# ROAD-NETWORK LOCATION HEURISTICS FOR THE TACTICAL HARVEST-SCHEDULING MODEL

by

Junbeom Shin

Faculty of Natural Resource Management
Lakehead University
Thunder Bay, Ontario
Canada

October, 2022

# ABSTRACT

Shin, JB. 2022. Finding fast-Steiner heuristics suitable for an integrated tactical level harvest-scheduling and road-planning model.

Keywords: integrated harvest-scheduling, road networks, Steiner tree heuristics.

In tactical planning in hierarchical forest management, cut-blocks are selected for maximizing revenue and road networks are allocated at minimal cost in order to maximize profit. The selected cut-block set and requisite road-network, connecting the cut-blocks, therefore have an interdependent relationship. The location of these two elements in tactical planning must therefore be considered simultaneously in a tactical harvest-scheduling model. This integration presents a major computational challenge, especially with regard to the execution time required to find an optimal solution to the tactical harvest-scheduling model.

The objective of this thesis is to explore the influence of different road location heuristics, used within the tactical harvest scheduling model, upon the model's execution time and solution quality. We nested the three different types of road-location heuristics within the harvest-scheduling model in order to evaluate their effectiveness by three criteria: execution time, road construction cost and objective function value. In addition, after the tactical model was run, we executed and evaluated the usefulness of a road network repair algorithm, designed to improve further the solution of the road-network location generated by the tactical harvest-scheduling model. The thre heuristics were evaluated on a real-world dataset, representing a section of the Kenogami forest in Ontario, Canada. Our result show: i) that the Shortest Path Origin Heuristic (SPOH) achieved the fastest execution time and lowest construction cost when integrated within the tactical harvest-scheduling model; and ii) that the road network repair algorithm successfully lowered the road network costs and thereby increased the objective function value of all solutions generated using the tactical planning model. These results are significant for two reasons: first, they show that the choice of the road network heuristic used within a tactical planning model can have a major influence on the model's solution quality; and second, that the use of a road repair algorithm, on the solution generated using a tactical model, is of major economic value in forest management planning.

# CONTENTS

# TABLES

# FIGURES

# 1. Introduction

## 1.1 Problem defined

The purpose at the tactical level plan in forest management is to schedule harvest operations to specific areas and on a finer time scale than in strategic plans (Martell *et al.* 1998). This requires the building of new roads and the delineation of harvest-blocks (Beaudoin, Frayret and LeBel 2008). At the tactical level in forest management planning, the road construction and transportation costs are the main costs which affect the profit of the tactical plan (Bjørndal *et al*. 2012). According to Bjørndal *et al.* (2012), the road construction cost is more than 40% of the total forest operational cost. To achieve the objective function of maximizing profit (revenue *minus* road construction), road construction costs must therefore be minimized.

The relationship between the optimal spatial allocation of cut-blocks and the optimal spatial allocation of a road-network is one of interdependence. Therefore, at the tactical planning level, selecting the cut-block set and the road network's location must be done simultaneously in a harvest-scheduling model. Earlier research has shown the superiority of an *integrated model* (in which decisions on cut block location and road network locations are made simultaneously) over a sequential approach to these decisions (Weintraub and Navon 1976; Kirby *et al*. 1980, 1986).

The major problem confronting researchers on the integrated tactical harvest-scheduling model is that both problems within this model (i.e., the cut block location problem and road network location problem) are NP-hard (Murray and Church 1995). An NP-hard hard problem entails that, as the number of binary variables in an optimization model increases, the required computing time to find the optimal solution increases exponentially (Reeves 1993). The practical implication of this is that, in forest management planning, where datasets typically contain a large number of binary decision variables, the mathematically optimal solutions cannot be found using exact solution methods (Martell et al. 1998). Hence, heuristic solution methods have been used to solve the tactical harvest-scheduling problem on large problems (Murray and Church 1995). Heuristic algorithms are used to solve NP-hard optimization models such that near-optimal solutions can be found within reasonable computing times (Reeves 1993).

The specific problem addressed in this thesis is the evaluation of different heuristic algorithms that can be used to solve the road network allocation problem for the integrated tactical harvest-scheduling problem.

### 1.2 Objectives

The objective of this thesis is to develop and evaluate different heuristic algorithms for solving the road network allocation problem for the integrated tactical harvest-scheduling problem. In particular, our objective is to address two problems; i) finding an efficient and fast heuristic algorithm that, at minimal construction cost, connects the current set of cut-blocks, that have been selected by a standard simulated annealing algorithm, within the tactical harvest-scheduling model; and ii) developing a heuristic algorithm that improves the road network after the tactical model has finished executing.

### 1.3 Significance

The work in this thesis is significant for three reasons. First, this work is innovative, as will be demonstrated through the literature review. Second, this work has economic importance. Most instances of tactical planning problems in the real world of forest management planning require metaheuristic algorithms to be solved. The work in this thesis therefore shows how, through efficient heuristic algorithms, to reduce the major operational cost of road-construction in forest tactical planning. Third. this work is practical. At the tactical planning level of forest planning, many commercial software packages are available. The road network repair algorithm developed in this thesis, can be used on the solutions generated by these software packages. In other words, the repair algorithm can easily be used after and independently of the software packages used to generate a tactical planning solution. This makes it easy to implement in practice.

# 2. Literature Review

The literature review section is divided into two parts. The first part of the literature review concerns road location and road network location models. These papers are reviewed because a road location model is used in this thesis to prepare the dataset with a network of candidate roads to be used by the integrated model. The second part of the literature review concerns the integrated harvest-scheduling model. This topic is reviewed because it is the focus of the research and innovation in this thesis.

## 2.1 Road Location and Road Network Location Models

A forest road location model connects two points in the forest at minimal cost. A forest road network location model connect multiple forest roads. Forest roads should be feasible for the transportation of harvested wood. According to Akay *et al* (2013), designing forest roads should consider steep terrain, which affects the slope, geology and machine limitations. Machine limitations are also affected by maximum road grade and minimum turning radius. Forest roads should therefore follow both horizontal and vertical constraints.

Since manually drawing forest roads can be extremely time-consuming and labour-intensive, road location models, to be executed quickly on computers, have been developed. The first experimental attempt at computer-aided forest road location was in the late 1960s (Akay et al. 2013). In the 1970's, a much-advanced road location computer program, that satisfied horizontal and vertical constraints, and addressed economic objectives, was applied to a 100-mile highway corridor in western New York and Pennsylvania (Turner 1978). In the 2000's, the development of remote sensing technology allowed high-resolution digital terrain models (DTM) obtained from airborne or LiDAR, to be used in designing forest road networks (Akay and Sessions 2005). Since the development of DTM's, multiple modeling methods have been developed in the literature on the road location model. These models, which use DTM's, will now be reviewed. The review of the road location model is divided into four sections, by the type of optimization techniques used: branch and bound algorithm, dynamic programming, heuristic and metaheuristic.

### 2.1.1 Branch and Bound Algorithm

Murray (1998) approached the multiple target access problem (MTAP) to build a road network using mixed integer programming model solved using the branch and bound algorithm. An eight-by-eight grid cell surface was used as a map, and in the map, there were three subsets; harvest site, unharvested site and access location.

The result showed that the intersection points that reduced the length of the road network was a a Steiner node. However, since the Steiner problem is NP-hard, the more intersection nodes there are, the less feasible is the computation of an optimal solution within a reasonable period of computing time.

Najafi and Richard (2013) applied a mixed integer programming model to a real forest, connecting 298 nodes. The objective function was to minimize the sum of the road network, timber transportation, and skidding costs. They used the MIP gap to compare the obtained solution to its optimality, and the solution found was within 80% of the optimal solution's objective function value.

### 2.1.2 Dynamic Programming

Dynamic programming is a recursion process which reduces the size of the problem to smaller subsets and then solves each small problem and combines them (Akay *et al*. 2013). Dynamic programming solutions yield the mathematically optimal solution to a model. Teasley (2002) showed that dynamic programming could be used successfully to solve road location models. Tan (2000) compared dynamic programming to heuristic algorithms in determining an optimum route for connecting a remote landing-node to the current road network. Unlike heuristic methods, if only one road is to be located, dynamic programming can ensure that the solution is the mathematical optimum. Tan (2002) observed that the computation times for the dynamic programming model take much longer than heuristic methods.

### 2.1.3 Metaheuristic Algorithms

Metaheuristics do not guarantee optimal solutions, but their advantage is in solving large-sized problems in a reasonable time period by finding near-optimal solutions (Reeves 1993). Chung and Sessions (2001) used two metaheuristic algorithms; simulated annealing and a genetic algorithm, to search for an optimal road location with criteria based on accessing timber harvest sites using the lowest cost. They used a forest map of small-sized grid cells, of 25m by 25m, due to the complexity of the road network problem. Akay (2004) used two methods: linear programming and simulated annealing, to find the optimal path connecting 2 points in 55 ha of forested land. Two constraint types were used: i) balancing cut and fill; and ii) selecting paths within vertical and horizontal alignment constraints. The objective function was to minimize total road costs, including earthwork costs conforming to design specifications, environmental impact, and driver safety. Meignan *et al.* (2012) used a greedy randomized adaptive search procedure (GRASP) based on a local search procedure to select a set of road locations to minimize harvesting and construction costs. Then, by comparing the cost of solutions to the solutions manually obtained, they reduced the cost of the road network by 8.37%. Even if the computerized solutions may need to be adjusted by an expert, it provided good initial solutions to planners.

### 2.1.4 Heuristics

Heuristic algorithms are used to speed up the process of computing a satisfactory solution, when an exhaustive search is impractical (Desale *et al.* 2015). Heuristics have been researched extensively and applied to designing road networks in forest management.

Dean (1997) developed three different heuristics for solving forest road problems by minimizing building road cost: fully independent path technique, independent paths with reduction, and branch evaluation. The heuristics were compared with a complete enumeration method including all possible branching points for each branching pattern. Among the three heuristics, branch evaluation was closest to the optimal solution obtained from the complete enumeration.

Shirasawa and Hasegawa (2014) exhaustively compared 14 different heuristics for locating a forest road network of minimum cost in the multiple target access problem (MTAP). Most of the 14 heuristics in this research were originally designed to solve the Steiner tree problem. The results showed that the performance of the repetitive shortest path heuristics were superior to other heuristics.

A popular heuristic algorithm, widely used in road location models, is the shortest path algorithm. The shortest path algorithm is used for finding the shortest path between two points on a weighted graph. Among the several shortest path algorithms, the Dijkstra algorithm (Dijkstra 1959) has been used broadly to locate the shortest path between two points in the problem of forest road location. This algorithm has been extensively used since the 1970s with the introduction of computer-aided engineering in designing road networks (Heinimann *et al*. 2003, Turner 1978). Chung *et al*. (2008) introduced an optimization model which designs an economic road network, and its purpose is to maximize the economic service area of a road network rather than connecting specific destination points. The algorithm applied the shortest path algorithm to calculate the minimum cost network. Liu and Sessions (1993) used Dijkstra's shortest path algorithm to find the least-cost road segments from entry points to destinations while considering construction, maintenance and transportation costs over multiple time periods and various topographic conditions. They used the computer program NETWORK II (Sessions in 1987) developed for solving transportation routing problems. Tan (1999) developed a shortest path heuristic for the road location problem. Tan (1999) used a metric representing road cost in the objective function and demonstrated how it could be used to select the optimal location for a forest road. The cost metric included extraction, transportation and road construction costs, and these costs are related to the road length and travel distance. To locate forest roads that have minimal cost, the Dijkstra algorithm was used. Anderson and Nelson (2004) developed an algorithm for creating forest road networks based on the shortest path algorithm. Based on a gridded on map, roads connecting neighbouring cut-blocks were generated using Dijkstra's algorithm subject to horizontal and vertical constraints. Implementing sensitive analysis verified the robustness, stability and long-term effect of the roads. Stuckelberger *et al.* (2007) also introduced a similar approach to Anderson and Nelson (2004) and used the minimum spanning tree and Steiner tree heuristic to connect cut blocks. In this thesis, a library of candidate forest roads was generated for our dataset based on the method used in Anderson and Nelson (2004).

## 2.2 Tactical Harvest Scheduling Models

The literature on harvest-scheduling models is immense. In this review, we restrict ourselves to reviewing tactical level models which schedule cut-blocks and roads simultaneously, since this is the type of model used in this thesis.

The earliest models which integrated timber harvest scheduling and road building were developed in the 1970’s. Weintraub and Navon (1976) used a mixed integer linear programming model and used binary variables to represented whether roads were constructed or not. An increase in the number of binary variables became a computational burden as the number of roads increased. This is because the problem of finding the optimal road network is NP-hard; i.e., the computing time needed to solve the problem increases exponentially as the number of binary decision variables increase (Weintraub and Navon 1976). Also, Weintraub and Navon (1976) were the first to show that solving the two problems of (i) scheduling cut blocks and (ii) locating a road network, simultaneously, produced a better solution than solving these problems sequentially (i.e., scheduling cut-blocks first and then locating a road network). The model in which these two problems are solved simultaneously is called the integrated model. They used a branch and bound algorithm and showed that the integrated model produced solutions 7% more valuable, on average, than solutions generated using the sequential approach.

Kirby *et al.*(1986) and Jones *et al.*(1986) used an integrated tactical harvest-scheduling model to further demonstrate the advantage of solving the cut-block scheduling and road network location problems simultaneously rather than sequentially. They solved the model using the branch and bound algorithm. They showed that the integrated model produced solutions that were 45%, more valuable than the same problem instances solved sequentially.

In the 1990s, adjacency constraints became commonly used in tactical planning (Goycoolea *et al*. 2009). Adjacency constraints entail that no pair of cut-blocks that are adjacent to one another can be cut in the same period (Murray 1999). The purpose of adjacency constraints was to control the area of cut-blocks in order to satisfy the environmental and conservation objectives. Constraints incorporating adjacency constraints made the harvest scheduling problem more difficult because binary variable were now required to represent whether a discrete cut-block was harvested in a given period or not. As a result of using binary variables in the harvest scheduling problem, the model represented a combinatorial optimization problem, which was NP-hard (Murray 1999). Hence, as the problem size (i.e., number of cut-blocks) of the harvest-scheduling problem increased, the solution space and computing time needed for finding an optimal solution increased exponentially (Lockwood and Moore 1992, Murray 1999). A model with adjacency constraints must comply with a maximum harvest area policy. Many papers have been written about modeling solving the harvest-scheduling problem with adjacency constraints(e.g., Lockwood and Moore 1992, McDill and Braze 2000, Murray 1999, Nelson and Brodie 1990).

Metaheuristic algorithms were introduced to solve the harvest-scheduling model because of the computational challenge resulting from adjacency constraints. Metaheuristic algorithms are used to solve NP-hard problems with near-optimal solutions, rather than mathematically optimal solutions (Richards and Gunn 2000). Even though using metaheuristic algorithms can obtain feasible solutions in large-sized problem instances in reasonable periods of computing time, they cannot guarantee proximity to the optimal solution, unlike exact algorithms, such as branch and bound. Relatively few papers have used metaheuristics to solve the integrated model (Naderializadeh and Crowe 2020). Clarke et al. (2000) used a minimum spanning tree algorithm to find a road network in an integrated harvest-scheduling model. They nested the minimum spanning tree algorithm within a three stage-heuristic harvest-scheduling algorithm. The minimum spanning tree (MST) algorithm was suitable for nesting within the integrated model because it executes quickly. Fast execution is required because metaheuristic algorithms are iterative and as a result the MST must be executed thousands of times on different cut-block schedules as the metaheuristic search algorithm explores the value of diverse solutions. Richards and Gunn (2000) used a fast-executing minimum Steiner tree algorithm (SPOH) for finding economical road networks and integrated it within a tabu search metaheuristic for solving the integrated model. There were no benchmarks to compare its proximity to optimum; but Richards and Gunn (2003) later developed a diverse and robust tabu search algorithm, showing a better objective function value. Naderializadeh *et al*.(2020) integrated transportation costs and road construction costs with a simulated annealing metaheuristic to solve the integrated model. Even though it was implemented in a small dataset, Naderializadeh *et al*. (2020) showed that transportation costs can be reduced with a metaheuristic algorithm.

This literature review on the integrated harvest-scheduling model shows the innovation of the objectives of this thesis in two respects. First, the work in this thesis uses a heuristic algorithm for the Steiner tree problem which has never been used within the metaheuristic solution method for the integrated harvest-scheduling model. This heuristic is called Reduction Techniques(RT) (Rehfeldt 2015) and it is compared with previously published road network location algorithms used within the integrated model (i.e., its solutions are compared to solutions in which SPOH and the minimum spanning tree algorithms are used within the integrated model). Second, the work in this thesis uses a road network repair algorithm, Triple contraction heuristic (Zelikovsky 1993), to be used on the road network generated using an integrated model. This algorithm executes slowly but generates higher quality solutions than fast-executing algorithms. The exploration of the road network repair algorithm in this thesis is a major innovation.

## 3. Methods

The Methods section is divided into three major parts: first, we describe how the candidate forest roads were generated before the harvest scheduling model was solved. Second, we describe how the integrated forest harvest-scheduling model was formulated and solved. Third, we describe how the road network in the final solution was repaired and improved.

### 3.1 Generating a Set of Candidate Forest Roads

To generate a set of candidate forest roads, we relied heavily on the methods described in two papers: Anderson and Nelson (2004) and Stuckelberger et al. (2007).

To generate the set of roads, three layers of information were used in a geographic information system. First, we used a Provincial Digital Elevation Model (PDEM) from Ontario GeoHub to represent the elevation value. An elevation gap between two adjacent nodes is used to calculate a vertical constraint. The second layer is the world topographic map, which shows unavailable areas for building candidate roads, such as lakes, streams and rivers. The basic rule used is that every road cannot be built to cross lakes or rivers, but can cross streams with a penalty cost. The third layer represents forest polygons in a real-sized forest. In this paper, 900 polygons were used, and the forest polygons represent portions of the Kenogami forest, located in the boreal forest of Ontario, Canada. Each forest polygon contains a particular stand type, area, age, and yield curve.

Upon these three layers, a squared grid with 50 m X 50 m, was overlaid. On this grid, one node exists in the center of every square, and a total of 16 direct road-links emanate from each node to connect its adjacent nodes. The approach of sixteen directions used in this paper offers much more flexibility to road networks due to the increased number of candidate road-links. The basic design of the grid used is presented below the below in Figure 1.

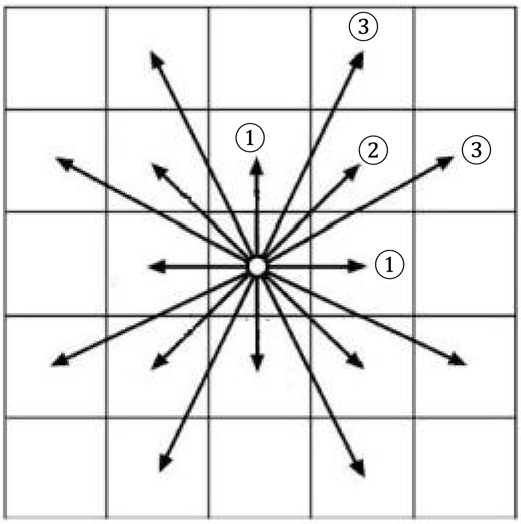


**Figure 1:** Grid of sixteen road links (arcs) used to generate candidate roads.

Second, some of the generated road-links (arcs) and nodes may be located in an unworkable area for building roads, such as rivers or lakes. Therefore, these arcs and nodes were removed. Also, some arcs have infeasibel elevation differences between their two nodes. We judged that road-links could not be constructed if the elevation difference was more than 15% or less than -15%. Thus, road-links with such unfavorable values were removed from the set. Based on this rule, a total of 65,150 nodes and 408,778 candidate road arcs were generated. From these arcs and nodes, the set of candidate roads were built.

After removing unnecessary or infeasible road-links, landing nodes, which represent each harvested polygon, were placed on the largest and flattest terrain within each forest polygon. The location of the landing node is quite critical in the road network. Since roads connect two landing nodes, the length of the road can vary, depending on the placement of the landing nodes.

The third step for designing candidate roads is to assign costs to each road-link (arc). In this thesis, we used the simplifying assumption that costs are proportionate to the length of the road. Input from a forest engineer would be needed for a more realistic appraisal of these costs. For example, assuming that a road is built on a flat area with no penalty, the weights for the road are calculated as follows. First, the road length connecting immediately right or left or upper or below vertices is set to 50. Second, the road length connecting to diagonal vertices multiply a square root of 2 by the first road length (see Table 1 below and Figure 1 above).

**Table 1:** Assigning Road construction cost based on the its road length

| **Road type**<br>(See figure 1) | **Assigned value of Cost based on Length (m)** |
|---|---|
| ① | 50 (a distance to immediate right or left node) |
| ② | 71 (a distance to diagonal nodes, 50 X a root square of 2) |
| ③ | 112 (a distance to diagonal nodes, 50 X a root square of 5) |

The final step is to generate a set of candidate roads that connect all pairs of adjacent polygons. This is the set of candidate roads that is to be used in the harvest-scheduling model. The method of generating a set candidate road is based on Dijkstra's (1959) shortest path algorithm. The shortest path between all pairs of landing nodes in adjacent polygons was found, subject to vertical and horizontal constraints. This use of the shortest path algorithm was developed by Anderson and Nelson (2004) and Stuckelberger et al. (2007). The constraints on the shortest path algorithm are presented in Table 2, below.

**Table 2:** Road standards constraining shortest path algorithm.

| Road Standards |
|---|
| – Favourable grade: limited by penalties for steeper grades.<br>– Abrupt grade change: limited by penalties for abrupt changes in grade.<br>– Radius of curve: limited by penalties for abrupt changes in horizontal alignment.<br>– Minimal stream-crossings: links that cross streams were penalized.<br>– Minimal switchbacks: penalize switchbacks |

To meet the standards listed in Table 2, we set three components into the shortest path algorithm: a threshold value, a penalty coefficient, and a limit.

First, for the threshold value, any edge which violates the limits on the road standards was assigned a penalty value. Any arc crossing water streams or more than 15% of slope change was assigned a penalty cost. Second, even though roads do not violate the limit, if the value exceeds the threshold value, we impose penalties proportionately based on the value deviating from the target value (penalty coefficient). We set the threshold value in the vertical constraint to 10%. Every edge exceeding the threshold value of the slope change received a penalty coefficient multiplied by the difference between its slope change and threshold. Therefore, the actual road distance of two arbitrary roads will be different based on the terrain, even though the two roads are represented as the same length on a map.

Horizontal constraint on the shortest path were set to 90°. Only directional changes more than 90° are allowed, and any abrupt change with angles equal to or less than 90° are prohibited. Unlike the vertical constraint, the horizontal constraint does not use a penalty. We implemented this constraint by dividing the roads into sixteen types based on their direction and allowing only selected types to be combined. Therefore, through setting the three types of constraints (vertical, horizontal, and stream crossing) every generated road by Dijkstra's shortest path algorithm meets the road building standards in the above list (Table2). These roads are not of the quality generated by an operational forest engineer, but they suffice for tactical planning in which approximate feasibility of a road's location is needed.

## 3.2 Integrated harvest scheduling model.

The mathematical formulation of the integrated model used in this thesis is presented below.

***Indices and sets***

$k, K$ = *Indices and set of polygon k*
$I, j, I$ = *Indices and set of nodes*
$s, S$ = *Indices and set of species demanded*
$t, T$ = *Indices and set of time periods*
$B$ = *Set of intermediate (transshipment) nodes*
$C$ = *Set of destination nodes*
$D_t$ = *Set of stands not eligible for harvest in period t*
$A$ = *Set of directed arcs*
$I_i$ = *Set of arcs directed into node i*
$I_k$ = *Set of arcs directed into stand k*
$O_i$ = *Set of arcs directed out of node i*
$O_k$ = *Set of arcs directed out of stand k*
$O_{kt}$ = *Set of stands, including stand k, that are harvested during period t*

***Parameters***

$APC_{st}$ = *Allowable period cut of species s in period t* ($m^3$ *per period*)

$v_{skt}$ = *Volume harvestable of species s from stand k, period t (m*$^3$*)*

$c_{ijt}$ = *Discounted cost of building an arc from node i to node j in period t ($)*

$r_{kt}$ = *Discounted revenue from harvesting stand k in the period t ($)*

$a_k$ = *Area of stand k (ha)*

$O_{max}$ = *Allowed maximum opening area for harvest stands (ha)*

$M$ = *Maximum value of allowable period cut for all the species in each period (m*$^3$*)*

***Decision Variables***

$x_{kt}$ *1 if stand k is harvested in period t, 0 otherwise*

$y_{ijt}$ *1 if arc from node i to node j is built in period t, 0 otherwise*

The objective function of this model is to maximize present revenue minus the discounted costs of road construction.

**Maximize**

$$\sum_{k\in K}\sum_{t\in T} r_{kt}\, x_{kt} - \sum_{(i,j)\in A}\sum_{t\in T} c_{ijt}\, y_{ijt} \qquad (1)$$

**Subject to:**

Each stand may be harvested only one time during the planning horizon.

$$\sum_{t\in T} x_{kt} \le 1 \quad \forall k \in K \qquad (2)$$

Some of the stands could be not eligible for harvest, either by age or other reasons.

$$x_{kt} = 0 \qquad \forall\, t \in T, \quad k \in D_t \qquad (3)$$

The adjacency constraint is given in equation (4). For adjacency constraints, we used the area-restricted model (ARM). The ARM entails that when harvesting more than one adjacent polygon in the same period, the total area cannot exceed the allowed maximum opening size (McDill et al. 2002). This constraint is more computationally expensive than URM (Unit-restricted model) because, in order to follow this constraint, examining a combination of the adjacent polygons is necessary.

$$\sum_{k \in u} x_{kt} a_k \leq O_{max} \qquad \forall k \in K, \qquad t \in T \tag{4}$$

The volume to be harvested in a given period for each species cannot exceed the target volume.

$$\sum_{k \in K} x_{kt} v_{skt} \leq APC_{st} \qquad \forall\, t \in T, \qquad s \in S \tag{5}$$

A road may be built only once during the planning horizon.

$$\sum_{t \in T} y_{ijt} \leq 1 \qquad \forall\, (i,j) \in A \tag{6}$$

A road connecting two nodes(*i* and *j*) must be built in either period prior to the period or in the period of its harvest.

$$\sum_{s \in S} x_{kt} \leq M \sum_{t \in P_t} y_{ijt} \qquad \forall\, (i,j) \in A, \qquad t \in T \tag{7}$$

Only one road of the multiple possible roads connecting node *i* to *j* can be built.

$$\sum_{(i,j) \in O_i} \sum_{t \in T} y_{ijt} \leq 1 \qquad \forall\, i \in I, \qquad i \notin C \tag{8}$$

At least only one road connecting the forest's entry node must be built.

$$\sum_{(i,j) \in I_i} \sum_{t \in T} y_{jit} \geq 1 \qquad \forall\; i \in C \tag{9}$$

In the equation (10) and (11), the decision variables are binary.

$$x_{kt} \in \{0, 1\} \qquad \forall\, k \in K, \quad t \in T \tag{10}$$

$$y_{ijt} \in \{0, 1\} \qquad \forall\, (i, j) \in A \quad t \in T \tag{11}$$

## 3.3 Solving the Integrated Harvest Scheduling Model

To solve the integrated model a metaheuristic algorithm called simulated annealing (Kirkpatrick 1984) was used. Simulated annealing has been broadly used to solve the tactical harvest scheduling model (without roads), and several papers have shown its proximity to the optimum (generated using the branch and bound algorithm) to be within 10% (Murray and Church 1995, Boston and Bettinger 1999, Crowe & Nelson 2005). The simulated annealing algorithm explores the solution space by iteratively evaluating neighbouring solutions and accepting inferior neighbouring solutions (as the current solution) with a probability that decreases as the search persists over time. This decreasing probability is analogous to the decreasing temperature of an annealing metal as it cools over time—hence the name, simulated annealing. An outline of the simulated annealing algorithm is as presented below (Dong et al. 2015):

1) Set parameters:

Start temperature (T), the cooling rate (r) with $0 < r < 1$ and the number of iterations allowed at each temperature (nrep).

2) Generate an initial feasible solution S and calculate the objective function value.

3) Set this solution as a current solution (S) and best solution (S*)

4) While T is not frozen,

(i) Perform the following loop nrep times: pick a random neighbour S′ of S. If S′ is feasible, let delta = f(S′) – f(S). If delta $\geq 0$, set S = S′. If delta $< 0$, set S = S′ with probability $e^{(-\text{delta}/T)}$

(ii) Set T = rT

5) Return the best solution S*

To select a neighbour of S is to randomly select any binary decision variable, $x_{ij}$ from the current solution S, and flip 1 to 0 or 0 to 1 (where $x_{ij} = 1$ if polygon $i$ is harvested in period $j$; 0 otherwise). That is,

(i) If $x_{ij} = 0$, let $x_{ij} = 1$.

(ii) If $x_{ij} = 1$, let $x_{ij} = 0$.

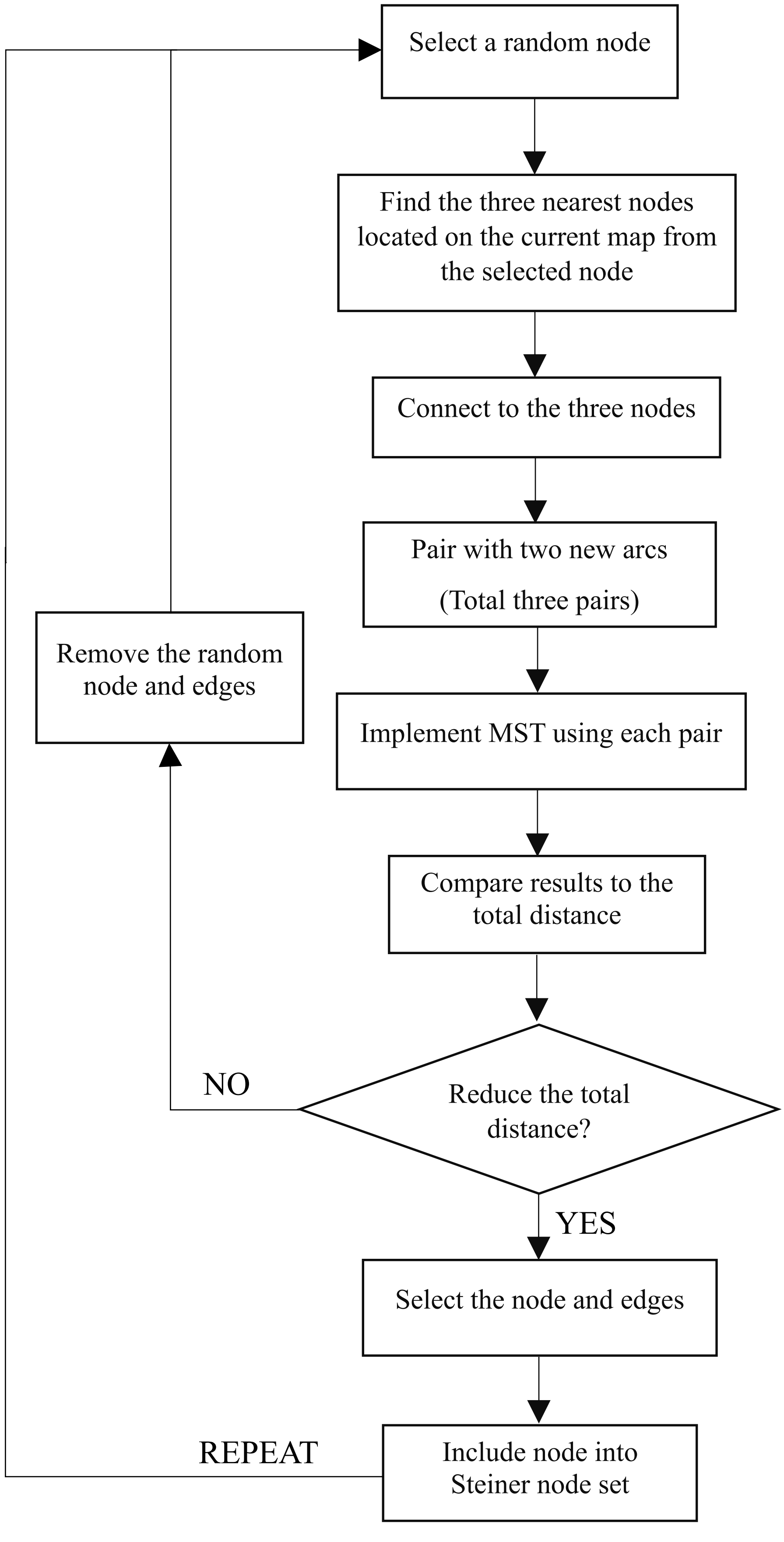
It is important to understand that the simulated annealing algorithm is an iterative search algorithm. That is, it typically generates and evaluates tens of thousands (often millions) of candidate solutions in order to find the solution of highest value. To generate and evaluate a harvest-scheduling solution is computationally quite simple; but for each harvest-schedule generated, a required road network must also be generated and its cost calculated. The computational cost of allocating a new road network and calculating its cost for each solution is major and slows down the search for an optimal solution. For this reason, it is imperative that the algorithm used for calculating road networks, within the metaheuristic search, must execute quickly. In this thesis, we experimented with three fast heuristics for building the road network and compared their results on execution time and solution quality. These three heuristics are: i) the minimum spanning tree (MST); ii) shortest path origin heuristic (SPOH); and iii) the reduction technique (RT).

The minimum spanning tree algorithm (Prim 1957) selects a set of road arcs that connect all required nodes, and the forest's point of entry, with minimum total cost for all selected arcs. SPOH (Takahashi and Matsuyama 1980) builds a candidate road network connecting all scheduled polygons using the shortest path from the entry point. SPOH starts with an arbitrarily picked terminal node, then selects a node with the shortest path distance to it and iterates the process until connecting all terminal nodes (Sadeghi & Fröhlich 2013). RT (Rehfeldt 2015) is an algorithm based on the minimum spanning tree. Unlike MST and SPOH, this algorithm has not been used within the integrated harvest-scheduling model, prior to this thesis. The algorithm is a network cost reduction technique that is based on a network established using the Kruskal minimum spanning tree rather than Prim's (Kruskal 1956). The basic idea of RT is that if an unselected polygon is connected by only one road, the corresponding polygon and road must be removed. In the same manner, after removing the polygon and road, if its adjacent polygon is an unselected polygon and has only one road, this polygon and road also should be removed using a recursive method. Hence, the unselected nodes are removed sequentially and exhaustively. Therefore, removing the branches which consist of unselected nodes would decline the total distance effectively. In addition, since this algorithm is based on the MST, it is useful to compare it with the SPOH modified in the shortest path algorithm.

## 3.4 Repairing the Road-Network

To reduce the building road network cost obtained from the three heuristics, we used the triple contraction heuristic (Zelikovsky 1993) to improve the road networks after the tactical harvest-scheduling model had finished executing. Figure 1 shows, in a flow chart, the procedure of the triple contraction heuristic.

**Figure 2:** Flowchart of the Triple contraction heuristic



New connecting roads have flexibility since the roads were not limited by the preprocessed road library and were built at the 50m x 50m grid level. On the other hand, this flexibility creates computational challenges. It means that to connect a Steiner node, the shortest path algorithm (Dijkstra's algorithm) should be executed, which is subject to a horizontal constraint, at least nine times:

i) Calculate the shortest path from the selected candidate Steiner node to the closest node on the map (1 time).

ii) Calculate the distance from the closest node to each end of the edge on the map (2 times).

This process repeats three times to find one Steiner node and the resulting distance. In addition, as the distance between the selected candidate Steiner node and the closest node is far, the search space increases exponentially. This step leads to a considerable computational burden as the repetitions increase. Using the repair algorithm can, however, improve the objective function value and could be much more realistic in terms of the shape of roads. This is because it removes traversing, unnecessary areas on a detailed level and makes the road’s path smoother. Therefore, we added a sixth step to our algorithm for solving the tactical planning model.

6) After T is frozen:

6-1) Divide the nodes into terminal nodes and candidate Steiner nodes. The entry node and scheduled nodes are designated as terminal nodes, and remaining nodes are designated as candidate Steiner nodes, except for nodes on the current graph.

6-2) Randomly select one node from the set of candidate Steiner nodes. The selected nodes are then connected with the three closest nodes in the current graph. Pair each with one other edge and divide the three edges into three subsets. The minimum spanning algorithm then runs, and the overall length is compared. If the total distance decreases, accept the new edge, otherwise discard it.

6-3) Iterate the above processes while specific conditions are satisfied.

## 3.5 Case Study

The case study for this thesis was a portion of the Kenogami Forest, located in the boreal forest of Ontario, Canada. The total area is 16,300 ha. This area of the forest was divided into 900 polygons ranging in size from 5 ha to 133 ha. The dataset for the forest contained eight different yield curves, and stand ages ranged from 5 to 210 years. The forest also contains several rivers and small lakes. The terrain of this forest is uneven, with elevations of merchantable wood ranging from 300 to 540 metres above sea level. The minimum rotation age for each stand was set at 70 years. Timber revenue were divided into three classes, based on age: \$75 per $m^3$ (ages 70 to 120 years), \$90 per $m^3$ (ages 121 to 180 years), and \$110 per $m^3$ (ages greater than 180). The road-building cost was simplified by basing it on the length of the road constructed (\$30,000 per km). Adjacent cut-blocks were defined as blocks sharing a common boundary, and the maximum allowable area of a cut-block was limited to 80 ha.

The model and search algorithms described in the methods were written in the JAVA programming language, and all program instances were executed using an Intel® Core™ i5 CPU (2.30GHz), with 8 gigabytes of RAM.

# 4. Results

The Results are presented in three parts: i) the set of candidate roads generated; ii) the solutions to the harvest scheduling model using three different roading heuristics and results from the repair heuristic; and iii) maps of the solutions.

## 4.1 Results From Preprocessed Road Library

The number of possible road links (arcs) generated for the Kenogami forest was 840,000. The density of these arcs varied depending on the terrain. From these arcs, a set of candidate roads was generated linking all pairs of adjacent polygons. The total number of candidate roads, connecting all pairs of adjacent polygons, generated was 3,556. After the roads were generated, they were exported to into a GIS viewer and inspected for feasibility. Figure 3 (below) represents one candidate road (in red) connecting the landings of two polygons (a green polygon and a red polygon). The underlying green arcs are the candidate road-links. Inspection of Figure 3 show that the road follows the three constraints. That is, the road:

a) does not violate a gradient constraint of 15%;

b) avoids crossing a large water body; and

c) does not violate the radius of the curve constraint.

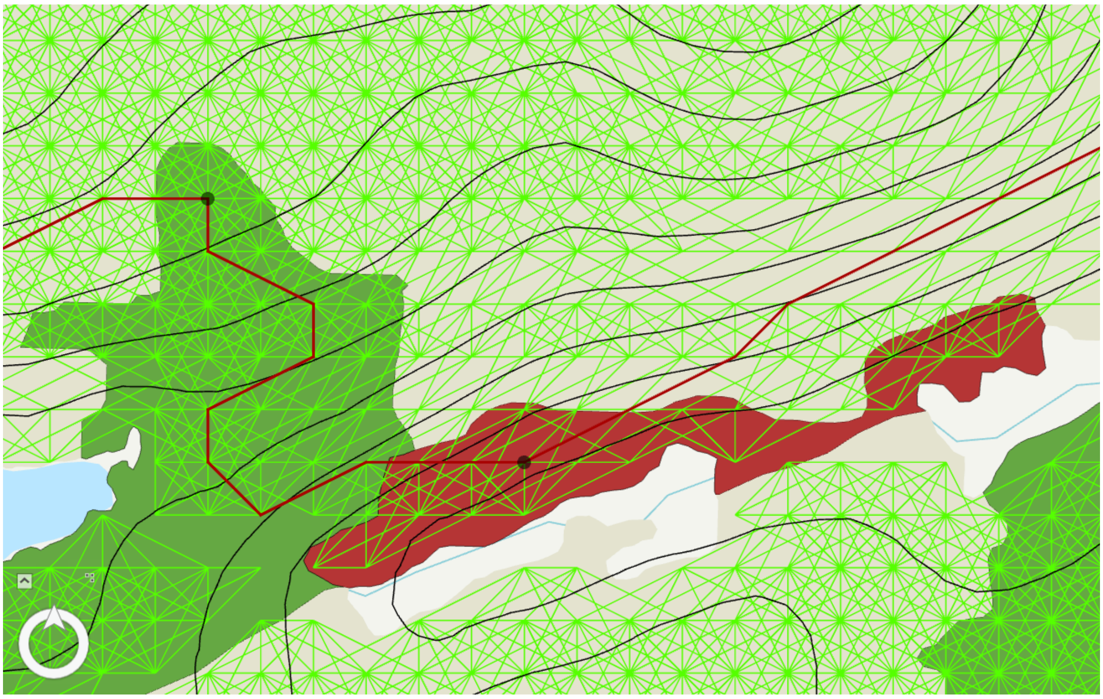


**Figure 3:** Example of candidate road generated using shortest path algorithm.

### 4.2 Solutions of Harvest-Scheduling Model

#### 4.2.1 Progress of Search Algorithms over Time

The first result to be reviewed in the solutions to the harvest-scheduling model is the progress of the objective function value over time as the model was solved, using the simulated annealing algorithm and the road network construction algorithms using the following road-construction heuristics: i) the MST algorithm; ii) the SPOH algorithm; and iii) the RT algorithm.

Figure 4 (below) shows the progress of the harvest-scheduling model's search using the MST algorithm.

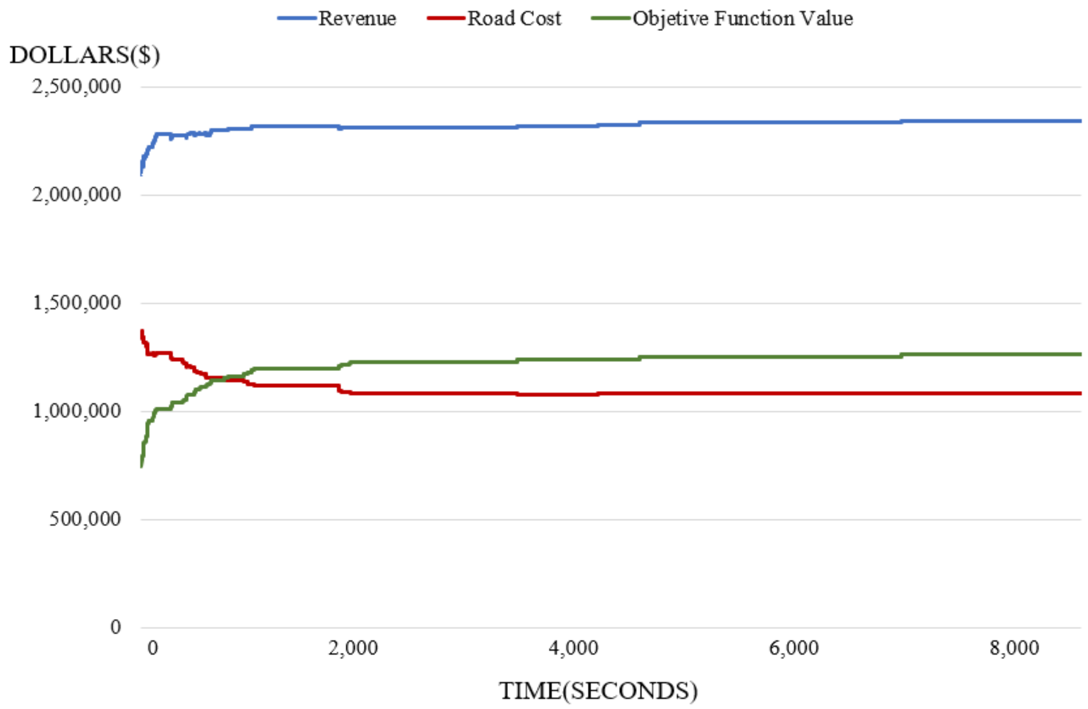


**Figure 4:** Progress of MST over time.

In Figure 4 we can observe that the increase in the objective function value over time correlates to the increase in revenue, rather than the decrease in road construction cost. The overall pattern of the road construction cost does not fluctuate significantly after reducing costs to a certain level in the initial phase of the model's execution.

Three heuristics were executed under the same conditions, and the execution time of the harvest-scheduling model using MST slightly exceeded 8,000 seconds. The average number of iterations per minute was approximately four thousand. This long execution time stems from the dataset size. SPOH uses individual adjacent paths connecting two adjacent polygons, while MST uses every possible combination which can be made by individual adjacent paths. As a result, the number of elements in the dataset increases drastically and leads to increasing the execution time when using MST. In the 900 polygons forest, when combining every possible adjacent route, the number of elements of the dataset increased from 4,000 to 80,000.

Figure 5 (below) shows the progress of the harvest-scheduling model's objective function over time while using the SPOH algorithm.

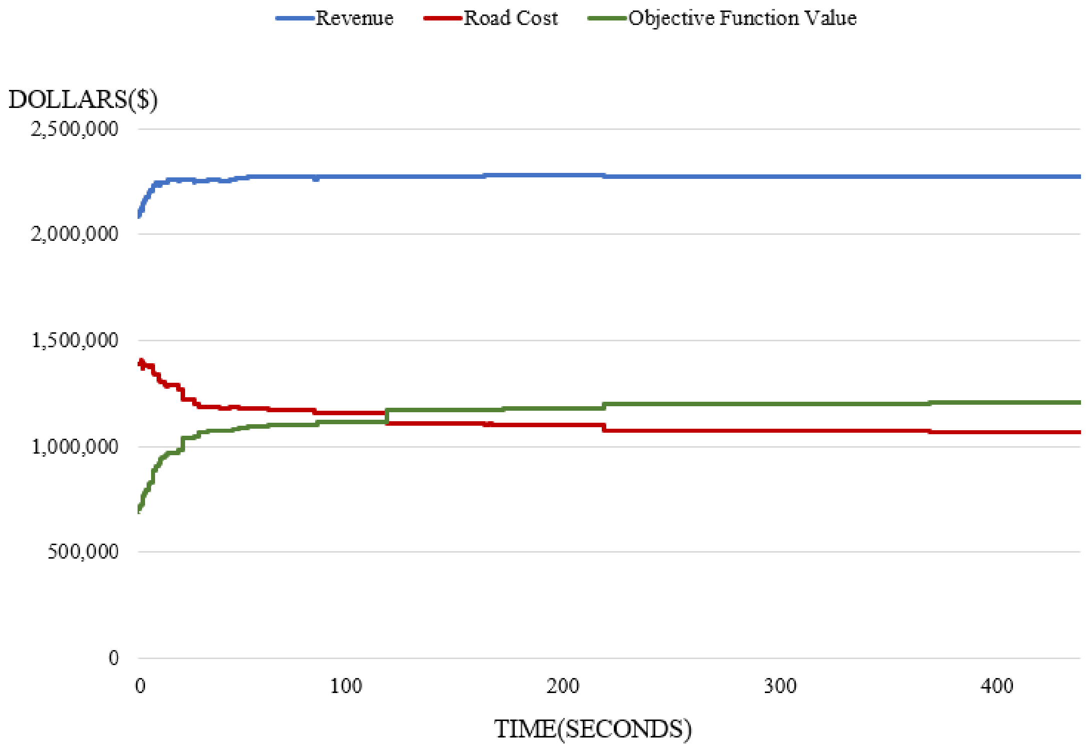


**Figure 5:** Progress of search over time when using the SPOH algorithm.

Figure 5 shows the progress of the search using the SPOH algorithm. Here we observe that, as distinct from the MST algorithm (Figure 4, above) the SPOH algorithm tends to regularly decrease road construction costs over time. The increase in the objective function value in Figure 5 mainly stems from the decrease in road construction costs despite maintained revenue.

Another noticeable result using SPOH is a faster execution time versus MST.

With the same condition, the execution time was faster than MST by a factor of 20. The number of iterations per minute of the model using SPOH about was 70 thousand. Since SPOH takes advantage of the pre-processed road network library, such as the shortest route to its adjacent nodes, it does not use the shortest path algorithms separately to calculate the optimal route. Therefore, this heuristic showed the fastest execution time among the three roading-construction algorithms used, and the total elapsed time was 420 seconds. The pattern of the road construction cost over time in the two heuristics is similar to the results found by Rouhfza (2015).

To observe the progress over time of the decrease in road construction costs, SPOH was executed using a longer run-time, and road construction cost decreased slightly until 30 minutes after execution (over 2,000,000 iterations) and then levelled off.

Figure 6 (below) shows the progress of the harvest-scheduling model's objective function over time using the RT algorithm

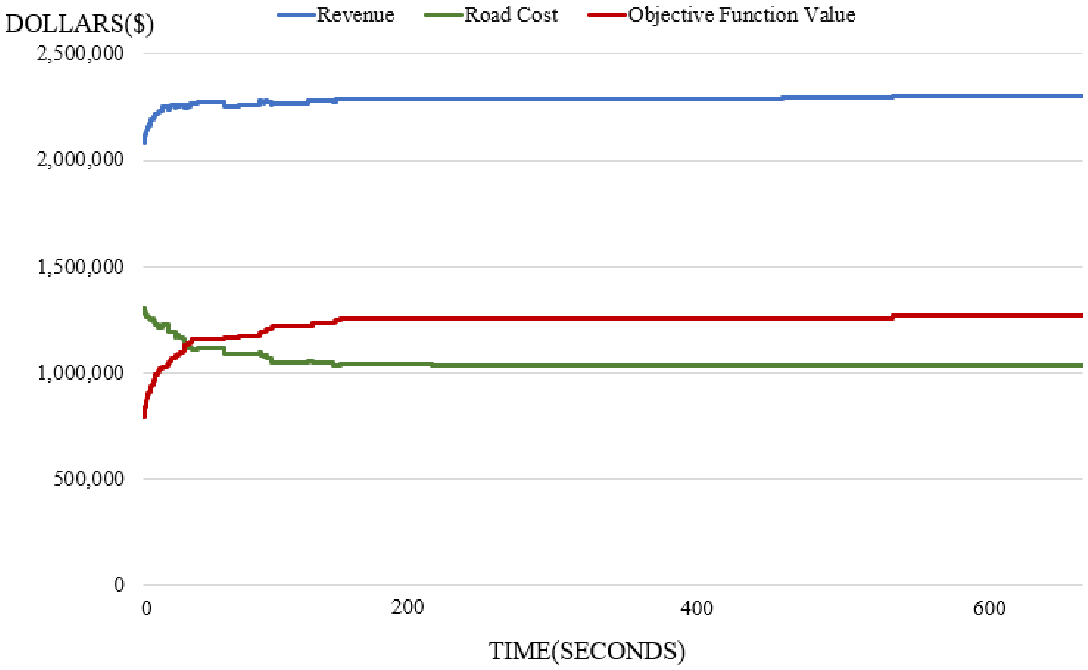


**Figure 6:** Progress of RT over time.

In Figure 6, we observe that the overall progress of the objective function when using RT is similar to MST (Figure 4, above). This result shows the increase in the objective function value over time relies on the rise in revenue rather than the decrase in road construction costs. The total execution time exceeded 600 seconds, thus, RT was slightly slower than SPOH but faster than MST. The average number of iteration per minute of RT was 45 thousand. This means that using RT entails a longer execution time than SPOH by a factor of 1.5. Since RT uses the same dataset as SPOH, therefore one would expect the execution times of SPOH and RT to be the same due to the same number of edges and nodes. However, the recursive method of RT finding and eliminating nodes sequentially and exhaustively slows its execution versus SPOH.

### 4.2.2 Objective Function Values of Solutions Found.

Table 3 present the objective function values of the solutions found using the three different roading algorithms: MST, SPOH, and RT.

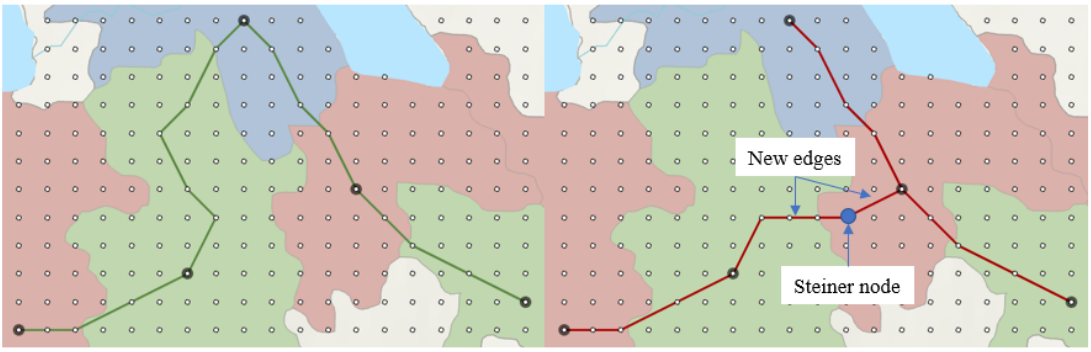
**Table 3:** Comparison of three heuristics (Note: The difference was measured by comparison to the value in RT's row).

| Heuristics | Objective Function ($) | Revenue ($) | Road cost ($) | Difference in objective function% | Difference in Road cost% |
|---|---|---|---|---|---|
| SPOH | 4,946,775 | 6,464,535 | 1,517,760 | -5.1% | 11.3% |
| MST | 5,199,995 | 6,762,185 | 1,562,190 | -0.3% | 14.6% |
| RT | 5,213,725 | 6,577,135 | 1,363,410 | 0% | 0% |

Table 3 offers several noteworthy observations. First, we observe that the solution generated using the RT algorithm had the highest objective function value and the lowest road construction cost. Second, we observe that the solution found using the MST algorithm had the highest revenue, but that, because its road costs were so high (compared to RT), it did not achieve the highest objective function value. Third, we observe that SPOH had the lowest objective function and the second lowest road construction cost. Finally, a general trend that can be observed in Table 3 is that the different road building heuristics had a strong influence not only on the value of the road network solution, but also on the revenue generated from the harvest-schedule.

### 4.3 Results From Repairing Road Network

In order to repair and thereby improve the objective function values of the three solutions generated using MST, SPOH, and RT, we used the Triple Contraction Heuristic (TPH). An example from the results, comparing a road "before repair" versus a road "after repair", is presented in Figure 8 (below).



(a) before repair (b) after repair

**Figure 7:** An example of the application of the triple contraction heuristic

In Figure 7 we observe that, in order to execute a possible repair, a blue point (a.k.a, a Steiner node) was randomly chosen between two polygon landings whose polygons were scheduled to be harvested in the same period and already connected by a road. Next, the blue point was connected to the two nearest landings (i.e., large black nodes), thereby creating a new road link. Finally, since the cost of the new road link reduced the total cost of the existing road, the existing road was replaced with the new, lower cost road.

As shown in Figure 7, the Steiner node was located at a random point between two harvested polygons. The new connecting roads generated using the TPH algorithm were therefore not limited by the preprocessed road library. In addition, they were built directly on the 50m x 50m grid. This procedure therefore creates flexibility in finding new roading options, but it comes with a tremendous computational burden, compared to the fast heuristics. The resulting execution time to generate a repaired road network for this case study was approximately 300 minutes

The results of applying the TPH repair algorithm to solutions generated using MST, SPOH, and RT, are presented in Table 4 (below).

**Table 4**: Results of applying the TPH algorithm to three existing solutions.

| **Road cost** | **SPOH($)** | **MST($)** | **RT($)** |
|---|---|---|---|
| Before | 1,517,760 | 1,562,190 | 1,363,410 |
| After | 1,108,433 | 1,352,236 | 1,183,853 |
| Reduction rate(%) | 26.9% | 13.4% | 13.1% |

Table 4 shows that the reductions in the cost of the road networks occurred in all three solutions and ranged from 13.1% to 26.9%. This shows that the repair algorithm was consistently successful, regardless of whether it started with a high-quality solution (e.g., RT) or a low-quality solution (e.g., SPOH).

The influence of the reduced road construction costs upon the objective function values of three harvest scheduling solutions is presented in Table 5.

**Table 5:** The objective function values after repairing algorithm.

| **Objective function value** | **SPOH($)** | **MST($)** | **RT($)** |
|---|---|---|---|
| Before | 4,946,775 | 5,199,995 | 5,213,725 |
| After | 5,356,102 | 5,409,949 | 5,393,282 |
| Rate of increase(%) | 8.3% | 4.0% | 3.4% |

Table 5 shows that increases in the value of the objective function ranged from 3.4% to 8.3%. These values show that the use of the TPH algorithm road repair step can make a meaningful contribution to improving the value of a tactical plan in forest management.

## 4.4 Mapped Solutions

Six mapped solutions are presented in this section: each of the three solutions generated using MST, SPOH, and RT are presented; and each of the three repaired solutions are also presented.

Figures 8 and 9 (below) present the mapped solutions using the integrated model with the SPOH algorithm. Several observations are noteworthy from these figures. First, the road networks form a tree (containing no cycles and disconnection). Second, several routes connect to nodes (Steiner nodes) other than the scheduled landing nodes (terminal nodes) to reduce the total distance. Third, the opening size does not exceed the set value of 80ha. Fourth, every road does not violate the horizontal constraints except for the intersection points and landing points. Finally, one can observe that the repairs occurred in all three periods and tend to make the paths between cut-blocks more direct and therefore less costly.

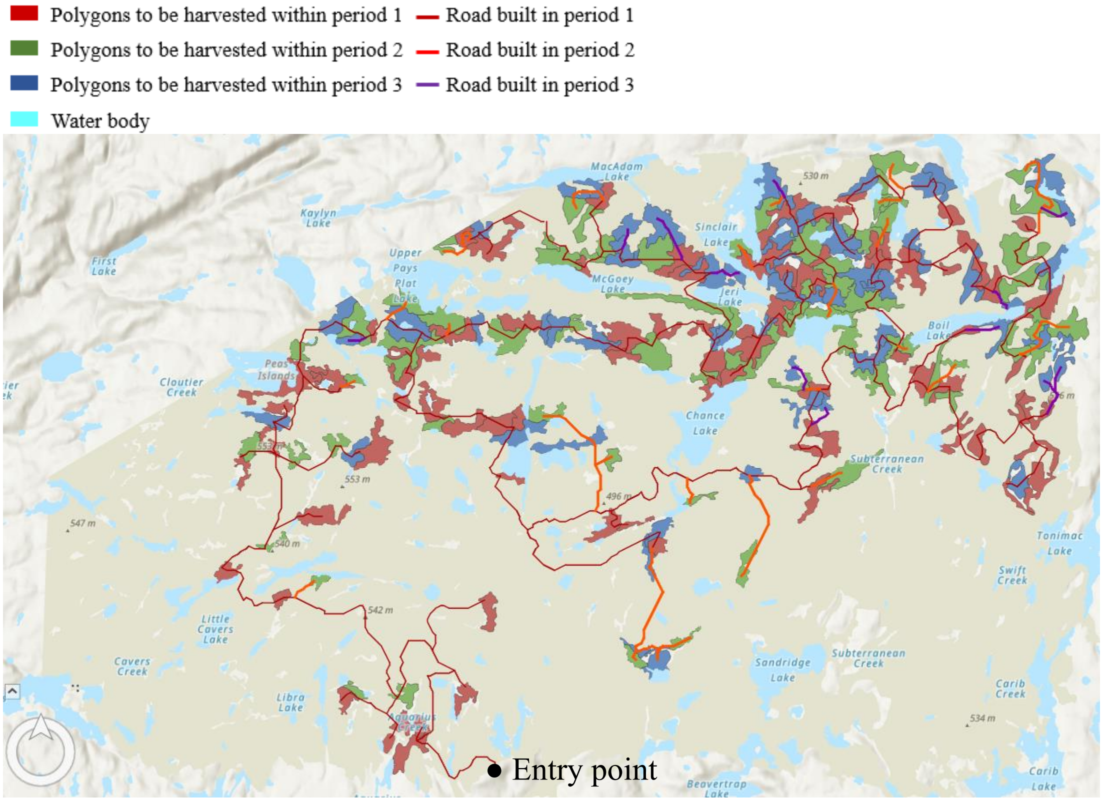


**Figure 8:** SPOH solution mapped before repair algorithm.

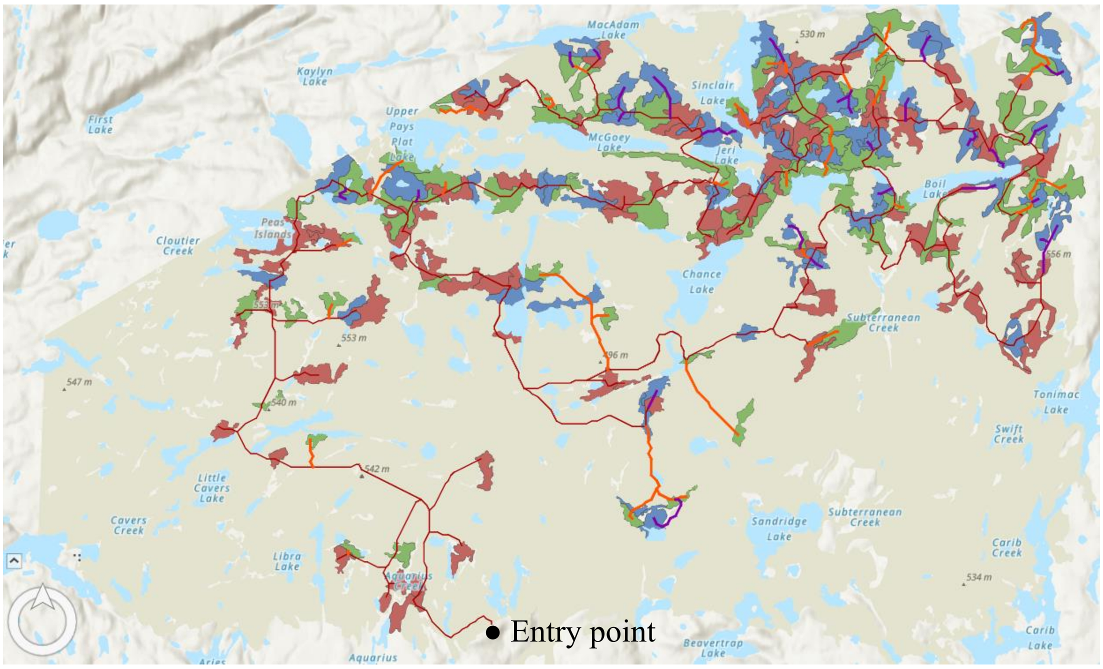


**Figure 9:** SPOH solution after the repairing algorithm

The MST solution before repair is presented in Figure 10 (below); and its solution after repair is presented in Figure 11.

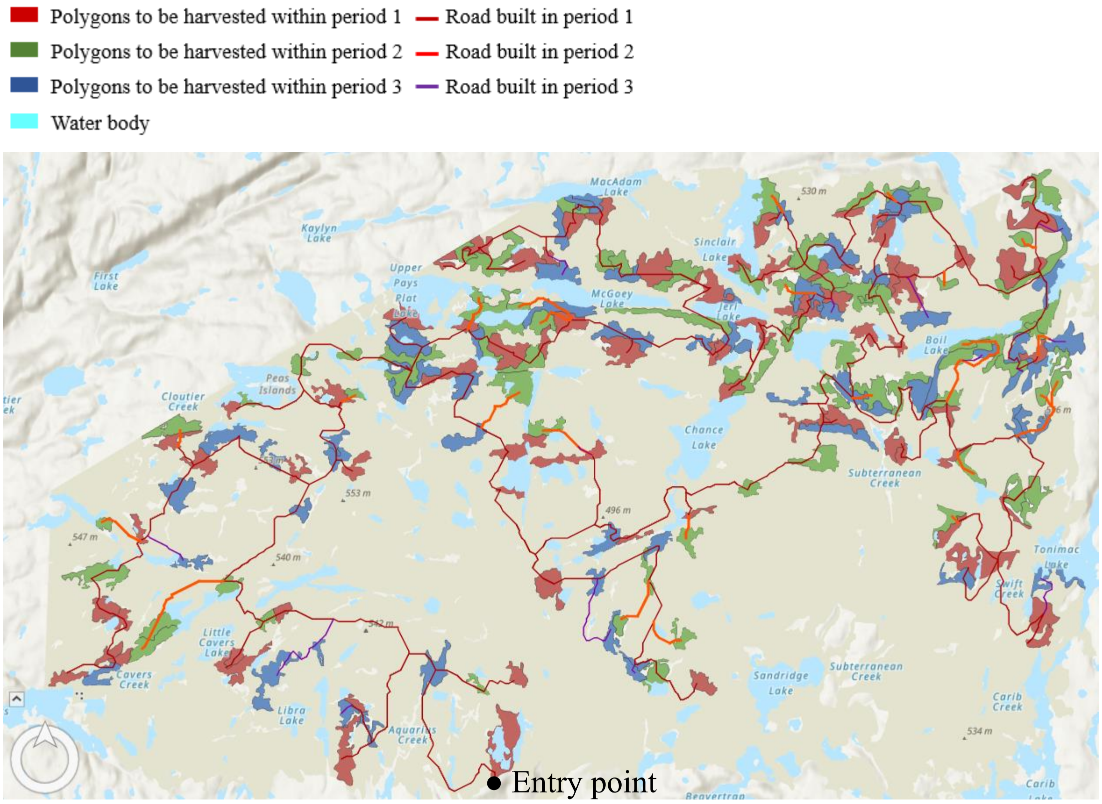


**Figure 10:** MST solution before the repairing step

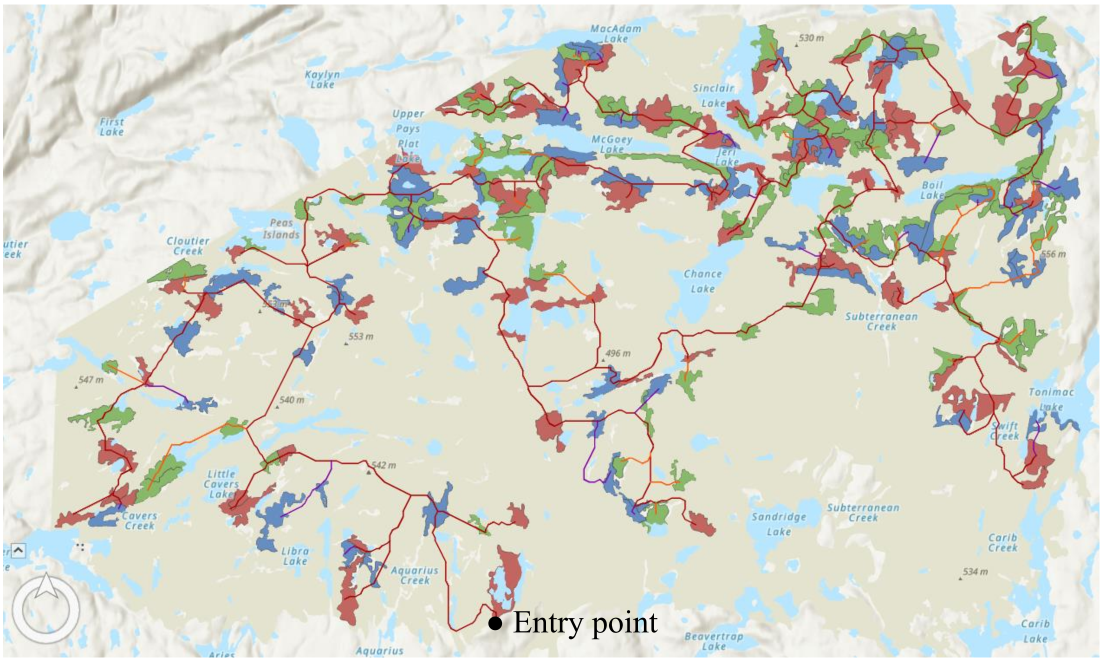


**Figure 11:** MST solution after the repairing algorithm

The RT solution of the harvest-scheduling model before repair presented in Figure 12 and the solution after repair is presented in Figure 13.

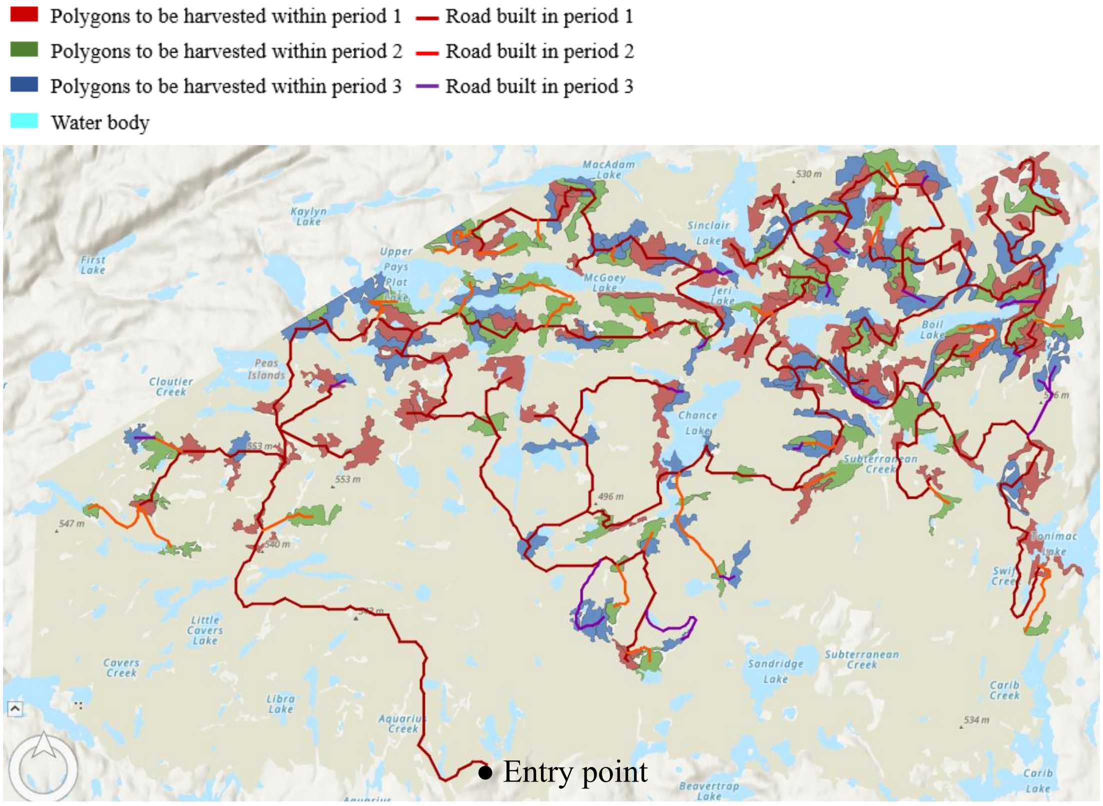


**Figure 12:** Before the repairing step in RT

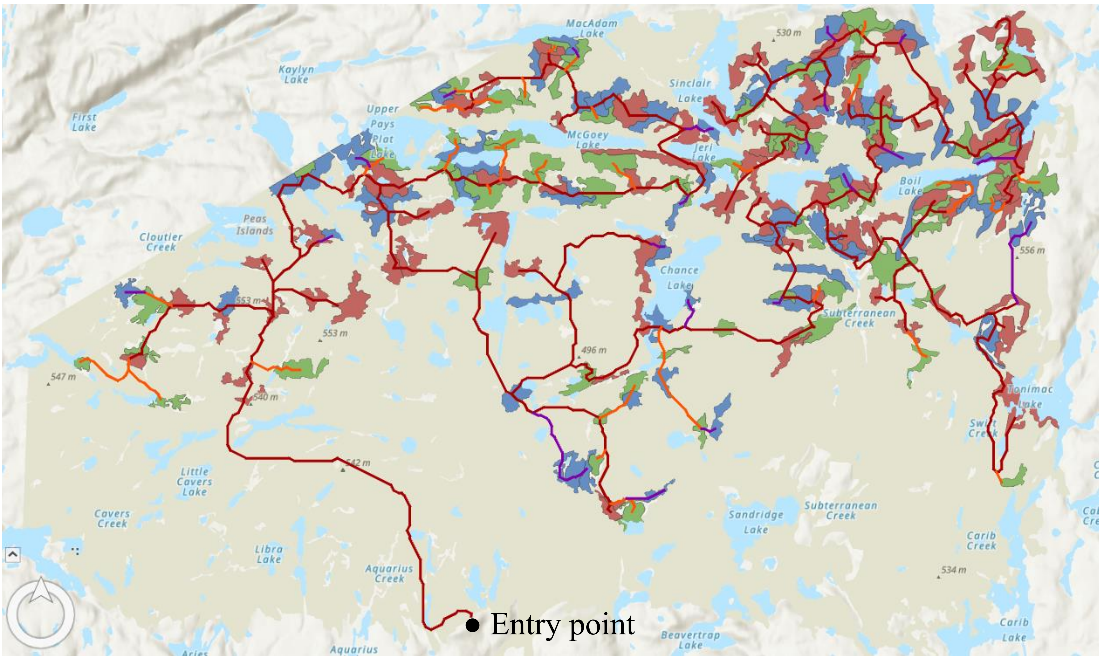


**Figure 13:** RT solution after the repairing algorithm

The mapped solutions above illustrate the following three observations:

i) Unrepaired roads tend to be built somewhat detouring from the immediate landing nodes since the preprocessed road library is based on the adjacent roads.

ii) Unrepaired roads branch only at landing nodes. This is because each unrepaired road has a unique source node and destination node.

iii) Repaired roads tend to have a more direct path.

# 5. Discussion

In this Discussion we address three practical topics that have a direct bearing on the usefulness of this research: i) the possible effect of larger datasets on the computational feasibility of the algorithms reviewed in this thesis; ii) the possible effect of the road-network heuristics on transportation costs; and iii) the practical benefits that might arise from this research.

The first topic of discussion concerns addressing the question of whether the road network heuristics used in this thesis can be used on much larger datasets than the 900-polygon forest of the case study; i.e., whether the computational challenge of using these heuristics would slow down the generation of solutions so much as to be impracticable if applied to much larger datasets. We will address this question in three parts: namely, the effect of larger datasets on: i) the number of candidate roads generated; ii) the use of the roading heuristics nested within the harvest-scheduling model; and iii) the use of the road network repair heuristic.

First, in the stage of generating large-scale road libraries, there is no computational difficulty of increasing the size of the dataset. This is because the number of roads to be built equals the number of pairs of adjacent polygons, and the growth in the number of roads would increase linearly as the number of polygons increases.

Second, to evaluate the practicality of the three fast road-building heuristics nested in the integrated tactical model, we need to examine the execution time of each heuristic. For the three fast heuristics, we used two general types of algorithms, the shortest path-based (used in SPOH) and minimum spanning tree-based (used in MST and RT). To compare the algorithm efficiency in the computer science field, we measure the worst-case time complexity, which is a determination of the maximum amount of time that an algorithm requires to solve problems of size n, called Big 'O' notation (Devi et al. 2011). According to Shirasawa (2014), the worst-case efficiency of MST and SPOH is the same. As shown in Table 4, the execution time is directly and strongly affected by the number of edges rather than nodes. Therefore, the long execution time in MST is not surprising.

**Table 6:** The time complexity of MST and SPOH

| Heuristic name | Worst-case time complexity |
|---|---|
| Minimum spanning tree heuristic | $O(\|E\| + \|V\| \log\|V\|)$ |
| Shortest paths origin heuristic | $O(\|E\| + \|V\| \log\|V\|)$ |

where |E|=the number of edges in the dataset , |V| = the number of vertices in the dataset.

As mentioned above, the worst-case time complexity was the same. However, the execution time will vary depending on the constraints or dataset size. For example, the execution time in MST was longer SPOH by a factor of 20. This is due to the difference in the number of edges used in the dataset. Moreover, to obtain a more stable, near-optimal optimal solution on a larger dataset, the number of iterations at each temperature (in the simulated annealing algorithm) must be increased linearly as the size of the dataset increases. As the number of iterations increases, the execution time increases proportionally.

Therefore, in large-scale problem cases, SPOH and RT may still show fast and practical execution times. However, the execution time using MST, which uses a larger dataset to traverse only selected polygons, would not be practical on much larger problem instances. We can therefore conclude that the use of the heuristics within much larger datasets will increase the execution time of the integrated model; but that, since this increase will not be exponential relative to the increase in the number of polygons, it should remain practicable. To be empirically certain of this theoretical conclusion, however, these heuristics should be further tested on much larger problem instances of the harvest-scheduling problem.

Third, we now consider the effect of increased problem size on the execution time of the road network repair heuristic. In this thesis, the repair algorithm searched for nodes on a 50m X 50m scale grid. Therefore, for the case study, 65,150 nodes were potential Steiner nodes. The procedure of repeatably calculating the shortest route from selected Steiner nodes increases linearly relative to problem size. As a result, we conclude that the use of the repair heuristic on a much larger dataset would increase linearly relative to an increase in the size of the forest. Such an increase in computing time may be extensive, but the benefits of using the repair heuristic, as shown in the results, can make this extra computing time a practical and beneficial undertaking for the tactical forest planner.

The second topic of discussion concerns evaluating the possible effects of the road network heuristics on transportation costs. Even though, in this paper, the objective function did not include wood transportation costs, this is a very important cost that should be considered when evaluating the merits of a road network heuristic. To perform this evaluation, we first we calculated the transportation costs of all solutions and then evaluated the implication of these costs on the total costs of both construction and transportation (which are conflicting objectives in a road network's design).

The transportation cost for each repaired solution, from the landing nodes in each harvested polygon to the forest's entry node, was calculated as the distance times \$0.30 per $m^3$ of harvested wood *per* km. Table 8 shows each transportation obtained from each heuristic.

**Table 8:** Discounted transportation costs for the three roading heuristics.

| Transport cost | SPOH(\$) | MST(\$) | RT(\$) |
|---|---|---|---|
| Before Repair | 3,225,220 | 2,068,137 | 3,216,131 |
| After Repair | 2,096,393 | 1,571,784 | 2,283,453 |
| Reduction rate | 35% | 24% | 29% |

Table 8 shows the reduction in transportation costs before and after applying the repair algorithm, and this result is somewhat interesting. From Table 8, we observe that the application of the repair algorithm decreased transportation costs for all three solution, The reduction in transportation costs stems from removing unnecessary detours by the repair algorithm. Also, the reduction rate of transportation costs is larger than the reduction rate in road construction costs (see Table 4). The reduction of road construction costs can be inferred from the mapped solution (Figure 8-13, above). These maps show that reduction in transportation costs stem from from: i) making main roads much straighter; and ii) relocating branch roads. The former occurs mainly on the main road close to the node entry and the latter is observed in the dense set of harvested polygons located in the upper region. Therefore, in the former case, since most vehicles can exploit the improved roads, the transportation costs are reduced significantly.

From these results, we can conclude two things regarding the indirect effect of the algorithms used in this thesis upon transportation costs: first, that, for the algorithms of practical use on larger datasets (i.e., SPOH and RT), there is no meaningful difference in transportation costs; and secondly, that the repair algorithm leads to major reductions in transportation costs, for all three solutions.

The third topic of discussion concerns evaluating the practical benefits that might arise from this research. Here we conclude that the greatest practical benefit of this work is to be found in the usefulness of the TPH algorithm that was used in the repair of the road network after the harvest-scheduling model was run. We present two reasons to support this claim.

The first reason we conclude that that the TPH algorithm has practical benefits is the scale of the reduced costs in the construction of the road network that was achieved by its application. Recall that our results showed that reductions resulting from using the TPH algorithm to the total cost of road construction ranged from 13.1%

to 26.9%. Given the magnitude of road construction costs in forestry, this provides a major economic incentive to use a the TPH algorithm in the practice of producing tactical plans.

A second, less obvious reason for the practical benefits of using the TPH algorithm, is that it is easy to implement in practice. This is because many of the tactical plans generated by practitioners are completed using commercially available software packages (e.g., FSOS, Woodstock-Stanley, Patchworks); and it would be impossible for a practitioner to incorporate TPH into these software packages. But it is not necessary to add the TPH algorithm to these software packages, since it is executed upon the solutions which are output from the software packages. In other words, the TPH road network repair algorithm could be added to the tactical planning procedure as a stand-alone program which takes the output from the tactical planning model as its input. This makes the adaptation of the TPH algorithm into the planning process a practical reality for planners.

# 6. Conclusion

The objective of this thesis was to find more efficient and economically beneficial heuristics for building road networks in tactical forest management planning models. To meet this objective, we; i) compared the effects of three different road-building heuristics (SPOH, MST, and RT), nested within a harvested scheduling model, upon the model's objective function and execution time; and ii) evaluated the effect of a road-network repair heuristic (TPH), executed after the harvest-scheduling model was run, upon the model's objective function. The results showed that: i) of the three nested heuristics tested, RT produced solutions with the highest objective function value and lowest road-network construction cost; and ii) that the repair heuristic reduced road-construction costs between 13.1% and 26.9%. It was also found that RT had an indirect effect on transportation costs, reducing them between 24% and 35%.

From this research we conclude two things. First, that of the three nested heuristics examined in this thesis, RT proved to be most effective in reducing road construction costs within a tactical planning model. A cautionary note is needed on this conclusion; for further research is needed to test its effectiveness on larger data sets. Nonetheless, the preliminary results in this thesis are promising because the execution time of this algorithm is expected to increase linearly with the increase in the problem size. Our second conclusion is that the most valuable result of this research was the illustration of the effectiveness of the repair algorithm, TPH. We draw this conclusion for two reasons: first, TPH led to major reductions in both road-construction and transportation costs; and second, the practical adaptation of this heuristic, by real world planners, is feasible because the algorithm can be run independently of a commercial forest planning software package to make major improvements in a tactical plan's road network.

Future research based on the results of this thesis should be taken in two directions. First, the RT algorithm should be tested on larger datasets in order to reach a firm conclusion on its empirical feasibility. Second, the TPH algorithm should also be tested on larger datasets in order to fully evaluate the effectiveness and empirical feasibility of using this algorithm in forest management planning.

# LITERATURE CITED

Andalaft, N., Andalaft, P., Guignard, M., Magendzo, A., Wainer, A., & Weintraub, A. (2003). A problem of forest harvesting and road building solved through model strengthening and Lagrangean relaxation. Operations Research, 51(4), 613-628.

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

Andersson, D. (2005). Approaches to integrated strategic/tactical forest planning (Vol. 16, No. 16).

Akay, A. E. (2004). A new method of designing forest roads. Turkish Journal of Agriculture and Forestry, 28(4), 273-279.

Akay, A. E., Aruga, K., Bettinger, P., & Sessions, J. (2013). Using optimization techniques in designing forest roads and road networks. Bartın Orman Fakültesi Dergisi, 15(1), 49-62.

Akay, A. E., Erdas, O., Reis, M., & Yuksel, A. (2008). Estimating sediment yield from a forest road network by using a sediment prediction model and GIS techniques. Building and Environment, 43(5), 687-695.

Akay, A. E., & Sessions, J. (2005). Applying the decision support system, TRACER, to forest road design. Western Journal of Applied Forestry, 20(3), 184-191.

Beaudoin, D., Frayret, J. M., & LeBel, L. (2008). Hierarchical forest management with anticipation: an application to tactical–operational planning integration. Canadian journal of forest research, 38(8), 2198-2211.

Bettinger, P., & Chung, W. (2004). The key literature of, and trends in, forest-level management planning in North America, 1950–2001. International Forestry Review, 6(1), 40-50.

Bjørndal, T., Herrero, I., Newman, A., Romero, C., & Weintraub, A. (2012). Operations research in the natural resource industry. *International Transactions in Operational Research*, *19*(1-2), 39-62.

Chung, W., & Sessions, J. (2001). Designing a forest road network using heuristic optimization techniques. Appalachian Hardwood: Managing Change, Corvallis, Oregon.

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

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

Crowe, K. A., & Nelson, J. D. (2005). An evaluation of the simulated annealing algorithm for solving the area-restricted harvest-scheduling model against optimal benchmarks. Canadian Journal of Forest Research, 35(10), 2500-2509.

Dean, D. J. (1997). Finding optimal routes for networks of harvest site access roads using GIS-based techniques. Canadian Journal of Forest Research, 27(1), 11-22.

Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. Int. J. Comput. Eng. Res. Trends, 351(5), 2349-7084.

Devi, S. G., Selvam, K., & Rajagopalan, S. P. (2011). An abstract to calculate big o factors of time and space complexity of machine code.

Dong, L., Bettinger, P., Liu, Z., & Qin, H. (2015). A comparison of a neighbourhood search technique for forest spatial harvest scheduling problems: A case study of the simulated annealing algorithm. Forest Ecology and Management, 356, 124-135.

Goycoolea, M., Murray, A., Vielma, J. P., & Weintraub, A. (2009). Evaluating approaches for solving the area restriction model in harvest scheduling. Forest Science, 55(2), 149-165.

Graham, R. L., & Hell, P. (1985). On the history of the minimum spanning tree problem. Annals of the History of Computing, 7(1), 43-57.

Heinimann, H. R., Stückelberger, J., & Chung, W. (2003). Improving automatic grid cell-based road route location procedures. Proceedings Austro, 5-9.

Kirby, M. W., Hager, W. A., & Wong, P. (1986). Simultaneous planning of wildland management and transportation alternatives. TIMS Studies in the Management Sciences, 21, 371-387.

Kirby, M. W., Wong, P., Hager, W. A., & Huddleston, M. E. (1980). A guide to the integrated resources planning model. Management Sciences Staff, Berkeley, CA, USA, a USDA Forest Service publication, January.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7(1), 48-50.

Liu, K., & Sessions, J. (1993). Preliminary planning of road systems using digital terrain models. Journal of Forest Engineering, 4(2), 27-32.

Lockwood, C., & Moore, T. (1993). Harvest scheduling with spatial constraints: a simulated annealing approach. Canadian journal of forest research, 23(3), 468-478.

Martell, D. L., Gunn, E. A., & Weintraub, A. (1998). Forest management challenges for operational researchers. European journal of operational research, 104(1), 1-17.

McDill, M. E., & Braze, J. (2000). Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions. Forest Science, 46(3), 423-436.

Meignan, D., Frayret, J. M., Pesant, G., & Blouin, M. (2012). A heuristic approach to automated forest road location. Canadian Journal of Forest Research, 42(12), 2130-2141.

Murray, A. T. (1998). Route planning for harvest site access. Canadian Journal of Forest Research, 28(7), 1084-1087.

Murray, A. T. (1999). Spatial restrictions in harvest scheduling. Forest Science, 45(1), 45-52.

Najafi, A., & Richards, E. W. (2013). Designing a forest road network using mixed integer programming. Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering, 34(1), 17-30.

Naderializadeh, N., & Crowe, K. A. (2020). Formulating the integrated forest harvest-scheduling model to reduce the cost of the road-networks. *Operational Research*, *20*(4), 2283-2306.

Naderializadeh, N., Crowe, K. A., & Rouhafza, M. (2020). Solving the integrated forest harvest scheduling model using metaheuristic algorithms. Operational Research, 1-27.

Nelson, J., & Brodie, J. D. (1990). Comparison of a random search algorithm and mixed integer programming for solving area-based forest plans. Canadian journal of forest research, 20(7), 934-942.

Reeves, C. R. (Ed.). (1993). Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc..

Rehfeldt, D. (2015). A generic approach to solving the Steiner tree problem and variants (Doctoral dissertation).

Richards, E. W., & Gunn, E. A. (2000). A model and tabu search method to optimize stand harvest and road construction schedules. Forest Science, 46(2), 188-203.

Richards, E. W., & Gunn, E. A. (2003). Tabu search design for difficult forest management optimization problems. Canadian Journal of Forest Research, 33(6), 1126-1133.

Rouhafza, M. (2015). Incorporating operational road planning constraints into an integrated tactical level harvest-scheduling and road-planning model (Doctoral dissertation).

Sadeghi, A., & Fröhlich, H. (2013). Steiner tree methods for optimal sub-network identification: an empirical study. BMC bioinformatics, 14(1), 1-19.

Shan, Y., Bettinger, P., Cieszewski, C. J., & Li, R. T. (2009). Trends in spatial forest planning. Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS), 1(2), 86-112.

Shirasawa, H., & Hasegawa, H. (2014). A comparative study of heuristic algorithms for the multiple target access problem. Journal of forest research, 19(5), 437-449.

Takahashi, H., and Matsuyama, A. 1980. An approximate solution for the Steiner problem in graphs. Math Jap 24:573–577

Tan, J. (1999). Locating forest roads by a spatial and heuristic procedure using microcomputers. Journal of Forest Engineering, 10(2), 91-100.

Tan, J. (2000). Application of dynamic programming to optimum location of a forest road. Journal of Forest Engineering, 11(2), 33-42.

Teasley, R. L. (2002). Managing forest road removal using dynamic programming: a pilot study. American Journal of Undergraduate Research, 1(2), 23-30.

Turner, A. K. (1978). A Decade of Experience in Computer Aided Route selection. *Photogrammetric Engineering and Remote Sensing*, *44*(12), 1561-1576.

Weintraub, A., & Navon, D. (1976). A forest management planning model integrating silvicultural and transportation activities. Management Science, 22(12), 1299-1309.

Zelikovsky, A. Z. (1993). An 11/6-approximation algorithm for the network Steiner problem. Algorithmica, 9(5), 463-470.