A Heuristic Approach to Automated Forest Road Location

David Meignan, Jean-Marc Frayret, Gilles Pesant, and Mathieu Blouin

Abstract: An optimization problem arising when planning forest harvesting operations is the location of new access roads. The new roads must cover several areas to be harvested at minimum cost. This problem is of economical and environmental relevance in the domain of forestry. In this study, the problem is expressed as a P-forest problem in a graph. It consists in determining a set of tree structures in a graph that covers a set of vertices corresponding to harvest areas. The objective is to minimize the sum of construction costs and harvesting costs. In addition to the location of roads, the P-Forest problem has several relevant applications including public transport, electricity transmission, roads, pipelines and communication networks design. This paper presents a Greedy Randomized Adaptive Search Procedure (GRASP) to solve this problem. The heuristic was implemented on a decision support system and computational experiments were conducted on randomly generated and real instances to demonstrate the performance and practical efficiency of the proposed approach. A comparison with manually designed forest road networks on the real instances shows a clear advantage for the proposed method.

Résumé: Un problème important d'optimisation intervenant dans le cadre de la planification des opérations forestières est la localisation de nouveaux chemins d'accès pour le débardage. Les nouveaux chemins forestiers doivent couvrir un ensemble de zones de récolte à un coût minimum. Ce problème est d'une importance économique et environnementale majeure pour le secteur forestier. Dans cet article, le problème est exprimé sous la forme d'un problème de P-arbres dans un graphe. Il consiste à déterminer un ensemble de structures arborescentes dans un graphe de manière à couvrir indirectement les points qui correspondent aux zones à débarder. L'objectif est de minimiser les coûts de débardage et de construction de chemin. Ce problème d'optimisation trouve également des applications dans différents domaines tels que la conception de réseaux de transports, de réseaux électriques, de routes, de conduites et de réseaux de télécommunications. Nous proposons une procédure de recherche gloutonne adaptative pour résoudre ce problème. Cette heuristique a été intégrée à un système d'aide à la décision et testée sur des instances générées aléatoirement ainsi que sur des scénarios réels. Afin de démontrer l'efficacité ainsi que l'aspect opérationnel de l'approche, les résultats ont été comparés à des réseaux conus manuellement. Cette comparaison indique un net avantage en faveur de notre approche.

1. Introduction

Due to the evolution of environmental and economical aspects in forestry, there is a continuing need to find efficient optimization methods in this application field. Forest harvesting planning, that considers all operations, from cutting to delivering at processing plants, addresses several significant optimization problems. In [Rönnqvist, 2003] the author reviews the major optimization problems in forestry. Most of these problems are hard combinatorial optimization problems of large size or problems that require a high quality solution in a small amount of time. In addition, these problems involve a wide variety of data (geographical, economical, legal and agronomical).

The problem considered in this paper is a network design problem that consists in determining the roads that will be used to transport trees or logs from harvested areas to processing plants or intermediate storage. This network design problem appears at different scales of harvest planning. When the entire supply chain of forest products is considered, the problem consists in determining the flows on a given road network to satisfy the demand of the forest product industry. This stage of planning is at the tactical level [D'Amours et al., 2007]. In this case, the objective of the network design problem is usually to minimize transportation and road maintenance costs over several time periods [Karlsson et al., 2004]. This paper addresses a planning problem that is closer to the operational scale and that consists in determining the location of new roads to access harvest areas. Contrary to the network design problem in the global supply chain, the costs to optimize when determining new roads are road construction costs and total harvesting costs that include logging and hauling costs. In addition, the location of the new roads must satisfy several constraints such as environmental, topological and soil constraints. This road location problem is of great economic and environmental importance. Environmental issues include the respect of environmental regulations, the reduction of the access road network size and the minimization of environmental impact such as the limitation of erosion.

The underlying optimization problem consists in minimizing harvesting cost and road construction cost. These two costs result in two conflicting objectives that should not be addressed

David Meignan and Jean-Marc Frayret.[1] Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, 2500 chemin de Polytechnique, H3T 1J4, Montréal, Canada.
Gilles Pesant. Département de Génie Informatique et Génie logiciel, École Polytechnique de Montréal, 2500 chemin de Polytechnique, H3T 1J4, Montréal, Canada.
Mathieu Blouin. FPInnovations, 570 St-Jean Blvd., H9R 3J9, Pointe-Claire, Canada.

[1]Corresponding author (e-mail: jean-marc.frayret@polymtl.ca).

DOI: 10.1139/x2012-140

separately. The harvesting cost may be reduced by implementing a dense road network that facilitates the access of harvest areas. However, such a dense network generates high construction costs. Conversely, limiting the number and length of new roads may reduce the total construction cost, but a sparse road network increases harvesting costs because harvest areas are less accessible. In this sense, the design of a forest road network must achieve a trade-off between harvesting cost and construction cost [Chung et al., 2008]. The integration of both costs is a challenging aspect of the problem.

In [Clark et al., 2000], the authors investigate the problem of access road network design in combination to the problem of scheduling the stands to harvest. The problem consists in determining the temporal planning of road building and stands to harvest, with the objective to maximize the revenue generated from harvesting stands and minimizing road costs. In the study of [Clark et al., 2000], the spatial road design problem is reduced to a minimum spanning tree problem, i.e. the problem of finding a network that connects a set of points without any intermediate point. In this model, harvesting costs depend on the temporal planning of road building and not to the distance between access roads and areas to harvest. Similarly, Weintraub and Murray [2006] describe a model for the spatial and temporal road design problem and review some exact and approximate solution methods. The proposed model does not take into account the determination of the location of the roads which is determined manually by the planner.

Dean [1997] investigates the forest road network design problem using a model similar to the Steiner tree problem. He assumes that a set of destination points for the roads is known and thus does not consider the minimization of the distance between access roads and areas to harvest. He proposes some heuristic methods based on the minimum path heuristic for the Steiner tree problem [Takahashi and Matsuyama, 1980]. Later, Murray [1998] provides a formal definition of the problem defined by Dean [1997].

Along the same line, Anderson and Nelson [2004] as well as Stückelberger et al. [2007] propose some heuristics to solve the forest road network design problem. They consider only the construction cost that reduces the problem to a Steiner tree problem in a graph. However, they propose some methods to generate the underlying graph according to topographical and operational constraints. In these two studies, the heuristics are also based on the minimum path heuristic.

Epstein et al. [2006] and Legues et al. [2007] investigate the machinery location and road network design problem. This problem consists in determining the location of harvesting machinery and the location of access roads considering topographical constraints, harvesting costs and construction costs. These two studies integrate in the problem model construction costs and harvesting costs. This last cost depends on the distance between areas to harvest and machinery. The proposed heuristic solution approaches are based on the same strategy that iteratively determines machinery location and then connects it to existing roads or exit points. However, these approaches are more suited to steep terrains that necessitate the location of equipment such as towers to perform aerial cable harvesting.

The access road location problem addressed in the rest of the paper is modelled as a P-Forest Problem (PFP). In this problem, potential road segments form a graph and areas to harvest correspond to vertices to cover. The problem consists in determining a road network of minimum cost that indirectly covers all vertices to cover, i.e. determining a road network from which all areas to be harvested are accessible. The PFP model includes both harvesting costs and construction costs. Contrary to the model presented in [Epstein et al., 2006], indirect covering implies that harvesting may start at any point of the new roads or existing roads. This aspect of the model appears most appropriate to represent hauling operations with ground equipment such as tractors or skidders instead of aerial equipment. Furthermore, the PFP is able to represent the access of harvest areas from existing roads. This last case is particularly interesting to determine whether it is advantageous to extend an existing road network with new access roads.

In this study, a Greedy Randomized Adaptive Search Procedure (GRASP) is proposed to solve the PFP. In comparison to previous heuristics approaches for solving the forest road network design problem, the proposed heuristic simultaneously optimizes construction and harvesting costs through a local search procedure. On real instances, GRASP provides an initial solution within a few seconds, and the solution can be improved by running GRASP for several iterations.

The proposed heuristic has been implemented on a decision support system and tested on randomly generated and real problem instances. Three sets of experiments are performed in order to validate the proposed heuristic approach. First, the robustness of the heuristic strategy is evaluated. Then, the proposed GRASP is compared with a branch-and-bound approach to evaluate the performance in terms of solution quality and computational time. Finally, results on real problem instances are compared to solutions manually planned using PlaniRoute [FPInnovations] a Geographical Information System (GIS) application. This last part of the experiments is aimed at demonstrating the practical efficiency of the proposed approach on real instances.

2. Access road location

2.1. Description of the problem

The studied problem of determining the location of new access roads corresponds to decisions at the tactical level since the planning horizon covers several months and additional operational decisions must be taken to implement a solution to the problem. However, this problem necessitates a high level of detail usually related to the operational scale.

On a practical level, the access road network is elaborated by an expert considering a planning horizon of approximately 6 to 12 months. Then, this plan is validated by authorities that examine the compliance with applicable laws and regulations. Finally, the validated road network is implemented with possible minor adjustments made on the ground. The map of new access roads, in most cases, is determined with the help of a Geographical Information System (GIS). In addition, specialized software can also be used to support the expert in this task. These tools aim at reducing the costs of planned operations and at better integrating legal, environmental and operational constraints [Sessions et al., 2006]. Despite the use of a GIS, the location of access roads remains a tedious task because of the large number of input data and constraints, and of the combinatorial complexity of the problem. The planner must also find a good balance between harvesting and construction costs.

Three types of input are needed to design a forest road network in the context of the study: legal and environmental constraints, geographical data, and operational data and constraints.

First, legal and environmental constraints mainly consist of the definition of harvestable areas and protected areas. Land use planning and environmental protection regulations can limit the location of forest activities including harvesting and road building. For instance, in Quebec, forest operations are subject to the Forest Act [Gouvernement du Québec, 2010]. This set of regulations cover the protection of lakes and watercourses by defining some minimum distances allowed for forest operations, and establish the rules for locating and dimensioning bridges and culvers. It also concerns the protection of ecological reserves, archeological sites and natural habitats of some species such as caribous and herons, by prohibiting forest operations around these territorial units.

Second, the design of a forest road network also requires a large amount of geographical information typically stored in a GIS. These data include the areas to harvest, topological and soil data, the existing road network, and forest characteristics. Among these data, soil type, drainage and slope are the major criteria to determine the feasibility of roads and estimate construction cost. Soil type is determined by its composition, for instance rocks, gravels, sand, silt or clay, and the surface soil thickness. A good soil texture with sufficient depth is necessary to construct and exploit an access road. The drainage is related to the slope and the soil type. A well-drained soil is essential to ensure a sustainable use of the roads. Moreover, the road grade should be kept as low as possible. For the scenarios studied in this paper, the access roads cannot exceed a slope of 15% or be constructed on unstable soil or soil without deposits.

Finally, operational data is necessary to determine the type, costs and productivity of harvesting machinery. For harvesting operations, appropriate equipment must be selected in order to minimize soil deterioration and ensure productivity. For instance, grapple skidders should be used on small extraction distances, and clambunk forwarders on longer distances. Irrespective of the harvesting machinery, the major criterion of the harvesting cost is the distance between the access road and the area to harvest. In addition to productivity and cost parameters, a maximum harvesting distance to road is also considered. In most situations, skidding or yarding at a distance greater than 400 meters is too expensive and therefore should be avoided.

Costs and constraints of a forest road network are evaluated from these three sources of data. As a first constraint, the new roads have to be connected to existing roads or exit points defined by the planner. The exit points serve as potential junctions between new roads and existing roads not visible on the map. The set of new access roads must form a forest structure (in graph theory terminology) in such a way that all new roads can be reached from an existing road or an exit point. Then, the new roads cannot be constructed on protected areas, steep slopes, and other natural barriers for road construction such as lakes or large rivers. Finally, the forest road network, including existing roads and exit points, must cover all harvest areas. A harvest area is covered when the entire area is within the maximum harvesting distance to a road, taking into account all harvest barriers. Barriers considered for this coverage constraint may be different from those for construction. For instance, rivers are natural barriers for harvesting operations but not necessary for road construction.

The cost of a road network includes construction costs and harvesting costs. Since the optimization problem focuses on the location of new forest roads, regardless of the routing problem to transport trees or logs to processing plants or intermediate storage, the on-road transportation cost is neglected to focus on legal, environmental and topological aspects of roads location.

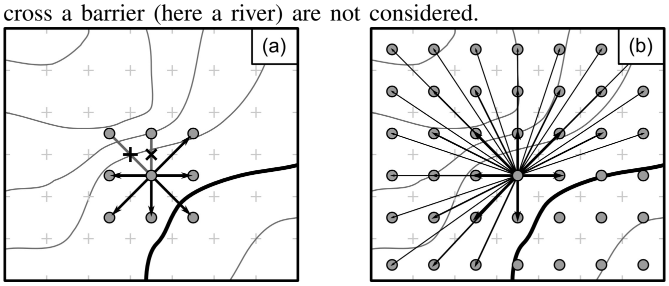
The construction costs are determined by the length of road segments and a combination of three other factors: soil type, slope and drainage. The cost of water crossing structures is included into road construction costs as a penalty when a forest road crosses a watercourse. Optimization of the location of water crossing structures is not investigated in this paper as it is an optimization problem in itself. However, the water crossing penalties can be adjusted to reflect the costs and constraints of these structures. Note that the PlaniRoute software [FPInnovations] offers a module for sizing and evaluating bridges and culverts in accordance with environmental regulations of Quebec and Ontario Provinces. Harvesting costs correspond to operating costs for cutting and limbing trees as well as the costs of moving trees to the roadside by skidding, yarding, or forwarding them. The harvesting costs of a forest section are calculated from its average distance to the nearest road, the volume of timber that can be extracted, and the harvesting productivity. The optimization problem's objective is to minimize the sum of these two antagonistic costs. Indeed, if harvesting costs are optimized independently of construction costs, lots of new roads are designed to reduce the distance to the areas to harvest, resulting in high construction cost, and conversely.

2.2. Problem formulation

In order to tackle the problem, the studied terrain is divided into square cells 50 meters wide. This width has been chosen because it corresponds to the minimum curve radius of roads and also to the average precision between the planning and the implementation of roads. Using this grid, the data are then projected to a directed graph as in [Epstein et al., 2006] and [Chung et al., 2008]. A vertex is defined at each cell's center, and two types of arcs are defined: potential road segments and harvesting arcs. As shown in Figure 1, the road segment arcs connect each vertex to its eight neighbors while harvesting arcs connect the vertices up to the maximum harvesting distance. A construction cost and harvesting cost is associated to road arcs and harvesting arcs respectively. Some arcs are removed from the resulting graph in order to consider construction and harvesting barriers. These barriers include, for instance, maximum slope (steep terrain) for road arcs, and rivers for harvesting arcs.

Based on the graph defined above, the problem consists in determining a set of tree-like paths of minimum cost that indirectly covers the vertices located on cells to harvest. A vertex is said to be indirectly covered if one of its harvesting arcs leads to a road at a distance not exceeding the maximum harvesting distance. The problem can be viewed as a forest location problem or P-Forest Problem (PFP) [Tamir and Lowe, 1992]. The PFP belongs to the category of extensive facility location

Fig. 1. Generation of the graph; (a) road arcs on steep slopes or non-building land areas are removed, (b) harvesting arcs that cross a barrier (here a river) are not considered.



problems, which includes location problems where facilities are generalized to paths, trees, cycles or subtrees instead of points [Mesa and Boffey, 1996], [Labb et al., 1998]. Following the classification introduced in [Mesa and Boffey, 1996], the studied problem is defined as follow:

- Network: the underlying network structure is a general graph,
- Demand: the demand, here the cells to be harvested, is located at the vertices,
- Facilities: the structure to be located is a forest,
- Facility extremities: the structural extremities are vertices,
- Decision criterion: the decision criterion is the construction cost (related to the total length of the new roads) added to the weighted min-distance-sum (sum of the minimum distances between cells to be harvested and access roads),
- Restrictions: each tree-like path must be connected to a existing road or an exit point, and each harvest vertex must be indirectly covered,
- Type of facilities: the facilities are central i.e. the roads must be as close as possible to the harvest vertices.

Several papers in the operations research literature have studied variants of the PFP. However, most of these studies consider the underlying network as a tree. A review of the most important contributions can be found in [Boffey and Narula, 1998], [Hutson and ReVelle, 1993] (indirect covering tree problem), [Kim et al., 1996] (single-tree location problems) and [Tamir and Lowe, 1992] (P-forest problem). These problems can fall into the category of bi-level network design problems, considering that the higher level consists in designing the facilities, and the lower one deals with the location of the facilities [Balakrishnan et al., 1991]. In addition to the location of roads in a logging region, these optimization problems have several relevant applications including public transport, electricity transmission, roads, pipelines and communication networks [Boffey and Narula, 1998].

2.3. Mixed-integer programming model

The following mixed-integer programming model represents the forest road network design problem as a PFP.

Let $G(V, E \cup A)$ be a bi-level directed graph with V the set of vertices, E and A two sets of arcs. Two subsets of vertices, $R \subset V$ and $B \subseteq V$ are defined. R is the set of vertices where existing roads and exit points are located. Vertices in B correspond to harvest vertices, i.e. the cells to be harvested. For each harvest vertex $i \in B$, w_i denotes the quantity of timber to be harvested in the corresponding cell.

The set of arcs E are potential road segments and A are harvest arcs, i.e. the segments that can be used to harvest cells from access roads. With each road arc in E is associated a strictly positive cost c_{ij} corresponding to the construction cost along the arc (i, j). For the set of harvest arcs A, the cost d_{ij} corresponds to the unit cost to harvest the cell i from an access road located at vertex j. The set of arcs A is defined in such a way that no arc longer than the maximal harvesting distance D is included. Thus, the maximum harvesting distance constraint is implicitly contained in the model.

Two sets of binary decision variables, x_{ij} and y_{ij}, and integer decision variables z_{ij} are introduced to represent a solution. The new access roads are described by x_{ij}, and y_{ij} indicates from which road location a cell is harvested. The values of these two decision variables are defined as follow:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is a new access road segment,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if the cell at } i \text{ is harvested from } j, \\ 0 & \text{otherwise.} \end{cases}$$

The integer decision variable z_{ij} corresponds to the timber flow from cells to be harvested to existing roads and exit points. The value of z_{ij} corresponds to the total timber quantity that pass through the arc (i, j). The PFP can be stated as follows:

$$[1] \qquad \text{Minimize} \left\{ \sum_{(i,j)\in E} c_{ij} x_{ij} + \sum_{(i,j)\in A} d_{ij} y_{ij} w_j \right\}$$

Subject to:

$$[2] \qquad \sum_{i \in V:(i,j)\in E} z_{ij} + \sum_{i\in V:(i,j)\in A} y_{ij} w_i = \sum_{k\in V:(j,k)\in E} z_{jk} \quad \forall j \in V \backslash R$$

$$[3] \qquad \sum_{(i,j)\in E: j \in R} z_{ij} + \sum_{(i,j)\in A:j\in R} y_{ij} w_i \geq T$$

$$[4] \qquad z_{ij} \leq x_{ij} T \quad \forall (i,j) \in E$$

$$[5] \qquad \sum_{k\in V:(j,k)\in E} x_{jk} \geq y_{ij} \quad \forall (i,j) \in A : j \notin R$$

$$[6] \qquad \sum_{j\in V:(i,j)\in A} y_{ij} \geq 1 \quad \forall i \in B$$

$$[7] \qquad x_{ij} \in \{0,1\} \quad \forall (i,j) \in E$$

$$[8] \qquad y_{ij} \in \{0,1\} \quad \forall (i,j) \in A$$

$$[9] \qquad z_{ij} \in \{0,\ldots,T\} \quad \forall (i,j) \in E$$

The first term of the objective function corresponds to the construction cost and the second one is the harvesting cost. Constraints 2 to 4 ensure the arborescent structure of the new roads in the graph. More precisely, the flow balance is guaranteed with Constraints 2, each flow ends at an existing road vertex or an exit point with Constraints 3, and Constraints 4 link the flows and the arcs of the new access roads. In Constraints 3 and 4, the parameter T corresponds to the total timber quantity ($\sum_{i \in B} w_i$). Constraints 5 and 6 ensure that all demands are allocated to a vertex in the solution or a root vertex. Finally, Constraints 7 to 9 require all decision variables to be integer with specific bounds.

In addition to the interest of a formal definition of the problem, this mixed-integer programming model has been used to obtain optimal solutions on small instances. These results are presented in Section 4.

3. Solution methodology

Several aspects of the problem have been considered to choose the optimization method. The first requirement of the optimization procedure consists in solving large problem instances in a limited amount of time. The real instances used for computational experiments, contain between 8 000 and 12 000 cells, which correspond to a map of 3 000 hectares for the largest instance. Another aspect considered for choosing the optimization method is the limit in modelling all features of the problem with the provided data. Despite the large amount of accurate data used for instantiating a problem instance, it is hardly possible nor even desirable to avoid planner's intervention for obtaining a realistic solution. For instance, in some cases, the choice between the construction of a road on a steep slope and the construction of a water crossing structure cannot be decided only on the objective function and must be determined by a forester. Thus, solutions need to be adjusted by an expert. The objective, in this context, is to develop an optimization procedure suitable for a decision support system. In this kind of system, an expert must be able to iteratively adjust parameters, constraints and initial data, according to solutions, in order to better fit the reality. Expert's interactions can also be useful to balance the weight of different inputs that are aggregated in the optimization model. A heuristic approach seems appropriate for these different aspects.

This section presents a Greedy Randomized Adaptive Search Procedure (GRASP). The choice of this simple metaheuristic without sophisticated memory has been motivated by the small number of solutions the method can obtain in a few minutes considering the size of the instances and software requirements. In addition, GRASP is well suited for a decision support system as it requires a limited parameter setting. Finally, despite the simplicity of the approach, the results obtained by GRASP are satisfactory according to the experts comments.

3.1. GRASP

GRASP is a metaheuristic that iteratively performs two steps until a stopping criteria, usually a maximum number of iterations, is reached. The first phase consists in a greedy randomized construction of a solution, and the second one is a local search that improves the solution until a local minimum is obtained. A review of GRASP, its variants, and some applications is given in [Resende and Ribeiro, 2003].

Contrary to trajectory or population-based metaheuristics, in GRASP the exploration of the search space is mainly performed during the construction of the solutions. This construction involves a "greedy function", and also a part of randomness. The construction phase consists in incorporating elements in a partial solution until a complete solution that satisfies constraints is obtained. To select the element to add to a partial solution, a list of candidate elements is created. Then, the best elements according to the greedy function are selected to form the restricted candidate list (RCL). Finally, an element in the RCL is randomly chosen to be incorporated into the partial solution.

In GRASP the size of the RCL is determined by a parameter α. In the implementation of GRASP for solving the PFP, this parameter corresponds to the proportion of candidates selected to form the RCL. If α is set to the value 0, the construction phase is equivalent to a pure greedy construction. At the opposite, the case $\alpha = 1$ corresponds to a random construction. The appropriate choice of this value is critical, and using the same value to solve different problem instances may hinder finding high-quality solutions [Resende and Ribeiro, 2003]. In the proposed GRASP, the value of this parameter is determined by an adaptive parameter strategy inspired by Reactive-GRASP [Prais and Ribeiro, 2000] as well as Estimation of Distribution Algorithms (EDA) [Pelikan et al., 1999]. The value of α is randomly determined using a normal probability distribution bounded between 0 and 1. The mean and variance parameters of the normal distribution dynamically evolve during the solving process and are computed from a sample of size E of the values of α that lead to the previous best solutions. The normal probability distribution is exploited to favor the convergence of the variable in the same way as the univariate marginal distribution model [González et al., 2002]. This self-adjustment method allows replacing the parameter α with a less sensitive parameter E. An evaluation of this adaptive parameter strategy is presented in Section 4.1.

The proposed implementation of GRASP for the PFP is described in Algorithm 1. An iteration starts by determining the value of the parameter α for the construction procedure based on the results of the previous iterations (line 3). Then, a new solution is generated and improved (lines 4-6). This solution is stored if its cost is the best found over the iterations (lines 7-9). A randomized two-step greedy construction procedure and a variable neighborhood descent procedure are respectively used for the construction phase and the improvement phase. These two procedures are described in the following sub-sections.

3.2. Two-step greedy randomized construction procedure

The two-step greedy randomized procedure provides initial solutions for the PFP which are then improved by local search. This procedure is based on the minimum path heuristic for the Steiner tree problem [Takahashi and Matsuyama, 1980], [Hwang et al., 1992] and is also related to Prim's algorithm for the minimum spanning tree problem. The basic principle of the minimum path heuristic and Prim's algorithm is to grow a tree by iteratively adding new branches that connect the current tree to a remaining terminal vertex. At each iteration, the

Algorithm 1: GRASP, greedy randomized adaptive search procedure for the PFP

```
Data: graph, the graph in which roads have to be located.
      maxIterations, the number of iterations of the GRASP.
Result: bestSolution, sets of arcs that represents the best
        found road network.
1  bestSolution ← ∅
2  for k = 0 to maxIterations do
3      α ← DetermineRCLParameter( )
4      coveringVertices ← SelectCoveringVertices(graph, α)
5      currentSolution ← MinimumPathHeuristic(graph, coveringVertices)
6      currentSolution ← VariableNeighborhoodDescent(currentSolution)
7      if bestSolution= ∅ or f(currentSolution) < f(bestSolution) then
8          bestSolution ← currentSolution
9      end
10 end
11 return bestSolution
```

Algorithm 2: SelectCoveringVertices, greedy randomized procedure for the selection of covering vertices

```
Data: graph, the graph in which covering vertices have to
      be located. α, relative size of the restricted
      candidate list (RCL).
Result: coveringVertices, a set of vertices that indirectly
        covers harvest vertices.
1  coveringVertices ← ∅
2  toCover ← graph.harvestVertices −
   VerticesCovered(graph.existingRoadVertices)
3  while toCover ≠ ∅ do
4      rcl ← CoveringVerticesRCL(α, toCover)
5      vertex ← RandomSelection(rcl)
6      coveringVertices ← coveringVertices ∪{vertex}
7      toCover ← toCover − VerticesCovered(vertex)
8  end
9  return coveringVertices
```

Algorithm 3: MinimumPathHeuristic, connects covering vertices with minimum path heuristic

```
Data: graph, the graph in which roads have to be located.
      coveringVertices, the set of vertices to be connected
      by roads.
Result: forest, a set of arcs that corresponds to the
        minimum cost roads between existing road
        vertices and covering vertices.
1  terminals ← coveringVertices
2  forest ← ∅
3  forestVertices ← existingRoadVertices
4  while terminals ≠ ∅ do
5      path ← MinimumPath(forestVertices, terminals)
6      forest ← forest ∪ path
7      forestVertices ← forestVertices ∪ Vertices(path)
8      terminals ← terminals − Vertices(path)
9  end
10 return forest
```

algorithm chooses the smallest-cost branch and adds it to the tree.

Contrary to the Steiner tree problem and the spanning tree problem, there is no set of vertices to be spanned for the PFP. Instead, harvest vertices must be indirectly covered, i.e. each harvest vertex must be harvestable from a road vertex. To overcome this difference, the first step of the greedy randomized procedure consists in determining a set of covering points (Algorithm 1, line 4). These points are defined so that all vertices to be harvested are covered. Then, the second step uses the minimum path heuristic to determine a set of access roads that span the covering points (Algorithm 1, line 5). In other words, the algorithm first locates a set of crossing points from which all areas to be harvested are covered. Then, these points are connected to the existing roads and exit points to form the new access road network.

In order to be used in a GRASP, the construction procedure must produce different solutions on successive calls. The randomization, necessary for the exploration of the search space, occurs at the first step of the construction procedure. This first step, that locates a set of covering points, is described by Algorithm 2. In this greedy randomized procedure, the set of vertices to cover is initialized with harvest vertices that are not covered by existing roads or exit points (line 2). While there remain uncovered vertices to be harvested, a RCL of potential covering vertices is determined (line 4). The candidate selection to form the RCL is based on the number of remaining demands to cover (vertices that do not cover any remaining demands are not considered). A portion α of the best candidate vertices is selected to form the RCL. Then, a vertex is randomly chosen in the RCL and added to the set of covering vertices (lines 5-6). Finally, covered demands are removed from the set of demands to cover (lines 6-7).

The second step, that spans the covering vertices with the minimum path heuristic, is described by Algorithm 3. In this procedure, the terminals to span correspond to covering vertices determined by the previous procedure (line 1). Since the result of the second step is a forest, the set of forest vertices is initialized with root vertices (line 3). In the algorithm's core, while all terminals are not included in the forest, a path from a forest vertex to a remaining terminal is computed and added to the forest. The added path, computed by the MinimumPath function (line 5), is a minimum cost path determined using Dijkstra's algorithm between forest vertices and remaining terminals. Note that the successive calls to Dijkstra's algorithm are optimized by re-using previous data of graph exploration.

This two-step greedy randomized construction procedure may have a limitation for solving the PFP. The procedure takes into account construction costs c_{ij} but harvesting costs d_{ij} are not directly used. Harvesting costs may be integrated in the heuristic function to choose the covering vertices in the first step. But, as the objective is to determine the minimum number of covering points with the minimum harvesting costs, the ratio between the number of covered demands and the harvesting costs is difficult to tackle as a heuristic function. In addition, it leads to inferior results than the adopted heuristic function based on the number of covered vertices. In any case, this

Fig. 2. Moves in neighborhood structures.

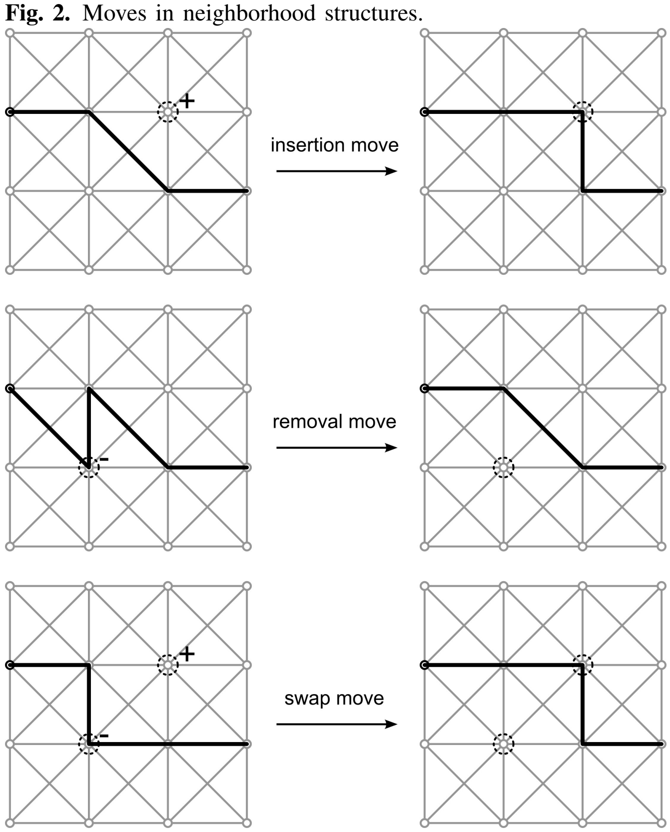
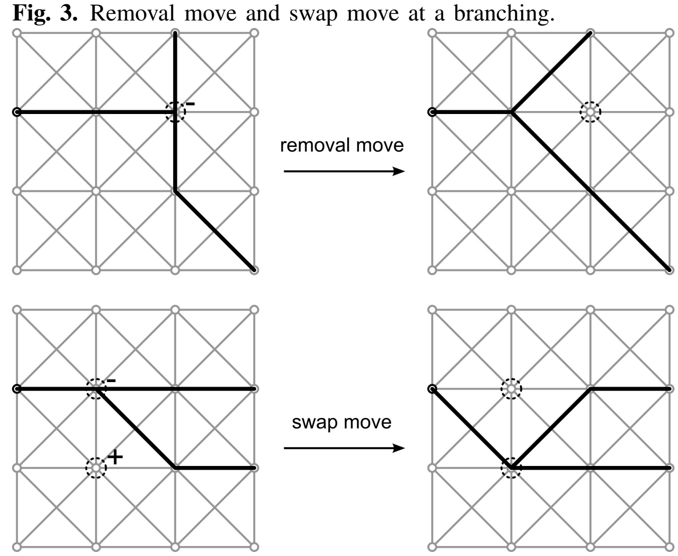


Fig. 3. Removal move and swap move at a branching.



limitation is resolved by applying a variable neighborhood descent procedure that considers both construction and harvesting costs.

3.3. Variable neighborhood descent procedure

Variable neighborhood descent is a local search procedure that uses several neighborhood structures [Hansen and Mladenović, 2003]. A neighborhood structure defines a set of local modifications or moves that can be applied to a solution in order to generate new solutions. A local search procedure uses such moves to explore the solution space. In the proposed implementation of GRASP, a variable neighborhood descent procedure is used to improve the cost of the solutions generated by the two-step greedy randomized construction procedure. Thus, the cost of an initial road network can be improved by applying successive small modifications. The implemented variable neighborhood descent exploits three neighborhood structures in which a new solution is obtained respectively by adding, removing or swapping (exchanging a vertex with another one) a vertex in an initial solution. On these three neighborhood structures, the neighborhood solutions considered are restricted to the solutions that satisfy constraints, and thus, removal or swapping operations are performed only if all harvest vertices are covered by the resulting roads. Moves using these neighborhood structures are illustrated in Figure 2.

For the three examples in Figure 2, the addition, removal or swapping of vertices is performed on a path, but these operations can also be performed at a leaf or a branching. For instance, Figure 3 illustrates a removal move and a swap move performed at a branching.

The three neighborhood structures are used in a variable neighborhood descent procedure described in Algorithm 4. In this procedure, the current solution is improved until a local optimum with regard to the three neighborhood structures is reached. Note that the evaluation of solutions considers both construction and harvesting costs, and thus, contrary to the greedy construction procedure the local search is guided by the entire objective function.

In the algorithm, $\mathcal{N}_k, k = \{1, 2, 3\}$ is the set of neighborhood structures. The variable neighborhood descent consists, first, in finding a local optimum using the first neighborhood structure $\mathcal{N}_1$. Then, the local descent continues with $\mathcal{N}_2$. If a better solution has been obtained with this neighborhood structure, the first structure is used for the next local descent, otherwise, it uses $\mathcal{N}_3$. This process continues until all neighborhood structures have been explored in a row with no improvement of the solution.

In order to obtain better computational time performance, the selection of a neighborhood solution (lines 6 to 10) uses a first improvement policy, i.e. the first neighbor whose cost value is smaller than that of the current solution is selected. In addition, the neighborhood structures are sorted by size as suggested in [Hu and Raidl, 2006], and thus, vertices swapping are performed first, then removal, and finally insertion.

4. Computational experiments

Three sets of experiments have been conducted to analyze the efficiency of the proposed approach. Small generated problem instances have been used to evaluate the adaptive parameter strategy of GRASP, and to compare the GRASP with results obtained with the commercial MIP solver ILOG CPlex. Real problem instances were used in the third experiment part aimed to evaluate the GRASP solutions against manually planned solutions. The goal of these experiments is twofold. First, the performance of GRASP is quantitatively evaluated on small instances. Second, the practical efficiency of the proposed approach is demonstrated on real instances.

Algorithm 4: VariableNeighborhoodDescent, variable neighborhood descent procedure

```
Data: graph, the graph in which roads have to be located.
      initialSolution, the initial road network to be
      improved by local search.
Result: s, an improved road network, which is a local
        optimum solution.
s ← InitialSolution
k ← 1
while k ≤ k_max do
    move ← false
    while HasNeighbor(s, N_k) and not move do
        s' ← NextNeighbor(s, N_k)
        if f(s) > f(s') then
            move ← true
            s ← s'
        end
    end
    if move then
        k ← 1
    else
        k ← k + 1
    end
end
return s
```

4.1. Adaptive parameter setting of GRASP

In comparison to a classical GRASP procedure, the implemented GRASP includes an adaptive parameter strategy presented in Section 3.1. This strategy allows to dynamically adapt the parameter α that determines the size of the restricted candidate list. In order to analyse the influence of this parameter α of GRASP and to evaluate the efficiency of the proposed adaptive parameter strategy, the results of the adaptive parameter strategy have been compared to the results of six fixed parameter values: 10%, 20%, 40%, 60%, 80% and 100%.

A benchmark containing 15 small size instances, with properties similar to real ones, have been generated. They contain between 25 and 100 vertices. Based on the analyses of real scenarios, the generated instances consist of graphs with a maximum of 8 connected road arcs to each vertex. The density of vertices to be harvested is very high on harvestable areas, and one to five percents of the vertices are existing roads or exit points. The average and standard deviation of the ratio between construction costs and harvesting costs on real instances have been used to generate the costs on small instances.

The GRASP was run with 200 iterations and the parameter E of the adaptive strategy set to 25. The benchmark composed of the 15 generated instances have been used for the computational experiments. The results on these instances are reported in Table 1 for the adaptive parameter strategy and the six fixed parameter settings.

For each problem instance, the table indicates the average deviation on ten runs to the best parameter setting considered. For instance, on problem number 11, the adaptive parameter strategy obtains an average deviation of 0.61% to the best parameter setting considered which corresponds to the α value of 10%. The best parameter settings are shown in bold type and naturally correspond to an average deviations of 0%. In addition, for each problem instance the rank of each parameter setting is indicated. This rank gives additional information on the performance of each setting independently of the size of the gaps.

For fixed values of α, the maximum average deviation attains 6% (instance number 3 with $\alpha = 10\%$). Considering all problem instances, no fixed parameter value really dominates the other ones. For instance, on problem number 6, small values of alpha give the best results, while on problem number 7 it is the larger values of alpha that give the best results. The setting $\alpha = 60\%$ gives the best average result for fixed α values, but has the fifth rank on three problem instances. These points confirm that the value of the parameter α has an impact on the results and, furthermore, that a fixed value of α does not provide a robust parameter setting for GRASP.

For the proposed adaptive parameter strategy, the number of problem instances for which this strategy obtains the best average deviation and the first rank is not higher than for the other parameter settings. However, the average deviation never exceed 1%. In addition, the adaptive strategy reaches the best overall average result with an average deviation of 0.23% and an average rank of 1.93. This result indicates that on the set of small instances the adaptive parameter strategy is more robust than the considered fixed parameter settings.

4.2. GRASP and exact approach

In a second set of experiments, the results of the implemented GRASP have been compared to the results of an exact approach. Generated instances presented in the previous section have been solved using ILOG CPlex 11 and the MIP formulation described in Section 2.3. The computational time was limited to ten hours on a PC with an Intel Core 2 Quad (2.83 GHz - 8Go of memory), using the default settings of the commercial solver. The best integer solutions obtained by branch-and-bound with CPlex are compared to the results of GRASP in Table 2. The GRASP was limited to 500 iterations on the same computer with E, the single parameter, sets to 25. The computer used for the experiments has four cores, and thus, can simultaneously run four threads. Both ILOG CPlex and the implementation of GRASP exploit this parallelism. The parallel implementation of GRASP consists in distributing the iterations among the available processors [Martins et al., 2000].

Computational results are reported in Table 2. Columns five and six respectively give the best integer solution values and computational times in seconds obtained by branch-and-bound on ILOG CPlex. The last three columns indicate the average computational results for GRASP on ten runs. The penultimate column reports the deviations between GRASP and branch-and-bound solution values. For the branch-and-bound approach, computational times become very important for solving instances containing 100 vertices. A value of 36 000 seconds indicates that the branch-and-bound failed to solve optimally the instance within the limit of ten hours of computation. This result indicates that a "naïve" branch-and-bound approach is impracticable for solving real-size problems. The maximum computational time for GRASP is 6.67 seconds, and the average time of 1.13 second to complete 500 iterations suggests that the implemented GRASP is suitable for solving large-size problems. Considering solution values, the GRASP

Table 1. Computational comparison of the adaptive parameter setting strategy and fixed parameter settings of GRASP on the 15 generated problem instances.

Instance		Adaptive α		Fixed α											
				$\alpha = 10\%$		$\alpha = 20\%$		$\alpha = 40\%$		$\alpha = 60\%$		$\alpha = 80\%$		$\alpha = 100\%$	
No.	Nb. vertices	Avg. dev. (rank)		Avg. dev. (rank)		Avg. dev. (rank)		Avg. dev. (rank)		Avg. dev. (rank)		Avg. dev. (rank)		Avg. dev. (rank)	
1	25	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
2	25	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
3	25	**0.00%**	(1)	6.00%	(7)	2.83%	(6)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
4	25	**0.00%**	(1)	5.45%	(6)	5.45%	(6)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
5	25	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
6	49	0.07%	(3)	**0.00%**	(1)	**0.00%**	(1)	0.20%	(6)	0.13%	(4)	0.23%	(7)	0.13%	(4)
7	49	0.68%	(3)	3.79%	(6)	3.79%	(6)	0.68%	(3)	0.68%	(3)	**0.00%**	(1)	0.30%	(2)
8	49	**0.00%**	(1)	3.02%	(7)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
9	49	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
10	49	0.88%	(4)	4.88%	(7)	3.18%	(6)	1.91%	(5)	0.61%	(3)	0.38%	(2)	**0.00%**	(1)
11	100	0.61%	(3)	**0.00%**	(1)	0.51%	(2)	1.10%	(4)	1.80%	(5)	1.85%	(6)	2.25%	(7)
12	100	0.53%	(4)	0.40%	(3)	**0.00%**	(1)	0.13%	(2)	0.70%	(5)	1.07%	(6)	1.37%	(7)
13	100	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)	**0.00%**	(1)
14	100	0.51%	(2)	**0.00%**	(1)	0.77%	(3)	1.10%	(6)	0.95%	(5)	1.55%	(7)	0.80%	(4)
15	100	0.24%	(2)	3.86%	(7)	1.20%	(6)	0.39%	(4)	**0.00%**	(1)	0.33%	(3)	0.49%	(5)
Average		0.23%	(1.93)	1.83%	(3.40)	1.18%	(2.87)	0.37%	(2.53)	0.33%	(2.27)	0.36%	(2.67)	0.36%	(2.53)

Table 2. Computational results on generated instances comparing branch-and-bound (ILOG CPlex) and GRASP.

Instance				Branch and bound			GRASP (500 it.)		
No.	Nb. vertices	Nb. demand points	Nb. roots	Best found		Duration (sec.)	Avg. value	Dev. (%)	Duration (sec.)
1	25	24	1	8 534.00	*	1.20	8 534.00	0.00%	0.33
2	25	24	1	5 933.00	*	0.22	5 933.00	0.00%	0.19
3	25	24	1	8 513.00	*	1.64	8 513.00	0.00%	0.25
4	25	23	2	5 197.00	*	0.14	5 197.00	0.00%	0.17
5	25	23	2	8 530.00	*	8.36	8 620.00	1.06%	0.26
6	49	48	1	15 066.00	*	1 231.81	15 066.00	0.00%	0.62
7	49	48	1	14 257.00	*	125.63	14 257.00	0.00%	0.34
8	49	47	2	13 918.00	*	217.31	13 918.00	0.00%	0.36
9	49	47	2	11 646.00	*	99.83	11 758.00	0.96%	0.22
10	49	46	3	18 360.00	*	16 728.17	18 632.00	1.48%	0.60
11	100	99	1	34 050.00		36 000.00	34 700.30	1.91%	1.96
12	100	97	3	29 254.00		36 000.00	27 922.50	-4.55%	1.38
13	100	97	3	24 405.00		36 000.00	24 405.00	0.00%	1.35
14	100	96	4	37 360.00		36 000.00	39 165.00	4.83%	2.25
15	100	96	4	24 334.00		36 000.00	24 388.40	0.22%	6.67
Average						13 227.62		0.39%	1.13

* Optimal solution value

reaches an optimal solution for seven of the ten instances optimally solved using branch-and-bound. The deviations between GRASP and branch-and-bound solution values do not exceed five percent. In addition, the GRASP reaches a deviation of 0.39% on average in a few seconds. These results illustrate the efficiency of the implemented GRASP in terms of solution quality and computational times.

4.3. Real instances

Three harvesting maps located in Quebec have been considered to evaluate the GRASP on real instances. The studied maps cover approximately $3\ 000$ hectares each, which correspond to a graph containing $12\ 614$ vertices for the largest instance. These maps correspond to sectors within larger forest management units. A size of $3\ 000$ hectares seems representative of sectors' size in the province of Quebec. However, sizes may vary from less than 50 hectares for specific tree species, to more than $20\ 000$ hectares in case of agreements with the Government. A sector covers several cutting blocks that share the same primary road network. For the three considered maps, the sectors are units in which operational decisions are taken.

The inputs grids of these real problem instances have been produced using PlaniRoute [FPInnovations] a commercial software of FPInnovations for which a demo version is available upon request. In addition, real cost parameters have been used to generate the graph.

From a planner's point of view, the first map is the easiest scenario among the three scenarios. Soil type, drainage and slope are adequate for road construction, except for some rocky hillsides. The second map contains many steep slopes, but the deposits are sufficient. It is a map of average difficulty. The last one is very difficult due to the steep terrain and the low deposit.

For the three problems, forest roads have already been located using a GIS by an expert in forestry. The GIS used for these manually planned solutions gives an estimation of road construction and harvesting costs to guide the planner, but do not provide a computerized method to optimize the location of roads. Note that the three manually planned solutions considered correspond to forest roads that have been designed in a real situation, independently of this study, and implemented with some possible minor adjustments in the field. In order to compare the costs of manual and computerized solutions, the road networks manually designed were evaluated using the objective function on the implemented software. Covering constraints are not entirely respected in manual solutions, and in these cases the costs to harvest uncovered areas are not taken into account. Therefore the manual solutions give a lower estimation of the real cost, which is not the case for the GRASP approach. The GRASP was run using the same computer and parameter setting than for previous experiments on generated instances and computation time was limited to two hours.

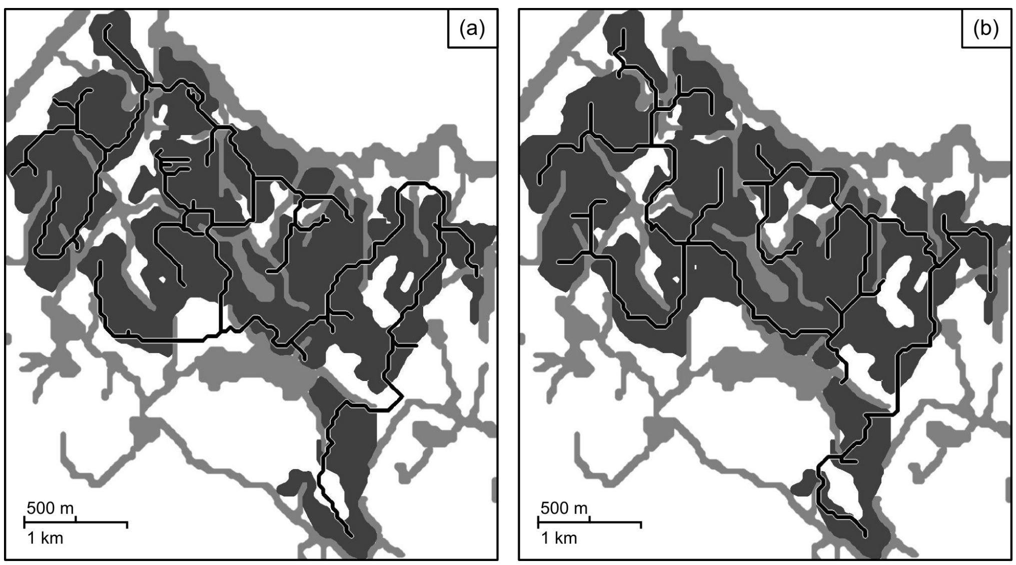
The costs of manual solutions are compared with the ones from the GRASP in Table 3. The evaluation considers both construction and harvesting costs. In the table, the Gap columns indicate the difference in percentage between the solutions manually obtained and the computerized method. The GRASP obtained an average gap of 11%, i.e. for these three real instances, the GRASP generates solutions whose cost is on average 11% lower than the solutions manually planned. Manual and computerized solutions for the third problem instance are represented in Figure 4. For this scenario, the computerized solution reduces the cost of the road network by 8.37%. This significant gap is obtained by reducing the total length of the roads by almost 3 kilometers and reducing the number of water crossing structures. Even if the computerized solutions may need to be adjusted by an expert, they are good initial solutions to support decisions. In addition, these solutions allow saving time for the planner in the design of an initial network.

The advantages of the GRASP approach reported by the expert are, first, to suggest globally good solutions that respect slopes, building difficulty and coverage constraints. Then, the

Table 3. Results on real instances comparing manual and GRASP approaches.

	Map		Manual	GRASP	
	Map size (ha)	Nb. vertices	Solution cost ($)	Solution cost ($)	Gap Manual vs. GRASP (%)
Laurier	2 900	8 084	408 882	364 039	10.97%
Meunier	2 881	11 526	697 329	599 699	14.00%
Parent	3 153	12 614	700 381	641 788	8.37%
Average					11.11%

Fig. 4. Solutions for scenario 3 "Parent", (a) manual solution, (b) GRASP solution. Light gray areas correspond to construction or harvest barriers such as lakes and rivers. Dark gray zones correspond to areas to be harvested. Black lines are the access roads manually designed on the left image, and GRASP solution on the right image.



heuristics ensure a good balance between harvesting costs and construction costs, and thus generate neither too much nor too few forest roads. Finally, solutions with lower costs than manual ones are obtained in a reduced amount of time.

5. Conclusion and perspectives

This paper presented a heuristic approach for solving a network design problem in the field of forestry. The problem was modeled as a P-Forest Problem (PFP) that belongs to the category of extensive facility location problems. It consists in determining a set of tree structures in a graph that covers a set of vertices so as to minimize the harvesting and construction costs. The proposed heuristic approach for solving the PFP is a Greedy Randomized Adaptive Search Procedure (GRASP). This metaheuristic consists in iteratively constructing a solution and improving it using a local search procedure. A two-step greedy construction procedure and a variable neighborhood descent procedure are respectively used for the construction phase and the improvement phase of the GRASP.

Three sets of experiments were conducted to analyze the efficiency of the proposed approach. Small generated problem instances were used to compare the GRASP with results obtained by branch-and-bound. In addition, an experiment was performed on real problem instances and aimed at evaluating the GRASP against manually planned solutions. The GRASP leads to significant improvements both in solution quality and computational time in comparison to manual solutions.

The proposed approach was implemented on a decision support system and can be effectively used to help the design of forest roads. However, due to the complexity of the studied instances, some aspects of the solutions provided by the heuristic are not yet realistic or convenient for a direct implementation of the roads. Based on observations of experts in forestry, the main adjustments needed concern the location of water-crossings, the adjustment of some roads to allow the logs to be dragged in the direction of the slope, and the definition of shorter road connections to exit points. Even if specific parts of solutions necessitate a manual edition by an expert, the GRASP provides solutions that respect slopes, construction difficulty and coverage constraints. Then, the heuristic ensures a good balance between harvesting costs and road construction costs. And finally, low costs solutions are generated in a reduced amount of time.

Future works will focus on two aspects. First, we will introduce on-road transportation costs. This additional cost is expected to improve the validity of generated solutions by penalizing long road connections. It will also improve the accuracy of the evaluation of costs. Second, we will consider an interactive heuristic approach to deal with the complexity of the problem. Contrary to the GRASP, which is a fully automated approach, in the interactive heuristic approach the user may contribute to the optimization process. The objective is to exploit the problem-domain expertise of the user in order to generate more realistic solutions that integrate aspects not captured by the objective function.

Acknowledgements

The authors would like to thank the reviewers for their valuable comments and suggestions that helped to improve this manuscript. The authors would also acknowledge the participation of Jean Favreau as well as the contribution of FQRNT, MITACS and NSERC discovery grant.

References

A. E. Anderson and J. Nelson. Projecting vector-based road networks with a shortest path algorithm. *Canadian journal of forest research*, 34:1444–1457, 2004. doi: 10.1139/x04-030.

A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. The multi-level network design problem. Technical Report 3366-91-MSA, Sloan School of Management M. I. T., 1991.

B. Boffey and S. C. Narula. Models for multi-path covering-routing problems. *Annals of Operations Research*, 82(12): 331–342, 1998. doi: 10.1023/A:1018923022243.

W. Chung, J. Stückelberger, K. Aruga, and T. W. Cundy. Forest road network design using a trade-off analysis between skidding and road construction costs. *Canadian Journal of Forest Research*, 38:439–448, 2008.

M. M. Clark, R. D. Meller, and T. P. McDonald. A three-stage heuristic for harvest scheduling with access road network development. *Forest Science*, 46(2):204–218, 2000.

S. D'Amours, M. Rönnqvist, and A. Weintraub. Supply chain planning of the forest product industry using operations research. Technical Report CIRRELT-2007-52, CIRRELT, 2007.

D. J. Dean. Finding optimal routes for networks of harvest site access roads using gis-based techniques. *Canadian Journal of Forest Research*, 27(1):11–22, 1997. doi: 10.1139/x96-144.

R. Epstein, A. Weintraub, P. Sapunar, E. Nieto, J. B. Sessions, J. Sessions, F. Bustamante, and H. Musante. A combinatorial heuristic approach for solving real-size machinery location and road design problems in forestry planning. *Operations Research*, 54(6):1017–1027, 2006. doi: 10.1287/opre.1060.0331.

FPInnovations. Planiroute [online], available from `http://www.feric.ca/planiroute-en`. [accessed 10 March 2012].

C. González, J. A. Lozano, and P. L. naga. Mathematical modelling of umdac algorithm with tournament selection. behaviour on linear and quadratic functions. *International Journal of Approximate Reasoning*, 31:313–340, 2002.

Gouvernement du Québec. Regulation respecting standards of forest management for forests in the domain of the state, r.q. c. f-4.1, r.7, 2010.

P. Hansen and N. Mladenović. *Handbook of Metaheuristics*, chapter Variable Neighborhood Search, pages 145–184. Kluwer Academic, 2003.

B. Hu and G. R. Raidl. Variable neighborhood descent with self-adaptive neighborhood-ordering. In *7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics*, 2006.

V. A. Hutson and C. ReVelle. Indirect covering tree problems on spanning tree networks. *European Journal of Operational Research*, 1(1):20–32, 1993. doi: 10.1016/0377-2217(93)90141-9.

F. K. Hwang, D. S. Richards, and P. Winter. *The steiner tree problem*, volume 53 of *Annals of discrete mathematics*. Elsevier, 1992.

J. Karlsson, M. Rönnqvist, and J. Bergström. An optimization model for annual harvest planning. *Canadian Journal of Forest Research*, 34(8):1747–1754, 2004.

T. U. Kim, T. Lowe, A. Tamir, and J. Ward. On the location of a tree-shaped facility. *Networks*, 28:167–175, 1996. doi: 10.1002/(SICI)1097-0037(199610)28:3¡167::AID-NET5¿3.0.CO;2-L.

M. Labb, G. Laporte, and I. Rodríguez-Martín. *Fleet Management and Logistics*, chapter Path, tree and cycle location, pages 187–204. Kluwer, 1998.

A. D. Legues, J. A. Ferland, C. C. Ribeiro, J. R. Vera, and A. Weintraub. A tabu search approach for solving a difficult forest harvesting machine location problem. *European Journal of Operational Research*, 179(3):788–805, 2007. doi: 10.1016/j.ejor.2005.03.071.

S. L. Martins, M. G. C. Resende, C. C. Ribeiro, and P. M. Pardalos. A parallel grasp for the steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, 17:267–283, 2000. doi: 10.1023/A:1026546708757.

J. A. Mesa and T. B. Boffey. A review of extensive facility location in networks. *European Journal of Operational Research*, 95:592–603, 1996. doi: 10.1016/0377-2217(95)00321-5.

A. T. Murray. Route planning for harvest site access. *Canadian Journal of Forest Research*, 28(7):1084–1087, 1998. doi: 10.1139/x98-122.

M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. Technical report, Illinois Genetic Algorithms Laboratory, 1999.

M. Prais and C. C. Ribeiro. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing*, 12(3):164–176, 2000. doi: 10.1287/ijoc.12.3.164.12639.

M. G. C. Resende and C. C. Ribeiro. *Handbook of metaheuristics*, chapter Greedy randomized adaptive search procedure, pages 219–249. Kluwer, 2003.

M. Rönnqvist. Optimization in forestry. *Mathematical programming*, 97:267–284, 2003.

J. Sessions, A. Akay, G. Murphy, W. Chung, and K. Aruga. *Computer Applications in Sustainable Forest Management*, chapter Road and harvesting planning and operations, pages 83–99. Springer, 2006.

J. Stückelberger, H. R. Heinimann, and W. Chung. Improved road network design models with the consideration of various link patterns and road design elements. *Canadian Journal of Forest Research*, 37:2281–2298, 2007.

H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Japonica*, 24(6):573–577, 1980.

A. Tamir and T. J. Lowe. The generalized p-forest problem on a tree network. *Networks*, 22:217–230, 1992.

A. Weintraub and A. T. Murray. Review of combinatorial problems induced by spatial forest harvesting planning. *Discrete Applied Mathematics*, 154(5):867–879, 2006. doi: 10.1016/j.dam.2005.05.025.