Two-Edge Disjoint Survivable Network Design Problem with Relays: A Hybrid Genetic Algorithm and Lagrangian Heuristic Approach

Abdullah Konak
Associate Professor of Information Sciences and Technology
Penn State Berks
Tulpehocken Rd. PO Box 7009
Reading, PA19610 USA
konak@psu.edu

Two-Edge Disjoint Survivable Network Design Problem with Relays: A Hybrid Genetic Algorithm and Lagrangian Heuristic Approach

Abdullah Konak
Penn State Berks, USA

***Abstract*:** This paper presents a network design problem with relays considering the two-edge network connectivity. The problem arises in telecommunication and logistic networks where a constraint is imposed on the distance that a commodity can travel on a route without being processed by a relay, and the survivability of the network is critical in case of a component failure. The network design problem involves selecting two edge-disjoint paths between source and destination node pairs and determining the location of relays to minimize the network design cost. The formulated problem is solved by a hybrid approach of a genetic algorithm (GA) and a Lagrangian heuristic such that the GA searches for two-edge disjoint paths for each commodity, and the Lagrangian heuristic is used to determine relays on these paths. The performance of the proposed hybrid approach is compared to the previous approaches from the literature with promising results.

***Keywords*:** Network Design, Network Survivability, Relays, Genetic Algorithm

1. Introduction

The network design problem with relays (NDPR) arises in telecommunication and distribution networks where commodities routed through a network must be processed within certain intervals. In fiber-optic networks, for example, lightwave signals must be regenerated periodically due to transmission impairments such as signal distortion and noise (Winters et al., 1993). Regeneration of lightwave signals is a complex and costly process which requires expensive equipment (Mukherjee, 2000). In natural gas distribution networks, minimum levels of pressure should be maintained at various pipeline segments. This goal can be achieved by strategically locating compressors on the network and properly adjusting pipeline diameters (Andre et al., 2009, Kabirian and

Hemmati, 2007, Tabkhi et al., 2009). In multi-zone truck dispatching systems, a truckload may be handled by multiple drivers or trucks during the trip from its origin to its destination (Taylor et al., 2001). The common aspect of these three problems is that a commodity cannot travel more than a certain distance without visiting a processing unit, which is called a relay in this paper (i.e., regeneration points in fiber networks, compressors in gas distribution systems, and hubs where truckloads switch drivers or trucks in multi-zone truck dispatching systems)

The NDPR is introduced by Cabral et al. (2007) in the context of a wireless telecommunication network design problem. In this paper, the two-edge connected network design problem with relays (2ECON-NDPR), which is an extension of the NDPR by incorporating network survivability, is considered. Network survivability is the ability of a network to continue serving its users in the case of catastrophic component failures. Network survivability is an important concern for telecommunication networks as network topologies have become sparser because of high capacity fiber-optic links. The survivability of a network in case of component failures is achieved through the availability of redundant paths between nodes. In a network, two nodes are said to be two-edge connected if there exist two edge-disjoint paths between the nodes. Two-edge connectivity ensures the connectivity of network in the case of a single edge failure. For many real-world networks, a level of redundancy to protect against a single edge or node failure is sufficient because component failures are so rare that the probability of observing another failure during a repair is almost zero (Magnanti et al., 1995, Monma and Shallcross, 1989, Konak and Smith, 2004). Therefore, the survivable network design literature mainly focuses on two-edge or two-node connectivity problems. In this paper, two-edge connectivity (2ECON) in the context of the NDPR is studied, and a hybrid solution approach between a genetic algorithm (GA) and a Lagrangian heuristic is proposed. The proposed hybrid approach is referred to as the GA-LH in the following sections of the paper.

The 2ECON-NDPR is defined as follows. An undirected network $G=(V, E)$ with node set $V=\{1, 2, \ldots, N\}$ and edge set E is given. Each edge (i, j) is associated with a cost of c_{ij} and a distance of d_{ij}. A set of commodities (K), representing point-to-point traffics, are to be simultaneously routed on the network. Each commodity k has a single

source node $s(k)$ and a single destination node $t(k)$. For each commodity k, two edge-disjoint paths, $\pi_1(k)$ and $\pi_2(k)$, from node $s(k)$ to node $t(k)$ are to be determined. An upper bound λ is imposed on the distance that a commodity k can travel on paths $\pi_1(k)$ and $\pi_2(k)$ without visiting a relay. Fixed cost ρ_i is incurred if a relay is located at node i. The objective function of the 2ECON-NDPR is to minimize the network design cost while making sure that each commodity k can be routed from node $s(k)$ to node $t(k)$ such that the distances between node $s(k)$ and the first relay, between any consecutive relays, and between the last relay and node $t(k)$ do not exceed the upper bound λ on both paths $\pi_1(k)$ and $\pi_2(k)$.

The integer programming formulation (Problem 2PATH-NDPR) given below presents a path-based formulation for the 2ECON-NDPR where relay constraints are modeled as set covering constraints as described by Konak (2012a) for the NDPR. Problem 2PATH-NDPR enables to solve the problem in two phases. In the GA-LH approach defined in this paper, a GA is used to search two edge-disjoint paths for each commodity, and a Lagrangian heuristic is used to determine relay assignments for the encoded paths in the GA's chromosomes. The 2ECON-NDPR is a more challenging problem than the NDPR with single paths not only because of additional edge-disjoint path constraints but also because of the difficulty of solving resulting set covering sub-problems. The computational experiments in Section 4 show that the GA-LH performs well and can effectively solve large problem instances.

<u>Decision Variables & Parameters:</u>

$\boldsymbol{P}(k)$	set of all paths from source node $s(k)$ to sink node $t(k)$ of commodity k.
$\boldsymbol{V}(k, p)$	set of the nodes, including nodes $s(k)$ and $t(k)$, on path p of commodity k.
$\boldsymbol{V}(k, p, j, \lambda)$	set of the nodes that can be traversed on path p of commodity k starting from node j, in the direction from node $s(k)$ to node $t(k)$ without violating upper bound λ.
$\delta_{ij}(p,k)$	edge-path indicator parameter such that $\delta_{ij}(p,k)=1$ if path p of commodity k includes edge (i, j), and $\delta_{ij}(p,k)=0$ otherwise.
x_{ij}	binary edge decision variable such that $x_{ij}=1$ if edge (i, j) is selected in the solution, and $x_{ij}=0$ otherwise.

y_i binary relay decision variable such that y_i=1 if a relay is located at node i, and y_i= 0 otherwise.

v_{kp} binary path decision variable such that v_{kp}=1 if path p, $p \in P(k)$, is used by commodity k, and v_{kp}=0, otherwise.

<u>Problem 2PATH-NDPR:</u>

$$\text{Min} \sum_{i \in V} \rho_i y_i + \sum_{(i,j) \in E} c_{ij} x_{ij}$$

s.t:

$$\sum_{p \in P(k)} v_{kp} = 2 \qquad k \in K \tag{1}$$

$$\sum_{p \in P(k)} \delta_{ij}(k,p) v_{kp} \le x_{ij} \qquad (i,j) \in E, k \in K \tag{2}$$

$$\sum_{i \in V(k,p,j,\lambda)} y_i \ge v_{kp} \qquad k \in K, p \in P(k), j \in V(k,p): j \ne t(k), t(k) \notin V(k,p,j,\lambda) \tag{3}$$

$$y_i, x_{ij}, v_{kp} \in \{0,1\}$$

Constraint (1) makes sure that two paths are chosen for each commodity k. Constraint (2) has two functions. Firstly, constraint (2) states that edge (i, j) must be included in the solution if a path including edge (i, j) is selected. Secondly, constraint (2) ensures that two paths sharing a common edge (i, j) cannot be selected for commodity k. Note that Problem 2PATH-NDPR is defined for an undirected network, and edge (i, j) represents directed arcs (i, j) and (j, i). By definition, $\delta_{ij}(p,k)$=1 if path p from node $s(k)$ to node $t(k)$ consecutively visits nodes i and j in any order. Therefore, selecting two paths, one including directed arc (i, j) and the other including directed arc (j, i), for a commodity is not permissible due to constraint (2). Constraint (3) represents set covering constraints by considering each node j on path p of commodity k if this path is selected. Figure 1 illustrates an example of constraint (3) where $s(k)$=1 and $t(k)$=6, and λ=4. If the commodity starts at node 1, a relay has be located at node 2 or node 3 to make sure that the commodity can be routed beyond node 3. Therefore, $V(k, p, 1, 4)$={2, 3}. Similarly, starting from node 2, a relay has to be located at node 3 or node 4, otherwise the commodity cannot be routed farther than node 4 (i.e., $V(k, p, 2, 4)$={3, 4}). A relay constraint is not needed for nodes 4 and 5 because if the commodity starts at either of

these nodes, it can be routed to its destination without violating the relay constraint (i.e., $6 \in V(k,p,4,4)$ and $6 \in V(k,p,5,4)$).

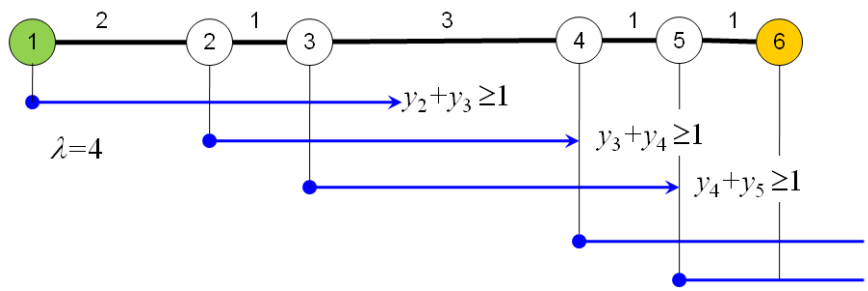

Figure 1. An example of generating set-covering constraints from paths.

2. Previous Work on the NDPR and 2ECON-NDPR

In the literature, several network design problems are closely related with the 2ECON-NDPR although they are not directly applicable. One of the related problems is the regenerator assignment problem, which is defined as to determine the location of regeneration points in a given network. Particularly in digital telecommunication networks, digital signals must be regenerated into its original forms after being transmitted over a certain distance. Ramamurthy et al. (1997, 1998) study the regenerator assignment problem in optical networks, where the objective is to determine the minimum number of regenerators and their exact locations in the network to cope with light-wave attenuation during transmission. Gouveia et al. (2003) formulate an assignment problem for label-switching routers in optical networks considering quality-of-service and distance requirements. In this formulation, the total distance that a light path travels between two label switching routers must not exceed an upper bound due to transmission loss, and there is limited on the number of routers through which a light path can be routed because of processing delays at label switching routers. Yetginer and Karasan (2003) propose heuristic approaches to determine regenerator points in a network under traffic load uncertainty. Chen et al. (2010) define the minimum regenerator assignment problem in optical networks, where the objective is to minimize the number of regenerators under a total distance constraint that the path between each node pair can travel without visiting a regenerator. Chen et al. (2010) also show that the

minimum regenerator assignment problem can be formulated as the Steiner arborescence problem with a unit degree constraint.

The NDPR is also related with the hop-constrained network design problem and the rooted distance-constrained minimum spanning tree problem. In the hop constrained network design problem, hop constraints impose an upper bound on the number of edges between source and destination nodes due to reliability concerns (LeBlanc and Reddoch, 1990) or performance concerns (Balakrishnan and Altinkemer, 1992). Several papers (Choplin, 2001, Kwangil and Shayman, 2005, Randall et al., 2002) address optical network design problems considering restricted transmission range due to optical impairments, which is also one of the main motivations in the NDPR. Voss (Voss, 1999) considers the hop-constrained Steiner tree problem and proposes a solution approach based on tabu search and mathematical programming. Gouveia and Magnanti (2003) study the diameter-constrained minimum spanning and Steiner tree problems where an upper-bound is imposed on the number of edges between any node pairs. Gouveia (1996) proposes two different mathematical models for the hop-constrained minimum spanning tree problem as well as Lagrangian relaxation and heuristic approaches to solve the problem. Subsequently, Gouveia and Requejo (2001) present an improved Lagrangian relaxation approach to the problem. Recently, Gouveia et al. (2011) has shown that the hop-constrained minimum spanning tree problem can be formulated as a Steiner tree problem (STP) in an appropriate layered network, which leads to a tighter formulation of the problem.

In the rooted distance-constrained minimum spanning tree problem (Gouveia et al., 2008, Ruthmair and Raidl, 2011, Leitner et al., 2011), each edge has an associated cost and a distance, and the objective is to find a spanning tree on a given network with a minimum total cost such that the path from a specified root node to any other node has a total distance or delay not exceeding an upper-bound. However, relays are not considered in the rooted distance-constrained minimum spanning tree problem.

In the literature, limited work has been published on the NDPR considering survivability constraints. As mentioned in the previous section, the NDPR is defined by Cabral et al. (2007) in the context of wireless telecommunication networks. The authors formulate a path-based integer programming model and propose a column generation

approach to solve the NDPR with a single path for each commodity. Unfortunately, Cabral et al. (2007)'s column generation approach cannot be practically used for large problem instances. Therefore, Cabral et al. (2007) propose four different construction heuristics in which a solution is constructed by taking into consideration one commodity at a time. Konak et al. (2009) provide a flow-based formulation for the 2ECON-NDPR, and they have modified Cabral et al. (2007)'s construction heuristics to solve the 2ECON-NDPR such that the flow-based formulation is used to optimally solve subproblems that arise in the solution construction process of these heuristics. In addition, they develop a GA for the 2ECON-NDPR based on an earlier hybrid local search-GA (Kulturel-Konak and Konak, 2008) where a specialized crossover operator and local search are used to create new solutions. The construction heuristics and the GA reported by Konak et al. (2009) are the only work in the literature addressing 2ECON in the context of the NDPR. Therefore, in the computational experiments the performance of the proposed hybrid approach is compared to these previous approaches. Recently, Konak (2012a) proposes a path-based formulation with set covering constraints for the NDPR where commodities are routed through a single path. In Konak (2012a)'s approach, the relay problems are exactly solved using Integer Programming, which is computationally feasible since commodities are routed through a single path. When two-edge connectivity is considered, however, determining optimal relay assignments becomes computationally difficult. Therefore, a Lagrangian heuristic is used to efficiently solve relay assignment subproblems in the GA-LH.

3. A Genetic Algorithm Approach to 2ECON-NDPR

In this section, the GA-LH is introduced to find good solutions to the 2ECON-NDPR. The parameters and notation used in the GA-LH are given below.

z	a solution
$x_{ij}(z)$	edge decision variable of solution z such that $x_{ij}(z)=1$ if edge (i,j) is selected in solution z, and $x_{ij}(z)=0$ otherwise.
$y_i(z)$	relay decision variable of solution z such that $y_i(z)=1$ if a relay is located at node i, 0 otherwise.
$\pi_p(k,z)$	ordered set edges representing the pth path of commodity k in solution z

$n(p,k,z)$	number of nodes in the pth path of commodity k in solution z
$E(z)$	edge set of solution z, $E(z) \subset E$
$V(z)$	node set of solution z, $V(z) \subset V$
PP	parent population
OP	offspring population
μ	population size
$rank(z)$	rank of solution z in the population
$t_{\max}$	maximum number of generations (stopping criteria)
$U(0,1)$	uniform random number between 0 and 1
$\|A\|$	cardinality of set A

3.1. Encoding and Solution Representation

If the primary and secondary paths for each commodity are known, Problem 2PATH-NDPR reduces to a set covering problem. This observation is used in the GA-LH to solve the problem in two steps: identifying the paths in the first step and assigning relays in the second step. Therefore, the GA-LH chromosome encodes the primary and secondary paths of commodities, but not relays which are determined by solving a set covering problem. A chromosome z is represented as $z=\{\pi_1(1),\ldots,\pi_1(|K|); \pi_2(1),\ldots,\pi_2(|K|)\}$ where $\pi_1(k)$ and $\pi_2(k)$ are ordered sets of edges representing the primary and secondary paths of commodity k, respectively. A set covering problem is constructed and solved to determine the relays for a given chromosome.

3.2. Determining Relays and Solution Evaluation

For a given chromosome z, relays are determined by solving the following set covering problem:

<u>Problem Relay(z):</u>

$$\text{Min} \sum_{j \in V(z)} \rho_j y_j(z)$$

s.t:

$$\sum_{j \in V(z)} a_{rj}(z) y_j(z) \geq 1 \quad r = 1,\ldots,M(z)$$

$$y_j \in \{0,1\}$$

where $a_{rj}(z)$ is the binary coefficient indicating whether node j covers constraint r (i.e., $a_{rj}(z)=1$) or not (i.e., $a_{rj}(z)=0$) if a relay is located at node j, and $M(z)$ denotes the number of set covering constraints for chromosome z. Coefficient matrix $a_{rj}(z)$ can be constructed by sequentially analyzing nodes on paths $\pi_1(k,z)$ and $\pi_2(k,z)$ as follows:

```
Procedure Construct_Problem_Relay(z) {
    Set a_rj(z)=0 for all possible values of r and j ;
    r=1 ;
    for k=1,…,|K| do {
        for p=1,..,2 do {
            for l=1,…,(n(p,k,z)-1) do {
                td=0 ; q=l ;
                do {
                    Let i and j denote the qth and (q+1)th nodes, respectively, in path π_p(k,z);
                    td=td+d_ij;
                    if td ≥ λ and j≠t(k) then {
                        if td=λ then q=q+1; //include the (q+1)th node as well
                        for u=l+1,…,q do{
                            Let node i be the uth node in path π_p(k,z);
                            a_ri(z)=1;
                        }
                        r=r+1; //add a new constraint
                    }
                    q=q+1; //move to the next node in the path
                } while (td < λ and j≠t(k) )
            }
        }
    }
    Set M(z)=r and return coefficients a_rj(z) ;
}
```

Problem Relay(z) can be solved using various heuristic and exact methods. In this paper, two approaches are considered: Integer Programming and a Lagrangian heuristic adopted from Beasley (1990) with minor modifications. During the search, the Lagrangian heuristic is used to solve Problem Relay(z), and the at the termination of the GA-LH, the problem is optimally solved for the best solution found during the search. Before solving Problem Relay(z), all set covering constraints are preprocessed by eliminating duplicate constraints and setting $y_j(z)=1$ if node j is a single node in a constraint of Problem Relay(z). Let Relay(z') represent the reduced set covering problem

for chromosome z after eliminating all nodes with $y_j(z)=1$, all satisfied constraints, and all duplicate constraints. The Lagrangian lower bound problem is given as follows:

<u>Problem RelayLLB(z'):</u>

$$\text{Min LB}=\sum_{j\in V(z')} C_j y_j(z') + \sum_{r=1}^{M(z')} \mu_r a_{rj}(z')$$

$$y_j(z') \in \{0,1\}$$

where $\mu_r \geq 0$ is the Lagrange multiplier associated with the rth constraint of the reduced problem, and $C_j = \left(\rho_j - \sum_{r=1}^{M(z')} \mu_r a_{rj}(z')\right)$ is the reduced cost of node $j \in V(z')$. Problem RelayLLB(z') can be solved by setting $y_j(z')=1$ if $C_j \leq 0$ and $y_j(z')=0$ if $C_j > 0$. The subgradient procedure given by Beasley (1990) is used to update the Lagrange multipliers. In the subgradient procedure, the step size parameter is halved if the lower bound has not been improved in the last 20 iterations (30 is used by Beasley (1990)).

3.3. Crossover/Mutation Operator

GAs rely on crossover and mutation operators to create new solutions from existing ones in the population. The function of a crossover operator is to create new solutions by recombining building blocks (called genes) of two or more chromosomes (called parents). For example, the single point crossover operator creates a new solution from two parents by copying all genes before a randomly selected crossover point from one parent and the rest from the other parent. Parents are usually selected randomly with a bias toward fittest solutions in the population to encourage passing their characteristics that make them fittest to the next generations. Repeatedly applying crossover and selecting solutions with high fitness values for the next generation, the population eventually converges. In GAs, crossover operators do not have the ability to introduce new solution characteristics to the population because they only use genes available in the population to create new solutions. The function of mutation in GAs is to introduce new solution characteristics to the population through random small changes in chromosomes. Thereby, a solution characteristic that might be missing in the current

population may appear in the future populations. Without a mutation operation, GAs tend to converge quickly and would not be able to escape from local optima.

Selecting a good combination of crossover and mutation is critical for the performance of a GA. However, it is challenging to design effective crossover and mutation operators for problems based on network structures. Traditional GA crossover and mutation operators are likely to generate unacceptable network topologies which have to be repaired. For example, the single point crossover can create a disconnected network from two parent networks which are connected. In the case of the 2ECON-NDPR, a crossover operator should maintain connectivity of paths and ensure that the primary and secondary paths are edge-disjoint. In addition, the dependencies among paths should be considered to design an effective crossover in the case of the 2ECON-NDPR. Because of these reasons, a special crossover/mutation operator is proposed by extending the crossover operator given by Konak (Konak, 2012b) for the NDPR. To maintain the integrity of paths in the crossover/mutation operator of the GA-LH, the Suurballe-Tarjan algorithm (Suurballe and Tarjan, 1984) is used. The Suurballe-Tarjan algorithm determines the shortest pair of edge-disjoint paths between two nodes in polynomial time. The crossover/mutation operator of the GA-LH is in fact a construction heuristic that generates an offspring from two randomly selected parents and randomly selected edges by considering one commodity at a time in a random order of commodities. The details of the crossover/mutation operator are given in the procedure below. In Figure 2, a step-by-step example of the crossover/mutation operator is illustrated for a small problem with two commodities.

```
Procedure CrossoverMutation(a, b) {
   V′= V(a)∪V(b) and E′=E(a)∪E(b);
   Randomly and uniformly select an edge (i+, j+) from E\E′ such that i+∈V′ and j+∈ V′;
   //edge mutation
   E′= E′∪(i+, j+);
   //node mutation
   Randomly and uniformly select a node i+ from V\V′ such that |{(i+, j): j∈V′}| ≥ 2 ;
   V′= V′∪ i+ and E′= E′∪{(i+, j): j∈V′};
   //set random temporary costs
   for (i, j)∈E′ do {
      if (i, j)∈E(a)∩E(b) then tc_ij=U(0,1)/2;
                          else tc_ij=U(0,1);
   }
   //construct the offspring
   K′=K;
   while K′≠∅ do {
      Randomly and uniformly select a commodity k from K′;
      Find edge-disjoint shortest paths π1(k,z) and π2(k,z) from s(k) to t(k) in G′;
      Set x_ij(z) =1 and tc_ij=0 for all (i, j)∈{π1(k,z)∪π2(k,z)};
      K′=K′ \ k;
   }
Return offspring z;
}
```

In Procedure *CrossoverMutation*, given two parent solutions a and b, first a partial network $G'=(V', E')$ is constructed by combining all nodes and edges of the parents as seen in Figure 2-(*ii*). Offspring z is constructed from partial network G' one commodity at a time by determining two edge-disjoint paths and adding these paths to the offspring. Before the solution construction step, partial network G' is mutated by adding random edges and a random node. First, an edge (i^+, j^+) is added to the partial network (Figure 2-(*iii*)). Edge (i^+, j^+) is selected randomly and uniformly from the set of edges which do not exist in the either of the parents (i.e., $(i^+, j^+)\in(E\backslash E')$), but whose end-nodes are in partial network G' (i.e., $i^+\in V'$ and $j^+\in V'$). The latter condition is required to make sure that edge (i^+, j^+) can be used by a path. Next, partial network G' is mutated by adding a random node. A node i^+ from node set $V\backslash V'$ and its all edges connected to partial network G' are added to partial network G' (Figure 2-(*iv*)). Node i^+ is randomly and uniformly selected among the set of nodes that can be connected to partial network G' through at least two edges (i.e., $|\{(i^+, j): j\in V'\}| \geq 2$). This condition is required so that

node i^+ can be on a path as well. Otherwise, it would be impossible to include node i^+ in the offspring even though this node is added to partial network G'.

Adding random edges and a node, which do not exist in either of the parents, to the partial network is a type of mutation because the offspring can inherit these edges in the solution construction process. In the next step, random temporary costs are assigned to the edges of partial network G'. Uniform random variables between 0 and 0.5 are assigned to edges that exist in both parents, and uniform random variables between 1 and 0 are assigned to all other edges of partial network G'. Therefore, the edges common in both parents are more likely to be selected for the offspring. Finally, the offspring is constructed by determining two edge-disjoint shortest paths for each commodity on partial network G' one commodity at a time. A commodity k is selected randomly and uniformly from unassigned commodities (K'), and two edge-disjoint shortest paths, $\pi_1(k,z)$ and $\pi_2(k,z)$, are found using the Suurballe-Tarjan algorithm (Suurballe and Tarjan, 1984), and these paths are assigned to the offspring. In the example given in Figure 2-(*vi*), two edge-disjoint shortest paths are determined for commodity 1 first and then commodity 2. Next, temporary edge costs are set to 0 for all edges $(i, j) \in \pi_1(k,z) \cup \pi_2(k,z)$ for the assigned commodity k to encourage the paths of the unassigned commodities to use the edges already included in the offspring. In Figure 2-(*vii*), the temporary costs of the edges used by commodity 1 are assigned to zero. These steps continue until the paths for all commodities are assigned to the offspring. Figure 2-(*ix*) illustrates the final offspring constructed from parents *a* and *b* after assigning commodity 2. Note that the offspring is not a complete solution until relays are determined.

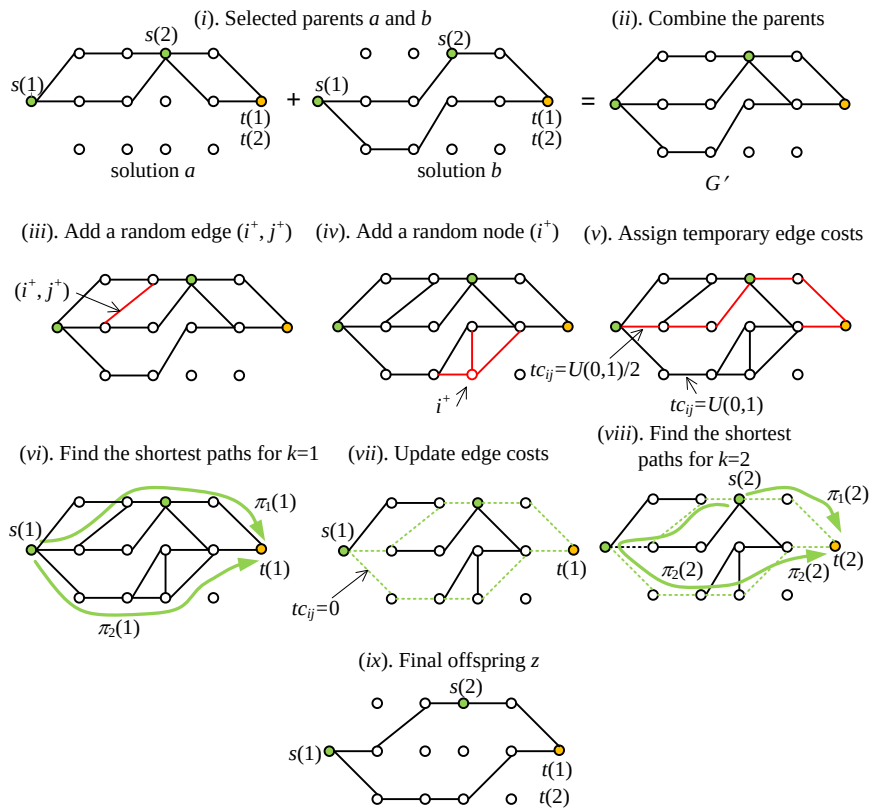

Figure 2. An illustration of Procedure *CrossoverMutation*(a, b) for a problem with two commodities.

3.4 Overall Algorithm

The crossover/mutation operator ensures that there exists at least two edge-disjoint paths for each commodity, and the set covering problem always provides feasible relay assignments. Therefore, the feasibility of solutions is guaranteed in the GH-LH. The pseudo code of the GA-LH is given below. The population is initialized by randomly generating μ solutions. To generate a random solution, a uniform random number is assigned to each edge $(i, j) \in E$ as its temporary cost tc_{ij}, and then a solution is constructed in the similar fashion to the crossover/mutation operator. In each generation of the GA-LH, μ offspring are generated using the crossover/mutation operator. Solutions are ranked based on their objective function values such that the best solution has the rank of 1. For crossover/mutation, two parent solutions (solutions a and b in the procedure below), are randomly selected from the parent population (PP). The first parent is selected using a roulette-wheel selection strategy where solution ranks instead of objective function values are used to calculate selection probabilities. Therefore, the

selection probability of solution a as the first parent is calculated as $2(\mu\text{-}rank(a)+1)/(\mu(\mu+1))$. The second parent is selected randomly and uniformly from the population. After generating μ offspring, parent and offspring solutions are combined together and ranked, and the best ranked μ unique solutions are selected for the next generation. If the population has a multiple copies of a solution, only one of them is actually ranked, and the others are placed at the bottom of the population, and they are compared with the other duplicate solutions. With this approach, the population is discouraged to converge to a single solution. In initial experiments, better results were found by avoiding identical solutions in the population. The GA-LH terminates after $t_{\max}$ generations, and the best solution is evaluated one more time by optimally solving problem Relay(z).

```
Produce GA-LH(μ, t_max){
    Randomly generate μ solutions and add them to population PP;
    Rank solutions in PP;
    for t=1,…,t_max do{
        Set OP=∅;
        for i=1,…,μ do {
            Randomly select solution a from PP with probability 2(μ-rank(a)+1)/(μ(μ+1));
            Randomly and ungormly select solution b from PP such that a≠b;
            z=CrossoverMutation(a,b);
            Determine the relays for chromosome z using the Lagrangian heuristic;
            Add offspring z to OP ;
        } }
        Set PP=PP∪OP;
        Rank solutions in PP;
        Select the best unique μ solutions and remove the others from PP;
    }
    Use Problem Relay(z) to determine the optimal relays for the best solution in PP;
    Return the best solution;
}
```

4. Experimental Results & Discussions

Computational experiments focus on the two contributions of the paper. In the first group of experiments, the performance of the GA-LH is studied with respect to various approaches to solve Problem Relay(z) to demonstrate the contribution of the Lagrangian Heuristic and the proposed hybrid approach. The performance of the proposed hybrid approach depends on how efficiently and effectively set covering sub-problems can be solved. Initial experiments showed that dependencies among commodity paths in the 2ECON-NDPR are much more complicated than the NDPR with single paths. Therefore, solving set covering sub-problems is more challenging in the 2ECON-NDPR than the NDPR with single paths. In the second set of experiments, the performance of the GA-LH is compared to the previous GA as well as three construction heuristics given by Konak et al. (Konak et al., 2009).

In the computational experiments, four different problem groups, with 50, 60, 80, and 160 nodes, were used. For each problem group, the x and y coordinates of nodes were randomly generated from integer numbers between 0 and 100. Distance d_{ij} of edge (i, j) was defined as the Euclidian distance between nodes i and j, and $c_{ij}=d_{ij}$ for each edge (i, j). Three different values of λ, i.e., 30, 35, and 40, were tested. Because edges that are longer than λ should not be considered, the number of edges (M) for each problem depends on λ. For example, the 160 node problem with λ=35 has 3,624 edges and the one with λ=30 has 2,773 edges. In addition, each problem was run with 5, 10 and 15 commodities. The source and the destination nodes of commodities were selected among nodes far apart from one another so that the test problems are not trivial to solve. In Tables 1 and 2, the problems are named using their parameters. For example, the problem with 160 nodes, 10 commodities and λ=35 is represented by (160, 10, 35). The computational experiments were performed on the Research Computing and Cyberinfrastructure Clusters of the Pennsylvania State University. Median CPU times given in the tables are comparable with Intel Xeon E5450 Quad-Core 3.0 GHz with 32GB Memory.

4.1 Performance of the Proposed Hybrid Approach and the Lagrangian Heuristic

To study the contribution of the Lagrangian Heuristic and the proposed hybrid approach, three versions of the GA-LH were developed as described below.

GA-OPT: In this version, CPLEX v12.1 is used to solve problem Relay(z) optimally. Any performance difference between the GA-OPT and GA-LH can be considered as a drawback of the Lagrangian heuristic. If the Lagrangian heuristic is effective, the performances of the GA-OPT and GA-LH should be close.

GA-H: In this version, an efficient heuristic is used to determine relays for a given chromosome. The heuristic solves problem Relay(z) by assigning relays to the nodes that appear in the largest number of constraints (breaking ties by cost) one at a time until all constraints are satisfied. Any performance difference between the GA-H and GA-LH is due to the Lagrangian Heuristic.

GA-E: In this version, relay decision variables are directly encoded in the chromosome using a binary string encoding (y_i=1 if a relay is located at node i, and y_i= 0 otherwise). A uniform crossover operator and a bit flip mutation operator are used to determine relay assignments for offspring. The mutation rate of each bit is set as $1/|V(z)|$ which corresponds to one bit mutation on the average as usually used in the GA literature. After creating the paths of offspring z from two parents a and b using Procedure *CrossoverMutation*(a, b), the relays of offspring z are determined as follows:

for $i \in V(z)$ **do {**
  if $U(0,1) < 0.5$ **then** $y_i(z)=y_i(a)$ **else** $y_i(z)=y_i(b)$
  if $U(0,1) < 1/|V(z)|$ **then** $y_i(z)=1- y_i(z)$
}

Because the uniform crossover does not ensure the feasibility of relay assignments, infeasible solutions are penalized in the GA-E using a penalty function as follows:

$$fitness(z)= \sum_{i \in V(z)} \rho_i y_i(z) + \sum_{(i,j) \in E(z)} c_{ij} x_{ij}(z) + \left(\sum_{i \in V(z)} \rho_i \right) \times \left(\sum_{k \in K} \sum_{p \in \{1,2\}} \sum_{i \in V(z)} \omega_i^{kp} \right)$$

where $\omega_i^{kp} = 1$ if the relay constraint of path $\pi_p(k,z)$ is violated at node i, and $\omega_i^{kp} = 0$ otherwise. The GA-E is not a hybrid approach because both paths and relays of offspring are determined by crossover and mutation. The GA-E represents a pure GA approach to

the problem. Therefore, any performance difference between GA-E and GA-LH is due to the proposed hybrid approach.

Table 1 presents the average and best results found by the four GAs with parameters μ=50 and $t_{\max}$=200 over 30 random replications. Two-sided t-test was used to compare the GA-LH with the GA-OPT, GA-H, and GA-E in pairwise comparisons. In the table, superscript (*) indicates that the average values in the same row are significantly different from the ones found by the GA-LH with p-value < 0.05. Averages without a superscript are not statistically different. As seen in Table 1, the best results were found by the GA-OPT and GA-LH. The performances of the GA-OPT and GA-LH are virtually identical. As expected, the GA-OPT requires significantly longer CPU times than the GA-LH. Although the Lagrangian heuristic is not guaranteed to converge to optimality, the optimality of relay assignments could be established using the lower and upper bounds, and relay assignments turned out to be optimal for majority of solutions.

The performance of the GA-H was on a par with the GA-LH in some problems, but the GA-LH performed much better than the GA-H particularly for the problems where $\lambda \geq 35$ and/or $|K| \geq 15$. These results demonstrate the advantages of the Lagrangian heuristic as opposed to a computationally more efficient heuristic to determine relay assignments. For λ=30, the number of feasible relay assignments is limited. As λ and $|K|$ increase, however, it becomes increasingly challenging to determine near optimal relay assignments. This observation may explain why the GA-H provided good solution for λ=30, but performed poorly for $\lambda \geq 35$. On the other hand, the GA-LH was robust with respect to the problem parameters. The performance of the proposed hybrid approach depends on how efficiently and effectively set covering sub-problems can be solved. It is important to use a robust approach to solve problem Relay(z) because the outcome of problem Relay(z) is used in the GA to guide the search. If the approach to solve problem Relay(z) is biased in some ways, the GA might be misguided. The Lagrangian heuristic was shown to be very robust in this respect. In addition, the Lagrangian Heuristic is computationally efficient. The GA-LH was able to find virtually identical results with GA-OPT in a fraction of time.

The comparisons between the GA-LH and GA-E demonstrate the benefits of the hybrid approach used in this paper. As seen in Table 1, the GA-E performed poorly. It is

challenging to use traditional GA crossover and mutation operators in the 2ECON-NDPR due to dependencies among commodity paths as discussed in Section 3. In addition, path and relay decision variables also depend on one another. Therefore, a meta-heuristic approach that solely depends on direct manipulation of the decision variables as the search strategy is likely to be ineffective for the 2ECON-NDPR. For instance, a GA with conventional crossover and mutation is likely to generate many unacceptable solution structures (e.g., long paths without relays or relays that are not on a path) during the search. The proposed hybrid approach alleviates such problems by solving the problem in two phases as in the GA-LH. The computational results in Table 1 support the paper's main claim that the proposed hybrid approach is promising.

Another interesting result in Table 1 is that the computational effort for solving problem Relay(z) seems to be independent from the problem size in the GA-OPT and a linear function in the GA-LH. This observation justifies solving a very difficult problem in two steps, first determining paths and then assigning relays. Note that the proposed approach is computationally feasible as long as problem Relay(z) can be solved efficiently and effectively. Computationally, the most expensive operation of the GA-LH is the Suurballe-Tarjan algorithm to find the shortest pairs of edge-disjoint paths, which can be performed in $O(M \log_{1+M/N} N)$ (Suurballe and Tarjan, 1984). Since the Suurballe-Tarjan algorithm is run for each commodity, CPU time depends on the number of the commodities routed in the network. Overall, the GA-LH is capable of solving large problems in reasonably short CPU times.

4.1 Comparison of the GA-LH with Earlier Approaches.

The computational results summarized above justify the proposed hybrid approach in terms of computational feasibility. Next, the performance of the GA-LH is compared with three construction heuristics, namely Increasing Order Construction Heuristic (IOCH), Decreasing Order Construction Heuristic (DOCH), Random Order Construction Heuristic (ROCH) and a previous GA given by Konak et al. (Konak et al., 2009). In the previous GA (Konak et al., 2009), the path and relay decision variables are bundled in the GA's chromosome (both paths and relays are searched by the GA), and a special crossover operator is used for relays. Both GAs were run for t_{max}=1000 iterations

and with the population size of 50 as used in (Konak et al., 2009). Again, 30 random replications were performed using the same initial solution in each corresponding replication of both GAs. In the construction heuristics, sub-problems were solved using CPLEX v12.1 with a time limit of 1,800 CPU seconds per sub-problem. The details of the construction heuristics are given in (Konak et al., 2009). In Table 2, the best and average results found by the GA-LH and the previous GA are presented. The standard deviations are also provided to gauge the robustness of the GAs. In addition, the best results found by three construction heuristics are given. As seen in Table 2, the GA-LH outperformed the construction heuristics and the previous GA in all problems. Excluding problem (50, 5, 35), the average results of the GA-LH are statistically better than the ones of the previous GA in all problems. In addition, the GA-LH has a smaller variability within the replications. As seen in the CPU times, the GA scales well. The CPU times of the construction heuristics are in the magnitudes of hours since sub-problems were solved to optimality.

5. Conclusions and Future Research

In this paper, a hybrid approach of a GA and a Lagrangian heuristic is proposed to solve the network design problem with relays under the two-edge connectivity constraint. The computational experiments have shown that proposed approach is computationally efficient and outperforms the earlier approaches to the problem. The proposed specialized crossover/mutation operator can be extended to other survivable network design problems such as problems with mixed-connectivity requirements or node-disjoint network design problems. The capacitated version of the problem is an interesting and challenging further research avenue.

References:

Andre, J., Bonnans, F. & Cornibert, L. (2009) Optimization of capacity expansion planning for gas transportation networks. *European Journal of Operational Research,* 197**,** 1019-1027.

Balakrishnan, A. & Altinkemer, K. (1992) Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing,* 4**,** 192-205.

Beasley, J. E. (1990) A Lagrangian heuristic for set-covering problems. *Naval Research Logistics,* 37**,** 151-64.

Cabral, E. A., Erkut, E., Laporte, G. & Patterson, R. A. (2007) The network design problem with relays. *European Journal of Operational Research,* 180**,** 834-844.

Chen, S., Ljubic, I. & Raghavan, S. (2010) The regenerator location problem. *Networks,* 55**,** 205-220.

Choplin, S. (2001) Virtual path layout in ATM path with given hop count. *Proceedings of the First International Conference on Networking-Part 2.* Colmar, France, Springer-Verlag.

Gouveia, L. (1996) Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research,* 95**,** 178-190.

Gouveia, L. & Magnanti, T. L. (2003) Network flow models for designing diameter-constrained minimum spanning and Steiner trees. *Networks,* 41**,** 159-173.

Gouveia, L., Paias, A. & Sharma, D. (2008) Modeling and solving the rooted distance-constrained minimum spanning tree problem. *Computers and Operations Research,* 35**,** 600-613.

Gouveia, L., Patricio, P., de Sousa, A. F. & Valadas, R. (2003) MPLS over WDM network design with packet level QoS constraints based on ILP models. *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, 30 March-3 April 2003.* Piscataway, NJ, USA, IEEE.

Gouveia, L. & Requejo, C. (2001) A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research,* 132**,** 539-552.

Gouveia, L., Simonetti, L. & Uchoa, E. (2011) Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming,* 128**,** 123-148.

Kabirian, A. & Hemmati, M. R. (2007) A strategic planning model for natural gas transmission networks. *Energy Policy,* 35**,** 5656-5670.

Konak, A. (2012a) Network Design Problem with Relays: A Genetic Algorithm with a Path-Based Crossover and a Set Covering Formulation. *European Journal of Operational Research,* 218**,** 829-837

Konak, A. (2012b) Network Design Problem with Relays: A Genetic Algorithm with a Path-Based Crossover and a Set Covering Formulation *European Journal of Operational Research,* 218**,** 829-827.

Konak, A., Kulturel-Konak, S. & Smith, A. (2009) Two-Edge Disjoint Survivable Network Design Problem with Relays. IN CHINNECK, J. W., KRISTJANSSON, B. & SALTZMAN, M. (Eds.) *Operations Research and Cyber-Infrastructure (Operations Research/Computer Science Interfaces Series Vol. 47).* New York, USA., Springer.

Konak, A. & Smith, A. E. (2004) Capacitated Network Design Considering Survivability:

An Evolutionary Approach. *Journal of Engineering Optimization,* 36**,** 189-205.

Kulturel-Konak, S. & Konak, A. (2008) A Local Search Hybrid Genetic Algorithm Approach to the Network Design Problem with Relay Stations. IN RAGHAVAN, S., GOLDEN, B. L. & WASIL, E. (Eds.) *Telecommunications Modeling, Policy, and Technology.* New York, Springer.

Kwangil, L. & Shayman, M. A. (2005) Optical network design with optical constraints in IP/WDM networks. *IEICE Transactions on Communications,* E88-B**,** 1898-1905.

LeBlanc, L. & Reddoch, R. (1990) Reliable link topology/capacity design and routing in backbone telecommunication networks. *First ORSA Telecommunications SIG Conference.*

Leitner, M., Ruthmair, M. & Raidl, G. R. (2011) Stabilized Branch-and-price for the Rooted Delay-constrained Steiner Tree Problem. *Network Optimization. 5th International Conference, INOC 2011, 13-16 June 2011.* Berlin, Germany, Springer-Verlag.

Magnanti, T. L., Mirchandani, P. & Vachani, R. (1995) Modeling and solving the two-facility capacitated network loading problem. *Operations Research,* 43**,** 142-157.

Monma, C. L. & Shallcross, D. F. (1989) Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research,* 37**,** 531-541.

Mukherjee, B. (2000) WDM optical communication networks: Progress and challenges. *IEEE Journal on Selected Areas in Communications* 18**,** 1810-1824.

Ramamurthy, B., Iness, J. & Mukherjee, B. (1997) Minimizing the number of optical amplifiers needed to support a multi-wavelength optical LAN/MAN. *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies.* Kobe, Japan Published by the IEEE Computer Society.

Ramamurthy, B., Iness, J. & Mukherjee, B. (1998) Optimizing amplifier placements in a multiwavelength optical LAN/MAN: the unequally powered wavelengths case. *IEEE/ACM Transactions on Networking (TON),* 6**,** 755-767.

Randall, M., McMahon, G. & Sugden, S. (2002) A simulated annealing approach to communication network design. *Journal of Combinatorial Optimization,* 6**,** 55-65.

Ruthmair, M. & Raidl, G. R. (2011) A Layered Graph Model and an Adaptive Layers Framework to Solve Delay-Constrained Minimum Tree Problems. *Integer Programming and Combinatoral Optimization. 15th International Conference, IPCO 2011, 15-17 June 2011*. Berlin, Germany, Springer Verlag.

Suurballe, J. W. & Tarjan, R. E. (1984) A quick method for finding shortest pairs of disjoint paths. *Networks,* 14**,** 325-336.

Tabkhi, F., Pibouleau, L., Azzaro-Pantel, C. & Domenech, S. (2009) Total cost minimization of a high-pressure natural gas network. *Journal of Energy Resources Technology,* 131**,** 043002 (12 pp.).

Taylor, G. D., Whicker, G. L. & Usher, J. S. (2001) Multi-zone dispatching in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review,* 37**,** 375-390.

Voss, S. (1999) The Steiner tree problem with hop constraints. *Annals of Operations Research,* 86**,** 321-345.

Winters, J. H., Gitlin, R. D. & Kasturia, S. (1993) Reducing the effects of transmission impairments in digital fiber optic systems. *IEEE Communications Magazine* 31**,** 68-76.

Yetginer, E. & Karasan, E. (2003) Regenerator placement and traffic engineering with restoration in GMPLS networks. *Photonic Network Communications,* 6**,** 139-149.

Table 1. Performance of the GA-LH with respect to other approaches to solve Problem Relay(z)

	GA-OPT			**GA-H**			**GA-E**			**GA-LH**		
Problem Name (N, K, λ)	*Average*	*Best*	*Median CPU Sec*	*Average*	*Best*	*Median CPU Sec.*	*Average*	*Best*	*Median CPU Sec.*	*Average*	*Best*	*Median CPU Sec*
(60, 5, 30)	797.3	786.0	8617	794.6	786.0	8	1246.8*	1006.0	6	797.3	786.0	7
(60, 5, 35)	681.8	647.0	9029	686.9	647.0	7	1157.6*	791.9	6	681.8	647.0	7
(60, 5, 40)	598.5	576.9	6487	596.6	584.0	5	1044.2*	817.0	5	599.2	576.9	5
(60, 10,30)	1111.9	1032.0	5932	1124.9	1062.0	10	1967.4*	1338.3	10	1113.1	1032.0	10
(60,10,35)	899.1	830.0	1898	920.7*	871.0	10	1932.6*	1405.9	10	899.1	830.0	10
(60,10,40)	859.2	805.2	8027	881.1*	840.0	8	1879.4*	1509.4	10	859.2	805.2	9
(60,15,30)	1517.4	1411.0	8935	1533.8	1483.0	15	2477.0*	2247.2	14	1517.5	1411.0	16
(60,15,35)	1285.0	1145.0	11046	1350.4*	1217.0	16	2572.3*	2386.9	14	1285.0	1145.0	18
(60,15,40)	1145.7	1037.3	8158	1225.9*	1147.0	14	2578.3*	2231.8	13	1145.7	1037.3	17
(80,5,30)	617.4	611.0	9463	620.2	612.0	10	1099.8*	920.7	9	617.4	611.0	9
(80,5,35)	531.2	522.0	8843	531.0	528.0	9	1072.7*	885.5	8	531.2	522.0	9
(80,5,40)	496.0	482.6	6664	495.6	483.0	11	1133.8*	885.2	8	496.0	483.0	12
(80,10,30)	723.6	706.0	6739	728.8	714.0	14	1476.1*	1084.5	15	723.6	706.0	14
(80,10,35)	639.6	625.0	6735	643.5	625.0	15	1546.0*	1267.6	13	639.6	625.0	18
(80,10,40)	630.8	612.0	7202	636.5*	614.0	16	1645.2*	1284.1	10	630.7	612.0	23
(80,15,30)	864.5	848.0	9841	870.8	844.0	20	1803.5*	1490.8	17	864.7	848.0	27
(80,15,35)	859.3	832.0	8973	878.1*	828.0	21	1941.9*	1712.4	25	858.6	829.0	57
(80,15,40)	863.6	808.1	8210	910.2*	852.0	24	2126.3*	1888.8	25	867.2	808.1	46
(160,5,30)	463.4	426.0	10331	454.8*	421.0	21	1173.4*	1030.9	21	463.4	426.0	19
(160,5,35)	426.0	411.0	8623	428.1	413.0	21	1177.1*	1048.9	22	426.0	411.0	20
(160,5,40)	438.3	409.1	8926	430.4	410.0	19	1207.9*	1051.1	22	438.2	409.1	19
(160,10,30)	616.3	592.0	7367	629.1*	606.0	34	1903.8*	1655.7	40	616.3	592.0	38
(160,10,35)	581.0	569.0	11862	590.5*	567.0	36	2043.8*	1699.9	46	581.0	569.0	37
(160,10,40)	607.0	584.3	9471	633.1*	573.0	34	2117.3*	1502.3	38	606.8	584.3	44
(160,15,30)	761.8	723.0	8498	771.4	737.0	49	2545.6*	2362.3	46	760.9	723.0	77
(160,15,35)	768.2	715.0	7782	813.2*	734.0	53	2759.1*	2505.0	52	767.7	700.0	78
(160,15,40)	795.6	719.9	7893	851.5*	775.0	54	2959.0*	2591.2	58	795.6	719.9	77

(*) indicates that the average values in the same row are significantly different from the ones found by the GA-LH with p-value < 0.05.

Table 2. Comparison of the GA-LH with the previous approaches (bold characters are used for the best solutions)

		GA-LH				Previous GA (Konak et al., 2009)			Construction Heuristics
Problem (N, K, λ)	*M*	*Best*	*Average*	*Standard Deviation*	*Median CPU Seconds*	*Best*	*Average*	*Standard Deviation*	*Best*
(50, 5, 30)	279	**593.53**	594.48	3.55	31	**594.23**	610.07*	11.82	652.93
(50, 5, 35)	372	**491.39**	496.40	13.00	30	**491.39**	498.59	8.46	513.72
(50, 10, 30)	279	**931.91**	962.81	18.26	137	1122.37	1193.49*	28.47	1078.99
(50, 10, 35)	372	**728.25**	790.64	39.78	50	955.73	1075.64*	41.91	901.52
(50, 15, 30)	279	**1169.65**	1193.55	22.24	106	1564.37	1636.21*	33.63	1281.09
(50, 15, 35)	372	**1061.25**	1130.78	40.56	80	1440.82	1602.76*	70.79	1247.37
(60, 5, 30)	305	**785.84**	785.84	.00	37	817.19	867.34*	16.18	911.28
(60, 5, 35)	412	**641.14**	668.85	18.67	36	694.37	737.76*	21.68	817.44
(60, 10, 30)	305	**1020.66**	1059.45	25.53	54	1234.81	1294.49*	26.74	1217.44
(60, 10, 35)	412	**797.35**	825.26	19.39	53	1066.91	1121.10*	25.89	1044.39
(60, 15, 30)	305	**1298.89**	1396.23	45.81	82	1655.45	1718.87*	35.22	1419.52
(60, 15, 35)	641	**989.21**	1052.07	43.98	110	1502.86	1607.73*	36.82	1247.53
(80, 5, 30)	853	**608.55**	612.93	3.85	56	616.39	626.36*	4.42	753.47
(80, 5, 35)	641	**520.58**	527.23	2.51	44	528.17	540.00*	7.15	701.87
(80, 10, 30)	853	**688.30**	703.79	8.49	80	757.77	803.10*	15.00	1025.52
(80, 10, 35)	641	**607.63**	621.96	11.76	138	693.76	763.67*	20.20	1004.91
(80, 15, 30)	853	**809.63**	836.78	10.43	166	1005.32	1072.50*	25.31	1265.06
(80, 15, 35)	2773	**780.42**	803.46	11.75	594	1030.27	1080.49*	25.95	1169.78
(160, 5, 30)	3624	**420.70**	434.70	11.71	99	457.55	493.69*	17.79	598.59
(160, 5, 35)	2773	**410.80**	416.45	9.11	104	450.92	482.95*	15.48	568.85
(160, 10, 30)	3624	**558.03**	576.11	8.99	212	693.16	734.82*	18.92	879.84
(160, 10, 35)	2773	**551.09**	561.48	4.24	200	696.40	745.80*	20.13	908.06
(160, 15, 30)	3624	**665.77**	699.78	21.06	496	919.28	1008.33*	30.33	1100.38
(160, 15, 35)	2773	**659.94**	682.44	17.84	761	956.94	1039.12*	31.61	1113.93

(*) indicates that the average is significantly different from the one found by the GA-LH with *p*-value < 0.05.