A New Hybrid Genetic Algorithm For The District Heating Network Problem

Mitică CRAUS[1], Florin LEON[1] Dragos AROTĂRIȚEI[2]
[1]*Gheorghe Asachi Technical University*
Bd. Dimitrie Mangeron nr. 53A, 700050 Iaşi, Romania
[2]*Gr. T. Popa University of Medicine and Pharmacy*
Str.Universitatii nr. 16, 700115 Iasi, Romania
[1]*craus@cs.tuiasi.ro,* [1]*fleon@cs.tuiasi.ro,* [2]*dragos_aro@yahoo.com*

***Abstract* — In this paper we propose a new hybrid genetic algorithm with an adaptive fitness function to solve the problem of extending district heating networks, related to the prize-collecting Steiner tree problem. The multi-criterial optimization is achieved by means of weighting the profit and costs, depending on the application. The experimental results showed that the algorithm works well for small and sparse graphs and the running time is reasonable.**

***Index Terms* — Water heating systems, Genetic algorithms, Optimization methods, Trees (graphs), Prize-collecting Steiner tree**

I. INTRODUCTION

The district heating networks (DHN) as part of the district heating systems (DHS) are very important in the perspective of finding solutions for the combustible consummation reduction and building ecological and safe residential areas. Making new networks is very expensive. The subject of this paper is the problem of DHN that evolves in order to be extended in a new area of consumers, using the DHN in use and the existing plants, or a new plant in a new district.

The main step of the process of extending such networks leads to the prize-collecting Steiner tree problem (PCST), a modified variant of the Steiner tree problem. There are some solutions for PCST, but very few based on genetic algorithms.

A primal-dual method is proposed in [2]. The algorithm is based on Goemans and Williamson algorithm and has the time-complexity of $O(n^2 \log n)$. In [3] the authors present and compare three heuristics for a variant of the Steiner tree problem with revenues that is a Steiner tree with budget and hop constraints. In the first step the authors use a greedy method which obtains good approximations in short computational times. This solution is then improved by means of a destroy-and-repair method or a tabu-search algorithm. Some branch-and-cut algorithms based on a direct graph model are proposed in [7] and [8]. The algorithms give good results for a set of significant real-world instances, but the algorithm implementation is laborious.

The evolutionary algorithms proved to be a good option for single objective or multi-objective optimization problems. However, PCST has extremely few approaches by using genetic algorithms (GAs) or other evolutionary algorithms (EAs). One of the most important phases in designing the encoder-decoder step for an evolutionary algorithm is its genotypic representation. The difficulties of efficient coding-decoding schema for PCSTs explain the few proposals which use GA for Steiner trees and in special PCST problem. A distance network heuristic is proposed in [4]. It is based on the distance network heuristic designed by Kou [6]. Prufer encoding [5] is used by some researchers as well. However, some criticism arises from other authors, which assert that some good results reported sometimes are due to sparse graphs and small graphs. We can conclude that the characteristics of the graph in the PCST problem have an important influence in choosing the encoding-decoding schema for GA used in solving minimal PCST.

The paper is organized as follows. In the second section the PCST problem and our solution are presented. The third section is reserved to the description of the genetic algorithm. In the next section some experiments are explained and analyzed. The last section contains conclusions and directions for future work.

II. PRIZE-COLLECTING STEINER TREE PROBLEM (PCST) AND OUR APPROACH

The Steiner minimal tree is an important optimization problem which consists in finding a tree that connects a given set of points with minimum costs.

Given a connected and undirected graph $G=(V, E)$, a cost function c: $E \rightarrow R^+$ and a subset $S \subseteq V$, the Steiner tree problem (STP) is defined as follows:

Find a tree $T = (V_T, E_T)$ *where* $S \subseteq V_T \subseteq V$, $E_T \subseteq E$ *and c(T) is minimal.*

A typical DHS consists in one or more production units, a pipe network and possible heat-exchanger stations. In our approach we will consider the case with one production unit and a heating network with n customers [1].

When a new district has to be integrated in a DHN, keeping the existent production unit, a number of consumers (terminals in the network) will be connected with minimum costs and maximum profit. At limit, all the consumers will be connected. The problem is Prize-Collecting Steiner Tree (PCST) type and the solution is a rooted tree.

The common way to solve the problem is to select a subset of profitable customers from the set of all customers and then to proceed to maximize the profit. The DHN is extended using this subset as terminal nodes and the cost of network is evaluated. Many constraints have to be satisfied. The edges have to follow the streets and the nodes must represent the street intersections and potential customers.

One node (the root) represents the heating producer.

Let $G = (V, E, c, p)$ be an undirected graph, where vertices are associated with profits defined by a profit function $p: V \rightarrow R^+$ and the edges have costs given by the function $c: E \rightarrow R^+$.

There are two types of nodes: customer nodes and non-customer nodes. A street node (non-customer node) has a zero profit. The customer and the non-customer node sets are denoted by R and Q, respectively:

$$R = \{v \in V \mid p(v) \neq 0\}, R \neq \phi \quad (1)$$

$$Q = \{v \in V \mid p(v) = 0\} \quad (2)$$

A subset S of customer nodes has to be connected by a tree T. The problem is to maximize the difference between profits and costs, without overflowing the budget:

$$profit(T) = \sum_{v \in V_T} p(v) - \sum_{e \in E_T} c(e) \quad (3)$$

$$subject\ to\ \ C = \sum_{e \in E_T} c(e) \leq B(buget) \quad (4)$$

In order to be sure that the root is included in the solution, we allocate a high value of the profit for this node.

For the proposed solution based on genetic algorithms, the encoding is binary and the chromosomes have a length equal to all the nodes of the problem, both with and without profit. A gene with a "1" value indicates that the node was part of the solution, while a "0" value indicates that the node is *temporarily* excluded. Since the GA performs a search without any domain knowledge, a solution will most likely be an invalid network graph. Therefore, the partial solution obtained must be completed to form a connected graph that can represent an acceptable solution. The graphs are internally represented by their adjacency matrices.

In order to find the intermediary nodes between the profit nodes selected by the GA, we use the Dijkstra algorithm to find both the path between any two disconnected profit nodes, and the minimum cost between them. In this way, a new graph is constructed. However, we chose to allow the GA to operate on disconnected graphs as well, and not only on connected graphs, because this post-processing step would usually introduce many "1" genes, which in turn would significantly lower the genetic diversity of the population, leading to premature convergence and decreasing the chances of finding the optimal solution.

The next step is to compute the minimum spanning tree, using Prim's algorithm, starting with the first node that represents the root. The tree is then evaluated, and the sum of profits P is computed only for the profit nodes, as well as the sum of costs of the network edges C, including those connecting the non-profit nodes.

The problem is a multi-criterial one. We aim simultaneously at minimizing the cost and maximizing the profit, but these objectives are contradictory. The simplest approach is to give weights to these two criteria. Therefore, the fitness function F is:

$$F = P \cdot w_P - C \cdot w_C, \quad (5)$$

where w_P and w_C are the corresponding weights of the total profit and the total cost, respectively.

We also included the possibility of establishing a maximum allowed cost. In this case, many solutions found by the GA will be rejected. If we assigned a constant negative fitness to all these invalid solutions, again the genetic diversity would be greatly reduced. It is very likely that the GA could not find a solution at all.

In order to solve this problem, we chose an adaptive fitness function. If the solution does not meet the maximum cost limitation, the fitness will be:

$$F = -C \quad (6)$$

Therefore, at the beginning of the search process, if no valid solution is found, the GA tries to decrease the cost, with no regard to the profit. After a few generations, when the cost has decreased enough to respect the imposed criterion, the fitness function changes to the expression in equation 1, and the GA starts to maximize the weighted profit. In this way, no solutions are excluded a-priori, and this helps decrease the number of individuals in the population, which is an important factor regarding the speed of the algorithm.

III. HYBRID GENETIC ALGORITHM FOR THE PRIZE-COLLECTING STEINER TREE PROBLEM

The genetic algorithm has the classical structure given by the pseudo-code from Fig. 1.a. Given the GA solution (the started graph), the pseudo-code of the specialized algorithm is presented in Fig. 1.b. :

```
create initial population
while not maximum number of generations
     eached
       evaluate fitness of individuals
 save best individual into new generation
 while not new generation complete
     select 2 parents by tournament
     generate child by crossover
     change child by mutation
     insert child into new generation
 end while
 current generation = new generation
end while
solution = best individual
```

Figure 1.a.

```
while exist 2 disconnected profit nodes in
graph
     apply Dijkstra algorithm to find
     shortest path between these nodes in
initial graph
     insert links into the graph
end while
apply Prim's algorithm on graph starting
with node 1
solution = minimum spanning tree
```

Figure 1.b.

Figure 1. Genetic Algorithm for the Prize-Collecting Steiner Tree Problem.

In order to show how the algorithm works, let us consider the graph in Fig 2. After applying the genetic algorithm, the selected nodes are those presented in Fig 3.

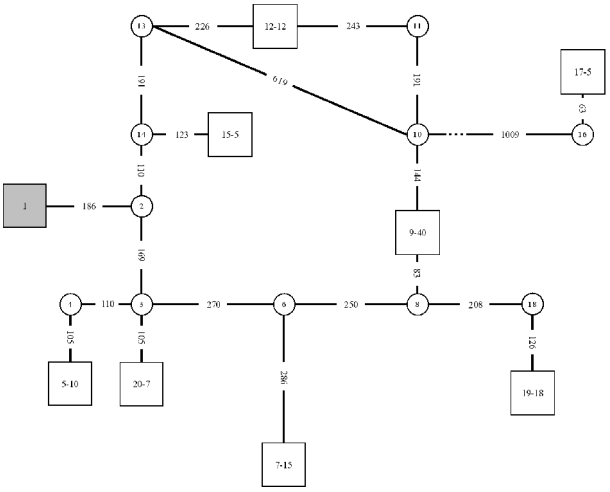

Figure 2. Simple test network.

This in only a preliminary solution, since the GA does not guarantee that the profit nodes are connected. If we impose a positive weight for the cost, the GA will eliminate any unnecessary links, or unprofitable nodes. It will only select the profit nodes that give the best outcome. The first node, the root, is included by default. Using the algorithm previously presented, the selected profit nodes are connected using the shortest path in the initial graph of the problem.

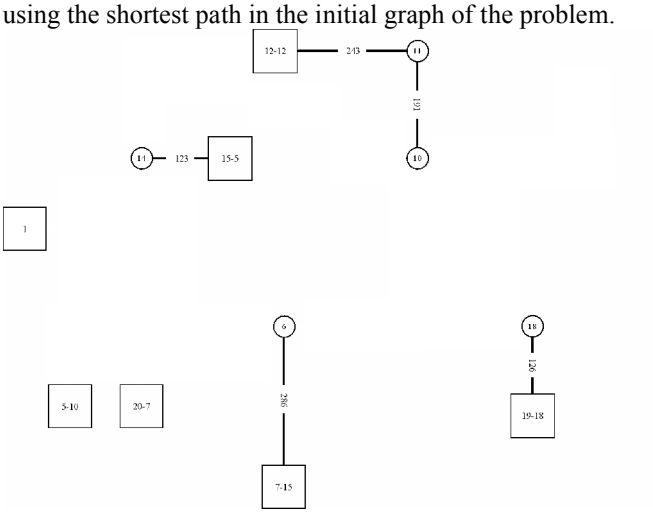

Figure 3. Preliminary GA solution.

The final phase of the approach is to apply Prim's algorithm to find the minimum spanning tree on the resulting connected graph, and this represents the solution. For the network considered in this example, the solution is displayed in Fig 4.

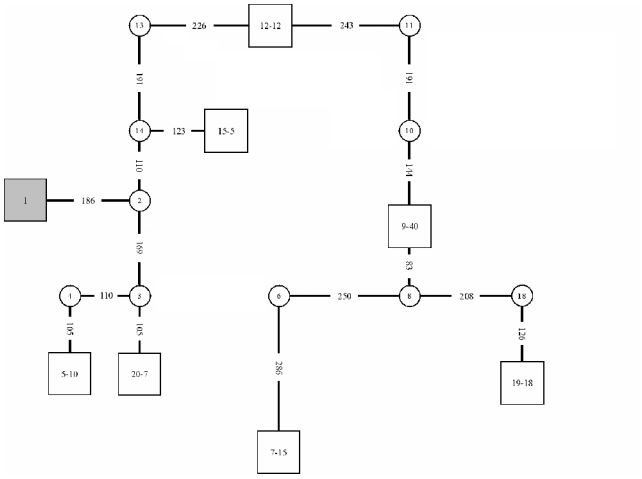

Figure 4. Final network tree.

IV. EXPERIMENTAL RESULTS

We applied our algorithm to a simple network in order to analyze the behavior of the algorithm, and to three benchmark problems, to see how it performs in complex cases. The results are presented in what follows.

A. SIMPLE NETWORK

In order to test the performance of the algorithm, we start with a simple test network, displayed in Fig 4. For the genetic algorithm, we used a configuration of 50 individuals, a stopping criterion of maximum 50 generations, a tournament selection with 3 individuals, a crossover rate of 0.9, and a mutation rate of 0.02. In order not to lose the best solution from a generation to another, we also used elitism, i.e. the best solution in a generation was directly copied into the next one.

For the first test, we only considered the profit of the network: $w_P = 100$, $w_C = 0$. We did not impose any cost limitation either. The results of the algorithm are displayed in Fig 5. The vertices with profit are marked in the form *vertex_index*:*profit*. The same notation is used to present the cost of a network edge.

```
Best fitness: 11200
Best chromosome: 00001011000100101011
Selected vertices: 1 2 3 4 5:10 6 7:15 8
9:40 10 11 12:12 13 14 15:5 16 17:5 18
19:18 20:7
Profit: 112
Network edges: 1-2:186  2-3:169  2-14:110
3-4:110  3-20:105  4-5:105  6-7:286  6-
8:250  8-9:83
8-18:208  9-10:111  10-11:191  10-16:1009
11-12:243  12-13:226  13-14:191  14-15:123
16-17:63
18-19:126
Cost: 3895
```

Figure 5. The results of the algorithm for $w_P = 100$, $w_C = 0$.

Next, we also took into account the cost, using $w_P = 100$ and $w_C = 1$. The results are presented in Fig. 6. One can notice a decrease in the total cost. The profit node 17 was not included in the solution because it is very distant to the rest of the nodes in the network.

```
Best fitness: 7877
Best chromosome: 00001110011101100111
Selected vertices: 1 2 3 4 5:10 6 7:15 8
9:40 10 11 12:12 13 14 15:5 18 19:18 20:7
Profit: 107
Network edges: 1-2:186  2-3:169  2-14:110
3-4:110  3-20:105  4-5:105  6-7:286 6-
8:250  8-9:83
8-18:208  9-10:111  10-11:191  11-12:243
12-13:226  13-14:191  14-15:123  18-
19:126
Cost: 2823
```

Figure 6. The results of the algorithm for $w_P = 100$, $w_C = 1$.

Finally, we imposed a cost limit of 500. In this case, only a few nodes have been found (Fig. 7).

```
Best fitness: 240
Best chromosome: 10000000000000000001
Selected vertices: 1 2 3 20:7
Profit: 7
Network edges: 1-2:186  2-3:169  3-20:105
Cost: 460
```

Figure 7. The results of the algorithm for cost limits 500.

In Fig 8, the evolution of the best fitness of the GA is displayed for 3 different runs of the algorithm. The negative values represent invalid solutions, which however are improved and soon become valid without being eliminated from the population. As explained before, the algorithm first tries to decrease the cost, and then to increase the profit. One can also notice the fast convergence of the GA, in less than 20 generations for this test problem.

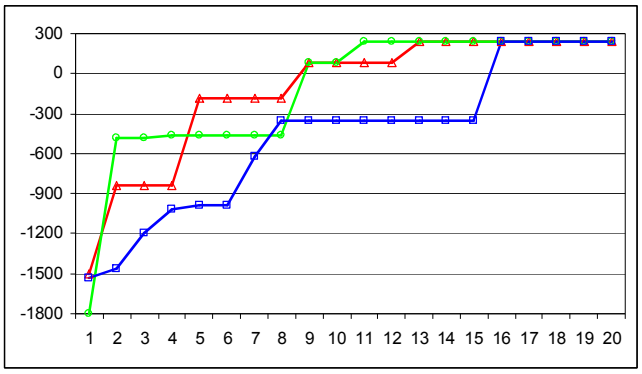

Figure 8. Best fitness evolution for a cost-limited problem.

Fig. 8 displays the fitness, which is 240. The value of 500 is the limit for the cost, not for the fitness, since the latter is the weighted sum of profits and negative costs. The profit of this configuration is 460, which is close to the imposed limit.

B. THE BENCHMARK PROBLEMS

For more complex test problems, we used some of the benchmarks files proposed by Resende [9]: *K100.3*, *K100.6*, and *K100.9*. For all the problems, we used the weights of $w_P = 100$ and $w_C = 1$, with no cost limitation.

The graph in problem *K100.3* is not connected. The algorithm detects this and introduces network edges as needed between the root (the first node) and each of the unconnected parts of the graph, with a very large cost. The solution of the algorithm is presented below (Fig. 9).

```
Connected: 1-2
Connected: 1-21

Best fitness: 3984062
Best chromosome:
01000000111001010111010000101110010101001 1
1100110111111101011001100001100011011 0
11000001000000101110
Selected vertices: 1 2 6 7 8 9 10 11:20628
14 15 16 18 19 20 22 23 25 27 29 30:16155
31 32 34 35 36 38 41 42 43 44:3259 46 47
48 49 50 51 52 53 54 55 56 58 59 60
61:16456 62:17797 63:28396 64 65 68 70 71
73 75 76 77 78:15719 79 81:13488 82 85 86
88 89 90 91 93 94 95:3222 96 97 98:9229 99
100
Profit: 144349
Cost: 10450838
```

Figure 9. The solution of the algorithm in the *K100.3* benchmark case.

For the K100.6 problem, the algorithm finds the following solution:

```
Best fitness: 15430863
Best chromosome:
11000110110110011100101011010001100010101 1
00000101011100101101000000110001000011
11001011010000001000
Selected vertices: 1 2 5 6:16311 7:249 8
9:10244 10 11 12:11697 13 15 16 17 18 20
21 22 23 24:7954 25 26 27 28 29 30 32 33
37:24691 38 39 41:15668 42 44 45 48 50 52
53 54 55 56 57 59 60 62 63 64 65:9092 67
68 69 70 72 73 74:20033 77 78 79 80 81 82
84 85:21197 86 87 88 90 91 92 94 95
96:22756 97 98 99
Profit: 159892
Cost: 558337
```

Figure 10. The solution of the algorithm in the *K100.6* benchmark case.

The solution for the *K100.9* problem is presented in Fig.11.

```
Best fitness: 14077005
Best chromosome:
00110110011100011000111101001010001000110 1
00110110110100100100000010101111000000
10101100010001011101
Selected vertices: 1 2 3:23535 4 5 6 7 9
10 11 12 13 14 16 17 21:11599 22 23
24:17216 26 27 28 29 30 31 32 33:17175 35
37 39 40 41 42 43 44 45 46 47 48:17750
49:18876 50:2678 51 52 54 55:6003 57:18584
59 60 61 63 64 67 68 69 71 72 73 74 77 79
80 81 82 83 84 85 86:10416 87 88 90:2620
91 93 94 95 96 97 98 99 100
Profit: 146452
Cost: 568195
```

Figure 11. The solution of the algorithm in the *K100.11* benchmark case.

V. CONCLUSIONS

We propose a new algorithm for district heating network extension with new customers. The experimental results show that the algorithm works well for small and sparse graphs and the running time is reasonable.

The choices of w_p and w_c are made experimentally and depend on application. The cost limit restricts the results, so the usually preferred solution is no cost limits.

The running time is high for problems K100.3, 6, 9 and

the optimization is one of the future research. The extension of the algorithm for dense and massively graph is another objective of the future research.

The problem is intensively computational. Therefore, the parallel algorithms for shorter paths and minimum spanning tree could be very useful.

REFERENCES

[1] D. Arotăriței, M. Craus, "An evolutionary Algorithm for Multi-criteria Optimization of District Heating Networw", Buletinul Institutului Politehnic din Iaşi, tomul LII (LVI), fasc.1-4, Automatic Control and Computer Science Section, pp. 97 -106, 2006.

[2] P. Feofiloff, C.G. Fernandes, C.E. Ferreira, J.C. de Pina, "Primal-Dual Approximation Algorithms for the Prize-Collecting Steiner Tree Problem", Information Processing Letters, Vol. 103, Issue 5, pp. 195-202, 2007.

[3] A.M. Costa, J.-F. Cordeau, G. Laporte, "Fast heuristics for the Steiner tree problem with revenues, budget and hop constraints", European Journal of Operational Research, Vol. 190, Issue 1, pp. 68-78, 2008.

[4] H. Esbensen, P. Mazumder, "A Genetic Algorithm for the Steiner Problem in a Graph", European Conf. on Design Automation, pp. 402-406,1994.

[5] J. Gottlieb, B.A. Julstrom , G.R. Raidl, F. Rothlauf, "Prüfer Numbers: A Poor Representation of Spanning Trees for Evolutionary Search", Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001, pp. 343-350, 2001

[6] L. Kou, G. Markovski, L. Berman, "A Fast Algorithm for Steiner Trees", Acta Informatica, 15, pp. 141-145, 1981.

[7] Ljubic, et.al, "An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem", Mathematical Programming, Series B, 105(2-3), pp. 427-449, 2006.

[8] Ljubic, et.al, "Solving the prize-collecting Steiner tree problem to optimality", in: C. Demetrescu, R. Tamassia, and R. Sedgewick, editors, Proceedings of ALENEX 2005, Seventh Workshop on Algorithm Engineering and Experiments, Vancouver, British Columbia, Canada, January 22, 2005.

[9] M. G. C. Resende, "Test problem distribution: Prize collecting Steiner tree", AT&T Labs – Research, Internet and Network Systems Research Center, Algorithms and Optimization Research Department, http://www.research.att.com/~mgcr/data/pcstp.tar.gz, 2003.