



# Minimum Cost $\leq k$ Edges Connected Subgraph Problems

Firdovsi Sharifov<sup>1</sup>

*Department of Cybernetics  
V.M. Glushkov Institute of Cybernetics  
Kiev, Ukraine*

Hakan Kutucu<sup>2</sup>

*Department of Mathematics  
Izmir Institute of Technology  
Izmir, Turkey*

---

## Abstract

The minimum-cost network design problem is considered in the case where an optimum network remains connected, after deleting any  $\leq k$  edges which form a matching in the optimum network. For the case  $k = 1$ , we develop heuristic algorithms to compute a lower and an upper bounds for optimal value of objective function. These algorithms are used in the branch and bound methods to find a solution to the considered problem. We also present computational results.

*Keywords:* Network Models, Isomorphic Graph, Matching

---

<sup>1</sup> Email: [F-sharifov@yandex.ru](mailto:F-sharifov@yandex.ru)

<sup>2</sup> Email: [hakankutucu@iyte.edu.tr](mailto:hakankutucu@iyte.edu.tr)

## 1 Introduction

Let  $G = (V, E)$  be an undirected graph and  $H = (V(H), E(H))$  be any graph for which the set  $E(H)$  of edges forms a matching with cardinality  $|E(H)| \leq k$  ( $k \geq 1$ ). Let  $N$  ( $N \subseteq V$ ) be the set of vertices and a cost  $c_e \geq 0$  be associated with each edge  $e$  of the set  $E$ . Sometimes, the vertices in  $N$  are called terminal nodes. For any  $E(F) \subseteq E$  a subgraph  $F$  of  $G$  with the set of edges  $E(F)$  and the set of vertices  $V(F)$  is denoted by  $F = (V(F), E(F))$ , where  $V(F)$  contains all the end vertices of edges in  $E(F)$ . The cost  $c(E(F))$  of a subgraph  $F$  is the sum of the costs of the edges in the set  $E(F)$ .

Here we consider a minimum cost  $\leq k$  edge connected subgraph problem which consists of finding  $E(G_*) \subseteq E$  (a subgraph  $G_* = (V(G_*), E(G_*))$  of  $G$ ) such that  $c(E(G_*))$  is minimum under the conditions that  $N \subseteq V(G_*)$  and after deleting the edges of a subgraph  $\Gamma$  from  $G_*$ , the obtained subgraph  $(V(G_*), E(G_*) \setminus E(\Gamma))$  of  $G$  contains at least one path between every pair of terminal nodes of  $N$  for every subgraph  $\Gamma$  of  $G$  that is isomorphic to the graph  $H$ , where the set  $E(H)$  of edges forms a matching with cardinality  $|E(H)| \leq k$ .

We abbreviate this problem by ECSP ( $\leq k$ ). The edge connected network design problems that are considered in [1,2,3,4,5,6,7,8] can be formulated by the same way as ECSP ( $\leq k$ ). For example, when  $H$  is the null graph, i.e.  $E(H) = \emptyset$ , the ECSP ( $\leq k$ ) is the Steiner tree problem (STP) on the graph  $G$  and if  $H$  is a graph with  $|E(H)| = k - 1$  and  $N = V$ , then the ECSP ( $\leq k$ ) is equivalent to the minimum cost  $k$ -edge connected network design problem. It is easy to see that any feasible solution of the latter problem is a feasible solution of ECSP ( $\leq k$ ), but the converse is not true.

## 2 Network flow model of the ECSP ( $\leq k$ )

Let  $\mathcal{H}$  be the set of all subgraphs of the graph  $G$  that are isomorphic to  $H$ , i.e.  $\mathcal{H}$  is the set of all matchings in  $G$  with cardinality not greater than  $k$ . Let a node  $s$  from  $N$  be fixed as a source, and let the other nodes from  $N$  be considered as sinks. Then the ECSP ( $\leq k$ ) can be formulated as follows: to find

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

subject to

$$\sum_{j \in \delta_{G-\Gamma}(i)} x_{ij}^r - \sum_{j \in \delta_{G-\Gamma}(i)} x_{ji}^r = \begin{cases} 1 & \text{for } i = s, \\ 0 & \text{for } i \neq s, r, \quad i \in V, r \in N_0, \Gamma \in \mathcal{H}, \\ -1 & \text{for } i = r, \end{cases} \quad (2)$$

$$0 \leq x_{ij}^r \leq x_e, \quad r \in N_0, \quad (ij) = e \in E, \quad (3)$$

$$x_e = 0 \vee 1, \quad e \in E. \quad (4)$$

where  $N_0 = N \setminus \{s\}$ ,  $\delta_{G-\Gamma}(i)$  is the set of edges incident with a common vertex  $i$  in the graph obtained from  $G$  by removing all the edges of  $E(\Gamma)$ . We can reduce the number of constraints in (2) as follows:

Let  $\mathcal{H}(i, j)$  be the set of all subgraphs that are isomorphic to  $H$  and contain the edge  $(i, j)$  in  $E(G)$ , and let  $\bar{\mathcal{H}}(i, j) = \mathcal{H} \setminus \mathcal{H}(i, j)$ . The dual problem of (1)-(3) can be transformed in the following form: to find

$$\max \sum_{r \in N_0} (u_r^r - u_s^r) \quad (5)$$

subject to

$$u_j^r - u_i^r \leq w_{ij}^r + \sum_{\Gamma \in \mathcal{H}(i,j)} (u_j^r(\Gamma) - u_i^r(\Gamma)), \quad r \in N_0, (i, j) \in E, \quad (6)$$

$$\sum_{r \in N_0} w_{ij}^r \leq c_{ij}, \quad (ij) \in E, \quad (7)$$

$$w_{ij}^r \geq 0, \quad r \in N_0, (ij) \in E. \quad (8)$$

**Proposition 2.1** *There exists an optimal solution to the problem (5)-(8) such that a value of  $u_j^r(\Gamma) - u_i^r(\Gamma) \neq 0$  only for one subgraph  $\Gamma \in \mathcal{H}(i, j)$ .*

**Proof.** Suppose that there are two subgraphs  $\Gamma_1, \Gamma_2 \in \mathcal{H}(i, j)$  such that  $u_j^r(\Gamma_t) - u_i^r(\Gamma_t) \neq 0$  for  $t = 1, 2$ . Then we can define

$$u_j^r(\Gamma_1) - u_i^r(\Gamma_1) := u_j^r(\Gamma_1) - u_i^r(\Gamma_1) + u_j^r(\Gamma_2) - u_i^r(\Gamma_2),$$

$$u_j^r(\Gamma_2) - u_i^r(\Gamma_2) := 0.$$

It is easy to see that after these definitions the optimal value of (5) did not change and the constraints in (6) hold. By repeating this process for all pairs of  $\Gamma \in \mathcal{H}(i, j)$  we complete the proof of the proposition.  $\square$

By Proposition 2.1, the constraints (6) can be written in the following form:

$$u_j^r - u_i^r \leq w_{ij}^r + u_j^r(\Gamma(i, j)) - u_i^r(\Gamma(i, j)), \quad r \in N_0, (i, j) \in E,$$

where  $\Gamma(i, j)$  is a subgraph that contains an edge  $(i, j)$  and is isomorphic to  $H$ . Since  $E(H)$  is a matching with cardinality  $|E(H)|$ , then for any edge  $(i, j)$ , we can find a maximum cardinality matching in the graph  $G - \kappa(i, j)$  by any matching algorithm, where  $\kappa(i, j)$  is the set of edges incident with vertices  $i, j$ . Then we obtain a linear programming problem which has  $|E||N|$  variables. One can use this problem to find a lower bound for the ECSP ( $\leq k$ ). In the next section, we consider the case  $k = 1$ .

### 3 ECSP ( $\leq k$ ) when $k = 1$

The case  $k = 1$  means that the graph  $H$  has two vertices and one edge. Hence the graph  $G_*$  has to be connected when any edge  $e$  of  $E(G_*)$  is deleted from  $G_*$ . When  $N = \{s, r\}$  ( $|N| = 2$ ), the ECSP ( $\leq k$ ) can be solved in time  $O(|V|^3)$ , by calculating the minimum-cost flow in the network  $G$  with the source  $s$  and the sink  $r$ , while all the edges in  $G$  have unit capacities, and the value of a flow is equal to 2. If  $N = V$  and  $G$  is a complete graph and the costs of its edges satisfy the triangle inequality, linear relaxations of the Traveling Salesman Problem (TSP) and the ECSP ( $\leq k$ ) are equivalent.

#### 3.1 An algorithm to find a lower bound

The dual problem can be formulated as the following linear programming problem: to find

$$\max \quad 2 \sum_{r \in N_0} (u_r^r - u_s^r) - \sum_{(ij) \in E} z_{ij} \tag{9}$$

subject to

$$u_j^r - u_i^r \leq w_{ij}^r, \quad (ij) \in E, \quad r \in N_0, \tag{10}$$

$$\sum_{r \in N_0} w_{ij}^r \leq c_{ij} + z_{ij}, \quad (ij) \in E, \tag{11}$$

$$w_{ij}^r \geq 0, \quad z_{ij} \geq 0, \quad (ij) \in E, \quad r \in N_0. \tag{12}$$

It is clear that the value of the objective function (9) is a lower bound  $F(\text{lower})$  for the ECSP ( $\leq k$ ).

**Proposition 3.1** *There is an optimum solution  $u_v^r, w_{ij}^r, v \in V, r \in N_0, (i, j) \in E$ , to problem (9)-(12) such that the equality*

$$w_{ij}^r = \max\{0, u_j^r - u_i^r\}, r \in N_0, (i, j) \in E. \tag{13}$$

holds.

By this proposition, after computing the value of  $u_v^{r_0}$  for all  $v \in V$  for any fixed  $r_0 \in N_0$ , we use (13) to define the values of  $w_{ij}^{r_0}$ , for all  $(i, j) \in E$ . In order to find the optimum value of  $u_v^{r_0}$ , we solve the following problem

$$\max \quad 2(u_{r_0}^{r_0} - u_s^{r_0}) - \sum_{(ij) \in E} z_{ij}$$

subject to

$$u_j^{r_0} - u_i^{r_0} \leq c_{ij} + z_{ij}, \quad (ij) \in E,$$

$$z_{ij} \geq 0, \quad (ij) \in E.$$

This is the dual of the well known problem of minimum-cost two edge-disjoint paths between nodes  $s$  and  $r_0$ . Suppose that  $u_v^{r_0}; v \in V$  and  $z_{ij}^0; (ij) \in E$  are an optimal solution. In the next iteration, we fix a new sink  $r_1 \neq r_0$  in  $N_0$  and set

$$u_v = u_v^{r_0}, \text{ if } u_v > u_v^{r_0}$$

$$c_{ij} := \max\{0, c_{ij} - u_j^{r_0} + u_i^{r_0}\} \text{ for } (i, j) \in E. \tag{14}$$

We again find two edge-disjoint paths with minimum total cost in the network  $G$ , in which a source is the node  $s$ , a sink is the node  $r_1$ , and  $c_e$  is the cost of the edge  $e$  for  $e \in E$ . Therefore, we find two edge-disjoint paths  $|N| - 1$  times and at every iteration  $t$ , we have  $u_v^{r_t}; v \in V$  and  $z_{ij}^t; (ij) \in E$  and define the value of  $w_{ij}^{r_t}$  by (13). From (14), it follows that the constraints (11) hold for the values of  $w_{ij}^{r_t}$ . Finally, we define the value of  $z_{ij}$  by

$$z_{ij} = \sum_{t=1}^{|N_0|} z_{ij}^t.$$

Since constraints (11) hold for  $w_{ij}^r$ , we conclude that the values of  $u_v^r, z_{ij}, w_{ij}^r$  are found by the above algorithm are the feasible solution to problem (9)-(12), and the value of (9) corresponding to this solution is a lower bound for the ECSP ( $\leq k$ ). This algorithm computes the lower bound in  $O(|V|^3|N_0|)$  time.

### 3.2 Heuristic algorithm when $k = 1$

Now we present heuristic algorithm for constructing an initial feasible network and local improvement heuristics for reducing the cost of the network, while preserving its feasibility. These heuristics are used in a local search approach to obtain low-cost network designs. Our algorithm is based on the approach suitable for constructing a two-edge connected network  $\bar{G}$ . Then we delete some edges from the network  $\bar{G}$  in such a way that  $\bar{G}$  remains to be a feasible network. After that, we reduce (if it is possible) the cost of the network  $\bar{G}$  using a locally improving heuristic algorithm.

Let  $E_0 \subseteq E$  be a set of edges  $(i, j)$  for which the constraints in (11) hold as an equality in the defining  $F(\text{lower})$  and let  $G(E_0)$  be the subgraph of  $G$  which is defined by the set  $E_0$ . It is easy to show that the graph  $G(E_0)$  is at least two-edge connected. Now we describe the algorithm to construct a two-edge connected network  $\bar{G} = (\bar{V}, \bar{E})$  such that the underlying graph  $\bar{G}$  is the subgraph of  $G(E_0)$ .

At the beginning of the operation of the algorithm, we set  $\bar{V} = \emptyset$  and  $\bar{E} = \emptyset$ . Then we find the shortest paths between every pair of terminal nodes in the network  $G(E_0)$ . Let  $L$  be a list of all these shortest paths. In this list, we choose the minimum path  $P_1$ . Let this path connect nodes  $v_1$  and  $v_2$ . We delete this path from the list  $L$  and include all its edges and all vertices to the sets  $\bar{E}$  and  $\bar{V}$ , respectively. In the list, we again choose the minimum path  $P_2$ . Let it connect the nodes  $v_3$  and  $v_4$ . If the degrees of the nodes  $v_3$  and  $v_4$  (with respect to the current graph  $\bar{G} = (\bar{V}, \bar{E})$ ) are greater or equal to 2, we delete this path from the list  $L$  and its vertices and edges are not included into  $\bar{V}$  and  $\bar{E}$ . Otherwise, we include its edges and vertices  $\bar{E}$  and  $\bar{V}$ , respectively. We repeat this procedure while list  $L$  is not empty. Let  $\bar{G}$  be the graph which is constructed in such a way. It is clear that  $d_f \geq 2$  for all vertices  $f$  in  $\bar{V} \setminus N$ .

In order to verify whether the graph is two-edge connected, we find two disjoint shortest paths between every pair of distinct nodes in  $N$  as follows: let  $w, v \in N$  and  $P(v, w)$  is the first shortest path in the graph  $\bar{G} = (\bar{V}, \bar{E})$ . We set weights of edges of  $P(v, w)$  to a big number  $M$ . Again find the shortest path between nodes  $w, v \in N$ . If the length of this path is not less than  $M$ , then we find common edges on these paths. Let  $(i, j)$  be the common edge. Then we find a shortest path that connects nodes  $i$  and  $j$  in the graph induced by set  $(E_0 \setminus \bar{E})$  and its vertices and edges are included into  $\bar{V}$  and  $\bar{E}$ . We repeat this procedure at most  $O(|N|^2)$  times to find  $\bar{G}$  which are feasible solutions to ECSP ( $\leq k$ ) when  $k = 1$ .

### 3.3 Computational results

In Table 1, the costs of edges are randomly chosen in the interval  $[0, 100]$ . In Table 1,  $n$  is the number of vertices,  $m$  is the number of edges,  $node$  is the number of terminal nodes,  $F_0(lower)$  is the value of a lower bound computed at zero iteration,  $F_0(upper)$  is the value of the upper bound computed at iteration zero,  $F_r$  is the record value of the objective function that is defined by the branch and bound algorithm,  $nF_r$  is the number of changes for the record value of the objective function of ECSP ( $\leq k$ ) when  $k = 1$ . We choose a new branch to continue the process of branching if  $(F_c(upper) - F_c(lower))/F_c(lower) \leq 0.05$ , where  $F_c(upper)(F_c(lower))$  current upper (lower) bound at a iteration.

Table 1  
Computational Results

#instance	$n$	$m$	node	$F(lower)$	$F(upper)$	$F_r$	$nF_r$	time
1	8	12	4	20	20	20	0	0.001 sec.
2	17	29	5	40	44	44	0	0.002 sec.
3	38	65	8	552	553	553	0	0.002 sec.
4	10	45	10	140	190	170	5	0.011 sec.
5	15	105	10	113	118	116	1	0.092 sec.
6	15	105	15	122	296	182	4	0.148 sec
7	20	190	10	54	112	104	5	6.478 sec.
8	30	435	10	60	100	95	3	18.419 sec.
9	40	780	10	32	63	50	3	4.298 sec.
10	50	1225	10	30	55	52	2	1.249 sec.
11	50	1225	20	65	115	113	1	1.749 sec.
12	50	1225	30	87	131	131	0	1.397 sec.
13	50	1225	40	91	181	180	1	1.525 sec.
14	50	1225	50	97	211	211	0	1.804 sec.

Computational analysis is performed on a workstation with 2.4 Ghz Intel processor and 2 GB of ram.

## 4 Conclusion

There are many problems on graphs that can be formulated as (1)-(4) by defining different types of graphs  $H$ . If  $H$  is a graph such that one can find a subgraph of  $G$  that is isomorphic to  $H$  by a polynomial-time algorithm then by proposition 1, the dual LP problem has  $O(|E||N|)$  constraints. Hence,  $H$  can be used to compute a lower bound in the latter problem. In order to find an optimal solution for considered problems, one can use the same framework of branch and bound algorithm that we are used for the case  $k = 1$ . In the future, we are going to implement this approach to the ECSP ( $\leq k$ ) for the case  $k > 1$ .

## References

- [1] Balakrishnan, A., T. L. Magnanti and P. Mirchandani, *Connectivity-Splitting Model for Survivable Network Design*, Network, **43** (2004), 10–27.
- [2] Bienstock, D., E. F. Brickell and C.L. Monma, *On the structure of minimum-weight  $k$ -connected spanning networks*, SIAM J. Discr. Math., **3** (1990), 320–329.
- [3] Chopra, S., *The  $k$ -edge connected spanning subgraph polytope*, SIAM J. Discr. Math., **7** (1994), 245–259.
- [4] Didi Biha, M., and A. R. Mahjoub,  *$k$ -Edge connected polyhedra on seriesparallel graphs*, Oper. Res. Lett. **19** (1996), 71–78.
- [5] Didi Biha, M., and A. R. Mahjoub, *Steiner  $k$ -edge connected subgraph polyhedra*, J. Combinatorica, **4** (2000), 131–144.
- [6] Grötschel, M., and C. L. Monma, *Integer polyhedra arising from certain network design problems with connectivity constraints*, SIAM.J.Disc. Math., **4** (1990), 502–522.
- [7] Mahjoub, A. R., *Two-edge connected spanning subgraphs and polyhedra*, Math. Prog., **64** (1994), 199–208.
- [8] Monma, C. L., and D. F. Shallcross, *Methods for designing communications networks with certain two-connected survivability constraints*, Oper. Res., **3** (1989), 531–541.