Clique and Independent Set Based GRASP Approaches for the Regenerator Location Problem

Peter Jahrmann, Günther R. Raidl[1]

Institute of Computer Graphics and Algorithms
Vienna University of Technology
Favoritenstraße 9/1861, 1040 Vienna, Austria
p.jahrmann@net1220.at, raidl@ads.tuwien.ac.at

Abstract

We consider the Regenerator Location Problem (RLP) in optical fibre communication networks: As optical signals deteriorate in dependence of the distance from the source, regenerator devices need to be installed at a subset of the network nodes so that no segment of any communication path without an intermediate regenerator exceeds an allowed maximum length. The objective is to place a smallest possible number of regenerators in order to satisfy this condition. We propose two new construction heuristics based on identifying and exploiting cliques and independent sets of the network graph. These strategies are further extended to Greedy Randomized Adaptive Search Procedures (GRASP) that also include new destroy and recreate local search phases. Excellent results are obtained in an experimental comparison with a previously described GRASP.

1 Introduction

Over the past twenty years the internet has become a key technology for the modern information society. It has sustainably influenced our every day life, having a strong impact on the way how we behave, communicate and think. This formative potential is integrated in numerous applications and services like video sharing, online gaming, voice over IP, mobile internet and social media in its full extent. As a consequence an exponential growth of traffic for telecommunication networks is obligatory. To satisfy the needs of growing demands for these underlying networks, optical fibre technology has proven to be the best choice. Compared to copper wire it offers substantially higher bandwidths, less signal degradation and is cheaper in acquisition. These characteristics enable optical networks to be the state-of-the-art of high speed data transmission.

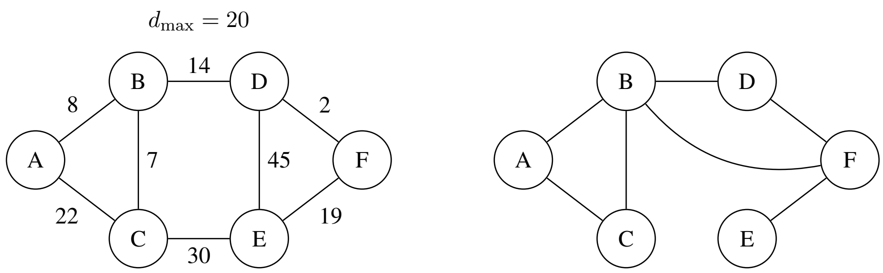
In this article we address the *Regenerator Location Problem* (RLP), which deals with the geographical transmission of information in optical networks. Despite the fact that optical fibre has less signal degradation than copper wire, the strength of an optical signal still deteriorates as it gets farther from the source. The distance a signal can travel without too much loss of quality is limited by a certain value. For larger distances optical signal regenerators need to be installed throughout the network at a subset of the nodes. Since regenerators are considered to be expensive the RLP aims to deploy as few such devices as possible, while ensuring that all nodes in the network can communicate with each other.

After introducing the problem more formally in the next section and reviewing the literature, we present two new concepts for a heuristic solution construction based on exploiting cliques and independent sets, respectively. These strategies are then further extended to different variants of a Greedy Randomized Adaptive Search Procedure (GRASP). Experimental comparisons indicate that a hybrid GRASP variant in which clique as well as independent set based mechanisms are used together work best and yield new state-of-the-art results.

2 Regenerator Location Problem and Communication Graph

An instance of the RLP is given by an undirected, connected and weighted graph $G = (V, E, d)$, where V is the set of nodes, $E \subseteq V^2$ is the set of edges and $d_{i,j} \geq 0$ is the length of each edges $(i, j) \in E$.

[1]The second author is partly funded by the Vienna Science and Technology Fund (WWTF) through project ICT10-027.



original graph G communication graph M

Figure 1: Exemplary RLP instance and its corresponding communication graph.

Moreover a constant $d_{\max}$ specifies the maximum distance a signal can travel without the need of regeneration. A solution to the RLP is a subset of nodes $R \subseteq V$ where regenerators are to be installed. Such a solution is feasible if each pair of nodes $u, v \in V$ can communicate, i.e., either the length of the shortest path does not exceed $d_{\max}$, or a path exists with regenerator nodes placed on it in such a way that no subpath without an intermediate regenerator node exceeds the limit $d_{\max}$. Among all feasible solutions, we are interested in a cheapest one, i.e., we minimize the number of selected regenerators $|R|$.

Chen et al. [3] introduced the *communication graph* for the RLP, which is obtained by the following transformation from G and provides a more convenient and efficient way to check feasibility:

1. Initialize the communication graph $M = (V, E')$ with V being the same set of nodes as in G and E' being an empty edge set.

2. Utilizing an all-pair shortest path algorithm, an edge $(u, v) \in V \times V$ is added to E', if and only if the shortest path length from u to v does not exceed $d_{\max}$, i.e., if no regenerators are required on this path.

The complementary set of E', i.e., $\overline{E'} = V \times V \setminus E'$, contains all *not directly connected node pairs* (NDC pairs), which must be connected via regenerators to obtain a feasible solution to the RLP. Since the communication graph only consists of edges having lengths not exceeding $d_{\max}$, the feasibility of a solution can now also be defined in a simpler way: For each pair of nodes $u, v \in V$, $u \neq v$, there must exist a path in M from u to v which either corresponds to a single edge or has regenerators placed on all its inner nodes. Figure 1 shows an example for an RLP instance and its corresponding communication graph.

Having a closer look at the example the set of NDC pairs in the communication graph is $\overline{E'} = \{(\mathrm{A}, \mathrm{D}), (\mathrm{A}, \mathrm{E}), (\mathrm{A}, \mathrm{F}), (\mathrm{B}, \mathrm{E}), (\mathrm{C}, \mathrm{D}), (\mathrm{C}, \mathrm{E}), (\mathrm{C}, \mathrm{F}), (\mathrm{D}, \mathrm{E})\}$. To connect the NDC pair (A, D) a regenerator placed at node B is necessary. Since this regenerator also connects the NDC pairs (C,D), (A,F) and (C,F) only the pairs $\{(\mathrm{A}, \mathrm{E}), (\mathrm{B}, \mathrm{E}), (\mathrm{C}, \mathrm{E}), (\mathrm{D}, \mathrm{E})\}$ remain. Another regenerator placed at node F connects all NDC pairs left and for this reason an optimal solution for this example is $R = \{\mathrm{B}, \mathrm{F}\}$.

A successive preprocessing of the communication graph frequently reduces the number of nodes, fixes nodes on which regenerators must be placed, adds further edges, and may divide the original graph in several independently solvable communication graphs [3]:

- Assuming $|V| \geq 3$, for each node $u \in V$ having degree one in M, we must install a regenerator at its single neighbor since otherwise u cannot communicate with the remaining node(s); node u can then be removed as it will be able to communicate with all other nodes if and only if its neighbor is able to do so.

- More generally, regenerator nodes need to be installed at any *cut point* in M. A cut point is a node whose removal would disconnect the graph into separated components. All these cut points must be regenerator nodes as otherwise no communication would be possible between the different components. The cut points of a graph can be identified by a classic algorithm for determining its biconnected components, such as the algorithm from Hopfroft and Tarjan [10] that is based on depth-first search and runs in $O(|E|)$ time.

- Once we have installed a regenerator at a node u by one of the above steps, all its neighbors $N(u)$ in M can communicate with each other via u. Any node pair $v, v' \in N(u)$ that was so far not directly connected can now be considered connected, and consequently we add a corresponding edge (v, v') to E'. In this way, a regenerator node and its neighbors become a fully connected clique.

- Even more generally, if nodes with installed regenerators are directly connected, all their individual neighbors are able to communicate with each other, and they are therefore connected to a clique.

- Knowing the biconnected components allows for an exact decomposition of the RLP into independent subproblems: The RLP can be solved for each biconnected component independently, and the overall solution is obtained as the union of the subproblem's solutions.

Especially on sparse input graphs, this preprocessing typically reduces the complexity considerably. The decomposition into independent subproblems even allows for a straight-forward parallelization. The nodes the preprocessing already identifies as necessary places for regenerators do not need to be paid further special attention when solving the remaining reduced problem(s): As observed in [3], an optimal solution to the reduced problem will not have another regenerator placed there since such a regenerator would not decrease the number of NDC pairs further – all its neighbors are already connected in M.

3 Related Work

While communication network design obviously is a huge research area, the RLP as we consider it here has only recently gained attention and relatively few approaches for solving it exist so far.

Chen et al. [3] prove its NP-completeness via a reduction from the vertex cover problem and propose three construction heuristics as well as an exact branch-and-cut algorithm. The first heuristic is a straight-forward greedy approach that iteratively selects a node where the placement of a regenerator reduces the number of NDC pairs the most; after each regenerator placement, the communication graph is updated accordingly. The second heuristic, called H1, exploits the observation that an optimal solution to the RLP on the communication graph M can also be represented as a spanning tree with regenerators placed at all internal nodes of the tree. The RLP thus corresponds to finding a spanning tree in M with a smallest number of internal nodes, or in other words, to finding a maximum leaf spanning tree [8]. Heuristic H1 builds a spanning tree iteratively by starting with a node with smallest degree and always appending a reachable node having maximum degree and placing a regenerator on it. The procedure terminates when all nodes are reachable. The third heuristic H2 is derived from H1 but also includes the updating of the communication graph as in the greedy approach. Solutions obtained from one of the construction heuristics are locally improved by utilizing a 2-for-1 neighborhood structure, which tries to replace any pair of placed regenerators by a single regenerator placed at some other node without becoming infeasible. The branch-and-cut approach is based on a cut-formulation of the RLP modeling it as a Steiner arborescence problem with a unit degree constraint. Experimental results show that the branch-and-cut is a viable approach for problems with up to 100 nodes, while especially H2 scales also well to much larger instances.

Duarte et al. [5] propose a GRASP using a randomized variant of the greedy algorithm and the 2-for-1 local search from [3]. The randomization of the greedy heuristic takes place in a GRASP-typical way: In each iteration the set of all feasible candidate elements, i.e., the nodes considered for placing a regenerator, is reduced to a restricted candidate list RCL and one element is selected from it uniformly at random. A regenerator is placed on the corresponding node, the communication graph updated and the process repeated until M is complete. The reduction to the RCL is done by including only candidate nodes where the installation of a regenerator would reduce the number of NDC pairs above a cutoff-value that is determined from the respective minimum and maximum values.

In an earlier technical report [4], Duarte et al. also investigated other variants of GRASP based on the H1 and H2 from [3] and a random-key genetic algorithm. They perform tests with instances of up to 100 nodes and in particular conclude that GRASP is superior to the genetic algorithm.

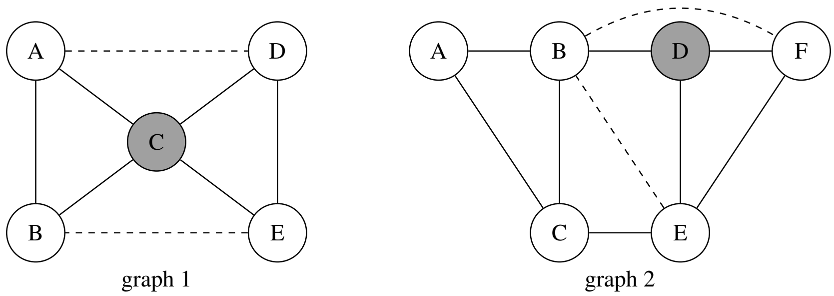


Figure 2: Placing regenerators merges adjacent maximal cliques.

Chen et al. [2] introduce the Generalized RLP in which they distinguish between two different kinds of nodes, the nodes for which communication needs to be ensured and nodes that may be selected for placing regenerators. The RLP as defined before can be considered the special case where these two node sets coincide. The authors present a greedy heuristic as well as a multi-commodity flow based mixed integer linear programming approach for it, which are based on similar ideas as their work in [3].

In the related *Regenerator Placement Problem* (RPP) [1, 12, 13, 15], a set of requests is given, e.g., in the form of a connection demand matrix, in addition to the network graph G. The objective is then to find appropriate routes in combination with installing a minimum number of regenerators to realize the specified communication demands. Thus, RPP differs from our RLP by considering specific routes. Flammini et al. [7] investigated the complexity of diverse variants of this problem.

Further related work addresses the issue of regenerator placement within larger network design problems. E.g., Yetginer and Karasan [9, 14] consider regenerator placement in the context of traffic engineering with restoration, and Gouveia et al. [9] focus on an MPLS over WDM network design problem in which the so-called WDM path constraint forbids path segments between two components that are longer than a given maximum length.

4 Clique and Independent Set Based GRASP

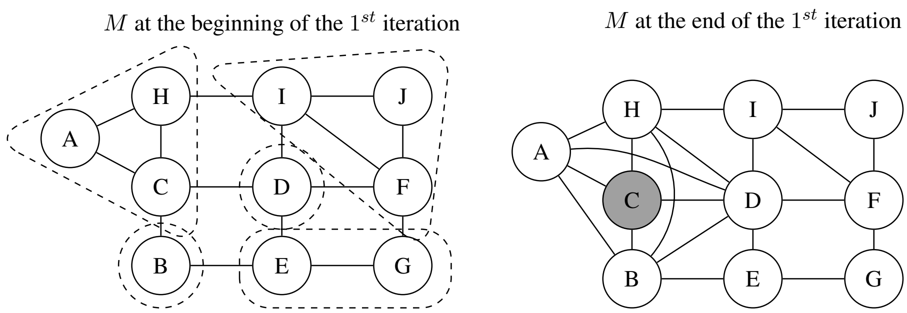
We utilize two basic structures of graph theory, cliques and independent sets, to exploit further structural aspects of the RLP on the communication graph by guiding the heuristic construction of solutions in promising ways. Considering the communication graph $M = (V, E')$, a *clique* is a subset of the nodes $C \subseteq V$ such that for every pair of nodes $u, v \in C,\ u \neq v$, there exists an edge $(u, v) \in E'$. An *independent set* in M is a node set $I \subseteq V$ where no two nodes are adjacent, i.e., for every pair of nodes $u, v \in I$, $(u, v) \notin E'$ holds.

We consider two different concepts for utilizing cliques and independent sets in construction and local search procedures,

- one which subsequently merges existing cliques by the introduction of regenerators until only one big clique is left,
- and another strategy which tries to identify independent sets whose nodes should be connected with as few as possible regenerators.

4.1 Clique Based Solution Construction

Updating the communication graph iteratively by adding additional edges to E' each time a regenerator is introduced implies that M is fully connected once a feasible solution is found; i.e., in the end the communication graph forms one large clique. The placement of a regenerator at some node $u \in V$ can therefore be perceived as a merge process of two or more maximal cliques incident to u into a bigger one. Looking at the example graph 1 in Fig. 2, the two maximal cliques $\{\mathrm{A,B,C}\}$ and $\{\mathrm{C,D,E}\}$ share



u	$\Phi(u)$
A	0
B	0.83
C	2
D	1.5
E	2
F	1.5
G	0.33
H	0.33
I	0.33
J	0

Figure 3: Clique based solution construction heuristic.

the common node C. The placement of a regenerator in node C leads to the single maximum clique $\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$. In graph 2 the two cliques $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and $\{\mathrm{D}, \mathrm{E}, \mathrm{F}\}$ are disjoint, and it is not possible to merge them by placing a single regenerator. A first regenerator installed in node D would be a possible choice yielding the cliques $\{\mathrm{B}, \mathrm{D}, \mathrm{E}, \mathrm{F}\}$ and $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, both sharing now the common node B, where a second regenerator would finally yield the complete graph. This observations lead to the assumption that it is desirable to identify as large cliques as possible while getting along with few clique merge steps. Another noticeable fact is that a node within a clique having no adjacent nodes outside the clique needs not to be considered as a candidate to house a regenerator. Therefore candidate nodes can frequently be eliminated when bigger cliques are obtained.

Our construction procedure exploiting this clique-merge point-of-view determines in each iteration cliques in M so that each node belongs to exactly one clique; i.e., the node set V is partitioned into disjoint cliques. Note that single nodes may also form individual cliques in this context. Each clique is greedily constructed by starting with a node $u \in V$ not being part of a clique so far and having highest degree. All its neighbors $U = N(u)$ are determined, and the clique is iteratively grown by always adding a node $v \in U$ which has the most adjacent nodes in U. U is updated by removing v and all nodes that are not adjacent to v. Ties are broken randomly.

Having determined the partitioning of M into cliques, a new regenerator is deployed on a node that is chosen according to a valuation function $\Phi(u)$, $u \in V$. As motivated before, the next regenerator should be one that is expected to lead to a small overall number of clique merges. Therefore $\Phi(u)$ first considers all neighbors $N(u)$ and their associated cliques. Since we cannot expect that placing a regenerator at u will in general be able to merge all these cliques, we define $\Phi(u)$ as the sum of the number of adjacent nodes per neighboring clique divided by the cardinality of the neighboring clique. More specifically, let $C(u) = \{C_1, \ldots, C_{|C(u)|}\}$ be the set of all neighboring cliques of u, then

$$\Phi(u) = \sum_{i=1}^{|C(u)|} \frac{|N(u) \cap C_i|}{|C_i|}. \tag{1}$$

The node $u \in V$ with the highest score $\Phi(u)$ is added to the set of regenerator nodes R. In case of ties the node having higher degree is favored, and if node degrees are also equal, ties are broken randomly.

Finally the communication graph is updated accordingly, adding for each NDC node pair which is now connected an edge to E'. The whole process is iterated until M is complete.

Considering the example in Fig. 3, the clique construction procedure determines at the beginning of the first iteration cliques $\{(\mathrm{A}, \mathrm{C}, \mathrm{H}), (\mathrm{F}, \mathrm{I}, \mathrm{J}), (\mathrm{E}, \mathrm{G}), (\mathrm{B}), (\mathrm{D})\}$. The score for each candidate node is listed in the table on the right hand side. For example, candidate node C has a score of $\Phi(\mathrm{C}) = \frac{2}{2} + \frac{1}{1} = 2$, which is together with the equal score of E highest among all nodes' scores. Since node C has higher degree than E, C is considered most promising, and a regenerator is installed on it.

4.2 Independent Set Based Solution Construction

Our second strategy follows a dual point-of-view and is based on independent sets. Since the identification of NDC node pairs which would become connected in case of a regenerator placement can be computationally relatively costly, we consider a faster alternative. In principle, an NDC node pair corresponds to an independent set of cardinality two. An independent set of cardinality larger than two therefore can be interpreted as a generalization; i.e., it is a larger set of nodes among which no communication is possible so far at all. If we can place a regenerator at a node with many of its neighbors being part of an identified independent set, this is in general highly desirable, since a quadratic number of NDC pairs can be connected in this way. An evaluation based on the number of nodes being part of an independent set may be computationally significantly less expensive than iteratively determining the exact number of NDC pairs that would be connected for each candidate node.

In more detail, our approach determines in each major iteration maximal independent sets by greedily finding maximal cliques on the *complementary graph* $\overline{M}$ of the communication graph M. Note that any clique of some graph always is an independent set for its complementary graph. For finding the cliques, we essentially use the same algorithm as in the clique based heuristic described above, but with one major difference: The cliques are now not required to be disjoint anymore, i.e., one node can be part of several cliques, and the construction of each clique is thus stopped only when a maximal clique is reached. Further cliques are constructed until each node appears in at least one clique.

Next it is checked if there is an independent set consisting of only a single node. As all independent sets are now maximal, this would mean that there is no other node which is not adjacent to the particular candidate node in M. In this case we therefore place a regenerator at this node which immediately connects all remaining NDC node pairs.

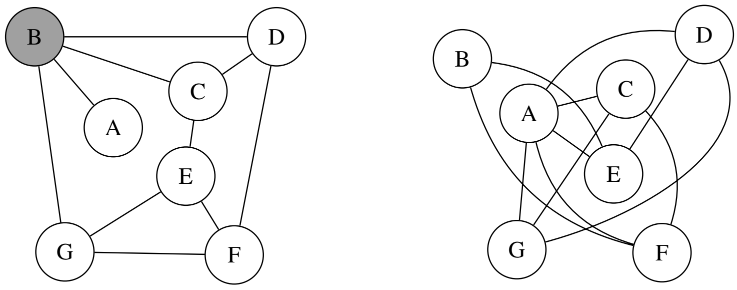
Otherwise we evaluate all candidate nodes $u \in V$ again by a valuation function $\Psi(u)$, which is now aimed at finding the place where an additional regenerator would connect most independent set nodes. More precisely, let I be the set of independent sets, then

$$\Psi(u) = \prod_{j=1}^{|I|} \frac{\max(1, |N(u) \cap I_j|)}{|I_j|}. \tag{2}$$

We consider each independent set $I_j,\ j = 1, \ldots, |I|$, and determine its fraction of nodes that are neighbors of u and would therefore be connected by placing a regenerator at u. These fractions are multiplied for all independent sets so that more emphasis is put on nodes u connecting several nodes from several independent sets instead of possibly more nodes from only a single or very few independent set(s). This latter aspect is important as in this way, typically the more "central" nodes are also preferred. Again, the candidate node $u \in V$ with highest score is added to the set of regenerator nodes R. In case of ties nodes with higher degree are favored or, in case of also equal degree, ties are broken randomly. Finally, the communication graph and its complementary graph are updated, i.e., edges are added for newly connected nodes to M and removed from $\overline{M}$, respectively. The whole process is repeated until M is complete.

In Fig. 4 the independent set construction procedure first determines all independent sets using the described strategy and then calculates $\Psi(u)$ for each node $u \in V$, see the table at the right side. As node B has the highest score, a regenerator is installed on it.

A simplified variant of the independent set construction heuristic determines only a single independent set in each iteration. While it is considerably faster, we lose some quality in the selection of the next regenerator node. This single maximum independent set is derived using a marking algorithm on the communication graph: The node which has the fewest adjacent umarked neighbors is marked as next independent set node and its neighbors are marked as "already used". To partly compensate the loss of quality in the selection of the best candidate node the valuation function $\Phi(u)$ is enhanced taking an adaptive aspect into account. Let I_1 be the single independent set and $SP(v, w)$ be the length of a



communication graph M *complement graph* $\overline{M}$

u	$\Psi(u)$
A	0.08
B	0.25
C	0.17
D	0.17
E	0.17
F	0.17
G	0.17

independent sets: $\{(\mathrm{A}, \mathrm{C}, \mathrm{G}), (\mathrm{B}, \mathrm{F}), (\mathrm{D}, \mathrm{E})\}$

$\Psi(\mathrm{B}) = \frac{1}{2} \cdot \frac{3}{3} \cdot \frac{1}{2} = 0.25$

Figure 4: Independent set based solution construction heuristic.

shortest path between two vertices $v, w \in V$ in terms of the number of edges in M, then

$$\Phi(u) = \frac{|N(u) \cap I_1|}{|I_1|} + \frac{|R|}{|V|} \cdot \frac{\max_{v \in V, w \in R} SP(v, w)}{\max(\min_{w \in R} SP(u, w), 1)}. \tag{3}$$

At the beginning of the solution construction the cardinality of R is small and the second term therefore also tends to be small. In this phase the preferred regenerator is the one, which primarily connects most independent set nodes. The more regenerators are identified, the more it is important for a candidate node to have a short (relative) distance to another regenerator, which is expressed by the second term. This assumption makes sense since a feasible solution to the RLP can always be obtained by constructing a spanning tree on the communication graph M and placing regenerators on all its internal nodes. The selection of the next regenerator node as well as updating the communication graph and its complementary graph is done as usual.

4.3 Advanced Local Search

As mentioned, previous work [5] only considered a 2-for-1 neighborhood structure for locally improving candidate solutions. Frequently, only few improvement steps are possible in such a local search. Sometimes, a reordering of regenerator nodes would allow for further reductions, but this option has so far not been considered. Hence we introduce larger *destruct and recreate* neighborhood structures based on our clique and independent set construction mechanisms.

Each one consists of two phases, starting with a destruction phase where some regenerators are temporarily removed followed by a recreation phase where the previously presented clique or independent set based strategy is used to complete the infeasible solution again. Note that this approach can also be perceived as a Very Large Neighborhood Search. We used a combination of two different destruction methods, i.e., ways to select the regenerators to be removed:

The first one is based on the observation that a solution to the RLP on M can also be interpreted as maximum leaf spanning tree. Since a feasible set of regenerators corresponds to all the internal nodes of such a spanning tree, the idea is to flatten its branches and to remove redundant regenerators. Therefore each regenerator node $u \in R$ is evaluated w.r.t. its number of adjacent regenerators $r(u)$. Given a cutoff value α and the minimum as well as maximum values $r_{\min}$ and $r_{\max}$, a regenerator node u is removed from R, if

$$r(u) \leq r_{\min} + \alpha \cdot (r_{\max} - r_{\min}). \tag{4}$$

In case of the situation that all regenerator nodes share exactly the same number of adjacent regenerators, only those with the highest node degree remain in the temporary solution.

In addition we use as second destruction mechanism a uniform random selection of regenerator nodes, where each node is selected with probability α. Based on preliminary experiments we finally

RN1				
Algorithm	⊘-Iter	⊘-$\|R\|$	#-Best	⊘-t_{Best}
CG-GRASP	365.1	4.54	985 (98.5%)	5.43
CL-GRASP	2280.9	**4.53**	996 (99.6%)	1.10
IS-GRASP	2604.1	**4.53**	**1000 (100.0%)**	0.33
SIS-GRASP	3009.8	**4.53**	**1000 (100.0%)**	0.31
Hyb-GRASP	2506.9	**4.53**	**1000 (100.0%)**	**0.24**

EN1				
Algorithm	⊘-Iter	⊘-$\|R\|$	#-Best	⊘-t_{Best}
CG-GRASP	4626.2	11.97	783 (97.9%)	9.05s
CL-GRASP	970.3	11.95	793 (99.1%)	3.85s
IS-GRASP	1049.0	**11.94**	**800 (100.0%)**	0.81s
SIS-GRASP	1769.0	11.95	795 (99.4%)	2.96s
Hyb-GRASP	1014.0	**11.94**	**800 (100.0%)**	**0.33s**

Table 1: Results on the small instance sets RN1 and EN1.

RN2				
Algorithm	⊘-Iter	⊘-$\|R\|$	#-Best	⊘-t_{Best}
CG-GRASP	21.7	7.97	428 (42.8%)	354.75s
CL-GRASP	29.1	7.17	741 (74.1%)	193.48s
IS-GRASP	187.9	7.10	819 (81.9%)	131.49s
SIS-GRASP	297.6	7.24	720 (72.0%)	188.90s
Hyb-GRASP	85.0	**7.07**	**848 (84.8%)**	**115.00s**

EN2				
Algorithm	⊘-Iter	⊘-$\|R\|$	#-Best	⊘-t_{Best}
CG-GRASP	9.0	12.57	149 (18.6%)	575.72s
CL-GRASP	18.2	11.88	377 (47.1%)	394.22s
IS-GRASP	41.5	11.39	670 (83.8%)	135.81s
SIS-GRASP	143.9	11.80	375 (46.9%)	387.59s
Hyb-GRASP	34.4	**11.37**	**688 (86.0%)**	**123.17s**

Table 2: Results on the large instance sets RN2 and EN2.

decided to apply the spanning tree based selection with a probability of 2/3 and otherwise the random selection.

4.4 GRASP

For a general introduction to GRASP see e.g. [6]. Its basic principle is to iteratively apply a randomized version of a construction heuristic for the problem at hand to create a set of diverse, meaningful initial solutions. These are typically further improved by some local search. The overall best solution is finally returned as result.

Our construction procedures described in the paragraphs above already include random decisions: As mentioned, ties are finally always broken uniformly at random. Experimental investigations indicated that such ties occur relatively frequently, and consequently, solutions created in different runs are already relatively diverse. We considered a further randomization as it is usually done in GRASP by randomly selecting from restricted candidate lists, but in the end decided to stay with the already originally available randomness. A further significant randomization appeared to most likely degrade performance.

As local improvement within our GRASP, we apply the above advanced local search. In the course of this paper we studied the following GRASP variants: CG-GRASP is an efficient reimplementation from [5], parameterized by $\alpha = 0.6$ as suggested. CL-GRASP, IS-GRASP, and SIS-GRASP make use of the clique, independent set, and simplified independent set mechanisms, respectively, using them for the construction of starting solutions as well as in the destroy-and-recreate local search. Hyb-GRASP is a fifth variant in which we combine the independent set based construction approach with the clique based destroy and recreate neighborhood structure.

For further algorithmic details and variants, especially also a Variable Neighborhood Search making use of the clique and independent set based mechanisms, we refer to the first author's master thesis [11].

5 Experimental Results

In our computational experiments we use four test sets. Sets RN1 and RN2 are the smaller and larger random network instances from [3], respectively. There are 10 instances for each combination of $|V| \in \{40, 60, 80, 100\}$ (RN1) as well as $|V| \in \{200, 300, 400, 500\}$ (RN2) and proportion of NDC node pairs $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The other two benchmark sets contain small and large self-generated Euclidean network instances EN1 and EN2. In these cases, nodes correspond to randomly selected points in a square of size 100×100. There are again 10 instances for each combination of $|V| \in \{40, 60, 80, 100\}$ (EN1) as well as $|V| \in \{200, 300, 400, 500\}$ (EN2) and $d_{\max} \in \{20, 30, 40, 50\}$.

For a fair comparison among the different GRASP variants, we terminated each metaheuristic run after the same instance-specific CPU-time, which was chosen by preliminary tests so that the slowest GRASP variant could achieve at least 100 iterations in RN1 and EN1. Due to time limitations for the evaluation of RN2 and EN2 the metaheuristic runs had to be terminated after a maximum period of two hours per instance. For each instance five independent runs were executed with each metaheuristic, collecting the information on the average total number of iterations $\oslash$-Iter, the average number of needed regenerators $\oslash$-$|R|$, the number of times a finally obtained solution corresponds to the overall best-known solution #-Best, and the average runtime until a best-known solution was found $\oslash$-t_{Best} (or the total runtime if the best-known solution was not found). The destruct and recreate local search was continued until ten consecutive iterations had not been able to improve the solution, and the cutoff value α for selecting nodes in the destruction phase was set to 0.55. Results are listed in Table 1 and Table 2. Since the test sets RN1 and RN2 consists of 200 instances, the maximum number of best-known solutions is 1000 in these cases, and for EN1 and EN2 it is 800.

We can observe that the GRASP-variants using the new clique and independent set based strategies perform in general much more average iterations per instance than the CG-GRASP. There is only one outlier in the EN1 test set, where the CG-GRASP performs on instances consisting of 40 nodes considerably more average iterations than the other heuristics. Due to the effects of preprocessing on sparse graphs CG-GRASP is able to evaluate candidate nodes in these cases in a very fast way, without having the overhead of determining cliques or independent sets first. On instances consisting of 60,80 and 100 nodes, our new GRASP variants are faster again and thus perform more iterations.

Especially on the large instance sets RN2 and EN2 our approaches in general obtain significantly more often best-known solutions than the previously introduced CG-GRASP. Overall, the best results are achieved by Hyb-GRASP. Its independent set based strategy for constructing starting solutions is complemented very well by the clique based destroy and recreate local search, and due to the combination of these different strategies more diverse solutions are investigated. It is also remarkable that SIS-GRASP performs very well in comparison to CG-GRASP and consistently executes the most iterations per instance due to its short running times, although it cannot catch up with Hyb-GRASP.

To check if observed performance differences w.r.t. the numbers of obtained best-known solutions are significant among the different GRASP variants, we applied paired Wilcoxon signed rank tests. Results indicate that Hyb-GRASP and IS-GRASP dominate all other algorithms on error levels of less than 2.5%. Hyb-GRASP dominates IS-GRASP on an error level of 3.9%. The third place was shared by CL-GRASP and SIS-GRASP achieving significantly more best-known solutions than CG-GRASP on an error level of again less than 2.5%.

6 Conclusions

In this article we have introduced new strategies for selecting regenerator nodes in the course of heuristically constructing solutions to the RLP. These strategies make use of two fundamental principles from graph theory, i.e., cliques and independent sets. The construction techniques are further utilized in an advanced local search based on a destroy and recreate concept. Ultimately, we iteratively apply our construction procedures together with the destroy and recreate local search, representing different variants of GRASP. No explicit additional randomization of the construction heuristics turned out to be necessary due to the already originally included randomized tie breaking. Our experimental results indicate that within the same time, the new GRASP variants obtain significantly more best-known solutions than CG-GRASP from [5]. Especially Hyb-GRASP, where the independent set based construction of starting solutions is combined with the clique-based destroy and recreate local search, achieves the best results, closely followed by IS-GRASP which uses independent set concepts only. The simplified independent set heuristic is fastest and therefore the most iterations can in general be executed with it within the allotted time. SIS-GRASP competes very well with CG-GRASP, although it cannot catch up with Hyb-GRASP and IS-GRASP.

In future work it seems promising to exploit the proposed clique and independent set based solution construction principles also in approaches for related problems like the generalized RLP, the RPP, or more general network design problems where regenerator placement is a considered aspect.

References

[1] A. Beshir, F. Kuipers, A. Orda, and P. Van Mieghem. Survivable routing and regenerator placement in optical networks. In *4th International Congress on Ultra Modern Telecommunications and Control Systems*, pages 684–690. IEEE, 2012.

[2] S. Chen, I. Ljubic, and S. Raghavan. The generalized regenerator location problem. In *Proceedings of the 2009 International Network Optimization Conference (INOC 2009)*, Pisa, Italy, 2009.

[3] S. Chen, I. Ljubic, and S. Raghavan. The regenerator location problem. *Networks*, 55(3):205–220, 2010.

[4] A. Duarte, R. Marti, M. Resende, and R. Silva. Randomized heuristics for the regenerator location problem. Technical report, AT&T Labs Research, Florham Park, New Jersey, 2010. URL http://www.research.att.com/ mgcr/doc/gpr-regenloc.pdf, accessed: 2012-11-30.

[5] A. Duarte, M. Resende, and R. Marti. Grasp for the regenerator location problem. In *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pages 111–121, Udine, Italy, 2011.

[6] T. A. Feo and M. G. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[7] M. Flammini, A. Marchetti Spaccamela, G. Monaco, L. Moscardelli, and S. Zaks. On the complexity of the regenerator placement problem in optical networks. In *Proceedings of the 21st Annual Symposium on Parallelism in Algorithms and Architectures*, pages 154–162, Calgary, AB, Canada, 2009. ACM Press.

[8] T. Fujie. The maximum-leaf spanning tree problem: Formulations and facets. *Networks*, 43:212–223, 2004.

[9] L. Gouveia, P. Patricio, A. Sousa, and R. Valadas. MPLS over WDM network design with packet level qos constraints based on ILP models. In *Proceedings of the 22nd Annual IEEE Conference on Computer Communications (INFOCOM 2012)*, pages 576–586. IEEE Press, 2003.

[10] J. Hopcroft and R. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

[11] P. Jahrmann. Metaheuristics for the Regenerator Location Problem. Master's thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, March 2013.

[12] S.-W. Kim, S.-W. Seo, and S. C. Kim. Regenerator placement algorithms for connection establishment in all-optical networks. In *IEEE Global Telecommunications Conference*, pages 1205–1209. IEEE Communications Society, 2000.

[13] F. A. Kuipers, A. Beshir, A. Orda, and P. V. Mieghem. Impairment-aware path selection and regenerator placement in translucent optical networks. In *IEEE International Conference on Network Protocols (ICNP 2010)*, pages 11–20. IEEE Computer Society, 2010.

[14] E. Yetginer and E. Karasan. Regenerator placement and traffic engineering with restoration in GMPLS networks. *Photonic Network Communication*, 6:139–149, 2003.

[15] W. Zhang, J. Tang, K. E. Nygard, and C. Wang. REPARE: Regenerator placement and routing establishment in translucent networks. In *Proceedings of the Global Communications Conference, 2009 (GLOBECOM 2009). Honolulu, Hawaii, USA*, pages 1–7. IEEE, 2009.