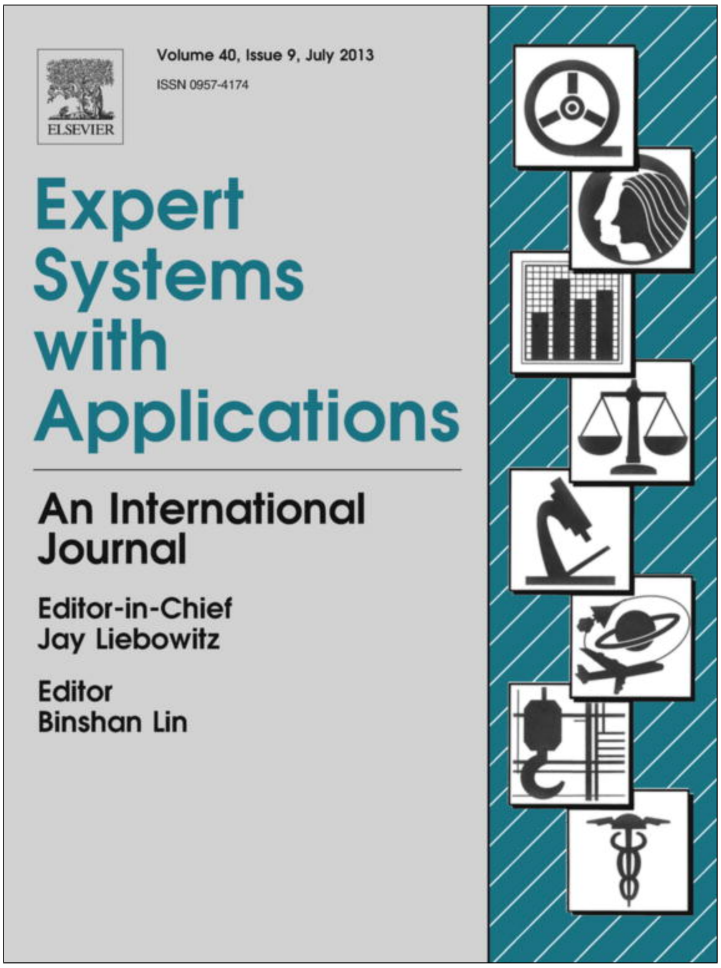
Volume 40, Issue 9, July 2013
ISSN 0957-4174
ELSEVIER
Expert Systems with Applications
An International Journal
Editor-in-Chief
Jay Liebowitz
Editor
Binshan Lin

Expert Systems with Applications 40 (2013) 3471–3480

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Network infrastructure design with a multilevel algorithm

Hideson Alves da Silva [a,b], Alceu de Souza Britto Jr. [b,*], Luiz Eduardo Soares de Oliveira [c], Alessandro L. Koerich [b,c]

[a] *Companhia Paranaense de Energia – COPEL, Brazil*
[b] *Pontifícia Universidade Católica do Paraná – PUC-PR, Brazil*
[c] *Universidade Federal do Paraná – UFPR, Brazil*

ARTICLE INFO

Keywords:
Network design
Multilevel optimization process
Genetic algorithm
Particle swarm optimization

ABSTRACT

This paper presents a novel algorithm to deal with the network design problem, which optimizes the network levels considering their interdependency. The idea is to design a low cost and optimized network providing the number and the geographic location of devices as well as the links among them for each network level, while taking into account the existing dependency among them. In addition, a new database composed of real and georeferenced data is created and make available for the research community. This database contains three datasets that represent distinct projects related to geographic regions of the city of Curitiba (Brazil). The experimental results show that the proposed algorithm provides a significant cost reduction in the network design. The savings of this proposal go from 1% to 40% (depending on the network size, number of levels and demand nodes), making it attractive for companies that spend a considerable amount of resources in network projects and deployment.

1. Introduction

The Network Design Problem (NDP) is present in important areas of the human activity, such as the planning of telecommunication, electric power, water, and gas networks. The potential number of applications and the importance of them have been the main motivation for the proposal of many network design methods (Chen, Kim, Lee, & Kim, 2010; Chen & Xu, 2012; Ding & Ishii, 2000; Esbensen, 1995; Huy & Nghia, 2008; Martins, Pardalos, Resende, & Ribeiro, 1999; Martins, Ribeiro, & Souza, 1998; Ribeiro & Souza, 2000; Xu, Chiu, & Glover, 1995; Xu, Wei, & Wang, 2009; Zhan, Zhang, Li, & Chung, 2009; Zhong, Huang, & Zhang, 2008). These interesting contributions share the same challenge related to provide a low cost infrastructure plan for a network by scaling and defining the devices and the links among them in order to meet a known demand. The most common approach found in the literature treats the NDP as a Steiner-tree Problem (STP) and, in addition, it considers the planning of each network level separately. In this context, an interesting approach is proposed by Huy and Nghia (2008), where the authors apply a parallel genetic algorithm that uses a *fitness* function based on the *Distance Network Heuristic* (DNH). They achieved promising results through experiments carried out on the OR-library (Beasley, 2010) when compared with related works based on different meta-heuristics, such as the Esbensen's genetic algorithm with graph reduction (Esbensen, 1995), the greedy randomized adaptive search procedure (GRASP) Martins et al. (1999), a parallel approach of the GRASP method (PGRASP) Martins et al. (1998), and the Tabu Search (TS) Ribeiro and Souza (2000).

Similar to the approach proposed by Huy and Nghia (2008), Ding and Ishii (Ding & Ishii, 2000) employ the *Steiner-tree* to model the NDP in their solution. However, they have used a dynamic version of this structure and an Online Genetic Algorithm (OLGA) combined with the PRIM algorithm for *fitness* evaluation. They have achieved promising results that surpassed other *fitness* functions, such as the DNH, the Shortest Path Heuristic (SPH) and the Average Distance Heuristic (ADH). Another interesting approach proposed by Wen-Liang Zhong (2008) introduces a new discrete Particle Swarm Optimization (PSO) algorithm to deal with the *Steiner-tree Problem*. The experimental results using the OR-library have shown that the proposed algorithm is better than the Genetic Algorithm (GA) and the original PSO. In the same direction, Zhan et al. (2009) propose a modified PSO algorithm that speeds up the processing by considering only the promising solutions (particles) for *fitness* evaluation. Besides GA and PSO, other searching techniques have been employed to solve the STP, such as the Tabu Search (Ribeiro & Souza, 2000; Xu et al., 1995). Other proposals for traffic networks (Kampstra, van der Mei, & Eiben, 2011; Zhang, Lu, & Xiang, 2008; Xiao, Wang & Du, 2008). Urban railway networks (Marin & Rodenas, 2009), water networks (Bolognesi, Bragallia, Marchia, & Artinaa, 2010), power distribution (Cadini, Zio, & Petrescu, 2010; Li, Wang,

* Corresponding author. Tel.: +55 41 3271 1669; fax: +55 41 3271 2121.
E-mail addresses: hideson@copel.com (H. A. d. Silva), alceu@ppgia.pucpr.br (A. de Souza Britto Jr.), lesoliveira@inf.ufpr.br (L. E. S. d. Oliveira), alekoe@ppgia.pucpr.br (A.L. Koerich).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.055

Xie, & Xie, 2009) and telecommunication networks (Kampstra et al., 2011; Mateus, Luna, & Sirihal, 2000) may be found in the literature.

However, the drawback of the aforementioned contributions is that they consider the optimization of each network level in separate, what is not suitable, since most of time the networks present multiple levels. In addition, these levels are mutually dependent. In other words, the infrastructure of each level depends on the infrastructure of the adjacent ones. Thus, we consider that the network design is still an open problem. To make it clear, let us focus on the design of the infrastructure of telecommunication networks, which is the case study of this paper. Fig. 1 shows a diagram that represents a three-level network. At the first level (*last mile*) are the clients (*demands*), which must be met by facility nodes represented by red circles. At the second level, the red squares are facility nodes, while the crossed square represents a facility node of the third level of the network.

A cost-effective design of the network infrastructure should address the interdependence between adjacent levels. In other words, to plan the *last-mile* a set of access devices (facilities) is necessary to meet a set of clients (demands). To design the second level, a definition of middle devices is necessary to support the access devices. Finally, the definition of the main devices is necessary to design the third level.

With this in mind, the contribution of this paper is twofold. Firstly, we have built a new database composed of real and georeferenced data, which is currently available at (http://www.ppgia.pucpr.br/~net_datasets). This database contains three datasets that represent distinct projects related to geographic regions of the city of Curitiba in Brazil. Secondly, we propose an optimization algorithm to deal with the NDP that considers the interaction between the devices of the multiple levels of the network infrastructure. The idea is to design a low cost network knowing for each level the number and the geographic location of devices and the links among them. Besides, the existing dependency and interaction among the devices of each level has to be taken into account. The same problem has been tackled by (Silva, Britto, Jr., L. E. S., & A. L., 2011), however, the new approach proposed in this paper considers the binary PSO algorithm as an alternative for the internal optimization process. In addition, the experimental protocol is now based on new datasets and the experiments are extended to networks with more than three levels. The experimental results show that the proposed algorithm provides a significant cost reduction in the network project. The savings achieved by the proposed approach go from 1% to 40% (depending on the network size, number of levels and number of demand nodes). This makes the proposed approach very attractive to companies that spend a considerable amount of resources in network projects and deployment.

This paper is organized into five sections. Section 2 describes the modeling of the NDP. Section 3 presents the proposed algorithm, while Section 4 presents the experiments used to evaluate the proposed method. Conclusions and future works are presented in the last section.

2. Problem modeling

Let us consider a network with L levels, where each individual level (ℓ) is represented by a graph denoted by $G(V_\ell, E_\ell)$, where V_ℓ represents the vertices (or nodes) and E_ℓ all possible edges. In this special graph, V_ℓ is organized into distinct subsets according to three possible kinds of nodes: M_ℓ contains the demand nodes, N_ℓ contains the facility nodes, and A_ℓ contains the ascending nodes. This last subset enables the idea of planning each network level considering information from the ascending levels. Beyond that, let us consider some additional data structures necessary to keep information related to the elements of G: x_{ij} that represents the cost of the path from the ith demand to the jth facility node; y_{ij} that represents the cost of the path from the ith facility to the jth ascending node; n_i that keeps the current binary status of the ith facility node (1-on/0-off); a_i that maintains the binary status of the ith ascending node (1-on/0-off); t_i that represents the cost of the ith facility node; and c_i that represents the cost of the ith ascending node.

From this modeling strategy, a possible solution is represented by a subgraph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$. However, looking for such a subgraph is considered a NP-complete problem (Johnson, Lenstra, & Rinnoov Kan, 1978). An alternative is to consider optimization strategies to solve it, such as the proposed algorithm described in the next section.

3. Proposed multilevel algorithm

The main contribution of the algorithm described in this section is the strategy used to optimize the entirely network, reprocessing levels when necessary. In other words, it takes into account the

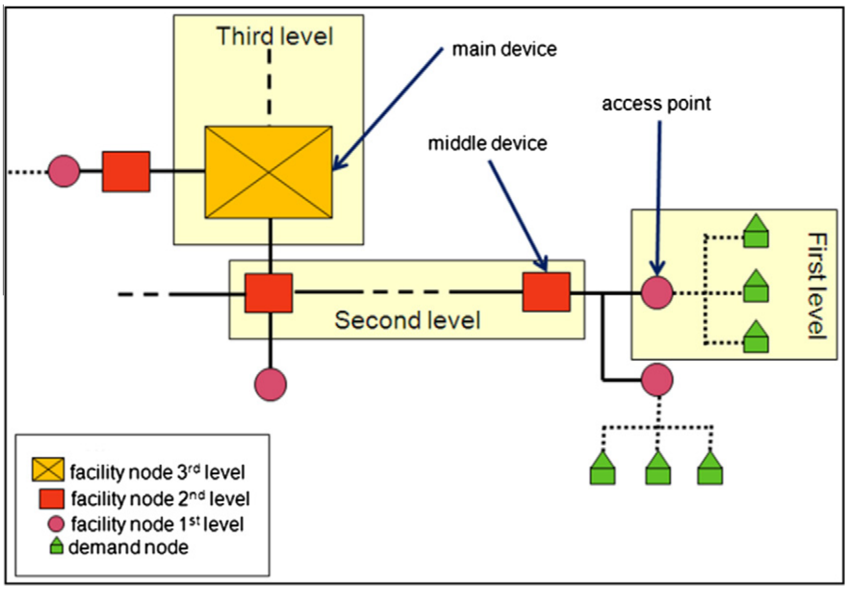

Fig. 1. Diagram of a three-level network structure.

existing interdependency of the L levels of a network. This means that the design of one level affect the design of the adjacent ones. Algorithm 1 presents the proposed Multilevel Algorithm for Network Design (MAND).

Algorithm 1: Multilevel Algorithm for Network Design

```
1   MAND ( M )
2   L: number of levels;
3   ℓ: current level;
4   cost_level[L] = {max ,…, max};
5   got_improved[L] = {true, …, true};
6   begin
7     ℓ = 1;
8     while ( (ℓ+1) < L )
9       while (got_improved[ℓ] or got_improved[ℓ+1] or got_improved[ℓ+2])
10        while (got_improved[ℓ] or got_improved[ℓ+1])
11          cost = Level_Processing(ℓ);
12          if (cost < cost_level[ℓ]) then
13            cost_level[ℓ]=cost;
14            got_improved[ℓ]=True;
15          else
16            got_improved[ℓ]=False;
17          endif
18          cost = Level_Processing(ℓ+1);
19          if (cost < cost_level[ℓ+1]) then
20            cost_level[ℓ+1]=cost;
21            got_improved[ℓ+1]=True;
22          else
23            got_improved[ℓ+1]=False;
24          endif
25        endwhile
26        while (got_improved[ℓ+2] or got_improved[ℓ+1])
27          cost = Level_Processing(ℓ+2);
28          if (cost < cost_level[ℓ+2]) then
29            cost_level[ℓ+2]=cost;
30            got_improved[ℓ+2]=True;0
31          else
32            got_improved[ℓ+2]=False;
33          endif
34          cost = Level_Processing(ℓ+1);
35          if (cost < cost_level[ℓ+1]) then
36            cost_level[ℓ+1]=cost;
37            got_improved[ℓ+1]=True;
38          else
39            got_improved[ℓ+1]=False;
40          endif
41        endwhile
42      endwhile
43      if (got_improved[ℓ] and ℓ > 1) then
44        ℓ=ℓ-1;
45      else
46        ℓ=ℓ+1;
47      endif
48    endwhile
49    return ∑cost_level[L];
50  end.
```

As one may see in line 1, the input data is the set of demands (M), which represents the clients to be met at the first level of the network. This data is used to process the level (ℓ) in order to find an initial set of facilities (also known as *last-mile*). Initially, besides the path costs, the only information used to compute the cost is the value associated with the set of facility nodes at level ℓ. In a further iteration, when level $\ell+1$ has already been processed at least once, it is possible to re-evaluate the facilities defined for level ℓ, but now considering the cost of the ascending nodes, that are the facility nodes estimated for the level $\ell+1$.

The proposed algorithm has four main loops. The first loop (*while* $(\ell+1) < L$, in line 8), is used to control the number of iterations based on the number of levels (L) of the network. The next loop (*while got_improved* $[\ell, \ell+1$ and $\ell+2]$, in line 9) is used to control three adjacent levels while it is still possible to find better solutions. The two innermost loops implement the execution of two levels each, and control their iterations. The first loop executes levels ℓ and $\ell+1$ (line 10), and the second loop executes level $\ell+2$ and $\ell+1$ (line 26). In short, in the proposed algorithm there is a conditional control to check the number of levels to design (first loop), and there is a loop to control the production of better solutions (second loop). The innermost loops control the interaction between adjacent levels by using the ascending nodes.

In other words, by using the information from ascending levels, it is possible to search for better solutions considering the interdependence among levels. Thus, when the current level is ℓ, it is possible to use information from level $\ell+1$, if the current level is $\ell+1$, it is possible to use information from level $\ell+2$, and so on. This allows us to optimize the entire network infrastructure. The set of ascending nodes is only known after the first iteration of subsequent levels. This means that in the first iteration of a level, the solution is always penalized, as it does not have the information about facilities nodes of the subsequent level.

Algorithm 2: Level Processing Module

```
Level_Processing (ℓ)
  if (exist_level(ℓ + 1)) then
    A[ℓ] ← N[ℓ + 1]
    cost_level=Optimization(ℓ,M,N,E,A)
    cost_level+=Pathfinder (ℓ,M,N,E,A)
  return cost_level;
end.
```

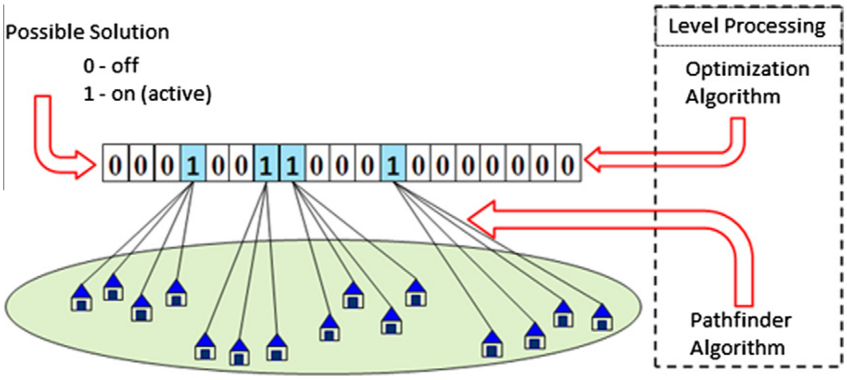

Fig. 2. overview of the level processing subroutine.

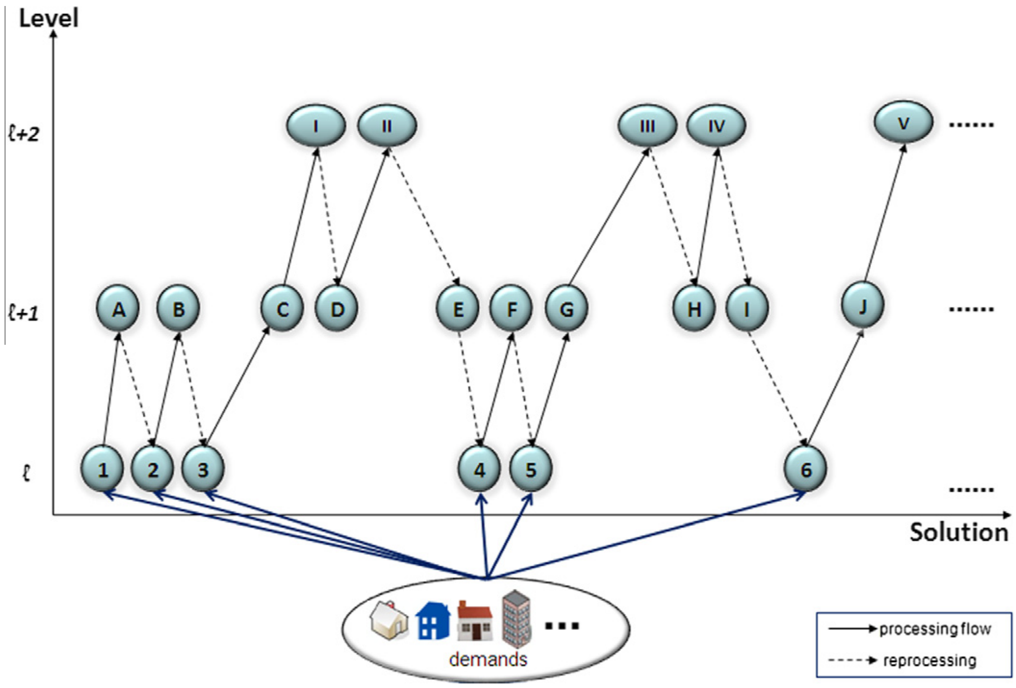

Fig. 3. Trellis of the MAND.

3.1. Level processing

The *Level_Processing*(ℓ) subroutine (see Algorithm 2) is responsible for searching the number and geographic location of the devices; and defining the path among them. It means to find paths between devices, devices and demands, and devices and their ascending nodes. Fig. 2 presents the two stages of the level processing. A binary vector represents the number of devices available at each level. In this case, 0 (zero) and 1 (one) stands for active and not active devices, respectively. In the first stage, for finding the facility nodes, it is necessary to know which demand nodes must be met (optimization stage). Different evolutionary algorithms may be used, such as GA (Goldberg, 1989) and PSO (Kennedy & Eberhart, 1942). Afterwards, with the demand and facility nodes obtained in the first stage, it is necessary to find the best paths among them (*pathfinder stage*). For this second stage, a graph search algorithm is necessary.

The optimization algorithm used in *Level_Processing* assesses its objective function. For such an aim, it takes into account the set of demand nodes of the current level and the set of facility nodes of the next level (the ascending nodes). With such a data, the optimization algorithm finds a set of facility nodes (where for each element is defined the status on or off), then it is defined the best path, and finally it is obtained the respective cost of the current level.

Fig. 3 illustrates the trellis of the proposed MAND. It represents the dealings between levels. Firstly, it generates the solution '1' at level ℓ, in which the found arrangements are related directly to the first demands (the clients). After that, it generates the solution 'A', at level $\ell + 1$ that meets the facility nodes of solution '1'. Now with the facility nodes of solution 'A' that represents ascending nodes for level ℓ, solution '1' is reprocessed and it generates solution '2', and so on. When it is not possible to improve a solution for the first innermost loop (level ℓ and level $\ell + 1$), the next innermost loop is processed; in this case levels $\ell + 1$ and $\ell + 2$ are processed with the same logic of the previous process. Following the example in Fig. 3, solution '6' in level ℓ is the best solution to meet the demands of the first level of the network. The next level ($\ell + 1$) has the best solution 'J' to meet solution '6'. Finally, the level $\ell + 2$ has the best solution 'V' to meet solution 'J'.

In the experiments presented in this paper, we evaluate two single objective algorithms (AG and PSO) at the first stage of the Level Processing subroutine. The following section describes the objective function and the pathfinder algorithm used in the experiments.

3.2. Objective function

An optimization process depends on a fitness function to indicate how close a given solution is to achieving the defined specification. Eq. (1) is the objective function used by the optimization process embedded into the proposed algorithm. It represents the cost of all activated nodes plus the cost of the path to link them. The idea is to minimize the cost of the network implementation. It sums up the path length that is necessary to connect demand nodes with facilities nodes. When there are ascending nodes, the cost of the corresponding connections is also taken into account.

$$C_\ell = MIN\left(\sum_{i\in M_l}\sum_{j\in N_l} x_{ij}n_i + \sum_{i\in N_\ell} t_i n_i + \sum_{i\in N_\ell}\sum_{j\in A_\ell} y_{ij}a_i + \sum_{i\in N_\ell} c_i a_i\right) \tag{1}$$

Recalling that M_ℓ, N_ℓ and A_ℓ denote the demand, the facility, and the ascending nodes, respectively. In addition, x_{ij} represents the cost of the path from the ith demand to the jth facility node; y_{ij} represents the cost of the path from the ith facility to the jth ascending node; n_i keeps the current binary status of the ith facility node (1-on/0-off); a_i maintains the binary status of the ith ascending node (1-on/0-off); t_i represents the cost of the ith facility node; and c_i represents the cost of the ith ascending node.

Eqs. (2) and (3) impose constraints to the problem, that only facility and ascending nodes that are activated (status on) must be considered in the Eq. (1)

$$n_i \in \{0,1\}, \quad i \in N_\ell \tag{2}$$

$$a_i \in \{0,1\}, \quad i \in A_\ell \tag{3}$$

The cost values inside the arrays t and c may be real or hypothetical ones. In the experiments carried out in this paper, the used monetary costs increase from the first to the last level proportionally. In fact, the installation cost of devices at any level is ten times more expensive than the previous one.

3.3. Path finding

The best path between nodes is determined taking into account the cost estimated to link the nodes using suitable cables. Thus, the general idea is to minimize the distance between nodes. A common strategy is to use the Euclidian distance to calculate the final cost of each path. However, when geographic restrictions must be considered during the network planning such a metric is not suitable. An alternative is to use more elaborated algorithms such as (Dijkstra, 1959), A* (A Star) Hart, Nilsson, and Raphael (1968) or (Johnson, 1977). The Euclidian distance is less time consuming, when compared with more sophisticated pathfinder algorithms. To overcome this problem, we have added in the Dijkstra algorithm some additional data structures to store the paths already calculated in order to reduce the time consumed.

The final cost of a path is calculated based on the distance and the monetary cost of cables. In real-world networks, the cable cost varies through the different levels of the network. Thus, we have considered monetary cost of 1 for the first level, 2 for second level, 3 for third level, and so on. The relation between the monetary costs of device installation and cables has influence on the number of devices in the solution. If more expensive devices are used, then it is advisable to use less devices and build more paths. In another situation, the cables could be more expensive, and then it is better to add more devices in the solution.

4. Experimental results

This section presents the experiments undertaken to evaluate the proposed algorithm. The MAND was implemented in C programming language, and the reported results were obtained on a PC-Pentium Dual-Core E5300, 2.6 GHz with 2 Gb of memory, the operational system is Ubuntu Linux release 8.04.4. The experiments considers in the level processing module single optimization algorithms (GA or PSO), and Dijkstra as a pathfinder. The new database containing three subsets of georeferenced data was used. All the results are the average of five runs and the MAND is compared with the most common strategy available in the literature, named CS in the experiments. The CS represents the algorithms where the planning of each level is done individually.

4.1. Database

The databases available in the literature, such as the OR-library, are not suitable for the assessment of MAND, since they are not georeferenced. Thus, we have built a new database that is currently available at (http://www.ppgia.pucpr.br/~net_datasets). Table 1 shows the characteristics of the three datasets available. These datasets represent real projects, corresponding to different geographic regions of the city of Curitiba, Brazil. Fig. 4 shows the city map with the selected neighborhoods.

4.2. Experiments with GA

The experiments presented in this section use a bit representation GA (Kangal., 2011) with the following parameters:

- Chromosome length: number of geographic coordinates
- Population size: 60
- Generation: 10,000
- Probability of crossover: 0.95
- Probability of mutation: 0.001
- Sharing: false
- Selection: tournament selection.

The chromosome length is defined based on the number of geographic coordinates that represent the locations where the devices may be installed. The values of the other parameters were empirically defined. Table 2 presents the experimental results for a network with three levels. All the results of the proposed algorithm are compared with a conventional strategy (CS) of optimizing each level in separate (by level). Here, the CS is a GA-based solution. For each level is possible to compare: (1) the number of demands to be met, (2) the number of facilities found and (3) the solution cost in monetary units. At the three last lines of Table 2, it is possible to observe the total cost, time consumed and further statistics about the cost reduction. As one can see, the proposed algorithm provides a very interesting cost reduction in all datasets.

Table 1
Georeferenced datasets.

Dataset	# Demands (clients to be meet)	# Coordinates (local for possible devices)	Area (km^2)
D1	50	390	2.1
D2	105	583	2.5
D3	405	1624	6.1

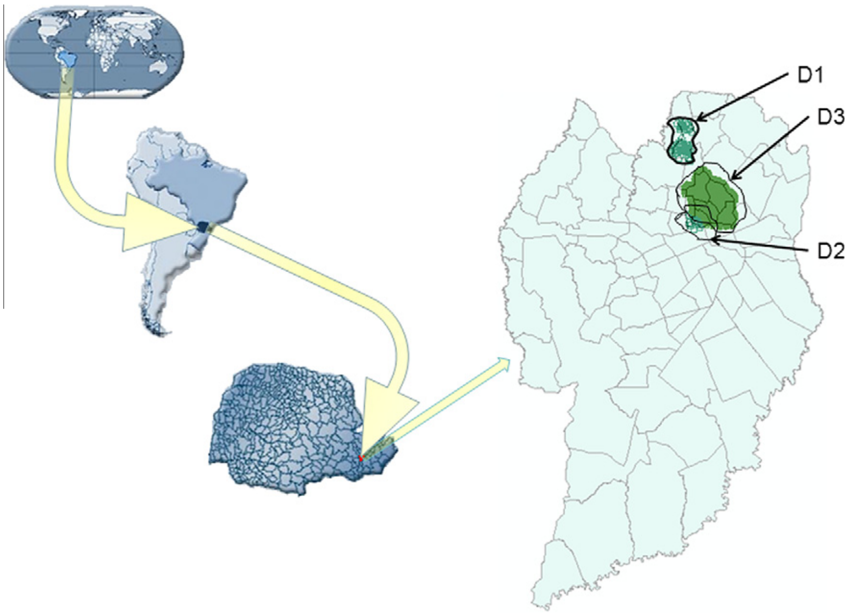

Fig. 4. Geographic view of the dataset examples.

Table 2
Experimental results considering a GA-based single objective optimization algorithm for planning a network with three levels. The average of five runs of the proposed MAND and the most common strategy (CS) in the literature for the datasets D1, D2 and D3.

-MAND with a built-in AG (three-level network)			Dataset D1		Dataset D2		Dataset D3	
			MAND	CS	MAND	CS	MAND	CS
Levels	1	Demands	50	50	105	105	405	405
		Facilities	28	41	46	71	132	184
		Cost	8273	7265	13,968	12,175	52,619	47,373
	2	Demands	28	41	46	71	132	184
		Facilities	6	9	10	16	81	118
		Cost	19,208	25,000	30,023	39,113	122,155	163,191
	3	Demands	6	9	10	16	81	118
		Facilities	1	2	1	1	75	89
		Cost	24,769	32,588	29,535	42,561	788,608	945,602
Total cost			52,250	64,854	73,526	93,849	963,382	1,156,166
Time (in sec.)			5589	245	17,835	742	213,827	8746
Cost reduction			19.43%		21.65%		16.67%	

Table 3
Experimental results considering a PSO-based single objective optimization algorithm for planning a network with three levels. The average of five runs of the proposed MAND and the most common strategy (CS) in the literature for the datasets D1, D2 and D3.

-MAND with a built-in PSO (three-level network)			Dataset D1		Dataset D2		Dataset D3	
			MAND	CS	MAND	CS	MAND	CS
Levels	1	Demands	50	50	105	105	405	405
		Facilities	14	40	23	71	67	176
		Cost	11,457	7803	19,020	13,045	67,700	45,110
	2	Demands	14	40	23	71	67	176
		Facilities	4	10	4	16	15	40
		Cost	14,902	25,331	19,809	39,287	54,641	101,138
	3	Demands	4	10	4	16	15	40
		Facilities	1	2	1	1	3	6
		Cost	20,117	33,412	17,837	43,313	62,645	126,015
Total cost			46,476	66,546	56,666	95,646	184,987	272,262
Time (in sec.)			5326	606	13,922	1575	136,583	14,487
Cost reduction			30.16%		40.75%		32.06%	

4.3. Experiments with PSO

Here, a PSO algorithm replaces the GA inside MAND. The particles are based on a bit representation, and the source code is adapted from the original standard source code of Shi (2011). The parameters used are:

- Dimension: number of geographic coordinates
- Maximum iterations: 10,000
- Number of particles: 60
- Maximum velocity: 10
- Weight: 1.0
- C_1: 2.0 and C_2: 2.0

Table 4
Experimental results considering a AG-based single objective optimization algorithm for planning a network with five levels. The average of five runs of the proposed MAND and the most common strategy (CS) in the literature for the datasets D1, D2 and D3.

-MAND with built-in AG (five-level network)			Dataset D1		Dataset D2		Dataset D3	
			MAND	CS	MAND	CS	MAND	CS
Levels	1	Demands	50	50	105	105	405	405
		Facilities	24	41	46	71	126	184
		Cost	8815	7265	13,979	12,175	52,585	47,373
	2	Demands	24	41	46	71	126	184
		Facilities	6	9	11	16	87	118
		Cost	17,565	25,000	27,397	39,113	123,459	163,191
	3	Demands	6	9	11	16	87	118
		Facilities	1	2	1	1	75	89
		Cost	25,767	32,588	31,241	42,561	795,129	945,602
	4	Demands	1	2	1	1	75	89
		Facilities	1	1	1	1	68	83
		Cost	100,328	107,450	100,141	100,156	6,854,629	8,358,646
	5	Demands	1	1	1	1	68	83
		Facilities	1	1	1	1	68	74
		Cost	1,000,279	1,000,065	1,000,169	1,000,195	67,863,019	74,066,648
Total cost			1,152,755	1,172,368	1172,926	1,194,200	75,688,821	83,581,460
Time (in sec.)			7339	752	26,328	803	390,967	8768
Cost reduction			1.67%		1.78%		9.44%	

Table 5
Experimental results considering a PSO-based single objective optimization algorithm for planning a network with five levels. The average of five runs of the proposed MAND and the most common strategy (CS) in the literature for the datasets D1, D2 and D3.

-MAND with built-in PSO (five-level network)			Dataset D1		Dataset D2		Dataset D3	
			MAND	CS	MAND	CS	MAND	CS
Levels	1	Demands	50	50	105	105	405	405
		Facilities	20	40	23	71	64	174
		Cost	9856	7755	19,316	13,057	69,394	45,101
	2	Demands	20	40	23	71	64	174
		Facilities	6	10	6	16	16	41
		Cost	15,562	24,792	17,552	39,599	50,212	101,888
	3	Demands	6	10	6	16	16	41
		Facilities	1	2	1	2	3	6
		Cost	24,259	33,457	21,594	42,740	67,094	129,803
	4	Demands	1	2	1	2	3	6
		Facilities	1	1	1	1	2	2
		Cost	100,347	104,633	100,180	102,708	207,035	221,049
	5	Demands	1	1	1	1	2	2
		Facilities	1	1	1	1	2	2
		Cost	1,000,236	1,000,270	1,000,167	1,000,259	2,000,588	2,001,753
Total cost			1,150,260	1,170,907	1,158,810	1,198,363	2,394,322	2,499,595
Time (in sec.)			7339	752	19,232	1806	220,884	11,733
Cost reduction			1.76%		3.30%		4.21%	

Similar to the GA-based experiments, all the parameters were obtained empirically from exploratory experiments. Table 3 shows the experimental results of planning a network with three levels. Like in the first set of experiments, all the results of the proposed algorithm are compared with a conventional strategy (CS) of optimizing each level in separate (by level). However, a PSO-based solution is used as CS. As observed before, the MAND surpasses the CS approach in all datasets.

4.4. Experiments with more than three levels

The motivation for these experiments is to show that the proposed algorithm may work with more than three levels. For this purpose, we have carried out new experiments using both versions of the MAND, GA and PSO-based, considering now a five-level network. Tables 4 and 5 show the results. Note that, even with a low cost reduction, the final monetary saving for large networks still represents a significant value.

4.5. Discussion

The experiments undertaken to evaluate MAND does not intent to compare GA and PSO optimization algorithms. However, during the experiments, we have observed that the best solutions for the first network levels were found after a significant different number of iterations of the evolutionary algorithms. Figs. 5 and 6 present, for each level, at which iteration (in average) the best solutions were found. It is possible to observe that the PSO-based MAND has found the best solutions with less than 1000 iterations (one thousand) for each level, while the GA-based run approximately 10,000 (ten thousand) generations just for the first level. In addition, the PSO-based provided the best result for each dataset. This corroborates the observations reported in Papagianni, Papadopoulos, and Tselikas (2008), where GA is compared against PSO.

As one can also see, no matter the optimization algorithm used inside the proposed MAND, it shows a significant increase in the processing time when compared with the strategy where each level is processed in separate. The reason is that in the proposed algorithm some levels are computed more than once as the information from ascending nodes is used to reprocess the previous levels. However, the cost reduction achieved in the experiments justifies the increasing in the computing time. Tables 6 and 7 summarize the cost reduction provided by MAND for each dataset (D1, D2 and D3), considering different optimization strategies.

Fig. 7 presents a comparison between solutions provided by the MAND and the CS approach for the dataset D2. It shows the facility

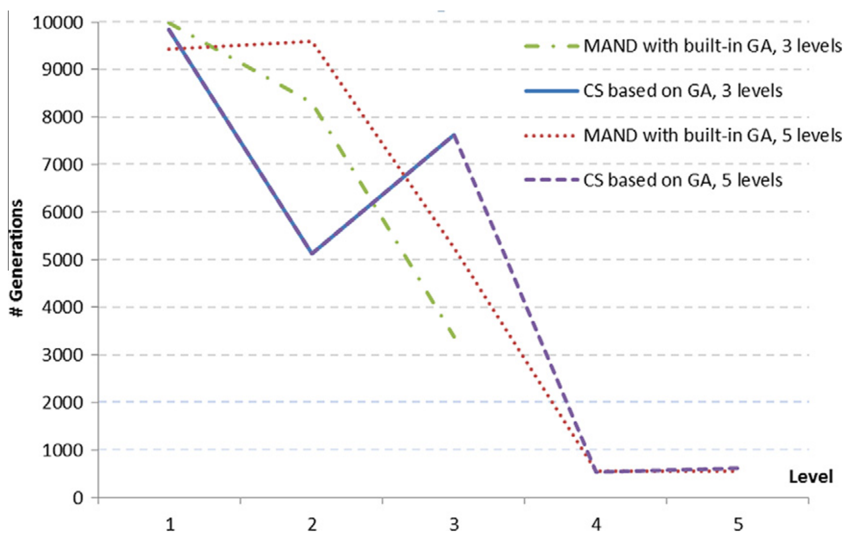

Fig. 5. Number of generations of the GA-based MAND for each network level.

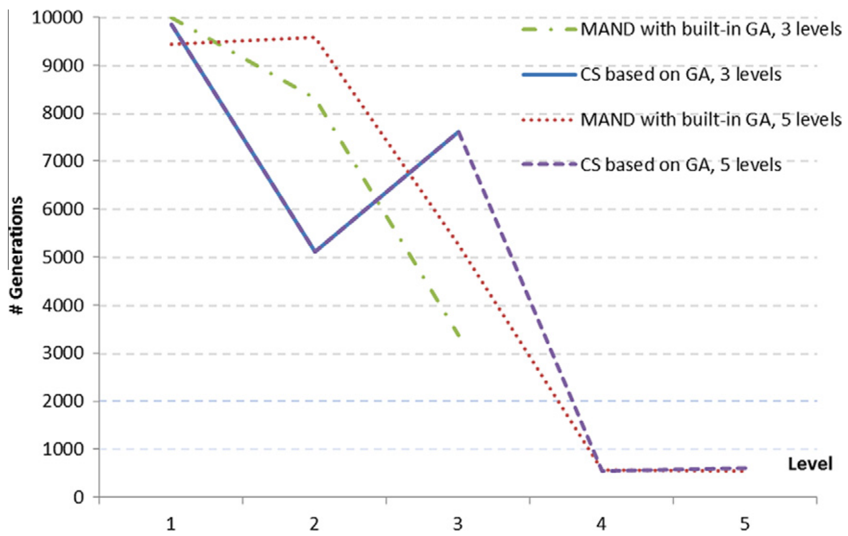

Fig. 6. Number of iterations of the PSO-based MAND for each network level.

Table 6
Observed cost reduction for a three-level network.

Dataset	GA (%)	PSO (%)
D1	19.43	30.16
D2	21.65	40.75
D3	16.67	32.06

Table 7
Observed cost reduction for a five-level network.

Dataset	GA (%)	PSO (%)
D1	1.67	1.76
D2	1.78	3.30
D3	9.44	4.21

nodes of level one (purple circles) to meet the clients (green circles). The nodes of level one met by nodes of the second level (red squares); and finally, the nodes of level two met by nodes of the third level (orange diamond). It is also possible to observe that CS presents a low cost for the first network level, since its solution for this level contains more facility nodes when compared with MAND (72 against 23) reducing the cost of the path to reach the demand nodes. By contrast, a high number of facility nodes in the first level increase the cost of the adjacent levels, as they become demand nodes. At the second level, the MAND is already better than the CS approach (4 against 16). In this case, it is possible to observe a final cost reduction of 42.6%, when MAND is used.

A similar comparison between MAND and the human being performance is done in Fig. 8. Two engineers provided their solutions for the dataset D2. The presented solution corresponds to the best one in terms of cost reduction. This kind of comparison shows that even with a high time consuming, the proposed algorithm is still faster than an engineer for planning the same network using only basic tools, like CAD software and maps. In addition, it should be considered that when an engineer designs a network manually, it is hard to analyze different scenarios.

			MAND	CS
Levels	1	Demands:	105	105
		Facilities:	23	72
		Cost:	19.291	13.013
	2	Demands:	23	72
		Facilities:	4	16
		Cost:	19.149	39.533
	3	Demands:	4	16
		Facilities:	1	1
		Cost:	15820	42.089
	Total Cost		54.260	94.635
	Time (in hours)		5	0.22

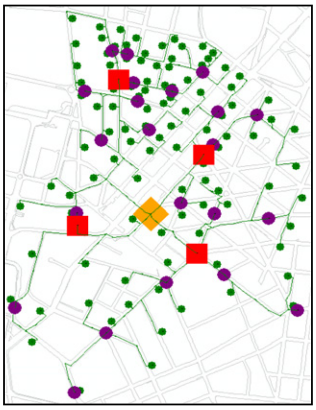
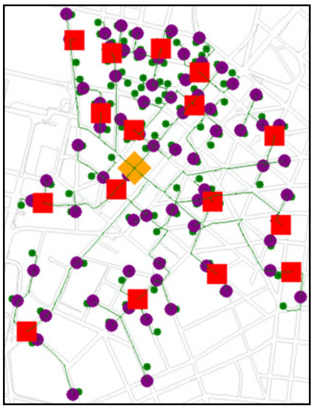

Fig. 7. Comparison of the MAND and the CS approach.

			MAND	Engineer
Levels	1	Demands:	105	105
		Facilities:	23	48
		Cost:	19,291	17,231
	2	Demands:	23	48
		Facilities:	4	3
		Cost:	19,149	59,781
	3	Demands:	4	3
		Facilities:	1	1
		Cost:	15,820	14,737
	Total Cost		54,260	91,570
	Time (in hours)		5	10

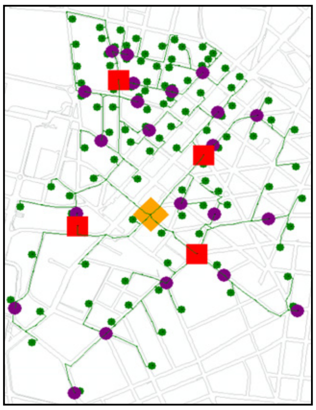
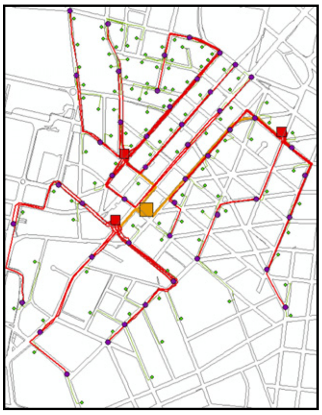

Fig. 8. Comparison of the MAND and the human being (Engineer).

The cost reduction provided by MAND for networks with five levels is lower than the one of a three-level network, but in practice, this percentage is still significant. A statistical evaluation was carried out to check whether there is a significant difference between the costs presented in Table 6. Using the Friedman test (Friedman, 1940), we obtained $p = 0.0070$ for the test set D1; $p = 0.0029$ for the test set D2; and $p = 0.0018$ for the test set D3. These values of $p < 5\%$ for all the datasets prove that the results are significant.

5. Conclusion and future works

In this paper, we have presented a multilevel algorithm to deal with the network design problem. The algorithm performs the design of different levels of a network considering the existing interdependence among them. For this purpose, the information of ascending levels is used to process again a previous one. The experimental results have shown a significant monetary cost reduction when compared with a conventional approach in which the network levels are processed in separate. The savings provided by the proposed algorithm goes from 19.43% to 40.75% in the experiments for networks with three levels, and for networks with five levels, the cost reduction goes from 1.67% to 9.44%.

One aspect that could not be neglected is the increase in the processing time that goes up to 44 times higher than the time consumed by the conventional algorithms to design a network. However, it can be argued that the monetary cost reduction provided by the proposed algorithm is worthwhile. In addition, another benefit of the proposed algorithm is that it opens up the opportunity for engineers to generate several network designs based on different scenarios for further evaluation. Based on that, we may conclude that the MAND algorithm can be applied in the design of real-world instances of network infrastructure, aiding engineers to save time and money.

The proposed method is also an option to build an additional tool for engineers who use Geographic Information Systems (GIS). While GIS is generally used to visualize and print stored data, the proposed method adds intelligence to the software, providing a designed network and many scenarios that can be analyzed by the engineers.

In spite of the good results achieved, the proposed algorithm demands further work to be improved in some interesting aspects. For instance, it still needs improvement to work with multi-objective problems and to consider the benefits of the dynamic programming, allowing saving processed solutions, which could be restored and exploited in new situations as the process matures.

Acknowledgment

The authors wish to thank COPEL (Companhia Paranaense de Energia) and CNPq (Brazilian National Council for Scientific and Technological Development).

References

Beasley, J. E. (2010). OR-LIBRARY: Distributing test problems, URL: http://www.people.brunel.ac.uk/~mastjjb/jeb/info.html, accessed in January/2010.

Bolognesi, A., Bragallia, C., Marchia, A., & Artinaa, S. (2010). Genetic heritage evolution by stochastic transmission in the optimal design of water distribution networks. *Advances in Engineering Software, 41*, 792–801.

Cadini, F., Zio, E., & Petrescu, C. A. (2010). Optimal expansion of an existing electrical power transmission network by multi-objective genetic algorithms. *Reliability Engineering and System Safety, 95*, 173–181.

Chen, A., Kim, J., Lee, S., & Kim, Y. (2010). Stochastic multi-objective models for network design problem. *Expert Systems with Applications, 37*(2), 1608–1619.

Chen, A., & Xu, X. (2012). Goal programming approach to solving network design problem with multiple objectives and demand uncertainty. *Expert Systems with Applications, 39*(4), 4160–4170.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik, 1*, 269–271.

Ding, S., & Ishii, N. (2000). An online genetic algorithm for dynamic Steiner tree problem. *IECON-Industrial Electronics Society, 2*, 812–817.

Esbensen, H. (1995). Computing near-optimal solutions to the Steiner problem in graph using a genetic algorithm. *Networks, 26*, 173–185.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *Annals of Mathematical Statistics, 11*, 86–89.

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley.

Hart, P. E., Nilsson, N., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions Systems Science and Cybernetics SSC, 4*(2), 100–107.

Huy, N. V., & Nghia, N. D. (2008). Solving graphical Steiner tree problem using parallel genetic algorithm. In: *IEEE international conference on research, innovation and vision for the future* (pp. 29–35).

Johnson, D. B. (1977). Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM, 24*(1), 1–13.

Johnson, D. S., Lenstra, J. K., & Rinnoov Kan, A. H. G. (1978). The complexity of the network design problem. *Networks* (Vol. 8, pp. 279–285). New Jersey: John Wiley & Son (4).

Kampstra, P., van der Mei, R. D., & Eiben, A. E. (2011). Evolutionary computing in telecommunication network design: A survey, 2006. Available in http://www.cs.vu.nl/~mei/articles/2006/kampstra/art.pdf, accessed in August/2011.

Kangal. (2011). Kanpur genetic algorithms laboratory. Available at http://www.iitk.ac.in/kangal/, accessed in April/2011.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In: *Proceedings of the international conference on neural networks* (pp. 1942–1948).

Li, Y., Wang, L., Xie, H., & Xie, Q. (2009). Distribution network optimal planning based on clouding adaptive ant colony algorithm. In *Power and Energy Engineering Conference-APPEEC* (pp. 1–4).

Marin, A., & Rodenas, R. G. (2009). Location of infrastructure in urban railway networks. *Computers & Operations Research, 36*, 1461–1477.

Martins, L. S., Pardalos, P., Resende, M. G., & Ribeiro, C. C. (1999). Greedy randomized adaptive search procedures for the Steiner problem in graphs. *DIMACS Series Discrete Mathematics and Theoretical Computer Science, 43*, 133–146.

Martins, S. L., Ribeiro, C. C., & Souza, M. C. (1998). A parallel GRASP for the Steiner problem in graphs. *Lecture notes in computer science* (1457, pp. 310–331). Springer-Verlag.

Mateus, G. R., Luna, H. P. L., & Sirihal, A. B. (2000). *Heuristics for distribution network design in telecommunication* (6). MA, USA: Kluwer Academic Publishers Hingham (pp. 131–148).

Papagianni, C., Papadopoulos, C. P. K, & Tselikas, N. D. (2008). Communication network design using particle swarm optimization. In *Proceedings of the international multi-conference on computer science and information technology* (pp. 915–920).

Ribeiro, C. C., & Souza, M. C. (2000). Tabu search for the Steiner problem in graphs. *Networks, 36*(2), 138–146.

Shi, Y. (2011). Source code of PSO. Available in http://www.engr.iupui.edu/~shi/pso.html, accessed in June/2011.

Silva, H., Britto, A. S., Jr., Oliveira, L. E. S., & Koerich, A. L. (2011). A multilevel algorithm for the network design problem. In *37th Annual conference on IEEE industrial electronics society* (pp. 2300–2304).

Xiao, H., Wang, X., & Du, W. (2008). A new discrete traffic network design problem with evolutionary game algorithm. *Intelligent Computation Technology and Automation (ICICTA), 1*, 3–7.

Xu, J., Chiu, S. Y., & Glover, F. (1995). Tabu search heuristics for designing a steiner tree based digital line network. Colorado University, Colorado, Waltham, Technical Report, p. 35.

Xu, T., Wei, H., & Wang, Z.-D. (2009). Study on continuous network design problem using simulated annealing and genetic algorithm. *Expert Systems with Applications, 36*(2), 2735–2741.

Zhan, Z.-H., Zhang, J., Li, Y., & Chung, H. S. (2009). Adaptative particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics, 39*(6), 1362–1381.

Zhang, G., Lu, J., & Xiang, Q. (2008). Application of genetic algorithm to network design problem. In *International conference on intelligent computation technology and automation* (Vol. 1, pp. 26–29).

Zhong, W.-L., Huang, J., & Zhang, J. (2008). A novel particle swarm optimization for the Steiner tree problem in graphs. In: IEEE-congress on evolutionary computation, pp. 2460–2467.