A Probabilistic Tree-Based Representation for Non-convex Minimum Cost Flow Problems

Behrooz Ghasemishabankareh[1(✉)], Melih Ozlen[1], Frank Neumann[2], and Xiaodong Li[1]

[1] School of Science, RMIT University, Melbourne, Australia
{behrooz.ghasemishabankareh,melih.ozlen,xiaodong.li}@rmit.edu.au
[2] School of Computer Science, The University of Adelaide, Adelaide, Australia
frank.neumann@adelaide.edu.au

Abstract. Network flow optimisation has many real-world applications. The minimum cost flow problem (MCFP) is one of the most common network flow problems. Mathematical programming methods often assume the linearity and convexity of the underlying cost function, which is not realistic in many real-world situations. Solving large-sized MCFPs with nonlinear non-convex cost functions poses a much harder problem. In this paper, we propose a new representation scheme for solving non-convex MCFPs using genetic algorithms (GAs). The most common representation scheme for solving the MCFP in the literature using a GA is priority-based encoding, but it has some serious limitations including restricting the search space to a small part of the feasible set. We introduce a probabilistic tree-based representation scheme (PTbR) that is far superior compared to the priority-based encoding. Our extensive experimental investigations show the advantage of our encoding compared to previous methods for a variety of cost functions.

Keywords: Representation scheme · Genetic algorithm
Minimum cost flow problem · Mixed integer nonlinear programming

1 Introduction

Network flow problems have numerous applications in electrical and power networks, telecommunication, road and rail networks, and airline services [2]. Different types of network flow problems exist, e.g., the shortest path problem, the maximum flow problem, the assignment problem, the transportation problem, and the minimum cost flow problem (MCFP), among which MCFP is one of the most general cases with applications such as distribution problems, optimal loading of a Hopping aeroplane and the racial balancing of schools [2].

MCFPs can be formulated and solved by Linear Programming (LP) techniques, when the underlying cost function is linear or can be approximated by a linear function [17]. However, many real-world MCFPs are nonlinear and require

A. Auger et al. (Eds.): PPSN 2018, LNCS 11101, pp. 69–81, 2018.
https://doi.org/10.1007/978-3-319-99253-2_6

formulation using a nonlinear cost function, instead of a linear approximation. For example, in a transportation problem, the nonlinearity of a cost function is due to the economy of scale phenomenon, which occurs when cost per unit of the transportation flow decreases with an increasing amount of the total flow [7]. Many studies suggest the appropriateness of employing nonlinear cost functions in the network design problems [4,15].

Some attempts have been made in using genetic algorithms (GAs) to solve the network flow problems [1,7,13]. Among these works, the *representation scheme* plays a critical role in their success. Several representation schemes exist for the network flow problems such as variable-length encoding [19], fixed-length encoding [3], and priority-based representation (PbR) [16]. The most common representation scheme for solving MCFPs is PbR [8]. PbR scheme has been used to solve the shortest path problems, the transportation problems, as well as the network design problems [8,13,16]. Although PbR is widely used for solving network flow problems, it has some serious drawbacks (when dealing with MCFP), most noticeably its restriction on any search algorithm from reaching some parts of the feasible search space (see Sect. 2 for details).

To counteract the above limitations, in this paper we propose a probabilistic tree-based representation (PTbR) for solving nonlinear non-convex MCFP instances using the GA. The PTbR allows all possible feasible solutions to be generated, instead of being restricted to a small part of the feasible region (e.g., PbR scheme). This paper first examines the capabilities of PTbR and compare it with that of the PbR scheme. Then a comparative study is carried out on the performance of the GA employing these two different representation schemes on a set of 35 benchmark instances. This paper has the following contributions: (1) proposing a novel representation scheme (PTbR) to deal with MCFP; (2) providing a close examination between PTbR and PbR to find out which one is more effective for handling MCFPs; (3) conducting extensive experiments to compare the performance of the PTbR-based GA (PtGA) variants with the PbR-based GA (PrGA) for solving non-convex MCFP instances. We also compare our results with those of the mathematical solver packages.

The rest of the paper is structured as follows: Sect. 2 gives the preliminaries and Sect. 3 describes our proposed probabilistic tree-based representation and the GA employing PTbR scheme for solving MCFPs. The experimental studies are presented in Sects. 4 and 5 provides the conclusion.

2 Preliminaries

This section describes the problem definition, the PbR, and finally discusses the drawbacks of PbR. Let $G(N, A)$ be a network consisting of a set N of n nodes and a set A of m directed arcs. The maximum and minimum amount of flow on each arc (i, j) are equal to u_{ij} and 0, respectively. $b(i)$ denotes the amount of supply or demand for source or sink node. $b(i) > 0$ denotes that node i is a supply node and $b(i) < 0$ shows that node i is a demand node with a demand of $-b(i)$ and $b(i) = 0$ denotes the transshipment node i. Figure 1 shows an example of the

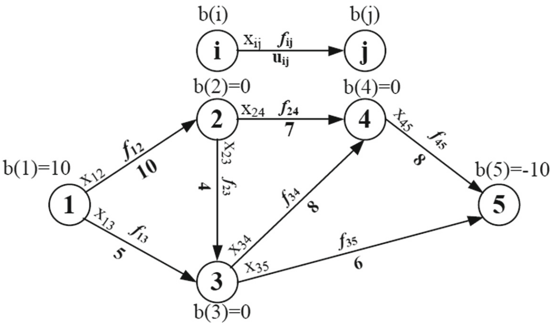


Fig. 1. An example of the MCFP ($n = 5$, $m = 7$).

MCFP with $n = 5$ nodes and $m = 7$ arcs, which has one supplier node ($b(1) = 10$) and one demand node ($b(5) = -10$). In this example, we aim to satisfy the demand by sending all supplies through the network while minimising the total cost. The integer flow on an arc (i,j) is represented by x_{ij} and the associated cost for the flow (x_{ij}) is denoted by $f_{ij}(x_{ij})$. The formulation of the MCFP is as follows [2]:

$$Minimise : z(\mathbf{x}) = \sum_{(i,j)\in A} f_{ij}(x_{ij}), \tag{1}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b(i) \qquad \forall \quad i \in N, \tag{2}$$

$$0 \le x_{ij} \le u_{ij} \quad \forall \quad (i,j) \in A, \tag{3}$$

$$x_{ij} \in Z \quad \forall \quad (i,j) \in A, \tag{4}$$

where Eq. 1 minimises the total cost through the network. Equation 2 is a flow balance constraint which states the difference between the total outflow (first term) and the total inflow (second term). The flow on each arc should be between an upper bound and zero (Eq. 3), and finally all the flow values are integer numbers (Eq. 4). In this paper we consider the following assumptions for the MCFP: (1) the network is directed; (2) there are no two or more arcs with the same tail and head in the network; (3) the single-source single-sink MCFP is considered; (4) the total demands and supplies in the network are equal, i.e., $\sum_{i=1}^{n} b(i) = 0$.

2.1 Priority-Based Representation

Priority-based representation (PbR) is the most commonly-used representation method for MCFPs [8]. In order to represent a candidate solution for an MCFP, PbR lets the number of genes to be equal to n and the value of each gene is generated randomly between 1 and n, which represents the priority of each node for constructing a path among all possible nodes [8]. Figure 2a illustrates the PbR chromosome for the network presented in Fig. 1. In order to obtain a feasible solution, a two-phase decoding procedure is followed. In phase I, a path is

(a) An example of a chromosome for the network in Fig. 1.

(b) A feasible solution for the given chromosome.

Fig. 2. The PbR chromosome and its corresponding solution.

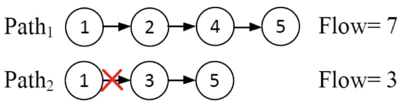


Fig. 3. A feasible solution that PbR fails to represent (for the network in Fig. 1).

generated based on the priorities and the maximum possible flow is sent through the generated path in phase II. After sending the flow on the network, the upper bound (u_{ij}), supply and demand should be updated. If the supply/demand is not equal to 0, the next path should be generated. The above procedure repeats until all demands are satisfied. Figure 2b presents a feasible solution for the given chromosome in Fig. 2a.

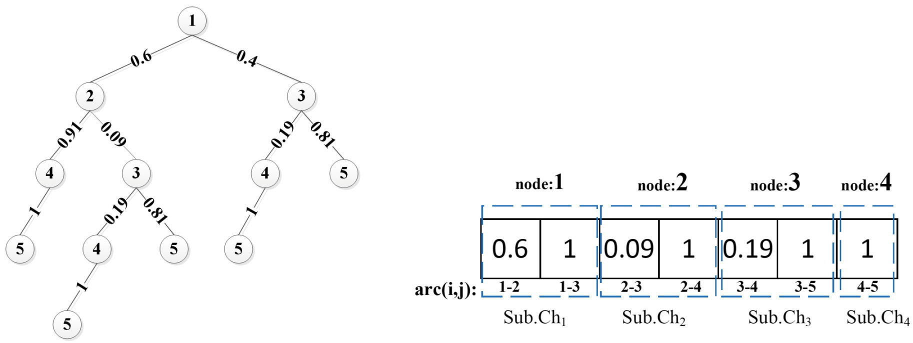
Although PbR has been commonly used in the network flow problems, it has some limitations in representing the full extent of the feasible space for MCFP. Figure 3 shows an example (for the network presented in Fig. 1) that PbR is unable to represent. Here the first path is generated as follows: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$. Since in Path_1 after node 1, node 2 is selected, it shows that node 2 has a higher priority than node 3. Hence, if arc (1, 2) is not saturated, PbR will not allow any flow to be sent through arc (1, 3), essentially blocking this possibility completely (Fig. 3, Path_2). This means that PbR is unable to represent a potential feasible solution such that the flow would go through arc (1, 3) (as shown in Fig. 3). Another limitation for PbR is that each time a path is generated, we are supposed to send the maximum possible amount on the generated path. These limitations would restrict a search algorithm from reaching the full extent of the feasible space.

3 Proposed Method

Representation plays a critical role before applying an optimisation algorithm, and this applies to GA too. In this section we first propose a probabilistic tree-based representation (PTbR) scheme for solving MCFPs, which alleviates the deficiency of using PbR. Then we describe the GA employing PTbR for solving MCFP instances.

3.1 Probabilistic Tree-Based Representation

To counteract the above-mentioned limitations of the PbR, we propose the PTbR scheme, where a probability tree is adopted to represent a potential MCFP



(a) A probability tree structure. (b) The PTbR based on the probability tree.

Fig. 4. Probability tree and its corresponding PTbR for the network in Fig. 1.

solution. Unlike the PbR scheme which is restricted to a small part of the feasible space, the PTbR is able to represent all possible feasible solutions. Figure 4a shows an example of the probability tree for the network presented in Fig. 1. Here, the probability of each successor node to be selected is defined on each branch.

The tree structure can be converted to a chromosome with several sub-chromosomes. Figure 4b shows the PTbR chromosome converted from the probability tree presented in Fig. 4a. The PTbR chromosome has $n - 1$ sub-chromosomes (Sub.Ch) and the value of each gene is a random number between 0 and 1 which is then accumulated to 1 in each sub-chromosome. In order to obtain a feasible solution from PTbR, in phase I, a path is first constructed, and then a feasible flow is sent through the constructed path in phase II. For example, to obtain a feasible solution for the chromosome in Fig. 4b, we generate the first path from node $i = 1$ (Sub.Ch$_{i=1}$). A random number is generated in [0,1] ($rand = 0.2$), and since $0 \leq rand = 0.2 \leq 0.6$, we move through arc (1,2) and node 2 is selected. From node 2 (Sub.Ch$_{i=2}$) another random number is generated ($0.09 \leq rand = 0.85 \leq 1$) and the selected successor node is 4. From node 4 the only available node is 5. Hence, the following path is generated: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$.

In Phase II, we attempt to send a feasible flow through the generated path. First the capacity of the generated path is defined ($U = min\{u_{12} = 10, u_{24} = 7, u_{45} = 8\} = 7$). Then, there are three possible approaches to send a feasible flow on the generated path: (1) send a random flow between 1 and U (random(**R**)); (2) send a flow 1-by-1 (one-by-one (**O**)); (3) send the maximum possible amount of the flow on the generated path (maximum(**M**)), which is the same as PbR. In the above example, we follow the first approach (random(**R**)) and after calculating $U = 7$, we send a random flow in [1, 7] (e.g., flow = 6) and the network, supply and demand are updated. Since the demand has not been fully met (i.e., not equal to 0 yet), the above procedure is repeated.

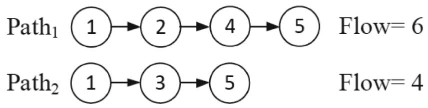


Fig. 5. A feasible solution generated based on the PTbR chromosome in Fig. 4b.

Figure 5 shows a feasible solution for the chromosome presented in Fig. 4b. Note that in Fig. 5, after generating Path_1, although arc (1,2) is not saturated, the second path picks node 3 as the successor of node 1, unlike the PbR. This example illustrates that PTbR allows all potential solutions to be generated probabilistically, instead of being restricted by using PbR.

3.2 Genetic Algorithm with PTbR

This section describes the GA employing the new representation scheme PTbR for solving MCFPs, i.e., PtGA. The key distinction between the PtGA and the PbR-based GA (PrGA) is that PrGA employs the PbR [8]. This PtGA can be described by the following procedure:

Initialisation: First a population with *pop_size* individuals (chromosomes) is randomly generated. The process of creating a chromosome based on the PTbR is explained in Subsect. 3.1.

Crossover and Mutation: In order to explore the feasible region, crossover and mutation operators are applied to create the new offspring at each generation. For PtGA, a two-point crossover operation is applied, where two blocks (sub-chromosomes) of the selected chromosome (parents) are first randomly selected. Then, two parents swapping the selected sub-chromosomes to generate new offspring. To perform mutation for PtGA, first a random parent is selected and the randomly chosen sub-chromosome is regenerated to create a new offspring.

Fitness Evaluation and Selection: For each chromosome in the population, after finding a feasible solution ($\mathbf{x}$) by applying the decoding procedure for PTbR, the value of cost function is evaluated using the following equation: $Minimize : z(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} f(x_{ij})$. After calculating the fitness values for all individuals in the population, the tournament selection procedure is applied to select individuals for the next generation.

Termination Criteria: The termination criteria for the PtGA are as follows: (1) no further fitness value improvement in the best individual of the population for β successive iterations; (2) the maximum number of function evaluations (NFEs) reached. If any of the above conditions is satisfied first, the algorithm stops and the best solution ($\mathbf{x}^*$) and its corresponding cost function value are reported.

Note that for PrGA, it is common to employ a weight mapping crossover (WMX) and inversion mutation [8]. The termination criteria can be the same for both PrGA and PtGA.

4 Experimental Studies

This section first describes the MCFP instances and cost functions that have been adopted, followed by some discussion about the mathematical solver packages used in our experiments. We then describe the parameter settings, experimental comparisons and result analysis on the performances of PrGA, PtGA, and mathematical solvers in solving these MCFP instances.

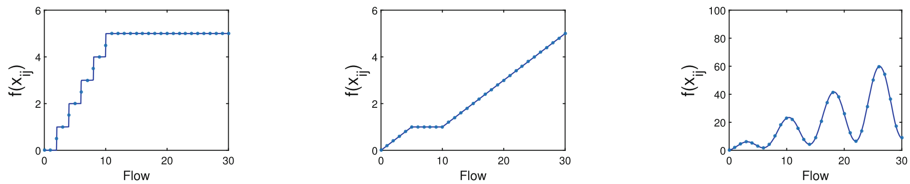
Since our focus is to solve nonlinear non-convex MCFP, we adopt a set of nonlinear non-convex cost functions which are commonly-used in the literature [9,10,14]. Michalewicz et al. [14] categorised the nonlinear cost functions as (1) piece-wise linear cost functions; (2) multimodal (nonlinear non-convex) cost functions; (3) smooth cost functions which are mostly used for Operations Research (OR) problems. In this paper we chose the nonlinear non-convex and arc-tangent approximation of the piece-wise linear cost functions from [9,10,14] to evaluate the performances of PrGA and PtGA. The formulation of these functions are as follows [9,10,14]:

$$\begin{aligned} F_1 : f(x_{ij}) = c_{ij}\big(&arctan(P_A(x_{ij}-S))/\pi + 0.5 + arctan(P_A(x_{ij}-2S))/\pi + 0.5+ \\ &arctan(P_A(x_{ij}-3S))/\pi + 0.5 + arctan(P_A(x_{ij}-4S))/\pi + 0.5+ \\ &arctan(P_A(x_{ij}-5S))/\pi + 0.5\big). \end{aligned} \tag{5}$$

$$\begin{aligned} F_2 : f(x_{ij}) = c_{ij}\big(&(x_{ij}/S)(arctan(P_B x_{ij})/\pi + 0.5) + (1 - x_{ij}/S)(arctan(P_B(x_{ij}-S))/\pi + 0.5)+ \\ &(x_{ij}/S - 2)(arctan(P_B(x_{ij}-2S))/\pi + 0.5)\big). \end{aligned} \tag{6}$$

$$F_3 : f(x_{ij}) = 100 \times c_{ij}\big(x_{ij}(sin\Big(\frac{5\pi x_{ij}}{4S}\Big) + 1.3)\big). \tag{7}$$

Note that c_{ij} is non-negative coefficient, P_A and P_B are set to 1000 and S is set to 2 for F_1, and 5 for F_2 and F_3, respectively [10]. All cost functions F_1, F_2 and F_3 are illustrated in Fig. 6. A set of 35 single-source single-sink MCFP instances is randomly generated with different number of nodes ($n = \{5, 10, 20, 40, 80, 120, 160\}$) and presented in Table 1 (*No.* denotes the instance number, and each instance has n nodes and m arcs). Note that, for each node size (n), five different networks are randomly generated. The number of supply/demand for nodes $1/n$ are set to $q = 20/-20$ in the test instances up to 20 nodes and for all other test problems supply/demand are set to $q = 30/-30$.



(a) Cost function F_1 (b) Cost function F_2 (c) Cost function F_3

Fig. 6. Shapes of different cost functions.

Table 1. A set of 35 randomly generated single-source single-sink MCFP instances.

No.	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*	*No.*	*n*	*m*
1		8	6		24	11		114	16		369	21		1484	26		3419	31		4882
2		8	7		34	12		98	17		385	22		1406	27		3166	32		4718
3	**5**	8	8	**10**	32	13	**20**	105	18	**40**	373	23	**80**	1560	28	**120**	3326	33	**160**	4986
4		9	9		27	14		99	19		406	24		1353	29		3212	34		4835
5		8	10		29	15		101	20		406	25		1526	30		2911	35		5130

This paper focuses on solving nonlinear non-convex MCFPs, which could be considered as mixed integer nonlinear programming (MINLP) problems. However, only very few mathematical solver packages exist for solving MINLP problems, such as CPLEX, Couennn, Baron, LINDOGlobal and AlphaECP [5,12,18]. Some of these solvers have serious limitations. For instance, CPLEX is only capable of solving quadratic optimisation problems, BARON cannot handle the trigonometric functions $sin(x)$, $cos(x)$, while Couenne is not able to handle the *arctangent* function [5]. Among these solvers, AlphaECP and LINDOGlobal are able to handle general MINLPs [12,18]. As a result, we choose to compare our PtGA and PrGA results with those of LINDOGlobal and AlphaECP.

4.1 Parameter Settings

Both PrGA and PtGA are implemented in MATLAB on a PC with Intel(R) Core(TM) i7-6500U 2.50 GHz processor with 8 GB RAM and run 30 times for each problem instance. In order to solve MCFP instances using mathematical solvers, AlphaECP is applied through a high level mathematical language general algebraic modelling system (GAMS) [11] and LINDOGlobal [12] is applied directly on all problem instances.

The parameter settings for the PrGA are as follows: maximum number of iterations ($It_{max} = 200$), population size ($pop\_size = min\{n \times 10, 300\}$), crossover rate ($P_c = 0.95$), mutation rate ($P_m = 0.3$) and maximum number of function evaluations ($NFEs = 100{,}000$). The parameter settings for the PtGA are $It_{max} = 200$, $pop\_size = min\{n \times 5, 300\}$, $P_c = 0.95$, $P_m = 0.3$ and $NFEs = 100{,}000$. The *pop_size* value depends on the number of nodes (n) and increases for the larger networks and the $P_m = 0.3$ value decreases linearly in each iteration. If the results are not improved in $\beta = 30$ successive iterations for PrGA or PtGA, the algorithm is terminated. The run time limit for LINDOGlobal and AlphaECP is set to 3600 seconds (s). Other parameters for AlphaECP and LINDOGlobal are set as default settings.

4.2 Results and Analysis

As mentioned in the procedure of PTbR, after finding a path, there are three possible ways to send the flow over the generated path, i.e., send possible flow (1) randomly (**R**), (2) one-by-one (**O**), or (3) by a maximum possible amount (**M**).

Table 2. Results for cost function F_1.

No.	*n*	*m*	PtGA-R			PtGA-O			PTGA-M			PrGA			LINDOGlobal		AlphaECP		*h*
			t	*mean*	*std*	*t*	*mean*	*std*	*t*	*mean*	*std*	*t*	*mean*	*std*	*t'*	*OBJ*	*t'*	*OBJ*	
1		8	5	**30.1752**	7.29E-15	12	**30.1752**	7.29E-15	3	**30.1752**	7.29E-15	7	**30.1752**	7.29E-15	1	**30.1752**	1	**30.1752**	0
2		8	5	**32.2126**	0.00E+00	13	**32.2126**	0.00E+00	3	**32.2126**	0.00E+00	7	**32.2126**	0.00E+00	2	**32.2126**	1	**32.2126**	0
3	5	8	5	**33.0507**	7.29E-15	15	**33.0507**	7.29E-15	4	**33.0507**	7.29E-15	6	**33.0507**	7.29E-15	1	**33.0507**	1	**33.0507**	0
4		9	5	**33.1016**	7.29E-15	13	**33.1016**	7.29E-15	4	**33.1016**	7.29E-15	7	**33.1016**	7.29E-15	1	**33.1016**	1	**33.1016**	0
5		8	6	**40.3756**	2.19E-14	15	**40.3756**	2.19E-14	3	**40.3756**	2.19E-14	6	**40.3756**	2.19E-14	1	**40.3756**	1	**40.3756**	0
6		24	45	29.2974	2.08E-01	76	29.1682	8.03E-02	42	29.2961	9.99E-02	32	30.0343	3.16E-05	3600	**29.135**	268	30.021	-1
7		34	57	19.8109	2.03E-01	88	20.1956	3.09E-01	45	20.7775	2.94E-01	42	24.1056	6.01E-01	3600	**19.5957**	3600	20.184	-1
8	10	32	48	23.4834	1.16E-01	85	23.6681	1.46E-01	37	24.0765	2.34E-01	28	25.3353	1.39E-01	3600	**23.3061**	350	24.018	-1
9		27	48	26.1797	2.31E-01	69	26.2407	1.52E-01	34	26.4196	7.96E-02	29	28.3965	2.01E-01	3600	**25.9165**	525	25.929	-1
10		29	49	19.8438	9.12E-02	68	20.2202	2.30E-01	34	21.4404	1.55E-01	32	23.3819	2.95E-01	3600	**19.6325**	740	**19.6325**	-1
11		114	160	8.8767	4.60E-01	172	10.9784	4.74E-01	178	13.0274	7.23E-01	153	15.8245	2.03E-01	3600	8.3929	3600	**7.0800**	-1
12		98	141	11.7898	2.18E-01	167	12.4354	3.02E-01	156	13.2006	4.27E-01	160	15.4705	2.40E-01	3600	11.5964	3600	**11.372**	-1
13	20	105	213	8.2120	4.40E-01	239	9.619	5.36E-01	191	10.7931	4.36E-01	133	11.7116	4.25E-01	3600	7.0803	3600	**6.4935**	-1
14		99	187	**10.1773**	3.85E-01	227	11.3854	7.18E-01	202	12.881	6.11E-01	175	15.1102	9.30E-01	3600	10.5551	3600	10.7775	1
15		101	132	14.9139	3.17E-01	191	15.5517	5.14E-01	104	16.5822	7.43E-01	154	18.4694	5.16E-01	3600	13.8717	3600	**13.6730**	-1
16		369	340	1.6119	3.08E-01	437	3.1972	4.35E-01	362	4.1695	5.34E-01	411	6.6595	7.00E-01	3600	**0.4433**	3600	5.366	-1
17		385	316	**3.5652**	4.57E-01	376	4.853	4.95E-01	285	6.0036	3.92E-01	407	10.1381	8.84E-01	3600	4.9674	3600	10.375	1
18	40	373	393	0.5091	3.26E-01	523	1.9296	4.46E-01	406	3.0951	7.99E-01	432	5.1361	1.09E+00	3600	**0.1937**	3600	4.717	-1
19		406	330	0.8437	3.93E-01	370	3.22	6.36E-01	279	6.5308	4.73E-01	342	10.684	6.18E-01	3600	**0.2118**	3600	2.298	-1
20		406	359	3.5742	4.14E-01	435	6.177	6.60E-01	320	9.1484	6.27E-01	360	12.2394	7.58E-01	3600	8.7013	3600	**2.7180**	-1
21		1484	336	**0.7333**	2.71E-04	401	0.8181	1.67E-01	375	2.3523	7.75E-01	459	3.2517	1.32E+00	3600	NF	3600	5.703	1
22		1406	326	**0.6737**	4.90E-04	336	0.806	1.56E-01	369	2.2178	4.64E-01	506	5.4074	1.03E+00	3600	NF	3600	4.982	1
23	80	1560	279	**0.8085**	3.36E-04	361	1.6541	4.84E-01	278	4.128	8.63E-01	429	5.6396	9.22E-01	3600	NF	3600	7.476	1
24		1353	342	**0.6585**	3.50E-04	464	1.793	5.88E-01	354	4.3911	7.70E-01	779	8.1006	1.32E+00	3600	NF	3600	4.18	1
25		1526	322	**0.7628**	3.39E-04	394	1.0477	3.20E-01	302	2.7758	5.56E-01	583	5.0059	1.11E+00	3600	NF	3600	8.642	1
26		3419	725	**1.0924**	5.37E-04	725	2.8201	3.69E-01	711	5.525	4.39E-01	877	8.5144	1.17E+00	3600	NF	3600	6.585	1
27		3166	728	**1.0818**	9.44E-04	805	2.4484	3.89E-01	624	4.9784	6.08E-01	754	7.6565	1.16E+00	3600	NF	3600	2.103	1
28	120	3326	892	**1.0152**	6.75E-04	893	2.3935	3.03E-01	636	5.0317	6.01E-01	906	7.5819	7.76E-01	3600	NF	3600	13.321	1
29		3212	748	**1.0532**	7.88E-04	877	2.5385	5.05E-01	673	5.2673	6.27E-01	777	10.2727	1.74E+00	3600	NF	3600	3.414	1
30		2911	774	**0.9446**	4.06E-04	828	2.0145	4.28E-01	697	4.2784	6.21E-01	817	7.5239	1.37E+00	3600	NF	3600	4.297	1
31		4882	837	**12.2598**	2.55E+00	914	14.5003	4.81E-01	950	15.6057	5.54E-01	922	13.8152	2.74E-01	3600	NF	3600	14.18	1
32		4718	961	**6.1413**	1.42E+00	919	15.6089	9.73E-01	952	17.0593	7.89E-01	927	14.6681	9.81E-01	3600	NF	3600	10.578	1
33	160	4986	902	**8.5483**	1.21E+00	912	16.2194	6.18E-01	947	17.2818	8.47E-01	1345	15.1022	7.50E-01	3600	NF	3600	14.45	1
34		4835	853	**6.3798**	1.01E+00	1064	11.199	4.81E-01	1067	12.1144	6.70E-01	942	10.3464	4.48E-01	3600	NF	3600	14.1422	1
35		5130	994	**10.6176**	1.22E+00	1068	19.7703	7.79E-01	1081	20.5212	8.59E-01	896	18.9247	5.97E-01	3600	NF	3600	15.043	1

Table 3. Results for cost function F_2.

No.	*n*	*m*	PtGA-R			PtGA-O			PTGA-M			PrGA			LINDOGlobal		AlphaECP		*h*
			t	*mean*	*std*	*t*	*mean*	*std*	*t*	*mean*	*std*	*t*	*mean*	*std*	*t'*	*OBJ*	*t'*	*OBJ*	
1		8	4	**7.525**	1.82E-15	8	**7.525**	1.82E-15	2	**7.525**	1.82E-15	5	**7.525**	2.92E-15	1	**7.525**	1	**7.525**	0
2		8	5	**8.65**	0.00E+00	12	8.7085	1.56E-01	3	**8.65**	0.00E+00	5	**8.65**	0.00E+00	1	**8.65**	1	**8.65**	0
3	5	8	4	**8.225**	3.65E-15	8	**8.225**	3.65E-15	2	**8.225**	3.65E-15	5	**8.225**	3.65E-15	1	**8.225**	1	**8.225**	0
4		9	5	**9.925**	1.82E-15	14	10.1713	3.84E-01	3	**9.925**	1.82E-15	5	**9.925**	1.82E-15	1	**9.925**	1	**9.925**	0
5		8	5	**11.375**	5.47E-15	10	**11.375**	5.47E-15	2	**11.375**	5.47E-15	6	**11.375**	5.47E-15	1	**11.375**	1	**11.375**	0
6		24	34	11.0999	1.82E-15	41	12.3986	5.47E-15	33	11.5073	5.26E-01	22	12.2462	2.76E-01	90	**11.05**	230	12.054	-1
7		34	28	11.185	1.10E-01	37	11.2515	9.10E-02	17	11.0976	8.46E-02	25	11.3236	3.65E-15	360	**11.014**	310	11.024	-1
8	10	32	40	10.8903	1.09E-01	54	11.1688	1.09E-01	25	11.0155	5.40E-02	21	11.2944	3.65E-15	580	**10.725**	431	12.224	-1
9		27	26	**11.3939**	3.65E-15	36	**11.3939**	3.65E-15	20	11.3990	1.55E-02	26	11.8862	9.16E-02	610	**11.3939**	581	12.774	0
10		29	42	9.8438	1.03E-01	56	10.4882	7.34E-02	22	10.0002	1.30E-01	23	9.9807	1.57E-01	840	**9.7293**	743	**9.7293**	-1
11		114	127	**9.8945**	3.65E-15	171	10.7446	1.05E-02	90	10.6551	8.82E-02	159	10.5488	1.18E-01	3600	**9.8945**	3600	12.054	0
12		98	134	**11.4485**	0.00E+00	190	11.5245	2.89E-02	101	11.5641	4.27E-02	157	11.595	4.56E-02	3600	**11.4485**	3600	11.653	0
13	20	105	99	**10.8133**	3.65E-15	113	**10.8133**	3.65E-15	87	**10.8133**	2.32E-05	161	11.1547	4.77E-02	3600	**10.8133**	3600	10.8932	0
14		99	119	**10.3787**	0.00E+00	115	**10.3787**	0.00E+00	91	**10.3787**	0.00E+00	168	10.7202	1.88E-01	3600	**10.3787**	3600	11.209	0
15		101	115	**10.7386**	3.65E-15	123	**10.7386**	3.65E-15	94	**10.7386**	3.65E-15	172	10.759	6.63E-03	3600	**10.7386**	3600	10.828	0
16		369	348	**9.7242**	3.65E-15	395	13.8021	2.75E-01	281	10.9094	2.19E-02	372	10.9402	1.35E-02	3600	**9.7242**	3600	10.973	0
17		385	368	**10.4645**	2.27E-01	427	12.0613	3.25E-01	232	10.5797	9.81E-02	373	10.8525	1.16E-01	3600	11.1484	3600	11.244	1
18	40	373	490	**9.4592**	1.82E-15	537	13.9817	2.84E-01	348	11.0849	4.99E-02	381	11.1263	1.69E-02	3600	**9.4592**	3600	11.213	0
19		406	299	**9.5696**	1.82E-15	348	14.144	1.01E-01	230	10.8622	1.34E-01	379	11.1395	7.01E-02	3600	**9.5696**	3600	11.294	0
20		406	314	**10.0842**	1.82E-15	424	13.2452	3.54E-01	221	10.908	1.29E-01	372	11.0853	1.38E-01	3600	**10.0842**	3600	11.219	0
21		1484	323	**10.4395**	1.88E-01	364	14.4468	2.41E-01	243	10.6247	1.77E-02	325	10.7098	4.24E-02	3600	13.6678	3600	10.589	1
22		1406	326	**11.4700**	1.43E-01	396	14.4949	4.10E-02	250	11.5366	1.76E-02	369	11.5842	2.24E-02	3600	13.6122	3600	13.982	1
23	80	1560	249	**9.3868**	1.01E-02	360	14.4684	8.50E-02	209	10.6202	1.55E-02	395	10.6933	3.05E-02	3600	9.5945	3600	10.603	1
24		1353	364	**10.9509**	2.92E-01	510	14.8001	5.86E-02	286	11.0069	2.04E-02	343	11.135	5.69E-02	3600	12.2486	3600	12.623	1
25		1526	332	**9.8808**	6.44E-01	340	14.7078	1.23E-01	250	10.6271	4.95E-02	400	10.7367	5.29E-02	3600	9.9196	3600	14.412	0
26		3419	645	11.5658	2.21E-01	699	14.8154	7.21E-02	603	11.9241	2.23E-02	670	12.0306	3.47E-02	3600	NF	3600	**10.568**	-1
27		3166	555	**11.8297**	8.19E-01	617	15.1228	1.32E-01	402	11.9766	5.31E-02	595	12.1168	5.61E-02	3600	NF	3600	12.861	1
28	120	3326	670	**11.9377**	1.62E-01	716	15.1615	8.53E-02	461	12.1081	4.14E-02	467	12.2275	5.72E-02	3600	NF	3600	12.038	1
29		3212	745	11.8559	4.17E-03	864	15.3256	7.73E-02	396	12.1214	4.13E-02	791	12.2555	6.60E-02	3600	**11.8478**	3600	12.028	-1
30		2911	731	**12.5460**	1.59E-01	699	15.3006	7.35E-02	478	12.7238	4.79E-02	877	12.7510	6.14E-02	3600	14.2069	3600	16.797	1
31		4882	989	**11.4402**	3.48E-01	1004	26.5183	5.87E-01	914	16.903	5.39E-01	908	18.4306	7.00E-01	3600	NF	3600	12.4545	1
32		4718	946	12.0739	2.05E-01	1021	28.0915	5.06E-01	999	19.0685	5.14E-01	912	20.5485	1.07E+00	3600	NF	3600	**11.435**	-1
33	160	4986	846	**12.0254**	3.27E-01	929	28.4358	6.94E-01	912	18.9474	6.31E-01	914	20.4764	1.01E+00	3600	NF	3600	12.241	1
34		4835	916	**11.6837**	1.78E-01	1029	25.521	4.70E-01	1005	16.7851	7.29E-01	994	18.3083	7.24E-01	3600	NF	3600	11.884	1
35		5130	995	**12.1966**	9.78E-01	1122	30.2795	6.41E-01	1082	19.7566	6.03E-01	951	21.2144	7.46E-01	3600	NF	3600	12.691	1

This creates three different variants of the PTbR-based GA, namely PtGA-R, PtGA-O, and PtGA-M respectively.

To compare the effectiveness of these representation methods, we evaluate these variants, as well as PrGA using a set of 35 MCFP instances. Tables 2, 3 and 4 present the results of PtGA-R, PtGA-O, PtGA-M, PrGA, LINDOGlobal, and AlphaECP on a total of 35 test problems using cost functions F_1, F_2, F_3. The *std* and t (for PrGA and PtGA) denote the standard deviation of the results and the average of running time in seconds respectively, and the *mean* represents the average of cost function values over 30 runs. The t' and OBJ for LINDOGlobal

Table 4. Results for cost function F_3.

No.	n	m	PtGA-R			PtGA-O			PTGA-M			PrGA			LINDOGlobal		AlphaECP		h
			t	mean	std	t	mean	std	t	mean	std	t	mean	std	t'	OBJ	t'	OBJ	
1		8	5	**114.3819**	7.29E-14	11	**114.3819**	7.29E-14	3	**114.3819**	7.29E-14	9	**114.3819**	7.29E-14	1	**114.3819**	1	**114.3819**	0
2		8	6	**111.8675**	0.00E+00	14	**111.8675**	0.00E+00	4	**111.8675**	0.00E+00	9	**111.8675**	0.00E+00	1	**111.8675**	1	**111.8675**	0
3	5	8	7	**138.2576**	0.00E+00	16	**138.2576**	0.00E+00	3	138.285	0.00E+00	6	138.285	0.00E+00	1	**138.2576**	1	142.79	0
4		9	7	**107.3083**	0.00E+00	14	**107.3083**	0.00E+00	3	120.0322	5.83E-14	6	123.3197	2.92E-14	1	**107.3083**	1	123.32	0
5		8	6	**150.9943**	2.92E-14	15	**150.9943**	2.92E-14	3	**150.9943**	2.92E-14	7	**150.9943**	2.92E-14	1	**150.9943**	1	**150.9943**	0
6		24	66	106.7312	1.44E+00	108	107.7509	2.18E+00	43	118.2249	5.35E-01	41	135.3015	1.87E+00	3600	**104.5113**	332	141.64	-1
7		34	36	81.7436	3.64E-01	69	85.7071	3.81E+00	39	89.5549	7.08E+00	32	117.8976	1.06E+01	3600	**80.97226**	431	108.13	-1
8	10	32	49	89.4711	1.47E+00	65	88.569	6.40E-01	36	90.3497	2.08E+00	33	131.963	6.02E-01	3600	**86.41261**	491	135.434	-1
9		27	45	112.2958	1.31E+00	74	115.9811	4.28E+00	37	114.265	1.58E+00	27	152.5404	1.28E+00	3600	**110.3257**	288	133.92	-1
10		29	49	90.1242	8.41E-01	78	90.2275	8.48E-01	39	90.4574	1.21E+00	35	104.8298	8.38E+00	3600	**88.05281**	333	135.81	-1
11		114	90	69.7972	1.25E+00	163	105.9186	8.93E-01	81	103.9751	3.76E+00	86	103.4472	4.32E+00	3600	**66.6795**	3600	102.375	-1
12		98	111	79.5198	5.91E-01	194	116.2849	4.75E-01	79	116.3386	5.97E+00	78	122.6103	3.04E+00	3600	**79.3041**	3600	92.146	0
13	20	105	137	96.3334	2.15E+00	234	123.3836	2.84E+00	137	120.5739	1.11E+00	93	116.6993	5.49E+00	3600	**90.1959**	3600	137.339	-1
14		99	92	**79.4689**	0.00E+00	213	103.5296	1.40E+00	86	99.9841	3.27E+00	80	115.5684	1.98E+00	3600	**79.4689**	3600	115.578	0
15		101	145	82.9519	1.56E+00	199	108.6625	1.11E+00	99	111.233	2.93E+00	75	115.2218	2.27E+00	3600	**78.4144**	3600	116.503	-1
16		369	299	**75.7446**	6.09E-01	384	119.4979	4.80E+00	323	105.1299	2.87E+00	223	108.7354	3.66E+00	3600	77.1499	3600	145.164	1
17		385	253	**89.9066**	3.16E-01	396	108.3421	1.07E+00	241	111.1345	2.21E+00	294	119.3059	3.82E+00	3600	90.1749	3600	148.655	1
18	40	373	203	83.1843	3.89E+00	582	133.7859	3.33E+00	433	106.883	4.90E+00	215	93.0854	3.19E+00	3600	**79.8989**	3600	90.777	-1
19		406	282	**72.447**	5.14E-01	355	119.7419	4.45E+00	231	99.0947	3.44E+00	285	103.488	2.99E+00	3600	73.0978	3600	92.443	1
20		406	288	65.1725	4.73E-01	359	127.004	3.93E+00	277	102.8105	6.24E+00	201	109.9314	3.55E+00	3600	**65.0667**	3600	117.312	0
21		1484	325	**83.9645**	1.85E+00	404	138.502	5.27E+00	296	109.589	1.40E+00	336	107.3701	2.04E+00	3600	90.1854	3600	124.549	1
22		1406	238	**93.3544**	1.13E+00	332	128.5883	6.91E+00	246	109.0366	1.42E+00	347	108.2733	2.49E+00	3600	93.4224	3600	129.357	0
23	80	1560	296	**106.1091**	3.03E+00	304	143.037	2.91E+00	243	112.9184	1.57E+00	272	111.1676	1.20E+00	3600	148.6104	3600	107.493	0
24		1353	261	**63.7182**	4.32E-01	464	134.5088	7.03E+00	331	108.1399	2.83E+00	314	94.7708	5.95E+00	3600	65.3662	3600	132.307	1
25		1526	365	**59.5713**	2.19E-14	293	134.002	6.85E+00	241	101.3695	7.98E+00	335	106.9928	5.50E+00	3600	**59.5713**	3600	125.508	0
26		3419	544	**86.9299**	1.93E+00	595	130.1538	2.76E+00	491	84.2045	2.36E+00	686	89.0996	3.92E+00	3600	NF	3600	91.267	1
27		3166	464	**55.0274**	1.79E+00	475	129.6023	2.81E+00	437	83.2053	3.92E+00	544	87.5812	5.25E+00	3600	66.009	3600	92.576	1
28	120	3326	530	80.7884	8.00E+00	693	131.3889	3.27E+00	535	88.2298	5.87E-01	685	88.4319	5.24E-01	3600	NF	3600	**80.755**	0
29		3212	486	**84.6939**	5.19E+00	566	133.1519	2.59E+00	528	90.2997	5.19E-01	535	89.1701	4.72E-01	3600	NF	3600	136.381	1
30		2911	481	**89.7215**	1.09E+00	525	136.358	4.51E+00	455	90.4635	1.55E+00	582	95.9605	6.85E+00	3600	NF	3600	102.4492	1
31		4882	885	**80.6756**	5.27E+00	994	257.169	5.66E+00	631	145.3273	6.83E+00	907	158.2989	1.14E+01	3600	145.654	3600	98.52.90	1
32		4718	835	**82.8948**	9.12E+00	944	270.3601	8.35E+00	651	156.9115	8.70E+00	895	174.8577	9.43E+00	3600	144.216	3600	123.975	1
33	160	4986	910	**80.4273**	5.40E+00	1006	266.059	6.02E+00	694	159.3669	5.94E+00	800	170.5026	1.20E+01	3600	NF	3600	117.75	1
34		4835	992	**82.432**	3.75E+00	946	247.8246	4.70E+00	740	143.5167	7.49E+00	949	158.1368	9.00E+00	3600	NF	3600	102.685	1
35		5130	988	**88.063**	4.01E+00	1040	272.3081	5.92E+00	856	162.7781	7.43E+00	917	181.6462	9.77E+00	3600	173.679	3600	117.999	1

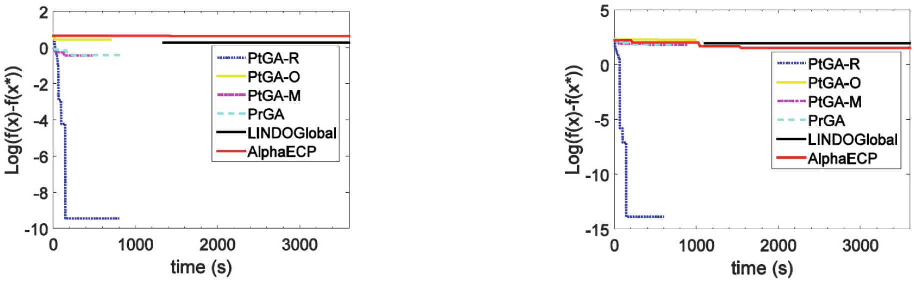
Table 5. The Friedman test's results for PtGA-R, PtGA-O, PtGA-M and PrGA.

	p-value	Mean column ranks			
		PtGA-R	PtGA-O	PtGA-M	PrGA
F_1	0.000E+00	16.49	34.13	50.94	60.43
F_2	0.000E+00	18.99	59.27	32.69	51.05
F_3	0.000E+00	16.28	54.94	41.05	49.72

and AlphaECP (exact methods) denote the running time and the cost function value, respectively. "NF" denotes that the mathematical solver cannot find any feasible solution in the time limit of an hour (3600 s). The best cost function value for each instance is presented in boldface.

To carry out a comprehensive comparison among PtGA-R, PtGA-O, PtGA-M, and PrGA, we use Friedman test [6]. For each function (F_1, F_2 and F_3) we perform the Friedman test with the significance level set to 0.05, and the results are shown in Table 5. Since the p-values in all three functions are almost zero (less than 0.05), there are overall statistically significant differences between the mean ranks of the algorithms (PtGA-R, PtGA-O, PtGA-M and PrGA). The mean column rank values of the PtGA-R is less than those of the PtGA-O, PtGA-M and PrGA (Table 5) which indicates that PtGA-R's performance is better than those of the other GA variants. It is clearly evident that the superior performance of the PrGA-R comes from utilising PTbR in its procedure and sending a random possible flow.

We also compare the performance of PtGA-R with LINDOGlobal and AlphaECP by applying a one-sample t-test with the significance level set to 0.05. After performing the one-sample t-test, if PtGA-R has statistically better or worse performance than that of the mathematical solvers, the parameter h is



(a) F_2 on instance No.30. (b) F_3 on instance No.35.

Fig. 7. Convergence graphs for PtGA, PrGA, LINDOGlobal and AlphaECP.

set to 1 and −1 respectively, otherwise h is set to 0. The last column of Tables 2, 3 and 4 presents the value of h for all instances.

For cost function F_1, Table 2 shows that PtGA-R has better performance on all instances with $n = \{80,120,160\}$ compared with that of PtGA-O, PtGA-M, PrGA, LINDOGlobal and AlphaECP. Furthermore, LINDOGlobal fails to find any feasible solutions when the problem size is increased ($n = \{80,120,160\}$). For F_2, Table 3 shows that on 28 out of 35 instances (80%), the PtGA-R has equal or better performance than the two mathematical solvers.

With regard to cost function F_3, Table 4 shows that even on instances 3 and 4 (small-sized instances), PrGA failed to find the optimal solutions due to the limitations of PbR in searching the feasible region, which is consistent with our analysis in Subsect. 2.1. In all large-sized instances ($n = \{80,120,160\}$), the PtGA-R has similar or better performance than that of the mathematical solvers.

Figure 7 shows the convergence graphs of PtGA-R, PtGA-O, PtGA-M, PrGA and the mathematical solvers for large-sized instances on F_2 and F_3. Since LINDOGlobal is not able to find any feasible solution for all large-sized problems on F_1, we are not able to provide the convergence graph for that cost function. As shown in Fig. 7, PtGA-R converges to a good solution faster than other GA variants as well as LINDOGlobal and AlphaECP. Based on Fig. 7, LINDOGlobal cannot find any feasible solution after about 1000 s. Once a solution is found, mathematical solvers (specially LINDOGlobal) are not able to improve it.

5 Conclusion

This paper has proposed a new encoding scheme called probabilistic tree-based representation (PTbR) for more effective handling of MCFPs. We examine the commonly-used priority-based representation (PbR), and compare it with PTbR to demonstrate that PTbR is superior to PbR for solving MCFPs. To validate our analysis on these representation schemes, the PTbR-based GA (i.e., PtGA) and PbR-based GA (i.e., PrGA) are evaluated over a set of 35 single-source single-sink network instances with up to five thousand variables. The experimental results demonstrate that PtGA with a random flow (i.e., PtGA-R) has better performance than PrGA on all problem instances. In addition, PtGA-R has also been shown to produce better solutions and have better efficiency than mathematical solvers such as LINDOGlobal and AlphaECP when considering the large-sized instances. For future research, one can focus on solving large-sized real-world MCFP using the proposed representation method.

References

1. Abdelaziz, M.: Distribution network reconfiguration using a genetic algorithm with varying population size. Electr. Power Syst. Res. **142**, 9–11 (2017)
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications, pp. 4–6. Prentice Hall, Upper Saddle River (1993)
3. Aiello, G., La Scalia, G., Enea, M.: A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. Expert Syst. Appl. **39**(12), 10352–10358 (2012)
4. Amiri, A.S., Torabi, S.A., Ghodsi, R.: An iterative approach for a bi-level competitive supply chain network design problem under foresight competition and variable coverage. Transp. Res. Part E: Logist. Transp. Rev. **109**, 99–114 (2018)
5. Burer, S., Letchford, A.N.: Non-convex mixed-integer nonlinear programming: a survey. Surv. Oper. Res. Manag. Sci. **17**(2), 97–106 (2012)
6. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011)
7. Fontes, D.B., Gonçalves, J.F.: Heuristic solutions for general concave minimum cost network flow problems. Networks **50**(1), 67–76 (2007)
8. Gen, M., Cheng, R., Lin, L.: Network Models and Optimization: Multiobjective Genetic Algorithm Approach. Springer, London (2008). https://doi.org/10.1007/978-1-84800-181-7
9. Klanšek, U.: Solving the nonlinear discrete transportation problem by minlp optimization. Transport **29**(1), 1–11 (2014)
10. Klanšek, U., Pšunder, M.: Solving the nonlinear transportation problem by global optimization. Transport **25**(3), 314–324 (2010)
11. Lastusilta, T., et al.: GAMS MINLP solver comparisons and some improvements to the AlphaECP algorithm. In: Process Design and Systems Engineering Laboratory, Department of Chemical Engineering Division for Natural Sciences and Technology, Abo Akademi University, Abo, Finland (2011)
12. Lin, Y., Schrage, L.: The global solver in the LINDO API. Optim. Methods Softw. **24**(4–5), 657–668 (2009)
13. Lotfi, M., Tavakkoli-Moghaddam, R.: A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. Appl. Soft Comput. **13**(5), 2711–2726 (2013)
14. Michalewicz, Z., Vignaux, G.A., Hobbs, M.: A nonstandard genetic algorithm for the nonlinear transportation problem. ORSA J. Comput. **3**(4), 307–316 (1991)
15. Reca, J., Martínez, J., López-Luque, R.: A new efficient bounding strategy applied to the heuristic optimization of the water distribution networks design. In: Congress on Numerical Methods in Engineering CMN (2017)
16. Tari, F.G., Hashemi, Z.: A priority based genetic algorithm for nonlinear transportation costs problems. Comput. Ind. Eng. **96**, 86–95 (2016)

17. Vegh, L.A.: A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. SIAM J. Comput. **45**(5), 1729–1761 (2016)
18. Westerlund, T., Pörn, R.: Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. Optim. Eng. **3**(3), 253–280 (2002)
19. Zhang, Y.H., Gong, Y.J., Gu, T.L., Li, Y., Zhang, J.: Flexible genetic algorithm: a simple and generic approach to node placement problems. Appl. Soft Comput. **52**, 457–470 (2017)