REVIEW ARTICLE

Combinatorial optimization in science and engineering

Julius Beneoluchi Odili

Department of Mathematical Sciences, Faculty of Natural and Applied Sciences, Anchor University, Lagos, Nigeria

This article is a review of combinatorial optimization in science and engineering applications. Combinatorial optimization has found wide applicability in most of our day-to-day affairs, ranging from industrial, academic, logistic to manufacturing applications, etc. This study introduces the concepts of optimization identifying the different types of optimization in the literature, before focusing on discrete optimization methods. Moreover, much emphasis is placed on the application areas, examples and the development of mathematical models in combinatorial optimization. The study concludes by highlighting the merits and demerits of combinatorial optimization models and recommends further studies on the development of more efficient and user-friendly combinatorial optimization methods.

Keywords: Combinatorial optimization models, discrete optimization, mathematical models, science and engineering applications.

THERE is hardly any field today – ranging from medicine, pharmacy, science, engineering to business – that does not require optimization. It is at the heart of decision-making in manufacturing and industrial sectors, and is a veritable tool in the analysis of physical systems[1]. Simply put, optimization is concerned with discovering the best solution among several feasible ones. In mathematical terms, optimization deals with the search of the optimal object among several objects, especially in situations where a complete feasible search is impossible[2]. The domain of optimization is usually in situations where the feasible solutions could be discrete or continuous with a definite optimal solution. Whether in continuous or discrete optimization, the overall goal is to minimize or maximize a function. In other words, optimization is the economics of science and engineering with the aim of minimizing costs and maximizing profit, time usage or industrial procedures[3].

Several optimization categories have been identified in the literature. Some of these are combinatorial optimization[4], complementarity problems[5], constrained optimization[6], unconstrained optimization[7], continuous optimization[8], discrete optimization[9], global optimization[10], integer linear programming[11], linear programming (LP)[12], network optimization[13], non-differentiable optimization nonlinear equations[14], optimization under uncertainty[15], quadratically constrained quadratic programming (QCQP)[16], quadratic programming (QP)[17], semidefinite programming (SDP)[18], semi-infinite programming (SIP)[19], stochastic linear programming[20], second-order cone programming (SOCP)[21], stochastic programming[22], nonlinear programming[23], nonlinear least-squares problems[24], mixed integer nonlinear programming (MINLP)[25], bound constrained optimization[26], mathematical programs with equilibrium constraints (MPEC)[27], multi-objective optimization[28] and derivative-free optimization[29]. Here, we shall group all these methods into two broad categories, namely continuous and discrete optimization.

Continuous optimization minimizes or maximizes a function using continuous real numbers that accept value points from an integer set to another and that include negative values, decimals and fractions[10]. That is, continuous optimization accepts numerical values that can appear in the real world as well as in the abstract mathematical world. Some experts believe continuous optimization is more accurate and more complex than discrete optimization[30]. However, many others argue to the contrary[31].

Discrete optimization, on the other hand, refers to a subclass of optimization that is concerned with the use of integers instead of fractions or decimals, to execute minimization or maximization of functions. Discrete optimization can be further subdivided into combinatorial optimization and integer programming (IP)[32]. Since the focus of this article is on combinatorial optimization, we shall elaborate on these two classes of discrete optimization.

Integer programming

An IP problem basically refers to mathematical optimization programs where all variables are integers. However, there are special cases where fractions or decimals could form part of the constraints. Usually such cases are technically referred to as mixed integer programming (MIP). In most settings, IP refers to ILP, since in most of the formulations of IP both the objective function and the constraints are linear[33]. That is not to mention that nonlinear integers do not exist[34]. Here we will concentrate on LIP problems.

e-mail: odili_julest@yahoo.com

An IP takes a form similar to the following

Maximize $\boldsymbol{C}^{\mathrm{T}} j$,

Subject to $B_j \leq \boldsymbol{d}$,

where $j \geq 0, \quad j \in \Re^n$. (1)

The ILP equivalent of eq. (1) will take the form

Maximize $\boldsymbol{C}^{\mathrm{T}} j$,

Subject to $B_j + s = \boldsymbol{d}$,

$s \geq 0, \quad j \in \Re^n$. (2)

where $\boldsymbol{C}$ and $\boldsymbol{d}$ are vectors and B is an integer matrix.

The frequent use of IP is due basically to its wide applicability in real-life situations, where most decisions are discrete in terms of a yes or no, true or false, move or stop, etc. Hence IP is sometimes called binary programming (BP), because it usually assumes 0–1 decision variables[35]. Let us consider a practical example on budgeting. A company has four building projects in four different locations requiring different capital investment and financial returns on investment with statistics as available in Table 1. Let us assume we are required to determine the project to embark upon to maximize profit with the available finances.

From Table 1, the objective to maximize is

$$0.2x^1 + 0.3x^2 + 0.4x^3 + 0.5x^4. \quad (3)$$

This being a decision problem requires binary variables (0, 1) to represent the final decision. So we define the financial constraints representing the projects by x thus

$$0.5x^1 + 1.0x^2 + 1.5x^3 + 2.1x^4 \leq 3.1 \text{ (first year)}, \quad (4)$$

$$0.3x^1 + 0.6x^2 + 1.5x^3 + 1.7x^4 \leq 2.3 \text{ (second year)}, \quad (5)$$

$$0.2x^1 + 0.5x^2 + 0.3x^3 + 0.8x^4 \leq 1.4 \text{ (third year)}, \quad (6)$$

$$x_k = 0 \text{ or } 1,\ k = 1, \ldots, 4. \quad (7)$$

Here we assume that whenever the decision is not to execute any project x_k, k is 0; otherwise it is 1.

Combinatorial optimization

Combinatorial optimization which deals with problems on discrete structures such as mastoids and graphs is concerned with finding the best option among a set of options. In mathematics, artificial intelligence, operations research, algorithm theory and software engineering, combinatorial optimization deals with finding the optimal solution out of a set of discrete feasible solutions, especially in situations where an exhaustive search is not possible. It is safe to say, therefore, that combinatorial optimization problems refer to such problem instances where the feasible solutions are defined with combinatorics concepts (e.g. combinations, permutations, sequences, sets and subsets), or graph theory concepts such as nodes, arcs, paths cycles, etc. that occur between two discrete values rather than on the smoothness of the graph[36]. Generally, therefore, the graph of continuous optimization is smoother due to the concrete and fixed nature of the discrete values in discrete optimization which hinders smooth transitions from one value to the other.

Areas of application of combinatorial optimization

Combinatorial optimization has wide applicability in our everyday life, especially in proffering solutions to problems in science and engineering. Some of the problems that can be solved using combinatorial optimization include:

- A manufacturing firm producing high-quality goods that are in demand in different parts of a large country, will need to find the optimal route to service its customers in a way to minimize time and costs in such distribution service taking into cognizance the transportation models available, transportation costs, transportation network, haulage capacity constraints, available human resources, customer conveniences, applicable taxes, etc. This is a clear graph theory problem that requires combinatorial optimization.
- An industrial establishment having a goal to maximize its profit, bearing in mind the possible production costs, human resources, equipment and raw materials availability and demand–forecast constraints. This is another case where multi-objective combinatorial optimization is required to achieve the company's goals.

Table 1. Financial investment

		Capital requirements (US$ m) Year		
Project	Return (US$ m)	1	2	3
1	0.2	0.5	0.3	0.2
2	0.3	1.0	0.6	0.5
3	0.4	1.5	1.5	0.3
4	0.5	2.1	1.7	0.8
Total capital available (US$ m)		3.1	2.3	1.4

Note. All capital requirements and returns are quoted in millions of US dollars. The returns are calculated after three years of investment.

- Weather forecast is an inevitable requirement in the aviation industry all over the world. To ensure good weather forecast, experts are required to explore the weather conditions with particular emphasis on wind speed, temperature, humidity, etc. in a given number of locations for a given number of hours to predict the weather in the next couple of hours. This requires setting up an objective function to measure the correlation between observations and the atmospheric condition, bearing in mind the available constraints and the stated variables – a clear case of combinatorial optimization.
- In image processing, computer applications are developed to identify handwritten characters in pixelated image form and then output the best guess of the represented digit of the image. This situation requires solving an optimization problem to adjust the training parameters in order to reduce the error count so as to attain dependable digit recognition.
- In economics, a prudent businessman can only invest in a given venture after an exhaustive preview of available funds, variances and projected returns on investment depending on the projected risks on such investment. This analysis involves combinatorial optimization.
- In commercial transportation, whether it is road, rail or air transportation, a firm needs to choose its fleet routes in the most efficient manner so as to maximize efficiency and effectiveness. This requires the application of combinatorial optimization.

In summary, combinatorial optimization in real-life situations determines the most efficient way of allocating resources in order to obtain maximum returns on investment. Other recent applications of combinatorial optimization include scheduling of production plants in industries, examination time-tabling in schools, colleges and universities, staff scheduling in large corporations, etc. In a nutshell, the broad areas of application of combinatorial optimization include determining the best cost, weight, distance, profit, value, utility, yield, production, capacity, etc. in various fields.

Development of combinatorial optimization models

Solving combinatorial optimization problems requires the development of algorithms that use a sequence of values of the variables (iterates) to arrive at a solution to the problem. To achieve this, the algorithm may require some previous knowledge gained in a previous iteration and intimate information about the problem being solved, including knowledge about its sensitivity to noise in the variables, etc. To successfully develop a good algorithm, therefore, the primary goal is the development of a good mathematical model of the targeted problems which could be designed from economic, statistical, biological or other natural principles. The model should describe the relationship between variables and may place restriction on states (constraints). Similarly, the model should have an objective function that needs to be minimized or maximized. The objective function evaluates the desirability of a set of values of the variables[4].

Essentials required for constructing mathematical models

In constructing a mathematical model, it is necessary to understand that a good model should consist of three principal elements. These are the objective function, variables and constraints of a problem.

The objective function refers to a quantifiable evaluation of operation of the system that is the target of minimization or maximization. The index in the case industrial system could be maximization of profit or minimization of production, labour or distribution costs, etc.

Variables refer to the unknown quantity/components of the system of interest for which we must find the right values to meet the set objective. In an industrial system, the variables may be the number of human resources required to solve a given problem, demand for a particular product, production time, storage facilities, distribution channels, etc. For data-fitting in an experimental situation, the variables could be the parameters for the model.

Constraints describe the relationships among the variables and should also indicate the permissible values. A constraint in industrial firms could be the labour and other resources which should be less than or equal to the ones available.

Mathematical formulation

In developing a mathematical formulation for a combinatorial optimization model taking into cognizance the essential elements, a combinatorial optimization problem K is a quadruple consisting of I, FS, M and GF, where I represents a set of instances, FS is a set of feasible solutions, M is a measure of the feasible solution and GF is the goal function either to minimize or maximize an objective. That is, for every combinational problem, there is always a decision problem that studies whether there exists a feasible solution for a particular measure of the problem leading to an answer that could be a yes or a no[37].

Examples of combinatorial optimization problems

There exist several combinatorial problems that have been identified in the literature, some of which are the travelling salesman problem, minimum spanning tree, packing problem, cutting stock problem, etc.

The travelling salesman problem

This is the problem of a particular salesman whose duty is to visit a given number of his customers located in different parts of a particular city or in different towns across a given geographical location using the cheapest possible route in terms of time and cost of the journey[38]. Another constraint in this problem is that the salesman is expected to visit each of the cities only once, as much as possible. The traveling salesman is allowed to return to his initial starting city at the end of his trip. To ensure fairness, a cost is placed on each of the routes (edges)[39].

The travelling salesman problem (TSP) is a hard non-deterministic polynomial time (NP-hard) problem generates a decision problem that is NP-complete[40]. This graph theory problem represents the cities as vertices and the routes connecting them as edges on the graph[41]. Several optimization algorithms have been used to successfully solve this problem. Examples of such combinatorial algorithms are ant colony optimization[42], African buffalo optimization[41], artificial bee colony optimization[43], particle swarm optimization[44], etc.

Application areas of TSP include network routing[45], parcel delivery by courier companies[46], job-shop scheduling[47], circuit board and oil-rig drilling operations[48], vehicle routing[49], urban route planning[50], etc.

Minimum spanning tree

This is another combinatorial problem that has become popular in recent years. A tree in graph theory is a simply a way devised to connect all the nodes of a graph together in order to create just a path from one node to the other throughout the whole graph. In this way, if it is a practical situation involving different cities connected by road networks, the tree should be created in such a way to enable a person connect from one city to another within the minimum spanning tree. The constraint is that each city can only be reached from a point by one particular path. Figure 1 shows an example of minimum spanning for a four-node graph. This four-node graph has 16 spanning trees (Figure 2).

A minimum spanning tree, therefore, is a special kind of tree that minimizes the sum of the individual weights/costs attached to individual edges in the graph. A graph like the four-node graph above could contain many minimum spanning trees if all the edges have equal weights, since in that case every tree is a minimum spanning tree. On the other hand, if all the edges have different weights such that no two edges have the same weight, then the graph has one minimum spanning tree. The basic idea of minimum spanning is a spanning tree that has a weight that is less than or equal to the weight of all other spanning trees in the graph[51].

Three prominent algorithms have proven to be successful in solving the minimum spanning tree problems. These are the Kruskal algorithm[52], Prim's algorithm[53] and Boruvka's algorithm[54]. Recent application areas of the minimum spanning tree include communication networks, transportation networks, water supply, electricity supply and computer networks environments.

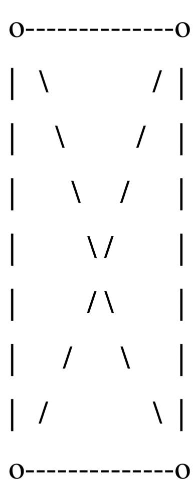

Figure 1. A four-node graph.

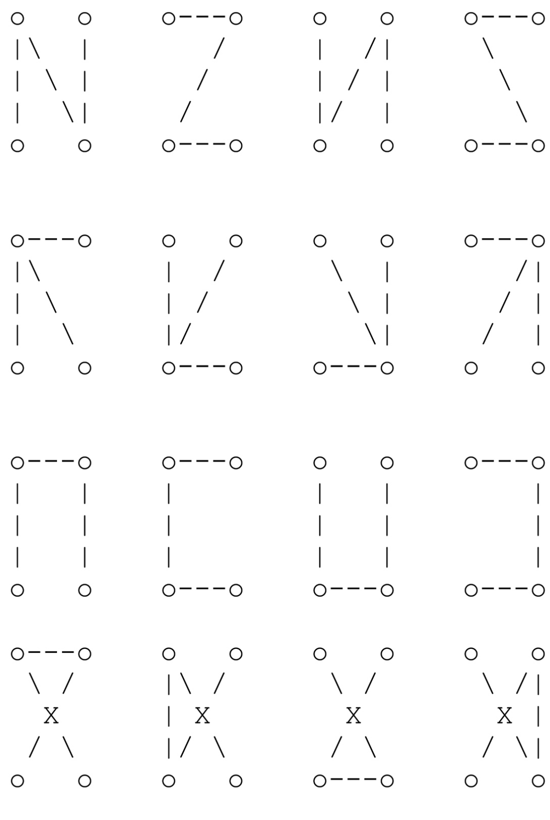

Figure 2. Four-nodes graph indicating 16 spanning trees.

The cutting stock problem

This problem has its origin in industries involved in cutting large pieces of clothing, metal, film, glass or paper materials into smaller shapes and sizes as required by the customers, while ensuring minimum wastage.

In formulating a model to solve the cutting stock problem, one may first list the number of j orders placed, each customer requiring k_i sizes and quantities ($i = 1,\ldots,j$ sizes). The IP equation could be represented in this form

$$\min \sum_{i=1}^{n} j_i k_i. \tag{8}$$

Subject to

$$\sum_{i=1}^{n} b_{ij} x_i \geq k_i,\ i, j = 1,\ldots,m, \tag{9}$$

$$x_i \geq 0, \tag{10}$$

here, b_{ij} represents i patterns appearing in order j, j_i the wastage recorded in the cutting process of pattern i, and k_i are the sizes to which the materials are cut[55].

Packing problem

There are several variants of the packing problem. Basically, it refers to filling a box, ship or haulage vehicles with specified smaller-shaped items in a given manner. Some variants of the problem are concerned with filling the larger items in such a way as to avoid overlaps. In other variants of the packing problem, overlaps are permitted, but the packing must not have gaps in the larger container. In general, packing problem is a merger of mathematics, puzzles and computer science in providing solutions to practical real-life problems. Some variants of the packing problem includes the parallel parking problem that involves parking cars in such a way as to allow as may cars as possible within a given parking lot. Other variants are packing infinite space, hexagonal packing, circles in circle, tiling of floors, etc. In Packing problem, the constraint is not always the breadth or length sometimes it is the weight or height of the items[56]. In a way, packing problem is similar to cutting stock problem, since both require managing a larger space with smaller shapes in a most efficient (profitable) manner.

Merits and demerits of combinatorial optimization

From the foregoing discussion and analysis, it is obvious that combinatorial optimization has become a part of our daily lives. However a critical observer may notice that that are some obvious benefits and shortcomings in the use of combinatorial optimization in attempting solutions to our practical day-to-day problems.

Merits

Among the benefits derivable from the use of combinatorial optimization is solving practical problems are the following:

- *Portability:* Most combinatorial application models have wide applicability. That is, once a reliable model has been developed, it requires just tuning of the appropriate parameters of the same model to solve many other problems. This saves cost, especially in a large conglomerate/corporation as the same model can be used in different branches and departments of the industrial establishment.
- *Cost-saving:* It is generally agreed by economists that it is cheaper to develop a model to attempt solutions to manufacturing problems than to use the real system for study. In cases of errors, the huge loss associated with learning-on-the-job using the real system can be enormous. The cost in an event of failure through the use of a combinatorial optimization model is by far cheaper than that of the real system.
- *Efficiency:* In many cases, in spite of the relative simplicity of the combinatorial optimization models, they have proven to be efficient such that they could even produce more reliable results than some real-life system. Generally, it is much easier to replicate the same experimental process when using a combinatorial optimization model than when using human subjects.
- *Time-saving:* Industrial process modelling and simulation using combinatorial optimization is usually much faster than using human subjects. Experiments using human subjects, machines, etc. in order to forecast outcomes could take several years. This explains the popularity of combinatorial optimization methods in solving real-life problems in the past few decades.
- *User-friendliness:* Another high point regarding the use of combinatorial optimization is the ease of use. Once the model is developed, a user does not have to be an expert in mathematics or computer science to analyse and interpret the output from the model.

The above-mentioned benefits explain the popularity of combinatorial optimization models in virtually every aspect of finance, supply-chain management, logistics, manufacturing and educational institutions today. However, there are some observed lapses that need to be addressed in order to maximize the benefits of combinatorial optimization.

Demerits

Though combinatorial optimization models are popular in decision-making today, much needs to be done in order to maximize the benefits of use of such models. Some of the observed lapses include:

- *Over-generalization:* Practical industrial processes are unique and the effects of different manufacturing parameters differ from place to place. Combinatorial optimization models are theoretical and as such may be unable to capture special consideration leading to cases of over-generalization of computational results.
- *Inaccurate computational results:* Combinatorial optimization models are generally stochastic. That is, they do not lay claim to discovering the optimal solutions. At best, they obtain near-optimal solutions. Making huge investments based on the use of such stochastic methods may not, in the long-run, be a wise decision.
- *Software and other costs:* Due to the rising popularity of combinatorial optimization models, the cost of combinatorial optimization software is prohibitive. Investing in such software may constitute a huge drain on the financial resources of an industrial firm. Besides, there are personnel costs and other associated costs.
- *Inaccurate analysis and interpretation of results:* Another challenge with the use of combinatorial optimization models in decision making is the problem of inaccurate analysis and interpretation of computational results in order to make informed decisions. This may require some level of professionalism, which further increases the investment cost.

Conclusion

This study is a review of combinatorial optimization with special emphasis on the practical application areas of combinatorial optimization models, the processes required for building a good model. It also discusses the existing combinatorial optimization problems in mathematics, operations research and computer science. The study concludes with a discussion on the merits and demerits application of combinatorial optimization models in different practical aspects of our day-to-day lives.

In view of the observed weaknesses of combinatorial optimization in real-life applications we recommend that further research needs to be done with a view to developing simpler, faster, efficient and more user-friendly combinatorial optimization models.

1. Gigerenzer, G. and Gaissmaier, W., Heuristic decision making. *Annu. Rev. Psychol.*, 2011, **62**, 451–482.
2. Vazan, P. and Tanuska, P. (eds), A short reflection on the strengths and weaknesses of simulation optimization. *Int. J. Comput. Inform. Eng.*, 2012, **6**(5), 640–644.
3. Miller, B. M. and Rubinovich, E. Y., *Impulsive Control in Continuous and Discrete–Continuous Systems*, 2012; www.books.google.com
4. Wolsey, L. A. and Nemhauser, G. L., *Integer and Combinatorial Optimization*, John Wiley, 2014.
5. Huang, Z.-H. and Ni, T., Smoothing algorithms for complementarity problems over symmetric cones. *Comput. Optim. Appl.*, 2010, **45**(3), 557–579.
6. Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 2014.
7. Tuba, M., Subotic, M. and Stanarevic, N. (eds), Modified cuckoo search algorithm for unconstrained optimization problems. In Proceedings of the 5th European Conference on European Computing Conference, World Scientific and Engineering Academy and Society, Paris, France, 2011, pp. 263–268.
8. Crandall, D. *et al.* (eds), Discrete-continuous optimization for large-scale structure from motion. Computer Vision and Pattern Recognition, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011.
9. Kouvelis, P. and Yu, G., *Robust Discrete Optimization and its Applications*, Springer Science & Business Media, 2013.
10. Horst, R. and Tuy, H., *Global Optimization: Deterministic Approaches*, Springer Science & Business Media, 2013.
11. Morais, H., Kádár, P., Faria, P., Vale, Z. A. and Khodr, H. M., Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renew. Energy*, 2010, **35**(1), 151–156.
12. Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D., *Linear Programming and Network Flows*, John Wiley, 2011.
13. Xie, C., Lin, D.-Y. and Waller, S. T., A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transp. Res. Part E*, 2010, **46**(3), 295–316.
14. Rao, R., Savsani, V. and Vakharia, D., Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inf. Sci.*, 2012, **183**(1), 1–15.
15. Conti, S., Held, H., Pach, M., Rumpf, M. and Schultz, R., Shape optimization under uncertainty – a stochastic programming perspective. *SIAM J. Optim.*, 2009, **19**(4), 1610–1632.
16. Anstreicher, K. M., On convex relaxations for quadratically constrained quadratic programming. *Math. Program.*, 2012, **136**(2), 233–251.
17. Rodriguez-Lujan, I., Huerta, R., Elkan, C. and Cruz, C. S., Quadratic programming feature selection. *J. Mach. Learn. Res.*, 2010, **11**, 1491–1516.
18. Wolkowicz, H., Saigal, R. and Vandenberghe, L., *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Springer Science & Business Media, 2012; www.springer.com/gp/book/9780792377719 (accessed on 8 December 2017).
19. Sivaramakrishnan, K. K., Linear programming approaches to semidefinite programming problems, Doctoral thesis, Rensselaer Polytechnic Institute, New York, USA, 2002.
20. Higle, J. L. and Sen, S., *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, Springer Science & Business Media, 2013; www.springer.com/gp/book/9780792338406 (accessed on 8 December 2017).
21. Le, C. V., Gilbert, M. and Askes, H., Limit analysis of plates using the EFG method and second-order cone programming. *Int. J. Num. Methods Eng.*, 2009, **78**(13), 1532–1552.
22. Birge, J. R. and Louveaux, F., *Introduction to Stochastic Programming*, Springer Science & Business Media, 2011.
23. Kuhn, H. W., Nonlinear programming: a historical view. In *Traces and Emergence of Nonlinear Programming*, Springer, 2014, pp. 393–414; https://link.springer.com/978-1-4614-0237-4 (accessed on 8 December 2017).
24. Gratton, S., Lawless, A. S. and Nichols, N. K., Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM J. Optim.*, 2007, **18**(1), 106–132.
25. Lee, J. and Leyffer, S., *Mixed Integer Nonlinear Programming*, Springer Science & Business Media, 2011; http://neos-guide.org/content/mixed-integer-nonlinear-programming (accessed on 8 December 2017).

26. Morales, J. L. and Nocedal, J., Remark on Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Trans. Math. Software (TOMS)*, 2011, **38**(1), 7.
27. Luo, Z.-Q., Pang, J.-S. and Ralph, D., *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, 1996.
28. Deb, K., Multi-objective optimization. In *Search Methodologies*, Springer, 2014, pp. 403–449.
29. Rios, L. M. and Sahinidis, N. V., Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Global Optim.*, 2013, **56**(3), 1247–1293.
30. Streiner, D. L., Norman, G. R. and Cairney, J., *Health Measurement Scales: A Practical Guide to their Development and Use*, Oxford University Press, Kenya, 2014.
31. Davenport, T. H., *Process Innovation: Reengineering Work through Information Technology*, Harvard Business Press, 2013.
32. Nemhauser, G. and Bienstock, D., *Integer Programming and Combinatorial Optimization*, Springer, 2005; https://link.springer.com/book/10.1007/978-3-319-07557-0 (accessed on 8 December 2017).
33. Margot, F., Symmetry in integer linear programming. In *50 Years of Integer Programming 1958–2008*, Springer, 2010, pp. 647–686.
34. Hemmecke, R., Köppe, M., Lee, J. and Weismantel, R., Nonlinear integer programming. In *50 Years of Integer Programming 1958–2008*, Springer, 2010, pp. 561–618.
35. Finkelman, M. D., Kim, W., Roussos, L. and Verschoor, A., A binary programming approach to automated test assembly for cognitive diagnosis models. *Appl. Psychol. Meas.*, 2010, **34**(5), 310–326.
36. Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer Science & Business Media, 2012; www.springer.com/gp/book/9783540137221 (accessed on 8 December 2017).
37. Bianchi, L., Dorigo, M., Gambardella, L. M. and Gutjahr, W. J., A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.*, 2009, **8**(2), 239–287.
38. Odili, J. B., Kahar, M. N. M. and Anwar, S., African buffalo optimization: a swarm-intelligence technique. *Proc. Comput. Sci.*, 2015, **76**, 443–448.
39. Odili, J. B. *et al.*, Application of ant colony optimization to solving the traveling salesman's problem. *Sci. J. Electr. Electron. Eng.*, 2013, 175–176.
40. Odili, J. B. and Mohmad Kahar, M. N., Solving the traveling salesman's problem using the African buffalo optimization. *Comput. Intell. Neurosci.*, 2016, 1–12.
41. Odili, J. B. and Kahar, M. N. M., African buffalo optimization (ABO): a new meta-heuristic algorithm. *J. Adv. Appl. Sci.*, 2015, 101–106.
42. Dorigo, M. and Birattari, M., Ant colony optimization. In *Encyclopedia of Machine Learning*, Springer, 2010, pp. 36–39.
43. Akay, B. and Karaboga, D., A modified artificial bee colony algorithm for real-parameter optimization. *Inf. Sci.*, 2012, **192**, 120–142.
44. Kennedy, J., Particle swarm optimization. In *Encyclopedia of Machine Learning*, Springer, 2010, pp. 760–766; https://link.springer.com/referencework/10.1007%2F978-0-387-30164-8 (accessed on 8 December 2017).
45. Nagy, G. and Salhi, S., Location-routing: issues, models and methods. *Eur. J. Oper. Res.*, 2007, **177**(2), 649–672.
46. Mitrović-Minić, S., Krishnamurti, R. and Laporte, G., Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transp. Res. Part B*, 2004, **38**(8), 669–685.
47. Whitley, L. D., Starkweather, T. and Fuquay, D. A. (eds), Scheduling problems and traveling salesmen: the genetic edge recombination operator. In Proceedings of the 3rd International Conference on Genetic Algorithms, George Mason University, Fairfax, Virginia, USA, June 1989.
48. Applegate, D. L., Bixby, R. E., Chvatal, V. and Cook, W. J., *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, 2011.
49. Laporte, G., The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.*, 1992, **59**(3), 345–358.
50. Nuortio, T., Kytöjoki, J., Niska, H. and Bräysy, O., Improved route planning and scheduling of waste collection and transport. *Expert Syst. Appl.*, 2006, **30**(2), 223–232.
51. Singh, A., An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Appl. Soft Comput.*, 2009, **9**(2), 625–631.
52. Han, D.-M. and Lim, J.-H., Smart home energy management system using IEEE 802.15. 4 and zigbee. *IEEE Transactions Consumer Electronics*, 2010, **56**(3), 1403–1410.
53. Manen, S., Guillaumin, M. and Van Gool, L. (eds), Prime object proposals with randomized Prim's algorithm. In IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 2013.
54. Bergantiños, G. and Vidal-Puga, J., The folk solution and Boruvka's algorithm in minimum cost spanning tree problems. *Discrete Appl. Math.*, 2011, **159**(12), 1279–1283.
55. Belov, G. and Scheithauer, G., A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *Eur. J. Oper. Res.*, 2002, **141**(2), 274–294.
56. Wäscher, G., Haußner, H. and Schumann, H., An improved typology of cutting and packing problems. *Eur. J. Oper. Res.*, 2007, **183**(3), 1109–1130.

ACKNOWLEDGEMENTS. We thank anonymous reviewers for their useful inputs and the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Malaysia for providing funds through Grant PGRS 1403118.

Received 7 January 2017; revised accepted 31 August 2017

doi: 10.18520/cs/v113/i12/2268-2274