Emerging Techniques in Evolutionary Scheduling

David Corne
Parallel, Emergent and Distributed Architectures Laboratory
Department of Computer Science
University of Reading, UK
Email: D.W.Corne@reading.ac.uk

Abstract

This article briefly presents and discusses a selection of emerging ideas for approaches to scheduling optimisation problems. Those discussed mainly arise from the 'evolutionary computing' research community. They are: Optima Linking, Optima Contouring, an Immune System Approach, and 'Mutagenetic Algorithms'. What each of these methods have in common is simply that they are new, novel (modulo, in some cases, clear conceptual links with other established methods), seem promising for general schedule optimisation, and are as yet only sparsely described in the research literature, if at all. Each of the selected new techniques is actually a 'general', or 'weak' algorithm, but happen to be in development by researchers active in the scheduling domain. It is rather unclear whether any of these will endure to earn a reputable place in the library of scheduling optimisation tools, but for the moment they each certainly seem worth further investigation. Some of them are currently under investigation as part of a project by the author dealing with the massive nuclear waste reprocessing and maintenance scheduling problems faced by British Nuclear Fuels Ltd.

Introduction

This article briefly presents and discusses a selection of emerging ideas for approaches to scheduling optimisation problems. The selection is quite eclectic, and represents the authors' personal 'top four' solutions to a multiobjective function taking into account: novelty, general interest, and promise.

Evolutionary scheduling, in general, is a successful and fast growing research area which has already led to several practical systems in place [1]. There are several 'standard' evolutionary scheduling techniques which are now quite well documented and established. The essence of such an 'evolutionary scheduling technique' is a particular method for representing candidate schedules as 'chromosomes' and a particular set of methods for operating on such chromosomes to produce new candidate schedules. One of the new techniques briefly discussed, 'Mutagenetic Algorithms', is an extension of these established evolutionary techniques incorporating the idea of 'mutagens': special agents, themselves subject to artificial evolution, which perform specific operations on candidate schedules.

Two other of the discussed methods, Optima Linking and Optima Contouring, arise from the evolutionary computing research community in a slightly different way. They arise from certain observations about the structure of schedule optimisation fitness landscapes, noticed in [2], for example, which in turn lead to promising ideas for organising the direction of local search.

Another of the methods, the Immune System approach, is inspired by recent work (for example [3]), showing how ideas from the immune system can be applied in computer science and optimisation.

Each of these methods is relatively new, and seem promising for general schedule optimisation, and hence worth bringing to the attention of scheduling practitioners. In fact, however, each of the selected new techniques is really a 'general', or 'weak' method, but it happens to be the case that each is in development by researchers active in the scheduling domain.

No claims are made here as to whether any of these techniques will endure to earn a reputable place in the library of scheduling optimisation tools, but for the moment they each certainly seem worth further investigation. Some of them are currently under investigation as part of a project by the author dealing with the massive nuclear waste reprocessing and maintenance scheduling problems faced by British Nuclear Fuels Ltd.

In the remainder of this article, each are described in turn.

Mutagenetic Algorithms

What practically all evolutionary computation or local search applications have in common is the notion of a fixed set of 'operators'. For example, imagine a simple scheduling application in which a candidate solution can be represented as a sequence of 9 tasks, such as:

3 2 5 1 6 7 9 8 4

In the application, such a list of tasks might, for example, be interpreted as the input to a greedy schedule builder. A typical operator on such a list is the 'adjacent swap' operator. This involves choosing a random pair of adjacent tasks, and swapping them, perhaps changing the above into:

3 2 5 6 1 7 9 8 4

Very many other such operators exist. There are inversion operators, for example, which choose a random contiguous chunk of a sequence and reverse the ordering of tasks in that chunk. There are 'shunt' operators, which shift an entire chunk of a sequence from one place to another, and so on. Similarly, there are a variety of recombination (mult-parent) operators, which have some way of producing a 'child' sequence by somehow combining aspects of two parent sequences.

The idea of a *Mutagenetic Algorithm* (MA) is to add further complexity and specificity to the mutation operators in an evolutionary approach, taking inspiration from [4]. In [4], the role of 'mutagens' in natural evolution is discussed. These are biological vectors which perform specific, specialised mutation operations on chromosomes. In nature, mutagens are themselves subject to natural selection, and [4] develops the thesis that, loosely speaking, mutagens and their targets co-evolve over time with the result that mutagens become better at making mutations which are useful to their targets. A straightforward implementation of this idea in our simple 'sequence' example would be to have a structured class of mutation operators which each specialise in a particular set of operations. These operators vie with each other, and based on feedback concerning the successes or failures of their use, certain operators become more favoured and hence used more often.

A particularly simple example of such a structured class of operators is as follows:

Class: (X X X)
Instances: (1 2 5) , (3 6 8), (1 3 9), ...

In this case, a class is simply a list of three numbers, intended to refer to particular *tasks* in the sequence. An instance of this class, such as "(1 2 5)", describes a mutation operator which takes those particular tasks, wherever they happen to appear in the candidate chromosome, and randomly reorders them.

In general, a structured mutation operator, or 'mutagen', involves a pattern and an action. The pattern describes which elements of the chromosome it operators on, and the action describes what it does. In the above illustration the 'action' is simply assumed to be random reorder. Further possibilities for patterns and actions can be designed at will, of course.

The point of all this is the underlying notion that certain specific types of operation may be particularly useful in an instance of a general scheduling problem. With a mutagenetic algorithm, the idea is to unleash a collection of different such operators, and co-evolve these in line with the population of schedules, in order to draw out and use these particularly useful operators. There is a fair degree of intuitive and other evidence for this view. For example, one of the more successful recent evolutionary algorithm approaches to job shop scheduling problems makes particular use of operators which reorder the tasks along the critical path [5], borrowing from work recommending similar operators within other search techniques [6,7]. This focusses mutation operations towards a particular set of tasks, which itself varies between schedules, and over time, during the course of an algorithm. Whereas the the relative conceptual simplicity of a 'pure' job shop scheduling problem, however, allows the notion of 'tasks on the critical path' to be pinned down as ideal candidates for mutation, more complex and messy real-world problems are also likely to involve particular sets of task more fruitful for mutation in certain ways than others, but far less easily spotted by analytic or argumentative means.

A mutagenetic algorithm attempts to directly find such operators. In its 'adaptive' form. the mutagenetic algorithm will involve a continually changing set of such operators, adapting themselves to the stage of search. However, there is reason to explore a simpler, 'fixed' form, in which a particular set of specialised operators is designed in advance, and then used in ensuing search as a fixed collection. Preliminary evidence (to be further described in a later version of this article) is promising for both methods.

Optima Linking

Optima Linking is a new idea which has been found to show excellent promise in application to scheduling problems. As such, it joins the armoury of methods being researched for scheduling, but Optima Linking arrives with the extra benefit of being useful for probing the structure of scheduling landscapes.

The idea of Optima Linking has been developed independently by Darrell Whitley and by Colin Reeves. It is first described in Rana and Whitley [8], which tests it along with other methods on a suite of standard function optimisation tasks. The essential idea is to generate a path between two previously found local optima; a good point on this path then becomes the starting point for a new local search. Its merit seems clear from the intuitively agreeable notion that the basins of attraction of undiscovered optima are often likely to lie 'between' other local optima. Similar ideas underlay recent work by Glover [9], called 'path-relinking', by Reeves (personal communication), and by Yamada and Nakano [5], achieving excellent results on scheduling benchmarks. In more recent work, Whitley has found Optima Linking to produce better than previous known best results in a benchmark project management scheduling problem (personal communication). These impressive results from early use of Optima Linking and similar ideas are very promising.

In addition to its raw potential as an optimisation technique, Optima Linking is also distinctly useful for exploring fitness landscape structure. To a degree, of course, other search techniques are viewable in this light. Repeated runs of simple hillclimbing, for example, can indicate the relative preponderance of different local optima, and the sizes of their respective basins of attraction. With more sophisticated search methods, however, ability to inform of landscape characteristics is typically compromised by the complexity of the technique itself. The performance of repeated runs of a genetic algorithm on a scheduling problem, for example, may tell us that it was more successful at navigating the 'bumps' than some other method, but more precise landscape information becomes lost, being 'smoothed over' in the dynamics of the algorithm. Optima Linking, in contrast, delivers strong performance while retaining understandable dynamics; its performance is intimately related with the shape of the fitness landscape in a way which can directly suggest pertinent details of landscape structure. Therefore, research on the development of optima linking based algorithms has a natural potential to be able to explain its results and explain the relative successive of the techniques developed in a useful way. Via 'mapping' landscapes with simple Optima Linking, such explanation can also feed back usefully into understanding the relative performance of other techniques on the same problems.

A 'canonical' Optima Linking operation works by iterating the following process: 1) generate a pair of optima using a local search method. 2) Generate a path of solutions between these optima 3) Continue local search starting from the best solution found along this path.

There is clearly massive scope for potentially powerful variations on this theme which retain the essential idea. One tunable aspect is the number of initial points to begin linking from, and regime used to determine the choices of optima to link (analogous to population and selection in a population-based evolutionary algorithm). Another important variable aspect is the method used to generate the linking path. Rana and Whitley's results on function optimisation [8], for example, build the link by at each step finding the 'best' next move along the path. In contrast, [10] finds a faster and simpler Optima Linking method successful in which the path is made simply from successive random steps in the right direction. Meanwhile, the Optima Linking-like notion underlying the recombination operator in [5] uses a local search between two parents, where the search is slightly biased towards the direction of the 'destination' parent.

Further variable aspects of the Optima Linking idea concern the neighbourhood move operator used to generate the linking path, which need not be the same as the operator used in the local search component, the local search method itself, which need not remain fixed, and parametric aspects of the local search method. For example, it is not necessary for the pair of points to be linked to be strict optima; in [10], Corne shows that points generated from rather modest local search effort (and certainly not locally optimal) still yield linking paths with significantly fruitful result. This in itself suggests intriguing aspects of the locations and accessibilities of the basins of attraction of good optima.

Showing particular promise for scheduling, Whitley has investigated the use of an Optima Linking algorithm on a publicly available Resource Constrained Project Scheduling benchmark problem. This problem involves scheduling 575 jobs with precedence constraints and other complex restrictions.

Using a permutation-based representation, an 'adjacent-task swap' local search operator, and linking optima by linking their inverse permutations[1], a straightforward Optima Linking algorithm found a new best solution for this problem, and in fact found this solution in 70% of attempts.

Optima Contouring

Optima contouring is a variant of Optima Linking, differing in a rather critical respect: there is no 'linking' between 'optima'! It recognises the intuition and ap-

[1]The inverse of a permutation indicates, for each element i in turn, how many larger elements precede it in the permutation. Eg: the inverse of {3 0 2 1} is { 1 2 1 0 }

peal behind the notion that paths between diverse optima may yield new basins of attraction, but contends that the crucial aspect of such a path is that it *emerges from a local optimum*, rather than *goes between local optima*. Such a comment must of course be taken in careful context. The 'No Free Lunch' theorems [11] do well to remind optimisation researchers that the valid goal of our efforts is to find ideal matches between algorithms and problem classes, rather than to only find 'ideal algorithms'. Hence, the idea of Optima Contouring is to make a particular assumption about scheduling landscapes, (viz: that they are generally as observed by Mattfeld [2]) and use this assumption in its design. Simply put: Optima Contouring is designed to exploit landscapes in which the distribution of local optima is like a 'massif central', becoming gradually more numerous and densely packed with increasing fitness.

The way it works, inspired of course by Optima Linking, is to first find a single local optimum by local search, and then describe one or more contours (loci of points) at concentric, fixed distances around this optimum. The points found around these contours are used to choose the starting point for the next local search. This process simply iterates with each newly found local optimum. The use of a number of contours at different distances from the optimal point (rather than, say, using one contour and simply restarting search from the best point on it), is meant to avoid being deceived by local 'massif central' like structure, and hence search more successfully for the global massif central cluster. This is achieved by a computationally quick cluster analysis of the contour points, dealing with both genotypic distance and fitness, which yields the presumed most promising point from which to restart the search. In particular, the 'best' point found on a contour may not be chosen, since it may be outweighed by a more dense cluster of quite good points on a different 'side' of the previous optimum.

Early results with this idea on simple job shop scheduling problems seem to show that it improves on Optima Linking, particularly in terms of speed.

Immune System Methods

So far, we have looked at some techniques designed specifically for schedule optimisation with respect to the standard ranges of criteria, such as makespan. A very thorny problem in real world scheduling, however, which academic research has not really focussed on very much, is the need for continual re-scheduling. This translates into a need for *robust* schedules: solutions to scheduling problems which can be quickly and easily altered to provide new, good solutions when circumstances change in (at least) typical ways.

Current research at Edinburgh [12] is looking into an immune-system based method for robust schedule generation. The context is a dynamic job-shop scheduling problem in which jobs arrive continually, and the environment is subject to many unpredictable changes.

The human immune system defends the body against toxins by producing *antibodies*, which recognise foreign molecules (*antigens*). There are unimaginably many potential antigens, but the immune system is able to defend us from a very great number of different antigens with very limited resources. Similarly, in a scheduling situation, ideal solutions should be able to 'cope' with the wide range of potential disruptions that may occur. If we consider these possible disruptions (eg: late arrival of a required resource, breakdown of a particular workstation, etc ...) as antigens, then the natural furtherance of the mapping is to consider a schedule to be an antibody. We therefore ideally require a collection of possible schedules which together cover a great range of possible disruptions to the existing statement of the problem. In practice, this set of schedules can be used as a 'library'; initially, one schedule is put into operation. When a disruption occurs, we search through the schedule library to find a different schedule which can handle the now altered problem. This mirrors the immune system's search through its gene library to find antibodies which can handle a particular new foreign substance. Notice, however, that we require not only a workable new schedule from the library, but one which is maximally similar to the one previously in operation. The altered schedule can then be implemented with minimised overall disruption.

An immune system approach to robust scheduling therefore has the task of generating such a library of schedules, given standard scheduling problem data together with information about the typical changes that may occur. Early work on this notion is very promising [12]. In particular, it has been found to be at least as good (in terms of quality of replacement schedules) and far more efficient, than an earlier more standard range of techniques for rescheduling [13].

Summary

A handful of new techniques have been described which seem to show promise in scheduling. Naturally, this promise arises mainly from the fact that they in some way exploit ideas about the structure of scheduling landscapes. This is particularly true for Optima Linking and Optima Contouring, but also the case for Mutagenetic Algorithms, where, in its adaptive form, the idea is for the details of the operators to evolve over time so as to better grasp the neighbourhood structure. In its 'design' form. the Mutagenetic Algorithm idea is to directly tailor the neighbourhood sampling scheme to the problem instance in question. Meanwhile, in the Immune System case, the idea is not to exploit scheduling neighbourhood structure (although there are clear opportunities and good reasons for doing so in the general framework), but to attack a particularly important aspect of real-world scheduling problems.

These are all fairly 'late-breaking' ideas, which have proved promising in early work. The hope and purpose of this article is to encourage further use and exploration of these ideas. A later version of this article will incorporate more detailed discussion and results on each of these techniques, applied to a particularly nasty real world scheduling problem.

References

[1] Corne, Ross, Practical Issues and recent Advances in Job and Open-Shop Scheduling, in Dasgupta and Michalewicz (eds), Evolutionary Algorithms in Engineering Applications, Springer, 1997. 0-61749-3, pages 39–49.

[2] Mattfeld, Evolutionary Search and the Job Shop, University of Bremen PhD thesis, 1996.

[3] Hightower, Forrest, Perelson, The Evolution of Emergent Organization in Immune System Gene Libraries, in Proceedings of the Fifth International Conference on Genetic Algorithms, pages 344-350, 1993.

[4] Wills, The Wisdom of the Genes: New Pathways in Evolution, Oxford University Press, 1991.

[5] Yamada, Nakano, "Scheduling by genetic local search with multi-step crossover", in *Parallel Problem Solving from Nature - PPSN IV*, Voigt, Ebeling, Rechenberg, Schwefel (eds.), Springer LNCS 1141, 1996.

[6] Laarhoven, Aarts, Lenstra, Job Shop Scheduling by Simulated Annealing, Operations Research 40(1), pp. 113–125, 1992.

[7] Brucker, Jurisch, Sievers, A Branch and Bound Algorithm for the Job Shop Scheduling problem, Discrete Applied Mathematics 49, pp. 107–127, 1994.

[8] Rana, Whitley, "Bit Representation with a Twist", in *Proc. of the 7th Int'l Conf. on Genetic Algorithms*, Morgan Kaufmann, 1997.

[9] Glover, Laguna, "Tabu Search". Chapter 3 of "Modern Heuristic Methods", Reeves (ed.), Blackwell Scientific, 1993.

[10] Corne, Ogden, "Evolutionary and Other Approaches to the Preacher Planning Problem", submitted to *PATAT2, Proc. of the 2nd Int'l Conf. on the Practice and Theory of Automated Timetabling*, Springer, 1998 (to appear).

[11] Wolpert, MacReady, No Free Lunch Theorems for Optimization, IEEE Trans. on Evolutionary Computation 1(1), pp. 67–82, 1997.

[12] Hart, Ross, Nelson, Producing Robust Schedules via an Artificial Immune System, submitted to the 1998 IEEE Conference on Evolutionary Computation, Alaska, 1998.

[13]

Hsiao-Lan Fang, "Genetic Algorithms in Timetabling and Scheduling", PhD Thesis, Department of Artificial Intelligence, University of Edinburgh, 1994