domain or of algorithm behaviour. These low knowledge control methods make decisions based only on observations of algorithm performance. We claim these approaches reduce the engineering effort and required expertise to effectively apply optimization algorithms.
- We compare a low knowledge control approach to idealized high knowledge approaches in the domain of scheduling algorithms. We present an analysis against the best possible high knowledge approaches and observe competitive performance. Although such methods have been shown to provide good performance, we claim they are impractical on the basis that they shift expertise from analysis of algorithm performance to an analysis of high knowledge models.
- We apply low knowledge control methods to a large neighbourhood search configuration for solving industrial-sized scheduling problems. The control mechanisms are applied to the selection of neighbourhood heuristics during search. In addition, a tuning procedure for combining neighbourhood heuristics is presented. The best performing control methods perform well across all problem sets and time limits.

2 Motivation

This thesis is motivated by the observation that, often, modelling is easy but solving is hard. In practice, modelling becomes hard because we want solving to be easy and the two aspects of problem solving are certainly dependant on each other [15]. Commercial libraries of problem solvers are sold under the premise that they will solve your model using state-of-the-art algorithms. In some cases these work well, but in many cases they produce mediocre results, and in other cases they fail to produce solutions at all. So, in practice, the application of optimization technology requires the expertise to model the problem *and* to configure the algorithms to produce high quality solutions. The quality of the system is determined by the capability and experience of these experts, as shown in the applied study of Le Pape et al. [6].

In this dissertation we do not address the modelling issue. Instead, we address the issue of algorithm configuration, with the goal of producing a system that comes closer to the promise of declarative programming that 'the user states the problem, the computer solves it' [2]. The future, as we see it, is one where toolkits of algorithms are available off-the-shelf. Algorithm experts are used to build toolkits rather than configure them. Such toolkits provide the raw components for an automated system to configure and apply. What remains to be seen is if this configuration exercise can be performed while reducing the reliance on an algorithm expert to effectively use this technology.

The motivation for the work in this dissertation is as follows:

1. **The success of optimization technology is hampered by lack of experts -** The primary driver for our research is that optimization technology suffers from a bottleneck of available experts to implement solutions [2, 12, 3]. Experts are required to gather requirements, create a computational model of the problem, and then develop a system to produce high quality solutions to the computational model. In this dissertation, we address the challenge of reducing the expertise required to develop a system that produces high quality solutions for a given computational model.
2. **Algorithm performance is often brittle -** In many problem domains, the world is constantly changing. Because of change, optimization systems that perform well at first, may start to fail as

[1] University College Cork,Ireland, email: tom@carchrae.net

the problems they are solving change in subtle, or not so subtle ways. We are interested in producing an optimization system that is able to adapt to changes in the world. Such behaviour represents a significant practical cost savings, since it is not only expensive to repair a failing system; costs are incurred, before (or if) the failure is noticed, when the decisions produced by such a system are used in operations.

3. **There is no dominating algorithm** - There is a tendency in the field of optimization to focus on average performance and claim that the algorithm that is best on average is superior to all algorithms. This has led to many papers claiming that, for a particular time limit, method X is better than method Y on problem sets A, B and C. While such results are useful as a measure of incremental improvement, there is an implicit assumption that, eventually, a single algorithm will be developed that dominates all others on all problem sets. Perhaps this is the case on particular problem instances, or even entire problem sets, and maybe across time limits as well. However, as the range of problems to be solved increases, the chance that a single method will dominate decreases [16]. This is especially true when optimization methods are applied to real world problems rather than academic benchmarks. Most methods will exhibit strong performance on some problems at some time limits. Therefore, in a fashion similar to boosting in machine learning [14, 4], we are interested in control mechanisms to combine ensembles of algorithms to produce a system that is ultimately more robust, and performs better, than a single method that performs best on average. This observation, in relation to optimization algorithms, was first made in Leyton-Brown et al. [7] and we continue this direction of research.

3 Overview of Dissertation

This dissertation is structured as follows. First, we present a brief review of the concepts of constraint programming and their application to job shop scheduling. This gives the context for the algorithm control problem by introducing classes of algorithms that require significant expertise to use effectively. In the next two chapters, we discuss the algorithm control problem and point to failings of the existing approaches, with respect to the requirement of reducing expertise. We suggest a low knowledge approach to overcome this problem. In the remaining chapters, we explore the application of our approach and evaluate performance against idealized high knowledge approaches.

Chapter 2 reviews constraint programming and how it can be applied to solve optimization problems and, in particular, scheduling problems. We define the core concepts in constraint programming and a constraint model for the job shop scheduling problem. We then discuss three general classes of algorithms to solve constraint programming problems, which lays the groundwork for later chapters where we present concrete examples of these algorithms.

Chapter 3 presents a detailed review of algorithm control methods. First, we present a framework to classify control mechanisms based on when they make decisions and when they capture knowledge. Then we review control methods from the literature and place them in the context of this framework. We present a discussion on the benefits and limitations of each instantiation of the framework.

Chapter 4 states our thesis: that a low knowledge approach is a practical way to achieve the goal of expertise reduction. We present a discussion on the benefits of high versus low knowledge control, and show that although theoretically, a high knowledge approach is superior, in practical terms, it is inferior since it is more prone to error, in addition to requiring significantly more expertise and effort to implement.

Chapter 5 investigates the use of low knowledge control methods applied to scheduling algorithms. We present three state-of-the-art scheduling algorithms and evaluate the performance across time limits and problem sets. We show that, without the effort or expertise of knowledge engineering, the low knowledge control methods perform strongly when compared to optimal high knowledge selection methods. An ablation study is performed to determine the impact of the components of the best control method.

Chapter 6 investigates the use of low knowledge control methods applied to large neighbourhood search. We present a large neighbourhood search algorithm for scheduling and four neighbourhood heuristics and discuss the challenge of effectively configuring such a system. We introduce a low knowledge method for tuning combinations of neighbourhood heuristics. The neighbourhood heuristics and control methods are evaluated on medium (400 activities) and large (1600 activities) scheduling problems, and then on a set of standard benchmarks.

Chapter 7 concludes with a review of the contributions of the dissertation and a discussion of areas for future work.

Appendix A presents detailed results on Taillard's benchmark problems.

4 Summary of Contributions

The contributions of this dissertation are:

1. A framework that defines the structure of control methods through knowledge capture and control decisions. This framework provides a tool to categorize and understand the literature, resulting in an analysis of the benefits and shortcomings of on-line and off-line control approaches.
2. The introduction of the idea of low knowledge algorithm control. We propose the use of low knowledge control methods to directly address the issue of the expertise required for the effective use of optimization technology. We challenge the belief that more information leads to a better reasoning system and propose that low knowledge control systems are both easier to implement and more robust to change.
3. The demonstration that low knowledge control methods can reduce required expertise while achieving good performance. Several control methods are presented and evaluated against optimal high knowledge selection methods. Empirical results suggest that low knowledge approaches can perform as well or better than high knowledge approaches.
4. An extension of the algorithm selection problem [13] to the more general problem of algorithm control, where algorithm selection is repeated during search. The algorithm control approach allows the interleaving of knowledge capture and decision making. We show that low knowledge control methods are able to perform as well as high knowledge selection methods when applied to scheduling algorithms.
5. An analysis of the impact of components of a low knowledge control strategy through an ablation study. This analysis is executed by producing variants of the control method that lack one of the components. We show empirically that all of the components benefit performance.
6. The evaluation of low knowledge control applied to a large neighbourhood search to solve industrial sized scheduling problems. We present a tuning method for optimizing the parameters of combined neighbourhood heuristics. Two novel neighbourhood

heuristics are presented: a general purpose large neighbourhood search heuristic based on solution cost impact and a heuristic that focuses scheduling effort on the resources with the highest load.

REFERENCES

[1] E. K. Burke, S. Petrovic, and R. Qu, 'Case-based heuristic selection for timetabling problems', *Journal of Scheduling*, **9**(2), 115–132, (2006).

[2] E. C. Freuder, 'In pursuit of the holy grail', *Constraints*, **2**(1), 57–61, (1997).

[3] E. C. Freuder, 'Holy grail redux', *Constraint Programming Letters*, **1**, 3–5, (2007).

[4] Y. Freund, 'Boosting a weak learning algorithm by majority', in *Proceedings of the Third Annual Workshop on Computational Learning Theory*, (1990).

[5] C. Gebruers, B. Hnich, D. G. Bridge, and E. C. Freuder, 'Using CBR to select solution strategies in constraint programming.', in *Proceedings of the Sixth International Conference on Case-Based Reasoning*, pp. 222–236, (2005).

[6] C. Le Pape, L. Perron, J. C. Régin, and P. Shaw, 'Robust and parallel solving of a network design problem', in *Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming*, pp. 633–648, (2002).

[7] K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham, 'Boosting as a metaphor for algorithm design', in *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming*, (2003).

[8] K. Leyton-Brown, E. Nudelman, and Y. Shoham, 'Empirical hardness models: Methodology and a case study on combinatorial auctions'. Under Review, http://www.cs.ubc.ca/ kevinlb/papers/EmpiricalHardness.pdf, 2008.

[9] E. Nudelman, K. Leyton-Brown, A. Devkar, Y. Shoham, and H. H. Hoos, 'SATzilla: An algorithm portfolio for SAT', in *Proceedings of the Seventh International Conference on Theory and Applications of Satisfiability Testing, SAT 2004 Competition: Solver Descriptions*, pp. 13–14, (2004).

[10] E. Nudelman, K. Leyton-Brown, H. H. Hoos, A. Devkar, and Y. Shoham, 'Understanding random SAT: Beyond the clauses-to-variables ratio', in *Proceedings of the Tenth International Conference on Principles and Practice of Constraint Programming*, pp. 438–452, (2004).

[11] E. O'Mahony, E. Hebrard, A. Holland, C. Nugent, and B. O'Sullivan, 'Using case-based reasoning in an algorithm portfolio for constraint solving', in *Proceedings of the Nineteenth Irish Conference on Artificial Intelligence and Cognitive Science*, (2008).

[12] J. F. Puget, 'Constraint programming next challenge: Simplicity of use', in *Proceedings of the Tenth International Conference on Principles and Practice of Constraint Programming*, pp. 5–8, (2004).

[13] J. R. Rice, 'The algorithm selection problem', *Advances in Computers*, **15**, 65–118, (1976).

[14] R. Schapire, 'Strength of weak learnability', *Journal of Machine Learning*, **5**, 197–227, (1990).

[15] B.M. Smith, *Handbook of Constraint Programming*, chapter 11: Modelling, 377–406, Elsevier, 2006.

[16] D.H. Wolpert and W. G. Macready, 'No free lunch theorems for optimization', *IEEE Transactions on Evolutionary Computation*, **1**, 67–82, (1997).

[17] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown, 'SATzilla: portfolio-based algorithm selection for SAT', *Journal of Artificial Intelligence Research*, **32**, 565–606, (June 2008).