Glossary for Chapter 1

Algorithm A systematic solution procedure for solving a particular type of problem. (Section 1.4)

OR Courseware The overall name of the set of software packages that are shrink-wrapped with the book. (Section 1.4)

Glossary for Chapter 2

Algorithm A systematic solution procedure for solving a particular type of problem. (Section 2.3)

Constraint An inequality or equation in a mathematical model that expresses some restrictions on the values that can be assigned to the decision variables. (Section 2.2)

Data mining A technique for searching large databases for interesting patterns that may lead to useful decisions. (Section 2.1)

Decision support system An interactive computer-based system that helps managers use data and models to support their decisions. (Section 2.5)

Decision variable An algebraic variable that represents a quantifiable decision to be made. (Section 2.2)

Heuristic procedure An intuitively designed procedure for seeking a good (but not necessarily optimal) solution for the problem at hand. (Section 2.3)

Linear programming model A mathematical model where the mathematical functions appearing in both the objective function and the constraints are all linear functions. (Section 2.2)

Metaheuristic A general kind of solution method that provides both a general structure and strategy guidelines for designing a specific heuristic procedure to fit a particular kind of problem. (Section 2.3)

Model An idealized representation of something. (Section 2.2)

Model validation The process of testing and improving a model to increase its validity. (Section 2.4)

Objective function A mathematical expression in a model that gives the overall measure of performance for a problem in terms of the decision variables. (Section 2.2)

Optimal solution A best solution for a particular problem. (Section 2.3)

Overall measure of performance A composite measure of how well the decision maker's ultimate objectives are being achieved. (Section 2.2)

Parameter One of the constants in a mathematical model. (Section 2.2)

Retrospective test A test that involves using historical data to reconstruct the past and then determining how well the model and the resulting solution would have performed if they had been used. (Section 2.4)

Satisficing Finding a solution that is good enough (but not necessarily optimal) for the problem at hand. (Section 2.3)

Sensitive parameter A model's parameter whose value cannot be changed without changing the optimal solution. (Section 2.3)

Sensitivity analysis Analysis of how the recommendations of a model might change if any of the estimates providing the numbers in the model eventually need to be corrected. (Sections 2.2 and 2.3)

Suboptimal solution A solution that may be a very good solution, but falls short of being optimal, for a particular problem. (Section 2.3)

Glossary for Chapter 3

Additivity The additivity assumption of linear programming holds if every function in the model is the sum of the individual contributions of the respective activities. (Section 3.3)

Blending problem A type of linear programming problem where the objective is to find the best way of blending ingredients into final products to meet certain specifications. (Section 3.4)

Certainty The certainty assumption of linear programming holds if the value assigned to each parameter of the model is assumed to be a known constant. (Section 3.3)

Changing cells The cells in a spreadsheet model that show the values of the decision variables. (Section 3.6)

Constraint A restriction on the feasible values of the decision variables. (Section 3.2)

Corner-point feasible (CPF) solution A solution that lies at the corner of the feasible region. (Section 3.2)

Data cells The cells in a spreadsheet that show the data of the problem. (Section 3.6)

Decision variable An algebraic variable that represents a quantifiable decision, such as the level of a particular activity. (Section 3.2)

Divisibility The divisibility assumption of linear programming holds if all the activities can be run at fractional levels. (Section 3.3)

Feasible region The geometric region that consists of all the feasible solutions. (Sections 3.1 and 3.2)

Feasible solution A solution for which all the constraints are satisfied. (Section 3.2)

Functional constraint A constraint with a function of the decision variables on the left-hand side. All constraints in a linear programming model that are not nonnegativity constraints are called functional constraints. (Section 3.2)

Graphical method A method for solving linear programming problems with two decision variables on a two-dimensional graph. (Section 3.1)

Infeasible solution A solution for which at least one constraint is violated. (Section 3.2)

Mathematical modeling language Software that has been specifically designed for efficiently formulating large mathematical models, including linear programming models. (Section 3.7)

Nonnegativity constraint A constraint that expresses the restriction that a particular decision variable must be nonnegative (greater than or equal to zero). (Section 3.2)

Objective function The part of a mathematical model such as a linear programming model that expresses what needs to be maximized or minimized, depending on the objective for the problem. (Section 3.2)

Optimal solution A best feasible solution according to the objective function. (Section 3.1)

Output cells The cells in a spreadsheet that provide output that depends on the changing cells. (Section 3.6)

Parameter One of the constants in a mathematical model, such as the coefficients in the objective function or the coefficients and right-hand sides of the functional constraints. (Section 3.2)

Product-mix problem A type of linear programming problem where the objective is to find the most profitable mix of production levels for the products under consideration. (Section 3.1)

Proportionality The proportionality assumption of linear programming holds if the contribution of each activity to the value of each function in the model is proportional to the level of the activity. (Section 3.3)

Range name A descriptive name given to a block of cells in a spreadsheet that immediately identifies what is there. (Section 3.6)

Sensitivity analysis Analysis of how sensitive the optimal solution is to the value of each parameter of the model. (Section 3.3)

Simplex method A remarkably efficient solution procedure for solving linear programming problems. (Introduction)

Slope-intercept form For the geometric representation of a linear programming problem with two decision variables, the slope-intercept form of a line algebraically displays both the slope of the line and the intercept of this line with the vertical axis. (Section 3.1)

Solution Any single assignment of values to the decision variables, regardless of whether the assignment is a good one or even a feasible one. (Section 3.2)

Solver A software package for solving certain types of mathematical models, such as linear programming models. (Section 3.7)

Target cell The output cell in a spreadsheet model that shows the overall measure of performance of the decisions. (Section 3.6)

Unbounded *Z* (or unbounded objective) The constraints do not prevent improving the value of the objective function (Z) indefinitely in the favorable direction. (Section 3.2)

Glossary for Chapter 4

Adjacent BF solutions Two BF solutions are adjacent if all but one of their nonbasic variables are the same. (Section 4.2)

Adjacent CPF solutions Two CPF solutions of an n-variable linear programming problem are adjacent to each other if they share n-1 constraint boundaries. (Section 4.1)

Allowable range for a right-hand side The range of values for this right-hand side b_i over which the current optimal BF solution (with adjusted values for the basic variables) remains feasible, assuming no change in the other right-hand sides. (Section 4.7)

Allowable range to stay optimal The range of values for a coefficient in the objective function over which the current optimal solution remains optimal, assuming no change in the other coefficients. (Section 4.7)

Artificial variable A supplementary variable that is introduced into a functional constraint in = or ≥ form for the purpose of being the initial basic variable for the resulting equation. (Section 4.6)

Artificial-variable technique A technique that constructs a more convenient artificial problem for initiating the simplex method by introducing an artificial variable into each constraint that needs one because the model is not in our standard form. (Section 4.6)

Augmented form of the model The form of a linear programming model after its original form has been augmented by the supplementary variables needed to apply the simplex method. (Section 4.2)

Augmented solution A solution for the decision variables that has been augmented by the corresponding values of the supplementary variables that are needed to apply the simplex method. (Section 4.2)

Barrier algorithm (or barrier method) An alternate name for *interior-point algorithm* (defined below) that is motivated by the fact that each constraint boundary is treated as a barrier for the trial solutions generated by the algorithm. (Section 4.9)

Basic feasible (BF) solution An augmented CPF solution. (Section 4.2)

Basic solution An augmented corner-point solution. (Section 4.2)

Basic variables The variables in a basic solution whose values are obtained as the simultaneous solution of the system of equations that comprise the functional constraints in augmented form. (Section 4.2)

Basis The set of basic variables in the current basic solution. (Section 4.2)

BF solution See *basic feasible solution*.

Big M method A method that enables the simplex method to drive artificial variables to zero by assigning a huge penalty (symbolically represented by M) to each unit by which an artificial variable exceeds zero. (Section 4.6)

Binding constraint A constraint that holds with equality at the optimal solution. (Section 4.7)

Constraint boundary A geometric boundary of the solutions that are permitted by the corresponding constraint. (Section 4.1)

Convex combination of solutions A weighted average of two or more solutions (vectors) where the weights are nonnegative and sum to 1. (Section 4.5)

Corner-point feasible (CPF) solution A solution that lies at a corner of the feasible region, so it is a corner-point solution that also satisfies all the constraints. (Section 4.1)

Corner-point solution A solution of an n-variable linear programming problem that lies at the intersection of n constraint boundaries. (Section 4.1)

CPF solution See *corner-point feasible solution*.

Degenerate basic variable A basic variable whose value is zero. (Section 4.4)

Degenerate BF solution A BF solution where at least one of the basic variables has a value of zero. (Section 4.4)

Edge of the feasible region A line segment that connects two adjacent CPF solutions. (Section 4.1)

Elementary algebraic operations Basic algebraic operations (multiply or divide an equation by a nonzero constant; add or subtract a multiple of one equation to another) that are used to reduce the current set of equations to proper form from Gaussian elimination. (Section 4.3)

Elementary row operations Basic algebraic operations (multiply or divide a row by a nonzero constant; add or subtract a multiple of one row to another) that are used to reduce the current simplex tableau to proper form from Gaussian elimination. (Section 4.4)

Entering basic variable The nonbasic variable that is converted to a basic variable during the current iteration of the simplex method. (Section 4.3)

Exponential time algorithm An algorithm for some type of problem where the time required to solve any problem of that type can be bounded above only by an exponential function of the problem size. (Section 4.9)

Gaussian elimination A standard procedure for obtaining the simultaneous solution of a system of linear equations. (Section 4.3)

Initial BF solution The BF solution that is used to initiate the simplex method. (Section 4.3)

Initialization The process of setting up an iterative algorithm to start iterations. (Sections 4.1 and 4.3)

Interior point A point inside the boundary of the feasible region. (Section 4.9)

Interior-point algorithm An algorithm that generates trial solutions inside the boundary of the feasible region that lead toward an optimal solution. (Section 4.9)

Iteration Each execution of a fixed series of steps that keep being repeated by an iterative algorithm. (Sections 4.1 and 4.3)

Iterative algorithm A systematic solution procedure that keeps repeating a series of steps, called an iteration. (Section 4.1)

Leaving basic variable The basic variable that is converted to a nonbasic variable during the current iteration of the simplex method. (Section 4.3)

Minimum ratio test The set of calculations that is used to determine the leaving basic variable during an iteration of the simplex method. (Section 4.3)

Nonbasic variables The variables that are set equal to zero in a basic solution. (Section 4.2)

Optimality test A test of whether the solution obtained by the current iteration of an iterative algorithm is an optimal solution. (Sections 4.1 and 4.3)

Parametric linear programming The systematic study of how the optimal solution changes as several of the model's parameters continuously change simultaneously over some intervals. (Section 4.7)

Pivot column The column of numbers below row 0 in a simplex tableau that is in the column for the current entering basic variable. (Section 4.4)

Pivot number The number in a simplex tableau that currently is at the intersection of the pivot column and the pivot row. (Section 4.4)

Pivot row The row of a simplex tableau that is for the current leaving basic variable. (Section 4.4)

Polynomial time algorithm An algorithm for some type of problem where the time required to solve any problem of that type can be bounded above by a polynomial function of the size of the problem. (Section 4.9)

Postoptimality analysis Analysis done after an optimal solution is obtained for the initial version of the model. (Section 4.7)

Proper form from Gaussian elimination The form of the current set of equations where each equation has just one basic variable, which has a coefficient of 1, and this basic variable does not appear in any other equation. (Section 4.3)

Reduced cost The reduced cost for a nonbasic variable measures how much its coefficient in the objective function can be increased (when maximizing) or decreased (when minimizing) before the optimal solution would change and this nonbasic variable would become a basic variable. The reduced cost for a basic variable automatically is 0. (Appendix 4.1)

Reoptimization technique A technique for efficiently solving a revised version of the original model by starting from a revised version of the final simplex tableau that yielded the original optimal solution. (Section 4.7)

Row of a simplex tableau A row of numbers to the right of the *Z* column in the simplex tableau. (Section 4.4)

Sensitive parameter A model's parameter is considered sensitive if even a small change in its value can change the optimal solution. (Section 4.7)

Sensitivity analysis Analysis of how sensitive the optimal solution is to the value of each parameter of the model. (Section 4.7)

Shadow price When the right-hand side of a constraint in ≤ form gives the amount available of a certain resource, the shadow price for that resource is the rate at which the optimal value of the objective function could be increased by slightly increasing the amount of this resource being made available. (Section 4.7)

Simplex tableau A table that the tabular form of the simplex method uses to compactly display the system of equations yielding the current BF solution. (Section 4.4)

Slack variable A supplementary variable that gives the slack between the two sides of a functional constraint in ≤ form. (Section 4.2)

Surplus variable A supplementary variable that equals the surplus of the left-hand side over the right-hand side of a functional constraint in ≥ form. (Section 4.6)

Two-phase method A method that the simplex method can use to solve a linear programming problem that is not in our standard form by using phase 1 to find a BF solution for the problem and then proceeding as usual in phase 2. (Section 4.6)

Glossary for Chapter 5

Adjacent CPF solutions Two CPF solutions are adjacent if the line segment connecting them is an *edge of the feasible region* (defined below). (Section 5.1)

Basic feasible (BF) solution A CPF solution that has been augmented by the slack, artificial, and surplus variables that are needed by the simplex method. (Section 5.1)

Basic solution A corner-point solution that has been augmented by the slack, artificial, and surplus variables that are needed by the simplex method. (Section 5.1)

Basic variables The variables in a basic solution whose values are obtained as the simultaneous solution of the system of equations that comprise the functional constraints in augmented form. (Section 5.1)

Basis matrix The matrix whose columns are the columns of constraint coefficients of the basic variables in order. (Section 5.2)

BF solution See *basic feasible solution.*

Constraint boundary A geometric boundary of the solutions that are permitted by the constraint. (Section 5.1)

Constraint boundary equation The equation obtained from a constraint by replacing its ≤, =, or ≥ sign by an = sign. (Section 5.1)

Corner-point feasible (CPF) solution A feasible solution that does not lie on any line segment connecting two other feasible solutions. (Section 5.1)

Corner-point solution A solution of an *n*-variable linear programming problem that lies at the intersection of *n* constraint boundaries. (Section 4.1)

CPF solution See *corner-point feasible solution*.

Defining equations The constraint boundary equations that yield (define) the indicated CPF solution. (Section 5.1)

Degenerate BF solution A BF solution where at least one of the basic variables has a value of zero. (Section 5.1)

Edge of the feasible region For an *n*-variable linear programming problem, an edge of the feasible region is a feasible line segment that lies at the intersection of n-1 constraint boundaries. (Section 5.1)

Hyperplane A "flat" geometric shape in *n*-dimensional space for $n > 3$ that is defined by an equation. (Section 5.1)

Indicating variable Each constraint has an indicating variable that completely indicates (by whether its value is zero) whether that constraint's boundary equation is satisfied by the current solution. (Section 5.1)

Nonbasic variables The variables that are set equal to zero in a basic solution. (Section 5.1)

Glossary for Chapter 6

Allowable range for a right-hand side The range of values for this right-hand side b_i over which the current optimal BF solution (with adjusted values for the basic variables) remains feasible, assuming no change in the other right-hand sides. (Section 6.7)

Allowable range to stay optimal The range of values for a coefficient in the objective function over which the current optimal solution remains optimal, assuming no change in the other coefficients. (Section 6.7)

Complementary slackness A relationship involving each pair of associated variables in a primal basic solution and the complementary dual basic solution whereby one of the variables is a basic variable and the other is a nonbasic variable. (Section 6.3)

Complementary solution Each corner-point or basic solution for the primal problem has a complementary corner-point or basic solution for the dual problem that is defined by the complementary solutions property or complementary basic solutions property. (Section 6.3)

Dual feasible A primal basic solution is said to be dual feasible if the complementary dual basic solution is feasible for the dual problem. (Section 6.3)

Dual problem The linear programming problem that has a dual relationship with the original (primal) linear programming problem of interest according to duality theory. (Section 6.1)

Parametric programming The systematic study of how the optimal solution changes as several of the model's parameters continuously change simultaneously over some intervals. (Section 6.7)

Primal-dual table A table that highlights the correspondence between the primal and dual problems. (Section 6.1)

Primal feasible A primal basic solution is said to be primal feasible if it is feasible for the primal problem. (Section 6.3)

Primal problem The original linear programming problem of interest when using duality theory to define an associated dual problem. (Section 6.1)

Reduced cost The reduced cost for a nonbasic variable measures how much its coefficient in the objective function can be increased (when maximizing) or decreased (when minimizing) before the optimal solution would change and this nonbasic variable would become a basic variable. The reduced cost for a basic variable automatically is 0. (Section 6.7)

Sensible-odd-bizarre method A mnemonic device to remember what the forms of the dual constraints should be. (Section 6.4)

Sensitive parameter A model's parameter is considered sensitive if even a small change in its value can change the optimal solution. (Section 6.6)

Sensitivity analysis Analysis of how sensitive the optimal solution is to the value of each parameter of the model. (Section 6.6)

Shadow price The shadow price for a functional constraint is the rate at which the optimal value of the objective function can be increased by slightly increasing the right-hand side of the constraint. (Section 6.2)

SOB method See *sensible-odd-bizarre method*.

Glossary for Chapter 7

Dual simplex method An algorithm that deals with a linear programming problem as if the simplex method were being applied simultaneously to its dual problem. (Section 7.1)

Gradient The gradient of the objective function is the vector whose components are the coefficients in the objective function. Moving in the direction specified by this vector increases the value of the objective function at the fastest possible rate. (Section 7.4)

Interior-point algorithm An algorithm that generates trial solutions inside the boundary of the feasible region that lead toward an optimal solution. (Section 7.4)

Parametric linear programming An algorithm that systematically determines how the optimal solution changes as several of the model's parameters continuously change simultaneously over some intervals. (Section 7.2)

Projected gradient The projected gradient of the objective function is the projection of the gradient of the objective function onto the feasible region. (Section 7.4)

Upper bound constraint A constraint that specifies a maximum feasible value of an individual decision variable. (Section 7.3)

Upper bound technique A technique that enables the simplex method (and its variants) to deal efficiently with upper-bound constraints in a linear programming model. (Section 7.3)

Glossary for Chapter 8

Assignees The entities (people, machines, vehicles, plants, etc.) that are to perform the tasks when formulating a problem as an assignment problem. (Section 8.3)

Cost table A table that displays all the alternative costs of assigning assignees to tasks in an assignment problem, so the table provides a complete formulation of the problem. (Section 8.3)

Demand at a destination The number of units that need to be received by this destination from the sources. (Section 8.1)

Destinations The receiving centers for a transportation problem. (Section 8.1)

Donor cells Cells in a transportation simplex tableau that reduce their allocations during an iteration of the transportation simplex method. (Section 8.2)

Dummy destination An imaginary destination that is introduced into the formulation of a transportation problem to enable the sum of the supplies from the sources to equal the sum of the demands at the destinations (including this dummy destination). (Section 8.1)

Dummy source An imaginary source that is introduced into the formulation of a transportation problem to enable the sum of the supplies from the sources (including this dummy source) to equal the sum of the demands at the destinations. (Section 8.1)

Hungarian algorithm An algorithm that is designed specifically to solve assignment problems very efficiently. (Section 8.4)

Parameter table A table that displays all the parameters of a transportation problem, so the table provides a complete formulation of the problem. (Section 8.2)

Recipient cells Cells in a transportation simplex tableau that receive additional allocations during an iteration of the transportation simplex method. (Section 8.2)

Sources The supply center for a transportation problem. (Section 8.1)

Supply from a source The number of units to be distributed from this source to the destinations. (Section 8.1)

Tasks The jobs to be performed by the assignees when formulating a problem as an assignment problem. (Section 8.3)

Transportation simplex method A streamlined version of the simplex method for solving transportation problems very efficiently. (Section 8.2)

Transportation simplex tableau A table that is used by the transportation simplex method to record the relevant information at each iteration. (Section 8.2)

Glossary for Chapter 9

Activity A distinct task that needs to be performed as part of a project. (Section 9.8)

Activity-on-arc (AOA) project network A project network where each activity is represented by an arc. (Section 9.8)

Activity-on-node (AON) project network A project network where each activity is represented by a node and the arcs show the precedence relationships between the activities. (Section 9.8)

Arc A channel through which flow may occur from one node to another. (Section 9.2)

Arc capacity The maximum amount of flow that can be carried on a directed arc. (Section 9.2)

Augmenting path A directed path from the source to the sink in the residual network of a maximum flow problem such that every arc on this path has strictly positive residual capacity. (Section 9.5)

Augmenting path algorithm An algorithm that is designed specifically to solve maximum flow problems very efficiently. (Section 9.5)

Basic arc An arc that corresponds to a basic variable in a basic solution at the current iteration of the network simplex method. (Section 9.7)

Connected Two nodes are said to be connected if the network contains at least one undirected path between them. (Section 9.2)

Connected network A network where every pair of nodes is connected. (Section 9.2)

Conservation of flow The condition at a node where the amount of flow out of the node equals the amount of flow into that node. (Section 9.2)

CPM An acronym for critical path method, a technique for assisting project managers with carrying out their responsibilities. (Section 9.8)

CPM method of time-cost trade-offs A method of investigating the trade-off between the total cost of a project and its duration when various levels of crashing are used to reduce the duration. (Section 9.8)

Crash point The point on the time-cost graph for an activity that shows the time (duration) and cost when the activity is fully crashed; that is, the activity is fully expedited with no cost spared to reduce its duration as much as possible. (Section 9.8)

Crashing an activity Taking special costly measures to reduce the duration of an activity below its normal value. (Section 9.8)

Crashing the project Crashing a number of activities to reduce the duration of the project below its normal value. (Section 9.8)

Critical path The longest path through a project network, so the activities on this path are the critical bottleneck activities where any delays in their completion must be avoided to prevent delaying project completion. (Section 9.8)

Cut Any set of directed arcs containing at least one arc from every directed path from the source to the sink of a maximum flow problem. (Section 9.5)

Cut value The sum of the arc capacities of the arcs (in the specified direction) of the cut. (Section 9.5)

Cycle A path that begins and ends at the same node. (Section 9.2)

Demand node A node where the net amount of flow generated (outflow minus inflow) is a fixed negative amount, so flow is absorbed there. (Section 9.2)

Destination The node at which travel through the network is assumed to end for a shortest-path problem. (Section 9.3)

Directed arc An arc where flow through the arc is allowed in only one direction. (Section 9.2)

Directed network A network whose arcs are all directed arcs. (Section 9.2)

Directed path A directed path from node i to node j is a sequence of connecting arcs whose direction (if any) is toward node j. (Section 9.2)

Feasible spanning tree A spanning tree whose solution from the node constraints also satisfies all the nonnegativity constraints and arc capacity constraints for the flows through the arcs. (Section 9.7)

Length of a link or an arc The number (typically a distance, a cost, or a time) associated with a link or arc for either a shortest-path problem or a minimum spanning tree problem. (Sections 9.3 and 9.4)

Length of a path through a project network The sum of the (estimated) durations of the activities on the path. (Section 9.8)

Link An alternative name for *undirected arc*, defined below. (Section 9.2)

Marginal cost analysis A method of using the marginal cost of crashing individual activities on the current critical path to determine the least expensive way of reducing project duration to a desired level. (Section 9.8)

Minimum spanning tree One among all spanning trees that minimizes the total length of all the links in the tree. (Section 9.4)

Network simplex method A streamlined version of the simplex method for solving minimum cost flow problems very efficiently. (Section 9.7)

Node A junction point of a network, shown as a labeled circle. (Section 9.2)

Nonbasic arc An arc that corresponds to a nonbasic variable in a basic solution at the current iteration of the network simplex method. (Section 9.7)

Normal point The point on the time-cost graph for an activity that shows the time (duration) and cost of the activity when it is performed in the normal way. (Section 9.8)

Origin The node at which travel through the network is assumed to start for a shortest-path problem. (Section 9.3)

Path A path between two nodes is a sequence of distinct arcs connecting these nodes when the direction (if any) of the arcs is ignored. (Section 9.2)

Path through a project network One of the routes following the arcs from the start node to the finish node. (Section 9.8)

PERT An acronym for program evaluation and review technique, a technique for assisting project managers with carrying out their responsibilities. (Section 9.8)

PERT/CPM The merger of the two techniques originally know as PERT and CPM. (Section 9.8)

Project duration The total time required for the project. (Section 9.8)

Project network A network used to visually display a project. (Section 9.8)

Residual capacity The remaining arc capacities for assigning additional flows after some flows have been assigned to the arcs by the augmenting path algorithm for a maximum flow problem. (Section 9.5)

Residual network The network that shows the remaining arc capacities for assigning additional flows after some flows have been assigned to the arcs by the augmenting path algorithm for a maximum flow problem. (Section 9.5)

Reverse arc An imaginary arc that the network simplex method might introduce to replace a real arc and allow flow in the opposite direction temporarily. (Section 9.7)

Sink The node for a maximum flow problem at which all flow through the network terminates. (Section 9.5)

Source The node for a maximum flow problem at which all flow through the network originates. (Section 9.5)

Spanning tree A connected network for all *n* nodes of the original network that contains no undirected cycles. (Section 9.2)

Spanning tree solution A basic solution for a minimum cost flow problem where the basic arcs form a spanning tree and the values of the corresponding basic variables are obtained by solving the node constraints. (Section 9.7)

Supply node A node where the amount of flow generated (outflow minus inflow) is a fixed positive amount. (Section 9.2)

Transshipment node A node where the amount of flow out equals the amount of flow in. (Section 9.2)

Transshipment problem A special type of minimum cost flow problem where there are no capacity constraints on the arcs. (Section 9.6)

Tree A connected network (for some subset of the *n* nodes of the original network) that contains no undirected cycles. (Section 9.2)

Undirected arc An arc where flow through the arc is allowed to be in either direction. (Section 9.2)

Undirected network A network whose arcs are all undirected arcs. (Section 9.2)

Undirected path An undirected path from node *i* to node *j* is a sequence of connecting arcs whose direction (if any) can be either toward or away from node *j*. (Section 9.2)

Glossary for Chapter 10

Curse of dimensionality The condition that the computational effort for dynamic programming tends to "blow up" rapidly when additional state variables need to be introduced to describe the state of the system at each stage. (Section 10.3)

Decision tree A graphical display of all the possible states and decisions at all the stages of a dynamic programming problem. (Section 10.4)

Distribution of effort problem A type of dynamic programming problem where there is just one kind of resource that is to be allocated to a number of activities. (Section 10.3)

Optimal policy The optimal specification of the policy decisions at the respective stages of a dynamic programming problem. (Section 10.2)

Policy decision A policy regarding what decision should be made at a particular stage of a dynamic programming problem, where this policy specifies the decision as a function of the possible states that the system can be in at that stage. (Section 10.2)

Principle of optimality A basic property that the optimal immediate decision at each stage of a dynamic programming problem depends on only the current state of the system and not on the history of how the system reached that state. (Section 10.2)

Recursive relationship An equation that enables solving for the optimal policy for each stage of a dynamic programming problem in terms of the optimal policy for the following stage. (Section 10.2)

Stages A dynamic programming problem is divided into stages, where each stage involves making one decision from the sequence of interrelated decisions that comprise the overall problem. (Section 10.2)

State variable A variable that gives the state of the system at a particular stage of a dynamic programming problem. (Section 10.3)

States The various possible conditions of the system at a particular stage of a dynamic programming problem. (Section 10.2)

Glossary for Chapter 11

All-different constraint A global constraint that constraint programming uses to specify that all the variables in a given set must have different values. (Section 11.9)

Auxiliary binary variable A binary variable that is introduced into the model, not to represent a yes-or-no decision, but simply to help formulate the model as a (pure or mixed) BIP problem. (Section 11.3)

Binary integer programming The type of integer programming where all the integer-restricted variables are further restricted to be binary variables. (Section 11.2)

Binary representation A representation of a bounded integer variable as a linear function of some binary variables. (Section 11.3)

Binary variable A variable that is restricted to the values of 0 and 1. (Introduction)

BIP An abbreviation for *binary integer programming*, defined above.

Bounding A basic step in a branch-and-bound algorithm that bounds how good the best solution in a subset of feasible solutions can be. (Section 11.6)

Branch-and-cut algorithm A type of algorithm for integer programming that combines automatic problem preprocessing, the generation of cutting planes, and clever branch-and-bound techniques. (Section 11.8)

Branching A basic step in a branch-and-bound algorithm that partitions a set of feasible solutions into subsets, perhaps by setting a variable at different values. (Section 11.6)

Branching variable A variable that the current iteration of a branch-and-bound algorithm uses to divide a subproblem into smaller subproblems by assigning alternative values to the variable. (Section 11.6)

Constraint programming A technique for formulating complicated kinds of constraints on integer variables and then efficiently finding feasible solutions that satisfy all these constraints. (Section 11.9)

Constraint propagation The process used by constraint programming for using current constraints to imply new constraints. (Section 11.9)

Contingent decision A yes-or-no decision is a contingent decision if it can be yes only if a certain other yes-or-no decision is yes. (Section 11.1)

Cut An alternative name for *cutting plane*, defined below. (Section 11.8)

Cutting plane A cutting plane for any integer programming problem is a new functional constraint that reduces the feasible region for the LP relaxation without eliminating any feasible solutions for the integer programming problem. (Section 11.8)

Descendant A descendant of a subproblem is a new smaller subproblem that is created by branching on this subproblem and then perhaps branching further through subsequent "generations." (Section 11.6)

Domain reduction The process used by constraint programming for eliminating possible values for individual variables. (Section 11.9)

Either-or constraints A pair of constraints such that one of them (either one) must be satisfied but the other one can be violated. (Section 11.3)

Element constraint A global constraint that constraint programming uses to look up a cost or profit associated with an integer variable. (Section 11.9)

Enumeration tree An alternative name for *solution tree*, defined below. (Section 11.6)

Exponential growth An exponential growth in the difficulty of a problem refers to an unusually rapid growth in the difficulty as the size of the problem increases. (Section 11.5)

Fathoming A basic step in a branch-and-bound algorithm that uses fathoming tests to determine if a subproblem can be dismissed from further consideration. (Section 11.6)

Fixed-charge problem A problem where a fixed charge or setup cost is incurred when undertaking an activity. (Section 11.3)

General integer variable A variable that is restricted only to have *any* nonnegative integer value that also is permitted by the functional constraints. (Section 11.7)

Global constraint A constraint that succinctly expresses a global pattern in the allowable relationship between multiple variables. (Section 11.9)

Incumbent The best feasible solution found so far by a branch-and-bound algorithm. (Section 11.6)

IP An abbreviation for *integer programming*. (Introduction)

Lagrangian relaxation A relaxation of an integer programming problem that is obtained by deleting the entire set of functional constraints and then modifying the objective function in a certain way. (Section 11.6)

LP relaxation The linear programming problem obtained by deleting from the current integer programming problem the constraints that require variables to have integer values. (Section 11.5)

Minimum cover A minimum cover of a constraint refers to a group of binary variables that satisfy certain conditions with respect to the constraint during a procedure for generating cutting planes. (Section 11.8)

MIP An abbreviation for *mixed integer programming*, defined below. (Introduction)

Mixed integer programming The type of integer programming where only some of the variables are required to have integer values. (Section 11.7)

Mutually exclusive alternatives A group of alternatives where choosing any one alternative excludes choosing any of the others. (Section 11.1)

Problem preprocessing The process of reformulating a problem to make it easier to solve without eliminating any feasible solutions. (Section 11.8)

Recurring branching variable A variable that becomes a branching variable more than once during the course of a branch-and-bound algorithm. (Section 11.7)

Redundant constraint A constraint that automatically is satisfied by solutions that satisfy all the other constraints. (Section 11.8)

Relaxation A relaxation of a problem is obtained by deleting a set of constraints from the problem. (Section 11.6)

Set covering problem A type of pure BIP problem where the objective is to determine the least costly combination of activities that collectively possess each of a number of characteristics at least once. (Section 11.4)

Set partitioning problem A variation of a set covering problem where the selected activities must collectively possess each of a number of characteristics exactly once. (Section 11.4)

Solution tree A *tree* (as defined in Sec. 9.2) that records the progress of a branch-and-bound algorithm in partitioning an integer programming problem into smaller and smaller subproblems. (Section 11.6)

Subproblem A portion of another problem that is obtained by eliminating a portion of the feasible region, perhaps by fixing the value of one of the variables. (Section 11.6)

Yes-or-no decision A decision whose only possible choices are (1) yes, go ahead with a certain option, or (2) no, decline this option. (Section 11.2)

Glossary for Chapter 12

Bisection method One type of search procedure for solving one-variable unconstrained optimization problems where the objective function (assuming maximization) is a concave function, or at least a unimodal function. (Section 12.4)

Complementarity constraint A special type of constraint in the complementarity problem (and elsewhere) that requires at least one variable in each pair of associated variables to have a value of 0. (Sections 12.3 and 12.7)

Complementarity problem A special type of problem where the objective is to find a feasible solution for a certain set of constraints. (Section 12.3)

Complementary variables A pair of variables such that only one of the variables (either one) can be nonzero. (Section 12.7)

Concave function A function that is always "curving downward" (or not curving at all), as defined further in Appendix 2. (Section 12.2)

Convex function A function that is always "curving upward" (or not curving at all), as defined further in Appendix 2. (Section 12.2)

Convex programming problems Nonlinear programming problems where the objective function (assuming maximization) is a concave function and the constraint functions (assuming a $\leq$ form) are convex functions. (Sections 12.3 and 12.9)

Convex set A set of points such that, for each pair of points in the collection, the entire line segment joining these two points is also in the collection. (Section 12.2)

Fractional programming problems A special type of nonlinear programming problem where the objective function is in the form of a fraction that gives the ratio of two functions. (Section 12.3)

Frank-Wolfe algorithm An important example of sequential-approximation algorithms for convex programming. (Section 12.9)

Genetic algorithm A type of algorithm for nonconvex programming that is based on the concepts of genetics, evolution, and survival of the fittest. The Evolutionary Solver within the Premium Solver Excel add-in uses this kind of algorithm. (Sections 12.10 and 13.4)

Geometric programming problems A special type of nonlinear programming problem that fits many engineering design problems, among others. (Section 12.3)

Global maximum (or minimum) A feasible solution that maximizes (or minimizes) the value of the objective function over the entire feasible region. (Section 12.2)

Global optimizer A type of software package that implements an algorithm that is designed to find a globally optimal solution for various kinds of nonconvex programming problems. (Section 12.10)

Gradient algorithms Convex programming algorithms that modify the gradient search procedure to keep the search procedure from penetrating any constraint boundary. (Section 12.9)

Gradient search procedure A type of search procedure that uses the gradient of the objective function to solve multivariable unconstrained optimization problems where the objective function (assuming maximization) is a concave function. (Section 12.5)

Karush-Kuhn-Tucker conditions For a nonlinear programming problem with differentiable functions that satisfy certain regularity conditions, the Karush-Kuhn-Tucker conditions provide the necessary conditions for a solution to be optimal. These necessary conditions also are sufficient in the case of a convex programming problem. (Section 12.6)

KKT conditions An abbreviation for *Karush-Kuhn-Tucker conditions*, defined above. (Section 12.6)

Linear complementarity problem A linear form of the complementarity problem. (Section 12.3)

Linearly constrained optimization problems Nonlinear programming problems where all the constraint functions (but not the objective function) are linear. (Section 12.3)

Local maximum (or minimum) A feasible solution that maximizes (or minimizes) the value of the objective function within a local neighborhood of that solution. (Section 12.2)

Modified simplex method An algorithm that adapts the simplex method so it can be applied to quadratic programming problems. (Section 12.7)

Newton's method A traditional type of search procedure that uses a quadratic approximation of the objective function to solve unconstrained optimization problems where the objective function (assuming maximization) is a concave function. (Sections 12.4 and 12.5)

Nonconvex programming problems Nonlinear programming problems that do not satisfy the assumptions of convex programming. (Sections 12.3 and 12.10)

Quadratic programming problems Nonlinear programming problems where all the constraint functions are linear and the objective function is quadratic. This quadratic function also is commonly assumed to be a concave function (when maximizing) or a convex function (when minimizing). (Sections 12.3 and 12.7)

Quasi-Newton methods Convex programming algorithms that extend an approximation of Newton's method for unconstrained optimization to deal instead with constrained optimization problems. (Section 12.5)

Restricted-entry rule A rule used by the modified simplex method when choosing an entering basic variable that prevents two complementary variables from both being basic variables. (Section 12.7)

Separable function A function where each term involves just a single variable, so that the function is separable into a sum of functions of individual variables. (Sections 12.3 and 12.8)

Sequential-approximation algorithms Convex programming algorithms that replace the nonlinear objective function by a succession of linear or quadratic approximations. (Section 12.9)

Sequential unconstrained algorithms Convex programming algorithms that convert the original constrained optimization problem to a sequence of unconstrained optimization problems whose optimal solutions converge to an optimal solution for the original problem. (Section 12.9)

Sequential unconstrained minimization technique A classic algorithm within the category of sequential-approximation algorithms. (Section 12.9)

SUMT An acronym for *sequential unconstrained minimization technique*, defined above. (Section 12.9)

Unconstrained optimization problems Optimization problems that have no constraints on the values of the variables. (Sections 12.3-12.5)

Glossary for Chapter 13

Children The new trial solutions generated by each pair of parents during an iteration of a genetic algorithm. (Section 13.4)

Gene One of the binary digits that defines a trial solution in base 2 for a genetic algorithm. (Section 13.4)

Genetic algorithm A type of metaheuristic that is based on the concepts of genetics, evolution, and survival of the fittest. (Section 13.4)

Heuristic method A procedure that is likely to discover a very good feasible solution, but not necessarily an optimal solution, for the specific problem being considered. (Introduction)

Local improvement procedure A procedure that searches in the neighborhood of the current trial solution to find a better trial solution. (Section 13.1)

Local search procedure A procedure that operates like a local improvement procedure except that it may not require that each new trial solution must be better than the preceding trial solution. (Section 13.2)

Metaheuristic A general solution method that provides both a general structure and strategy guidelines for developing a specific heuristic method to fit a particular kind of problem. (Introduction and Section 13.1)

Mutation A random event that enables a child to acquire a feature that is not possessed by either parent during an iteration of a genetic algorithm. (Section 13.4)

Parents A pair of trial solutions used by a genetic algorithm to generate new trial solutions. (Section 13.4)

Population The set of trial solutions under consideration during an iteration of a genetic algorithm. (Section 13.4)

Random number A random observation from a uniform distribution between 0 and 1. (Section 13.3)

Simulated annealing A type of metaheuristic that is based on the analogy to a physical annealing process. (Section 13.3)

Steepest ascent/mildest descent approach An algorithmic approach that seeks the greatest possible improvement at each iteration but also accepts the best available non-improving move when an improving move is not available. (Section 13.2)

Sub-tour reversal A method for adjusting the sequence of cities visited in the current trial solution for a traveling salesman problem by selecting a subsequence of the cities and reversing the order in which that subsequence of cities is visited. (Section 13.1)

Sub-tour reversal algorithm An algorithm for the traveling salesman problem that is based on performing a series of sub-tour reversals that improve the current trial solution each time. (Section 13.1)

Tabu list A record of the moves that currently are forbidden by a tabu search algorithm. (Section 13.2)

Tabu search A type of metaheuristic that allows non-improving moves but also incorporates short-term memory of the past search by using a tabu list to discourage cycling back to previously considered solutions. (Section 13.2)

Temperature schedule The schedule used by a simulated annealing algorithm to adjust the tendency to accept the current candidate to be the next trial solution if this candidate is not an improvement on the current trial solution. (Section 13.3)

Traveling salesman problem A classic type of combinatorial optimization problem that can be described in terms of a salesman seeking the shortest route for visiting a number of cities exactly once each. (Section 13.1)

Glossary for Chapter 14

Cooperative game A nonzero-sum game where preplay discussions and binding agreements are permitted. (Section 14.6)

Dominated strategy A strategy is dominated by a second strategy if the second strategy is always at least as good (and sometimes better) regardless of what the opponent does. (Section 14.2)

Fair Game A game that has a value of 0. (Section 14.2)

Graphical solution procedure A graphical method of solving a two-person, zero-sum game with mixed strategies such that, after dominated strategies are eliminated, one of the two players has only two pure strategies. (Section 14.4)

Infinite game A game where the players have an infinite number of pure strategies available to them. (Section 14.6)

Minimax criterion The criterion that says to select a strategy that minimizes a player's maximum expected loss. (Sections 14.2 and 14.3)

Mixed strategy A plan for using a probability distribution to determine which of the original strategies will be used. (Section 14.3)

Non-cooperative game A nonzero-sum game where there is no preplay communication between the players. (Section 14.6)

Nonzero-sum game A game where the sum of the payoffs to the players need not be 0 (or any other fixed constant). (Section 14.6)

n-person game A game where more than two players may participate. (Section 14.6)

Payoff table A table that shows the gain (positive or negative) for player 1 that would result from each combination of strategies for the two players in a two-person, zero-sum game. (Section 14.1)

Pure strategy One of the original strategies (as opposed to a mixed strategy) in the formulation of a two-person, zero-sum game. (Section 14.3)

Saddle point An entry in a payoff table that is both the minimum in its row and the maximum of its column. (Section 14.2)

Stable solution A solution for a two-person, zero-sum game where neither player has any motive to consider changing strategies, either to take advantage of his opponent or to prevent the opponent of taking advantage of him. (Section 14.2)

Strategy A predetermined rule that specifies completely how one intends to respond to each possible circumstance at each stage of a game. (Section 14.1)

Two-person, constant-sum game A game with two players where the sum of the payoffs to the two players is a fixed constant (positive or negative) regardless of which combination of strategies is selected. (Section 14.6)

Two-person zero-sum game A game with two players where one player wins whatever the other one loses, so that the sum of their net winnings is zero. (Introduction and Section 14.1)

Unstable solution A solution for a two-person, zero-sum game where each player has a motive to consider changing his strategy once he deduces his opponent's strategy. (Section 14.2)

Value of the game The expected payoff to player 1 when both players play optimally in a two-person, zero-sum game. (Sections 14.2 and 14.3)

Glossary for Chapter 15

Alternatives The options available to the decision maker for the decision under consideration. (Section 15.2)

Backward induction procedure A procedure for solving a decision analysis problem by working backward through its decision tree. (Section 15.4)

Bayes' decision rule A popular criterion for decision making that uses probabilities to calculate the expected payoff for each decision alternative and then chooses the one with the largest expected payoff. (Section 15.2)

Bayes' theorem A formula for calculating a posterior probability of a state of nature. (Section 15.3)

Branch A line emanating from a node in a decision tree. (Section 15.4)

Crossover point When plotting the lines giving the expected payoffs of two decision alternatives versus the prior probability of a particular state of nature, the crossover point is the point where the two lines intersect so that the decision is shifting from one alternative to the other. (Section 15.2)

Decision conferencing A process used for group decision making. (Section 15.7)

Decision maker The individual or group responsible for making the decision under consideration. (Section 15.2)

Decision node A point in a decision tree where a decision needs to be made. (Section 15.4)

Decision tree A graphical display of the progression of decisions and random events to be considered. (Section 15.4)

Decreasing marginal utility for money The situation where the slope of the utility function decreases as the amount of money increases. (Section 15.6)

Event node A point in a decision tree where a random event will occur. (Section 15.4)

Expected value of experimentation (EVE) The maximum increase in the expected payoff that could be obtained from performing experimentation (excluding the cost of the experimentation). (Section 15.3)

Expected value of perfect information (EVPI) The increase in the expected payoff that could be obtained if it were possible to learn the true state of nature. (Section 15.3)

Exponential utility function A utility function that is designed to fit a risk-averse individual. (Section 15.6)

Increasing marginal utility for money The situation where the slope of the utility function increases as the amount of money increases. (Section 15.6)

Influence diagram A diagram that complements the decision tree for representing and analyzing decision analysis problems. (Section 15.7)

Maximum likelihood criterion A criterion for decision making with probabilities that focuses on the most likely state of nature. (Section 15.2)

Maximum payoff criterion A very pessimistic decision criterion that does not use prior probabilities and simply chooses the decision criterion that provides the best guarantee for its minimum possible payoff. (Section 15.2)

Node A junction point in a decision tree. (Section 15.4)

Payoff A quantitative measure of the outcome from a decision alternative and a state of nature. (Section 15.2)

Payoff table A table giving the payoff for each combination of a decision alternative and a state of nature. (Section 15.2)

Plot An option provided by SensIt for generating a graph that shows how an output cell varies for different values of a single data cell. (Section 15.5)

Posterior probabilities Revised probabilities of the states of nature after doing a test or survey to improve the prior probabilities. (Section 15.3)

Prior distribution The probability distribution consisting of the prior probabilities of the states of nature. (Section 15.2)

Prior probabilities The estimated probabilities of the states of nature prior to obtaining additional information through a test or survey. (Section 15.2)

Probability tree diagram A diagram that is helpful for calculating the posterior probabilities of the states of nature. (Section 15.3)

Risk-averse individual An individual who has a decreasing marginal utility for money. (Section 15.6)

Risk-neutral individual An individual whose utility function for money is proportional to the amount of money involved. (Section 15.6)

Risk-seeking individual An individual who has an increasing marginal utility for money. (Section 15.6)

Sensitivity analysis The study of how other plausible values for the probabilities of the states of nature (or for the payoffs) would affect the recommended decision alternative. (Section 15.5)

Spider graph A graph that provides helpful comparisons for sensitivity analysis. (Section 15.5)

States of nature The possible outcomes of the random factors that affect the payoff that would be obtained from a decision alternative. (Section 15.2)

Tornado diagram A diagram that organizes the data from sensitivity analysis in a readily understandable way. (Section 15.5)

Utility The utility of an outcome measures the true value to the decision maker of that outcome. (Section 15.6)

Utility function for money, $u(M)$ A plot of utility versus the amount of money M being received. (Section 15.6)

Glossary for Chapter 16

Absorbing state A state of a Markov chain such that, upon entering this state, the process never will leave this state again. (Section 16.4)

Accessible A state is said to be accessible from another state if there is a strictly positive probability of eventually reaching this state when starting from the latter state. (Section 16.4)

Aperiodic state A state of a discrete time Markov chain that has a period of 1. (Section 16.4)

Balance equations The equations for a continuous time Markov chain that specify that the rate at which the process enters each state must be in balance with (equal to) the rate at which the process leaves that state. (Section 16.8)

Chapman-Kolmogorov equations A set of equations that enable solving for the *n*-step transition probabilities for a discrete time Markov chain. (Section 16.3)

Classes of states The states of a Markov chain may be partitioned into one or more separate classes such that those states that communicate with each other are in the same class. (Section 16.4)

Communicating states States that are accessible from each other. (Section 16.4)

Continuous time Markov chain A Markov chain with a continuous time parameter so the evolution of the process is being observed continuously over time. (Section 16.8)

Continuous time transition probability function A function that specifies the transition probabilities for a continuous time Markov chain. (Section 16.8)

Discrete time Markov chain A Markov chain with a discrete time parameter t so the evolution of the process is being observed at discrete points in time labeled as $t = 0, 1, 2, \cdots$. (Section 16.2)

Ergodic Markov chain A discrete time Markov chain where all its states are ergodic states. (Section 16.4)

Ergodic state A recurrent state of a discrete time Markov chain that also is aperiodic. (Section 16.4)

Expected recurrence time The expected value (in the statistical sense) of the recurrence time of a state of a discrete time Markov chain. (Section 16.6)

First passage time The first passage time in going from state *i* to state *j* in a discrete time Markov chain is the number of transitions made by the process in going from state *i* to state *j* for the first time. (Section 16.6)

Irreducible Markov chain A Markov chain such that all the states communicate. (Section 16.4)

Markov chain A stochastic process that has the Markovian property. (Sections 16.2 and 16.8)

Markovian property A stochastic process has the Markovian property if the conditional probability of any future event, given any past events and the present state, is independent of the past events and depends only upon the present state. (Sections 16.2 and 16.8)

***n*-step transition matrix** A matrix that gives all the *n*-step transition probabilities for a specific value of *n* for a discrete time Markov chain. (Section 16.2)

***n*-step transition probability** A transition probability where the number of steps involved is specified to be *n*. (Section 16.2)

Period of a state Upon entering a state of a discrete time Markov chain, the number of steps until the process can be in this state again is a strictly positive integer multiple of the period of the state. (Section 16.4)

Probability of absorption Given that a Markov chain starts in a particular state, the probability of ever going to a certain absorbing state is called the probability of absorption into that absorbing state. (Section 16.7)

Random walk A discrete time Markov chain with the property that a single transition from any state always either remains in that state or moves to one of its two adjacent states. (Section 16.7)

Recurrence time The recurrence time for a state of a discrete time Markov chain is the first passage time in going from that state back to that same state. (Section 16.6)

Recurrent state A state of a Markov chain such that, upon entering this state, the process definitely will return to this state again. (Section 16.4)

States The mutually exclusive categories for the current status of a stochastic process. (Sections 16.1 and 16.2)

Stationary transition probabilities Transition probabilities that do not change over time. (Sections 16.2 and 16.8)

Steady-state equations The system of equations whose unique solution gives the steady-state probabilities of a Markov chain. (Sections 16.5 and 16.8)

Steady-state probabilities Probabilities of the state of a Markov chain that are independent of the probability distribution of the initial state and that will remain unchanged over time. (Sections 16.5 and 16.8)

Stochastic process A process that evolves probabilistically over time. (Section 16.1)

Transient state A state of a Markov chain such that, upon entering this state, the process may never return to this state again. (Section 16.4)

Transition intensities The transition rates for a continuous time Markov chain. (Section 16.8)

Transition matrix A matrix that gives all the transition probabilities ($n = 1$) for a discrete time Markov chain. (Section 16.2)

Transition probability The conditional probability that a Markov chain will be in a particular state *n* steps from now, given its current state. (If *n* is not specified, it is assumed to be 1.) (Sections 16.2 and 16.8)

Glossary for Chapter 17

Balance equation An equation for a particular state of a birth-and-death process that expresses the principle that the mean entering rate for that state must equal its mean leaving rate. (Section 17.5)

Balking An arriving customer who refuses to enter a queueing system because the queue is too long is said to be balking. (Section 17.2)

Birth An increase of 1 in the state of a birth-and-death process. (Section 17.5)

Birth-and-death process A special type of continuous time Markov chain where the only possible changes in the current state of the system are an increase of 1 (a birth) or a decrease of 1 (a death). (Section 17.5)

Calling population The population of potential customers that might need to come to a queueing system. (Section 17.2)

Commercial service system A queueing system where a commercial organization provides a service to customers from outside the organization. (Section 17.3)

Customers A generic term that refers to whichever kind of entity (people, vehicles, machines, items, etc.) is coming to the queueing system to receive service. (Section 17.2)

Death A decrease of 1 in the state of a birth-and-death process. (Section 17.5)

Erlang distribution A common service-time distribution whose shape parameter *k* specifies the amount of variability in the service times. (Sections 17.2 and 17.7)

Exponential distribution The most popular choice for the probability distribution of both interarrival times and service times for a queueing system. (Sections 17.4 and 17.6)

Finite calling population A calling population whose size is so limited that the mean arrival rate to the queueing system is significantly affected by the number of customers that are already in the queueing system. (Sections 17.2 and 17.6)

Finite queue A queue that can hold only a limited number of customers. (Sections 17.2 and 17.6)

Hyperexponential distribution A distribution occasionally used for either interarrival times or service times. Its key characteristic is that even though only nonnegative values are allowed, its standard deviation actually is larger than its mean. (Section 17.7)

Infinite queue A queue that can hold an essentially unlimited number of customers. (Section 17.2)

Input source The stochastic process that generates the customers arriving at a queueing system. (Section 17.2)

Interarrival time The elapsed time between consecutive arrivals to a queueing system. (Section 17.2)

Internal service system A queueing system where the customers receiving service are internal to the organization providing the service. (Section 17.3)

Jackson network One special type of queueing network that has a product form solution. (Section 17.9)

Lack of memory property When referring to arrivals, this property is that the remaining time until the next arrival is completely uninfluenced by when the last arrival occurred. Also called the Markovian property. (Section 17.4)

Little's formula The formula $L = \lambda W$, or $L_q = \lambda W_q$. (Section 17.2)

Mean arrival rate The expected number of arrivals to a queueing system per unit time. (Section 17.2)

Mean service rate The mean service rate for a server is the expected number of customers that it can serve per unit time when working continuously. The term also can be applied to a group of servers collectively. (Section 17.2)

Nonpreemptive priorities Priorities for selecting the next customer to begin service when a server becomes free, without affecting customers who already have begun service. (Section 17.8)

Number of customers in the queue The number of customers who are waiting for service to begin. Also referred to as the queue length. (Section 17.2)

Number of customers in the system The total number of customers in the queueing system, either waiting for service to begin or currently being served. (Section 17.2)

Phase-type distributions A family of distributions obtained by breaking down the total time into a number of phases having exponential distributions. Occasionally used for either interarrival times or service times. (Section 17.7)

Poisson input process A stochastic process for counting the number of customers arriving to a queueing system that is a Poisson process. (Section 17.4)

Poisson process A process where the number of events (e.g., arrivals) that have occurred has a Poisson distribution with a mean that is proportional to the elapsed time. (Section 17.4)

Pollaczek-Khintchine formula The equation for L_q (or W_q) for the $M/G/1$ model. (Section 17.7)

Preemptive priorities Priorities for serving customers that include ejecting the lowest priority customer being served back into the queue in order to serve a higher priority customer that has just entered the queueing system. (Section 17.8)

Priority classes Categories of customers that are given different priorities for receiving service. (Section 17.8)

Product form solution A solution for the joint probability of the number of customers at the respective facilities of a queueing network that is just the product of the probabilities of the number at each facility considered independently of the others. (Section 17.9)

Queue The waiting line in a queueing system. The queue does not include customers who are already being served. (Section 17.2)

Queue discipline The rule for determining the order in which members of the queue are selected to begin service. (Section 17.2)

Queue length See *number of customers in the queue*. (Section 17.2)

Queueing network A network of service facilities where each customer must receive service at some or all of these facilities. (Section 17.9)

Queueing system A place where customers receive some kind of service from a server, perhaps after waiting in a queue. (Section 17.2)

Reneging A customer in the queueing system who becomes impatient and leaves before being served is said to be reneging. (Section 17.5)

Server An entity that is serving the customers coming to a queueing system. (Section 17.2)

Service cost The cost associated with providing the servers in a queueing system. (Section 17.10)

Service mechanism The service facility or facilities where service is provided to customers in a queueing system. (Section 17.2)

Service time The elapsed time from the beginning to the end of a customer's service. (Section 17.2)

Social service system A queueing system which is providing a social service. (Section 17.3)

Steady-state condition The condition where the probability distribution of the number of customers in the queueing system is staying the same over time. (Section 17.2)

Transient condition The condition where the probability distribution of the number of customers in the queueing system currently is shifting as time goes on. (Section 17.2)

Transportation service system A queueing system involving transportation, so that either the customers or the server(s) are vehicles. (Section 17.3)

Utilization factor The average fraction of time that the servers are being utilized serving customers. (Section 17.2)

Waiting cost The cost associated with making customers wait in a queueing system. (Section 17.10)

Waiting time in the queue The elapsed time that an individual customer spends in the queue waiting for service to begin. (Section 17.2)

Waiting time in the system The elapsed time that an individual customer spends in the queueing system both before service begins and during service. (Section 17.2)

Glossary for Chapter 18

ABC control method A method of managing a multiproduct inventory system that begins by dividing the products into a high-priority (A) group, a medium-priority (B) group, and a low-priority (C) group. (Section 18.8)

Assembly system A multiechelon inventory system where some installations have multiple immediate predecessors in the preceding echelon. (Section 18.5)

Backlogging The situation where excess demand is not lost but instead is held until it can be satisfied when the next normal delivery replenishes the inventory. (Section 18.2)

Computerized inventory system A system where each addition to inventory and each sale causing a withdrawal are recorded electronically, so that the current inventory level always is in the computer. (Section 18.6)

Continuous review A continuous monitoring of the current inventory level. (Section 18.2)

Demand The demand for a product in inventory is the number of units that will need to be withdrawn from inventory for some use (e.g., sales) during a specific period. (Introduction)

Dependent demand Demand for a product that depends on the demand for other products. (Section 18.3)

Discount factor The amount by which a cash flow 1 year hence should be multiplied to calculate its net present value. (Section 18.2)

Discount rate The rate at which future income over time loses its current value because of the time value of money. (Section 18.2)

Distribution system A multiechelon inventory system where an installation might have multiple immediate successors in the next echelon. (Section 18.5)

Echelon A stage at which inventory is held in the progression of units through a multistage inventory system. (Section 18.5)

Echelon stock The stock of an item that is physically on hand at an installation plus the stock of the same item that already is downstream at subsequent echelons of the system. (Section 18.5)

Economic order quantity model A standard deterministic continuous-review inventory model with a constant demand rate so that an economic quantity is ordered periodically to replenish inventory. (Section 18.3)

EOQ model An abbreviation of economic order quantity model. (Section 18.3)

Holding cost The total cost associated with the storage of inventory, including the cost of capital tied up, space, insurance, protection, and taxes attributed to storage. (Sections 18.1 and 18.2)

Independent demand Demand for a product that does not depend on the demand for any of the company's other products. (Section 18.3)

Installation stock The stock of an item that is physically on hand at an installation. (Section 18.5)

Inventory A stock of goods being held for future use or sale. (Introduction)

Inventory policy A policy for when to replenish inventory and by how much. (Introduction)

Just-in-time (JIT) inventory system An inventory system that places great emphasis on reducing inventory levels to a bare minimum, so the items are provided just in time as they are needed. (Section 18.3)

Lead time The amount of time between the placement of an order and its receipt. (Section 18.3)

Material requirements planning (MRP) A computer-based system for planning, scheduling, and controlling the production of all the components of a final product. (Section 18.3)

Multiechelon inventory system An inventory system with multiple stages at which inventory is held. (Section 18.5)

Newsvendor problem A standard stochastic single-period model for perishable products. (Section 18.7)

No backlogging The situation where excess demand either must be met through a priority replenishment of inventory or it will be lost. (Section 18.2)

Ordering cost The total cost of ordering (either through purchasing or producing) some amount to replenish inventory. (Sections 18.1 and 18.2)

Periodic review The inventory level is checked only at discrete intervals and replenishment decisions are made only at those times. (Section 18.2)

Perishable product A product that can be carried in inventory for only a very limited period before it can no longer be sold. (Section 18.7)

Quantity discounts Discounts that are provided when sufficiently large orders are placed. (Section 18.3)

(R, Q) policy An abbreviation for reorder-point, order-quantity policy, where R is the reorder point and Q is the order quantity. (Section 18.6)

Reorder point The inventory level at which an order is placed to replenish inventory in a continuous-review inventory system. (Section 18.3)

Reorder-point, order-quantity policy A policy for a stochastic continuous-review inventory system that calls for placing an order for a certain quantity each time that the inventory level drops to the reorder point. (Section 18.6)

Safety stock The expected inventory level just before an order quantity is received. (Section 18.6)

Salvage value The value of an item if it is left over when no further inventory is desired. (Section 18.2)

Scientific inventory management The process of formulating a mathematical model to seek and apply an optimal inventory policy while using a computerized information processing system. (Introduction)

Serial multiechelon system A multiechelon inventory system where there is only a single installation at each echelon. (Section 18.5)

Set-up cost The fixed cost (independent of order size) associated with placing an order to replenish inventory. When purchasing, this is the administrative cost of ordering. When producing, this is the cost incurred in setting up to start a production run. (Sections 18.1 and 18.2)

Shortage cost The cost incurred when the demand for a product in inventory exceeds the amount available there. (Sections 18.1 and 18.2)

Stable product A product which will remain sellable indefinitely so there is no deadline for disposing of its inventory. (Section 18.7)

Supply chain A network of facilities that procure raw materials, transform them into intermediate goods and then final products, and finally deliver the products to customers through a distribution system that usually includes a multiechelon inventory system. (Section 18.5)

Two-bin system A type of continuous-review inventory system where all the units of a product are held in two bins and a replenishment order is placed when the first bin is depleted, so the second bin then is drawn on during the lead time for the delivery. (Section 18.6)

Glossary for Chapter 19

Average cost criterion A criterion for measuring the performance of a Markov decision process by using its expected average cost per unit time. (Sections 19.1 and 19.2)

Deterministic policy A policy that always remains the same over time. (Section 19.2)

Discounted cost criterion A criterion for measuring the performance of a Markov decision process by using its expected total discounted cost based on the time value of money. (Section 19.5)

Method of successive approximations A method for quickly finding at least an approximation to an optimal policy for a Markov decision process under the discounted cost criterion by solving for the optimal policy with n stages to go for $n = 1$, then $n = 2$, and so forth up to some small value of n. (Section 19.5)

Policy A specification of the decisions for the respective states of a Markov decision process. (Section 19.2)

Policy improvement algorithm An algorithm that solves a Markov decision process by iteratively improving the current policy until no further improvement can be made because the current policy is optimal. (Sections 19.3 and 19.4)

Randomized policy A policy where a probability distribution is used for the decision to be made for each of the respective states of a Markov decision process. (Section 19.3)

Stationary policy A policy that always remains the same over time. (Section 19.2)

Glossary for Chapter 20

Acceptance-rejection method A method for generating random observations from a continuous probability distribution. (Section 20.4)

Animation A computer display with icons that shows what is happening in a simulation. (Section 20.5)

Applications-oriented simulator A software package designed for simulating a fairly specific type of stochastic system. (Section 20.5)

Assumption cell An input cell (for a spreadsheet simulation) that has a random value so that an assumed probability distribution must be entered into the cell instead of permanently entering a single number. (Section 20.6)

Congruential methods A popular class of methods for generating a sequence of random numbers over some range. (Section 20.3)

Continuous simulation The type of simulation where changes in the state of the system occur continuously over time. (Section 20.1)

Cycle length The number of consecutive pseudo-random numbers in a sequence before it begins repeating itself. (Section 20.3)

Decision Table tool A Crystal Ball module that systematically applies simulation over a range of values of one or two decision variables and then displays the results in a table. (Section 20.6)

Discrete-event simulation The type of simulation where changes in the state of the system occur instantaneously at random points in time as a result of the occurrence of discrete events. (Section 20.1)

Distribution Gallery Crystal Ball's gallery of 17 probability distributions from which one is chosen to enter into any assumption cell. (Section 20.6)

Dynamic option A Crystal Ball option such that when cell references are entered into any of the parameter fields in the dialogue box for a distribution, choosing the dynamic option causes each of these cell references to be evaluated for each separate trial of the simulation run. (Section 20.6)

Fixed-time incrementing A time advance method that always advances the simulation clock by a fixed amount. (Section 20.1)

Forecast cell An output cell that is being used by a spreadsheet simulation to forecast a measure of performance. (Section 20.6)

General-purpose simulation language A general-purpose computer language for programming almost any kind of simulation model. (Section 20.5)

Inverse transformation method A method for generating random observations from a probability distribution. (Section 20.4)

Next-event incrementing A time advance method that advances the time on the simulation clock by repeatedly moving from the current event to the next event that will occur in the simulated system. (Section 20.1)

OptQuest A Crystal Ball module that systematically searches for an optimal solution for a simulation model with any number of decision variables. (Section 20.7)

Pseudo-random numbers A term sometimes applied to random numbers generated by a computer because such numbers are predictable and reproducible. (Section 20.3)

Random integer number A random observation from a discretized uniform distribution over some range. (Section 20.3)

Random number A random observation from some form of a uniform distribution. (Section 20.3)

Random number generator An algorithm that produces sequences of numbers that follow a specified probability distribution and possess the appearance of randomness. (Section 20.3)

Seed An initial random number that is used by a congruential method to initiate the generation of a sequence of random numbers. (Section 20.3)

Simulation clock A variable in the computer program that records how much simulated time has elapsed so far. (Section 20.1)

Simulation model A representation of the system to be simulated that also describes how the simulation will be performed. (Section 20.1)

Simulator A shorthand name for *applications-oriented simulator* (defined above). (Section 20.5)

State of the system The key information that defines the current status of the system. (Section 20.1)

Static option A Crystal Ball option such that when cell references are entered into any of the parameter fields in the dialogue box for a distribution, choosing the static option causes each of these cell references to be evaluated only once, at the beginning of the simulation run, so the parameter values at that point are used for all trials of the simulation. (Section 20.6)

Time advance methods Methods for advancing the simulation clock and recording the operation of the system. (Section 20.1)

Trial Crystal Ball's term for a single application of the process of generating a random observation from each probability distribution entered into a spreadsheet simulation and then calculating the output cells in the usual way and recording the results of interest. (Section 20.6)

Uniform random number A random observation from a (continuous) uniform distribution over some interval $[a, b]$, commonly where $a = 0$ and $b = 1$. (Section 20.3)

Warm-up period The initial period waiting to essentially reach a steady-state condition before collecting data during a simulation run. (Section 20.1)