Introduction to the Special Issue
Evolutionary Algorithms for Scheduling

David Montana
BBN Technologies (GTE Internetworking)
10 Fawcett Street
Cambridge, MA 02138
dmontana@bbn.com

Scheduling is a problem that occurs in a large variety of forms with huge cumulative economic and social consequences. Proper scheduling can provide better utilization of scarce and expensive resources as well as higher satisfaction for individuals such as customers and employees. For this reason, there has been a lot of effort expended over a large number of years on algorithms for automatic scheduling. However, except for a few isolated successes on some specialized problems, automatic scheduling remains an open problem.

There are a few reasons why scheduling is such a difficult problem. One is the size and complexity of the search space. For the generic problem of assigning N tasks to M resources with a particular ordering of tasks at each resource, the number of possible solutions is $\begin{pmatrix} N+M-1 \\ M-1 \end{pmatrix} N!$, which implies superexponential growth as a function of the number of tasks and resources. In addition to sheer size, this search space has a more complex topology than a Euclidean space or, when $M > 1$, a space of permutations.

A second reason why scheduling is a difficult problem is that scheduling is an inherently dynamic process. Schedules only remain valid for a limited amount of time. After a certain duration, the world generally has changed enough that the scheduling algorithm has to find a different schedule. The required update time varies greatly between applications, as illustrated by the following sample update time scales:

- milliseconds for scheduling computational tasks on one or more CPUs
- minutes for scheduling service calls
- weeks for scheduling commercial airline flights

In all cases, there are time constraints on how long it can take to schedule and reschedule that place limits on the amount of computation performed by a scheduling algorithm.

A third factor making scheduling difficult is that different domains and applications require solutions of different variations of the scheduling problem. These variations arise from a number of different sources including

- differences in the types of hard constraints, such as relative and absolute temporal restrictions and resource capabilities constraints
- the need for additional information beyond an ordered assignment of tasks to resources, such as absolute times, routes traveled, and manufacturing plans

- different sets of evaluation criteria, such as cumulative response time, throughput, time span, and cumulative employee satisfaction

Traditional operations research (OR) techniques have had very limited success on large, real-world scheduling problems due to the poor scaling properties of such techniques and their lack of flexibility. Special-purpose heuristics, most notably those developed for the traveling salesman problem (Lin & Kernighan, 1973), have very limited applicability to real-world scheduling problems because they are designed for very specific problems. Therefore, in the real world, most large scheduling problems are handled either manually or by simple "dispatch" rules, both of which usually produce suboptimal schedules.

Evolutionary algorithms (as well as other modern heuristics such as simulated annealing) offer the promise of producing optimal schedules for a wide variety of large, real-world problems. The five papers in this issue discuss applications to four very different domains:

- urban transit systems (Deb & Chakroborty)
- supply chain management for a chicken processing factory (Hart, Ross & Nelson)
- exam timetabling (Burke, Newall & Weare)
- flowshop manufacturing (Reeves & Yamada; Cotta & Troya)

Other problems to which evolutionary algorithms have successfully been applied include:

- job-shop scheduling (Davis, 1985; Bagchi et al., 1991; Bruns, 1993)
- scheduling computing tasks (Gonzalez & Wainwright, 1994; Kidwell, 1993)
- scheduling laboratory equipment (Syswerda, 1991)
- crew scheduling (Levine, 1996)
- maintenance/rehabilitation scheduling (Langdon, 1997; Halhal et al., 1997)
- talent/project scheduling (Nordstrom and Tufekci, 1994; Ramat et al., 1997)

The reason for evolutionary algorithms' success at a wide and ever growing range of scheduling problems is a combination of power and flexibility. The power derives from the empirically proven ability of evolutionary algorithms to efficiently find globally competetive optima in large and complex search spaces. The favorable scaling of evolutionary algorithms as a function of the dimension of the search space makes them particularly effective in comparison with other search algorithms for the large search spaces typical of real-world scheduling.

The flexibility of evolutionary algorithms has multiple facets. Even the "standard" genetic algorithm (i.e., bit string representation with traditional crossover and mutation operators) can effectively handle problems that many traditional optimization algorithms cannot including: (i) discrete spaces, (ii) nonlinear, discontinuous evaluation functions, and (iii) nonlinear, discontinuous constraints. The use of "non-standard" evolutionary algorithms and the tailoring of representation, operators, initialization method, etc. to fit the problem/domain greatly increases the range of problems to which evolutionary algorithms can be effectively applied.

Most of the early work on non-standard genetic algorithms was in scheduling and related areas. Goldberg and Lingle (1985) and Grefenstette et al. (1985) invented the order-based representation and associated permutation operators to solve the traveling salesman problem, which is a scheduling problem where an optimal ordering of tasks (cities) is selected for a single resource (salesman). Since then, the order-based approach has been expanded to handle problems where ordering of tasks is not the only information to find. Whitley, Starkweather & Fuquay (1989) and Syswerda (1991) used an order-based representation to solve such problems by utilizing the ordering of tasks in the chromosome not as an end in itself but rather as the input to a greedy schedule builder. Bagchi et al. (1991) used an alternative approach where the additional information is encoded in the chromosome itself; for example, for problems with multiple resources, the resource that each task is assigned to is included in the chromosome along with the ordering of tasks. For this approach, new operators that manipulate the additional information must be defined. Order-based representations have become a "standard" technique for genetic scheduling.

Davis (1985) initiated another approach to representation of schedules using a non-standard genetic algorithm, letting the elements of the chromosome be instructions for how to build a schedule that can be interpreted by a schedule builder. The genetic programming approach to schedule building (Langdon, 1997) also uses chromosomes consisting of sets of instructions on how to build a schedule, except that the instructions are in the form of a (parse) tree rather than Davis' sets of ordered lists.

Besides representation and operators, another important deviation from a standard genetic algorithm common in scheduling is non-random initialization of the population. Using heuristics to choose better-than-random individuals for the initial population often leads to significantly faster convergence to a good solution or, alternatively, better solution in same amount of time (Grefenstette, 1987). Speed is often a key issue in scheduling because of the large search spaces and the constraints on turnaround time, so it is often beneficial to use heuristic initialization (Bruns, 1993) and/or seeding with good individuals from the previous run in the case of rescheduling (Fang, Ross & Corne, 1993; Gonzalez & Wainwright, 1994).

Deb and Chakroborty (this issue) demonstrate the use a of standard (binary-string) genetic algorithm for a problem not suitable for traditional operation research techniques. Genetic algorithms can solve different versions of their bus scheduling problem despite the discreteness, nonlinearity, and high dimensionality of the search spaces. They can use a standard genetic algorithm rather than an order-based one because the orderings are fixed, i.e. buses always travel fixed routes. The free variables to optimize are the arrival times and transfer possibilities for each stop. It is often the case that fixing the ordering allows the use of a standard genetic algorithm to optimize the remaining schedule specifications, such as in (Kidwell, 1993) where what is optimized is the resource assignments.

Cotta and Troya (this issue) apply forma analysis to understanding the behavior of order-based representations and the associated operators. Genetic formae are a generalization of schemata that jointly consider the representation and operators (Radcliffe & Surry, 1995). Cotta and Troya use forma analysis to explain why certain genetic operators are better than others for the permutation flowshop problem. In the process, they help introduce some of the theoretical analysis applied mostly to standard string-based genetic algorithms to the order-based genetic algorithms central to evolutionary approaches to scheduling.

Reeves and Yamada (this issue) define special-purpose operators for an order-based representation based on knowledge of the search space. They investigate the structure of the search space for permutation flowshop problems and found a "big valley" structure, i.e. many closely packed local optima with the goodness of local optima tending to drop off with distance from the global optimum. To exploit this structure, the genetic operators employ path relinking, which is a stochastic search for local optima on a path between the two parents. These operators improve the genetic algorithm performance significantly on this problem (and potentially on other problems with similarly structured search spaces).

Hart, Ross and Nelson (this issue) tackle a large scheduling problem with multiple levels of detail. In practice, this problem is too large to solve with a single large optimization method that must handle all the levels of detail in the hierarchy. In a variation of the order-based representation plus schedule builder approach, one could optimize the order at the highest level of the hierarchy and use a heuristic schedule builder to fill in lower levels of detail. Instead, the authors define multiple heuristics for doing the detailed scheduling associated with a high-level task and include the choice of heuristic for each high-level task in the chromosome along with an ordering of these tasks. This is an interesting hybrid of including additional information in an order-based chromosome and providing instructions for building the schedule in the chromosome.

Burke, Newall and Weare (this issue) address the issue of heuristic initialization for the problem of timetabling. They compare a variety of different heuristics for generating good initial solutions. Each heuristic is stochastic (with a controllable amount of randomness) and hence capable of generating a variety of different solutions. Based on statistical measurements of diversity, they verify that the best initialization methods produce populations of high fitness individuals without sacrificing diversity. They also determine how much randomness in the heuristics is optimal, with too little yielding too little diversity and too much producing low fitness solutions.

Scheduling in its wide variety of forms is a critical problem in today's world. Evolutionary algorithms provide an approach to automatically finding optimal (or nearly optimal) solutions to many scheduling problems with search times that make optimal scheduling practical where it was previously impractical. The papers in this special issue provide important new insights into how to use evolutionary algorithms for optimal scheduling, a subject about which there is still much to learn. Thanks to the authors and reviewers for making this an outstanding issue.

References

Bagchi, S., Uckun, S., Miyabe, Y., & Kawamura, K. (1991). Exploring Problem-Specific Recombination Operators for Job-Shop Scheduling. In R. Belew & L. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms,* pp. 10–17. San Mateo, CA: Morgan Kaufmann.

Bruns, R. (1993) Direct Chromosome Representation and Advanced Genetic Operators for Production Scheduling. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 352–359. San Mateo, CA: Morgan Kaufmann.

Davis, L. (1985). Job Shop Scheduling with Genetic Algorithms. In J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms,* pp. 136–140. Hillsdale, NJ: Lawrence Erlbaum Associates.

Fang, H., Ross, P. & Corne, D. (1993). A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 375–382. San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E. & Lingle, Jr., R. (1985). Alleles, Loci, and the Traveling Salesman Problem. In J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms,* pp. 154–159. Hillsdale, NJ: Lawrence Erlbaum Associates.

Grefenstette, J. J., Gopal, R., Rosmaita, B. J. & van Gucht, D. (1985). Genetic Algorithms for the Traveling Salesman Problem. In J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms,* pp. 160–165. Hillsdale, NJ: Lawrence Erlbaum Associates.

Grefenstette, J. J. (1987). Incorporating Problem Specific Knowledge in Genetic Algorithms. In L. Davis (Ed.), *Genetic Algorithms and Simulated Annealing,* pp. 42–60. Los Altos, CA: Morgan Kaufmann.

Halhal, D., Walters, G. A., Ouazar, D. & Savic, D. A. (1997). Water Network Rehabilitation with Structure Messy Genetic Algorithm. *Journal of Water Resources Planning and Management,* 123(3), pp. 137–146.

Levine, D. (1996). Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations Research,* 23(6), pp. 547–558.

Kidwell, M. (1993). Using Genetic Algorithms to Schedule Distributed Tasks on a Bus-Based System. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms,* pp. 368–374. San Mateo, CA: Morgan Kaufmann.

Langdon, W. (1997). Scheduling Maintenance Using Genetic Programming. In K. Warwick, A. Ekwue & R. Aggarwal (Eds.), *Artificial Intelligence Techniques in Power Systems.* London: IEE Publishing.

Lin, S. & Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the TSP. *Operations Research,* 21(2), pp. 498–516.

Nordstrom, A.-L. & Tufekci, S. (1994). A Genetic Algorithm for the Talent Scheduling Problem. *Computers & Operations Research,* 21(8), pp. 927–940.

Radcliffe, N. J. & Surry, P. D. (1995). Fundamental Limitations on Search Algorithms: Evolutionary Computation in Practice. In J. Van Leeurwen (Ed.), *Computer Science Today: Recent Trends and Developments.* Springer-Verlag.

Ramat, E., Venturini, G., Lente, C. & Slimane, M. (1997). Solving the Multiple Resource Constrained Project Scheduling Problem with Hybrid Genetic Algorithm. In T. Baeck (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms,* pp. 489–496. San Mateo, CA: Morgan Kaufmann.

Syswerda, G. (1991). Schedule Optimization Using Genetic Algorithms. In L. Davis (Ed.), *Handbook of Genetic Algorithms,* pp. 332–349. New York: Van Nostrand Reinhold.

Whitley, D., Starkweather, T. & Fuquay, D. (1989). Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms,* pp. 133–140. San Mateo, CA: Morgan Kaufmann.