A Comparison of Heuristic Methods for the Prize-Collecting Steiner Tree Problem and Their Application in Genomics

Murodzhon Akhmedov, Ivo Kwee and Roberto Montemanni

Abstract The prize-collecting Steiner tree (PCST) problem is a broadly studied problem in combinatorial optimization. It has been used to model several real world problems related to utility networks. More recently, researchers have started using PCSTs to study biological networks. Biological networks are typically very large in size. This can create a considerable challenge for the available PCST solving methods. Taking this fact into account, we have developed methods for the PCST that efficiently scale up to large biological network instances. Namely, we have devised a heuristic method based on the Minimum Spanning Tree and a matheuristic method composed of a heuristic clustering phase and a solution phase. In this work, we provide a performance comparison for these methods by testing them on large gene interaction networks. Experimental results are reported for the methods, including running times and objective values of the solutions.

1 Introduction

The prize-collecting Steiner tree is a well known problem in combinatorial optimization and graph theory. Within the concept of the PCST, given an undirected network $G = (V, E)$, where nodes are associated with prizes $p_j \geq 0$ and arcs are associated with costs $c_e > 0$, the goal is to construct a sub-graph $G' = (V', E')$ that has a *tree* structure. The researchers have studied different variants of the PCST problem in the literature. One of the broadly studied variant is known as *Goemans–Williamson*

M. Akhmedov (✉) · I. Kwee · R. Montemanni
Dalle Molle Institute for Artificial Intelligence (IDSIA-USI/SUPSI),
Galleria 2, 6928 Manno, Switzerland
e-mail: murodzhon@idsia.ch

I. Kwee
e-mail: ivo.kwee@ior.iosi.ch

R. Montemanni
e-mail: roberto@idsia.ch

M. Akhmedov · I. Kwee
Institute of Oncology Research (IOR), Via Vela 6, 6500 Bellinzona, Switzerland

K.F. Dœrner et al. (eds.), *Operations Research Proceedings 2015*,
Operations Research Proceedings, DOI 10.1007/978-3-319-42902-1_14

Minimization [1], where the objective is to identify a *tree* for a given graph by minimizing the total cost of arcs in a *tree* and minimizing the total prize of nodes excluded from the *tree*. This corresponds to the minimization of the following expression:

$$GW(G') = \sum_{e \in E'} c_e + \sum_{v \notin V'} p_v \tag{1}$$

The PCST has been successfully applied to model several real-world problems in utility networks. Recently, researchers have realized its application to biological networks for discovering the hidden knowledge [2]. Based on this idea, we have applied the PCST to gene interaction networks, where nodes correspond to genes and arcs represent the mutual information between genes. The PCST potentially captures the portion of graphs where genetic aberrations and mutations are highly present. Basically, biological interaction networks are large in size, and this can be remarkable challenge for existing PCST methods. By considering this fact in our previous studies, we have developed methods for the PCST that efficiently scale up to large biological network instances for analyzing the function of genes. In this work, we extensively test previously developed methods on generated gene interaction networks, and compare their performance on large networks.

2 Related Work

The pioneering work was performed by [3] in the PCST literature. The *node weighted Steiner tree problem* was proposed in [4], in which the specific set of nodes have to be covered by output tee. The state-of-the-art exact methods were presented in [5, 6], where the PCST was formulated by means of mixed integer linear programming (MILP) and a branch-and-cut algorithms was employed to solve underlining MILP. Some heuristic and matheuristic algorithms were studied in [1, 7, 8].

There some studies in the literature [2, 9–11] that already applied the PCST for functional analyses of protein interaction networks. As a result of these studies, the authors identified unknown functions of some proteins. They validated their computational findings by biological experiments. This shows the potential of the PCST to generate promising results while analyzing interaction networks.

3 Methodology

Usually, biological interaction networks are complex and huge in size. The PCST belongs to the class of NP-hard problems, where it is time consuming to obtain solutions for large graphs. This was the primary limiting factor for available PCST methods being applied on gene interaction networks. To enable the application of the PCST on biological networks, we have developed a heuristic and a matheuristic solution methods in our previous studies. The methods are shortly outlined in the following subsection.

3.1 The MST-Based Heuristic

This heuristic method is based on the iterative solution of Minimum Spanning Tree (MST) problems. Given an undirected network $G = (V, E)$ and a user-defined parameter α, the heuristic constructs a complete network $G_1 = (V_1, E_1)$ within the first iteration, where $V_1 : v$ only composed of nodes with $p_v > \alpha$ and $E_1 : (i, j)$ corresponds to the shortest path distance between nodes i and j. The algorithm starts solving the problem by considering the nodes with prize $p(v) \geq \alpha$ at the first iteration. Afterwards, the algorithm solves a MST on G_1 and obtains a tree $T_1 = (V_1^{'}, E_1^{'})$ with the cost of $C1$. In the second iteration, the heuristic constructs next complete network $G_2 = (V_2, E_2)$, where $V_2 : v$ formed by all nodes of tree from previous iteration $v \in V_1^{'}$, and $E_2 : (i, j)$ corresponds to the shortest path distance between nodes i and j. Again, the algorithm computes a MST on G_2 and obtains a tree $T_2 = (V_2^{'}, E_2^{'})$ with the cost of $C2$. If $C2 \geq C1$, the algorithm terminates. Otherwise, the heuristic continues generating complete graph and solving MST problems until the cost of current tree gets bigger or equal to the cost of the previous tree. Then, the algorithm prunes the leaf nodes of the tree in order to further decrease the cost, and obtains final solution. The interested reader may refer to [12] for further details of the heuristic method.

3.2 The Clustering Matheuristic

The matheuristic algorithm was devised by combination of a heuristic clustering algorithm and an exact PCST solver. The main idea of the matheuristic was to divide the large graph into smaller graph clusters, and to solve each cluster separately using exact solver. The heuristic clustering algorithm clusters the nodes according to the all-pairs shortest path distance. Then, smaller graphs are constructed by inducing the nodes in the same cluster. Every smaller graph is solved by using exact PCST solver. Important to note that any exact solver could be used as inner solver at this stage. We have adapted the method proposed by [5] to our approach, and used it as an exact solver due to its efficiency. In [5], the PCST was formulated by MILP, and a branch-and-cut algorithm was proposed to solve the formulation. The interested reader may refer to [13] for further explanation of the matheuristic method.

4 Experimental Results

In this section, we test the MST-based heuristic and the clustering matheuristic method on large gene interaction network instances, and compare their performance. The benchmark instances are generated based on gene expression profiling data of Diffuse Large B-Cell Lymphoma (DLBCL) cancer patients available online in Gene Expression Omnibus repository.[1] There two subtypes of DLBCL cancer that are: the germinal center B cell (GCB) and an activated B cell (ABC). The goal is to identify a set of genes that are relevant for subtype classification. The networks are generated by using the multiplicative model of ARACNE [14] algorithm, which is a powerful tool for the reconstruction of gene interaction networks. ARACNE uses the mutual information among genes for the network reconstruction. We used two parameters ($eps = 0.01$, $eps = 0.05$) fed into ARACNE in order to generate the test instances. In these networks, every arc represents the interaction between two genes and its weight is labeled as the pairwise correlation of expression values of genes. Each node is labeled with prize $p_v = |E_{ABC} - E_{GCB}|$, where E_{ABC} and E_{GCB} are the mean value of gene expression of ABC and GCB cancer patients for corresponding gene, respectively. All of the nodes have positive prize $p_v > 0$ in generated instances.

The computational experiments have been performed on a machine equipped with an Intel(R) Xeon(R) CPU E5320 1.86 GHz processors and 32 GB of shared memory. A single core was used for the experiments.

Table 1 summarizes the results of both methods for gene interaction network instances generated with the parameter $eps = 0.01$. The first three columns of the table show the names and the sizes of test instances, respectively. From the fourth to the ninth columns we report the objective values and running times of the MST-based heuristic method [12], in which the algorithm employs different values for the parameter α. The tenth and eleventh columns present the objective values and execution times of the clustering matheuristic method [13].

Table 2 delivers the results of the MST heuristic and clustering matheuristic methods for interaction network instances generated with the parameter $eps = 0.05$.

According to the results of the tables, both methods, the MST heuristic and clustering matheuristic, were able to provide solutions in a reasonable time. The solutions obtained by the clustering matheuristic are considerably better than the MST heuristic in terms of solution cost for these instances. The MST heuristic also was able to obtain good quality solutions, and the running times of the instances are improved by decreasing the parameter α. The primary reason for elaborated execution times is the decay in parameter α, where the larger set of nodes are considered in computations during the first iteration. The general pattern of the solution cost is decreased by lowering the α from 0.5 to 0.3 and 0.1, however, lowering the α to 0.0 did not improve the cost further. The main reason for this is the MST heuristic was designed for large networks that have smaller number of positive nodes $p_v > 0$. In contrast, the clustering matheuristic method was developed for large networks where most of the nodes have positive prizes $p_v > 0$. The parameter α can be used to tune a trade

[1] http://www.ncbi.nlm.nih.gov/geo/.

Table 1 Results of DLBCL test instances generated with the parameter *eps* = *0.01*

			MST $\alpha = 0.5$		MST $\alpha = 0.3$		MST $\alpha = 0.1$		MST $\alpha = 0.0$		Clust. MATH	
Instance	V	E	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)
GSE4732	2407	32503	192.3	1	190.3	4	190.5	32	190.5	143	189.6	120
GSE4475	13211	246942	708.9	24	698.1	96	693.6	855	699.2	6048	692.0	6433
GSE22470	13211	266085	861.1	26	850.2	119	846.3	1158	851.8	6419	842.3	7419
GSE10172	13211	231444	346.3	2	345.7	3	339.0	181	339.8	5861	338.6	5859
GSE19246	21049	82282	1631.8	22	1625.1	102	1622.9	1550	1623.9	6540	1616.9	4686
GSE10846	21049	329576	1556.8	76	1544.7	319	1544.7	1468	1541.7	13908	1538.4	14269
GSE23501	21049	939891	1410.0	27	1406.2	273	1404.8	6593	1405.6	26866	1402.6	33446
GSE31312	21049	1055055	1665.8	212	1662.9	1035	1668.6	8736	1673.9	34988	1662.0	37280

Table 2 Results of DLBCL test instances generated with the parameter *eps* = *0.05*

			MST $\alpha = 0.5$		MST $\alpha = 0.3$		MST $\alpha = 0.1$		MST $\alpha = 0.0$		Clust. MATH	
Instance	V	E	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)	OBJ	t (s)
GSE4732	2407	40921	192	1	189.4	5	189.8	40	189.8	173	188.8	155
GSE4475	13211	347507	708.3	33	697.0	130	692.5	1134	697.0	8026	690.7	9389
GSE22470	13211	375097	652.5	14	648.3	78	646.7	980	652.9	8570	645.1	9178
GSE10172	13211	325423	364.5	3	363.5	5	354.3	278	355.3	7617	352.6	8336
GSE19246	21049	154469	987.4	4	986.6	34	986.8	643	987.3	8683	985.5	7144
GSE10846	21049	475173	1179.0	38	1173.5	217	1170.7	2341	1170.9	18115	1169.9	19060
GSE23501	21049	1050916	1409.9	30	1405.9	304	1404.6	7250	1405.4	31912	1402.2	36645
GSE31312	21049	1423267	1983.2	510	1979.2	2079	1984.1	14662	1990.4	45707	1977.7	52847

off between the quality and running time for the MST heuristic. The α can be set to a reasonably higher value in order to analyze large interaction networks fast, and also not losing too much from the optimality.

5 Conclusions

In this study, we have compared a MST-based heuristic and a clustering matheuristic methods developed for large prize-collecting Steiner tree problems generated from real biological data describing gene interaction networks. Experimental results support that the performance of the clustering matheuristic is better than the MST heuristic method in terms of solution quality for the interaction network instances, however, MST heuristic also can be used to analyze large interaction networks in a quick manner by tuning the α parameter.

Acknowledgements M. Akhmedov is supported by Swiss National Science Foundation through project 205321-147138/1: “Steiner Trees for Functional Analysis in Cancer System Biology”.

References

1. Johnson, D.S., Minkoff, M., Phillips, S.: The prize collecting Steiner tree problem: theory and practice. In: Proceedings of 11th ACM–SIAM Symposium on Discrete Algorithms, pp. 760–769 (2000)
2. Bechet, M.B., Borgs, C., Braunsteinc, A., Chayes, J., Dagkessamanskaia, A., François, J.M., Zecchina, R.: Finding undetected protein associations in cell signalling by belief propagation. PNAS **108**, 882–887 (2010)
3. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.: A note on the prize collecting traveling salesman problem. Math. Progr. **59**, 413–420 (1993)
4. Segev, A.: The node-weighted Steiner tree problem. Networks **17**, 1–17 (1987)
5. Ljubic, I., Weiskircher, R., Pferschy, U., Klau, G.W., Mutzel, P., Fischetti, M.: An algorithmic framework for the exact solution of the prize−collecting Steiner tree problem. Math. Progr. **105**(2), 427–449 (2006)
6. Ljubic, I., Weiskircher, R., Pferschy, U., Klau, G., Mutzel, P., Fischetti, M.: Solving the prize−collecting Steiner tree problem to optimality. Proceedings of ALENEX, Seventh Workshop on Algorithm Engineering and Experiments, pp. 68–76 (2005)
7. Canuto, S.A., Resende, M.G.C., Ribeiro, C.C.: Local search with perturbation for the prize-collecting Steiner tree problem in graphs. Networks **38**, 50–58 (2001)
8. Klau, G.W., Ljubic, I., Moser, A., Mutzel, P., Neuner, P., Pferschy, U., Raidl, G., Weiskircher, R.: Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem. Genet. Evol. Comput. GECCO **2004**(3102), 1304–1315 (2004)
9. Tuncbag, N., McCallum, S., Huang, S.C., Fraenkel, E.: SteinerNet: a web server for integrating omic data to discover hidden components of response pathways. Nucl. Acids Res. 1–5 (2012)
10. Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Mueller, T.: Identifying functional modules in protein-protein interaction networks: an integrated exact approach. Bioinformatics **26**, 223–231 (2008)
11. Beisser, D., Klau, G.W., Dandekar, T., Mueller, T.: T, and M. Dittrich. BioNet: an R-package for the functional analysis of biological networks. Bioinformatics **26**(8), 1129–1130 (2010)

12. Akhmedov, M., Kwee, I., Montemanni, R.: A fast heuristic for the prize-collecting steiner tree problem. Lect. Notes Manag. Sci. **6**, 207–216 (2014)
13. Akhmedov, M., Kwee, I., Montemanni, R.: A divide and conquer matheuristic algorithm for the prize-collecting Steiner tree problem. Computers and Operation Research (to appear)
14. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., Califano, A.: Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinform. **7**(Suppl 1), S7 (2006)