Tanzania Journal of Science 46(2): 281-289, 2020
ISSN 0856-1761, e-ISSN 2507-7961

Simulated Annealing Algorithm for the Linear Ordering Problem: The Case of Tanzania Input Output Tables

Allen R. Mushi
Department of Mathematics, University of Dar es Salaam, P. O. Box 35062, Dar es Salaam, Tanzania. E-mail: allenmushi66@gmail.com

Received 2 March 2020, Revised 4 May 2020, Accepted 6 May 2020, Published June 2020

Abstract

Linear Ordering is a problem of ordering the rows and columns of a matrix such that the sum of the upper triangle values is as large as possible. The problem has many applications including aggregation of individual preferences, weighted ancestry relationships and triangulation of input-output tables in economics. As a result, many researchers have been working on the problem which is known to be NP-hard. Consequently, heuristic algorithms have been developed and implemented on benchmark data or specific real-world applications. Simulated Annealing has seldom been used for this problem. Furthermore, only one attempt has been done on the Tanzanian input output table data. This article presents a Simulated Annealing approach to the problem and compares results with previous work on the same data using Great Deluge algorithm. Three cooling schedules are compared, namely linear, geometric and Lundy & Mees. The results show that Simulated Annealing and Great Deluge provide similar results including execution time and final solution quality. It is concluded that Simulated Annealing is a good algorithm for the Linear Ordering problem given a careful selection of required parameters.

Keywords: Combinatorial Optimization; Linear Ordering Problem; Simulated Annealing; Triangulation; Input Output tables

Introduction

Linear Ordering Problem (LOP) is a problem of ordering the rows and columns of a matrix in such a way that the sum of the upper triangle values is as large as possible. The problem has many applications including aggregation of individual preferences (Hurdy 2008), weighted ancestry relationships (Glover et al. 1974), scheduling with preferences (Boenchendorf et al 1982), triangulation of input-output matrices in economics (Chenery and Watanabe 1958) and many others. In economics, the sectors of the economy are normally divided into n sectors. A matrix called input-output table is constructed in such a way that the entries represent the amount of deliveries from one sector to another. The ordering of rows and columns of the matrix in such a way as to maximize the sum of entries in the upper triangle is called triangulation problem and is a direct application of the LOP. The triangulated matrix provides interesting economic interpretations and comparisons between countries (Grötschel et al. 1984).

LOP is normally modeled as a weighted directed graph where the task is to find a complete acyclic tournament with highest weight. That is, given a complete digraph $D_n = (V_n, A_n)$ of n nodes with a non-negative weight function $C: A \rightarrow R_+$, find an acyclic sub digraph of maximum total weight. Mathematically, it can be represented as an integer programming problem as in the

following formulation by Martí and Reinelt (2011):

Since the solution is a diagraph, define

$$x_{ij} = \begin{cases} 1 \text{ if } (i,j) \in D_n \\ 0 \quad \text{Otherwise} \end{cases}$$

Then the problem is formulated as follows:

$$Maximize \sum_{\substack{i,j \\ i \neq j}} c_{ij} x_{ij}$$

subject to

$$x_{ij} + x_{ji} = 1, \text{ for all } 1 \le i \le j \le n,$$

$$y_{ij} + y_{jk} + y_{ki} \le 2, \forall i, j, k \in V_n, j \neq k$$

$$x_{ij} \in \{0,1\} \text{ for all } 1 \le i, j \le n, i \neq j$$

Where: c_{ij} are weights in the matrix (input-output table in this case) and the constraint ensures that only one of x_{ij} or x_{ji} is selected in a solution. However, the solution space is exponential, making the problem highly complex. Many researchers have been working on this problem through exact methods. The approach is to design a mathematical model and solve using available exact algorithms; the most common being cutting planes and branch and bound. Since the problem is NP-Hard, no algorithm is known that can solve a general problem to optimality within reasonable time. However, many efforts have been made in the exact methods to try and improve the size of problems that can be solved to optimality. The integer constraints are relaxed in the LOP model and cutting planes added to prune infeasibilities from the relaxation. The deepest cutting planes are called facets and once identified they can greatly improve performance of the cutting planes method. Reinelt (1985) presented a number of facets to the LOP and proposed a cutting plane method coupled with branch and bound. The cutting planes component involved the generation of facets that included 3-dicycles, k-fences and Mobius ladder. The relaxation then became;

$$Maximize \sum_{\substack{i,j \\ i \neq j}} c_{ij} x_{ij}$$

subject to

$$x_{ij} + x_{ji} = 1, \text{ for all } 1 \le i \le j \le n,$$

$$x_{ij} \ge 0, \text{ for all } 1 \le i, j \le n, i \neq j,$$

$$x(C) \le 2, \text{ for all 3-Dicycles C in } A_n,$$

$$x(F) \le 7, \text{ for all 3-fences D = (V, F) in } D_{n,}$$

$$x(M) \le 8, \text{ for all Möbius ladders}$$

$$\text{D = (V, M) in } D_n$$

A test was done on small size instances which demonstrated great reduction in the size of the resulting polyhedron and thereby increasing the chance for obtaining an optimal solution when solving the relaxation through Linear Programming methods. Mushi (2005) implemented this algorithm on a real life problem of the Irish input-output tables and managed to solve a problem of 41×41 sectors to optimality. Other researchers have also been working on the problem by following the same exact procedures. Méndez-Díaz et al. (2019) analyzed a general LOP through integer programming by defining the convex hull, proposed a set of facets and applied branch and cut algorithm which is a combination of branch and bound and cutting planes algorithms. They provided extensive experimental results for randomly generated data of different structures where the results performed well on the generated samples. Mitchel and Botchers (1996) applied the primal-dual interior point cutting plane method to solve real world problems. They applied their method to input output tables of countries in the European Community and USA from 1954 to 1979 and produced some good results. The challenges of the branch and cut methods include the branching strategy that heavily affects performance. Agrawal et al (2019) proposed a primal heuristic procedure to generate feasible

integer solutions to be applied in the branch and bound algorithm. They presented results on standard problems with improved performance. However, the problem is NP-hard and therefore heuristic approaches are necessary for large problems which are typical of real-world situations.

Many heuristic approaches have been proposed from early years of the problem studies including the work by Chenery and Watanabe (1958), Aujac and Masson heuristic, Becker heuristic and many others are described in Martí and Reinelt (2011) with applications to mostly randomly generated problems. Recent heuristic techniques have also been presented in various articles with some success. Laguna et al. (1999) presented a Tabu Search algorithm for the LOP and applied intensification and diversification strategies to improve performance where they managed to solve 49 instances from LOP library (LOLIB). The results outperformed the work by Chanas and Kobylański (1996). Duarte et al. (2011) also applied Tabu Search to the LOP with cumulative costs. They experimented with 218 instances and managed to show that the Tabu Search procedure performed better in terms of solution quality with reasonable computing-time. Garcia et al. (2006) applied the Variable Neighbourhood Search (VNS) algorithm which is based on systematic change of neighbourhoods in local search procedures in search for global convergence. Experimentation with 249 instances from LOLIB revealed that the strategy is capable of producing good solutions. Scatter search which is a population based method has been applied by Campos et al. (1999) to the LOP and compared the results with Tabu Search algorithm using the LOLIB Library. The results showed that a careful scatter search implementation compares well with Tabu Search in terms of performance. Garcia et al (2019) presented hybrid heuristics that combines iterated local search and exact methods to the LOP. They applied their results to 78 problems in the LOLIB and managed to obtain better results for 77 out of them. Other heuristic algorithms include Local Search (Sakuraba and Yagiuri 2010), Multi-level algorithm (Safro et al 2009), Genetic algorithm (Cergibozan and Taşan 2019), Differential evolution algorithms (Baioletti et al 2020), Block-insertion (Qian et al 2020) and Memetic algorithms (Song et al 2018). A survey of heuristic algorithms for the LOP is given by Martí et al. (2012) together with a benchmark library LOLIB for further exploration of methods.

Simulated Annealing (SA) algorithm is a popular method which has been widely applied in combinatorial optimization problems. However, it has been hardly applied to the LOP; the author has been able to find one article which applied SA to the LOP and this is the work by Martí et al. (2012) when they surveyed heuristics and compared the results on the LOLIB Library. Furthermore, only one heuristic procedure has been applied to the Tanzanian Input-Output tables which have 79 sectors of the economy. That is the work by Amos and Mushi (2015) that applied the Great Deluge Algorithm. It is worth applying a different heuristic method and compare the results; a work that is the main objective of this article. Simulated Annealing has been chosen because of its popularity with many successful implementations to other combinatorial optimization problems.

The rest of the paper is organized as follows: a description of the Simulated Annealing algorithm is provided with its adaptation to the LOP. Then Summary of results is presented with comparison to previous work on the same data and finally a conclusion and future research directions are presented.

Simulated Annealing and the LOP implementation

Simulated Annealing mimics the cooling process of an object from gaseous to solid state. The cooling curve follows a particular path which is not always decreasing and the

process is called annealing. SA is one of global heuristic techniques that try to avoid falling into a local solution by accepting bad solutions by a probability function that depends on temperature and solution improvement. A good description is provided by Reeves (1993) where a general algorithm is as shown in the following pseudo-code:

Simulated_Annealing_Algorithm

Initialize parameters (Temperature T, freezing point F);
Get Initial Solution So;
While temperature T > freezing point F {
Get Solution in the neighborhood of S ($S \in N(So)$);
Calculate $\sigma = f(S) - f(So)$; //f is the objective function
If ($\sigma > 0$)
Accept new solution ($So = S$);
Else {Generate a random value x between 0 and 1;
If ($x < e^{-\frac{\sigma}{T}}$)
Accept new solution ($So = S$);
Else
Reject new solution
}
Update temperature ($T = \alpha(T)$);
}
End Simulated Annealing

The main challenges in the adaption of the SA to LOP are associated with the choice of solution structure, getting initial solution, neighbourhood structure, the kind of moves to be used, the cooling schedule, initial temperature, freezing point and choice of parameters in the cooling schedule as discussed next.

Solution structure and initial solution

A quick initial solution can easily be found by picking the upper triangle values of the original un-triangulated input-output table. This solution guarantees feasibility by making sure that it does not contain cycles and covers all nodes of the matrix. This is similar to the structures used in Amos and Mushi (2015) and is presented as follows:

$s_0 = (y_{ij}^0)$ where $y_{ij} = \begin{cases} 1 & \text{for all } i < j \\ 0 & \text{Otherwise} \end{cases}$

and therefore the objective function becomes

$$f(S_0) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} c_{ij} y_{ij}^0$$

Move selection

The same move type as in Amos and Mushi (2015) is applied where two randomly selected nodes i and j are swapped to get a new configuration. The new move is then checked for violation of constraints and accepted when no violation is found, otherwise it is rejected and a new set is swapped randomly. The move is feasible it satisfies two sets of constraints;

$$y_{ij} + y_{ji} = 1, \text{ for all } 1 \le i \le j \le n,$$

$$y(C) \le 2, \text{ for all 3-Dicycles C in } A_n$$

The first set of constraints is always satisfied because of the structure of move selection which ensures that only one of y_{ij} and y_{ji} becomes 1 in any solution. The second set of constraints (3-dicycles inequality) is expressed in the form;

$$y_{ij} + y_{jk} + y_{ki} \le 2, \forall i, j, k \in V_n, j \ne k.$$

Therefore, for any swapped indices i, j the algorithm uses another index $k \neq j$ and checks for violations. If no violations are found, then the move is feasible and is taken as a candidate move in the neighbourhood. Otherwise the move is rejected as soon as the first violation is detected. The process is repeated until a candidate move has been found.

Cooling schedules

A good cooling schedule must allow an ample time for exploration of solution space in the initial levels by accepting bad moves

and converges to an extreme point without wasting too much time on the final levels. Thus a cooling schedule that brings the temperature parameter from large initial value to freezing point too quickly may result into low solution quality because of the possible quick convergence to a local maximum without sufficient exploration of the search space. On the other hand, lowering the temperature too slowly may result into a large computation time which may not be necessary. Several authors have proposed different cooling schedules with different characteristics. Inspired by the arithmetic-geometric progression which is defined as a recurrence affine relation between consecutive terms of the sequence, Mahdi et al. (2017) defined a function $T(t) = \alpha t + b$. For convergence, the value of α must satisfy the relation $|\alpha| < 1$ and converges to $\frac{b}{1-\alpha}$ when the condition is satisfied. However, this is very similar to pure geometric progression with only a constant shift (b) in the schedule. Geman and Geman (1984) introduced a logarithmic cooling schedule. The schedule follows logarithmic distribution that asymptotically converges towards the global minimum and is defined as $T(t) = \frac{c}{\log(t+1)}$. The value of c is a positive constant independent of t but should be greater than the largest energy barrier in the problem (Mahdi et al. 2017, Pepra et al. 2017). However, according to Mahdi et al. (2017), this schedule converges very slowly and therefore requires a long computation time. The choice of a cooling schedule is therefore essential in the performance of SA.

In this article the following cooling schedules are compared due to their varying features and success in other SA implementations;

Linear –was introduced by Kirkpatrick et al. (1983) and is described as follows: given an initial temperature T_0, the temperature is reduced linearly by following the function $F(t) = T_0 - \alpha t$, where α is a decay rate whose value must be positive but close to zero for slow reduction. Different values of α are tested and results compared.

Geometric–was introduced by van Laarhoven and Aarts (1987) and follows a geometric function $T(t) = \alpha t$, where α is usually in the interval [0.8 – 0.9].

Lundy & Mees–came from an observation by Lundy and Mees (1986) that the stationary distributions between successive temperatures must be closed and therefore calling for alternate decrementing rules for selecting cooling factor. They proposed the cooling schedule as $T(t) = \frac{1}{t(1+\beta t)}$ where β is a very small constant; Aarts and van Laarhoven (1985) applied β in the range [0.5, 0.9] but lower values are possible depending on the characteristics of the problem.

Initial and final temperatures

Initial temperature is set to high value and experimented for the best value. This temperature must be high enough to explore all solution space. Final temperature (F) which is termed freezing point is chosen in such a way as to allow convergence without wasting unnecessary time. This is found through experimentation and may vary between cooling schedules.

Summary of Results

The algorithm was coded in C++ and run on a 3GHz processor PC, where three cooling schedules were experimented with different sets of parameters. The best parameters for the linear cooling schedule were; initial temperature (T_0 = 1,000) and freezing point (F = 0.1). The value of cooling rate (α) was found to be 0.0001 as shown in Table 1.

Table 1: Summary of results - Linear cooling

α	Solution	Sec.	Iterations
1	585,481	0	1
0.1	595,147	0.015	153
0.01	669,890	0.022	1,604
0.001	837,970	0.238	16,111
0.0001	839,842	2.442	161,173
0.00001	839,842	25.85	1,611,802

The best solution found is 839,842 after 161,173 iterations and 2.442 seconds which is a reasonable time. On geometric cooling schedule, the best initial temperature was 1,000,000 with freezing point 0.0001 and cooling rate $\alpha = 0.999$ as shown in Table 2.

Table 2: Summary of results - Geometric cooling

α	Solution	Sec.	Iterations
0.8	587,389	0	62
0.85	591,352	0	86
0.9	593,099	0	132
0.95	600,440	0	270
0.96	600,997	0	339
0.97	603,530	0	454
0.98	610,649	0.015	684
0.99	656,452	0.02	1,375
0.999	839,842	0.365	23,015
0.9999	839,842	2.091	138,149
0.99999	839,842	22.887	1,381,545

The best solution is the same as in previous case; however, the solution was obtained after 23,015 iterations with 0.365 seconds which is faster than linear cooling.

The Lundy & Mees cooling schedule results are as shown in Table 3 and were obtained with initial temperature of 1,000,000 and freezing point of 0.0001 and $\beta = 0.0004$.

Table 3: Summary of results – Lundy & Mees cooling

β	Solution	Sec.	Iterations
0.1	591,708	0	100
0.05	569,929	0	200
0.001	796,929	0.14	10,000
0.0005	839,693	0.31	20,000
0.0004	839,842	0.37	25,000
0.0003	839,842	0.5	33,334
0.0002	839,842	0.75	50,000

The convergence to the solution is similar in terms of iterations although linear schedule took more iterations and time to converge to the best solution. Figure 1 demonstrates the iteration steps during convergence to the best solution for the linear cooling schedule.

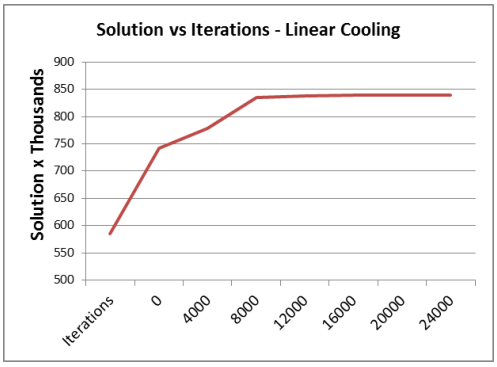

Figure 1: Iterations during solution search – Linear cooling.

In both cases the solution was obtained after very few seconds indicating that all three cooling schedules are useful for the Simulated Annealing to LOP. Figure 2 shows the solution search steps against time where Geometric and Lundy & Mees cases converged to the best solution within fractions of a second, while linear cooling converged after 2.4 seconds.

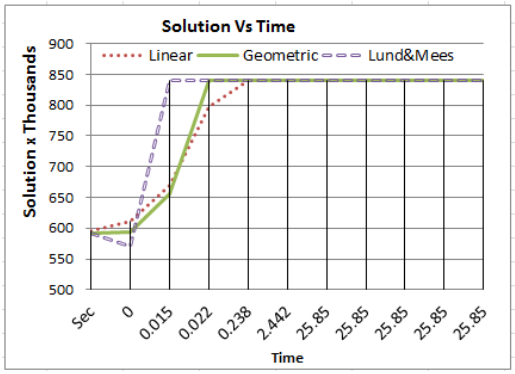

Figure 2: Solution search versus time

Although linear cooling is slightly slower than the other two cooling schedules it still performed well within tolerable range. The same solution is also found by Amos and

Mushi (2015) on great deluge algorithm, and therefore the same degree of linearity is observed with the same sector orders and similar time in seconds.

Lundy & Mees cooling schedule showed a slight fall in solution before improvements to convergence, which is a typical characteristic of Simulated Annealing where bad moves may be accepted in anticipation of better moves in future (Figure 2).

In general, the Simulated Annealing algorithm performed well for both cooling schedules, an evidence that it is a good algorithm for the LOP and has been able to provide a good solution to the Tanzanian input output tables within reasonable time.

Conclusion and Further Research Directions

The article intended to apply Simulated Annealing algorithm to the Linear Ordering Problem for the Tanzanian input output tables and compare results with previous work from Great Deluge Algorithm on the same data. To the best of knowledge, the algorithm has only been applied once to the standard library LOLIB data. The algorithm has been implemented and experimented with three different cooling schedules which are Linear, Geometric and Lundy & Mees. The results show that regardless of the choice of cooling schedule, the algorithm performed very well within a very short period of time. Linear cooling schedule is a little bit slower than the other two schedules but still gave a solution within 2.4 seconds which is tolerable. Simulated Annealing is therefore a good algorithm for the Linear Ordering Problem and has been able to generate the same solution as in the Great Deluge case. The results compare very well with the previous work including the solution, and therefore provide the same linear order and same degree of linearity (94.3%). More applications of LOP to real world problems are recommended such as aggregation of individual preferences and breaking ties in sports (Grötschel et al. 1984). There are variants of the LOP which have not been well explored, including Steiner Linear Ordering Problem (Magagnotti 2010), Checkpoint Ordering Problem (Hungerländer 2017) and Quadratic Linear Ordering Problem (Buchheim et al. 2010). Further studies in these variants especially applications of global heuristic techniques are recommended.

References

Aarts EHL and van Laarhoven P 1985 Statistical cooling: A general to combinatorial optimization problems. *Philips J. Res.* 40: 193-226.

Agrawal R, Iranmanesh E and Krishnamurti R 2019 Primal heuristic for the linear ordering problem. *8th International Conference on Operations Research and Enterprise Systems*, Prague Czech Republic. DOI: 10.5220/0007406301510156.

Amos M and Mushi AR 2015 Great Deluge Algorithm for the Linear Ordering Problem: The Case of Tanzanian Input-Output Table. *Int. J. Informat. Technol. Comput. Sci.* 07: 28-34.

Baioletti M, Milani A and Santucci V 2020 Variable neighborhood algebraic differential evolution: an application to the linear ordering problem with cumulative costs. *Informat. Sci.* 507: 37-52.

Boenchendorf K 1982 Reihenfolgenprobleme /Mean-flow-time sequencing. *Mathematical Systems in Economics* 74, Verlagsgruppe Athenaum, Hain, Scriptor.

Buchheim C, Wiegele A and Zheng L 2010 Exact algorithms for the quadratic linear ordering problem. *Inform. J. Comput.* 22(1): 168-177.

Campos V, Laguna M and Martí R 1996 Scatter Search for the Linear Ordering Problem. *New ideas in optimization*, McGraw-Hill Ltd. UK, 331–340.

Cergibozan C and Taşan AS 2019 Solving the linear ordering problem using a genetic algorithm with local search. In: Mula J, Barbastefano R, Díaz-Madroñero M and

Poler R (ed) *New Global Perspectives on Industrial Engineering and Management. Lecture Notes in Management and Industrial Engineering* Springer, Cham.

Chanas, S. and Kobylanski P 1996 A new heuristic algorithm solving the linear ordering problem. *Comput. Optim. Applicat.* 6: 191-205.

Chenery HB and Watanabe T 1958 International comparisons of the structure of production. *Econometrica* 26: 487-521.

Duarte A, Laguna M and Marti R 2011 Tabu search for the linear ordering problem with cumulative costs. *Comput. Optim. Applicat.* 48(3): 697-715.

Hungerländer P 2017 The checkpoint ordering problem. *Optimization* 66(10): 1699-1712.

Hurdy O 2008 NP-hardness results for the aggregation of linear orders into median orders. *Ann. Operat. Res.* 163: 63–88.

Garcia CG, Pérez-Britoa D, Campos V and Martí R 2006 Variable neighborhood search for the linear ordering problem. *Comput. Operat. Res.* 33: 3549-3565.

Garcia E, Ceberio J and Lozano JA 2019 Hybrid heuristics for the linear ordering problem. *2019 IEEE Congress on Evolutionary Computation (CEC)*. DOI: 10.1109/CEC.2019.8790280.

Geman, S and Geman D 1984 Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intel.* 6(6): 721-741.

Glover F, Klastorin T and Klingman D 1974 Optimal weighted ancestry relationships. *Manage. Sci.* 20: 1190-1193.

Grötschel M, Junger M and Reinelt G 1984 A cutting plane algorithm for the linear ordering problem. *Operat. Res.* 32(6): 1195-1220.

Kirkpatrick S, Gelatt JCD and Vecchi MP 1983 Optimization by Simulated Annealing. *Science* 220: 671-680.

Laguna M, Martí R and Campos V 1999 Intensification and diversification with elite tabu search solutions for the linear ordering problem, *Computers & Operations Research, 26(12)*: 1217-1230.

Lundy M and Mees A 1986 Convergence of an Annealing Algorithm. *Math.l Program.* 34: 111-124.

Magagnotti M 2010 The Steiner Linear Ordering Problem: Application to resource-constrained scheduling problems. *All Theses*, 811, https://tigerprints.clemson.edu/all_theses/811

Mahdi W, Medjahed SA and Quali M 2017 Performance analysis of simulated annealing cooling schedules in the context of dense image matching. *Computac. Sistem.* 21(3): 493-501.

Martí R, Reinelt G and Duarte A 2012 A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Comput. Optim. Applicat.* 51(3): 1297-1317.

Martí R and Reinelt G 2011 The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization. Springer-Verlag Berlin Heidelberg.

Méndez-Díaz I, Vulcano G and Zabala P 2019 Analysis of a generalized linear ordering problem via integer programming. *Discrete Appl. Math.* 271: 93-107.

Mitchel JE and Borchers B 1996 Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Ann. Operat. Res.* 62(1): 253-276.

Mushi AR 2005 The linear ordering problem: an algorithm for the optimal solution. *Afr. J. Sci. Technol. Sci. Engin. Ser.* 6(1): 51-64.

Peprah AK, Appiah SK and Amposah SK 2017 An optimal cooling schedule using a simulated annealing based approach. *Appl. Math.* 8: 1195-1210.

Qian Y, Lin J, Li D and Hu H 2020 Block-insertion-based algorithms for the linear ordering problem. *Comput. Operat. Res.* 115: 104861.

Reeves C 1993 Modern Heuristic Techniques for Combinatorial Problems. Blackwell Scientific Publications Oxford.

Reinelt G 1985 The Linear Ordering Problem: Algorithms and applications. Berlin Heldermann Verlag.

Safro I, Ron D and Brandt A 2009 Multilevel Algorithms for Linear Ordering Problems. *J. Exp. Algorithm. (JEA)* 13(4): doi.org/10.1145/1412228.1412232.

Sakuraba CS and Yagiuri M 2010 Efficient local search algorithms for the linear ordering problem. *Int. Trans. Operat. Res.* 17: 711-737.

Song J, Zhao H, Zhou T, Tao Y, Lü Z 2018 Solving the linear ordering problem via a memetic algorithm. In: Arai K, Bhatia R, Kapoor S (Ed) *Proceedings of the Future Technologies Conference (FTC) 2018. Advances in Intelligent Systems and Computing, 881*. Springer, Cham.

van Laarhoven PJM and Aarts EHL 1987 Simulated Annealing: Theory and Applications. *Reidel Publishing Company, Dordrecht.* https://doi.org/10.1007/978-94-015-7744-1.