A BIASED RANDOM-KEY GENETIC ALGORITHM FOR THE TREE OF HUBS LOCATION PROBLEM

LUCIANA S. PESSOA, ANDRÉA C. SANTOS, AND MAURICIO G. C. RESENDE

ABSTRACT. Hubs are facilities used to treat and dispatch resources in a transportation network. The main idea of Hub Location Problems (HLP) is to locate a number of hubs in a network and route resources from origins to destinations such that the total cost of attending all demands is minimized. In this study, we investigate a particular HLP, called the Tree of Hubs Location Problem in which hubs are connected by means of a tree and the overall network infrastructure relies on a spanning tree. This problem is particularly interesting when the total cost of building the hub backbone is high. In this paper, we propose a biased random key genetic algorithm for solving the tree of hubs location problem. Computational results show that the proposed heuristic is a robust and effective method to tackle this problem. The method was able to improve some best known solutions from the benchmark instances used in the experiments.

1. Introduction

Hub Location Problems (HLP) have been investigated for the past 30 years (Farahani et al., 2013; Alumur and Kara, 2008), and similar ideas can be traced back to the early sixties (Hakimi, 1964). Hubs are facilities used to treat and dispatch resources in a transportation network. The main idea of HLP is to locate a number of hubs in a network and route resources from origins to destinations such that the total cost of attending all demands is minimized. Several variants of HLP are found in the literature and are defined, among others, by their network infrastructure (e.g. star, trees), number of hubs (single or multiple hubs), resource availability (unlimited or limited) and their optimization criteria (min, min-max, min-sum).

HLP was firstly studied in the context of telecommunication networks, according to Farahani et al. (2013). However, HLP have had a growing interest in many real applications such as transportation systems (Cunha and Silva, 2007; Gelareh and Nickel, 2008; Limbourg and Jourquin, 2009), supply chain and logistics (Wang and Cheng, 2010; Ishfaq and Sox, 2012), emergencies (Berman et al., 2007), airlines and airport industries (Adler and Hashai, 2005; Costa et al., 2010). Obviously, the resources, also called flows, passing through the network are differentiated according to the applications, e.g., commodities for telecommunication networks, supplies for logistics, etc. In this study, we investigate a particular HLP, called the Tree of Hubs Location Problem (THLP) in which hubs are connected by means of a tree and the overall network infrastructure relies on a spanning tree. This problem

Key words and phrases. Genetic algorithm, random key, tree hub problem, network design, logistics.

The research of Luciana S. Pessoa was partially supported by PNPD/Capes and PUC-Rio. Work of Mauricio G. C. Resende was done when he was employed by AT&T Labs Research.

is particularly interesting when the total cost of building the hub backbone is high. Potential applications appear in telecommunications, transportation, logistics, and in the design of high-speed railways (Chen et al., 2008; Contreras et al., 2010).

The THLP is formally defined on an undirected connected finite and simple graph $G = (V, E)$, where V and E stand, respectively, for a set of vertices and a set of edges. Costs $c_{i,j} \geq 0$ are associated with every edge $[i, j] \in E$. Moreover, let P be the set of hubs, where $3 \leq p = |P| \leq |V|$ is the number of hubs to be deployed and $d_{i,j}$ the demand from $i \in V$ to $j \in V$. A tree $\mathcal{T} = (V', E')$ is a connected subgraph of G with no cycles, such that $V' \subseteq V$ and $E' \subset E$, while in a spanning tree $\mathcal{T}' = (V', E')$ of G, $V' = V$ and $E' \subset E$. Since the network infrastructure (topology) relies on a spanning tree, by definition there is a unique path between i and j. In the THLP, a demand from i to j is routed from i to a hub k, then uses a path $\mathcal{P}_{k...l}$ in the tree of hubs, and finally is sent to its destination j, whenever j is not a hub. Costs associated with edges in the tree of hubs path ($\mathcal{P}_{k...l}$) receive a discount α, where $0 \leq \alpha \leq 1$. The problem consists in locating p hubs and assigning $V \backslash P$ nodes, each to a unique hub such that the total cost of routing all the demands is minimized, the hubs are connected by means of a tree, and the overall network topology define a spanning tree. The THLP was proved to be NP-hard by Contreras et al. (2010).

Figure 1 illustrates an example of a partial solution for the THLP, where $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and the set of hubs is located at $P = \{3, 5, 6\}$. Furthermore, the dotted black lines correspond to the edges of the tree of hubs and the cost associated with each edge is the price of routing one resource unit. Suppose one has to decide to connect node "2" to hub "3" or "6". Let us consider $d_{2,1} = 2$ and $d_{2,4} = 10$, and $\alpha = 0$. Connecting "2" to hub "3" implies respectively costs of 10 and 50 to route 2 units to "1" and 10 units to "4". This results in a total cost of 60. Whenever "2" is connected to node "6", the total cost of routing the resources is equal to 108. One may note that even if $c_{2,6} < c_{2,3}$, it is better to connect "2" to hub "3". This example illustrates the demands and the total routing cost strongly impact the final result and illustrate the combinatorial nature of the problem.

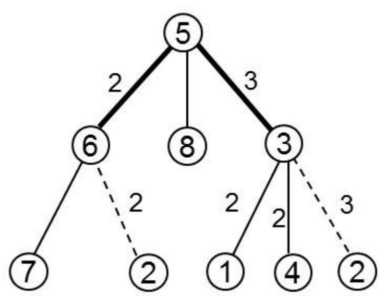

FIGURE 1. Example of a partial solution for the THLP.

The THLP problem was introduced by Contreras et al. (2010; 2009) inspired by the pioneering work of O'Kelly (1987) for the quadratic HLP. Contreras et al. (2009) **proposed** a Mixed Integer Linear Programming (MILP) formulation based on shortest paths between all pair of nodes, a Lagrangian relaxation, and a heuristic. Computational results **were** presented for instances with up to 100 vertices with optimality gaps of about 10%. Contreras et al. (2010) proposed a more efficient MILP applying valid inequalities. Experiments **were** addressed for instances with up to 25 nodes. The MILP of Contreras et al. (2009) has $O(|V|^4)$ variables and constraints and the one of Contreras et al. (2010) has $O(|V|^3)$ variables and constraints. In terms of exact algorithms, Sá et al. (2013) **worked on** a Benders decomposition algorithm using the formulation of Contreras et al. (2009) which was able to prove optimality for instances with up to 100 nodes in about 75 hours on an Intel Xeon with 8 cores of 2.53 GHz and 24 Gb of memory, running the Linux operating systems. Recently, Abreu Júnior et al. (2015) **proposed** an ad-hoc algorithm to solve the subproblems of the Benders decomposition introduced by Sá et al. (2013), which speed up the procedure by about 28%. The work of Sá et al. (2015) presents Benders based algorithms and heuristics for the multiple lines hub location. The study focuses on a generalization of the Hub Line Location Problem (HLLP), named q-line hub location problem (q-HLLP). The q-HLLP problem aims to locate a set of hubs in lines such that the total cost of routing demands from origins to destinations is minimized. The number of hubs is a decision variable of the model and upper and lower limits are given.

In this study, we propose a Biased Random-Key Genetic Algorithm (BRKGA) for the THLP. **BRKGA has produced high-quality results for network design problems (Resende, 2012) and is an interesting method for solving problems relying on trees and spanning trees (Ruiz et al., 2015; Coco et al., 2013; Fontes and Gonçalves, 2013) since well-known algorithms for such structures can be adapted and used as a decoder in a BRKGA.** To the best of our knowledge, this is the first metaheuristic-based heuristic proposed for solving the THLP. The remaining of this paper is organized as follows. An MILP formulation is given in Section 2. The main ideas of the BRKGA are shown in Section 3, followed by the proposed BRKGA for the THLP in Section 4. Computational experiments and concluding remarks are, respectively, described in Sections 5 and 6.

2. MILP Formulation

The mathematical formulation presented in this section was proposed by Contreras et al. (2010) and their results will be used to measure the performance of the **proposed** heuristic. It makes use of graph G, previously defined, with costs $c_{i,j} > 0$ associated with each edge $[i,j] \in E$. Some variables are defined in a support digraph G' obtained from G as follows: every edge $[i,j] \in E$ is transformed into two arcs (i,j) and (j,i) belonging to A with identical cost. Moreover, $d_{i,j}$ is the demand to be routed from $i \in V$ to $j \in V$. The oriented flow variables $y_{ij}^k \geq 0$ guarantee the final solution is connected and are also responsible for determining the amount of flow (resources) k passing through arc $(i,j) \in A$. The total flow O_i (D_i) leaving (entering) node i is given by

$$O_i = \sum_{j \in V} d_{i,j} \quad \forall i \in V, \tag{1}$$

and

$$D_i = \sum_{j \in V} d_{j,i} \quad \forall i \in V. \tag{2}$$

The formulation has decision variables x_{ij}, $[i,j] \in E$ that determine whether two hubs are connected ($x_{ij} = 1$) or not ($x_{ij} = 0$). The binary variables z_{ij} are used to identify links between hub and non-hubs node, as well determine if a node is a hub. If $i \neq j$, $z_{ij} = 1$ implies that non-hub node i is connected to hub j, $z_{ij} = 0$ otherwise. Whenever $i = j$, if $z_{ii} = 1$, then node i is set as a hub, $z_{ii} = 0$ otherwise. The MILP formulation for the THLP is given in (3) to (13).

$$\min \; Z = \sum_{i \in V} \sum_{j \in V} (c_{i,j}\, O_i + c_{j,i}\, D_i)\, z_{ij} + \sum_{k \in V} \sum_{i \in V} \sum_{j \in V\, i \neq j} \alpha\, c_{i,j}\, y^k_{ij} \qquad s.t. \tag{3}$$

$$\sum_{i \in V} z_{ii} = p \tag{4}$$

$$\sum_{j \in V} z_{ij} = 1 \quad \forall i \in V \tag{5}$$

$$z_{ij} + x_{ij} \leq z_{jj} \qquad \forall i, j \in V, i < j \tag{6}$$

$$z_{ji} + x_{ij} \leq z_{ii} \qquad \forall i, j \in V, i < j \tag{7}$$

$$y^k_{ij} + y^k_{ji} \leq O_k x_{ij} \qquad \forall i, j, k \in V, i < j \tag{8}$$

$$\sum_{i:(i,j) \in A} y^k_{ij} - \sum_{i:(j,i) \in A} y^k_{ji} = \sum_{i \in V} d_{k,i} z_{ij} - O_k z_{kj} \qquad \forall k, j \in V, k \neq j \tag{9}$$

$$\sum_{i \in V} \sum_{j \in V} x_{ij} = p - 1 \tag{10}$$

$$y^k_{ij} \geq 0 \qquad \forall (i,j) \in A, k \in V \tag{11}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in V \tag{12}$$

$$z_{ij} \in \{0,1\} \qquad \forall i, j \in V. \tag{13}$$

The objective function (3) has two parts, the first part represents the cost of routing all demands from a non-hub node to a hub while the second part gives the cost of routing flow on the tree of hubs with a discount α. Equation (4) guarantees that p hubs will be opened, while Constraints (5) to (7) ensure that non-hub nodes are connected to a unique and open hub. All flow from a node i is aggregated by means of O_i and its conservation on the tree of hubs is given by Inequalities (8). Constraints (9) are the flow conservation of the overall path leaving node k and entering all nodes $j \in V$. Equation (10) **ensures that the tree of hubs has the correct number of edges.** One may note that the overall topology is implicitly guaranteed to be a spanning tree since the hubs form a tree and all non-hubs are connected to a unique hub. The variables are defined by (11) to (13). This formulation has $O(|V|^3)$ variables and constraints.

3. Biased Random-key Genetic Algorithms

A genetic algorithm can be classified as a population-based heuristic (Beasley, 2002) which works with a population of solutions and combines them in some way to generate new solutions.

In this paper, we applied a genetic algorithm that encodes a solution (a chromossome) as a string of real-valued numbers in the interval [0,1] – a vector of random keys. A random-key plays the role of a gene in the chromossome. The *biased* aspect of the algorithm is due to the priority given to the best solutions to pass their characteristics to the future generations. Biased Random-Key Genetic Algorithms (or BRKGA, for short) have been successfully applied to a variety of optimization problems, as surveyed by Gonçalves and Resende (2011).

The first generation (initial population) is made up of I individual chromosomes composed of G genes. Each gene has a value (allele) generated uniformly at random in the interval [0,1]. Each chromossome is given to a *decoder* which converts the random-key sequence in a feasible solution and evaluates the solution cost, or fitness value.

The strategy to evolve the population begins partitioning the population into two sets: ELITE and NON-ELITE. A parameter I_e defines the elite set size. As illustrated in Figure 2, three operations are carried out on the current population to obtain the next generation:

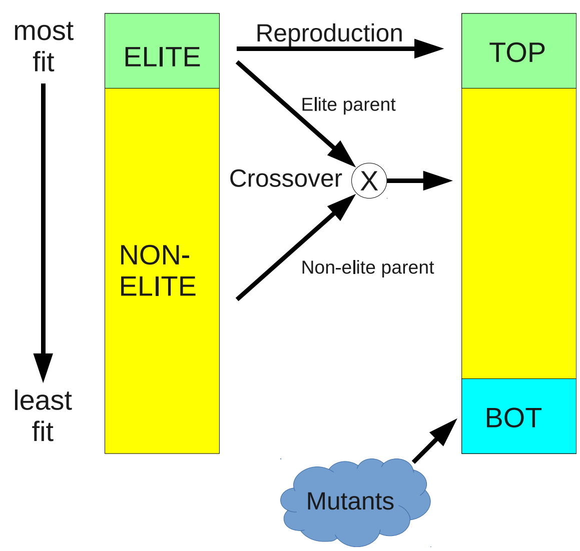

Figure 2. Transitional process between consecutive generations.

Reproduction is acomplished by copying all individuals from the elite set to the TOP partition, without change, to the population of the next generation.

Crossover combines two parent chromossomes and produces an offspring individual. Note, in Figure 2, that one parent comes from the elite set and the other parent is a non-elite individual. Both are chosen at random. The crossover process applies the *parametrized uniform crossover* scheme proposed by Spears and DeJong

(1991). A parameter $\rho_A > 0.5$ defines the probability of an allele of the elite parent to be passed to the offspring individual.

The concept of *mutants* is used instead of *mutation*. That is, at each generation, a group of I_m individuals are randomly generated in the same way as the original population. These new individuals are placed in set BOT.

This evaluation-selection-evolution cycle is repeated until a satisfactory solution is found. In practice, the algorithm stops when a maximum processing time is reached, a solution as good as given target is found, or a certain number of generations is produced.

4. Chromosome representation and decoding

In this section, we specialize a BRKGA for the tree of hubs location problem.

A chromosome that encodes a solution for the THLP is represented by a string having three parts, as shown in Figure 3. Each gene in the chromosome is a real-value number in the interval [0,1]. The first $|V|$ genes define hub and non-hub vertices. The next $|V| - p$ genes assign a non-hub node to a hub node. Finally, the last $p(p-1)/2$ genes decode into a tree with the hub vertices.

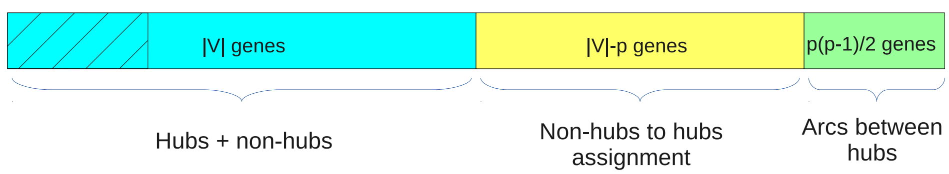

Figure 3. A chromosome for the Tree of Hubs Location

Figure 4 shows an example of the decoder for the Tree of Hubs Location, by considering a set of 10 vertices, among which 3 are chosen as hubs. In this case, a chromossome is made up 20 genes. Initially, the first $|V|$ genes are labeled $1, ..., |V|$ and sorted, so that after sorting, the first p labels are defined as hubs and, the rest, as non-hub vertices (Figure 4(a)). Then (see Figure 4(b)) a non-hub vertice is assigned to a hub by the association of each hub to a range of values in the interval $[0, 1]$, so that each hub has probability $1/p$ of being selected. Finally, each of the last $p(p-1)/2$ genes represents an arc between two hub vertices. After being labeled and sorted, these genes give the order in which the arcs are considered to grow a spanning tree, one arc at a time, until all vertices are connected (Figure 4(c)). Figure 4(d) gives the solution obtained for this example. After mapping the random-key vector into a feasible solution, the decoder evaluates the solution cost.

5. Computational experiments

The computational experiments were performed on a 3.16GHz Intel Core2-Duo processor with 4 GB RAM computer running Linux Ubuntu. Each run was limited to a single processor. All codes were implemented in C.

The experiments reported in this section aim to evaluate the quality of the solutions returned by different variants of BRKGA. Configuration parameters shown in Table 1 were considered to generate 15 variants of BRKGA for the tree of hubs location problem. Parameters I_e and I_m define different sizes for the partitions

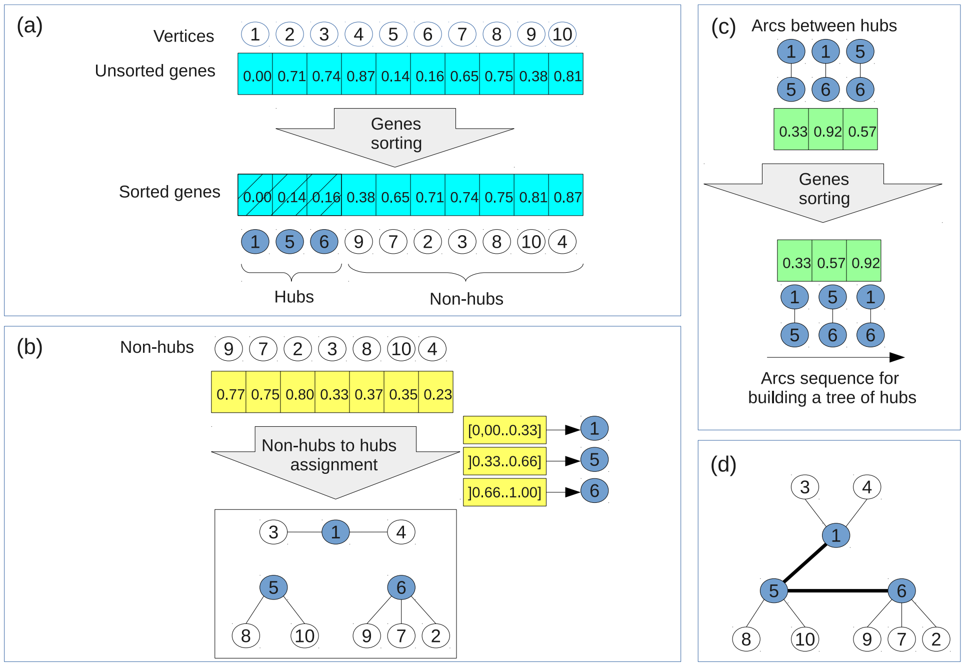

FIGURE 4. A decoder for the Tree of Hubs Location

TOP and BOT. We tested all the combinations of I_e and I_m. Population size I **was** kept proportional to the problem size. Gonçalves and Resende (2011) showed that this is a good strategy. Parameter ρ_A was set to 0.7 based on some preliminary experiments. The algorithm stops after 100 generations without improvement. To evaluate each variant of BRKGA, ten runs were carried out for each instance, varying the initial seed given to the random number generator.

TABLE 1. Configuration parameters for the BRKGA algorithm

Parameter	Value
$I_e =$	$\{0.15, 0.20, 0.25\} \times I$
$I_m =$	$\{0.15, 0.20, 0.25, 0.35, 0.45\} \times I$
$I =$	$100 \times \|V\|$
$\rho_A =$	0.7
Stopping Criterion =	100 generations without improvement
Number of seeds =	10

5.1. **Datasets.** We have considered two datasets described by Contreras et al. (2010). There are 63 instances: 36 derived from the CAB dataset and 27 from AP dataset. **Both test sets correspond to complete graphs** . Instance sizes vary from $n = 10$ to 25. For each instance, a number $p \in \{3, 5, 8\}$ is required to be chosen as hubs, and a discount factor $\alpha \in \{0.2, 0.5, 0.8\}$ is given. Contreras et al. (2010) present the optimal solution values for 59 out of 64 instances.

5.2. **Comparative Metrics.** We used the following metrics, as described by Resende et al. (2010) to compare the solutions obtained by each BRKGA variant with those obtained by Contreras et al. (2010):

- *BestValue*: for each instance, *BestValue* is the best solution value obtained over all executions of the variants considered.
- *Dev*: for each run of a variant, *Dev* is the relative deviation in percentage between *BestValue* and the solution value obtained in that run.
- *ADev*: average value of *Dev* over all instances and runs of a variant in a particular experiment.
- *#Best*: for each variant, this metric gives the number of instances for which the *BestValue* was found.
- *NScore*: as described by Ribeiro et al. (2002), for each variant and instance, this metric gives the number of variants that found better solutions than this specific variant for this instance. In case of ties, all variants receive the same score, equal to the number of variants strictly better than all of them.
- *Score*: for each variant, this metric gives the sum of the *NScore* values over all instances in the experiment. Thus, lower values of *Score* correspond to better variants.
- *ATime*: for each variant and instance, *ATime* is the average time taken to reach the best solution found by this variant over all runs of the same instance.
- *TTime*: for each variant, this metric gives the sum over all instances of *ATime*.

5.3. **Experimental results.** Table 2 summarizes the results obtained **by each of the 15 variants of BRKGA** over each set of instances. The fifteen variants were able to find good solutions of similar or better quality than those presented by Contreras et al. (2010), as demostrated by their ADev metric, which ranged from 0.49 to 1.24%. For the CAB dataset, the best results were obtained by setting I_e**=0.15 and** I_m**=0.35**. The relative deviation from the best known solutions was, on average, 0.49%. This variant was able to find the best solution for 25 out of 36 instances. Besides, it presented the smallest Score metric. For the AP dataset, the variant with I_e**=0.20 and** I_m**=0.35** obtained the best average deviation (0.66%) and Score metric, although it was not able to reach the largest number of best solutions. The last line of this table presents the metrics relative to Contreras' results. The results shows that BRKGA improved the best known solution for two instances (one instance of each dataset) which resulted in an average deviation of 0.0005% for CAB dataset and, 0.01% for AP dataset. **It is worth noting that TTime increases along with I_m. This can be explained by the fact that small values of I_m produce a small number of mutants and, consequently, a soft perturbation in the population. Therefore, the algorithm has a fast convergence to a local optimum. The opposite situation is observed as well. Larger values of I_m lead to a large perturbation in the population, which contributes to increase the computational time since the algorithm will spend more time to converge for local optima.**

The plots in Figures 5 and 6 summarize the results for all evaluated variants, displaying points whose coordinates are the values of the ADev and TTime metrics for each combination of parameter values. We note that there is no correlation between processing time and solutions quality.

TABLE 2. Summary of the numerical results obtained **by each of the** fifteen variants of the BRKGA

BRKGA	CAB dataset				AP dataset			
(I_e,I_m)	ADev(%)	#Best	Score	TTime	ADev (%)	#Best	Score	TTime
(0.15, 0.15)	0.67	21	115	1401.60	1.02	6	103	828.6
(0.15, 0.20)	0.80	22	110	1548.80	1.07	6	87	862.5
(0.15, 0.25)	0.61	24	81	1518.10	0.99	6	91	1035.2
(0.15, 0.35)	0.49	25	55	1822.10	0.92	7	94	1187.0
(0.15, 0.45)	0.68	22	109	2264.90	0.80	7	85	1674.9
(0.20, 0.15)	0.59	22	111	1410.00	0.90	6	103	868.0
(0.20, 0.20)	0.56	22	73	1516.20	1.00	7	76	951.8
(0.20, 0.25)	0.73	22	117	1700.10	0.83	8	93	1021.9
(0.20, 0.35)	0.64	21	102	1948.40	0.66	6	61	1615.1
(0.20, 0.45)	0.55	22	84	2983.70	0.88	7	88	2392.4
(0.25, 0.15)	0.52	22	99	1507.40	1.03	5	103	1040.7
(0.25, 0.20)	0.60	23	86	1764.50	0.89	5	91	1144.5
(0.25, 0.25)	0.69	23	103	1863.80	1.15	6	113	1330.4
(0.25, 0.35)	0.79	21	130	2386.50	1.24	5	125	1897.6
(0.25, 0.45)	0.54	22	85	4013.80	0.88	8	95	3464.6
Contreras	0.0005	35	1	-	0.01	26	2	-

Tables 3 and 4 detail the best results obtained over all BRKGA variants for each instance. The first three columns give the instances chacteristics. Column 4 presents the optimal values for these instances, except for those marked with an asterisk. The percent deviation with respect to the optimal solution is showed in column 5. The next two columns under *Time(seconds)* give the smallest time to **our** BRKGA and **the results on the MIP proposed by Contreras et al. (2010)** , respectively, to reach the best solution found. Even though the methods run on different machines (Contreras' methods run on a 2.33GHz Intel Core2 processor with 3GB RAM computer running Windows) and computational times reported might not be comparable we report them for completeness. Each table shows the new best solutions found by BRKGA. For the CAB (n=25, p=8, α=0.8) instance, BRKGA improved the best known solution by 0.02% . The improvement for the AP (n=25, p=8, α=0.8) instance was 0.26%. The average percent deviation over the AP dataset and the CAB dataset was only, 0.15% and 0.21%, respectively.

6. Concluding remarks

In this paper we presented a biased random-key genetic algorithm (BRKGA) for the tree of hubs location problem, whose potential applications appear in telecommunications and in transportation logistics.

The computational experiments showed that BRKGA **was** a robust and effective approach to tackle this problem. The solutions found presented small differences when compared among fifteen variants of BRKGA. Besides, our method was able to produce new best-known solutions for two out of five instances with unknown optimal values, and to reach the optimal solutions for 48 out of 58 instances. On the remaining instances, a small percent deviation to the optimal solutions was reached in small running times.

TABLE 3. Numerical results for CAB instances: the best solutions found among all variants of BRKGA and processing times. **Note that BRKGA experiments were performed on a 3.16GHz Intel Core2-Duo processor while MIP run on a 2.33GHz Intel Core2.**

n	p	α	Optimum	%Dev	Time(seconds)	
					BRKGA	**MIP**
10	3	0.2	494.50	0.00	0.2	0.2
		0.5	613.00	0.00	0.2	0.2
		0.8	719.00	0.00	0.2	0.7
	5	0.2	322.90	0.00	1.4	0.8
		0.5	499.40	0.00	2.1	2.9
		0.8	667.40	0.00	3.5	10.4
	8	0.2	190.50	0.00	2.6	10.5
		0.5	411.80	0.00	3.0	18.3
		0.8	631.60	0.00	3.1	34.8
15	3	0.2	1,915.20	0.00	3.6	3.4
		0.5	2,324.40	0.00	4.3	57.9
		0.8	2,666.10	0.00	7.2	93.9
	5	0.2	1,299.60	0.00	8.9	22.3
		0.5	1,935.10	0.00	7.9	415.9
		0.8	2,454.20	0.00	14.6	2953.7
	8	0.2	876.40	0.00	11.5	4320
		0.5	1,590.30	0.00	12.7	2317.8
		0.8	2,250.30	0.00	13.8	10980
20	3	0.2	4,170.10	0.00	12.3	7.6
		0.5	5,234.90	0.00	60.2	18.6
		0.8	6,279.40	0.49	31.8	351.5
	5	0.2	2,808.70	0.00	54.5	67.3
		0.5	4,384.30	0.00	34.9	604.0
		0.8	5,663.50	1.15	46.5	6660
	8	0.2	2,057.00	0.00	61.8	1798.2
		0.5	3,700.20	0.03	65.2	24240
		0.8	5,283.10*	0.56	60.7	144000
25	3	0.2	6,554.60	0.00	53.4	15.3
		0.5	8,274.00	0.00	177.8	36.6
		0.8	9,923.90	0.00	161.7	1131.2
	5	0.2	4,791.10	0.00	136.3	430.0
		0.5	7,190.70	0.34	130.5	16440
		0.8	9,173.40	0.00	192.4	28200
	8	0.2	3,752.90	1.72	165.9	15780
		0.5	6,272.90*	1.04	212.9	144000
		0.8	8,756.70*	**-0.02 (8755.0)**	216.83	144000
Average				0.15		

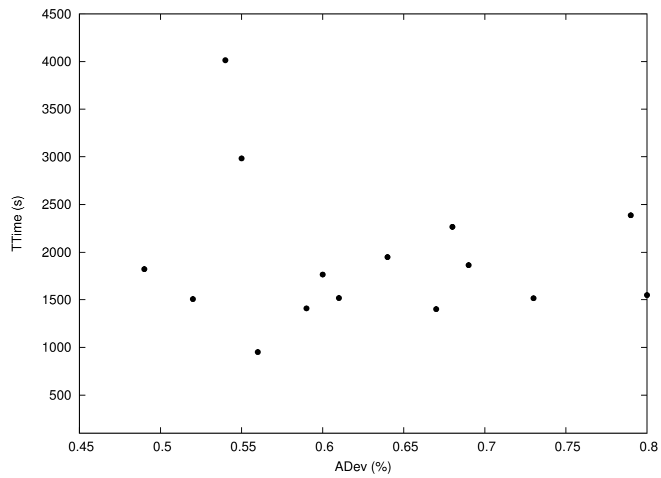

FIGURE 5. Average deviation from the best value and total running time for 15 different variants of BRKGA on the CAB instances: each point represents a unique combination of parameters I_e, and I_m **as detailed in Table 2.**

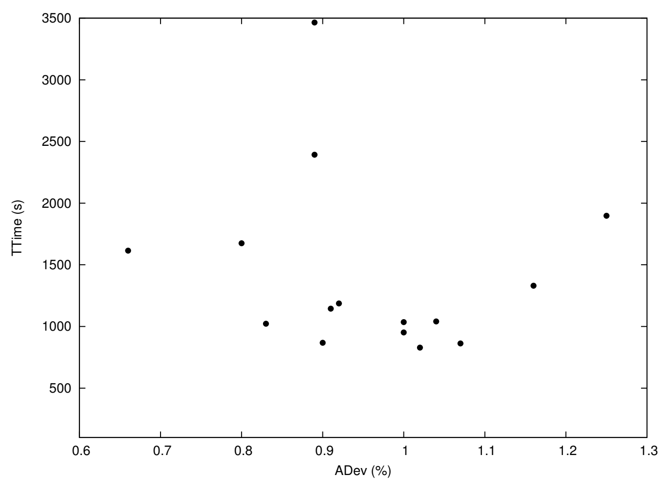

FIGURE 6. Average deviation from the best value and total running time for 15 different variants of BRKGA on the AP instances: each point represents a unique combination of parameters I_e, and I_m **as detailed in Table 2.**

TABLE 4. Numerical results for AP instances: the best solutions found among all variants of BRKGA and processing times. **Note that BRKGA experiments were performed on a 3.16GHz Intel Core2-Duo processor while MIP run on a 2.33GHz Intel Core2.**

n	p	α	Optimum	%Dev	Time(seconds)	
					BRKGA	**MIP**
10	3	0.2	52,541.0	0.00	1.7	0.5
		0.5	63,166.8	0.00	1.1	0.6
		0.8	72,640.8	0.00	0.2	1.1
	5	0.2	34,340.0	0.00	0.6	0.9
		0.5	49,418.7	0.00	2.2	3.2
		0.8	64,013.2	0.00	2.6	5.6
	8	0.2	20,513.4	0.00	3.6	11.5
		0.5	39,288.1	0.00	4.4	23.6
		0.8	57,953.4	0.00	2.9	36.6
20	3	0.2	58,761.2	0.00	35.5	5.9
		0.5	69,515.9	0.00	15.8	38.0
		0.8	78,177.6	0.00	39.6	60.1
	5	0.2	46,480.4	0.00	32.5	71.6
		0.5	61,061.4	0.13	30.8	436.8
		0.8	73,592.9	0.07	33.2	2780.5
	8	0.2	35,421.0	0.00	80.1	1637.7
		0.5	52,294.3	0.00	64.6	7980
		0.8	68,272.7	0.00	62.6	81060
25	3	0.2	60,602.3	0.00	95.6	26.3
		0.5	70,130.9	1.97	58.7	122.9
		0.8	79,442.4	0.31	89.2	488.1
	5	0.2	47,432.7	0.00	132.7	278.4
		0.5	61,046.7	1.58	100.8	1175.5
		0.8	73,569.9	0.51	138.5	9060
	8	0.2	37,295.6	1.36	128.6	9000
		0.5	54,318.7*	0.00	122.9	144000
		0.8	70,072.5*	**-0.26 (69,890.6)**	140.14	144000
Average				0.21		

References

J. C. Abreu Júnior, T. F. Noronha, and A. C. Santos. O problema de localização de concentradores em árvores: um procedimento para acelerar um algoritmo baseado em decomposição de Benders. In *XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, page 9p., Porto de Galinhas, Brazil, 2015.

N. Adler and N. Hashai. Effect of open skies in the middle east region. *Transportation Research Part A: Policy and Practice*, 39(10):878 – 894, 2005.

S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190:1–21, 2008.

J. E. Beasley. Population heuristics. In P.M. Pardalos and M.G.C. Resende, editors, *Handbook of Applied Optimization*, pages 168–183. Oxford University Press, 2002.

O. Berman, Z. Drezner, and G. O. Wesolowsky. The transfer point location problem. *European Journal of Operational Research*, 179(3):978 – 989, 2007.

H. Chen, A. M. Campbell, and B. W. Thomas. Network design for time-constrained delivery. *Naval Research Logistics*, 55:493–515, 2008.

A. A. Coco, T. F. Noronha, and A. C. Santos. Algoritmo baseado em BRKGA para o problema de árvore geradora mínima robusta. In *Proceedings of the XLV Simpósio Brasileiro de Pesquisa Operacional (SBPO), Natal, Brazil*, pages 3658–3669, 2013.

I. Contreras, E. Fernández, and A. Marín. Tight bounds from a path based formulation for the tree of hub location problem. *Computers & Operations Research*, 36(12):3117 – 3127, 2009.

I. Contreras, E. Fernández, and A. Marín. The tree of hubs location problem. *European Journal of Operational Research*, 2:390–400, 2010.

T. F.G. Costa, G. Lohmann, and A.V.M. Oliveira. A model to identify airport hubs and their importance to tourism in Brazil. *Research in Transportation Economics*, 26(1):3 – 11, 2010. ISSN 0739-8859. Economics of Transport for Tourism.

C. B. Cunha and M. R. Silva. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *European Journal of Operational Research*, 179(3):747 – 758, 2007.

R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096 – 1109, 2013.

D. B. M. M. Fontes and J. F. Gonçalves. A multi-population hybrid biased random key genetic algorithm for hop-constrained trees in nonlinear cost flow networks. *Optimization Letters*, 7(6):1303–1324, 2013. ISSN 1862-4472.

S. Gelareh and S. Nickel. A Benders decomposition for hub location problems arising in public transport. In Jörg Kalcsics and Stefan Nickel, editors, *Operations Research Proceedings 2007*, volume 2007 of *Operations Research Proceedings*, pages 129–134. Springer Berlin Heidelberg, 2008.

J. F. Gonçalves and M. G. C. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17:487–525, 2011.

S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

R. Ishfaq and C. R. Sox. Design of intermodal logistics networks with hub delays. *European Journal of Operational Research*, 220(3):629–641, 2012.

S. Limbourg and B. Jourquin. Optimal rail-road container terminal locations on the European network. *Transportation Research Part E: Logistics and Transportation Review*, 45(4):551 – 563, 2009.

M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 1987.

M. G. C. Resende. Biased random-key genetic algorithms with applications in telecommunications. *TOP*, 20:120–153, 2012.

M.G.C. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path-relinking for the max-min diversity problem. *Computers and Operations Research*, 37:498–508, 2010.

C. C. Ribeiro, E. Uchoa, and R. F. Werneck. A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS Journal on Computing*, 14:228–246, 2002.

E. Ruiz, M. Albareda-Sambola, E. Fernández, and M. G. C. Resende. A biased random-key genetic algorithm for the capacitated minimum spanning tree problem. *Computers & Operations Research*, 57:95 – 108, 2015.

E. M. Sá, R. S. Camargo, and G. Miranda. An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185 – 202, 2013.

E. M. Sá, I. Contreras, and J. F. Cordeau. Exact and heuristic algorithms for the design of hub networks with multiple lines. *European Journal of Operational Research*, 246(1):186–198, 2015.

W. M. Spears and K. A. DeJong. On the virtues of parameterized uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236, 1991.

J. J. Wang and M. C. Cheng. From a hub port city to a global supply chain management center: a case study of Hong Kong. *Journal of Transport Geography*, 18(1):104 – 115, 2010.

(Luciana S. Pessoa) Department of Industrial Engineering, PUC-Rio, Rua Marquês de São Vicente, 225, Gávea - 22453-900 Rio de Janeiro, RJ, Brazil.

E-mail address, Luciana S. Pessoa: `lucianapessoa@esp.puc-rio.br`

(Andréa C. Santos) ICD-LOSI, UMR CNRS 6281, Université de Technologie de Troyes, 12, rue Marie Curie, CS 42060, 10004, Troyes CEDEX, France.

E-mail address: `andrea.duhamel@utt.fr`

(Mauricio G. C. Resende) Mathematical Optimization and Planning (MOP), Amazon.com, Seattle, WA 98109, USA.

E-mail address: `resendem@amazon.com`