Les Cahiers du GERAD ISSN: 0711–2440

A seminal contribution of Ailsa Land and Alison Doig Harcourt to the field of mathematical programming

G. Laporte

G–2023–60

December 2023
Revised: June 2024

Citation suggérée : G. Laporte (Décembre 2023). A seminal contribution of Ailsa Land and Alison Doig Harcourt to the field of mathematical programming, Rapport technique, Les Cahiers du GERAD G– 2023–60, GERAD, HEC Montréal, Canada. Version révisée: Juin 2024

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2023-60) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: G. Laporte (December 2023). A seminal contribution of Ailsa Land and Alison Doig Harcourt to the field of mathematical programming, Technical report, Les Cahiers du GERAD G–2023–60, GERAD, HEC Montréal, Canada. Revised version: June 2024

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2023-60) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2023
– Bibliothèque et Archives Canada, 2023

Legal deposit – Bibliothèque et Archives nationales du Québec, 2023
– Library and Archives Canada, 2023

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A seminal contribution of Ailsa Land and Alison Doig Harcourt to the field of mathematical programming

Gilbert Laporte

Department of Decision Sciences, HEC Montréal & GERAD, Montréal, (Qc), Canada, H3T 2A7

gilbert.laporte@hec.ca

December 2023
Revised: June 2024
Les Cahiers du GERAD
G–2023–60

Abstract : In 1960, Ailsa Land and Alison Doig published the first linear programming-based branch-and-bound algorithm for the solution of mixed integer linear programs. This algorithm has had a major impact on the field of mathematical programming and remains to this day the only available solution methodology for generic mixed integer linear programs. This article recounts the genesis of this algorithm and describes some of the developments that have taken place since its publication.

Keywords : Mixed integer linear programming, simplex algorithm, Gomory cutting planes, branch-and-bound, branch-and-cut

Acknowledgements: Thanks are due to Jean-Marie Bourjolly for his comments on an earlier version of this paper, to Alysson M. Costa, Frank Land and Louis Ta who provided some useful references.

1 Introduction

Alison Doig Harcourt is an Australian statistician who spent most of her academic career as a faculty member at the University of Melbourne, where she worked until her retirement in 1994. In 2018 she received the title of Doctor of Science from that university, and in 2019 she was named Victorian Senior Australian of the year (Silberberg, 2019). She earned a Master's degree from the University of Melbourne on the solution of a practical paper trim problem which could be formulated as an integer linear program (Doig & Belz, 1956). During her "traveling year", she went to England where she met Ailsa Land at an Operational Research Conference held in Oxford in 1957 (Land & Powell, 2007; Shier, 2022). At that time Ailsa Land had just completed her Ph.D. dissertation at the London School of Economics (LSE) on the application of linear programming to the transportation of coking coal (Land, 1956). Alison Doig then joined the Economics Division of LSE to work with Ailsa Land.

Both were interested in linear programming and integer linear programming, and subsequently started working on an industrial problem at British Petroleum, which meant solving a linear programming problem related to refinery planning and involving the determination of shipments between crude sources and refineries, and storage over time (Land & Powell, 2007; Land, 2021a; Shier, 2022). This problem contained some integer variables, which entailed major complications. Not having access to a computer, they agreed with the company that they should forget trying to solve the problem and work instead on an algorithm capable of solving generic mixed integer linear programs (Land, 2021a). They developed a successful algorithm and published it soon after: Land, A.H. & Doig, A.G. (1960). An automatic method of solving discrete programming problems, *Econometrica* **28**, 497–520. Interestingly, they submitted their paper under their initials and surnames in order not to be identified as women, a precaution that would not be justified today (Land, 2021a). In what follows, I will expand on the significance of this contribution, but before I will provide some background information.

2 Linear programming

Mathematical programming is arguably the main methodology used in operational research. It encompasses several subfields, all rooted in linear programming (LP). The simplex algorithm, conceived by Dantzig in 1947, but apparently only published several years later (Dantzig, 1951), is without any doubt the most important development in this area. One motivation of Dantzig when designing the simplex method for the solution of linear programs was to automatize the solution process so that it could be executed by a computer, very much in the same way that several manufacturing processes had been automatized (Dantzig, 1963, p. 10).

Without loss of generality, a linear program can be expressed as the problem of maximizing (or alternatively of minimizing) an objective function cx (a profit), subject to $Ax = b$ and $x \geq 0$, where x is an n-vector of decision variables, A is an $(m \times n)$-matrix of coefficients of rank m, b is an m-vector of constants, and c is an n-vector of coefficients. The feasible domain is a polyhedron in $\mathbb{R}^n$. If an LP has optimal solutions, then at least one of them occurs at an extreme point of the polyhedron of feasible solutions. Figure 1 depicts a two-dimensional feasible domain with an objective function maximized at an extreme point. Geometrically, the simplex method consists of moving from an extreme point of the solution space to another. Algebraically, this means moving from a set of m linearly independent columns of A, called a basis, to another basis by removing a column and replacing it with another one, with the aim of improving the value of the objective function. Such an exchange is called a pivot. The algorithm ends when no further improvement is possible. In comparison with earlier algorithms such as the Fourier-Motzkin elimination method (Fourier, 1826a,b) for the solution of linear systems, the simplex method seeks to compute a solution that will maximize an objective function explicitly. For classical references on linear programming, see Dantzig (1963) and Schrijver (1986).

The first large-scale computer implementation of the simplex method appears to be that of Orchard-Hays (1955) who solved instances involving 200 constraints and 1,000 variables or more on an IBM 701 computer (Dantzig, 1963, p. 26).

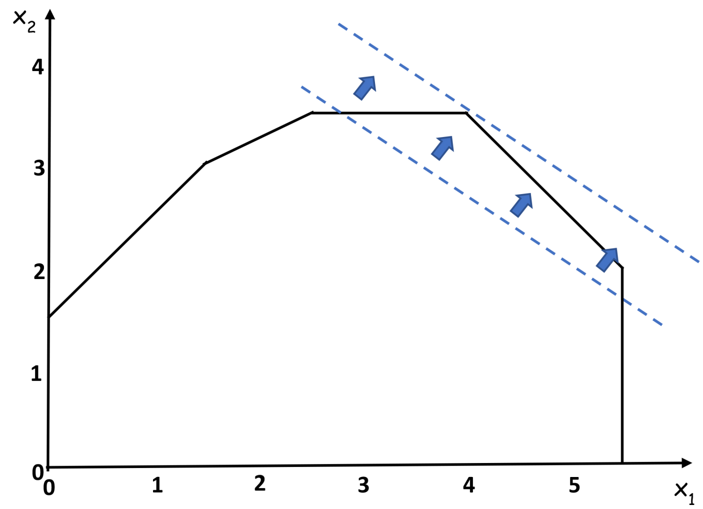

Figure 1: LP feasible region showing an objective function maximized at an extreme point

3 Integer linear programming

In several contexts, some or all variables of a linear program are required to take integer values. One distinguishes between integer linear programs (ILPs) in which all variables must be integer, and their generalization called mixed integer linear programs (MILPs) in which some of the variables may be continuous. Figure 2 depicts an ILP in which the constraints are the same as those of Figure 1, but the variables must also be integer. Solving MILPs as continuous LPs and rounding the values of the fractional variables may make sense in cases where the variables take very large values, but not when they are small. For example, producing 1,000.3 barrels of oil is about the same as producing 1,000 barrels, but rounding 4.3 tankers to 4 tankers may not be feasible. In addition, there exist several problems where binary variables are used to impose logical conditions (Dantzig, 1960) and cannot be rounded.

One appealing solution concept in integer linear programming is the use of cutting planes. These are constraints that shave off parts of the continuous domain without eliminating any feasible integer solution (or, in certain cases, without eliminating all optimal solutions). Figure 3 depicts a cutting plane that removes part of the feasible region of Figure 2 without eliminating any integer coordinate. Gomory (1958) developed the first theoretically convergent cutting plane algorithm for ILPs. Unfortunately, as observed by Land & Powell (2007) and numerous other researchers, despite its elegance Gomory's method suffers from severe convergence problems and typically fails to produce an optimal solution except on tiny instances. There exist, however, some problems, like the symmetric Traveling Salesman Problem (TSP), for which the application of Gomory cutting planes has proved very effective in solving relatively large instances (Miliotis, 1978; Land, 2021b).

4 The Land and Doig algorithm

Land & Doig (1960) devised the first ever linear programming-based branch-and-bound algorithm for the solution of generic MILPs. Branch-and-bound is a technique now commonly used for the solution of optimization problems with some discrete variables, in which the solution space is iteratively

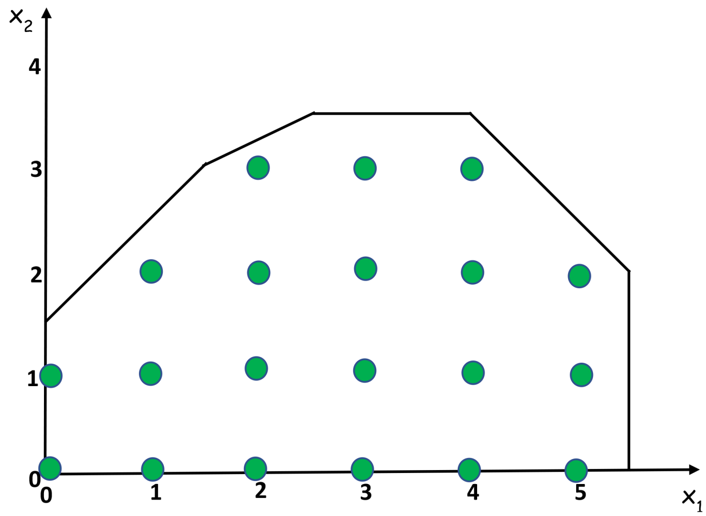

Figure 2: ILP feasible region

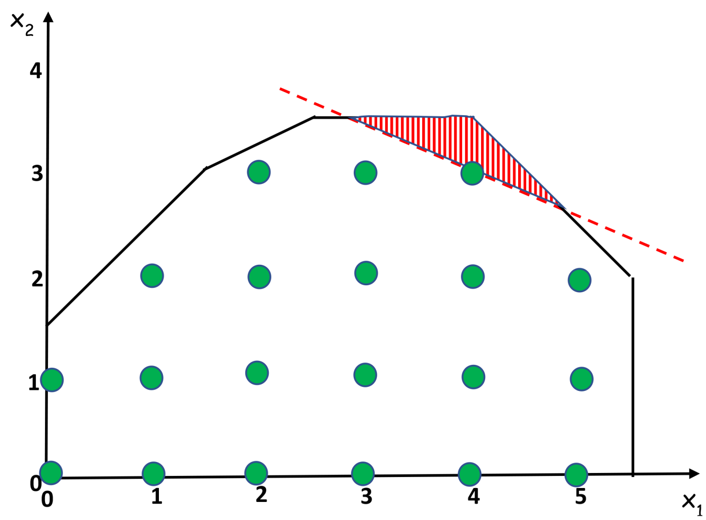

Figure 3: ILP feasible region with a cutting plane

partitioned in a *search tree* fashion into smaller and smaller subspaces, and lower and upper bounds are computed on the optimal solution value of the subproblem associated with each subspace. In a maximization context, those subproblems for which the upper bound is at most equal to the value of the best-known feasible solution (incumbent) can be discarded (fathomed). The process ends when it has been demonstrated that no subproblem can yield a better solution than the incumbent. The three main ingredients of branch-and-bound are *separation* (partitioning the space into subproblems), *branching* (selecting which subproblem to solve next), and *bounding* (computing lower and upper bounds on the optimum). Over the years, many heuristic rules have been suggested for efficient implementations of branch-and-bound algorithms. For references, see Geoffrion & Marsten (1972); Garfinkel (1979); Linderoth & Savelsbergh (1999); Johnson, Nemhauser & Savelsbergh (2000).

I will now sketch the Land and Doig algorithm using a minimum of notation. Here an "integer variable" denotes a variable that must take an integer value in a feasible solution. At the root node of the search tree the algorithm solves by the simplex method a linear program by considering all variables as continuous. If all integer variables take an integer value, the algorithm then terminates with a feasible and optimal solution. Otherwise, an integer variable x_k taking a fractional value $\bar{x}_k$ is selected for branching and a subproblem is created for each value of $x_k = l_k, \ldots, \lfloor \bar{x}_k \rfloor, \lceil \bar{x}_k \rceil, \ldots, u_k$, where $[l_k, u_k]$ is the domain of the variable x_k (Figure 4). The objective function is piecewise linear in x_k and attains a maximum at $\bar{x}_k$. By extrapolating this function in the left and right directions, one obtains an upper bound on the maximal profit attainable for each of the two branches $x_k = \lfloor \bar{x}_k \rfloor$ and $x_k = \lceil \bar{x}_k \rceil$, and indeed for all other branches. Figure 5 depicts this computation. If any of these two branches corresponds to a dominated solution because the upper bound value does not exceed that of the incumbent, it is fathomed, together with its neighbours which are then also dominated because the objective function is concave. This process is reiterated starting with the branches $x_k = \lfloor \bar{x}_k \rfloor$ and $x_k = \lceil \bar{x}_k \rceil$ if they are unfathomed, all other branches being kept for further potential exploration. This process ends with a provably optimal solution or with a proof that the problem is infeasible, assuming that the search tree can be fully explored within a given computing time.

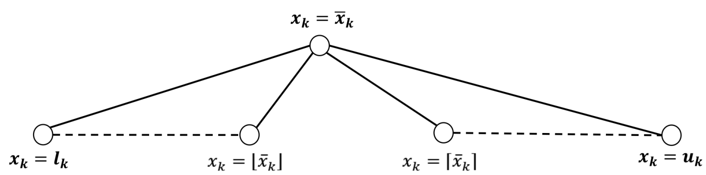

Figure 4: Branching on the integer variable in the feasible domain $[l_k, \ldots, \lfloor \bar{x}_k \rfloor, \lceil \bar{x}_k \rceil, \ldots u_k]$.

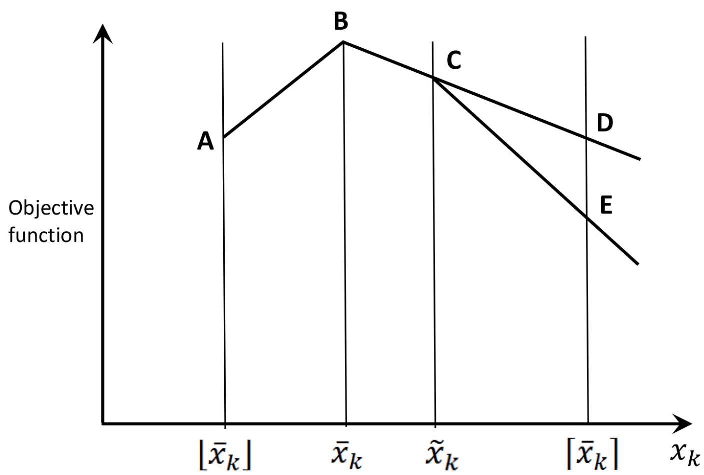

Figure 5: Computation of upper bounds A and D for the branches $x_k = \lfloor \bar{x}_k \rfloor$ and $x_k = \lceil \bar{x}_k \rceil$ obtained by extrapolation. The objective value at $\bar{x}_k$ is B. The upper bound computation for $\lceil \bar{x}_k \rceil$ does not take into account the pivot and change of slope occuring at $\tilde{x}_k$ (corresponding to C) and outputs the upper bound D which is weaker than E.

The authors called their algorithm "an automatic method", very much in the spirit of what Dantzig was advocating. Ironically, all their computations were performed by hand and the intermediate results were stored in paper form, which constituted quite a feat since the algorithm involves the solution of multiple linear programs. As put by Land & Doig (2010): "We were very well aware that the solution of this type of problem required electronic computation, but unfortunately LSE at that time did not have any access to such a facility. However, we had no doubt that using the same approach to computing could be achieved, if rather painfully, on desk computers, which were plentifully available. We became quite skillful at doing vector operations by multiplying with the left hand, and adding and subtracting with the right hand on another machine! Storage of bases and intermediate results did not present a limitation since it was all simply recorded on paper and kept in a folder. Hence we found it efficient to pursue each branch of our tree until its bound was no longer the best bound. To that extent our implementation was not exactly as we would later come to code it on a computer. It was efficient to make an estimate on the next bound in each direction of a branch before putting it aside for further development".

According to Balinski (1965), the Land and Doig algorithm had not as of February 1964 been tested computationally, save by hand. The first computer implementation appears to be that of Beale & Small (1965) (Magnanti, 2021).

Land & Doig (1960) are widely credited for the invention of branch-and-bound, but it is more accurate to say that what they developed is the first LP-based branch-and-bound algorithm for MILPs. The term *branch-and-bound* was coined three years later by Little et al. (1963) in relation to the asymmetric TSP, and Eastman (1958) had previously implemented in his Ph.D. thesis a branch-and-bound algorithm for the same problem. According to Cook (2012), the notion of branch-and-bound had already been laid out in an earlier paper by Markowitz & Manne (1957). However, these authors only provided ideas and did not propose a workable algorithm.

5 Further developments

The Land & Doig (1960) branch-and-bound algorithm lends itself to many variants depending on the rules applied for separation, branching, and bounding. Of particular interest is the dichotomous separation rule $x_k \leq \lfloor \bar{x}_k \rfloor$ or $x_k \geq \lceil \bar{x}_k \rceil$ proposed by Dakin (1965) which seems more efficient than partitioning the feasible space by creating a branch for each integer value of x_k within its domain $[l_k, u_k]$, as was done by Land and Doig. Most MILP branch-and-bound algorithms are now based on Dakin's rule, which was also implemented in the Land-Powell codes (Land & Powell, 1973).

Multiple computational studies have also been performed on various branching and bounding options (see, e.g., Lawler & Wood (1966); Mitten (1970); Linderoth & Savelsbergh (1999); Alvarez, Louveaux & Wehenkel (2017); Lodi & Zarpellon (2017)). In the words of Garfinkel (1979), "[...] the branch and bound concept is so simple that it allows ample room for the inclusion of many heuristic rules which can be devised for any particular instance of a combinatorial problem. These rules tend to be of little theoretical interest, but along with the cleverness of the implementation they are often the key elements in determining success or failure of these techniques".

In the early 1970s I was a Ph.D. student of Ailsa Land at LSE, together with a few others (see Laporte (2021) for an account of this period). We all had access to the Land-Powell software (Land & Powell, 1973) which was not conceived to be used as a black box but was rather meant to be modified with the aim of experimenting with new algorithmic ideas. Ailsa Land suggested to some of us that Gomory cuts, and indeed other types of linear inequalities, could be embedded within a branch-and-bound framework. This concept is now known as branch-and-cut and underlies several sophisticated solution methodologies for discrete optimization problems (Magnanti, 2021). Panagiotis Miliotis, who was also a Land student, was the first researcher to successfully combine branch-and-bound and Gomory cutting planes for the solution of the symmetric TSP (Miliotis, 1976). Branch-and-cut is now the most successful solution methodology for this problem (Applegate et al., 2006).

Related sophisticated solution frameworks have since emerged from the solution of combinatorial optimization problems involving huge numbers of discrete variables and constraints. These are decomposition algorithms that require the solution of multiple LPs or MILPs. One such algorithm is branch-and-price, often referred to as column generation (see, e.g., Barnhart et al., 1998), and another is branch-price-and-cut (or branch-cut-and-price) (see, e.g., Fukusawa et al., 2006). Essentially, branch-and-cut relaxes some constraints of the problem, branch-and-price relaxes variables, and branch-price-and-cut relaxes constraints and variables.

6 Conclusion

The seminal branch-and-bound algorithm of Land and Doig has had a major impact on the field of integer programming. The two most popular commercial MILP solvers, CPLEX and Gurobi, are both based on this methodology. Branch-and-bound has been qualified as "the most well-known method for discrete optimization" (Lodi & Zarpellon, 2017) and, more recently, Magnanti (2021) wrote: "The original approach for solving these problems [integer programs] which remains in place until today, is the use of branch and bound, introduced by Land & Doig (1960)". In fact, it remains to this day the *only* available methodology for the solution of generic mixed integer linear programs.

References

Alvarez, A.M., Louveaux, Q. & Wehenkel, L. (2017). A machine learning-based approximation of strong branching. *INFORMS Journal on Computing* **29**, 185–195.

Applegate, D.L., Bixby, R.E., Chvátal, V. & Cook, W.J. (2006). *The Traveling Salesman Problem. A Computational Study*. Princeton, NJ: Princeton University Press.

Balinski, M.L. (1965). Integer programming: Methods, uses, computation. *Management Science* **12**, 253–313.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsberg, M.W.P. & Vance, P.H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**, 316–329.

Beale, E.M.L. & Small, R.E. (1965). Mixed integer programming by a branch and bound technique. In *Proceedings of the IFIP Congress*, ed. W. Kalench. Macmillan, pp. 450–451.

Cook, W.J. (2012). Markowitz and Manne + Eastman + Land and Doig = branch and bound. *Documenta Mathematica. Extra Volume ISMP*, 227–238.

Dakin, R.J. (1965). A tree-search algorithm for mixed integer programming problems. *The Computer Journal* **8**, 250–255.

Dantzig, G.B. (1951). Maximization of a linear function of variables subject to linear inequalities. In *Activity Analysis of Production and Allocation*, ed. T.C. Koopmans. New York: Wiley, pp. 339–347.

Dantzig, G.B. (1960). On the significance of solving linear programming problems with some integer variables. *Econometrica* **28**, 30–44.

Dantzig, G.B. (1963). *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.

Doig, A.G. & Belz, M.H. (1956). *Report on the trim problem for May, 1956*. Department of Statistics, University of Melbourne.

Eastman, W.L. (1958). Linear programming with pattern constraints. Ph.D. thesis, Harvard University.

Fourier, J.B.J. (1826a). Analyse des travaux de l'académie royale des sciences, pendant l'année 1823. *Partie mathématique, Histoire de l'Académie Royale des Sciences de l'Institut de France* **6**, xxix–xli.

Fourier, J.B.J. (1826b). Solution d'une question particulière du calcul des inégalités. *Nouveau Bulletin des Sciences par la Société Philomatique de Paris*, 99–100.

Fukusawa, R., Longo, H., Lysgaard, J., Aragão, M.P.D., Reis, M., Uchuoa, E. & Werneck, R.F. (2006). Robust branch-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming* **106**, 491–511.

Garfinkel, R.S. (1979). Branch and bound methods for integer programming. In *Combinatorial Optimization*, eds. N. Christofides, A. Mingozzi, P. Toth & C. Sandi. Chichester: Wiley, pp. 1–20.

Geoffrion, A.M. & Marsten, R.E. (1972). Integer programming algorithms: A framework and state-of-the-art survey. *Management Science* **18**, 465–491.

Gomory, R.E. (1958). Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society* **64**, 275–278.

Johnson, E.L., Nemhauser, G.L. & Savelsbergh, M.W.P. (2000). Progress in linear programming-based algorithms for integer programming: an exposition. *INFORMS Journal on Computing* **12**, 2–23.

Land, A.H. (1956). An application of the techniques of linear programming to the transportation of coal. Ph.D. thesis, University of London.

Land, A.H. (2021a). *Ailsa's writings*. Unpublished document provided by her widower Frank Land.

Land, A.H. (2021b). The solution of some 100-city travelling salesman problems. *EURO Journal of Computational Optimization* **9**, 100017.

Land, A.H. & Doig, A.G. (1960). An automatic method of solving discrete programming problems. *Econometrica* **28**, 497–520.

Land, A.H. & Doig, A.G. (2010). An automatic method for solving discrete programming problems. In *50 years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*, eds. M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi & L. Wolsey. Berlin Heidelberg: Springer-Verlag, pp. 105–132.

Land, A.H. & Powell, S. (1973). *Fortran Codes for Mathematical Programming: Linear, Quadratic and Discrete*. New York: Wiley.

Land, A.H. & Powell, S. (2007). A survey of the operational use of ILP models. *Annals of Operations Research* **149**, 147–156.

Laporte, G. (2021). Some contributions of Ailsa H. Land to the study of the traveling salesman problem. *EURO Journal of Combinatorial Optimization* **9**, 100018.

Lawler, E.L. & Wood, D.E. (1966). Branch-and-bound methods: A survey. *Operations Research* **14**, 699–719.

Linderoth, J.T. & Savelsbergh, M.W.P. (1999). A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing* **11**, 177–187.

Little, J.D.C., Murty, K.G., Sweeney, D.G. & Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations Research* **11**, 972–989.

Lodi, A. & Zarpellon, G. (2017). On learning and branching: A survey. *TOP* **25**, 207–236.

Magnanti, T.L. (2021). Optimization: From its inception. *Management Science* **67**, 5349–5363.

Markowitz, H.M. & Manne, A.S. (1957). On the solution of discrete programming problems. *Econometrica* **25**, 84–110.

Miliotis, P. (1976). Integer programming approaches to the travelling salesman problem. *Mathematical Programming* **10**, 367–378.

Miliotis, P. (1978). Using cutting planes to solve the symmetric travelling salesman problem. *Mathematical Programming* **15**, 177–188.

Mitten, L.G. (1970). Branch-and-bound methods: General formulation and properties. *Operations Research* **18**, 24–34.

Orchard-Hays, W. (1955). *RAND Code for the Simplex*. Research Memorandum RM-1440, The RAND Corporation, February.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Chichester: Wiley.

Shier, D.R. (2022). A.H. Land, EURO Gold Medalist (1927–2021). *European Journal of Operational Research* **296**, 1–2.

Silberberg, V. (2019). Richard and Alison Harcourt. *Profile*, The Leo Baeck Centre for Progressive Judaism. February–March.