The Generalized Regenerator Location Problem

Si Chen* Ivana Ljubić⋆ S. Raghavan⋄

*College of Business and Public Affairs, Murray State University
Murray, KY 42071, USA

⋆Faculty of Business, Economics, and Statistics, University of Vienna
Brünnerstr. 72, 1210 Vienna, Austria

⋄Smith School of Business and Institute for Systems Research
University of Maryland, College Park, MD 20742, USA

Abstract

In an optical network a signal can only travel a maximum distance of d_{max} before its quality deteriorates to the point that it must be regenerated by installing regenerators at nodes of the network. As the cost of a regenerator is high we wish to deploy as few regenerators as possible in the network, while ensuring that nodes can communicate with each other. In the generalized regenerator location problem (GRLP), that we introduce in this paper, we are given a set S that corresponds to candidate locations for regenerators, and a set T that corresponds to the set of nodes that must communicate with each other. Together S and T make up the node set N of the network. If $S = T = N$, we obtain the regenerator location problem (RLP) which we have studied previously. We show that the GRLP is NP-Complete. We then devise reduction procedures, a heuristic, and improvement procedures for the GRLP. We also establish a correspondence between the directed Steiner forest problem (DSF) and the GRLP. Using this fact we model the GRLP as a DSF problem. We provide some preliminary computational results of our heuristic and MIP formulation for the GRLP problem.

Keywords: *optical network design, heuristics, directed steiner forest.*

1 Introduction

An optical network provides higher capacity and reduced costs for new applications such as the Internet, video and multimedia interaction, and advanced digital services. A critical problem that arises in the design of optical networks relates to the placement of regenerators. A signal can only travel a maximum distance of d_{max} before its quality deteriorates and needs to be regenerated. To accomplish this regenerators may be installed at nodes of the network. As the cost of regenerators is very high we wish to deploy as few regenerators as possible, while ensuring all nodes can communicate with each other (i.e., send a signal to each other). In this context within this paper we introduce and discuss the following problem that we call the generalized regenerator location problem (GRLP).

Mathematically, the generalized regenerator location problem (GRLP) can be described as follows. Given a network $G = \{N = S \cup T; F; D; d_{max}\}$, where N is the set of nodes, S and T are two subsets of N (S is the set of location nodes where regenerators can be placed, and T is the set of terminal nodes that must communicate with each other), F is the set of edges, and D is the associated distance matrix of edges, and a maximum distance of d_{max} that determines how far a signal can traverse before its quality

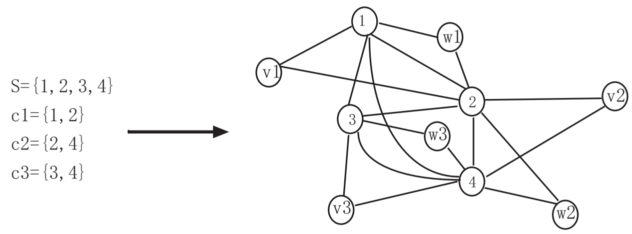


Figure 1: *Transform a HSP to GRLP*

deteriorates and needs to be regenerated. Determine a minimum cardinality subset of nodes $L \subseteq S$ such that for every pair of nodes in T there exists a path in G with the property that there is no subpath (i.e., a subsequence of edges on the path) with length $> d_{max}$ without internal regenerators (i.e, we do not include the end points of the subpath). When $S = T = N$ we obtain a special case of the GRLP problem that we refer to as the regenerator location problem (RLP). Here, all nodes are terminal nodes as well as candidate locations for placement of regenerators. We introduced the RLP in [1] and presented several heuristics and a branch-and-cut approach for it. In particular, we established a correspondence between the RLP and the maximum leaf spanning tree problem. In this paper, we focus on the GRLP problem. In this situation, the correspondence to the max leaf spanning tree problem does not hold. Consequently, the heuristics and exact solution procedures we developed for the RLP do not apply to the GRLP problem.

The rest of this extended abstract will be organized as follows: Section 2 proves that GRLP is NP-Complete; Section 3 describes our heuristic for solving GRLP; Section 4 discusses a MIP formulation for GRLP; and Section 5 presents some preliminary computational results from our heuristics and compares them with the optimal solutions obtained from the MIP formulation.

2 NP-Completeness of GRLP

In this section we prove that GRLP is NP-Complete.

Theorem 1. *The generalized regenerator location problem is NP-complete.*

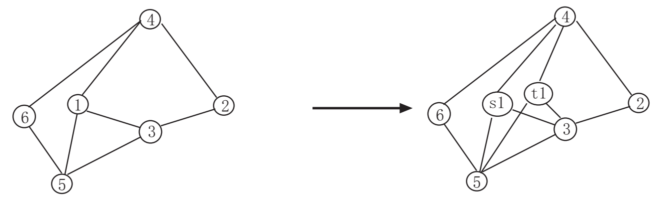
Proof. We consider a special case of Hitting Set Problem (HSP), which is stated as follows ([4]).
Instance: Collection C of subsets of a finite set S, where $|c| = 2$ for all $c \in C$, positive integer $K \leq |S|$.
Question: Is there a subset $S' \subseteq S$ with $|S'| \leq K$ such that S' contains at least one element from each subset in C?

We now construct the corresponding instance of the GRLP. Create a node for every $s \in S$. Connect all nodes in S. For every $c = \{c_i^1, c_i^2\}$ in C, create a pair of nodes v_i and w_i. Connect v_i to node c_i^1 and c_i^2 (the elements of c_i), and w_i, c_i^1 and c_i^2. Set the length of all edges in the resulting graph to d_{max}. Let $T = \{v_i, w_i \, |i = 1, 2, ..., |C|\}$.

The question is whether there is a feasible solution (a set of nodes $L \subseteq S$ where we place regenerators) to the GRLP problem with cardinality less than or equal to K. It is easy to observe that an instance of HSP has a "yes" answer if and only if the corresponding GRLP has a "yes" answer. As this is a polynomial transformation, the decision version of the GRLP is NP-complete. □

3 Proposed Heuristic

In this section, we discuss a heuristic procedure for the GRLP. Before we do, it is useful to consider the following graph transformation of GRLP.



(a) Graph M with S = {1, 3, 6} and T = {1, 2 ,4 ,5}

(b) Graph B with SB = {s1, 3, 6} and TB = {t1, 2 ,4 ,5}

Figure 2: The transformed graph B from M

Given a graph $G = \{N = S \cup T; F; D; d_{max}\}$, apply the all pairs shortest path algorithm. Replace edge lengths by the shortest path distance. If the edge length is less than or equal to d_{max} then keep the edge, and if the edge is greater than d_{max} delete the edge. Denote this new graph as M (with node set N and edge set E). We define the subgraphs induced by node sets S and T of M as $M_S = \{S, E(S)\}$ and $M_T = \{T, E(T)\}$), where $E(S)$ is the set of edges with both end points in S and $E(T)$ is the set of edge with both end points in T. Every node pair that is not connected by an edge in M_T requires regenerators to communicate. We call such node pairs "not directly connected" or NDC node pairs. It suffices to consider the GRLP problem on the transformed graph M and determine the minimum cardinality subset of nodes L, such that for the NDC node pairs in M there exists a path with regenerators at all internal nodes on the path.

Lemma 1. *Without loss of generality, we can assume that in any GRLP instance $S \cap T = \emptyset$.*

Proof. To show this, assume that we are given an instance M such that $S \cap T \neq \emptyset$. We now show the transformation from M into a new graph B where the set of potential regenerator locations and the set of terminal nodes are disjoint.

- For every node i in M
  1. if $i \in T \cap S$, create an s-node s_i and a t-node t_i, otherwise
  2. if $i \in S \setminus T \cap S$ or $i \in T \setminus T \cap S$, add i to B.
- For every edge (i, j) in M
  1. if $i, j \in T \cap S$, create four edges (s_i, s_j), (t_i, t_j), (s_i, t_j) and (s_j, t_i), add them to B, otherwise
  2. if exactly one end point, say $i \in T \cap S$, create two edges: (s_i, j) and (t_i, j) in B, otherwise
  3. Add (i, j) to B.

Let SB denote the set of S nodes in B and TB the set of T nodes in B. Figure 2 illustrates the transformation from M with $S = \{1, 3, 6\}$ and $T = \{1, 2, 4, 5\}$ to B with $SB = \{s1, 3, 6\}$ and $TB = \{t1, 2, 4, 5\}$. Observe that any feasible solution on M can be transformed into an equivalent one on B of the same cardinality and vice versa. □

Given the previous observation, in the rest of the paper we assume that S and T are disjoint. We call nodes in S, s nodes; and nodes in T, t nodes.

We now discuss the issue of checking feasibility of the GRLP problem. We first consider the two disjoint induced subgraphs in B, $B_S = \{S, E(S)\}$ and $B_T = \{T, E(T)\}$. The former contains only S nodes and edges between them, while the latter contains only T nodes and edges between them. An S component is a connected subgraph of B_S. Observe that for an instance of GRLP to be feasible the two end points of every NDC node pair in B_T must be connected to at least one common S component. Figure 3 identifies the S components of B in Figure 2 and replaces them with super nodes $S1$ and $S2$. Observe that the GRLP problem on B is feasible since the two end points of every NDC node pair are

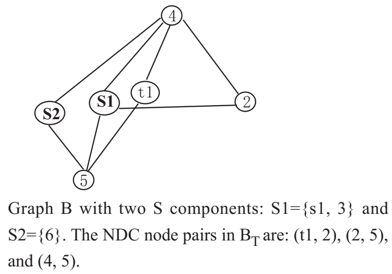


Figure 3: S components of B and NDC node pairs in B_T

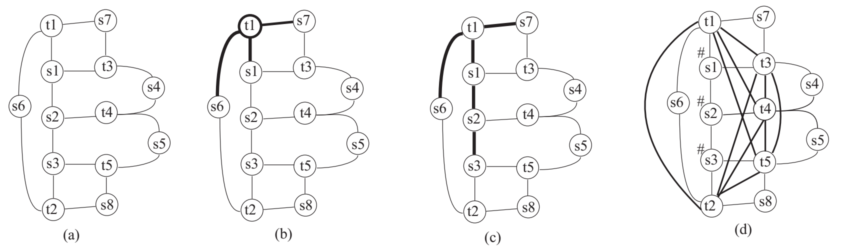


Figure 4: Illustration of Heuristic GH for GRLP

connected to at least one common S component. It is easy to show that the running time for checking feasibility is $O(|N|^3)$ (details are left for the full length paper).

We now discuss how preprocessing can be used to reduce the problem size and fix regenerators in the solution. Observe that (1) if t-node i in B is only connected to one other s-node j, every feasible solution must include a regenerator deployed at node j. Once a regenerator is deployed at node j, node i can be eliminated from B; (2) if a t-node i is connected to all other t-nodes, removal of i from B will not change the problem; (3) if all the neighbors of a s-node i are connected to each other, removal of i from B will not change the problem; (4) if the two end points of an NDC node pair (i, k) are connected to only one common S-component, and $i(k)$ is connected to only one s-node j from this S-component, any feasible solution must deploy a regenerator at j.

Heuristic GH: This heuristic focuses on S-components. First, it selects the t-node i that has the lowest t-degree. Next, for each S-component S_j connected to i, it identifies the subset of nodes $L_{S_{max}}$ with the largest value of $w_{L_{S_j}}$, which is defines as the ratio of the number of NDC node pairs that can be reduced after we deploy a regenerator at every node in $L_{S_j})/|L_{S_j}|$. It then adds regenerators to all nodes in $L_{S_{max}}$. The steps are repeated until all t nodes are connected.

We use Figure 4 to illustrate the heuristic. In (a) all t-nodes have t-degree zero, GH arbitrarily chooses node t_1. It is connected to three s-components: $S1 = \{s_6\}$, $S2 = \{s_1, s_2, s_3\}$ and $S2 = \{s_7\}$. Since $S1$ and $S3$ contain only single node and $S2$ is a spanning tree, $L_{S_1} = S1$, $L_{S_2} = S2$ and $L_{S_3} = S3$. Their weights are $w_{L_{S_1}} = 1$, $w_{L_{S_2}} = \frac{10}{3}$ and $w_{L_{S_3}} = 1$, respectively. As $w_{L_{S_2}} = \frac{10}{3}$ is the largest, we choose to deploy regenerators at every node in L_{S_2}. As shown in Figure 4(d) we now have a feasible solution.

After applying our heuristic GH we apply the following post-optimizer (to improve upon the heuristic solution) which consists of two subroutines: REMOVE and 2-for-1 SWITCH. REMOVE is applied to each S-component housing at least one regenerator location. It first tries to remove the entire S-component (i.e., remove all regenerators that are deployed at the s-nodes in the S-component). If the S-component cannot be completely removed, it then tries to remove some subset of the component. Finally, it tries to remove each regenerator location in the component one at a time. The 2-for-1 SWITCH modifies the a solution by swapping two s-nodes in the current solution with one that is not. It is first applied within each S-component and then in between two S-components.

4 Solving the GRLP to Optimality

In this section we describe a Multi-Commodity Flow Formulation for GRLP. Before we do so, it is useful to consider the following graph transformation.

Transforming the GRLP into the Directed Steiner Forest Problem: Given a directed graph $H = (V, A)$ with nonnegative arc costs and a collection $D \subseteq V \times V$ of ordered pairs, in the *Directed Steiner Forest* problem (DSF) we are looking for the minimum cost subgraph of H that contains a directed path for every pair of nodes in D. The DSF problem has not been investigated in the literature very intensively. In fact, there are only few approximation algorithms known for the DSF. Dodis and Khanna [2] have shown that the DSF cannot be approximated within $O(2^{\log^{1-\epsilon} n})$ for any fixed $\epsilon > 0$, unless NP-hard problems can be solved in quasi-polynomial time. The currently best known approximation ratio for the DSF is $O(n^{4/5+\epsilon})$ and it was shown by Feldman et al. [3].

Despite the name that seems to suggest that any solution of the DSF must be a forest, this is in general not always the case. In fact, the problem is a union of Steiner arborescences. In particular, consider the set D, and denote with $V_O = \{i \in B \mid \exists k \text{ s.t. } (i,k) \in D\}$, the set of all origins, and with $V_D = \{i \in B \mid \exists k \text{s.t. } (k,i) \in D\}$ the set of all destinations of the DSF. For each origin node $r \in V_O$, denote with $V_D(r) = \{i \in V_D \mid (r,i) \in D\}$ the set of its destination nodes. The DSF problem can then be viewed as the union of Steiner arborescences each rooted at a node $r \in V_O$ and connecting terminals $i \in V_D(r)$. Therefore an optimal solution on H is not always cycle-free. In a common example in which $D = V \times V$ and all costs are equal to one, the optimal solution is given by an oriented Hamiltonian cycle.

We now show how the generalized regenerator location problem can be transformed into the DSF problem, and then we propose a flow based formulation for it. We first transform the graph B into a directed graph H as follows:

- For every node i in B we create two nodes i_1 and i_2 in H, and, if $i \in S$, we add an arc $< i_1, i_2 >$ in H with cost $c_{i_1 i_2} = 1$. We denote the set of nodes with index 1 in H as V_1, the set of nodes with index 2 in H as V_2, and the set of all nodes in H as $V = V_1 \cup V_2$. This definition applies to T nodes as well, so we distinguish between S_1, S_2, T_1 and T_2 in the node sets. The nodes from T_2 will act as sources, the nodes from T_1 as destinations, whereas the nodes from $S_1 \cup S_2$ will help us in modeling the node-variables of S. We refer to the set of unit cost arcs as A_1.
- For every edge (i, j) in B, we add two arcs $< i_2, j_1 >$ and $< j_2, i_1 >$ to H with costs $c_{i_2 j_1} = c_{j_2 i_1} = 0$. We denote these arcs by A_2 and we set $A = A_1 \cup A_2$.

Consider the set of NDC node pairs in B. Each pair (u, v) of NDC nodes in B, corresponds to an ordered pair $(u_2, v_1) \in D$ in the graph H. We are looking for a minimum cost subgraph of H that contains a directed path for every pair of nodes in D.

Multi-Commodity Flow Formulation for the DSF: For every pair (i, j) in D create a commodity with origin node i, destination node j, and a requirement of 1. Let K denote the set of all commodities. It is easy to show that by solving the following uncapacitated fixed charge network design problem on H, we find the optimal solution of the DSF problem.

$$(MCF) \qquad \min \sum_{<i,j>\in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \qquad \sum_{<i,j>\in A} f_{ij}^k - \sum_{<j,i>\in A} f_{ji}^k = \begin{cases} -1, & i = D(k) \\ 1, & i = O(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V,\ \forall k \in K \tag{2}$$

$$0 \leq f_{ij}^k \leq x_{ij} \qquad \forall < i,j >\in A\ \forall k \in D \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad \forall < i,j >\in A \tag{4}$$

				GH		MIP		
n	nS	nT	m	NF	RT(sec)	NF	RT	Diff
30	10	20	75	3	0.00	3	0.20	0
30	10	20	126	5	0.00	5	0.60	0
50	10	40	375	5	0.00	4	4.70	1
70	10	60	831	5	0.00	5	37.50	0
90	10	80	1439	6	0.00	6	102.50	0
110	10	100	2229	6	0.00	6	194.40	0
70	20	50	1219	11	0.00	10	67.80	1
80	20	60	1724	12	0.00	12	146.00	0
100	20	80	3073	14	0.00	-	-	-
50	30	20	178	6	0.00	5	9.70	1
70	30	40	713	9	0.00	9	503.90	0
80	30	50	1144	11	0.00	-	-	-
90	30	60	1620	10	0.00	-	-	-
100	30	70	2092	10	0.00	-	-	-
100	30	70	2074	10	0.00	-	-	-
110	30	80	2790	12	0.00	-	-	-
80	40	40	762	12	0.00	10	808.00	2
100	40	60	1755	14	0.00	-	-	-
100	40	60	1724	14	0.00	-	-	-
120	40	80	3073	17	0.00	-	-	-

Table 1: Computational Results for GH v.s MIP

5 Computational Results

We now briefly discuss our computational experience with our heuristic. We compared our heuristic with the exact solution obtained by applying CPLEX 11.0 (a commercial MIP solver) to the Multi-Commodity flow formulation. All results were obtained on a 2GHz PC with 3GB memory running under Windows XP. We generated 20 instances where the underline graphs have all edges longer than d_{max} eliminated and only pure s and t nodes. All instances are also generated so that they are feasible.

We now discuss the performance of the heuristic GH. Table 1 summarizes our the results. "n" is the number of nodes; "nS" is the number of s-nodes; "nT" is the number of terminal nodes; "m" is the number of NDC node pairs; "NF" is the number of regenerators; "RT" is the running time in seconds; "Diff" is the difference between the heuristic solution and the optimal solution. The empty cells indicate that the MIP solver cannot produce the optimal solutions due to lack of memory.

Observe that CPLEX succeeds in finding the optimal solutions for 11 instances, for which our heuristic finds the optimal solutions for 7 but at a much faster pace. Diff ranges from 0 to 2. Based on these limited computational results, our heuristic appears to have good potential for application to large-sized problems. As part of the future work we wish to improve the performance of our heuristic and also find tighter lower bounds that can help us better evaluate the performance of our heuristic.

References

[1] S. Chen, I. Ljubić, and S. Raghavan. The regenerator location problem. *Networks*, 2008. to appear.

[2] Y. Dodis and S. Khanna. Designing networks with bounded pairwise distance. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 750–759. 1999.

[3] Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximating algorithms for directed Steiner forest. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 922–931. 2009.

[4] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. Freeman,NY, 1979.