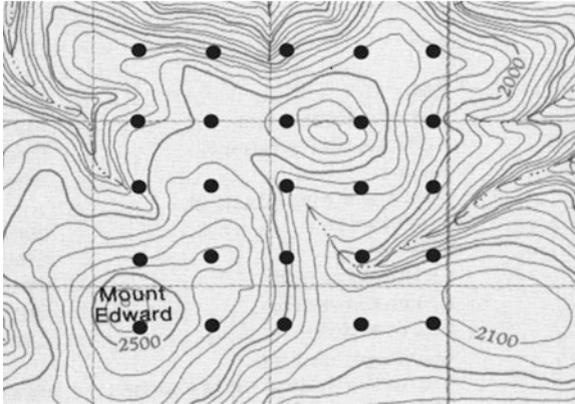


---

## Appendix A: Heuristic Algorithms

In this book, you will hear about or even directly encounter a number of solution algorithms. All of these algorithms fall into two broad categories: *exact algorithms* (sometimes also somewhat misleadingly referred to as *optimal algorithms*) and *heuristic methods* usually simply called *heuristics*. Exact algorithms have the obvious advantage of providing the best possible solution there is, given the user-defined constraints, whereas heuristics do not. Some heuristics do have error bounds, some actually proven, while others are empirical, i.e., they state that a certain heuristic usually (typically on average) finds solutions that have a certain quality. On the other hand, there is computing speed. Some models are such that it takes an exact algorithm exceedingly long to find the optimal solution. Is this relevant? Well, it depends. If the task at hand is to, say, locate a landfill for millions of dollars, you will not care if it takes a laptop 2 or 3 weeks to run, so that it can find a solution that may potentially save hundreds of thousands of dollars. There are limits to this argument, of course: if it takes years or even longer to find a solution, most problems have either solved themselves or have become irrelevant by that time. So, this is not acceptable.

In order to make the case for heuristics, consider the situation of automated guided vehicles (AGVs). Suppose you have a number of individual work stations on the shop floor, each of which processes a given piece from a semi-finished product to the finished good. For that purpose, it will be necessary to move pieces, on which work has been finished at one station to the next station. Note that it is not necessarily the case that all goods are running through the same sequence of jobs, not all tasks have to be performed at all work stations, and different levels of customization are possible. The movement of goods may be accomplished by automated guided vehicles that receive a message from a workstation whenever a piece is ready for pickup, and the machine knows where the piece has to go next. However, temporarily there may be more pieces to be transported than the machine can handle, so that it will put the tasks in a list. Whenever it is ready it will work on that list, depending on where it is at any point in time, how far it is to the destination for that particular transportation job, how many work stations will be idle if they have to wait for the next job, and many other considerations. It is apparent that solutions to that problem have to be found in real time, i.e., immediately. This is where heuristic algorithms come in.



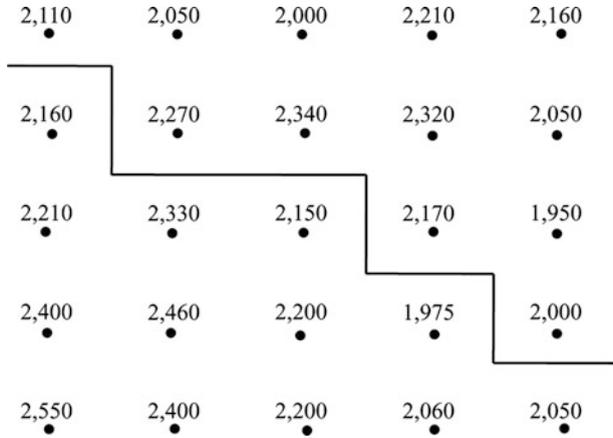
**Fig. A.1** Map for the lost hiker example

Heuristic algorithms typically have two phases. The first phase is the *construction phase*, in which a solution is established. This phase starts with nothing, and by the end of the phase, we have a solution. Phase 1 should be followed by Phase 2, which is an *improvement phase*, in which the method uses simple modifications of the present solution that improve the quality of the solution. The best known heuristics are the *Greedy Method* (a construction method), and the *Swap* (or *pairwise exchange*) method, which is an improvement method.

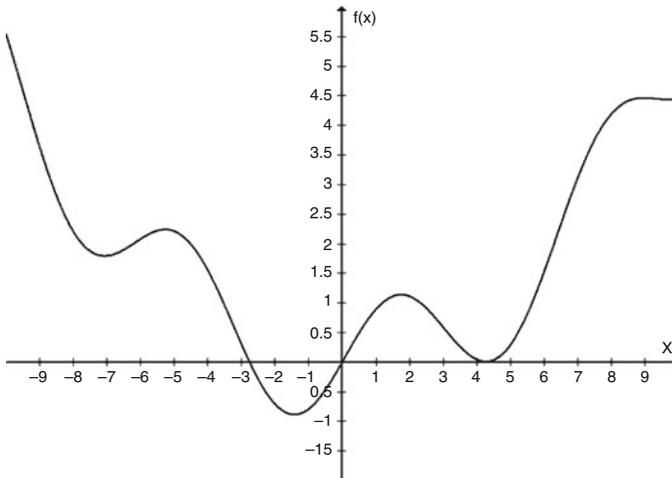
In order to illustrate the Greedy technique, consider the following example. Suppose that a hiker is lost in the woods. In order to be visible from the air for a rescue mission, he decides to climb as high as possible. Since he has neither a map, nor an idea of where he really is (and it is very foggy, so that visibility is very limited, making his vision myopic), he can only determine the shape of the land in close vicinity. At any point, he can examine the terrain to his North, Northwest, West, Southwest, South, Southeast, East, and Northeast. If there are higher points in any of these directions, he will go into that direction that features the highest nearby point (i.e., the best possible improvement, hence the name “Greedy”). If the points in all eight directions are lower than where he is right now, he will conclude that he is standing on top of a hill and stay there, assuming that he has arrived.

As an example, consider the situation shown in Fig. A.1. In order to simplify matters, we have determined the altitudes of the points shown as big black dots and have them displayed again in Fig. A.2.

Suppose now that the hiker is presently at the Northeasterly point in Fig. A.2 at an altitude of 2,160 ft. There are only three surrounding points, those to the West, the Southwest, and the South, with altitudes of 2,210, 2,230, and 2,050, respectively. The highest of these is the point to the Southwest, which is then where the hiker walks to. This point now has eight neighbors, the highest of which is located directly to the West of the hiker’s present position at an altitude of 2,340. Consequently, the hiker relocates to this point. Repeating the procedure, the hiker notices that all points in the vicinity of his present location are lower. So, he concludes that he is standing



**Fig. A.2** Altitudes (in feet) for the lost hiker example



**Fig. A.3** Function with three local minima

on top of a hill. From the topographic map we know that he is, but we also know that this is not the highest hill in the area of interest.

Formally, a point all of whose neighbors are lower (higher), is called a *local maximum (local minimum)*. If a local maximum (local minimum) is also the highest (lowest) point overall, it is referred to as a *global maximum (global minimum)*. As an illustration, consider the function  $y = \sin x + 0.05x^2$ . For values of  $x$  between  $-10$  and  $+10$ , the function is shown in Fig. A.3.

The function has three minima on the domain shown, they are at  $x = -7.0689$  (with  $y = 1.7911$ ), at  $x = -1.4755$  (with  $y = -.8879$ ), and at  $x = 4.2711$  (with  $y = .0079$ ). (Myopic) heuristics (as well as derivatives) will readily find

local optima, but not necessarily global optima. The reason for this is apparent: if any myopic method arrives at, say, the rightmost of the three local minima in Fig. A.3, how is the method to know whether or not there are better points located to the left? If a graph such as this were available, it is easy to see, but typically it is not, as almost all problems are multidimensional, and thus cannot be graphed in three or fewer dimensions.

The hiker's plight described above might now be considered his own problem and of little general interest, if it were not for the fact that each point on the map may represent a course of action (determined by the values of its coordinates), and the contour lines may represent their profit. So the search for the highest hill has now become a search for the point of maximal profit, which is of considerable general interest.

To make the problem even more interesting, had the hiker been in the extreme Southeast and had used the Greedy method for his progress, he would actually have ended up at Mount Edward, the overall highest point in the region of interest. As a matter of fact, the line that divides Fig. A.2 indicates what is usually referred to as *catchment areas*: if the hiker starts at any point to the Northeast of the line, he ends up at the local optimum at altitude 2,340 ft, if he starts at any point to the Southwest of the line, he ends up on Mount Edward at the global maximum. A similar situation applies to the function in Fig. A.3. Starting a Greedy minimization method any point to the left of  $x = -5.2671$  (a local maximum), will end up at the local minimum at  $x = -7.0689$ . Starting at any point between  $x = -5.2671$  and  $x = 1.7463$  will end at the global minimum at  $x = -1.4276$ , and starting anywhere to the right of  $x = 1.7463$  will lead to the local minimum at  $x = 4.2711$ .

This immediately suggests a technique that is called a *multistart method*. The idea is simply to apply a Greedy technique starting at a number of different points, compare the results, and choose the best (highest or lowest, depending on whether a minimum or a maximum is sought).

Next we will discuss an improvement method. The Swap method is easy to describe. Given a solution that has been obtained "somehow," it takes two components and exchanges them. Depending on the specific application, this may mean exchange their sequence, their inclusion/exclusion status, or whatever the problem commands. The method then computes the change of the value of the objective function. If the value has improved (i.e., has increased in case of a maximization problem or decreased in case of a minimization problem), the modified solution becomes the new starting point and the previous solution is discarded. This step is repeated until no further Swap step can improve the solution any further.

As an example, consider the map in Fig. A.4 and assume that the task at hand is to determine the shortest route from Memphis, Tennessee, to Reno, Nevada. The traveler has outlined the tour in layers, so that a drive from a city in one layer to a city in the next is about a day's drive.

The distances are as follows: From Memphis to Omaha, Wichita, Oklahoma City, and Dallas we have 650, 555, 460, and 460 miles, respectively. From Omaha to Denver and Albuquerque there are 540 and 860 miles. The distances from



**Fig. A.4** Map of the U.S. for the shortest route example

Wichita to Denver and Albuquerque are 510 and 585 miles, from Oklahoma City to Denver and Albuquerque are 630 and 550 miles, and from Dallas to Denver and Albuquerque are 780 and 640 miles, respectively. The distances from Denver to Salt Lake City and Phoenix are 505 and 835 miles, while from Albuquerque, there are 610 and 460 miles to Salt Lake City and Phoenix, respectively. Finally, the distance from Salt Lake City to Reno is 525 miles and from Phoenix to Reno there are 735 miles.

In order to obtain some solution, we use the Greedy algorithm, starting at the origin of our trip in Memphis. From here, Greedy will choose the nearest neighbor, which is either Dallas or Oklahoma City, both 460 miles away. We arbitrarily choose Dallas. The nearest neighbor of Dallas is Albuquerque, which is 640 miles away (as opposed to Denver, which is 780 miles from Dallas). From Albuquerque, we take the closest connection to Phoenix (460 miles), and from there we have no choice but take the last long trip to Reno (735 miles). The trip leads us on the route Memphis – Dallas – Albuquerque – Phoenix – Reno, and its total length is 2,295 miles.

At this point we start to swap. One possibility is to swap Phoenix and Salt Lake City. This means that we have to add the connections from Albuquerque to Salt Lake City (610 miles) and from Salt Lake City to Reno (525 miles), and subtract the connections from Albuquerque to Phoenix (460 miles) and from Phoenix to Reno (735 miles), for net savings of 60 miles. This is an improvement, and so our new route is Memphis – Dallas – Albuquerque – Salt Lake City – Reno, and its total length is 2,235 miles.

We can now use the new solution and try other Swap moves. For instance, we could attempt to swap Albuquerque and Denver. The net change of such a swap move is +35 miles, so we will not make this change. Another possibility is to swap Dallas and Oklahoma City. The net change is  $-90$  miles, so we make the change and obtain the route Memphis – Oklahoma City – Albuquerque – Salt Lake City – Reno, whose length is 2,145 miles.

At this time we may examine again the pair Albuquerque and Denver. This swap did not improve the solution the last time we tried it, but since then the solution has changed. In fact, swapping the two cities at this point results in a net change of  $+630 + 505 - 550 - 610 = -25$ , so that the change is made. This results in the new route Memphis – Oklahoma City – Denver – Salt Lake City – Reno, which is 2,120 miles long.

We may now try to exchange Oklahoma City and Wichita, which leads to a net change of  $+555 + 510 - 460 - 630 = -25$ , for another reduction in terms of the total distance, which is now 2,095 miles. The route leads from Memphis – Oklahoma City – Denver – Salt Lake City – Reno. At this point we may try to further reduce the length of the tour, which is no longer possible with swap moves. As a matter of fact (unbeknownst to us when we are just using heuristics), the tour is actually optimal.

Our final example of a heuristic method deals with a much-studied field called *bin packing*. We have an unspecified number of bins, all of which are of the same prespecified length. We also have a number of rods that are to be placed into the bins. The problem is one-dimensional, in that the bins and the rods have the same height and width, so that only the length of the bins and the rods that are placed into them will decide whether or not they actually fit. For instance, if the bin is 20 ft and there are one 6 ft rod, one 3 ft rod, and one 5 ft rod, then these three rods will occupy 14 ft of the bin and leave 6 ft unoccupied. The task at hand is now to put the existing rods into the smallest number of bins possible.

Despite its apparent simplicity, the problem has been proven to be very difficult from a computational point of view. A Greedy-like heuristic is the so-called *First Fit (FF) Algorithm*. In order to implement the method, we first assume that a sufficiently large number of bins is available. These bins are numbered 1, 2, ... The First Fit Algorithm that can be described as follows:

*FF Algorithm:* Put the next rod into the bin with the smallest number into which it will fit.

As an example, suppose that all bins are 19 ft long. In addition, we have six 11 ft rods, six 6 ft rods, and twelve 4 ft rods. In this type of situation, we can actually compute a very simple bound for the number of bins that will be needed. Here, the total length of the rods is  $6(11) + 6(6) + 12(4) = 150$  ft. Given that each bin is 19 ft long, we will need at least  $150/19 \cong 7.89$  bins. Since the number of bins must be integer, we will need at least eight bins. This also means that if we were to find a solution to the problem that requires eight bins, this solution must be optimal.

Apply now the First Fit Algorithm. Assigning the rods in order of their lengths (i.e., the 11 ft rods first, then the 6 ft rods, and finally the 4 ft rods), we notice that

only a single 11 ft rod fits into each bin. This means that we have to put each of the 11 ft rods into one bin each, so that we now have dealt with all 11 ft rods and have used parts of six bins. Next, we assign the six 6 ft rods. Since each of them fits into one of the already partially used bins, we now have six bins with one 11 ft and one 6 ft rod each, leaving 2 ft of free space in each of the six bins. This is not sufficient for any of the remaining 4 ft bins, so that we have to use additional bins. We can place four 4 ft rods in each bin, leaving an empty (and unusable for us) space of 3 ft each, which requires another three bins. We now have assigned all rods to the bins. This solution requires a total of nine bins. There is no apparent pairwise exchange (swap) step able to improve the solution.

On the other hand, if we were to put one 11 ft and two 4 ft rods into each of six bins, this would leave no empty space at all. The remaining six 6 ft rods can be put into two bins. Having again assigned all rods, this solution requires only eight bins and, given the bound computed above, must be optimal.

A variety of other heuristics exists for this problem. An excellent (albeit difficult) pertinent reference is the book by Garey and Johnson (1979). An interesting extension of the problem makes available the different rods over time. This is reminiscent of ready times in machine scheduling. The solution obtained in such a case will be no better than the one found in the case in which all rods are available at the beginning of the process (simply because the problem that makes rods only available over time is more restrictive than the problem discussed here).

The efficiency of the heuristics and their performance may be evaluated in different ways. An obvious evaluation is to use simulation (see Chap. 13 of this book). This will enable users to specify an average or expected *error bound* of the algorithm. For instance, in the above example the heuristic method uses nine instead of the optimal eight bins, i.e., 12.5 % more than optimal.

However, in some cases it is possible to determine theoretical error bounds, i.e., bounds that cannot be violated. For instance, it has been shown that the solution found by the First Fit Heuristic cannot be worse than about 70 % higher than the optimal solution. One problem associated with the theoretical bounds is that while they represent a reliable, provable property, they tend to be very high.

Many other heuristics have been presented in the literature. Many improvement algorithms are *neighborhood searches*, whose main distinguishing feature is that they start with a given solution and search for better solutions in the neighborhood of the present solution. The Swap Method described above belongs to this class. Another very successful heuristic in this class is *tabu search*. The idea of this method is to get out of a local optimum by temporarily allowing the current solution to deteriorate. In order to avoid cycling between solutions that are better and those that are worse, a list of prohibited moves (a tabu list) is set up that is updated as the algorithm progresses. This procedure allows to “get over the hump” from the present point to other solutions that are hopefully better than the best solution known at this point. For example, in Fig. A.3, if the best known minimum is  $x = -7.0689$ , we may allow worse solutions (i.e., those with higher functional values) in our move to the left. This may allow us to find the global minimum at  $x = -1.4755$ .

Other techniques are based on observations made in the technical or the natural world. Examples are *simulated annealing*, a technique modeled after the way molten metal cools. Similar to tabu search, it allows moves to solutions worse than the best presently known solution. Such moves are allowed with a certain probability that decreases during the course of the algorithm. The formulas ensure that the probability to accept a move to a very bad solution is very small. Other methods follow some behavioral patterns of ant colonies or bees.

---

## Appendix B: Vectors and Matrices

This appendix is intended to provide the reader with some basic refresher regarding some basic operations that involve matrices and vectors and the solution of systems of simultaneous linear equations. It is not designed to replace a text, but as a mere quick reference for some material used in this book.

**Definition B1.** An  $[m \times n]$ -dimensional matrix

$$\mathbf{A} = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

is a two-dimensional array of elements  $a_{ij}$  arranged in  $m$  horizontal rows and  $n$  vertical columns, so that the element  $a_{ij}$  is positioned in row  $i$  and column  $j$ . If  $m = n$ , the matrix is said to be *square*, if  $m = 1$ , it is called a *row vector*, if  $n = 1$ , it is a *column vector*, and if  $m = n = 1$ , it is a *scalar*.

It is common practice to denote scalars by italicized letters, vectors by lower-case bold letters, and matrices by capitalized letters in boldface.

**Definition B2.** Given an  $[m \times n]$ -dimensional matrix  $\mathbf{A}$  and an  $[n \times p]$ -dimensional matrix  $\mathbf{B}$ , the product  $\mathbf{C} = \mathbf{AB}$  is an  $[m \times p]$ -dimensional matrix  $\mathbf{C} = (c_{ij})$ , such that

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}, \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n.$$

---

### Example B1

With  $\mathbf{a} = [3, \sqrt{2}, -6]$  and

$$\mathbf{B} = \begin{bmatrix} 0 & 5 & \pi \\ 7 & -6 & 2 \\ \sqrt{3} & -11 & 1 \end{bmatrix},$$

we obtain  $\mathbf{aB} = [7\sqrt{2} - 6\sqrt{3}, 81 - 6\sqrt{2}, 3\pi + 2\sqrt{2} - 6]$ , and

$$\mathbf{B}^2 = \mathbf{BB} = \begin{bmatrix} 35 + \pi\sqrt{3} & -11\pi - 30 & 10 + \pi \\ -42 + 2\sqrt{3} & 49 & -10 + 7\pi \\ -77 + \sqrt{3} & 55 + 5\sqrt{3} & -21 + \pi\sqrt{3} \end{bmatrix}.$$

**Definition B3.** Given an  $[m \times n]$ -dimensional matrix  $\mathbf{A}$ , the *transpose*  $\mathbf{A}^T = (a_{ij}^T)$  is the  $[n \times m]$ -dimensional matrix with  $a_{ij}^T = a_{ji}$  for  $i = 1, \dots, m$  and  $j = 1, \dots, n$ .

---

**Example B2**

Use the vector  $\mathbf{a}$  and the matrix  $\mathbf{B}$  of Example B1, we then obtain

$$\mathbf{a}^T = \begin{bmatrix} 3 \\ \sqrt{2} \\ -6 \end{bmatrix} \text{ and } \mathbf{B}^T = \begin{bmatrix} 0 & 7 & \sqrt{3} \\ 5 & -6 & -11 \\ \pi & 2 & 1 \end{bmatrix}.$$



The system has either no solution, exactly one solution, or an infinite number of solutions.

### Example C2

Consider the system

$$\begin{aligned}2x_1 + 3x_2 &= 7 \\4x_1 + 6x_2 &= 10.\end{aligned}$$

It is known that if we simultaneously multiply right-hand side and left-hand side of an equation, we do not change its content. Multiplying the first equation by 2, we obtain  $4x_1 + 6x_2 = 14$ . Now the left-hand side of this equation and that of the second equation are equal, but its right-hand side differs, indicating that there is an inherent contradiction in the system. Thus it is no surprise that the system has no solution.

On the other hand, consider the system

$$\begin{aligned}2x_1 + 3x_2 &= 7 \\4x_1 + 6x_2 &= 14.\end{aligned}$$

Multiplying both sides of the first equation by 2 results in the second equation. In other words, the two relations have exactly the same informational content. This means that we really have only a single equation, and since we need one equation to specify the value of each unknown. Hence this system has an infinite number of solutions  $x_1$  and  $x_2 = \frac{1}{3}(7 - 2x_1)$ .

Consider now the case that has exactly one solution. Here, we are not concerned with conditions in which a system has exactly one solution, but our focus is on how to actually obtain such a solution, given that it exists. There are many different versions of the *Gaussian elimination technique* (named after the German mathematician Carl Friedrich Gauss, 1777–1855). In order to illustrate the technique, consider the system of simultaneous linear equations

$$\begin{aligned}2x_1 + 3x_2 - 5x_3 &= 1 && (I) \\x_1 - 2x_2 + 4x_3 &= -3 && (II) \\4x_1 + x_2 + 6x_3 &= 2 && (III)\end{aligned}$$

The idea is to first eliminate one variable, say  $x_3$ , from all equations but one. The system has then one equation in all (here: three) variables, whereas two equations include only the remaining variables (here:  $x_1$  and  $x_2$ ). Among these remaining variables, we now choose another variable to be eliminated (here:  $x_2$ ), and the procedure is repeated. In the end, we have a single equation in just one variable, which is then replaced by its value in all of the remaining equations, resulting in another system, in which again one of the equations is just a function of a single variable, which is replaced by its value everywhere, and so forth, until the system is solved.

Applying this idea to our example, we first eliminate the variable  $x_3$  from Eq. *II* by multiplying (*I*) by 4 and multiplying (*II*) by 5, adding them, and then replacing Eq. *II* by  $4 \times (I) + 5 \times (II)$ . The revised system can then be written as

$$\begin{array}{rcl} 2x_1 + 3x_2 - 5x_3 & = & 1 \quad (I) \\ 13x_1 + 2x_2 & = & -11 \quad (II') = 4 \times (I) + 5 \times (II) \\ 4x_1 + x_2 + 6x_3 & = & 2 \quad (III) \end{array}$$

Next, we eliminate  $x_3$  from Eq. *III* and replace Eq. *III* by  $6 \times (I) + 5 \times (III)$ . This results in the system

$$\begin{array}{rcl} 2x_1 + 3x_2 - 5x_3 & = & 1 \quad (I) \\ 13x_1 + 2x_2 & = & -11 \quad (II') \\ 32x_1 + 23x_2 & = & 16 \quad (III'') = 6 \times (I) + 5 \times (III) \end{array}$$

Since the Eqs. *II'* and *III''* now contain only the two variables  $x_1$  and  $x_2$ , we can eliminate  $x_2$  from (*III''*) by replacing (*III''*) by  $23 \times (II') - 2 \times (III'')$ . This process results in

$$\begin{array}{rcl} 2x_1 + 3x_2 - 5x_3 & = & 1 \quad (I) \\ 13x_1 + 2x_2 & = & -11 \quad (II') \\ 235x_1 & = & -285 \quad (III''') = 23 \times (II') - 2 \times (III'') \end{array}$$

We say that the system is now in triangular form, due to the pattern of coefficients on the left hand side. This allows us to obtain the values of the variables in a recursive procedure. First we can determine the value of  $x_1$  from Eq. *III'''*. Clearly,  $x_1 = -\frac{285}{235} = -\frac{57}{47}$ . Inserting the value of  $x_1$  into Eq. *II'* allows us to solve for the variable  $x_2$ . In particular, we have  $13(-\frac{57}{47}) + 2x_2 = -11$  results in  $x_2 = \frac{112}{47}$ . Finally, inserting the values of  $x_1$  and  $x_2$  into Eq. *I*, we can solve for  $x_3$ . The relation reads  $2(-\frac{57}{47}) + 3(\frac{112}{47}) - 5x_3 = 1$ , which results in  $x_3 = \frac{35}{47}$ . The system has now been completely solved. The solution is  $[x_1, x_2, x_3] = [-\frac{57}{47}, \frac{112}{47}, \frac{35}{47}] \cong [-1.2128, 2.3830, 0.7447]$ . Inserting the values of the unknowns into the original Eqs. *I*, *II*, and *III*, we can verify that the solution is indeed correct. In analogous fashion, we can determine the solution of any system of linear equations. For further details, see any pertinent introductory text on linear algebra or the short summaries in Eiselt and Sandblom (2007) or (2004).

---

## Appendix D: Probability and Statistics

While most chapters in this book deal with deterministic models, some include probabilistic models, in which concepts of probability are needed. This appendix is intended to briefly cover those probabilistic concepts needed in this book. It is by no means intended to replace a thorough knowledge of statistics.

**Definition D1.** The *probability* of an event (or outcome) of a random experiment is a number  $p$  between 0 and 1, which measures the likelihood that the event will occur.

A value of  $p$  close to 0 indicates that the event is unlikely to happen, while a value of  $p$  close to 1 means that the event is very likely. We can interpret the probability as the proportion of times that the even or outcome will occur, if the experiment is repeated a large number of times.

Suppose now that the events are *mutually exclusive* (i.e., the events are distinct and do not overlap) and *collectively exhaustive* (meaning that exactly one of them will occur). We then obtain the following result.

### Theorem D1

If  $p_1, p_2, \dots, p_n$  denote the probabilities of the mutually exclusive and collectively exhaustive outcomes of an experiment, then  $p_1 + p_2 + \dots + p_n = 1$ .

---

### Example D1

Consider an experiment that involves tossing a fair coin three times, suppose that each outcome is either head or tail, standing on edge does not occur. Denote  $H$  for “head” and  $T$  for tail, the outcome “first tail, then head, then tail” is written as  $THT$ . Then there are eight possible outcomes:  $HHH, HHT, HTH, HTT, THH, THT, TTH$ , and  $TTT$ . Given a fair coin, all outcomes are equally likely, and since  $p_1 + p_2 + \dots + p_8 = 1$ , we obtain the result that each outcome has a probability of  $P(HHH) = P(HHT) = \dots = P(TTT) = \frac{1}{8}$ .

The probability of a *composite event* that consists of several outcomes is the sum of probabilities of the outcome of the event. For instance, the event “obtain exactly one tail in three flips of a fair coin” refers to the event  $\{HHT, HTH, THH\}$ , so that the probability of such an event is  $P(\{HHT, HTH, THH\}) = \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{3}{8}$ . The *event space* is defined as the set of all possible events of an experiment.

**Definition D2.** A random variable  $X$  is a function defined on the event space of an experiment.

**Example D2**

Given the above coin tossing experiment, let  $X$  denote the number of heads that come up in the three tosses. There are four possibilities:  $X = 0$  (which occurs only in the event  $\{TTT\}$ , so that the probability of this event is  $P(X = 0) = \frac{1}{8}$ ),  $X = 1$  (which happens if one of the events  $\{HTT, THT, TTH\}$  occurs, so that the probability  $P(X = 1) = \frac{3}{8}$ ),  $X = 2$  (an event that occurs if one of  $\{HHT, HTH, THH\}$  occurs, so that  $P(X = 2) = \frac{3}{8}$ ), and finally  $X = 3$  (which occurs only if  $\{HHH\}$  happens, so that  $P(X = 3) = \frac{1}{8}$ ). Again, the sums of all possible events add up to 1.

In general, we distinguish between two different types of random variables, *discrete* and *continuous random variables*.

**Definition D3.** A random variable  $X$  is called *discrete*, if it can assume only one of (countably many) values  $a_1, a_2, \dots$ . The function  $P(a_j) = P(X = a_j) = p(a_j)$  is called the *discrete probability distribution* (function) of  $X$ . The function  $F(a_j) = P(X \leq a_j)$  is called the *cumulative probability distribution* (function) of  $X$ .

**Example D3**

Again, define the random variable  $X$  as the number of heads in three tosses of a fair coin. Table D.1 shows the probability distribution function for this event.

We note that  $F(a_j)$  is an increasing function of  $a_j$  that eventually reaches the value of 1 for the largest value of  $a_j$  (in case of finitely many outcomes), and that converges towards 1 for infinitely many outcomes  $a_j$ . Since  $P$  and  $F$  are probabilities, we must have  $0 \leq P(a_j) \leq 1$  and  $0 \leq F(a_j) \leq 1$  for all events  $a_j$ .

**Definition D4.** A discrete random variable  $X$  with a probability distribution function  $p(x) = P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}$   $x = 0, 1, 2, \dots$  is called *Poisson-distributed* with parameter  $\lambda$ . In case a random variable follows this distribution, we will write  $X \sim Po(\lambda)$ , where  $Po$  stands for Poisson.

The Poisson distribution is named in honor of Siméon Denis Poisson, a French mathematician, 1781–1840. The distribution is of major importance in queuing (Chap. 13 of this book) and will be extensively used in that context.

**Example D4**

If the random variable  $X \sim Po(\lambda = 1.3)$ , then  $P(X = 2) = p(2) = e^{-1.3} \frac{(1.3)^2}{2!} = 0.2303$ , and  $F(X \leq 1) = p(0) + p(1) = 0.6268$ .

**Table D.1** Probability distribution for the coin toss example

X	P	F
0	$\frac{1}{8}$	$\frac{1}{8}$
1	$\frac{3}{8}$	$\frac{4}{8}$
2	$\frac{3}{8}$	$\frac{7}{8}$
3	$\frac{1}{8}$	$\frac{8}{8}$

**Definition D5.** A random variable  $X$  is called *continuous*, if there exists a function  $f(x)$ , such that the cumulative distribution function  $F(x)$  for  $X$  can be written as  $F(x) = \int_{-\infty}^x f(t)dt$ . The function  $f(x)$  is called the (*probability*) *density function* of  $X$ .

There are two different continuous distributions that are used in this book, the *exponential* and the *normal distribution*.

**Definition D6.** A continuous random variable  $X$  with the density function  $f(x) = \lambda e^{-\lambda x}$  with  $x \geq 0$  is said to *exponentially distributed* with parameter  $\lambda$  (a positive constant). If  $X$  is exponentially distributed, we write  $X \sim Ex(\lambda)$ .

It is not difficult to demonstrate that the cumulative distribution function of an exponentially distributed variable with parameter  $\lambda$  is  $F(x) = 1 - e^{-\lambda x}$ .

**Definition D7.** A continuous random variable  $X$  with density function  $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  is said to follow a *normal distribution* (or, alternatively, *Gaussian distribution*) with parameters  $\mu$  and  $\sigma > 0$ . In such a case, we write  $X \sim N(\mu, \sigma)$ . The distribution  $N(0, 1)$ , i.e., the normal distribution with  $\mu = 0$  and  $\sigma = 1$  is called the *standard normal distribution* and is usually denoted by  $Z$ . Its density function is called the *normal curve* (or *bell curve*), given by the function  $f(x) = \frac{1}{\sqrt{2\pi}}e^{-1/2x^2}$ .

**Definition D8.** The *expected value* (or *mean* or *expectation*)  $E(X)$  or  $\mu$  of a discrete random variable  $X$  is given by  $E(X) = \mu = a_1P(X = a_1) + a_2P(X = a_2) + \dots$ , meaning we multiply each outcome by its associated probability and add them up. For a continuous random variable  $X$ , the expected value is given by  $E(X) =$

$$\mu = \int_{-\infty}^{\infty} tf(t)dt.$$

There are discrete and continuous random variables for which there exists no expected value.

#### Example D5

For the random variable  $X$  of Examples D2 and D3 where  $X$  denotes the number of heads in three tosses of a fair coin, we obtain the expected value  $E(X) = 0\left(\frac{1}{8}\right) + 1\left(\frac{3}{8}\right) + 2\left(\frac{3}{8}\right) + 3\left(\frac{1}{8}\right) = 1\frac{1}{2}$ .

#### Theorem D2

If  $X \sim Po(\lambda)$ , then  $E(X) = \lambda$ , if  $X \sim Ex(\lambda)$ , then  $E(X) = \frac{1}{\lambda}$ , and if  $X \sim N(\mu, \sigma)$ , then  $E(X) = \mu$ .

**Definition D9.** The *variance*  $V(X) = \sigma^2$  of a random variable  $X$  with mean  $\mu$  is defined as  $E((X - \mu)^2)$ . We call  $\sigma = \sqrt{V(X)}$  the *standard deviation* of  $X$ . The expression  $\sigma_\mu = \sigma/\mu$  is referred to as the *coefficient of variation* of  $X$ .

There are discrete and continuous variables for which the mean exists, but the variance does not. One can show that  $V(X) = E(X^2) - \mu^2$  and that  $V(X) \geq 0$  for all random variables  $X$ , as long as  $E(X^2)$ ,  $E(X)$ , and  $V(X)$  exist.

#### Example D6

For the random variable  $X$  of Examples D2 and D3 where  $X$  denotes the number of heads in three tosses of a fair coin, we obtain the variance  $V(X) = E(X^2) - \mu^2 = 3 - 2.25 = 0.75$ . The standard deviation  $\sigma = \sqrt{V(X)} = \sqrt{.75} \approx .8660$ . The coefficient of variation  $\sigma_\mu = \sqrt{.75}/1.5 \approx 0.5774$ .

#### Theorem D3

If the random variable  $X \sim Po(\lambda)$ , then  $V(X) = \lambda$ , if  $X \sim Ex(\lambda)$ , then  $V(X) = \frac{1}{\lambda^2}$ , and if  $X \sim N(\mu, \sigma)$ , then  $V(X) = \sigma^2$ .

#### Theorem D4

If the random variable  $X \sim N(\mu, \sigma)$ , then  $Z = [(X - \mu)/\sigma] \sim N(0, 1)$ .

#### Theorem D5

For any random variable  $X$  and for any given numbers  $a < b$ , we have  $P(a < X \leq b) = F(b) - F(a)$ . For any continuous random variable  $X$ ,  $P(a \leq X \leq b) = P(a \leq X < b) = P(a < X < b) = F(b) - F(a)$  as well as  $P(X = a) = P(X = b) = 0$ .

#### Example D7

Let the random variable  $X \sim N(2.3, 0.90)$  and compute the probability  $P(1.7 \leq X \leq 2.6)$ . We find that  $P(1.7 \leq X \leq 2.6) = P\left(\frac{1.7-2.3}{0.9} \leq \frac{X-2.3}{0.9} \leq \frac{2.6-2.3}{0.9}\right) = P(-0.6667 \leq Z \leq 0.3333) = F(0.3333) - F(-0.6667)$ . The function  $F(x) =$

$\int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-1/2t^2} dt$  is called area under the normal curve and it is tabulated at the end of this section for values  $x \geq 0$ . By virtue of symmetry of the normal curve, we find that  $F(-x) = 1 - F(x)$ , therefore  $F(0.3333) - F(-0.6667) = F(0.3333) - 1 + F(0.6667) = 0.6306 + 1 - 0.7475 = 0.3781$  by reading the table and interpolating as necessary.

It has been observed in practice that empirical data that are bell-shaped and symmetric share some common properties. In particular, the *empirical rule* holds, according to which about 68 % of all observations lie within one standard deviation about the mean, about 95 % of the observations are within two standard deviations about the mean, and virtually all observations are within three standard deviations about the mean. It is worth mentioning that this is not a provable property, but an observed rule that often occurs in practice.

Making now stronger assumptions about the distribution in particular that it is not only bell-shaped and symmetric, but normal, we are able to confirm the assertion of the empirical rule. In particular, for a standard normal variable  $Z \sim N(0, 1)$  we have  $F(0) = 0.5$ ,  $F(1) = 0.8413$ ,  $F(2) = 0.9773$ , and  $F(3) = 0.9987$ . Then  $P(-1 \leq Z \leq 1) = 2(0.8413 - 0.5) = 0.6826$ ,  $P(-2 \leq Z \leq 2) = 2(0.9773 - 0.5) = 0.9546$ , and  $P(-3 \leq Z \leq 3) = 2(0.9987 - 0.5) = 0.9974$ .

We will now consider events  $A, B, C, \dots$ , which are sets of outcomes of experiments.

**Definition D10.** Given the events  $A$  and  $B$ , the *union*  $A \cup B$  is the event consisting of outcomes that are in  $A$  or in  $B$  or both. In contrast, the *intersection*  $A \cap B$  is the event that consists of outcomes that are in both  $A$  and  $B$ .

**Theorem D6 (The addition law for probabilities)**

For any events  $A$  and  $B$ , we have  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ .

**Example D8**

Recall Example D4, in which  $X \sim P_o(\lambda = 1.3)$ . Furthermore, let  $A = \{X = 0 \text{ or } 1\}$  and  $B = \{1 \text{ or } 2\}$ . Then  $A \cup B = \{X = 0, 1, \text{ or } 2\}$ , while  $A \cap B = \{X = 1\}$ . We then find  $P(A) = p(0) + p(1) = 0.6268$ ,  $P(B) = p(1) + p(2) = 0.5846$ ,  $P(A \cup B) = p(0) + p(1) + p(2) = 0.8571$ ,  $P(A \cap B) = p(1) = 0.3543$ . In accordance with the theorem, we find that  $P(A) + P(B) - P(A \cap B) = 0.6268 + 0.5846 - 0.3543 = 0.8571 = P(A \cup B)$ .

**Definition D11.** The sets  $A$  and  $B$  are said to be *collectively exhaustive*, if their union  $A \cup B$  includes all possible outcomes of an experiment. The two sets are called *mutually exclusive*, if their intersection is empty. The *complement*  $\bar{A}$  of a set  $A$  (sometimes also written as  $\neg A$ ) is the set of all possible outcomes not in  $A$ .

As far as probabilities are concerned,  $P(\bar{A}) = 1 - P(A)$ .

The addition law for probabilities can be generalized. For instance, for mutually exclusive sets  $A_1, A_2, \dots, A_m$ , we obtain  $P(A_1 \cup A_2 \cup \dots \cup A_m) = P(A_1) + P(A_2) + \dots + P(A_m)$ . To establish a multiplication law for probabilities, we need the following

**Definition D12.** For any events  $A$  and  $B$  with  $P(B) \neq 0$ , the *conditional probability of  $A$  given  $B$*  is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

#### Example D9

Consider an experiment, in which the random variable  $X$  denotes that at most two heads come up in three tosses of a fair coin, i.e.,  $X \leq 2$ , and define  $B$  as an event that sees at least one head in three tosses of a fair coin, i.e.  $X \geq 1$ . Then  $A \cap B = \{X = 1 \text{ or } 2\}$ , so that  $P(A \cap B) = \frac{3}{8} + \frac{3}{8} = \frac{3}{4}$  and  $P(B) = \frac{3}{8} + \frac{3}{8} + \frac{1}{4} = \frac{7}{8}$ .

Therefore,  $P(A|B) = \frac{3/4}{7/8} = \frac{6}{7} \approx 0.8571$ .

#### Theorem D7 (The multiplication law for probabilities)

For any events  $A$  and  $B$ ,  $P(A \cap B) = P(A|B) P(B)$ .

Note that if  $P(B) = 0$ , then  $P(A|B)$  is not defined, but in this case the right-hand side is interpreted as being zero.

**Definition D13.** The events  $A$  and  $B$  are said to be *statistically independent*, if  $P(A \cap B) = P(A) P(B)$ .

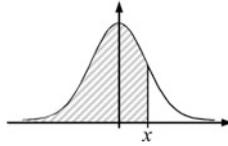
#### Theorem D8 (Bayes's theorem)

Let the events  $A_1, A_2, \dots, A_m$  be mutually exclusive and collectively exhaustive. Then for any event  $B$  with  $P(B) \neq 0$ , we have

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots + P(B|A_m)P(A_m)}, i = 1, \dots, m$$

The theorem of Bayes (Thomas Bayes, English clergyman, 1702–1761) will be used in Chap. 9 of this book. In the context of Bayes's theorem, the unconditional probabilities  $P(A_i)$  are called *prior probabilities*, while the conditional probabilities  $P(A_i|B)$  are referred to as *posterior probabilities*.

**Area under the normal curve**



<i>x</i>	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
.00	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
.10	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5754
.20	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
.30	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
.40	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
.50	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
.60	.7258	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7518	.7549
.70	.7580	.7612	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
.80	.7881	.7910	.7939	.7967	.7996	.8023	.8051	.8079	.8106	.8133
.90	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9430	.9441
1.6	.9452	.9463	.9474	.9485	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9700	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9762	.9767
2.0	.9773	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9865	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9978	.9980	.9980	.9981
2.9	.9981	.9982	.9983	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990
3.5	.999767									
4.0	.9999683									
4.5	.99999660									
5.0	.999999713									

---

## Bibliography

- Axsäter S (2006) Inventory control, 2nd edn. Springer, Berlin/Heidelberg/New York
- Blumenfeld D (2001) Operations research calculations handbook. CRC Press, Boca Raton
- Budnick FS, Mojena R, Vollmann TE (1988) Principles of operations research for management, 2nd edn. Richard D. Irwin, Homewood
- Chan Y (2005) Location, transport and land-use: modelling spatial-temporal information. Springer, Berlin/Heidelberg/New York
- Chhajed D, Lowe TJ (eds) (2008) Building intuition: insights from basic operations management models and principles. Springer Science + Business Media LLC, Boston
- Dantzig GB (1963) Linear programming and extensions. Princeton University Press, Princeton
- Dantzig GB, Thapa MN (1997) Linear programming: introduction. Springer, New York
- Dantzig GB, Thapa MN (2004) Linear programming: theory and extensions. Springer, New York
- Daskin MS (1995) Network and discrete location: models, algorithms, and applications. Wiley, New York
- Eiselt HA, Sandblom C-L (2000) Integer programming and network models. Springer, Berlin/Heidelberg/New York
- Eiselt HA, Sandblom C-L (2004) Decision analysis, location models, and Scheduling Problems. Springer, Berlin/Heidelberg/New York
- Eiselt HA, Sandblom C-L (2007) Linear programming and its applications. Springer, Berlin/Heidelberg/New York
- Erlenkotter D (1990) Ford Whitman Harris and the economic order quantity model. *Oper Res* 38:937–946
- Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. WH Freeman, San Francisco
- Garner Garille S, Gass SI (1981) Stigler's diet problem revisited. *Oper Res* 49:1–13
- Gass SI, Assad AA (2005) An annotated timeline of operations research: an informal history. Kluwer Academic, New York
- Hakimi SL (1964) Optimal location of switching centers and medians of a graph. *Oper Res* 12: 450–459
- Hillier FS, Lieberman GJ (2010) Introduction to operations research, 9th edn. McGraw-Hill/Irwin, New York
- Hillier FS, Hillier MS, Schmedders K, Stephens M (2008) Introduction to management science: a modeling and case study approach with spreadsheets, 3rd edn. McGraw-Hill/Irwin, Boston
- Lind D, Marchal W, Mason R, Gupta S, Kabadi S, Singh J (2004) Statistical techniques in business and economics. McGraw-Hill, Toronto, First Canadian Edition
- Olson DL (1996) Decision aids for selection problems, Springer series in operations research. Springer, New York
- Råde L, Westergren B (1990) Beta: mathematics handbook, 2nd edn. CRC Press, Boca Raton
- Walker RC (1999) Introduction to mathematical programming. Prentice Hall, Upper Saddle River
- Winston WL (2004) Operations research: applications and algorithms, 4th edn. Duxbury, Belmont

---

# Index

## A

- ABC classification, 343, 364
- Activity
  - critical, 258, 262–269, 273, 275–278
  - duration, 258–266, 268, 269, 271–277, 279
- Activity-on-arc (AOA) representation, 259
- Activity-on-node (AON) representation, 259
- AHP. *See* Analytic hierarchy process (AHP)
- Algorithms
  - exact, 285, 290, 297, 417
  - heuristic, 132, 231, 242, 246, 289, 291, 417–424
- All-integer linear programming (AILP), 124, 125, 127, 166
- Allocation problem, 26–30, 283
- Analytic hierarchy process (AHP), 336–340, 342
- Anticipated payoffs, 308–310, 326, 329
- AOA representation. *See* Activity-on-arc (AOA) representation
- AON representation. *See* Activity-on-node (AON) representation
- Arborescence, 190
- Arc, 176, 178, 179, 182–184, 186–190, 195–205, 208, 210, 211, 216, 217, 233, 236, 241, 259, 306, 371
- Arrival rate, 384–387, 389, 393, 394
- Arrival time, 284, 405
- Aspiration level, 114
- Assignment problems, 13, 40–56, 144, 157–159, 172, 173, 205, 215, 248

## B

- Backorders, 344, 346, 352–354, 360, 407–410
- Backward
  - pass, 261, 264
  - recursion, 180, 182, 262, 275, 321
  - sweep, 261–263, 265, 273

- Basis point, 82
- Bayes's rule, 310, 317–319, 321
- Bayes's theorem, 318, 437
- Bin packing, 422
- Bisection search, 236, 237, 252
- Blending problems, 37–41
- Bottleneck, 86, 88, 93, 183, 258, 263, 264
- Branch-and-bound methods, 149–154
- Breadth-first-search, 180
- Break-even analysis, 7
- Breakthrough, 123, 179, 180, 182
- Budget constraint, 4, 17, 27, 67, 113–115, 159

## C

- Calling population, 382, 384
- Catchment areas, 420
- Center-of-gravity, 239, 240, 253
- Center problems, 223, 233–236
- Certainty equivalent, 322, 323
- Changes
  - objective function coefficients, 76, 77
  - right-hand side values, 68, 81, 83, 84, 87
  - structural and parameter, 74
- Channel, 382, 386, 393, 403
- Chinese postman problem, 196, 197, 201, 214, 216
- Coefficient of variation, 339, 340, 434
- Column
  - dominated, 225, 227, 248
  - essential, 225, 227
- Competitive location, 247
- Complementary slackness conditions, 100, 103
- Conservation equations, 178, 184, 187
- Constraints
  - addition and deletion of, 67, 75, 76
  - binding, 25, 72
  - conditional, 129, 136, 137, 164
  - method, 110, 112, 120

Contour lines, 61, 69, 420  
 Convexity, 64  
 Corner point theorem, 63, 65, 70, 124, 126, 144, 241  
 Cost  
   carrying, 35, 50, 346, 352, 357–362, 407, 410  
   holding, 33, 35, 346–349, 352–354, 356, 361, 362, 365–367, 407, 408  
   ordering, 346–349, 352, 361, 362, 365–367, 407, 410  
   shortage, 346, 352–354, 360, 364–366, 407, 408, 410  
 Covering matrix, 226, 228, 248, 249  
 Covering problems, 225–230, 235, 236, 248  
 CPM. *See* Critical path method  
 Crashing, 265  
 Crash time, 265, 268  
 Critical path, 257–269, 273–281  
 Critical path method (CPM), 257–264, 273  
   project acceleration, 264–269  
 Cut, minimal, 183, 208, 209  
 Cutting plane method, 167  
 Cutting stock problems, 132–136  
 Cycle length, 347, 349, 350, 401

**D**

Decision analysis, 303–331, 333, 336, 369  
 Decision trees, 307, 317–319, 321, 323, 324, 328, 330  
 Decomposition principle, 7  
 Degeneracy  
   dual, 69, 70, 105, 146, 147  
   primal, 71, 82, 146, 166  
 Depth-first search, 180  
 Destination, 40–45, 95, 177, 185, 246, 247, 417  
 Diet problem, 20, 22, 91, 136–138, 140  
 Dijkstra method, 190, 211  
 Discrete event simulation, 397, 398  
 Distribution  
   exponential, 402  
   normal, 274, 433  
 Divisibility, 14, 15  
 Duality, 96–104  
 Dual problem  
   relations to primal problem, 97, 98, 100, 101, 103  
   setting up, 101–103  
 Due time, 284  
 Dynamic programming, 3, 179

**E**

Earliest due date algorithm (EDD algorithm), 287, 298  
 Earliest possible finishing times (EF), 260  
 Earliest possible starting times (ES), 260  
 Economic order quantity (EOQ), 346–363, 366  
 EDD algorithm. *See* Earliest due date algorithm (EDD algorithm)  
 Edge, 175, 176, 178, 195, 197–200, 215, 431  
 EF. *See* Earliest possible finishing times (EF)  
 Efficiency, 41, 105, 143, 246, 320, 321, 328, 330, 423  
 Either-or constraints, 129  
 Empirical rule, 435  
 Employee scheduling, 30–32  
 EMV. *See* Expected monetary values (EMV)  
 EOQ. *See* Economic order quantity  
 EPII. *See* Expected payoff with imperfect information (EPII)  
 EPPI. *See* Expected payoff with perfect information (EPPI)  
 Equity objectives, 143, 246  
 ES. *See* Earliest possible starting times (ES)  
 Euclidean distance, 222  
 EVPI. *See* Expected value of perfect information (EVPI)  
 EVSI. *See* Expected value of imperfect (or sample) information (EVSI)  
 Excess variable, 17, 85, 86, 114, 146, 147, 167  
 Expected monetary values (EMV), 310, 313–317, 320, 321, 327  
 Expected payoff with imperfect information (EPII), 320, 321  
 Expected payoff with perfect information (EPPI), 316  
 Expected value, definition, 26, 27, 321, 322, 338, 360, 433, 434  
 Expected value of imperfect (or sample) information (EVSI), 320  
 Expected value of perfect information (EVPI), 315, 316, 320, 321, 328, 330  
 Extreme points, 59, 60, 64, 65, 69, 70, 81, 82, 109–112, 126, 144  
   number of, 112

**F**

Facilities  
   extensive, 247  
   undesirable, 223, 245, 246

- Feasibility, 6–8, 23, 65, 66, 109, 110, 157, 387, 390
- Feasible
- direction methods, 64
  - set, 59, 63–67, 69, 70, 72, 73, 78, 81–83, 101, 109, 110, 112, 113, 118, 119, 124–126, 144, 146, 147, 149–152, 246
  - solution, nonexistence of, 66, 67, 83
- First fit algorithm, 422
- First passage time, 372, 373
- Fixed charges, 139–142
- Flow
- balancing equations, 178
  - mean time, 285, 286, 293, 296, 298, 299
  - pattern and value, 180, 183
- Floyd-Warshall method, 212
- Forward
- pass, 260, 264
  - recursion, 260, 275
  - sweep, 260, 261, 263, 265, 273
- G**
- Games against nature, 304
- Gantt chart, 2, 269, 276–277, 279–281, 287, 290, 296–301
- Gaussian elimination, 428
- Gini index, 246
- Goal programming, 113–115, 121–122
- Graph
- connected, 176, 194, 197, 199
  - directed, 175, 176, 196, 197, 199–201, 216
  - mixed, 175, 176
  - undirected, 175, 176, 178, 191, 194, 196, 197, 200, 214, 253
- Graphical solution method, 16, 57–66, 69, 71–72
- Greedy method, 154, 244, 418, 420
- Guillotine cuts, 135
- H**
- Halfplane, 58, 107, 108
- Halfspace, 58, 59, 67, 81, 82, 106
- Heuristic methods, 144, 154–173, 285, 417
- Hub location, 246
- Hyperplane, 57–61, 70, 71, 81, 82
- I**
- Implementability, 6
- Improvement cone, 107–110, 117, 119
- Incremental technique, 64
- Influence diagrams, 305, 306
- Institute for Operations Research and Management Science (INFORMS), 1
- Integer programming, 2, 15, 123–173, 176, 222, 226–228, 247, 272, 288, 289, 303
- Integrality gap, absolute and relative, 127
- Interarrival time, 385, 403–406
- Interchange heuristic, 155
- Interval constraints, 68
- Inventory models, 33–37, 50, 343–367
- Inventory position, 344, 345, 352, 363, 364
- Investment allocation, 26, 29
- Iso-profit lines, 61, 63, 106
- J**
- Jackson's job shop algorithm, 297
- Jackson's rule, 287
- Johnson's algorithm, 296, 297
- Just-in-time (JIT) inventory system, 344
- K**
- Kendall's notation, 383
- Kirchhoff node equations, 178
- Knapsack problems, 103, 130, 131, 154, 158, 159
- Königsberg bridge problem, 2, 175, 195
- Kruskal's method, 194
- L**
- Labeling methods, 179
- Land use problem, 138, 139
- LAPT algorithm. *See* Longest alternate processing time algorithm (LAPT) algorithm
- Lateness, 284, 285, 287, 294, 298, 299
- Layout problems, 248
- Lead time, stochastic, 359–363
- Linear congruence methods, 400
- Linearity, 14, 15, 41
- Linear programming relaxation, 127, 131, 144, 145, 147, 149, 152, 153, 166
- List scheduling methods, 290
- Little's formula, 385, 388
- Location
- continuous, 221
  - discrete, 222
  - models, 188, 190, 221–255
- Location-allocation heuristic, 243, 254
- Location set covering problem (LSCP), 225, 228

- Logical variables, 123, 137
- Longest alternate processing time algorithm (LAPT) algorithm, 294, 295
- Longest processing time (LPT) first algorithm, 290, 291, 298, 299
- Lorenz curve, 246
- LPT algorithm. *See* Longest processing time (LPT) first algorithm
- LSCP. *See* Location set covering problem (LSCP)
- M**
- Machine scheduling  
   dedicated, 294–302  
   parallel, 289–294  
   single, 285–289
- Makespan, 285, 286, 290, 291, 294, 295, 297, 299
- Management science, 1, 3
- Manhattan distance, 222
- Markov chains, 369–380
- Markovian property, 370
- Material requirements planning, 344
- Matrix, 42, 46, 55, 172, 173, 191–193, 201, 203, 206, 213, 215, 219, 225–228, 231, 232, 234, 241–244, 248, 249, 254, 305, 308–310, 312, 316, 318, 324–327, 330, 333–341, 370, 372, 374, 377–380, 425, 426
- Maximal covering location problem (MCLP), 230–233, 249
- Maximal flow problem, 183
- Maximin rule, 308, 309
- MCDM. *See* Multicriteria decision making problems (MCDM)
- MCLP. *See* Maximal covering location problem (MCLP)
- McNaughton's Wrap-Around Rule, 292
- Mean recurrence time, 372, 374, 378, 380
- Median problems, 223, 236–245, 251, 253
- Midsquare method, 400
- MILPs. *See* Mixed-integer linear programming problems (MILPs)
- Min cost feasible flow problem, 183, 186
- Min-cut max flow theorem, 183
- Minimax problems, 142, 236, 239, 252, 271, 309, 310, 325
- Mixed-integer linear programming problems (MILPs), 124
- Model, deterministic and stochastic, 14, 257, 383
- Modeling process, 1, 9–11, 22, 24, 74, 75, 115
- Modified distribution (MODI) method, 43
- Multicriteria decision making problems (MCDM), 105, 304
- Multiobjective programming, 105–122
- Multistart procedure, 207
- N**
- Nash equilibria, 247
- Neighborhood search, 423
- Network flows, 177–186
- Newspaper boy problem, 310
- Nodes, reachable, 176
- Nonbreakthrough, 179, 182, 209
- Nondominated frontier, 111, 117–119
- Normal duration, 264, 275, 276
- Normalization technique, 342
- O**
- Objective function, gradient of, 61–63, 68, 69, 77–79, 81, 106, 109, 118
- Operations research  
   definition, 1, 7  
   elements, 3–8  
   journals, 3, 9
- Opportunity costs, 84, 85, 99, 311
- Optimality, 6–8, 105, 166, 239, 290, 293, 302, 315
- Optimal solution  
   alternative, 70, 76, 79, 168, 203, 206, 217, 298  
   economic analysis of, 84–96  
   unbounded, 68, 69, 75, 76, 78, 101, 128  
   unique, 64, 79, 103, 158, 172, 173
- Optimist's rule, 309
- Origin, 40–45, 58, 61, 62, 69, 76, 82, 95, 96, 110, 177, 185, 246, 247, 421
- Overshipments, 44, 45
- P**
- Parameters, 5–7, 14, 15, 22, 33, 34, 42, 50, 74, 76, 77, 159, 222, 230, 283, 284, 343, 344, 347, 349, 353, 356, 359, 383, 386, 390, 400, 401, 406, 408, 433
- Path, 65, 69, 152, 175, 176, 180, 182, 186–194, 199–203, 208–213, 215, 217, 218, 222, 236, 241, 242, 257–269, 273–281, 371, 374
- Payoff table, 305
- PERT. *See* Program evaluation and review technique (PERT)

- Pessimistic rule, 308  
 Phase, construction and improvement, 418  
 Pollaczek-Khinchine formula, 386  
 Polytope, 59  
 Postoptimality analyses, 73–96, 101  
 Precedence relations, 258–260  
 Preemption, 286, 292, 299  
 Probabilistic, 214, 223, 258, 272, 283, 431  
 Probability  
   conditional, 316–318, 321, 328, 330, 370, 373, 401, 436, 437  
   definition, 431–438  
   distribution, 258, 272, 304, 305, 316, 359, 360, 363, 373, 432, 433  
   posterior, 318, 319, 321, 437  
   prior, 318, 321, 329, 437  
 Processing time, 3, 18, 19, 77, 139, 142, 284–288, 290–301  
 Process, stationary, 370  
 Production-inventory models, 36, 49, 344  
 Production-lot size model, 357–359  
 Production planning, 6, 18–19, 47  
 Program evaluation and review technique (PERT), 257, 258, 272–275  
 Project networks, 257–281, 285
- Q**
- Quantity discounts, 346, 354–357, 366
- R**
- Random numbers, true and pseudo, 399, 400  
 Random variables, 272, 402, 403, 432, 434  
 Ready time, 284, 294, 298, 423  
 Rectilinear distance, 237–240  
 Reduced costs, 84  
 Redundancy, 70  
 Regret criterion, 309, 325, 330  
 Release time, 284  
 Reorder point, 351, 359–366, 407–410  
 Reshipments, 44, 45  
 Resource requirement graph, 269, 279  
 Resources, scarce, 1, 26–28, 49, 233, 258  
 Review systems, continuous and periodic, 363, 364
- Risk  
   aversion, 4, 322  
   neutral, 322, 325  
   seeking, 322
- Robust optimization, 309
- Routing, arc and node, 196–219
- Row, dominated, 227, 248
- 100% Rule, 79–81, 83
- S**
- Satisficing, 113  
 Schedule length, 285, 290–299, 301, 302  
 Selection problem, 303–305  
 Sensitivity analyses, graphical, 74–84  
 Service  
   rate, 384–386, 391, 393, 395  
   station, 383–388, 403, 405  
   time, 287, 378, 385–387, 390, 393, 394, 403, 405, 406  
 Shadow prices, 86, 99  
 Shop  
   flow, 284, 295–297, 301, 302  
   job, 284, 297  
   open, 284, 294–296, 301, 302  
 Shortest path problems, 186–194  
 Shortest processing time (SPT) algorithm, 298, 586, 587  
 Simplex method, 3, 13, 14, 64, 65, 70, 71, 110  
 Simulated annealing, 424  
 Simulation, deterministic and stochastic, 397  
 Simultaneous linear equations, 59, 60, 63, 375, 427–429  
 Sink, 177–189, 210, 218, 259–261, 263, 268  
 Smith's ratio rule, 286  
 Solution  
   definition, 109–111  
   efficient, 109, 110, 119, 135, 145, 176  
   nondominated, 109–112, 118  
   noninferior, 109, 120, 121  
   pareto-optimal, 106, 109  
 Source, 9, 177–180, 182–187, 189, 190, 209, 210, 218, 260, 261, 263, 268  
 Spanning tree, minimal, 194, 213  
 SPT algorithm. *See* Shortest processing time (SPT) algorithm  
 Standard deviation, 272, 274, 275, 277, 279, 340, 387, 434  
 State space, absorbing, 370  
 Statistical independence, 437  
 Steady state solution, 374–375, 397  
 Stochastic processes, 369–380  
 Subtour elimination constraints, 205  
 Supply chain management, 344  
 Swap heuristic, 171
- T**
- Tabu search, 423, 424  
 Tardiness, 284, 285, 288, 296  
 Target value, 114, 311, 312, 325  
 Tool crib, 390, 391  
 Tradeoffs, 105, 106, 115, 233  
 Traffic intensity, 385, 406

Transient states, 384, 398  
Transition probabilities, 369–371, 376, 379  
Transportation problems, 40, 41, 45  
    (un-)balanced, 41  
Traveling salesman problem, 196, 204, 218  
Tree (graph), 176

## U

Uncertainty, 305, 308–312  
Undesirable facilities, 245  
Utility theory, 303, 322–331  
Utilization rate. *See* Traffic intensity

## V

Variables  
    addition and deletion of, 74, 75  
    deviational, 114–116  
    slack, 17, 100, 146, 147  
    surplus, 17, 114

Variance, 143, 246, 272–274, 277, 340, 383,  
    386, 387, 434  
Varignon frame, 240  
Vector, 61, 105–113, 117, 119, 231, 232,  
    241–244, 255, 309, 337–339, 342, 373,  
    376, 425–426

Vector optimization, 105–113, 117, 119  
Vertex substitution method, 253, 254  
Vertices, 67, 125, 150, 152, 175, 221  
von Stackelberg solutions, 247

## W

Wald's rule, 308–310, 323  
Weighted shortest processing time (WSPT)  
    algorithm, 286, 287  
Weighting method, 110–112, 115, 119  
Weiszfeld method, 240, 253  
Workload balancing, 142–143  
WSPT algorithm. *See* Weighted shortest  
    processing time (WSPT) algorithm