Les Cahiers du GERAD ISSN: 0711–2440

Approximation algorithms for the min-power symmetric connectivity problem

R. V. Plotnikov, A.I. Erzin,
N. Mladenović

G–2016–95

November 2016

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

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

Variable neighborhood search variants for min-power symmetric connectivity problem

R. V. Plotnikov [a]
A.I. Erzin [b]
Nenad Mladenović [c,d]

[a] *Sobolev Institute of Mathematics, Novosibirsk, Russia*

[b] *Sobolev Novosibirsk State University, Novosibirsk, Russia*

[c] *University of Valenciennes and Hainaut-Cambresis, Famars, France*

[d] *GERAD, Montréal (Québec), Canada*

nomad87@ngs.ru
adilerzin@math.nsc.ru
nenadmladenovic12@gmail.com

November 2016

Les Cahiers du GERAD
G–2016–95

Abstract: We consider the NP-hard problem of synthesis of optimal spanning communication subgraph in a given arbitrary simple edge-weighted graph. This problem occurs in the wireless networks while minimizing the total transmission power consumptions. We propose several new heuristics based on the variable neighborhood search metaheuristic for the approximation solution of the problem. We have performed a numerical experiment where all proposed algorithms have been executed on the randomly generated test samples. For these instances, on average, our algorithms outperform the previously known heuristics.

Acknowledgments: The research of R. Plotnikov is partly supported by the Russian Foundation for Basic Research (grant no. 16-37-60006). The research of A. Erzin is partly supported by the Russian Foundation for Basic Research (grant no. 16-07-00552). The research of N. Mladenović is partly supported by the Ministry of Education and Science, Republic of Kazakhstan (Institute of Information and Computer Technologies) (project no. 0115PK00546).

1 Introduction

The wireless communication networks related problems have been actively researched in recent years (see, e.g., [1, 2]). One of the most important issues is minimizing of the transmission energy consumption of the network elements per time unit. Since often the exact positions of the network elements and the topology of the network cannot be predefined, modern sensors have ability to adjust their transmission ranges in order to minimize the energy consumption without breaking the connectivity of the network. Herewith, usually the energy consumption of a network element is assumed to be proportional to d^s, where $s \geq 2$ and d is the transmission range [3]. But in the general case this condition may be not satisfied because of the inhomogeneity of the environment, radio interference and peculiar properties of the network elements (e.g., the signal may be spread unequally in all directions). Thus, the communication energy consumption for each connection could be arbitrary.

We assume that the communication network is represented as a graph $G = (V, E)$. In this paper we consider the symmetric case: an edge between two vertices means that the both of them can send a message to each other and the energy consumption for this communication is the same for both of them. If $c_{ij} \geq 0$ is a transmission-related energy consumption needed for sending data from $i \in V$ to $j \in V$ (as well as from j to i), then in the connected subgraph $T = (V, E')$, $E' \subseteq E$, the energy consumption of the node $i \in V$ equals to $E_i(T) = \max\limits_{j:(i,j)\in E'} c_{ij}$. The goal of this paper is development of the algorithms for the construction of a spanning subgraph T that minimizes the sum $\sum\limits_{i\in V} E_i(T)$. Without loss of generality, we assume that the subgraph T is a spanning tree.

In this paper we propose several new heuristic algorithms which use the variable neighborhood search (VNS) metaheuristic and different local search procedures. We compare the solutions obtained by these algorithms with the solutions obtained by other algorithms.

2 Problem formulation

Mathematically, the considered problem can be formulated as follows. Given a simple undirected edge-weighted graph $G = (V, E)$ with the vertex set V, $|V| = n$, and the edge set E, find such spanning tree T^* in G, which is the solution to the following problem:

$$W(T) = \sum_{i\in V} \max_{j\in V_i(T)} c_{ij} \to \min_T, \tag{1}$$

where $V_i(T)$ is the set of vertices adjacent to the vertex i in the tree T and $c_{ij} \geq 0$ is the weight of the edge $(i, j) \in E$.

In the literature, this problem is called the Min-Power Symmetric Connectivity Problem (MPSCP) [4]. Any feasible solution of (1), i.e., a spanning tree in G, will be called a *communication tree* (subgraph). It is known that (1) is strongly NP-hard [1, 5, 6, 7], and if P $\neq$ NP, then the problem is inapproximable within $1 + 1/260$ [6]. Therefore, the construction and analysis of efficient approximation algorithms are some of the most important issues regarding the research on this problem.

3 Related works

The more general Range Assignment Problem, where the goal is to find a strongly connected subgraph in a given oriented graph, has been considered in [7, 8]. MPSCP was first studied in [4], where the authors proved that Minimum Spanning Tree (MST) is 2-approximation for this problem. Also approximate polynomial algorithms with performance ratios of $1 + \ln 2 + \varepsilon \approx 1.69 + \varepsilon$ and $15/8$ were proposed in [4].

In [1] two local search heuristics were proposed. The first one is edge-switching (ES) and the second is edge and fork switching (EFS). In [9] ES-like heuristics ES1a and ES1b were proposed. It should be noticed that instead of finding a local optimum ES1a and ES1b perform a single loop on a fixed list of edges. Also, the authors proposed a faster sweep method (SW) and the most time-consuming double edge switching (ES2), which is said to be the generalization of EFS. Two another fast local search heuristics should be mentioned: subtree moving search (ST) [10] and local improvements (LI) [5]. They are very similar because they use the same idea: at each step an edge is removed from a tree and the root of obtained subtree is reconnected with some vertex in another subtree in such way that the decrease of the objective is maximum. The difference between ST and LI is the following: in ST the best replacement is performed at each step, but in LI all edges are sequently considered to be removed and replaced by another edge in the best way, and this loop is repeated while the solution is improved at least at one its iteration.

In [11] a way to filter the edges without impairment of the optimal solution was proposed. This method allows to reduce significantly the computation time. In [10] a new iterated local search (ILS) algorithm is presented. It uses ES and EFS during the local search phase, filtration technique from [11] and two different mutation operators. The numerical experiment results demonstrate that, on average, the best solution within acceptable time can be constructed by ILS with ES, filtration and so-called random increase mutation.

In [12] a hybrid genetic algorithm (GA), which uses variable neighborhood descent (VND) instead of mutation, was proposed. This algorithm is well parallelized and very fast.

4 Heuristics

As mentioned above, we use the VNS metaheuristic idea to get an approximate solution to (1). This metaheuristic consists of the local search and shaking phases (shaking phase consists of random movements, this procedure is briefly described below). We use two well-known schemes: basic VNS, wherein only one neighborhood structure is used in the local search phase, and general VNS, which uses a variable neighborhood descent (VND) approach in the local search phase. Detailed descriptions of both these methods can be found in [13]. Our variants of these procedures are described below. As a stopping criteria in VNS-based algorithms we used the following rule: if there were no any improvements during the last 3 iterations then algorithm stops.

In the shaking phase we perform the following procedure. Sequentially add at random one edge, and then remove another one from the appeared cycle in the best way. This is the variant of so-called intensified shaking. Such replacement is repeated k times. Let $k_{\max}$ be the maximum number of the edge replacements in the shaking procedure. Note that k_{max} is an independent parameter of all the considered VNS heuristics. The best value of this parameter is estimated experimentally.

For the local search phase we propose the following heuristics:

- *Adding and Best Removing (ABR).* In this algorithm all non-tree edges of the initial communication graph are considered sequently to be added in the tree, and after each adding the worst edge is removed from the obtained cycle. This procedure is repeated while the solution is improved.
- *Removing and Best Adding (RBA).* In this algorithm all edges in the tree are considered sequently to be removed, and after that the best edge, which connects the obtained two components, is added. This procedure is repeated while the solution is improved.

In order to reduce the computational complexity, we use the *filtration* of the edges presented in [11]. This filtration is applied to the communication graph as soon as the solution has been improved.

5 Simulation

All the proposed algorithms have been implemented in C++ using the Visual Studio 2010 Integrated Development Environment. A simulation was executed for $n = 10 - 250$, and in some cases for $n = 500$. For the same dimension, 100 different random instances were generated. For generation of an instance, a set of points were scattered on a square with uniform distribution, and the complete graph was used as initial communication graph. As an edge weight the squared distance between the corresponding points was taken. As the initial solution the best tree among the MST and the result of Prim-like Incremental Power Heuristic [14] was used. In order to compare the algorithms, we calculated the average improvement over the MST. It should be noticed, that because of the fact that often the initial solution was already better than MST, the results below do not represent the improvement over the *initial* solution. Of course, the contribution of Incremental Power Heuristic should be taken into account, and it was already estimated in [9]. The improvement over the MST was taken as the quality value, because this is the most commonly used characteristic [1, 9, 10] and its usage helps reader to compare our results with previous ones.

For the VNS-based heuristics, it is necessary to define the parameter $k_{\max}$. For this purpose, each algorithm was run on the same instances with different values of $k_{\max}$. It appeared that, beginning from $k_{\max} = 30$, on average, the objective of the obtained solution did not decrease significantly, whereas the runtime grown fast. Moreover, on average, the runtime of all algorithms remained accessible for $k_{\max} = 30$. Therefore, in all VNS-based algorithms, we set $k_{\max} = 30$.

Table 1 represents CPU time and improvement over MST using the basic VNS with different local search procedures. We have compared four basic VNS-based heuristics: B_ABR, B_RBA, B_ES and B_LI, where the local search procedures are respectively ABR, RBA, ES and LI. The general VNS-based heuristics results are presented in Table 2. We have run only two variants of general VNS. Both of them used ABR and RBA as local search procedures. G_AR is general VNS where in each iteration of the local search phase ABR was run at first and RBA was run next. G_RA is general VNS in each iteration of the local search phase RBA was run at first and ABR was run next.

Table 1: Basic VNS. Improvement over MST and CPU time.

n	B_ABR		B_RBA		B_ES		B_LI	
	Impr. to MST	CPU time	Impr. to MST	CPU time	Impr. to MST	CPU time	Impr. to MST	CPU time
10	3.94 %	0.00 s	3.98 %	0.00 s	3.98 %	0.00 s	3.77 %	0.00 s
30	5.71 %	0.05 s	5.77 %	0.05 s	5.76 %	0.06 s	4.14 %	0.00 s
50	6.16 %	0.23 s	6.26 %	0.21 s	6.27 %	0.21 s	3.82 %	0.00 s
100	6.02 %	1.44 s	6.24 %	1.44 s	6.12 %	1.21 s	3.42 %	0.01 s
250	6.01 %	15.41 s	6.27 %	21.46 s	5.96 %	12.09 s	3.45 %	0.02 s

Table 2: General VNS. Improvement over MST and CPU time.

n	G_AR		G_RA	
	Impr. to MST	CPU time	Impr. to MST	CPU time
10	3.98 %	0.00 s	3.96 %	0.00 s
30	5.74 %	0.07 s	5.78 %	0.08 s
50	6.15 %	0.28 s	6.29 %	0.29 s
100	6.03 %	1.74 s	6.20 %	1.86 s
250	6.05 %	21.27 s	6.30 %	31.14 s

We compared the solution obtained by the best VNS-based metaheuristics, basic VNS with RBA (B_RBA), with the solution obtained by ILS with ES (ILS_ES) from [10] in the same running time as B_RBA. These results are presented in Table 3. In the same manner, we have compared the G_RA (which appeared to be the best of general VNS-based heuristics) with ILS_ES, see Table 4. Except the improvement over MST, for each of two heuristics B_RBA and G_RA, we calculated the percentage of cases when its solution is better than ILS and the percentage of cases when it is worse than ILS_ES. One can see that, on average, B_RBA and G_RA both outperform ILS_ES, especially in large dimension cases. The advantages of the both VNS-based heuristics are most strongly shown when $n = 500$. In this case B_RBA yield more accurate solution than ILS_ES in 99 percent of cases, the average improvement of B_RBA over MST exceeds the same estimation of LI_ES by 0.44% which is about 7.5% of the improvement, and the maximum improvement over to MST exceeds the same estimation of LI_ES by 0.82% which is 10.16% of the improvement. The results obtained by G_RA in the case of $n = 500$ are very impressive as well: G_RA yields better solution than ILS_ES in 94 percent of cases, its average excess of improvement over MST is 0.38%, which is 6.37% of the improvement. It should be noted, that LI_ES had appeared to be too time-consuming in a case of $n = 500$. Its average running time on 10 instances exceeded 1200 seconds.

Table 3: Comparison of the results for the best of the basic VNS-based heuristics B_RBA and for the best of the iterated local search-based algorithms ILS_ES.

n	B_RBA is better	ILS_ES is better	B_RBA: Impr. to MST Min	Avg	Max	ILS_ES: Impr. to MST Min	Avg	Max	CPU time
10	2 %	0 %	0 %	3.98 %	19.82 %	0 %	3.95 %	19.82 %	0.00 s
30	11 %	2 %	0.82 %	5.78 %	15 %	0.82 %	5.7 %	14.58 %	0.05 s
50	30 %	10 %	1.20 %	6.28 %	13.56 %	1.20 %	6.2 %	13.41 %	0.20 s
100	54 %	26 %	2.48 %	6.23 %	10.42 %	2.38 %	6.15 %	10.86 %	1.43 s
250	85 %	15 %	3.64 %	6.29 %	9.50 %	3.46 %	6.04 %	9.36 %	22.63 s
500	99 %	1 %	4.10 %	6.34 %	8.89 %	3.72 %	5.90 %	8.07 %	208.2 s

Table 4: Comparison of the results for the best of the general VNS-based heuristics G_RA and for the best of the iterated local search-based algorithms ILS_ES.

n	G_RA is better	ILS_ES is better	G_RA: Impr. to MST Min	Avg	Max	ILS_ES: Impr. to MST Min	Avg	Max	CPU time
10	0 %	0 %	0 %	3.98 %	19.82 %	0 %	3.98 %	19.82 %	0.01 s
30	6 %	6 %	0.82 %	5.77 %	15 %	0.82 %	5.74 %	14.58 %	0.08 s
50	17 %	14 %	1.2 %	6.29 %	13.66 %	1.2 %	6.29 %	13.56 %	0.30 s
100	47 %	36 %	1.87 %	6.21 %	10.34 %	2.42 %	6.23 %	10.86 %	2.11 s
250	72 %	28 %	3.69 %	6.29 %	9.49 %	3.65 %	6.16 %	9.60 %	31.42 s
500	94 %	6 %	4.30 %	6.35 %	8.6 %	3.75 %	5.97 %	8.29 %	279.7 s

In [12] we proposed two hybrid genetic algorithms for the MPSCP. The best results had been obtained by the hybrid genetic algorithm which used VND-based heuristic instead of mutation (GA_VND). In Table 5 the results of this algorithm are compared with the best VNS-based heuristics: B_RBA and G_RA. One can see that GA_VND solve the problem significantly faster, but it should be taken into account that it was well parallelized and four parallel threads were used. However, VNS-based heuristics yield more accurate solutions, especially on large instances: in a case of $n = 500$ their average improvement over MST exceed the same estimation for the GA_VND by more than 0.6%, which is 10.5% of the improvement.

Table 5: Comparison of the results for the best of VNS-based heuristics and hybrid genetic algorithm GA_VND.

n	B_RBA Impr. to MST	B_RBA CPU time	G_RA Impr. to MST	G_RA CPU time	GA_VND Impr. to MST	GA_VND CPU time
10	3.98 %	0.00 s	3.98 %	0.01 s	3.98 %	0.06 s
30	5.78 %	0.05 s	5.77 %	0.08 s	5.75 %	0.14 s
50	6.28 %	0.20 s	6.29 %	0.3 s	6.20 %	0.31 s
100	6.23 %	1.427 s	6.21 %	2.11 s	5.96 %	1.12 s
250	6.29 %	22.63 s	6.29 %	31.42 s	5.87 %	6.35 s
500	6.34 %	208.2 s	6.35 %	279.7 s	5.71 %	31.8 s

6 Conclusion

In this paper we have proposed new variable neighborhood search-based heuristics for the Min-Power Symmetric Connectivity Problem. We used two known variants of the VNS metaheuristic: basic VNS and general VNS. As local search we used two new heuristics, ARB and RBA, as well as already known heuristics ES and LI. We also used filtration of the edges of the communication graph in our algorithms in order to reduce the computation time. The numerical experiment has shown that, on average, the best of the proposed VNS-based heuristics constructs significantly more accurate solutions during short time than the best of the known heuristics: iterated local search variant [10] and hybrid genetic algorithm [12]. In future we plan to implement the decomposition metaheuristics for this problem in order to solve the larger instances in acceptable time.

References

[1] E. Althaus, G. Călinescu, I. I. Măndoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky, Wireless Networks 12(3), 287–299 (2006).

[2] P. Santi, ACM Comput. Surv. 37(2), 164–194 (2005).

[3] T. Rappaport, Wireless Communications: Principles and Practices (Prentice Hall, 1996).

[4] G. Călinescu, I. I. Măndoiu, and A. Zelikovsky, "Foundations of information technology in the era of network and mobile computing: Ifip 17th world computer congress — tc1 stream / 2nd ifip international conference on theoretical computer science (tcs 2002) august 25–30, 2002, montréal, québec, canada," (Springer US, Boston, MA, 2002) Chap. Symmetric Connectivity with Minimum Power Consumption in Radio Networks, 119–130.

[5] A. Erzin, R. Plotnikov, and Y. Shamardin, Journal of Applied and Industrial Mathematics 7, 142–152 (2013).

[6] B. Fuchs, On the hardness of range assignment problems, Tech. Rep. TR05-113 (2005).

[7] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, Theoretical Computer Science 243(1–2), 289–305 (2000).

[8] A. E. F. Clementi, P. Penna, and R. Silvestri, "Hardness results for the power range assignmet problem in packet radio networks," in Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization Problems: Randomization, Approximation, and Combinatorial Algorithms and Techniques, RANDOM-APPROX '99 (Springer-Verlag, London, UK, UK, 1999), 197–208.

[9] J. Park and S. Sahni, "Power assignment for symmetric communication in wireless networks," in Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC), 620 (IEEE Computer Society, Los Alamitos, 2006), 591–596.

[10] S. Wolf and P. Merz, "Evolutionary computation in combinatorial optimization: 9th european conference, evocop 2009, tübingen, germany, april 15–17, 2009. proceedings," (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) Chap. Iterated Local Search for Minimum Power Symmetric Connectivity in Wireless Networks, 192–203.

[11] R. Montemanni and L. Gambardella, Computers & Operations Research 32(11), 2891–2904 (2005).

[12] A. Erzin and R. Plotnikov, Electronic Notes in Discrete Mathematics 47, 21–28 (2015).

[13] P. Hansen and N. Mladenovic, European Journal of Operational Research 130, 449–467 (2001).

[14] X. Cheng, B. Narahari, R. Simha, M. X. Cheng, and D. Liu, IEEE Transactions on Mobile Computing 2(3), 248–256 (2003).