CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Interactive Planning System for Forest Road Location

David Meignan
Jean-Marc Frayret
Gilles Pesant

June 2013

CIRRELT-2013-37

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Interactive Planning System for Forest Road Location

David Meignan[1], Jean-Marc Frayret[2,3,*], Gilles Pesant[2,4]

1 Department of Mathematics and Computer Science, University of Osnabrück, Albrechtstraße 28a, 49069 Osnabrück, Germany

2 Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

3 Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7

4 Department of Computer Engineering, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7

Abstract. This paper presents an interactive planning system for forest road location. This decision support system is based on an interactive heuristic approach, within which the user contributes in a cooperative manner to the optimization process. The objective of this cooperative optimization process is to exploit the problem-domain expertise of the user in order to, on the one hand, guide the search for a solution towards intuitively interesting parts of the solution space, and, on the other hand, generate more practical solutions that integrate aspects of the decision problem that are not captured by the heuristic objective function. This paper more specifically presents the user interface, the interaction mechanisms and the heuristic developed to support the cooperation between the computer and the user. We also present experimental results based on real problem instances, with an expert user. A comparison shows clear advantages for using the proposed interactive approach over a pure manual or pure automated approach.

Keywords: Interactive optimization, heuristics, forest road location, decision support.

Acknowledgements. The authors would like to thank Mathieu Blouin and Jean Favreau as well as the contribution of Fonds de recherche du Québec - Nature et technologies (FRQNT), Mitacs, and Natural Sciences and Engineering Research Council of Canada (NSERC) for having supported the project under the discovery grant program.

---

* Corresponding author: Jean-Marc.Frayret@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1 Introduction

Heuristic optimization methods have been proven successful to solve a variety of planning problems in areas such as transportation, logistics, manufacturing, energy and telecommunication (Ibaraki et al. 2005). In particular, Advanced Planning and Scheduling (APS) represent promising applications for heuristics considering the complexity and richness of underlying combinatorial optimization problems. However, they have several limitations with respect to solution implementation in practical applications. First, any deviation or simplification between the implemented optimization model and the real decision problem may lead to inconsistencies, which, in turn may lead to unpractical or even infeasible decisions. Multi-stakeholder and multi-criteria decision contexts are typical examples of decision problems, which require some form of adjustment or negotiation of tentative decisions in order to meet all constraints and preferences. Along the same line, the discrepancies between the information required and the information known a priori at the time of decision-making may lead to similar difficulties. For instance, some information may only be known when users analyze a tentative decision, which trigger their awareness or memory, which, in turn, can lead to the adjustment or rejection of a tentative decision. Similarly, heuristic optimization methods may benefit from users' ability to quickly analyze the general structure and performance of a solution and propose adjustments to improve it. For such reasons, APS systems have evolved from optimization with what-if analysis to fully interactive plan builder using interactive heuristics in order to take advantage of human expertise and tacit knowledge.

The planning of forest road location, which involves deciding the precise location of a non-existing road network, is a decision problem with several sources of explicit and implicit information, multiple objectives, complex constraints and multiple stakeholders. In order to address this rich decision problem, this paper proposes a series of general design principles to develop, and ultimately evaluate with real problem instances, an interactive optimization method.

This paper is organized as follow. Section 2 proposes a literature review of interactive optimization. Next, Section 3 introduces the general problem of forest road construction, and the specific decision problem addressed in this paper, and presents a general model of that problem. Section 4 presents the general design principles, as well as the interactive planning system for forest road location. Finally, Section 5 presents computational experiments, and Section 6 summarizes this work and identifies promising research directions in interactive optimization.

2 Overview of interactive optimization

Although it is a rather sparse research topic, several interactive optimization methods have been proposed in the literature. This section proposes an overview of the most prominent interactive optimization approaches. The interested reader is referred to Meignan et al. (2013) for an extended analysis of interactive optimization methods. Meignan et al. (2013) define Interactive optimization approaches as "optimization methods with which an end-user or decision-maker can interact". In other words, such methods combine, on the one hand, the process of finding an optimal or good solution to an optimization problem, and, on the other hand, the process of adapting such a solution to a practical decision-making context. With these optimization methods, the user, or users with different objectives or problem perceptions, can significantly modify the outcome or the performance of the optimization process. Therefore, this review focuses solely on interactive optimization methods in which the interactions take place between human decision-makers (also referred to as users) and optimization systems. Several classes of approaches have been identified.

2.1 Trial-and-error

The simplest ways to involve users in an optimization process is trial-and-error, also referred to as what-if analysis. It is a form of interactive optimization that involves some form of iterative and direct adjustment of optimization parameters by the user. However, it remains a simplistic interactive approach because the manual process of interacting with the solution is completely separated from the optimization process. In other words, such interactive methods can be implemented with any optimization system because the user's feedback only concerns input values of the optimization process. In this interactive process, there is no specific feedback information that could be used to guide the next steps of optimization. Similarly, the system does no attempt to generalize the user's actions to infer his or her preferences to guide the next steps of optimization. Trial-and-error approaches can be used to adjust optimization parameters including weights in a multi-objective decision problem. In trial-and-error approaches, the user constructs a mental model of the decision problem on hand, and learns about the relationship between parameters and response of the system. This implicit knowledge allows the user to progressively adjust the values of some parameter in order to meet his or her preferences, or explore different solutions.

Although it is quite simple and intuitive to implement, this forms of interaction has several weaknesses. First, the effect of parameters as well as the functioning of the optimization process might not be understood appropriately by the user. Second, the process of adjusting parameters may be rather long to focus the search of a solution towards the user's preferences. Therefore, only a limited number of parameters can actually be adjusted in such a manual manner. Finally, the adjustment of some parameters may require specific knowledge about the optimization method or the optimization model that may not be easily understandable by the user.

In order to mitigate these aspects, appropriate visualizations of results are necessary to support the trial-and-error process of adjusting the solution. Some visualization tools and graphical representation have been proposed in the context of optimization systems (Jones 1997; Miettinen 2012).

2.2 Interactive evolutionary computation

Interactive evolutionary computation is another class of interactive optimization methods. The reader is referred to Takagi (2001) for an extensive review of interactive evolutionary applications. Interactive evolutionary computation address optimization problems for which an objective function is difficult to model. In other words, it addresses problems that require the user's subjective evaluation of solutions.

More specifically, once the optimization process has generated a first set of solutions, the user evaluates solutions according to his or her own perception of their value. This evaluation provided by the user determines the new set of solutions selected for the next generation. Then, mutation and crossover operators are applied on the selected solutions to form a new population, in a similar manner as standard evolutionary algorithms. Randomly generated solutions can also be inserted to ensure diversity. Next, this new population is presented to the user for another iteration of evaluation. The process is repeated until the user identifies a final solution (Banzhaf 1997), as presented hereafter in Algorithm 1.

Algorithm 1 General interactive evolutionary procedure

```
generate the initial population of solutions
do until a stopping criterion is met or the user stops the process
    user evaluation of solutions
    selection of solutions for the new population
    crossover and mutation operations
end
return preferred solution
```

The main difference between trial-and-error and interactive evolutionary computation concerns the embeddedness of the optimization process and the manual process of interacting with the solution. Interactive evolutionary computation explicitly considers the user feedback within the optimization process, which is not the case in the previous approaches. More specifically, solutions are either discarded or selected by the user, before the process is repeated. However, unlike more advanced forms of interaction (see next sub-sections), there is no explicit user preference to guide the next generation of solutions. Therefore, the cognitive burden of the evaluation process for the user can be quite high. In a standard evolutionary algorithm, this evaluation process requires generally a large number of evaluations to converge to interesting regions of the search space. Consequently, different variants of interactive evolutionary algorithms have been proposes to address this issue. For instance, Banzhaf (1997) suggests adding a sorting process for pre-selecting a sub-set of solutions that is evaluated by the user. The fitness values are then generalized to the entire population of solutions. Lee and Cho (1999) use a clustering method to select some representative solutions that are evaluated by the user. Another approach to improve this evaluation process consists in learning a model of the user subjective evaluation, which is used to generalize the evaluation of new solutions. Different methods have been investigated to learn the user's fitness, such as artificial neural networks (Biles et al. 1996), support vector machines (Llorà et al. 2005) and case-based reasoning (Babbar-Sebens and Minsker 2010).

2.3 Interactive multi-objective optimization

Interactive multi-objective optimization (Branke et al. 2008) is maybe the most prominent class of interactive methods in the combinatorial optimization literature. The main principle of these methods, and the main difference with the other approaches, is to use the user's preference, with respect to the objectives, in order to guide the search for a Pareto optimal or good approximate solution. This also reduces the computational time as only a sub-set of Pareto optimal solutions are investigated (Miettinen et al. 2008). Several interactive approaches based on exact or heuristic search procedures have been proposed. In Miettinen et al. (2008), the authors differentiate between three categories of interactive multi-objective optimization methods according to the type of preference information provided, trade-off based methods, reference point approaches, and classification-based methods. For these three categories of methods, the basic principle of the interactive search is similar, although the information fed back by the user is different. This process consists in the following steps:

Algorithm 2 General interactive multiobjective optimization procedure

```
initialize user's preference model
generate initial solution (or set of solutions)
do until a stopping criterion is met or the user stops the process
   user evaluates solution(s) or provide preference on objective functions
   update of the user's preferences model
   generation of new solution(s)
end
return preferred solution
```

In trade-off based methods, the user's preference is expressed as relative variations of objective values between two solutions. Trade-off values can be provided by the user and then used to direct the search in the region of interest. This preference information is called subjective trade-off. Differently, trade-offs can also be obtained by comparing feasible solutions. In this case, the user's preference consists in an evaluation of these feasible trade-offs. For example, in the Guided Multi-Objective Evolutionary Algorithm (Branke et al. 2001), the user provides the maximum and minimum desired trade-off (unit objective degradation) between each pair of objectives in order to guide the search. The trade-off based method can also be implemented in the interior point method to solve multi-objective linear programming problems (Junior and Lins 2009). In this approach, the user observes the path of interior solutions and defines iteratively the step length of the general objective functions and the growth portion of each individual objective function in order to guide the search toward a non-dominated solution.

In reference point approaches, the user specifies his preferences with desired values, or ranges, for each objective. The goal of this type of approach is to find a solution as close as possible to the desired reference points, which are influenced by the intermediate solutions generated. Jaszkiewicz and Stowinski (1999), Deb and Sundar (2006), Miettinen et al. (2010), and Katagiri et al. (2008) propose examples of such interactive approaches.

Finally, in classification-based methods, the user identifies iteratively the objective function that should be improved, as well as the objective functions that could be deteriorated with respect to a given intermediate solution. Therefore, this preference information is expressed by classifying objective functions. The user may also indicate bounds for the desired or maximum variation of an objective function. For instance, in the NIMBUS method, Miettinen and Mäkelä (2000) propose five classes used by the user to classify each objectives function. The optimization process then uses this information to iteratively alter the current solution so as to meet the user's preference.

2.4 Human-guided search

The last class of interactive approaches presented in this overview aims at supporting the optimization by adding new information to the optimization process, which is not entirely captured by the explicit decision-making model. More specifically, in trial-and-error, interactive evolutionary computation, and interactive multi-objective optimization, the contribution of the user aim at enriching the decision model during the optimization process by, respectively, modifying the parameters defining the decision problem, providing the implicit evaluation of solutions, and adjusting the importance of an objective function.

Contrary to these classes of approaches, human-guided search procedures propose to improve the local search of the optimization process, by adding new heuristic information related to the specific instance of decision problem (Klau et al. 2010). Such a procedure is based on the assumption that the user knows specific characteristics of the decision problem that are not exploited by the automated optimization algorithm. Such a combination between human and computer solving strategy have been investigated in early works (Krolak et al. 1970). They generally consist in the steps presented in Algorithm 3.

Algorithm 3 General human-guided search procedure

```
generate initial solution
do until a stopping criterion is met or the user stops the process
   user provides search preference with respect to the current solution
   improve current solution
end
return preferred solution
generate initial solution
```

As suggested previously, human-guided search are local-search heuristics or metaheuristics. Like interactive evolutionary and multi-objective optimization methods, they alternate between an automated optimization process and a manual process of interacting with the solution, during which the user can express some preferences. More specifically, the user can restrain the search space by selecting the parts of the current solution where the search should focus on. For instance, in the method propose in Klau et al. (2010), the user can attribute different degrees of mobility and penalties to the parts of the solution in order to specify which parts are satisfactory and which ones need to be re-optimized. A local-search procedure is then applied to improve the current solution. This procedure can be viewed as an interactive variant of the guided local search proposed by Voudouris and Tsang (2003). The interactive optimization application described in this paper is another example of human-guided search, although it is different from this latter. Similarly, Hao and Miller-Hooks (2006) propose an interactive heuristic for the vehicle routing problem with solution shape constraints (i.e., visual attractiveness of tours) in order to maximize tour acceptance and also to exploit the expert knowledge of managers to design efficient tours. In this approach, the user guides the search for a solution through the iterative setting and adjusting of geographical reference points around which routes are heuristically built.

This paper proposes an interactive planning system, in which users contribute to the optimization process by iteratively adding constraints to the problem and by identifying the parts of the current solution to involve in the next iteration of optimization. The core interactive process between the optimization method and the user is similar to the above-mentioned Human-Guided Search approach. However, the proposed interactive planning system considers different levels of interaction. Before describing the proposed interactive solution approach, the following section presents the optimization problem addressed in this paper and its formal model.

3 Forest road location planning

The decision problem considered in this paper is a network design problem that consists of locating the roads that will be used to transport trees or logs from harvested areas to processing plants or intermediate storage. This problem appears at different decision levels of forest harvest planning. For instance, at the tactical decision level, the problem consists in determining timber flows on an existing road network in order to satisfy demand with the forest products supply chain (D'Amours et al. 2008). Here, the objective is to minimize transportation and road maintenance costs over several time periods (Karlsson et al. 2004). The problem that is addressed in this paper concerns the operational planning of road location in a context of a non-existent road network, as presented in Meignan et al. (2012). The objective is to locate new access roads that connect the existing road network to harvest areas. Several constraints must be considered including environmental, topological and soil constraints. As far as road construction is concerned, the two main cost functions are harvesting, which include logging and hauling to roadside costs, and road construction. However, these two costs result in two conflicting objectives that should not be addressed separately. Indeed, a dense road network leads to a lower harvest cost (thanks to shorter hauling to roadside distances), but a higher road construction cost. Therefore, the design of a forest road network must achieve a trade-off between harvest cost and construction cost (Chung et al. 2008), which is a challenging aspect of the problem.

Similar decision problems have been studied in the literature. Clark et al. (2000) investigate the problem of access road network design in combination with the problem of scheduling the harvesting of stands, which involves the temporal planning of road building and stand harvesting. 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 and proposes some heuristic methods based on the minimum path heuristic for the Steiner tree problem. Along the same line, Anderson and Nelson (2004) and Stückelberger et al. (2007) propose some heuristics based on the minimum path heuristic to solve the same problem, but only consider construction cost, which reduces the problem to a Steiner tree problem in a graph. Differently, Epstein et al. (2006) and Legües et al. (2007) investigate the machinery location and road network design problem, which consists in determining the location of harvesting machinery and the location of access roads considering topographical constraints, harvest costs, and construction costs. Finally, Meignan et al. (2012) propose a first efficient solution approach based on GRASP to the specific problem addressed in this paper.

Although harvest cost and construction cost are the main costs of the road construction problem, ignoring the cost of logs transportation to the plants may lead to road networks with several transportation inefficiencies. However, including the transportation cost in the objective function of this problem leads to some difficulties. More specifically, the computation of the transportation cost is, in itself, an optimization process, which includes selecting for each harvest elementary cell (see further), the optimal hauling to roadside path (i.e., distance). This can only be computed for a complete road network, which may in part explains why transportation cost is generally ignored in the literature for this specific problem. However, because expert users with appropriate visualization tools can easily identify solutions with high transportation cost, one approach to deal with the difficulties of adding transportation cost to the decision model, is to use their expert knowledge and the experience to guide the search for a solution. More specifically, expert users can easily identify the parts, or road segments, of a solution with higher contributions to transportation cost. The underlying idea is to use this information in the optimization process to search specific regions of the solution space in order to find a trade-off between, harvest cost, road construction cost and transportation cost.

3.1 Problem definition

Decisions about access roads construction are part of an operational planning process. In practice, a forest engineer, with the support of a Geographic Information System (GIS), elaborates a plan that layouts the road network on the map. This process requires a large amount of information stored in the GIS, including the areas to harvest, topological and soil data, the existing road network, and forest characteristics. Here, soil type; drainage and slope are the important criteria to determine roads feasibility and construction costs. In this work, they are aggregated into a road construction difficulty index, which is computed by *PlaniRoute*, a GIS-based planning support system (FPInnovations 2013).

3.2 Problem modeling

The forest road location problem can be assimilated to a P-Forest Problem (PFP) (Tamir and Lowe 1992). The PFP is an extensive facility location problem where facilities to locate in a graph are trees (Mesa and Boffey 1996). As modeled in Meignan et al. (2012), the problem aims at determining a set of tree-like paths of minimum cost in a graph that indirectly cover a set of demand points. A demand point located on a vertex is indirectly covered, if the distance to the closest path in the solution is lower or equal to a maximum coverage distance. Here, arcs represent road segments, while demand points represent harvest elementary cells, and the coverage distance represents the maximum skidding distance.

The problem is formally defined as follows. Let $G(V,E \cup A)$ be a bi-level graph with V the set of vertices, E and A two sets of directed arcs. The set of arcs E are potential forest road segments and A are covering arcs. Two subsets of vertices, $R \subset V$ and $B \subseteq V$ are defined. R is the set of vertices that can serve as root for tree-like path, and B is the set of demand points to cover. For each demand point i □□□, w_i denotes the cost per distance unit to cover the demand points. With each edge in E is associated a cost c_{ij} corresponding to the construction cost along the edge (i,j). For the set of edges A, the value d_{ij} is the distance to cover the demand point j from vertex i. Finally, D is the maximum coverage distance.

A feasible solution is a set of tree-like path such that, (a) each tree starts at a root vertex in R, (b) each demand is covered by a root vertex or a vertex in the set of trees, (c) each cover distance is lower than the maximum coverage distance D. The objective explicitly modeled in the system is to minimize the sum of construction and coverage (i.e. harvest) costs. In this model transportation cost is estimated automatically and taken care of by the user. Two binary decision matrices x_{ij} and y_{ij} are introduced to represent the set of trees (i.e, road network) and the allocations of the demand points, as follows:

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

$$y_{ij} = \begin{cases} 1 \text{ if vertex } i \text{ serves the demand point } j \\ 0 \text{ otherwise} \end{cases}$$

We also introduce flow variables z_{ij} that correspond to the total demand that pass through the road arc (i,j), which are used to estimated transportation cost. The PFP can be stated as follows:

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

Subject to:

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

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

$$x_{ij} T \geq z_{ij} \qquad \forall (i,j) \in E \qquad (4)$$

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

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

$$y_{ij} d_{ij} \leq D \qquad \forall (i,j) \in A \qquad (7)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \qquad (8)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (9)$$

$$z_{ij} \in \{0, \dots, T\} \qquad \forall (i,j) \in E \qquad (10)$$

The left part of the objective function corresponds to the construction cost and the right part is the harvest cost. Constraints (2) to (4) ensure the arborescent structure of the road network. More precisely, the flow balance is guaranteed with constraint (2), constraint (3) corresponds to flow termination, and constraint (4) connects flows with road segments to construct. The parameter T corresponds to the sum of all demand quantities w_i. Constraints (5) and (6) ensure that all demand points are covered. The maximum coverage distance is defined by constraint (6). This coverage distance constraint can be removed if the arcs in the set A are properly selected. Finally, constraints (8) to (10) require all decision variables to be integer with specific bounds.

In order to use this graph-based representation of the problem, the forest map is discretized into 50-meter wide square cells. Using this grid, data are then projected to a graph. The cells to harvest correspond to demand points, existing roads are root vertices and maximum harvesting distance is considered as the maximum coverage distance. Construction costs and harvest costs are computed to reflect costs and constraints of road construction and harvest operations. These costs aggregate a large panel of legal, environmental, geographical and operational data. The objective is to consider both environmental and economic aspects in the optimization problem. For the experiments presented in Section 5, the inputs grids were produced using *PlaniRoute*, and actual cost parameters have been used.

3.3 Problem issues and practical application

This P-forest problem is NP-Hard (Epstein et al. 2006). Therefore finding the optimal solution to real problem instances in reasonable amount of time is not practical. Furthermore, this specific decision context includes several objectives to be solved simultaneously, only part of which, as mentioned earlier, is formally captured in the P-Forest problem formulation. Objectives of the real decision problem include minimizing road construction cost (i.e., minimizing road length and water crossing), minimizing harvest cost (i.e., minimizing total skidding distance to road side) and minimizing transportation cost (i.e., minimizing total distance to haul the entire harvested volume).

If road construction cost minimization is simple to compute, harvest cost minimization is more complex. Indeed, it is necessary to find the shortest skidding distance to roadside for each harvest elementary cells. Similarly, transportation cost, which is ignored in the modeling of the P-Forest problem, can only be computed once a complete road network is found, as well as all shortest skidding point for all harvest cells (i.e., points of the road network where a harvest cell is skidded to). Furthermore, solution feasibility and quality require to be evaluated through qualitative and quantitative analysis with respect to several ground and other environmental characteristics, such as soil type and habitat protection, some which are aggregated into a single parameter representing road construction difficulty. Some parameters are also unknown until a solution is considered and evaluated by all stakeholders (including forest companies, hunting associations, cabin owners, and wildlife protection associations). Along this line, any solution to be considered must undergo a process of validation by forest stakeholders, resulting in adjustments and compromises. However, an interesting characteristic of this problem is that a complete solution can be easily shown in a single computer screen and contextualized within a GIS for analysis. Furthermore, experienced forest engineers can quickly evaluate the structure and quality of the solution.

4 Interactive solution approach

In practice, forest road location is a manual process carried out within a GIS that contains a large amount of information about forest and soil conditions, as well as areas to harvest. Some GIS, such as *PlaniRoute*, also provide specific functions that can estimate harvest cost, road construction cost, as well as transportation cost. However, these commercial GIS do not yet provide functions that automatically design road networks. Therefore, in order to develop an APS system that supports forest road location planning by taking advantage of users' experience and knowledge, we implicitly followed a series of general development principles.

4.1 General design principles

APS systems contain an optimization software module that aims to provide optimal or near optimal solution to a specific planning decision problem. Human-computer interactions are generally seen as part of a more or less formal decision process that is disconnected from the optimization function, through trial-and-error. Therefore, users perform scenario-based and what-if analysis by managing different sets of decision parameters and by selecting solutions. However, human-computer interaction can be implemented at a more fundamental level by embedding the human thought process and the computer heuristic search into a non-linear decision process. In order to do so and develop an advanced road planning system capable of taking advantage of users' experience, we implicitly followed a series of general development principles that address the decision support process, the user interface, as well as the planning heuristics. These general principles are summarized hereafter.

4.1.1 Decision support process

First, the decision support process must take advantage of computer heuristics in order to process the large amount of information that defines the planning problem. Heuristics can be used to compute or estimate a cost, improve a solution, optimize a problem, or provide any automated function required by the decision process, as it is discussed later. Second, the decision support process must take advantage of users' ability to quickly analyze a tentative solution, evaluate its feasibility and guide the optimization process toward what he or she believes is a promising region of the solution space. Third, the decision support process can lead to either a manual decision process, a computerized decision process (i.e., when the solution is entirely provided by the system), or cooperative (i.e., human-guided planning).

Fourth, planning decisions (e.g. solutions) are complex multi-component objects that can be, as a whole, communicated, analyzed, and evaluated. Their individual components, or any sets of components, can also be selected, discarded, adjusted, or added. The definition of such a structure is problem specific. It is an artifact, which specific architecture is design to serve the purpose of allowing the user to transform a solution as he or she sees fit.

The practical impact of these four general principles is that any tentative decisions can be the result of a human-computer cooperative process. These principles also model the decision process as an iterative process, in which a tentative decision is an object that can be:

- observed and evaluated according to many different criteria or perspectives, using human qualitative judgment or quantitative computerized process (e.g., simulation);
- built, negotiated and modified by computer heuristics, users or more generally software agents.

4.1.2 User interface

First, the human-computer interface must provide users with information, which allows them to contextualize and analyze any solution (i.e., with respect to decision parameters and external data). This general principle aims to support the thought process of the user in order for him or her to quickly identify the parts of a tentative decision that may lead to unsatisfactory decision. Along this line, the interface must provide users with a solution's attributes in order to facilitate its evaluation, such as by directly calculating attributes, or by simulating different scenario of operations based on this solution. This principle aims again to support the thought process of the user. However, it more specifically considers the use of external computerized functions (e.g., simulation) to support the analysis of a tentative decision. Finally, the interface must allow users to adjust a tentative solution by adding constraints, by selecting part of a solution to be modified or by reconstructing an altered or partial solution. This general principle is fundamental as it explicitly considers the interaction mechanisms users may exploit in order to provide the computer system with a feedback to converge towards a practical and efficient solution. Once again, it is problem specific, as interaction mechanisms must be specifically designed for each type of solution structure. For instance, the solution structure of a scheduling problem must take into account the temporal dimension and resources allocation. However, in the context of forest road construction, the geographic location is the most important dimension for the user.

Similarly, these user interface principles model solutions as an object that can be observed, analyzed, transformed, simulated or stored in memory. Thanks to these functions, the user interface is at the center of the decision process, allowing users, and even stakeholders, to manipulate and evaluate any tentative solutions, while heuristics can automatically manipulate them to improve them.

4.1.3 Heuristics

In order to create a reactive and collaborative human-computer environment, heuristic procedures must be able to quickly perform a set of general and specific functions to contribute to the decision process. These general functions include:

- providing complete solutions;
- reconstructing/adjusting partial solutions by taking new constraints into account or by focusing on specific parts of the tentative solution;
- computing the attributes of tentative solutions.

These general functions aim to meet the need to have a good solution to initiate or restart the optimization process, and take into account new constraints and users' input to adjust the current solution. The next section presents the implementations of all these principles into a decision support system for forest road location.

4.2 Application for the forest road location problem

The proposed planning application can be used in three different modes of decision support, from purely computerized to purely manual. Along the same line, specific information are computed and visualized in order to help the user understand the characteristics and the performance of tentative decisions. This section first introduces the developed decision support processes for the forest road construction problem. Next, we present the human-computer interface. Finally, we present the interactive heuristics.

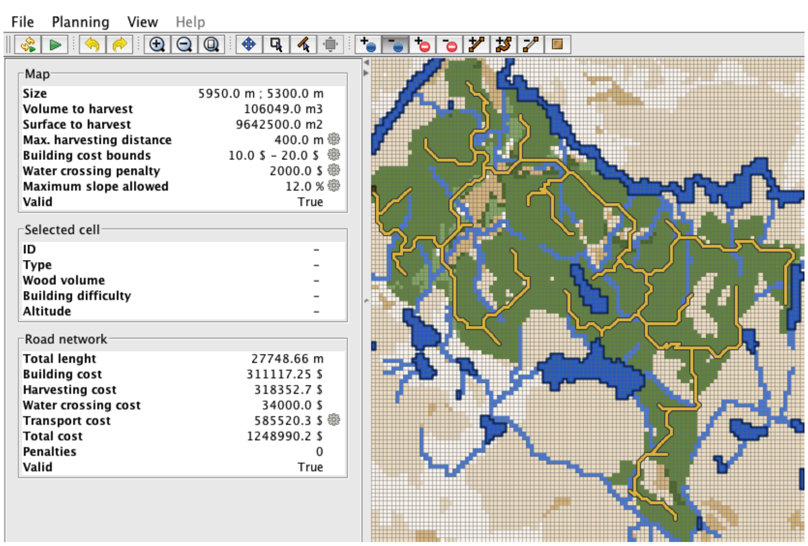
4.2.1 Decision support processes

The first decision support mode is purely automated. Here, the advanced planning system provides a first complete solution that meets all constraints initially formulated in the problem. Once a map of the region to harvest is loaded in the system, the user makes sure that the existing road network is modeled on that map. Then, the user initiates a first quick planning heuristic (H1). Figure 1(a) presents a first sub-optimal solution that satisfies the basic constraints of the problem, but has no regard for transportation cost. For instance, all harvest cells (i.e., demand points) are at a skidding distance smaller than the minimum required. Therefore, at this stage, the tentative solution could be considered for implementation although it is clearly not optimal. Unfortunately, because the proposed heuristic neglects transportation cost, the current tentative decision may require some adjustments in order to reduce transportation inefficiencies of the solution. At this point, the user enters the interactive mode of decision support.

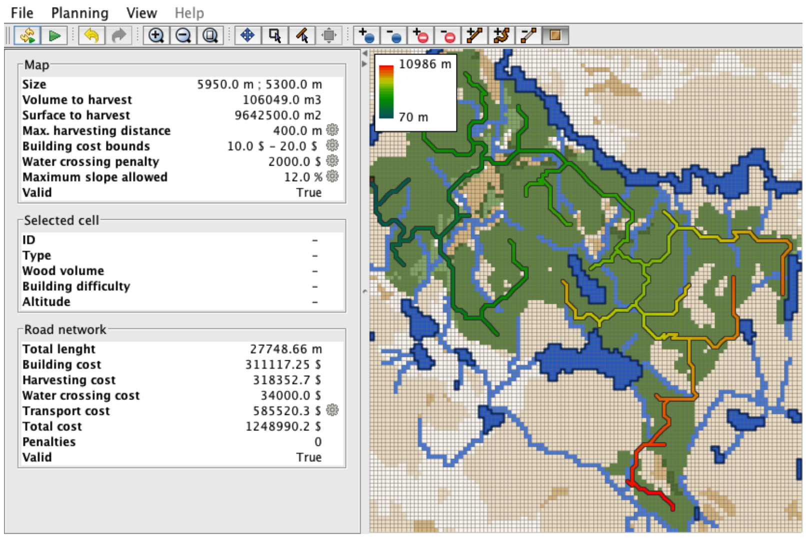
In the interactive decision support mode, the user can visualize some information in order to better understand the characteristics of the solution and analyze it in a larger context. This information is discussed in the next section. For instance, in the example considered in Figure 1(b), the distance of each road edges to the existing road network is indicated by a color, which allows the user to very quickly identify potential transportation inefficiencies (i.e., the red segments of the road network that are far from the root). Once one or many issues are identified, the user has three basic interactions mechanisms to adjust the solution. The first mechanism allows the user to **add mandatory points of passage**. For instance, as shown in Figure 2(a), the user added a mandatory point of passage (i.e., blue cell) between two road segments. In this example, near this cell, harvested logs could be skidded and hauled over very different distances to reach the root (i.e., one segment is much further from the root). Therefore, it might be interesting to link these two segments by adding a mandatory point of passage and reduce skidding distance.

Figure 1 Generation and visualization of an initial solution

(a) Visualisation of the initial solution. The road network generated by the heuristic procedure is on yellow.



(b) Visualization of the initial solution with an indicator of distance to the root. A color gradient is used to visualize the distance.



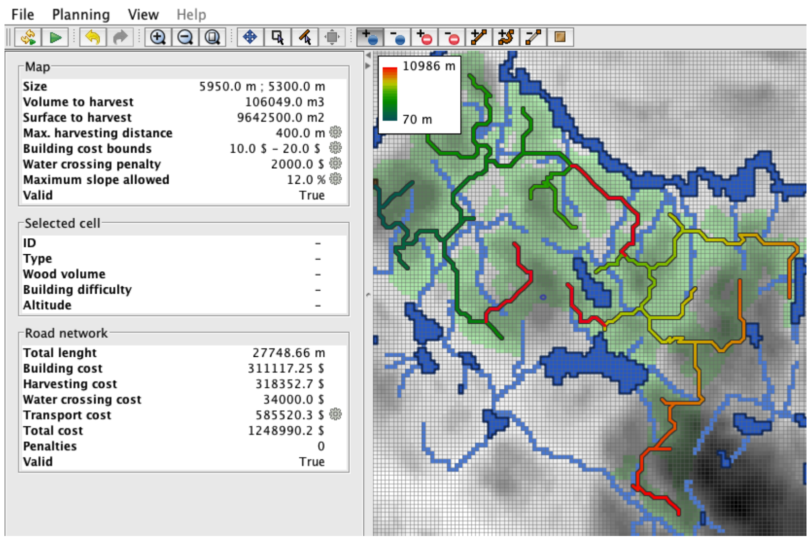
The second mechanism allows the user to **select parts of the road for a partial re-optimization** of the current solution. In order to do that, we defined a priori a structure to manipulate solutions. Table 1 presents this structure. The higher level represents the entire road network, which is a complete tentative solution. At this level, it can only be completely discarded or selected for further adjustments through the modification of one or several parts (i.e., lower level sub-structure). If the solution is not discarded, then the user can selects road segments (i.e., sub-set of road edges between two crossroads, or between a root of the network and a crossroad) to eliminate from the solution, and re-optimize locally. This mechanism allows the user to quickly reconstruct only the selected road segments, using a second heuristic (H2). Again, as shown in Figure 2(a), three road segments are selected (i.e., the dark red segments) in order to allow the computer to reconstruct the solution around the mandatory point of passage, and avoid having a loop.

Table 1 Levels of interaction with a solution

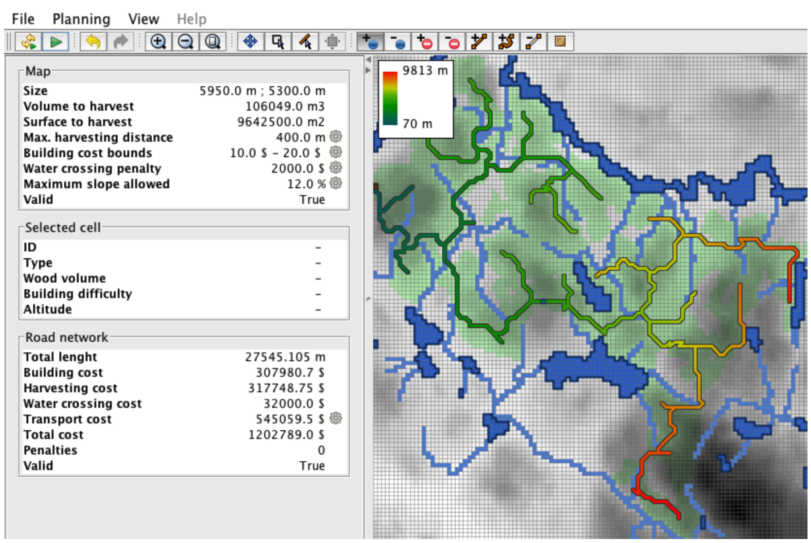
Level	Definition	Allowed actions
Road network	The entire solution, which comprises one or many roots (i.e., point of connection between the solution and an existing road) to each of which, several road segments are connected together to form a tree.	Discard; (Re-)optimize;
Road segment	Sub-set of road edges between two crossroads (i.e., when a road splits), or between a root of the network and a crossroad.	Create (manually by selecting two points on the map); Select (for elimination and local re-optimization);
Road edge	Elementary part of a road as defined by the granularity of the optimization model.	Forbid (for re-optimization);
Unselected forest cell	Elementary cell of forest that is not part of the solution (the cell is not selected to have a road segment built on)	Forbid (for re-optimization); Force road construction (for re-optimization);

Figure 2 Partial reconstruction of a solution with new constraints

(a) Selection of the road sections to re-plan and addition of a mandatory point of passage. Road sections to re-plan are indicated in red, and mandatory point of passage is the blue cell near the centre of the map.



(b) New solution obtained after a re-planning step.



The third interaction mechanism allows the user **to add forbidden points of passage**, as shown in Figure 3(b). This mechanism allows to user to add constraints that were not initially considered, due to lack of information. For instance, habitat protection requires forest road to avoid natural habitat for specific type of wildlife. This information is not necessarily stored in the GIS. It is therefore the responsibility of the forest engineer to acquire this knowledge through, for instance, site visit. Once this information is input directly on the map, the user can initiate the reconstruction of the solution with the second heuristic (H2) described thereafter.

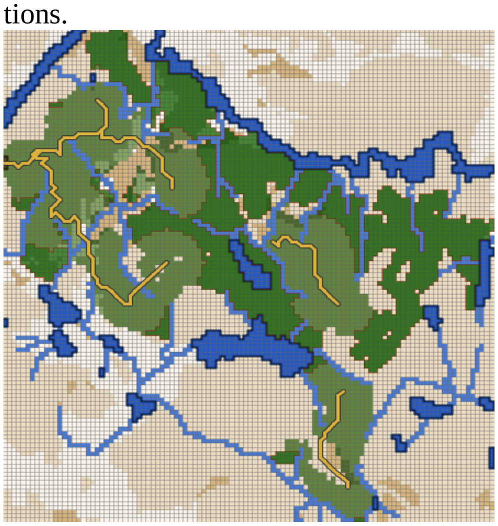
In the illustrating example shown in Figure 2(b), based on a real sector to harvest, a single step of interaction allow the system to identify a better solution (i.e., the harvest and skidding cost decrease by 3.7%, while estimated transportation cost decreased by 7%, by removing more than 200m of road). In this process, the user intervention is a means to restart the heuristic search procedure within a larger neighborhood (i.e., diversification), and by focusing the search for a solution towards a specific part of the current solution (i.e., intensification). This process can be repeated until the user considers that the solution satisfies the requirements in term of cost and feasibility. Furthermore, an “undo” function also allows the user to go back to any of the previous solutions, and explore manually other branches of the overall interactive optimization process. Also, solutions can be saved for comparison purpose.

In yet another interactive decision support mode, users can semi-manually build a road network by iteratively adding road segments defined by an origin and an end. This process can be entirely manual when the user just adds straight-line segments between these two points. However, this process requires the user to manually evaluate the slope of these road segments, because the system does not check this constraint in this edition mode. In order to avoid such a problem, the user can add optimized road segments, using a third heuristic that aims at minimizing construction cost between the origin and the end specified by the user. This third heuristic does not consider road edges which slope is higher than a user-defined threshold. For instance, as shown in Figure 3(a), optimized road segments can be added anywhere on the map, resulting in the construction of a partial road network, which segments may be disconnected. The user can either continue this process until all harvest points (i.e., demand points) are covered. In order to support this manual mode of constructing the road network, the interactive system can automatically show in a different color (i.e., darker colored cells in Figure 3) all harvest cells that are further than the user-defined minimum skidding distance.

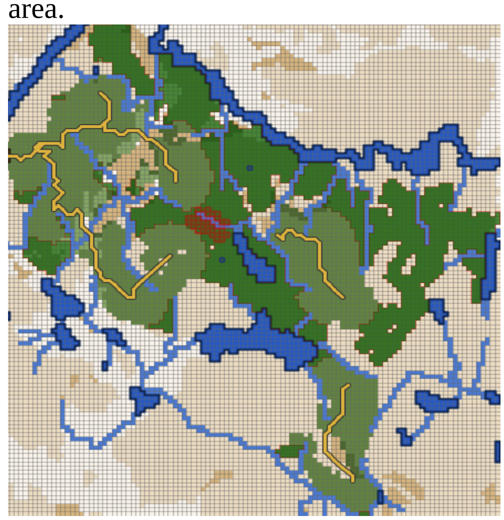
Instead of manually constructing the entire solution, the user can also initiate a reconstruction procedure of the partial road network to obtain a complete solution. At this point, the user can, like in the interactive decision support mode, add mandatory and forbidden points of passage as shown in Figure 3(b). This procedure uses the road segments in place in order to identify a complete network that covers all uncovered harvest points. Figure 3(c) shows a solution produced using such mode of interaction.

Figure 3 Reconstruction of a partial solution with mandatory points of passage

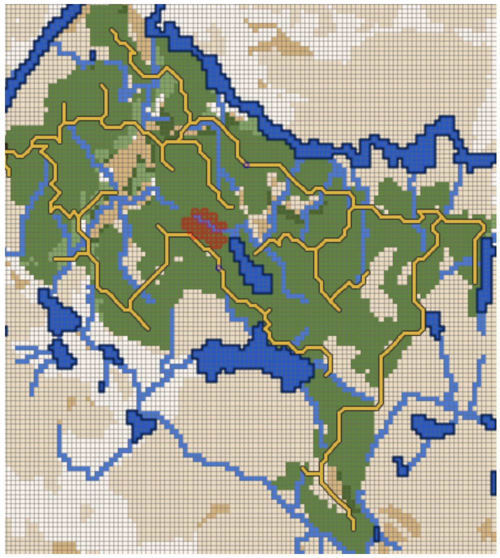
(a) Partial solution with manually added road sections.



(b) Addition of mandatory points and forbidden area.



(c) Solution obtained after the reconstruction step.

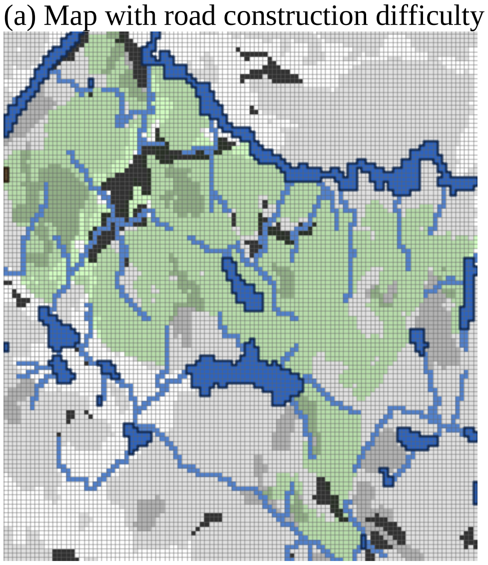


4.2.2 *Human-computer interface*

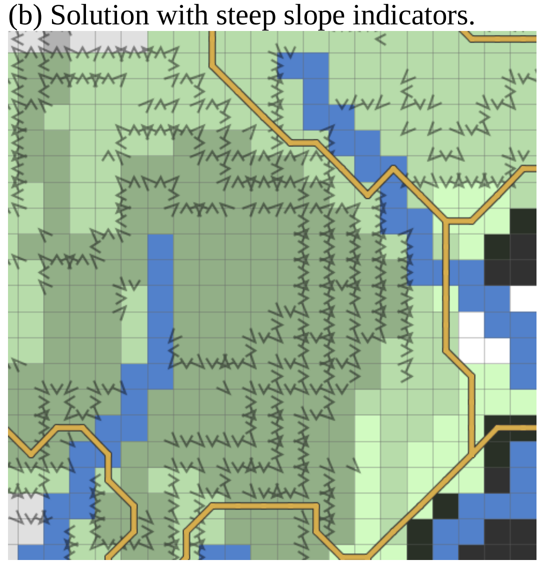
As shown in the previous section, the proposed system allows users, on the one hand, to visualize specific information according their need to understand the characteristics of a tentative solution and the constraints of the problem, and, on the other hand, to input specific information to guide the heuristic procedures to search for a solution. In order to support users with their need to understand the characteristics of a tentative solution, the system can display several types of information directly on the map, as in a GIS. Figure 4{a, b and c} shows three more types of information, including (a) the road construction difficulty index, which indicates road construction cost of these segment (dark segments have a high cost due to inappropriate soil conditions), (b) the direction of steep slope with road construction difficulty index (zoomed in), and (c) the total flow intensity of a tentative solution (i.e., calculated attribute of a solution). The user can visualize this information with the menu bar.

Figure 4 Modes of visualization with different information

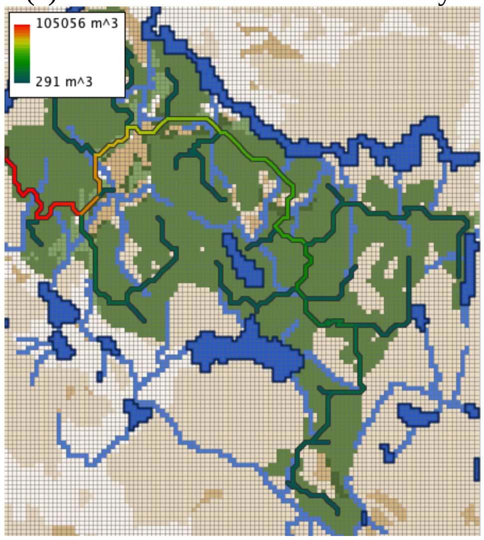
(a) Map with road construction difficulty index.



(b) Solution with steep slope indicators.



(c) Solution with total flow intensity.



Concerning feedback information to guide the heuristic search (Figure 3(b)), users can add or remove mandatory or forbidden points of passage in a point-and-click mode. Users can also add straight-lines or optimized road segments, and select road segments to be reconstructed from a tentative solution.

4.2.3 Interactive heuristic

The overall interactive decision support process consists of several steps as described here after. First, an initial solution is constructed using a first fast heuristic referred to as H1. This heuristic is a two-step greedy procedure with a local descent introduced in Meignan et al. (2012). It is based on the minimum path heuristic for the Steiner tree problem (Hwang et al. 1992).

The first step (H1, Algorithm 4) consists in locating a set of covering points, located so that all harvest points can either be skidded to an existing road (i.e., root vertices) or to one of these covering points considering the maximum skidding distance. The set of covering points is determined by iteratively adding the vertex in the graph that covers the maximum number of demands cells not yet covered. The second step (H2, Algorithm 5) uses the minimum path heuristic to connect the covering points obtained during the first step. The basic principle of the minimum path heuristic is to construct a forest (i.e., road networks with no cycle) by iteratively adding new branches that connect the current trees or root vertices to a remaining covering point. At each iteration the algorithm first chooses the smallest-cost branch, using Dijkstra's algorithm between forest vertices and remaining covering points, and adds it to the forest.

Algorithm 4 H1 – Greedy procedure for the selection of covering vertices

coveringVertices ← ø
toCover ← demands – VERTICESCOVERED(rootVertices)
while toCover ≠ ø **do**
  vertex ← MAXIMUMCOVERAGE(toCover)
  coveringVertices ← coveringVertices U {vertex}
  toCover ← toCover – VERTICESCOVERED(vertex)
end
return coveringVertices

Algorithm 5 H2 – Spanning covering vertices with minimum path heuristic

```
terminals ← coveringVertices
forest ← ø
forestVertices ← rootVertices
while terminals ≠ ø do
   path ← MINIMUMPATH(forestVertices, terminals)
   forest ← forest U path
   forestVertices ← forestVertices U VERTICES(path)
   terminals ← terminals – VERTICES(path)
end
return forest
```

This two-step greedy procedure has two limitations. First, it only takes into account construction costs c_{ij}, and not skidding costs d_{ij}. Skidding costs may be integrated in the heuristic function to choose the covering vertices in the first step. But, as we need to determine the minimum number of covering points with the minimum skidding costs, the ratio between the number of covered harvest cells and the skidding costs is difficult to tackle as a heuristic function. In addition, it leads to inferior results than the adopted heuristic function only based on the number of covered demands. However, this first limitation is resolved by applying a local descent procedure that considers both road construction and skidding costs. The second limitation is linked to the deterministic aspect of the procedure. Therefore, this greedy procedure provides a unique solution to each problem instance. This problem was addressed with the development of a Greedy Randomized Adaptive Search Procedure (GRASP), in which the first step was slightly modified in order to involve randomness to produce different solutions on the same problem instance. However, this application is outside the scope of this paper (Meignan et al. 2012).

The local descent procedure (H3, Algorithm 6) developed to resolve the first limitation is a variable neighborhood descent (Hansen and Mladenovic 2003). It exploits three neighborhood structures, in which a new solution is obtained respectively by adding or removing a vertex or swapping two vertices in the initial solution. The neighborhood solutions considered are restricted to the solutions that satisfy constraints, and thus, removal or swapping operations are performed only if all harvest points are covered by the resulting road network. In other words, this local descent procedure exploits a set $N_k,\ k = \{1,2,3\}$, neighborhood structures. It first consists in finding a local optimum using the k^{th} neighborhood structure, starting with $k=1$. Then, if the local descent using the k^{th} neighborhood structure does not lead to an improvement of the solution or, if $k=1$, the local descent continues with neighborhood $k+1$. Otherwise, k is set to the value 1 for the next local descent. 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 performances, the selection of a neighborhood solution uses a first improvement policy (i.e. the first neighbor which cost value is smaller than the current solution is selected. Here, the evaluation of solutions considers both road construction and skidding costs.

Algorithm 6 H3 – Variable neighborhood descent procedure

```
s ← InitialSolution()
k ← 1
while k ≤ k_max do
   move ← false
   while HasNeighbor(s, N_k) and !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
```

Once an initial solution is found using these three heuristics, the user analyze the solution directly within the computer interface and add constraints and select road segments to be reconstructed in order to guide the next iteration of the optimization using a fourth heuristic H4. Depending on the user's personal decision support preference, an iteration of the computerized optimization process corresponds either to an optimization limited to a part of the solution, or to the construction of an entire solution considering the new constraints. Therefore, user interactions can affect both problem data and optimization process. When the user adds new constraints, they either take the form of road cost penalties defined on vertices (i.e., to forbid points of passage, although they are not explicitly eliminated), or mandatory vertices (i.e., to force mandatory points of passage). Then, the user can limit the optimization procedure to parts of the current solution by defining a set of edges (i.e., through the selection of road segments) to modify in the current solution, while the rest of the solution is not modified in the next iteration. This restriction allows the user to focus on specific parts of the problem. It also improves the computation time of the optimization process by reducing the search space.

In order to develop heuristic procedures to take into account user feedback, we adapted the proposed two-step greedy construction procedure (H1 and H2) and the variable neighborhood descent (H3). These adaptations were made in order to take into account mandatory vertices constraints and optimization restrictions. As far as the adding of forbidden points of passage, the introduction of cost penalties does not require particular adaptation of the algorithms. At this point of the decision support process, users can call two different procedures.

On the one hand, if some mandatory vertices have been added before a new solution is required, a slightly modified version of the two-step greedy construction procedure is applied. In the first step, the mandatory vertices are initially added to the set of covering points in order to obtain, after the second step, a set of trees that pass through these vertices. This adapted greedy construction procedure is followed by a variable neighborhood descent with restrictions on neighborhoods. Indeed, the moves are not allowed to remove a mandatory vertex from the solution.

On the other hand, if the user restricts the optimization on a part of the solution, the selected road edges are removed from the solution. Then, a reconstruction procedure completes and connects the parts of the partial solution. This procedure is again an adaptation of the two-step greedy construction procedure. In the first step, demand points covered by the partial solution are not considered to determine the set of covering points. Then, the minimum path heuristic of the second step is adapted to connect both covering points and the partial solution. Finally, the variable neighborhood descent is then applied with restrictions on neighborhoods, so that mandatory vertices and vertices from the initial solution are not considered in the neighborhoods.

5 Computational results

A set of experiments on real problem instances was conducted in order to analyze the efficiency of the proposed approach. The efficiency of GRASP has been evaluated from a purely computational perspective in a previous study (Meignan et al. 2012). The focus of the following experiments is the interaction between the optimization process and the user. The goal of these experiments is to analyze in real situations how user can contribute to the optimization process.

To do so, three graphs of more than 8 000 vertices were considered. They correspond to maps of approximately 3 000 hectares. The inputs maps were produced using a commercial GIS, and real cost parameters were used to generate the graphs. For each of these problem instances, solutions are compared to road networks specifically planned manually by a forest engineer using *PlaniRoute*. In order to compare the costs of the manual and interactive solutions, all road networks were evaluated using the same objective function (1). For the interactive optimization approach, the forest engineer used the implemented software to construct realistic solutions.

After a short period of learning of the different functions and interaction mechanisms of the application, it took between 30 minutes and 1 hour to construct a satisfying solution, for each of the problem instance, which compares to several hours in a purely manual setting (without the support of the developed decision support system). These solutions are compared with manual solutions in Table 2. For a similar level of trust and confidence in the solution, the interactive approach yielded a significant average improvement of more than 9% with a significant improvement in total time to produce a road network.

Table 2 Experimental results comparing manual and interactive approaches

Problem			Manual	Interactive heuristic	
	Map size	Nb. Vertices	Solution cost	Solution cost	*Gap Manual vs. Int.*
#1	2 900 ha	8 084	$408 882	$372 474	*8.90%*
#2	2 881 ha	11 526	$697 329	$613 085	*12.08%*
#3	3 153 ha	12 614	$700 381	$645 041	*7.90%*
				Average	*9.63%*

6 Conclusion

In this paper, we proposed an interactive optimization method for the forest road location problem. In this interactive method, the user contributes to the optimization process by iteratively adding constraints on the problem and identifying the parts of the solution to involve in the next re-optimization. These interactions affect both problem data and optimization process.

A set of experiments was conducted to analyze the efficiency of the proposed approaches. It was performed on real problem instances and aimed at evaluating, for a similar level of trust and confidence in the solution, the interactive approach against manually designed solutions. The interactive heuristic obtained an average gap of 9% in comparison to the solutions manually constructed by a forest engineer, with a significant improvement in total time to produce a road network.

The advantages of the interactive approach are, first, to propose globally good solutions, which respect modeled constraints. Then, the interaction allows partial modifications of the solutions and re-optimizations that focus on specific parts of the networks to produce more realistic solutions. Finally, solutions with lower costs than manual ones are obtained in a reduced amount of time. Such interactive planning systems are not an alternative to highly efficient optimization techniques. They are rather a decision support approach for particular type of problems, which rich nature and intrinsic complexity require users to be involved at all stages of solution building to improve solution/decision acceptance.

6.1 Future work

The development of interactive decision support systems in the domain of operations planning is a promising research domain. Therefore, it would be interesting to study human-computer interactions from a cognitive and ergonomic perspective, in order to better understand the types of human-machine interfaces that foster cooperative work environment and efficient human-heuristic integration. For instance, one approach could be the creation of new solutions from the hybridization of others. In other words, users could save several tentative solutions, and select parts of any of them and reassemble them together in order to create a new hybrid partial solution that could be reconstructed.

Another second interesting aspect to study concerns the integration of human input within the heuristic. In other words, it would be interesting to better understand, identify and develop natural interaction mechanisms and human tacit expertise elements that could be integrated within a heuristic process of optimization.

A third aspect of interest in the context of interactive decision support systems concerns the development of interface agents that could learn the preference of its user in order to proactively propose adjustments to be made, or even entire plans. The introduction of an interface agent, that can learn over time its user’s preferences, could bring another modeling paradigm to improve the solving of complex problem. Similarly, such an interface agent could even be used in a training mode in order to support the training of a student or a beginner user by identifying rooky mistake and by proposing alternative planning solutions.

Acknowledgments

The authors would like to thank Mathieu Blouin and Jean Favreau as well as the contribution of FQRNT (Fonds Québécois de la Recherche sur la Nature et les Technologies), Mitacs, and NSERC (Natural Sciences and Engineering Research Council of Canada) for having supported the project under the discovery grant program.

References

Anderson, A.E., and Nelson, J.: Projecting vector-based road networks with a shortest path algorithm, Canadian journal of forest research 34(7), 1444–1457 (2004)

Babbar-Sebens, M., Minsker, B.: A Case-Based Micro Interactive Genetic Algorithm (CBMIGA) for interactive learning and search: Methodology and application to groundwater monitoring design, Environmental Modelling & Software 25, 1176–1187 (2010)

Banzhaf, W.: Interactive evolution, In: Handbook of Evolutionary Computation, Oxford University Press, 1–6 (1997)

Biles, J., Anderson, P., Loggi, L.: Neural network fitness functions for a musical IGA, In: Proceedings of the International ICSC Symposium on Intelligent Industrial Automation (1996)

Branke, J., Deb, K., Miettinen, K., Slowinski, R.: Multiobjective Optimization, Interactive and Evolutionary Approaches, Lecture Notes in Computer Science 5252, 470p. (2008)

Branke, J., Kaußler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization, Advances in Engineering Software 32, 499–507 (2001)

Chung, W., Stückelberger, J., Aruga, K., Cundy, T.W.: Forest road network design using a trade-off analysis between skidding and road construction costs, Canadian journal of forest research 38(3), 439–448 (2008)

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

D'Amours S., Rönnqvist, M., Weintraub.,A.: Using Operational Research for supply chain planning in the forest product industry, Information Systems and Operational Research (INFOR) 46(4), 265–281 (2008)

Dean, D. J.: Finding optimal routes for networks of harvest site access roads using GIS-based techniques, Canadian Journal of Forest Research 27, 11–22 (1997)

Deb, K. and Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms, In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 635–642 (2006)

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

FPInnovations: Planiroute [online], available at http://feric.ca [accessed June 2013] (2013)

Hansen, P. and Mladenovic, N.: Variable Neighborhood Search, In: Handbook of Metaheuristics, Kluwer Academic, pp. 145–184 (2003)

Hao, T., Miller-Hooks, E.: Interactive heuristic for practical vehicle routing problem with solution shape constraints, Transportation Research Record 1964, 9–18 (2006)

Hwang, F.K., Richards, D.S., Winter, P.: The Steiner tree problem, In: Annals of discrete mathematics 53 (1992)

Ibaraki, T., Nonobe, K., Yagiura, M. (Eds): Metaheuristics: Progress as Real Problem Solvers, Springer, 414 p. (2005)

Jaszkiewicz, A. & Słowiński, R.: The 'Light Beam Search' approach – an overview of methodology applications, European Journal of Operational Research 113(2), 300–314 (1999)

Jones, C. V.: Visualization and optimization, Kluwer, 434 p. (1997)

Junior H.V., Lins, M.P.E.: A win-win approach to multiple objective linear programming problems, Journal of the Operational Research Society 60 (5), 728–733 (2009)

Karlsson, J., Rönnqvist, M., and Bergström, J. 2004. An optimization model for annual harvest planning. Can. J. For. Res. 34(8): 1747–1754.

Katagiri, H., Sakawa, M., Kato, K., Nishizaki, I.: Interactive multiobjective fuzzy random linear programming: Maximization of possibility and probability, European Journal of Operational Research 188(2), 530–539 (2008)

Klau, G.W., Lesh, N., Marks, J., and Mitzenmacher M., (2010). Human-guided search. Journal of Heuristics, 16(3), 289-310.

Krolak, P., Felts, W., Marble, G.: A man-machine approach toward solving the traveling salesman problem, In: DAC '70 Proceedings of the 7th Design Automation Workshop, pp. 250-264 (1970)

Lee, J.-Y., Cho, S.-B.: Sparse fitness evaluation for reducing user burden in interactive genetic algorithm, In: Fuzzy Systems Conference Proceedings, IEEE International, Vol. 2, pp. 998-1003 (1999)

Legües, A.D., Ferland, J.A., Ribeiro, C.C., Vera, J.R., Weintraub, A.: A tabu search approach for solving a difficult forest harvesting machine location problem, European Journal of Operational Research 179(3), 788–805 (2007)

Llorà, X., Sastry, K., Goldberg, D.E., Gupta, A., Lakshmi, L.: Combating user fatigue in iGAs: partial ordering, support vector machines, and synthetic fitness, In: GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 1363–1370 (2005)

Meignan, D., Frayret, J.-M., Pesant, G., Blouin, M.: An Automated Heuristic Approach to Forest Road Location, Canadian Journal of Forest Research 42(12), 2130–2141 (2012)

Meignan, D., Knust, S., Frayret, J.-M., Pesant, G., Gaud, N.: A Review and Taxonomy of Interactive Optimization Methods, *submitted* (2013)

Mesa, J.A., Boffey, T.B.: A review of extensive facility location in networks, European Journal of Operational Research 95, 592–603 (1996)

Miettinen, K.: Survey of methods to visualize alternatives in multiple criteria decision-making problem, OR Spectrum (2012)

Miettinen, K., Eskelinen, P., Ruiz, F., Luque, M.: NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point, European Journal of Operational Research 206, 426–434 (2010)

Miettinen, K., Mäkelä, M. M.: Interactive multiobjective optimization system WWW-NIMBUS on the Internet, Computers & Operations Research 27(7-8), 709–723 (2000)

Miettinen, K., Ruiz, F., Wierzbicki, A. P.: Introduction to Multiobjective Optimization: Interactive Approaches, In: Multiobjective Optimization. Springer, pp. 27–58 (2008)

Stückelberger, J., Heinimann, H.R., Chung, W.: Improved road network design models with the consideration of various link patterns and road design elements, Canadian Journal of Forest Research 37(11), 2281–2298 (2007)

Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation, In: Proceedings of the IEEE 89(9), pp. 1275–1296 (2001)

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

Voudouris, C., Tsang, E. P. K.: Guided local search, In: Handbook of Metaheuristics, pp. 185–218 (2003)

Weintraub, A., Murray, A.T.: Review of combinatorial problems induced by spatial forest harvesting planning, Discrete Applied Mathematics 154(5), 867–879 (2006)