

Chapter 1

INTRODUCTION

Edmund K. Burke, Graham Kendall

*Automated Scheduling, Optimisation and Planning Research Group
The School of Computer Science and IT, University of Nottingham, UK*

1.1 INTER-DISCIPLINARY DECISION SUPPORT: MOTIVATION

The investigation of search and optimization technologies underpins the development of decision support systems in a wide variety of applications across industry, commerce, science and government. There is a significant level of diversity among optimization and computational search applications. This can be evidenced by noting that a very small selection of such applications includes transport scheduling, bioinformatics optimization, personnel rostering, medical decision support and timetabling. More examples of relevant applications can be seen in Pardalos and Resende (2002), Leung (2004) and Dell'Amico et al. (1997). The exploration of decision support methodologies is a crucially important research area. The potential impact of more effective and more efficient decision support methodologies is enormous and can be illustrated by considering just a few of the potential benefits: more efficient production scheduling can lead to significant financial savings; higher quality personnel rosters lead to a more contented workforce; more efficient healthcare scheduling will lead to faster treatment (which could save lives); more effective cutting/packing systems can reduce waste; better delivery schedules can reduce fuel emissions.

This research area has received significant attention from the scientific community across many different academic disciplines. Indeed, a quick look at any selection of key papers which have impacted upon search, optimization and decision support will demonstrate that the authors have been based in a number of different *departments* including Computer Science, Mathematics, Engineering, Business, Management, and others. It is clearly the case that the investigation and development of decision support methodologies is inherently multi-disciplinary. It lies firmly at the interface of Operational Research and Artificial Intelligence (among other disciplines). However, not only is the underlying methodology inherently inter-disciplinary but the broad range of

application areas also cuts across many disciplines and industries. We firmly believe that scientific progress in this crucially important area will be made far more effectively and far more quickly by adopting a broad and inclusive multi-disciplinary approach to the international scientific agenda in this field. The way forward is inter-disciplinary.

This observation provides one of the key motivations for this book. The book is aimed primarily at first-year postgraduate students and final-year undergraduate students. However, we have also aimed it at practitioners and at the experienced researcher who wants a brief introduction to the broad range of decision support methodologies that is available in the scientific literature. In our experience, the key texts for these methodologies lie across a variety of volumes. This reflects the broad range of disciplines that are represented here. We wanted to bring together a series of entry-level tutorials, written by world-leading scientists from across the disciplinary range, in one single volume.

1.2 THE STRUCTURE OF THE BOOK

This book was originally motivated by the thought of being able to give first year Ph.D. students a single volume that would give them a basic introduction to the various search and optimization techniques that they might need to use during their research. In this respect the book can be read in a sequential manner. However, each chapter also stands alone and so the book can be dipped into when you come across a technique which you are not familiar with, or just need to find some general references on a particular topic.

If you want to read the book all the way through, we hope that the way we have ordered the chapters makes sense. We start by introducing (in Chapters 2 and 3) some classical search and optimization techniques which, although not always suitable (particularly when your problem has a very large search space), are still important to have in your “tool box” of methodologies. Many of the other chapters introduce various search and optimization techniques, some of which have been used for over 30 years (e.g. genetic algorithms, Chapter 4) and some which are relatively new (e.g. artificial immune systems, Chapter 13). Some of the chapters consider some of the more theoretical aspects of search and optimization. The chapter by Darrell Whitley and John Paul Watson, for example, introduces *Complexity Theory and the No Free Lunch Theorem* (Chapter 11) whilst Colin Reeves considers *Fitness Landscapes* in Chapter 19.

One element of every chapter is a section called *Tricks of the Trade*. We recognize that it is sometimes difficult to know where to start when you first come across a new problem. Which technique or methodology is the most appropriate? This is a *very* difficult question to answer and forms the basis of much research in the area. It often requires experiments with a range of these

techniques. *Tricks of the Trade* is designed to give you some guidelines on how you should get started and what you should do, if you run into problems. Although tricks of the trade is towards the end of each chapter, we believe that it could be one of the first sections you read.

We have also included *Sources of Additional Information* in every chapter. These sections are designed as pointers to useful books, web pages, etc, which might be where you turn to next, once you have read the chapter in this book.

As this book is aimed primarily at the *beginner* (first-year Ph.D. student, final-year undergraduate, practitioner/researcher learning a new technique, etc) we thought it might be useful to explain some basic concepts which many books just assume that the reader already knows. Indeed, our own Ph.D. students and final-year undergraduates often make this complaint. We realize that the following list is not complete. Nor can it ever be, as we are not aiming to write a comprehensive encyclopedia. If you feel that any important terms are missing, please let the editors (authors of this introduction) know and we will consider including them in future editions. All of these concepts are, purposely, explained in an informal way so that we can get the basic ideas across to the reader. More formal definitions can be found elsewhere (see the *Sources of Additional Information* and *References*), including in the chapters of this book.

1.3 BASIC CONCEPTS AND UNDERLYING ISSUES

In this section we will go through a number of basic terms and issues and offer a simple description or explanation. In the spirit of attempting to explain these concepts to beginners, we will restrict the formal presentation of these concepts as much as possible. Instead, we will attempt to explain the basic ideas which underpin the terminology and the (often mathematical) formulations. Many of these terms are described and discussed throughout the book (see the index).

Artificial intelligence Artificial Intelligence is a broad term which can be thought of as covering the goal of developing computer systems which can solve problems which are usually associated with requiring human level intelligence. There are a number of different definitions of the term and there has been a significant amount of debate about it. However, the philosophical arguments about what is or is not Artificial Intelligence do not fall within the remit of this book. The interested reader is directed to the following (small) sample of general AI books: Negnevitsky (2005), Russell and Norvig (2003), Callan (2003), Luger (2002), MacCarthy (1996), Cawsey (1998), Rich and Knight (1991) and Nilsson (1998).

Operational research (Operations research) These two terms are completely interchangeable and are often abbreviated to OR. Different countries tend to use one or other of the terms but there is no significant difference. The field was established in the 1930s and early 1940s as scientists in Britain became involved in the *operational* activities of Britain's radar stations. After the war, the field expanded into applications within industry, commerce and government and spread throughout the world. Gass and Harris, in the preface to their excellent *Encyclopedia of Operations Research and Management Science* (Gass and Harris, 2001), present several definitions. However, as with Artificial Intelligence (above), we are not really concerned with the intricacies of different definitions in this book. The first definition they give says

Operations Research is the application of the methods of science to complex problems arising in the direction and management of large systems of men, machines, materials and money in industry, business, government and defense.

This presents a reasonable summary of what the term means. For more discussion, and a range of definitions, on the topic, see Bronson and Naadimuthu (1997), Carter and Price (2001), Hillier and Lieberman (2005), Taha (2002), Urry (1991) and Winston (2004). For an excellent and fascinating early history of the field see Kirby (2003).

Management science This term is sometimes abbreviated to MS and it can, to all intents and purposes, be interchanged with OR. Definitions can be found in Gass and Harris (2001). However, they sum up the use of these terms nicely in their preface when they say

Together, OR and MS may be thought of as the science of operational processes, decision making and management.

Feasible and infeasible solutions The idea of feasible and infeasible solutions is intuitive but let us consider the specific problem of cutting and packing, so that we have a concrete example which we can relate to. This problem arises in many industries: for example, in the textile industry where pieces for garments have to be cut from rolls of material, in the newspaper industry where the various text and pictures have to be laid out on the page and in the metal industry where metal shapes have to be cut from larger pieces of metal—see Dowsland and Dowsland (1992) for a more detailed review of this area. Of course, all these industries are different but let us consider a generic problem where we have to place a number of pieces onto a larger piece so that we can cut out the smaller pieces. Given this generic problem a feasible solution can be thought of as all the shapes being placed onto the larger sheet so that none of them overlap and all the pieces lie within the confines of the larger sheet. If some of the pieces overlap each other or do not fit onto the larger sheet, then the solution is infeasible. Of course, the problem definition is important when

considering whether or not a given solution is feasible. For example, we could relax the constraint that says that *all* of the shapes have to be placed on the larger sheet, as our problem might state that we are trying to cut out as many of the smaller shapes as possible, but it is not imperative that we include all the smaller pieces. A feasible solution is often defined as one that satisfies the *hard constraints* (see below).

Hard constraints For any given problem, there are usually constraints (conditions) that *have* to be satisfied. These are often called *hard constraints*. To continue with the cutting and packing example from above, the condition that no pieces can overlap is an example of a hard constraint. To take a new example, if we consider a nurse rostering problem, then an example of a hard constraint is the condition that no nurse can be allocated to two different shifts at the same time. If we violate a hard constraint, it leads to an *infeasible* solution. More information about research on nurse rostering problems can be seen in Burke et al. (2004).

Soft constraints and evaluation functions A *soft constraint* is a condition that we would like to satisfy but which is not absolutely essential. As an example, from nurse rostering again, we may have a soft constraint that says that we would like nurses to be able to express preferences about which shifts they would like to work. However, if this constraint is not fully met, a solution is still feasible. It just means that another solution which does meet the condition (i.e. more nurses have their personal working preferences met) would be of higher quality. Of course, there could be many competing soft constraints, which may provide a trade off in the *evaluation function* (measure of the quality of the solution which is also sometimes known as the *objective*, *fitness* or *penalty* function), as the improvement of one soft constraint may cause other soft constraint(s) to become worse. This is a situation where a multi-objective approach might be applicable (see Chapter 10).

Many problems have an evaluation function represented by a sum of each of the penalty values obtained for not satisfying each of the various constraints. Some problems simply ignore the hard constraints in the evaluation function and just disregard infeasible solutions. Another approach is to set a penalty value for the hard constraints but to set it very high so that any solution which violates the hard constraints is given a very high evaluation. A further possibility is to have dynamic penalties so that, at the start of the search, the hard constraints are given relatively low penalty values, so that the infeasible search space is explored. As the search progresses, the hard constraint penalty values are gradually raised so that the search eventually only searches the feasible regions of the search space.

Deterministic search This term refers to a search method or algorithm which always returns the same answer, given exactly the same input and starting conditions. Several of the methods presented in this book are not deterministic i.e. there is an element of randomness in the approach so that different runs on exactly the same starting conditions can produce different solutions. Note, however, that the term “*non-deterministic*” can mean something more than simply not being deterministic. See Chapter 11 for an explanation.

Optimization Within the context of this book, optimization can be thought of as the process of attempting to find the best possible solution amongst all those available. Therefore, the task of optimization is to model your problem in terms of some evaluation function (which represents the quality of a given solution) and then employ a search algorithm to minimize (or maximize, depending on the problem) that objective function. Most of the chapters in this book are describing methodologies which are aiming to optimize some function. However, most of the problems are so large that it is impossible to guarantee that the solution obtained is the optimal one. The term optimization can lead to confusion because it is sometimes also used to describe a process which returns the guaranteed optimal solution (which is, of course, subtly different from the process which just aims to find the best solution possible).

Local and global optimum Figure 1.1 illustrates the difference between a local and global optimum. A local optimum is a point in the search space where all neighboring solutions are worse than the current solution. In Figure 1.1, there are four local optima. A global optimum is a point in the search space where *all* other points in the search space are worse than (or equal to) the current one.

Exhaustive search By carrying out an exhaustive search, you search every possible solution and return the optimal (best) one. For small problems, this is an acceptable strategy, but as problems become larger it becomes impossible to carry out an exhaustive search. The types of problem that often occur in real world search and optimization problems tend to grow very large very quickly. We will illustrate this by considering a very well known problem: the traveling salesman problem (often referred to as TSP). This can be thought of as the problem of attempting to minimize the distance taken by a traveling salesman who has to visit a certain number of cities exactly once and return home. See Johnson and McGeoch (1997) or Lawler et al. (1990) for more details about the TSP. With a very small number of cities, the number of possible solutions is relatively small and a computer method can easily exhaustively check all possibilities (the search space) and return the best one. For example, the problem with five cities has a search space of size 12. So all 12 possibilities can

be very easily checked. However, for a 50-city problem (10 times the number of cities), the number of solutions rises to about 10^{60} . Michalewicz and Fogel, in their excellent book on modern heuristics, consider exactly this 50 city problem. They say,

There are only 1,000,000,000,000,000,000,000 [10²⁰] liters of water on the planet so a 50-city TSP has an unimaginably large search space. Literally, it's so large that as human, we simply can't conceive of sets with this many elements.

(Michalewicz and Fogel, 2004)

Therefore, for large problems (and large does not have to be that large), an exhaustive search is simply not an option. However, even if it is a possibility (i.e. the search space is small enough to allow us to carry out an exhaustive search) we must know how to systematically navigate the search space. This is not always possible.

Complexity This term refers to the study of how difficult search and optimization problems are to solve. It is covered in Chapter 11.

Order (Big O notation) This term and an associated notation is used at various places in this book and so we define it here. Suppose we have two functions $f(x)$ and $g(x)$ where x is, of course, a variable. We say that $g(x)$ is of the order of $f(x)$ written $g(x) = O(f(x))$ if, for some constant value K , $g(x) \leq Kf(x)$ for all values of x which are greater than K . This notation is often used when discussing the time complexity of search algorithms. In a certain sense, $f(x)$ *bounds* $g(x)$ once the values of x get beyond the value of K .

Heuristics When faced with the kind of problem discussed in the exhaustive search section above, we have to accept that we need to develop an approach to obtain high-quality solutions—but optimality cannot be guaranteed (without *checking out* all the possibilities). Such an approach is called a heuristic. The following two definitions provide good descriptions.

A heuristic technique (or simply heuristic) is a method which seeks good (i.e. near-optimal) solutions at a reasonable computation cost without being able to guarantee optimality, and possibly not feasibility. Unfortunately, it may not even be possible to state how close to optimality a particular heuristic solution is.

(Reeves, 1996)

A “rule of thumb” based on domain knowledge from a particular application, that gives guidance in the solution of a problem... Heuristics may thus be very valuable most of the time but their results or performance cannot be guaranteed.

(Oxford Dictionary of Computing, 1996)

There are many heuristic methods available to us. Some examples are simulated annealing (Chapter 7), genetic algorithms (Chapter 4), genetic programming (Chapter 5) and tabu search (Chapter 6). The term “approximate” is

sometimes used in connection with heuristic methods but it is important not to confuse with approximation methods (see Chapter 18).

Constructive heuristics Constructive heuristics refer to the process of building an initial solution from scratch. Take university examination timetabling as an example (Burke and Petrovic, 2002; Petrovic and Burke, 2004; Schaerf, 1999). One way to generate a solution is to start with an empty timetable and gradually schedule examinations until they are all timetabled. The order in which the examinations are placed onto the timetable is often important. Examinations which are more difficult to schedule (as determined by a heuristic measure of difficulty) are scheduled first in the hope that the *easier* examinations can *fit around* the difficult ones.

Constructive heuristics are usually thought of as being fast as they are often a single-pass approach.

Local search heuristics Local search can be thought of as a heuristic mechanism where we consider *neighbors* of the current solution as potential replacements. If we accept a new solution from this neighborhood, then we *move* to that solution and then consider its neighbors (see hill climbing (below) for some initial discussion of this point). What we mean by *neighbor* is dependent upon the problem solving situation that we are confronted with. Some of the techniques presented in this book can be described as local search methods. For example, see simulated annealing (Chapter 7) and tabu search (Chapter 6). Hill climbing is also a local search method (see below). For more information about local search see Aarts and Lenstra (1997). Note the difference between a constructive heuristic which builds a solution from scratch and a local search heuristic which moves from one solution to another. It is often the case that a constructive heuristic is used to generate a solution which is employed as the starting point for local search.

Hill climbing Hill climbing is probably the most basic local search algorithm. It is easy to understand and implement but suffers from getting stuck at a local optimum (see below). In the following discussion, we will assume we are trying to maximize a certain value. Of course, minimizing a certain value is just an analogous problem, but then we would be *descending* rather than *climbing*.

The idea behind hill climbing is to take the current solution and generate a neighbor solution (see local search) and move to that solution only if it has a higher value of the evaluation function (see above). The algorithm terminates when we cannot find a better-quality solution. The problem with hill climbing is that it can easily get stuck in a local optimum (see above). Consider Figure 1.1.

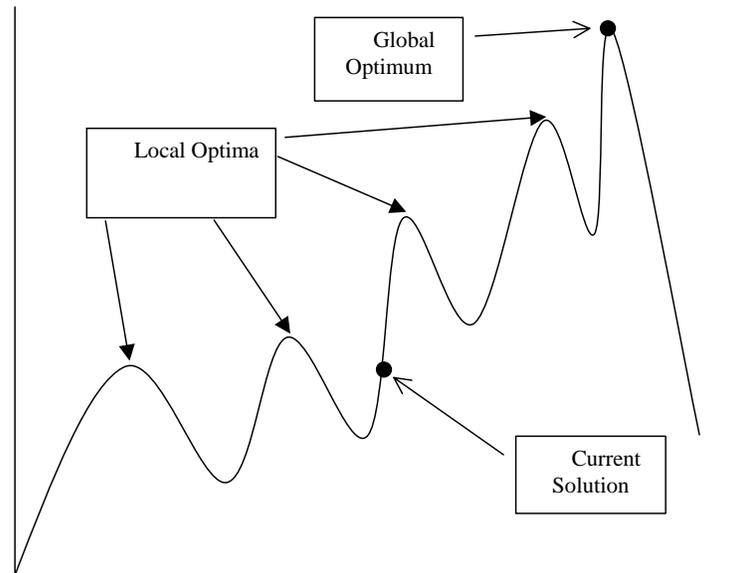


Figure 1.1. Hill Climbing getting stuck in a local optimum and the concept of local and global optima.

If the current solution is the one shown in Figure 1.1, then hill climbing will only be able to find one of the local optima shown (the one directly above it in this case). At that point, there will be no other better solutions in its neighborhood and the algorithm will terminate.

Both simulated annealing (Chapter 7) and tabu search (Chapter 6) are variations of hill climbing but they incorporate a mechanism to help the search escape from local optima.

Metaheuristics This term refers to a certain class of heuristic methods. Fred Glover first used it and he defines it (Glover, 1997) as follows:

A meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. The heuristics guided by such a meta-strategy may be high level procedures or may embody nothing more than a description of available moves for transforming one solution into another, together with an associated evaluation rule.

Osman and Kelly (1996) offer the following definition:

A meta-heuristic is an iterative generation process which guides a subordinate heuristic

The study and development of metaheuristics has become an extremely important area of research into search methodologies. In common usage, in the

literature, the term tends to be used to refer to the broad collection of relatively *sophisticated* heuristic methods that include simulated annealing, tabu search, genetic algorithms, ant colony methods and others (all of which are discussed in detail in this book). The term is employed sometimes with and sometimes without the hyphen in the literature. It is also sometimes interchanged with the term “modern heuristics” (see Rayward-Smith et al. (1996). For more information about metaheuristics, see Glover and Kochenberger (2003), Osman and Kelly (1996), Voss et al. (1999), Ribeiro and Hansen (2002) and Resende and de Sousa (2004).

Evolutionary methods Evolutionary methods can be thought of as representing a subset of the metaheuristic approaches and are typified by the fact that they maintain a *population* of candidate solutions and these solutions compete for survival. Such approaches are inspired by evolution in nature.

Some of the methods in this book are evolutionary. Chapter 4 (Genetic Algorithms) represents perhaps the best known evolutionary approach but there are many others including genetic programming (Chapter 5) and ant algorithms (Chapter 14).

Hyper-heuristics Hyper-heuristics can be confused with metaheuristics but the distinction between the two terms is quite clear. Hyper-heuristics are simply methods which search through a search space of heuristics (or search methods). They can be defined as *heuristics to choose heuristics*. Most implementations of metaheuristics explore a search space of solutions to a given problem but they can be (and sometimes are) employed as hyper-heuristics. The term hyper-heuristic only tells you that we are operating on a search space of heuristics. It tells you nothing else. We may be employing a metaheuristic to do this search and we may not. The actual search space being explored may include metaheuristics and it may not (but very little work has actually been done which includes metaheuristics among the search space being addressed). Chapter 17 describes hyper-heuristics in more detail and readers are also referred to Burke et al. (2003).

1.4 SOURCES OF ADDITIONAL INFORMATION

This section provides a list of journals (in alphabetical order) across a range of disciplines that regularly publish papers upon aspects of decision support methodologies. This list is certainly not exhaustive. However, it provides a starting point for the new researcher and that is the sole purpose of presenting it here. We have purposefully not provided URL links to the journals as many will change after going to press, but an internet search for the journal title will quickly locate the home page.

- ACM Journal of Experimental Algorithmics
- Annals Of Operations Research
- Applied Artificial Intelligence
- Applied Intelligence
- Artificial Intelligence
- Artificial Life
- Computational Intelligence
- Computer Journal
- Computers & Industrial Engineering
- Computers & Operations Research
- Decision Support Systems
- Engineering Optimization
- European Journal Of Information Systems
- European Journal Of Operational Research
- Evolutionary Computation
- Fuzzy Sets And Systems
- Genetic Programming and Evolvable Machines
- IEEE Transactions On Computers
- IEEE Transactions On Evolutionary Computation
- IEEE Transactions On Fuzzy Systems
- IEEE Transactions On Neural Networks
- IEEE Transactions On Systems Man And Cybernetics Part A—Systems And Humans
- IEEE Transactions On Systems Man And Cybernetics Part B—Cybernetics
- IEEE Transactions On Systems Man And Cybernetics Part C—Applications And Review
- IIE Transactions

- INFORMS Journal On Computing
- Interfaces
- International Journal Of Systems Science
- International Transactions On Operational Research
- Journal Of Artificial Intelligence Research
- Journal Of Global Optimization
- Journal Of Heuristics
- Journal Of Optimization Theory And Applications
- Journal of Scheduling
- Journal Of The ACM
- Journal Of The Operational Research Society
- Knowledge-Based Systems
- Machine Learning
- Management Science
- Neural Computation
- Neural Computing & Applications
- Neural Networks
- Neurocomputing
- Omega - International Journal of Management Science
- Operations Research Letters
- OR Spectrum
- SIAM Journal on Computing
- SIAM Journal On Optimization

The following list of references just includes those volumes and papers which give an overview of search and optimization methodologies and some well studied search/optimization problems. More detailed bibliographies and sources of additional information are given at the end of each chapter throughout the book.

References

- Aarts, E. and Lenstra, J. K. (eds), 1997, *Local Search in Combinatorial Optimization*, Wiley, New York.
- Bronson, R. and Naadimuthu, G, 1997, *Operations Research*, Schaum's Outlines, 2nd edn, McGraw-Hill, New York.
- Burke, E. K., De Causmaecker, P., Vanden Berghe, G. and Van Landeghem, R., 2004, The state of the art of nurse rostering, *J. Scheduling* 7:441–499.
- Burke, E. K., Kendall, G., Newall, J. P., Hart, E., Ross, P. and Schulenburg, S., 2003, Hyper-heuristics: An emerging direction in modern search technology, in: *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, eds, Chapter 16, pp. 457–474.
- Burke, E. K., Petrovic, S., 2002, Recent research directions in automated timetabling, *Eur. J. Oper. Res.* 140:266–280.
- Carter, M. W., and Price, C. C., 2001, *Operations Research: A Practical Introduction*, CRC Press, Boca Raton, FL.
- Callan R., 2003, *Artificial Intelligence*, Palgrave Macmillan, London.
- Cawsey A., 1998, *The Essence of Artificial Intelligence*, Prentice-Hall, Englewood Cliffs, NJ.
- Dell'Amico, M., Maffioli, F. and Martello, S. (eds), 1997, *Annotated Bibliographies in Combinatorial Optimization*, Wiley, New York.
- Dowsland, K. A. and Dowsland, W. B., 1992, Packing problems, *Eur. J. Oper. Res.* 56:2–14.
- Gass, S. I. and Harris, C. M., 2001, *Encyclopaedia of Operations Research and Management Science*, Kluwer, Dordrecht.
- Glover, F. and Kochenberger, G. (eds), 2003, *Handbook of Metaheuristics*, Kluwer, Dordrecht.
- Glover, F. and Laguna, M., 1997, *Tabu Search*, Kluwer, Dordrecht.
- Hillier F. S. and Liberman G. J., 2005, *Introduction to Operations Research*, McGraw-Hill, New York (8th edn).
- Johnson, D. S. and McGeoch, L. A., 1997, The travelling salesman problem: A case study, in: E. Aarts and J. K. Lenstra, eds, *Local Search in Combinatorial Optimization*, Wiley, New York, pp. 215–310.
- Kirby, M. W., 2003, *Operational Research in War and Peace: The British Experience from the 1930s to 1970*, Imperial College Press, London.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. and Shmoys, D. B. (eds), 1985, *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York (reprinted with subject index 1990).
- Leung, J. Y-T. (ed.), 2004, *Handbook of Scheduling*, Chapman and Hall/CRC Press, Boca Raton, FL.
- Luger G. F. A., 2002, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Addison-Wesley, Reading, MA, 4th edn.

- McCarthy, J., 1996, *Defending AI Research: A Collection of Essays and Reviews*, CSLI Publications, Stanford, CA.
- Michaelwicz, Z. and Fogel D. B., 2004, *How to Solve It: Modern Heuristics*, Springer, Berlin, 2nd edn.
- Negnevitsky M., 2005, *Artificial Intelligence: A Guide to Intelligent Systems*, Addison-Wesley, Reading, MA, 2nd edn.
- Nilsson, N., 1998, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann, San Mateo, CA.
- Oxford Dictionary of Computing, 1997, Oxford University Press, Oxford, 4th edn.
- Osman, I. H. and Kelly J. P. (eds), 1996, *Metaheuristics: Theory and Applications*, Kluwer, Dordrecht.
- Pardalos, P. M. and Resende, M. G. C. (eds), 2002, *Handbook of Applied Optimization*, Oxford University Press, Oxford.
- Petrovic, S. and Burke, E. K., 2004, University timetabling, Chapter 45 of J. Y-T. Leung, ed., *Handbook of Scheduling*, Chapman and Hall/CRC Press, Boca Raton, FL.
- Rayward-Smith V. J., Osman I. H., Reeves C. R. and Smith G. D., 1996, *Modern Heuristic Search Methods*, Wiley, New York.
- Reeves, C. R., 1996, Modern heuristic techniques, in: V. J. Rayward-Smith, I. H. Osman, C. R. Reeves and G. D. Smith, *Modern Heuristic Search Methods*, Wiley, New York, pp. 1–25.
- Resende, M. G. C. and de Sousa, J. P. (eds), 2004, *Metaheuristics: Computer Decision Making*, Kluwer, Dordrecht.
- Ribeiro, C. C. and Hansen, P. (eds), 2002, *Essays and Surveys in Metaheuristics*, Kluwer, Dordrecht.
- Rich E. and Knight K., 1991, *Artificial Intelligence*, McGraw-Hill, New York, 2nd edn.
- Russell, S. and Norvig, P., 2003, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ.
- Schaerf, A., 1999, A survey of automated timetabling, *Artif. Intell. Rev.* **13**:87–127.
- Taha H. A., 2002, *Operations Research: An Introduction*, Prentice-Hall, Englewood Cliffs, NJ, 7th edn.
- Urry S., 1991, *An Introduction to Operational Research: The Best of Everything*, Longmans, London.
- Voss, S, Martello, S., Osman, I. H. and Roucairol, C. (eds), 1999, *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer, Dordrecht.
- Winston, W. L., 2004, *Operations Research: Applications and Algorithms*, Duxbury Press, Philadelphia, PA, 4th edn.