Chapter 1
Introduction

Edmund K. Burke and Graham Kendall

1.1 Inter-disciplinary Decision Support: Motivation

Search and optimization technologies underpin 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 small selection of applications includes transport scheduling, bioinformatics optimization, personnel rostering, medical decision support and timetabling. Later in this introduction we present some recent survey papers for some of these areas and more examples of relevant applications are available in Pardalos and Resende (2002) and Leung (2004). The potential impact of more effective and 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;
- Efficient healthcare scheduling will lead to faster treatment (potentially saving 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. Looking at any selection of key papers

E.K. Burke (✉)
Computational Heuristics, Operational Research and Decision Support Group, Division of Computing and Mathematics, University of Stirling, Stirling, Scotland, UK
e-mail: e.k.burke@stir.ac.uk

G. Kendall
Automated Scheduling, Optimization and Planning Research Group, School of Computer Science, University of Nottingham, Jubilee Campus, Nottinghamshire, UK

Automated Scheduling, Optimization and Planning Research Group, School of Computer Science, University of Nottingham, Malaysia Campus, Jalan Broga, Semenyih, Malaysia

E.K. Burke and G. Kendall (eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, DOI 10.1007/978-1-4614-6940-7_1,

which have impacted upon search, optimization and decision support will reveal that the authors are based in a number of different departments including Computer Science, Mathematics, Engineering, Business and Management (among 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, Computer Science 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.

This observation provides one of the key motivations for this book, which 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 are in the literature. In our experience, the key texts for these methodologies lie across a variety of volumes. This reflects the wide 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

The first edition of this book was initially motivated by the idea of being able to present first-year PhD students with a single volume that would give them a basic introduction to the various search and optimization techniques that they might require during their program of research. This remains a primary motivation for this second edition.

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 with which you are not familiar, or if you 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. In Chaps. 2 and 3 we introduce 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. Indeed, recent (highly effective) approaches have hybridized such methods with some of the other techniques in this book. 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, Chap. 4) and some which are relatively new (e.g. artificial immune systems, Chap. 7). Some of the chapters consider more theoretical aspects of search and optimization. The chapter by Darrel Whitley, for example, introduces *Sharpened and Focused No Free Lunch and Complexity Theory* (Chap. 16) whilst Colin Reeves considers *Fitness Landscapes* in Chap. 22.

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. *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 asked authors to include a section entitled *Sources of Additional Information*. These sections are designed as useful pointers to resources such as books and web pages. They are intended as the next place to investigate, once you have read the chapter.

As this book is aimed primarily at the beginner (such as a first-year PhD student, final-year undergraduate or practitioner/researcher learning a new technique) we thought it might be useful to explain some basic concepts which many books just assume that the reader already knows. Indeed, our own PhD 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, purposefully, 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 later chapters of this book.

1.3 Basic Concepts and Underlying Issues

In this section we present a number of basic terms and issues and offer a simple description or explanation. In explaining the concepts to beginners, we will restrict the formal presentation of these concepts as much as possible.

1.3.1 Artificial Intelligence

Artificial intelligence (AI) 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 AI 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 (2009), Callan (2003), Luger (2005), McCarthy (1996), Cawsey (1998), Rich and Knight (1991), and Nilsson (1998).

1.3.2 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, 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 Liberman (2010), Taha (2010), Urry (1991), and Winston (2004). For an excellent and fascinating early history of the field see Kirby (2003).

1.3.3 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 in their preface when they say, "Together, OR and MS may be thought of as the science of operational processes, decision making and management."

1.3.4 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 can be related 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. 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 the smaller pieces can be cut out. 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 that it is not critical to include all the smaller pieces. A feasible solution is often defined as one that satisfies the *hard constraints* (see below).

1.3.5 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 an employee rostering problem, then an example of a hard constraint is the condition that no employee can be allocated to two different shifts at the same time. If we violate a hard constraint, it leads to an *infeasible* solution.

1.3.6 Soft Constraints and Evaluation Functions

A *soft constraint* is a condition that we would like to satisfy but which is not absolutely essential. To elaborate on the employee scheduling example, we may have a soft constraint that says that we would like employees 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 more (i.e. more employees have their 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 the situation where a multi-objective approach might be applicable (see Chap. 15).

Many problems have an evaluation function represented by a summation of the penalty values for the soft 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 penalty value. Of course, the goal here would be to *minimize* the overall penalty value but it is often the case that some search problems try to *maximize* an evaluation function. A further possibility is to have dynamic penalties so that, at the start of a (minimization) 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.

1.3.7 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 Chap. 16 for an explanation.

1.3.8 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 describe methodologies which aim to optimize some function. However, most of the problems are so large that it is impossible to guarantee that the solution obtained is optimal. 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).

1.3.9 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 Fig. 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. Of course, in Fig. 1.1, we are considering a *maximization* problem.

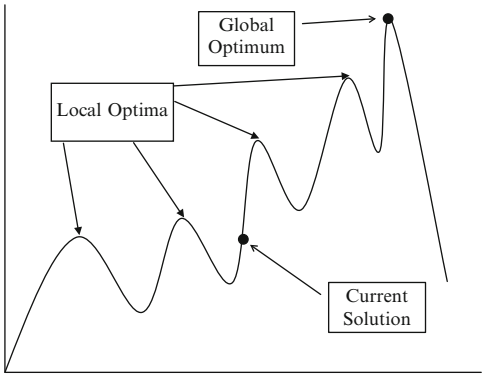

Fig. 1.1 An illustration of local optima and a global optimum

1.3.10 Exhaustive Search

By carrying out an exhaustive search, every possible solution is considered and the optimal (best) one is returned. For small problems, this is an acceptable strategy, but as problems become larger it becomes impossible to carry out such a search. The types of problem that often occur in the real world tend to grow very large very quickly and, of course, we can invent problems which are also too large to allow us to carry out an exhaustive search in a reasonable time. We will illustrate how problem sizes rise dramatically 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), Lawler et al. (1985) or Laporte (2010) for more details about the TSP. With a very small number of cities, the number of possible solutions is relatively small and an algorithm can easily check all possibilities (the search space) and return the best one. For example, a 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 (just 10 times the number of cities), the number of solutions rises to about 10^{60}. Michaelwicz and Fogel (2004), 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 [10^{20}] liters of water on the planet so a 50-city TSP has an unimaginably large search space. Literally, it's so large that as humans, we simply can't conceive of sets with this many elements."

Therefore, for large problems (and large does not have to be that large when considering the inputs), 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.

1.3.11 Complexity

This term refers to the study of how difficult search and optimization problems are to solve. It is covered in Chap. 16.

1.3.12 Order (Big O Notation)

This term and associated notation is used in various places in this book and so we define it here. Suppose we have two functions $f(x)$ and $g(x)$ where x is a variable. We say that $g(x)$ is of the order of $f(x)$ written $g(x) = \mathrm{O}(f(x))$ 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.

1.3.13 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 are often useful:

> 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 (Chap. 10), Genetic Algorithms (Chap. 4), Genetic Programming (Chap. 6) and Tabu Search (Chap. 9). The term *approximate* is sometimes used in connection with heuristic methods but it is important not to confuse it with approximation methods (see Chap. 21)

1.3.14 Constructive Heuristics

Constructive heuristics refer to the process of building an initial solution from scratch. Take university examination timetabling as an example. 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 often scheduled first in the hope that the *easier* examinations can *fit around* the difficult ones.

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

1.3.15 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 is meant by *neighbor* is dependent upon the problem-solving situation that is being addressed. Some of the techniques described in this book can be described as local search methods: for example, simulated annealing (Chap. 10) and tabu search (Chap. 9). Hill climbing is also a local search method. For more information about local search see Aarts and Lenstra (2003). 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.

1.3.16 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. In the following discussion, we will assume that we are trying to maximize a certain value. Of course, minimizing a certain value is 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. 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 Fig. 1.1.

If the current solution is the one as shown in Fig. 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 (Chap. 10) and tabu search (Chap. 9) are variations of hill climbing but they incorporate a mechanism to help the search escape from local optima.

1.3.17 Metaheuristics

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

> 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. For more information about metaheuristics, see Glover and Kochenberger (2003), Osman and Kelly (1996), Voss et al. (1999), Ribeiro and Hansen (2002), Resende and de Sousa (2004), Gendreau and Potvin (2010), and Rayward-Smith et al. (1996).

1.3.18 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 that 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 (Chap. 6). In the scientific literature, many metaheuristics are hybridized with other approaches (see, for example, Blum et al. 2011).

1.3.19 Exact Methods

This term is sometimes used to describe methods which can produce a solution that is guaranteed to be optimal (or which can show that no feasible solution exists).

1.3.20 Hyper-heuristics

Hyper-heuristics can be confused with metaheuristics but the distinction between the two terms is simple. 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* or *heuristics which generate 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 21describes hyper-heuristics in more detail and readers are also referred to Burke et al. (2003, 2010, 2013).

1.3.21 Matheuristics

This term refers to methods that hybridize metaheuristics with mathematical programming techniques. See Maniezzo et al. (2010) and Jourdan et al. (2009) for more details.

Sources of Additional Information

The goal of this book is to present clear basic introductions to a wide variety of search methodologies from across disciplinary boundaries. However, it will be the case that many readers will be interested in a specific application or problem domain. With this in mind, we provide a sample of domains, with a small selection of references to survey and overview papers which might be of interest. Of course, the list is far from exhaustive but its purpose is to point the interested reader to overview papers for a small selection of well studied problem areas.

- Cutting and Packing: Bennell and Oliveira (2008, 2009), Dyckhoff (1990), Dowsland and Dowsland (1992), and Wäscher et al. (2007).
- Employee Scheduling: Burke et al. (2004), Ernst et al. (2004), Gopalakrishnan and Johnson (2005), and Kwan (2004).
- Educational Timetabling: Burke and Petrovic (2002), Lewis (2008), Qu et al. (2009), Petrovic and Burke (2004), and Schaerf (1999).
- Healthcare Scheduling and Optimization: Cardoen et al. (2010) and Rais and Viana (2011).
- Printed Circuit Board Assembly: Ayob and Kendall (2008, 2009).
- Air Transport Scheduling: Qi et al. (2004) and Gopalakrishnan and Johnson (2005).

- Sports Scheduling: Dinitz et al. (2007), Drexl and Knust (2007), Easton et al. (2004), Kendall et al. (2010), Rasmussen and Trick (2008), and Wright (2009).
- Traveling Salesman Problem: Laporte 2010, Johnson and McGeoch (1997), and Lawler et al. (1985).
- Vehicle Routing: Laporte (2009), Potvin (2009), Marinakis and Migdalas (2007), and Bräysy and Gendreau (2005a,b).

In this section we will also provide 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 a search for a journal's title will quickly locate its home page.

- ACM Journal of Experimental Algorithmics
- Annals of Operations Research
- Applied Artificial Intelligence
- Applied Intelligence
- Applied Soft Computing
- Artificial Intelligence
- Artificial Life
- Asia-Pacific Journal of Operational Research
- Central European Journal of Operations Research
- Computational Intelligence
- Computational Optimization and Applications
- 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
- 4OR—A Quarterly Journal of Operations Research
- 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
- INFOR
- 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
- Journal of The Operational Research Society of Japan
- Knowledge-Based Systems
- Machine Learning
- Management Science
- Mathematical Methods of Operations Research
- Mathematics of Operations Research
- Mathematical Programming
- Naval Research Logistics
- Neural Computation
- Neural Computing & Applications
- Neural Networks
- Neurocomputing
- Omega—International Journal of Management Science
- Operations Research
- Operations Research Letters
- OR Spectrum
- RAIRO—Operations Research
- SIAM Journal on Computing
- SIAM Journal on Optimization
- Soft Computing
- Transportation Research
- Transportation Science

This bibliography presents a selection of 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 presented throughout the book.

References

Aarts E, Lenstra JK (eds) (2003) Local search in combinatorial optimization. Princeton University Press, Princeton, New Jersey, USA (first published by Wiley 1997)

Ayob M, Kendall G (2008) A survey of surface mount device placement machine optimisation: machine classification. Eur J Oper Res 186:893–914

Ayob M, Kendall G (2009) The optimisation of the single surface mount device placement machine in printed circuit board assembly: a survey. Int J Syst Sci 40:553–569

Bennell JA, Oliveira JF (2008) A tutorial in nesting problem: the geometry. Eur J Oper Res 184:397–415

Bennell JA, Oliveira JF (2009) A tutorial in irregular shape packing problems. J Oper Res Soc 60:S93–S105

Blum C, Puchinger J, Raidl G, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. Appl Soft Comput 11:4135–4151

Bräysy O, Gendreau M (2005a) Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transp Sci 39:104–118

Bräysy O, Gendreau M (2005b) Vehicle routing problem with time windows, part II: metaheuristics. Transp Sci 39:119–139

Bronson R, Naadimuthu G (1997) Operations research, Schaum's outlines, 2nd edn. McGraw-Hill, New York

Burke EK, Petrovic S (2002) Recent research directions in automated timetabling. Eur J Oper Res 140:266–280

Burke EK, Kendall G, Newall JP, Hart E, Ross P, Schulenburg S (2003) Hyper-heuristics: an emerging direction in modern search technology. In: Glover F, Kochenberger G (eds) Handbook of metaheuristics, chap 16. Kluwer, Dordrecht, pp 457–474

Burke EK, De Causmaecker P, Vanden Berghe G, Van Landeghem R (2004) The state of the art of nurse rostering. J Sched 7:441–499

Burke EK, Hyde M, Kendall G, Ochoa G, Ozcan E, Woodward JRA (2010) A Classification of hyper-heuristic approaches. In: Handbook of metaheuristics. Kluwer, Dordrecht, pp 449–468

Burke EK, Gendreau M, Hyde M, Kerdall G, Ochoa G, Ozcan E, QUR (2013) Hyper-heuristics: a survey of the state of the art. J Oper Res Soc, doi:10.1057/jors.2013.71

Callan R (2003) Artificial intelligence. Palgrave Macmillan, London

Cardoen B, Demeulemeester E, Beliën J (2010) Operating room planning and scheduling: a literature review. Eur J Oper Res 201:921–932

Carter MW, Price CC (2001) Operations research: a practical introduction. CRC, Boca Raton

Cawsey A (1998) The essence of artificial intelligence. Prentice-Hall, Englewood Cliffs

Dinitz JH, Fronček D, Lamken ER, Wallis WD (2007) Scheduling a tournament. In: Colbourn CJ, Dinitz JH (eds) Handbook of combinatorial designs, 2nd edn. CRC, Boca Raton, pp 591–606

Dowsland KA, Dowsland WB (1992) Packing problems. Eur J Oper Res 56:2–14

Drexl A, Knust S (2007) Sports league scheduling: graph- and resource-based models. Omega 35:465–471

Dyckhoff H (1990) A typology of cutting and packing problems. Eur J Oper Res 44:145–159

Easton K, Nemhauser GL, Trick MA (2004) Sports scheduling. In: Leung JT (ed) Handbook of scheduling. CRC, Boca Raton, 52.1–52.19

Ernst AT, Jiang H, Krishnamoorthy M, Owens B, Sier D (2004) An annotated bibliography of personnel scheduling and rostering. Ann Oper Res 127:21–144

Gass SI, Harris CM (2001) Encyclopaedia of operations research and management science. Kluwer, Dordrecht

Gendreau M, Potvin J-Y (eds) (2010) Handbook of metaheuristics, 2nd edn. Springer, Berlin

Glover F, Kochenberger G (eds) (2003) Handbook of metaheuristics. Kluwer, Dordrecht

Glover F, Laguna M (1997) Tabu search. Kluwer, Dordrecht

Gopalakrishnan B, Johnson EL (2005) Airline crew scheduling: state-of-the-art. Ann Oper Res 140:305–337

Hillier FS, Liberman GJ (2010) Introduction to operations research, 9th edn. McGraw-Hill, New York

Johnson DS, McGeoch LA (1997) The travelling salesman problem: a case study. In: Aarts E, Lenstra JK (eds) (2003) Local search in combinatorial optimization. Princeton University Press, Princeton, New Jersey, USA, pp 215–310

Jourdan L, Basseur M, Talbi E-G (2009) Hybridizing exact methods and metaheuristics: a taxonomy. Eur J Oper Res 199:620–629

Kendall G, Knust S, Ribeiro CC, Urrutia S (2010) Scheduling in sports: an annotated bibliography. Comput Oper Res 37:1–19

Kirby MW (2003) Operational research in war and peace: the British experience from the 1930s to 1970. Imperial College Press, London

Kwan R (2004) Bus and train driver scheduling. In: Leung JY-T (ed) Handbook of scheduling, chap 51. Chapman and Hall/CRC, Boca Raton

Laporte G (2009) Fifty years of vehicle routing. Transp Sci 43:408–416

Laporte G (2010) A concise guide to the traveling salesman problem. J Oper Res Soc 61:35–40

Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB (eds) (1985) The travelling salesman problem: a guided tour of combinatorial optimization. Wiley, New York (reprinted with subject index 1990)

Leung JY-T (ed) (2004) Handbook of scheduling. Chapman and Hall/CRC, Boca Raton

Lewis R (2008) A survey of metaheuristic-based techniques for university timetabling problems. OR Spectr 30:167–190

Luger GFA (2005) Artificial intelligence: structures and strategies for complex problem solving, 5th edn. Addison-Wesley, New York
Maniezzo V, Stützle T, Voss S (eds) (2010) Matheuristics. Springer, Berlin
Marinakis Y, Migdalas A (2007) Annotated bibliography in vehicle routing. Oper Res 7:27–46
McCarthy J (1996) Defending AI research: a collection of essays and reviews. CSLI Publications, Stanford
Michaelwicz Z, Fogel DB (2004) How to solve it: modern heuristics, 2nd edn. Springer, Berlin
Negnevitsky M (2005) Artificial intelligence: a Guide to intelligent systems, 2nd edn. Addison-Wesley, New York
Nilsson, N (1998) Artificial intelligence: a new synthesis. Morgan Kaufmann, San Mateo
Osman IH, Kelly JP (eds) (1996) Metaheuristics: theory and applications. Kluwer, Dordrecht
Oxford Dictionary of Computing (1996) Oxford dictionary of computing, 4th edn. Oxford University Press, Oxford
Pardalos PM, Resende MGC (eds) (2002) Handbook of applied optimization. Oxford University Press, Oxford
Petrovic S, Burke EK (2004) University timetabling. In: Leung JY-T (ed) (2004) Handbook of scheduling, chap 45. Chapman and Hall/CRC, Boca Raton
Potvin J-Y (2009) Evolutionary algorithms for vehicle routing. INFORMS J Comput 21:518–548
Qi X, Yang J, Yu G (2004) Scheduling problems in the airline industry. In: Leung JY-T (ed) Handbook of scheduling, chap 51. Chapman and Hall/CRC, Boca Raton
Qu R, Burke EK, McCollum B, Merlot LGT, Lee SY (2009) A survey of search methodologies and automated system development for examination timetabling. J Sched 12:55–89
Rais A, Viana A (2011) Operations research in healthcare: a survey. Int Trans Oper Res 18:1–31
Rasmussen RV, Trick MA (2008) Round robin scheduling—a survey. Eur J Oper Res 188:617–636
Rayward-Smith VJ, Osman IH, Reeves CR, Smith GD (1996) Modern heuristic search methods. Wiley, New York
Reeves CR (1996) Modern heuristic techniques. In: Rayward-Smith VJ, Osman IH, Reeves CR, Smith GD (eds) Modern heuristic search methods. Wiley, New York, pp 1–25
Resende MGC, de Sousa JP (eds) (2004) Metaheuristics: computer decision making. Kluwer, Dordrecht
Ribeiro CC, Hansen P (eds) (2002) Essays and surveys in metaheuristics. Kluwer, Dordrecht
Rich E, Knight K (1991) Artificial intelligence, 2nd edn. McGraw-Hill, New York
Russell S, Norvig P (2009) Artificial intelligence: a modern approach, 3rd edn. Prentice-Hall, Englewood Cliffs
Schaerf A (1999) A survey of automated timetabling. Artif Intell Rev 13:87–127

Taha HA (2010) Operations research: an introduction, 9th edn. Prentice-Hall, Englewood Cliffs
Urry S (1991) An introduction to operational research: the best of everything. Longmans, London
Voss S, Martello S, Osman IH, Roucairol C (eds) (1999) Meta-heuristics: advances and trends in local search paradigms for optimization. Kluwer, Dordrecht
Wäscher G, Hauβner H, Schumann H (2007) An improved typology of cutting and packing problems. Eur J Oper Res 183:1109–1130
Winston WL (2004) Operations research: applications and algorithms, 4th edn. Duxbury, Pacific Grove
Wright MB (2009) Fifty years of OR in sport. J Oper Res Soc 60:S161–S168