Finding Optimal Routes for Harvesting Tree Access

N. Picard
L. Gazull
V. Freycon
CIRAD
Montpellier Cedex 5, France

ABSTRACT

The layout of forest roads to access cut trees is often done manually in tropical forests, yielding suboptimal road networks with respect to the building cost. An alternative consists in using numerical optimization techniques to find a solution to this problem, also known as the multiple target access problem (MTAP). We used six numerical methods, three of which were found in the literature, to solve the MTAP. The six methods were compared on the basis on the building cost of the road network that they create, and on the basis of the computing time. They were used to solve randomly generated MTAP and also to solve a real case-study in an Indonesian rain-forest at Bulungan. The method that yielded the lowest building cost also required the longest computing time. Its computing time is actually so long that this method cannot be used in real situations. The fastest method poorly minimized the building cost. Among the four remaining methods, two methods were faster than the two others (by a factor 1.5 and 2). One of these two faster methods also yielded the lowest building costs among the four remaining methods.

Keywords: *Road, heuristic, optimization, shortest path, network, building cost.*

INTRODUCTION

Planning forest roads to access timber is part of forest planning [20]. At the strategic level [29], it consists in building roads to access each block of a production series; at the tactical level, it consists in building roads to access harvested trees. This paper addresses the automated methods for building an optimal road network to connect targets (trees or stands) to an existing road network. Optimality refers to a building cost, that has to be minimized. The question of road access optimization may be more or less simplified by considering or disregarding spatial and temporal constraints. Spatial constraints mean that the question is treated in the two-dimensional physical space [20]. Temporal constraints mean that the schedule of road construction is also optimized [3, 22]. In this paper, we shall disregard temporal constraints, and only spatial constraints will be faced.

The question of optimal road network with spatial constraints may be partitioned into two categories of problems: the road spacing problem, and the multiple target access problem (MTAP). In the road spacing problem [27], every point is potentially a target to reach. The question is to build a road network that covers the study area as densely as possible, with a stopping rule (typically a total cost that cannot be exceeded, or a balance between costs and expected benefit). It can be refined by considering different categories of roads, with a trade-off between each category [2]. Heuristic methods have been proposed to solve the road spacing problem [27]. The MTAP [6, 19] is dual of the road spacing problem: it consists in building a road network to reach a set of identified targets, without any stopping rule. In this paper, we shall focus on the MTAP.

Hereafter, we consider m point-shaped targets that have to be connected to an existing road network (the source). The source can be point-shaped, or have a linear or spatial extension. It can be composed of several disconnected parts. The MTAP primarily deals with the building cost. So if a route goes from a point A to a point B and then back from B to A, the cost between A and B is counted once. However, the MTAP can be extended to deal with other kinds of costs or criteria to minimize, including transport cost [2], destruction risk [31], negative impact on the environment [14], etc. The building cost is given as a surface (isotropic case) or as layers (anisotropic case). We suppose that it is known, and we do not consider the issue of estimating it from field characteristics such as substratum, hydrology, sideslope, road slope, altitude, topography, etc. [2, 5, 6, 9, 17, 23].

Dean [6] and Murray [19] have reviewed combinatorial optimization problems that are related to the MTAP. In particular the MTAP can be seen as an extension of the Steiner tree problem where the resulting tree can be composed of several disconnected components [19]. A degraded version of the MTAP is obtained when the only possible branching points of the road network are the targets. This degraded version was faced, e.g., by Clark et al. [3] and Gold and André [11]. Heuristic methods for solving the MTAP have been proposed by Anderson and Nelson [1], Dean [6], and Freycon [9]. The branch evaluation method proposed by Dean [6] was proven to

The authors are all researchers in the Forest Department, French Agricultural Research Centre for International Development, Campus de Baillarguet, France.

yield a solution close to the optimal solution. It is based on a complete enumeration of the possible branching patterns of the road network. The number of branching patterns to reach m targets increases quicker than an exponential (Appendix A). The computing time required to obtain the solution follows the same progression. This actually prevents from using the branch evaluation method, even for values of m that are not high. In his paper, Dean actually restricted to $m = 4$. Typical values of m are much larger. An alternative to the branch evaluation method is then required. The other proposed methods [1, 6, 9] require much less computing time than the branch evaluation method, but their solutions are not so close to the optimum. The MTAP may also be solved using integer optimization methods, as shown by Murray [19], but the amount of time required to solve large problems may also be problematic.

The goal of this study was to construct automated methods for solving MTAPs with many targets, and to compare them to three existing methods selected in the literature. This comparison has not been done previously in an extensive manner, thus preventing from choosing an efficient algorithm. Moreover the existing methods have not been assessed for large problems (when m is great). We deliberately focused on heuristic methods, leaving aside integer optimization methods. Three methods were developed on top of the three existing methods. All six methods were tested by comparing the road networks they produced in randomly generated problems. Finally, the automated methods were used to give a theoretical solution for a real case study, that consists of trees to harvest in a tropical rain-forest at Bulungan in Kalimantan, Indonesia. The Bulungan case study is a large MTAP since there are $m = 60$ trees to harvest in a 133 ha area.

METHODS FOR SOLVING THE MTAP

Continuous Space, Raster Or Graph Theory?

The first step is to clarify the mathematical space that is chosen to pose and solve the MTAP. At least three possibilities exist (Figure 1): space is continuous; space is discrete and made of pixels (raster approach); space is discrete and made of nodes of a network (graph approach). The first approach is convenient to solve embedded problems that contribute to solve the MTAP (such as the Launhard-Weber problem, or the Voronoï diagram, see later). In the second approach, that was used e.g. by Dean [6] or Freycon [9], a cost is associated to each pixel, and the cost of a road segment is the sum of the costs associated to the pixels that form the segment. In the last approach, that is the most frequently used [1, 11, 15, 19, 27], the nodes of the graph are located on a square grid and spatially neighbor nodes are connected by arcs. Costs are associated to arcs. The continuous space approach leads to difficult analytical treatment and inference of input data. It would require numerical optimization that would in turn require some discretization of space.

Thus we hereafter focus on the discrete space approach. Diagonal displacements are allowed, which means that each pixel (raster approach) or each node (graph approach) has eight neighbors (Moore neighborhood). Consider then in Figure 1b the path from pixel A to pixel C through pixel B, and the path from A to C through D. Let c_I be the cost associated to pixel I, and let $\mu_I = c_I / u$, where u is the pixel size, be the linear cost rate at I. In the raster approach, the cost associated to path ADC is $c_A + c_D + c_C$ and the cost associated to ABC is $c_A + c_B + c_C$. Path ADC is preferred to ABC if $c_D < c_B$, that is equivalent with

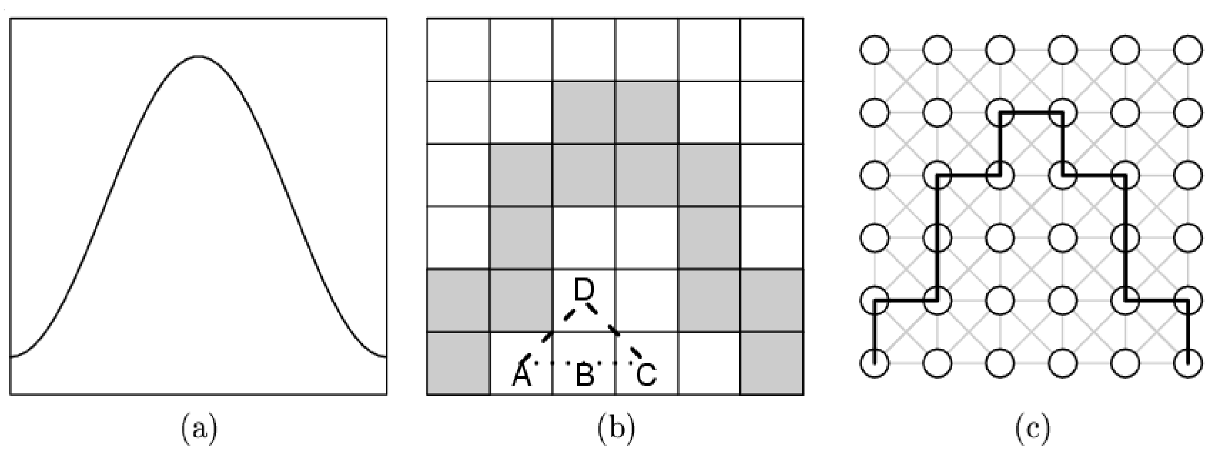

Figure 1. Description of the physical space for the MTAP: (a) continuous space, (b) space made of pixels (raster approach), (c) space made of nodes of a graph (graph theory approach). The line in (a), the grey pixels in (b) and the black arcs in (c) represent the same road segment.

$\mu_D < \mu_B$. However the length of path *ADC* is $2\sqrt{2}u$ and the length of *ABC* is $\sqrt{2}u$, so that the linear cost of *ADC* is less than that of *ABC* if $\sqrt{2}\mu_D < \mu_B$. As a consequence,if $\sqrt{1/2} < c_D / c_B < 1$, path *ADC* will be selected although it has a higher linear cost. The raster approach is thus biased for diagonal displacements [9].

This bias can be circumvented by prohibiting diagonal displacements (using a von Neumann neighborhood around each pixel), or by using different layers or pixels for each direction of displacement (anisotropic costs). It seemed easier, then, to use the graph approach that naturally allows anisotropic costs. Moreover the graph theory provides a theoretical background for it. Hereafter, we shall thus focus on the graph approach. Nevertheless, shifting from the graph approach to the raster approach remains quite natural [19]. The algorithms of the graph approach may be transferred to the raster approach; e.g. to Dijkstra's algorithm (see next section) in graph theory corresponds the cost spreading method in GIS.

Preliminary Algorithms

Before presenting the methods for solving the MTAP, some intermediary methods are presented. To compute the shortest (here "shortest" is taken as a synonym of "that minimizes the cost") path from a node to another node, Dijkstra's algorithm [7, 18] is used. Dijkstra's algorithm actually gives the shortest paths from a node to each of the other nodes of the graph.

To compute the shortest paths from a set of nodes (either connected or not) to each of the other nodes, a modified version of Dijkstra's algorithm is used. Dijkstra's algorithm sequentially updates a function $D(i)$, where i is a node, that gives the best current estimate of the cumulative cost to go from the origin node to i . Initially, $D(i_0) = 0$ where i_0 is the origin node, and $D(i) = 4$ for the other nodes i ...i_0. The modified version simply modifies the initialization step: $D(i) = 0$ if i is in the set of nodes, and $D(i) = 4$ otherwise.

The shortest path between two sets of nodes, denoted E_1 and E_2, is defined as the shortest possible path between a node of E_1 and a node of E_2. To compute it, one simply has to use the modified version of Dijkstra's algorithm using E_1 as the origin, and then to find the shortest among the paths from each node in E_2 to E_1.

Finally, to find the node that is the closest from n sets of nodes (n $3), we used the method proposed by Dean [6]. This method approximates the optimal solution. If the n sets of nodes are reduced to singleton sets, and if the cost to reach a node is proportional to the distance to it, this problem is known as the Launhard-Weber problem [28]. Its analytical solution in continuous space is known for $n = 3$ [12, 13]. Dean's method consists in applying n times the modified version of Dijkstra's algorithm, taking each set of nodes as an origin. Let $D_1 \ldots D_n$ be the resulting cost functions from each of the n sets of nodes to the other nodes. The node that is the closest from the n sets of nodes is taken as the node where $\Sigma_{k=1}^{n} D_k$ reaches its minimum value.

MTAP Solution Methods

Six methods for solving the MTAP are presented. All of them approximate the optimal solution. Three methods were already presented in the literature: the independent paths with reduction method [6], the branch evaluation method [6], and the source-to-closest-target method [1, 9, 10]. However they were defined in the raster approach, and we had to transfer them to the graph approach. Two other methods were presented in the literature, but we disregard them in this study: Dean's fully independent path method [6] was disregarded because it is obviously less efficient than the independent paths with reduction method; Dean's optimal solution was also disregarded because it requires tremendous (practically intractable) computing time even for small sized MTAP.

All six methods were implemented in C and Perl languages, and interfaced with R software [21]. The C language was used to perform all computations on graphs. The Perl language was used to derive all network branching patterns for the branch evaluation method.

We shall illustrate the six methods with a hypothetical example, that is shown in Figure 2a. It consists of a grid with 5 × 5 nodes with a source made of two neighbor nodes (existing road access) and three targets (trees to harvest). Space is uniform, so that the cost associated to horizontal and vertical arcs is one, and the cost associated to diagonal arcs is $\sqrt{2}$.

Independent Paths With Reduction Method [6]

This method consists in searching the shortest path from the source to each of the targets independently from each other, and then to remove the redundant segments. It thus requires only one call of the modified Dijkstra's algorithm. Figure 2b shows the resulting network for the hypothetical example. The total cost is $2 + 6\sqrt{2}$. 10.5. A shortcoming of this method is that it builds parallel paths, as appears with the hypothetical example (Figure 2b).

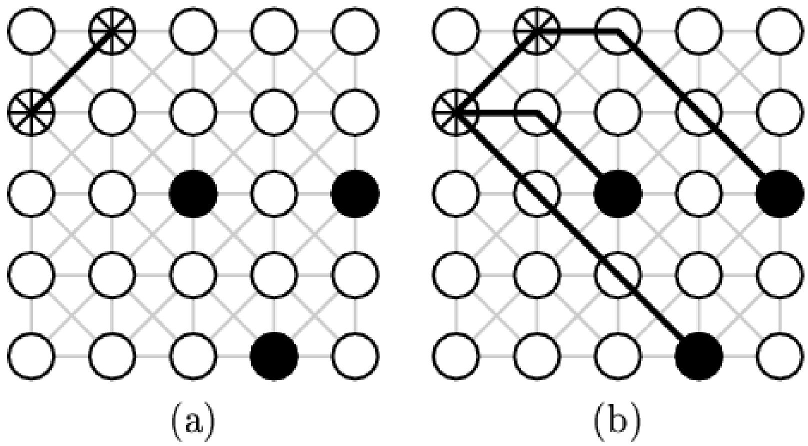

Figure 2. The independent paths with reduction method for solving the hypothetical example that in shown in (a): black nodes are the targets to reach; nodes with a star are the source. Black arcs in (b) are the solution network.

Source-to-Closest-Target Method [1, 9, 10]

This method is an iterative method: initially the network is made of the sources only. At each step, the target that is the closest to the current network is connected to it; the shortest path from this nearest target to the current network is merged with the network, and the shortest paths from the current network to the remaining nodes is updated [24]. It thus requires *m* calls of the modified Dijkstra's algorithm, where *m* is the number of targets. Figure 3 shows the current network at each step for the hypothetical example. The total cost of the final network is $5 + \sqrt{2}$. 6.4. A shortcoming of this method is that it build suboptimal branching points: the length of the sub-network that connects the three targets is four in Figure 3c, whereas it could be $1 + 2\sqrt{2}$. 3.8 if the branching point was located at line 2 column 4.

Branch Evaluation Method [6]

This method consists in considering all possible branching patterns to connect the targets and the source, and to build for each branching pattern the corresponding lowest cost network. The solution is the lowest cost among all the networks thus obtained. For instance there are eight branching patterns to connect three targets. They are indicated in Figure 4, together with the corresponding lowest cost networks for the hypothetical example. In this

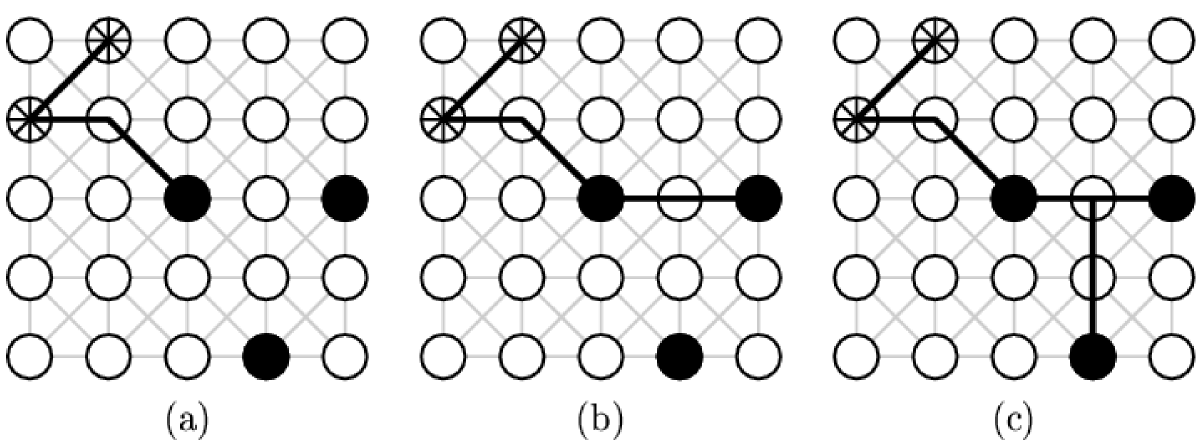

Figure 3. The source-to-closest-target method for solving the hypothetical example. At each step (a), (b), (c), the target that is the closest from the current network is connected to it. Black nodes are the targets; nodes with a star are the source. Black arcs are the current network.

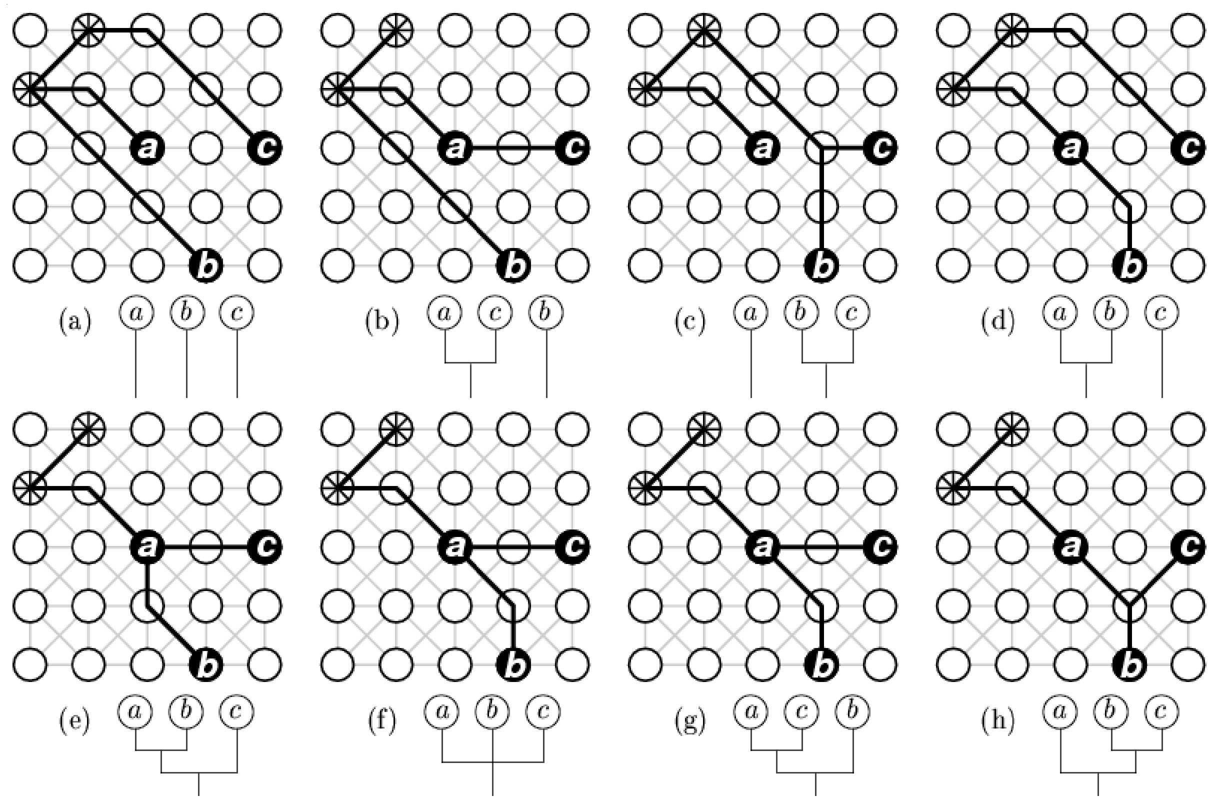

Figure 4. The branch evaluation method for solving the hypothetical example. Each subplot (a)-(h) shows the cheapest network that corresponds to each of the eight possible branching patterns. The diagram below each subplot is the corresponding branching pattern. The best among the eight networks is network (h). Black nodes are the targets; nodes with a star are the source.

example, the eighth pattern (Figure 4h) appeared to yield the lowest cost network among the eight possibilities. Its cost is $2 + 3\sqrt{2}$. 6.2.

Some clarification has to be brought on how the lowest cost network is obtained for a given branching pattern. First, one has to distinguish inner branching points from outer branching points. Inner branching points are not connected to the source, such as the branching point connected to a and b in the diagram of Figure 4e, while outer branching points are connected to the source, such as the lowest branching point in the diagram of Figure 4e. The location of an inner branching point is computed as the node that is the closest from its ascending branches only. The location of an outer branching point is computed as the node that is the closest from both its ascending branches and the source.

Let P_m be the number of branching patterns for the MTAP with m targets (Appendix A). Let B_{km} be the number of branching points in the kth branching pattern ($k = 1 \ldots P_m$), and let $s_{lkm} \geq 2$ be the number of sets of nodes that are connected to the lth branching point ($l = 1 \ldots B_{km}$).

Then the branch evaluation method requires

$$C_m = \Sigma_{k=1}^{P_m} \Sigma_{l=1}^{B_{km}} \psi(s_{lkm})$$

calls to the modified Dijkstra's algorithm, where ø(2) = 1 and ø(s) = s for $s \geq 3$. The numbers B_{km} and s_{1km} can be easily computed from the listing of all branching patterns (Appendix A shows how to obtain this listing). The number C_m increases very quickly: C_6 = 41298, C_7 = 681 066, C_8 = 13 021 576, etc. Its progression is quicker than an exponential relationship.

Hierarchical Methods

Two methods were inspired by the hierarchical methods in cluster analysis. Given n sets of nodes, we define the lowest cost network that connects them as the set of paths that connect the closest node from the n sets of nodes to each of them. The hierarchical method then proceeds in an iterative way: initially there are $m + 1$ unconnected sets of nodes that are the m targets plus the source. All C_{m+1}^{n} networks that connect n of them are computed, and the lowest cost one is selected. As n sets of nodes are connected, $m - n + 1$ unconnected sets of nodes are left.

Then again all C^{n}_{m-n+1} networks that connect *n* of them are computed, and the lowest cost one is selected, etc. The procedure is repeated until the number of unconnected sets of nodes is less than *n*. The remaining sets of nodes are then connected in a single network.

We implemented the hierarchical method for $n = 2$ and $n = 3$. To solve a MTAP with *m* targets, the hierarchical method with $n = 2$ requires $2m + 1$ calls of the modified Dijkstra's algorithm, whereas the hierarchical methods with $n = 3$ requires $(3m + 2)$ \2 calls of the modified Dijkstra's algorithm (where backslash indicates integer division).

Figure 5 and 6 show each step of the hierarchical method with $n = 2$ and 3 to solve the hypothetical example. The total cost of the final network is 6.4 for $n = 2$ and 6.2 for $n = 3$.

Approximation By A Minimum Spanning Tree

The last method consists in transforming the MTAP in a minimum spanning tree problem, by defining an adequate intermediary graph. This intermediary graph is defined as follows. Consider the $m + 1$ sets of nodes, called here "germs", that are the *m* targets plus the source. The nodes of the graph are partitioned into $m + 1$ cells, such that all the nodes in the *k*th cell are closer to the *k*th germ than to any other germ. This partition is the equivalent of the Voronoï diagram in continuous space. Two germs are considered neighbors if their cells share a common boundary. Then "triangles" between any three neighbor germs are built. These are not actual triangles, since the shortest path between two neighbor germs is not a straight line. This tessellation is the equivalent of the Delaunay tessellation in continuous space. Nodes that are the closest to the three vertices of each "triangle" are then introduced. We call them the "centers" of the "triangles", and let *t* be their number.

The intermediary graph then has $m + t + 1$ nodes that are the $m + 1$ germs plus the *t* centers of triangles. Its arcs connect neighbor germs to each other, and connect the centers of triangles to the germs that form the triangles. The costs associated to the arcs are the costs of the corresponding shortest paths. The minimum spanning tree of the intermediary graph is then computed using Kruskal's algorithm [16, 18]. Finally, the branches of the minimal spanning tree that end on a center of triangles are recursively pruned. The resulting tree is a solution of the MTAP. To build the intermediary graph, $m + 1$ calls to the modified Dijkstra's algorithm are required.

Figure 7 shows the tessellation of space around the germs, the intermediary graph, and the solution of the hypothetical example with this method. The total cost of the final network is $2 + 3\sqrt{2}$. 6.2.

MONTE CARLO EVALUATION OF MTAP SOLUTION METHODS

Experimental Plan

We used the same Monte Carlo experimental plan as Dean [6] to compare the six methods for solving the MTAP. Random MTAP were generated, and the six methods were used to solve them. The total cost of the solution computed by each method was recorded, along with the computing time needed to obtain it. Computations were performed by a laptop with a 466 MHz processor. The costs and time estimates were then analyzed statistically.

Four factors were included in the experimental plan: the size of the study area, that is the number *N* of nodes; the number *S* of existing access roads that is the number of sources; the number *m* of trees to harvest, that is the number of targets; and the method for solving the MTAP. The first factor had four modalities: the study area was a grid with 15×15 ($N = 225$), 20×20 ($N = 400$), 25×25 ($N = 625$) or 30×30 ($N = 900$) nodes. The number of sources has little influence on the computing time [6], but it has a great influence on the cost of the road network. Moreover, *S* source nodes distributed at random do not have the same effect as *S* connected nodes. The former will favor one-to-one connections between targets and sources, and will thus limit the complexity of the branching pattern. Hence, contrary to Dean [6], we used low values of the number of sources: $S = 1$, 2 or 3. The number of targets was $m = 3$, 4 or 5. It is cumbersome to use values of *m* greater than five with the branch evaluation method. Finally there are six methods for solving the MTAP. Thus, the experimental plan comprises 4×3×3×6 = 216 combinations of factors, or treatments.

For each treatment, one hundred MTAP were randomly generated. The locations of sources and targets were drawn at random without replacement. Horizontal and vertical displacement costs were uniformly distributed on [0,1], whereas diagonal displacement costs were uniformly distributed on $[0,\sqrt{2}]$. The cost of the solution of the MTAP and the computing time were then analyzed using a completely balanced four-way analysis of variance.

Results

Table 1 shows the results of the analysis of variance of the computing time, whereas Table 2 shows the results of the analysis of variance of the road building costs. Computing time depends on the number of nodes, on the number of targets and on the method, but not on the number of sources. Except for the method based on the minimum spanning tree, the computing time can be broken down into the number of times that Dijkstra's algorithm is called, times the computing time required for performing

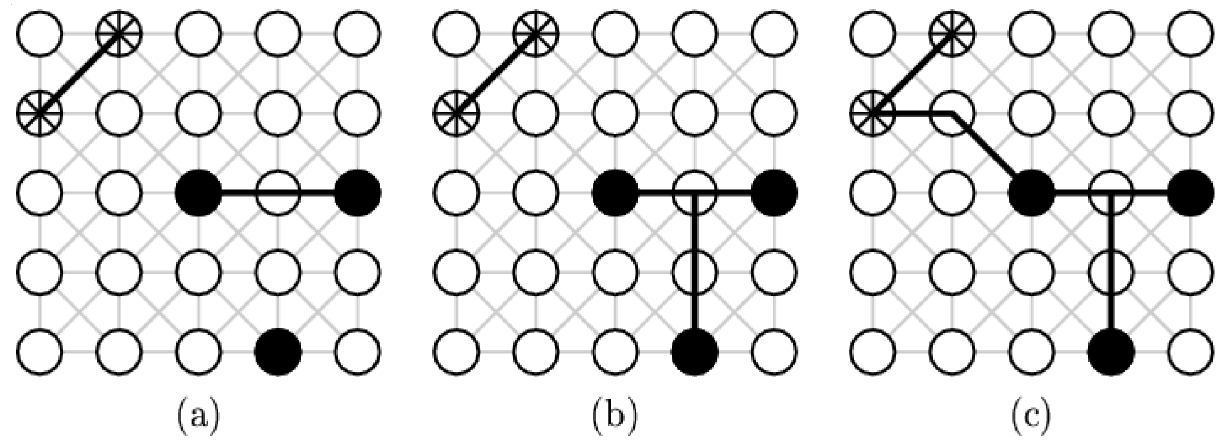

Figure 5. Hierarchical method with $n = 2$ for solving the hypothetical example. At each step (a), (b), (c), the two sets of nodes (either the targets or the source) that are the closest are connected. Black nodes are the targets; nodes with a star are the source. Black arcs are the current network.

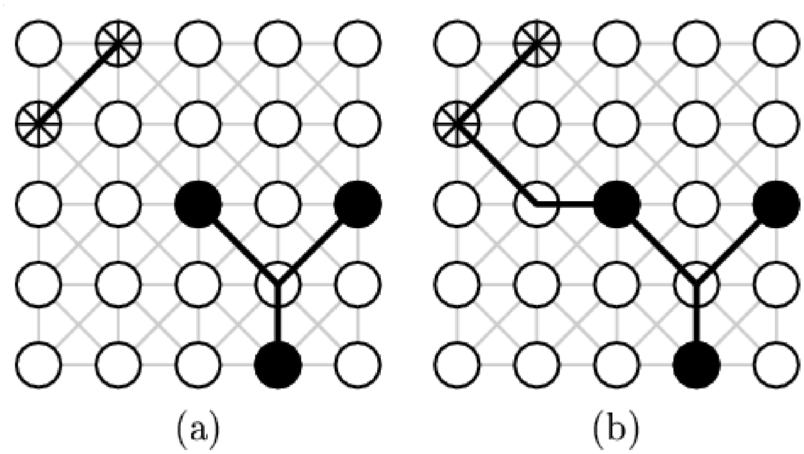

Figure 6. Hierarchical method with $n = 3$ for solving the hypothetical example. At each step (a), (b), the three sets of nodes (either the targets or the source) that are the closest are connected. Black nodes are the targets; nodes with a star are the source. Black arcs are the current network.

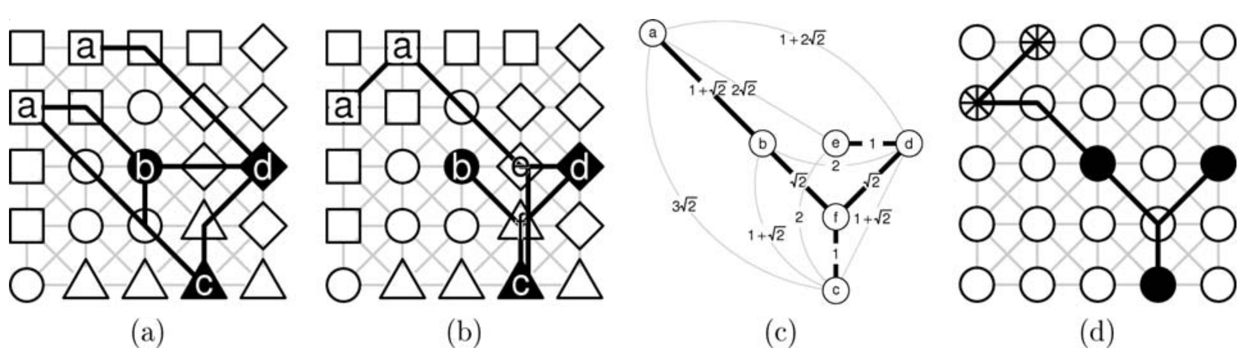

Figure 7. Method based on a minimum spanning tree for solving the hypothetical example. (a) Partition of the nodes around the germs *a-d* and associated tessellation; nodes with the same symbol (circle, square, lozenge, triangle) belong to the same cell; black arcs connect neighbor germs; black nodes are the targets; nodes labeled *a* are the source. (b) Additional "centers" of the "triangles" *e-f*: *e* is the center of the "triangle" (*a*, *c*, *d*), *f* is the center of the "triangle" (*b*, *c*, *d*); black arcs are the additional connections. (c) Intermediary graph that result from the connections in (a) and (b); black arcs are the minimum spanning tree of the graph. (d) Solution of the MTAP that follows from the minimum spanning tree in (c); the arc between *d* and *e* has been pruned since it ends on the center of a "triangle"; black nodes are the targets; nodes with a star are the source.

Table 1. Analysis of variance of the computing time.

(a) Analysis of variance results

Factor	df	Mean square	*F*-value	*p*-value
No. of nodes	3	771 642	296.53	< 0.0001
No. of sources	2	13	0.00	0.9951
No. of targets	2	2 512 433	965.48	< 0.0001
Method	5	3 284 420	1262.14	< 0.0001

(b) Multiple range test results (at the á = 0.05 level)

	Mean time (s)	*n*	Tukey test	Duncan test	Scheffé test
No. of nodes					
15 × 15	2.00	5400	A	A	A
20 × 20	5.84	5400	B	B	B
25 × 25	13.92	5400	C	C	C
30 × 30	29.00	5400	D	D	D
No. of targets					
3	0.59	7200	A	A	A
4	3.28	7200	B	B	B
5	34.21	7200	C	C	C
Method					
independent paths with reduction	0.07	3600	A	A	A
source-to-closest-target	0.27	3600	A	A	A
minimum spanning tree	0.41	3600	A	A	A
hierarchical $n = 3$	0.45	3600	A	A	A
hierarchical $n = 2$	0.61	3600	A	A	A
branch evaluation	74.35	3600	B	B	B

Dijkstra's algorithm. The latter approximately depends on the number of nodes only, whereas the former depends on the number of targets m and on the method, as clarified previously. Mean computing time was significantly different for all numbers of nodes, and for all numbers of targets. However, two groups of methods were distinguished as regards computing time: the branch evaluation method that requires a long computing time on one hand, and the other methods on the other hand. The long computing time required for the branch evaluation method directly results from the huge number of times that it calls Dijkstra's algorithm.

The building cost depends on all four factors (Table 2). Building cost was significantly different for all numbers of nodes, for all numbers of targets, and for all numbers of sources. However, depending on the multiple range test, three (Scheffé test) or four (Tukey and Duncan tests) groups of methods were distinguished as regards the building cost. The branch evaluation method provides the lowest cost road networks. Then the hierarchical method with $n = 2$ and the source-to-closest-target method provide the lowest cost alternative. The hierarchical method with $n = 3$ and the method based on the minimum spanning tree are in third position. Finally the independent paths with reduction method provides the most expensive networks.

Table 2. Analysis of variance of the network building cost.

(a) Analysis of variance results

Factor	df	Mean square	*F*-value	*p*-value
No. of nodes	3	1 110 837	5054.84	< 0.0001
No. of sources	2	302 541	1376.71	< 0.0001
No. of targets	2	828 714	3771.05	< 0.0001
Method	5	51 911	236.22	< 0.0001

(b) Multiple range test results (at the á = 0.05 level)

	Mean cost	*n*	Tukey test	Duncan test	Scheffé test
No. of nodes					
15 × 15	41.53	5400	A	A	A
20 × 20	52.89	5400	B	B	B
25 × 25	63.94	5400	C	C	C
30 × 30	74.88	5400	D	D	D
No. of sources					
1	64.87	7200	A	A	A
2	58.16	7200	B	B	B
3	51.91	7200	C	C	C
No. of targets					
3	47.26	7200	A	A	A
4	58.98	7200	B	B	B
5	68.69	7200	C	C	C
Method					
branch evaluation	55.33	3600	A	A	A
hierarchical $n = 2$	56.45	3600	B	B	A
source-to-closest-target	56.48	3600	B	B	A
hierarchical $n = 3$	57.66	3600	C	C	B
minimum spanning tree	58.16	3600	C	C	B
independent path with reduction	65.79	3600	D	D	C

APPLICATION TO TREE HARVESTING AT BULUNGAN

Study Area

The automated methods for solving the MTAP were used to design a road network to access harvested trees in a tropical rain-forest at Bulungan (2°52'–3°14'N, 116°–116°40'E) in Kalimantan (Borneo island), Indonesia. The climate is equatorial with an annual rainfall of 4000 mm. The topography is deeply eroded with a dense network of steep ridges and drainage gullies. Elevations at the study area range from 100 to 300 m above sea level [25]. The study site is the block 27 of a 50,000 ha forest concession managed by a state-owned timber company. Its area is 133 ha.

The MTAP and Its Hand-Made Solution

The study site was embedded in a rectangle that was covered with a square grid of 117× 128 nodes. Each node represents an area of 144 m^2. As the study area is characterized by steep terrain and dense river network, the main construction activities are earthworks (excavation, surfacing, and slope stabilization) and stream crossings (building culverts and bridges). The road building cost at a given place was estimated as the maximum of the earthwork cost and of the stream crossing cost. The former depends mainly on terrain gradient and the latter on stream width. To estimate building costs, a GIS was used in a first stage to create raster grid cells from different digital topographic data. A digital elevation model was generated from contour layers to calculate slopes. Table 3 gives the unit costs assigned to the different types of terrain condition encountered in the study area. These costs are expressed in arbitrary units. They come mostly from the FAO literature [8, 30] and from our field experience. The estimate of the cost grid is shown in Figure 8. In a second stage, costs estimated for each pixel were converted to costs associated to arcs between two nodes. Let c_{ij} be the cost associated to the pixel at line *i* and column *j* of the grid of pixels, and let $\tilde{a}_{ij\,6\,kl}$ be the cost associated to the arc between the node at line *i* column *j* and the node at line *k* column *l*. We used the following relationships: $\tilde{a}_{ij6i+1,j} = \tilde{a}_{ij6i,j+1,} = c_{ij}$ and $\tilde{a}_{ij6i+1,j+1} = \tilde{a}_{i+1,j6i,j+1,} = v2c_{ij}$, for all *i* and *j*. These relationships define the cost for all connections between nodes. They suppose that costs are isotropic. The $\sqrt{2}$ factor was introduced not to bias for diagonal displacement.

Table 3. Road building cost assigned to a cell, depending on the terrain condition. Cost is in arbitrary unit.

Terrain Condition	Slope Range	Cost
Easy	0-30%	10
Medium	30-50%	15
Difficult	50-70%	30
Very difficult	70-100%	45
Drain (primary)	any slope	200
Drain (secondary)	any slope	20
Existing road	any slope	0

Sixty trees have to be harvested on the study area. The number of targets of the MTAP thus is $m = 60$. There are two existing road accesses that join at the North of the study area. They are represented as white pixels in Figure 8. One road goes around the study area from North to South on its western limit. The other goes around the study area on its eastern limit. Together, the existing roads accesses cover 321 nodes. They form the source of the MTAP. A hand-made road network was designed by forest engineers to access the target trees. It is shown in Figure 9a and its cost is given in Table 4. The hand-made solution located the roads on crests and crossed primary streams only once (black pixels in Figure 8). The hand-made solution cannot be directly compared to the numerical solutions because we have no guarantee that the costs given in Table 3 match the engineers' objectives. The comparison between the hand-made solution and the numerical solutions should rather be seen as a way to assess the realism of the cost grid.

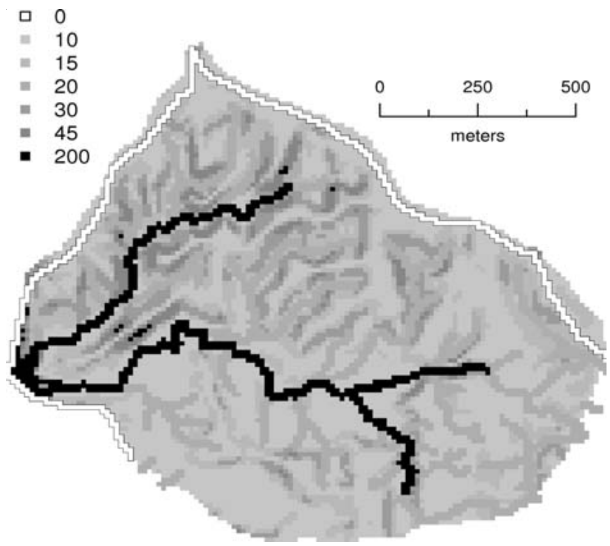

Figure 8. Cost grid at Bulungan. Cost units are arbitrary. White pixels show the existing road accesses. Black pixels show the primary streams.

Table 4. Cost of the road network and computing time required to obtain it by numerical methods.Cost is in arbitrary unit.

Method	Cost	Computing Time
Hand-made	11 844	–
Independent paths	11 549	18 s
Freycon no2	5 812	18 min 28 s
Hierarchical ($n = 2$)	5 794	36 min 37 s
Hierarchical ($n = 3$)	6 499	34 min 38 s
MST	6 376	18 min 37 s

Numerical Solutions

Due to the big size of the MTAP, the branch evaluation method is not applicable. We thus restricted to the five other automated methods for solving the MTAP. The solutions computed by the five methods are shown in Figure 9b–f. The costs and computing times are given in Table 4. The independent paths with reduction method is very fast but is as expensive as the hand-made solution. The four other numerical solutions are far less expensive. The source-to-closest-target method and the hierarchical method with $n = 2$ have a similar cost (Table 4). Their solutions are actually very similar and differ only in details (Figures 9c and d). The hierarchical method with $n = 3$ and the method based on the minimum spanning tree have a similar cost, that is higher than that of the previous methods. The source-to-closest-target method and the method based on the minimum spanning tree have a similar computing time. The two hierarchical methods have a similar computing time that is about twice longer than that of the previous methods.

Thus, for the MTAP at Bulungan, the best solution is provided by the hierarchical method with $n = 2$ when we do not take account of the computing time. If computing time is considered, the best trade-off is provided by the source-to-closest-target method.

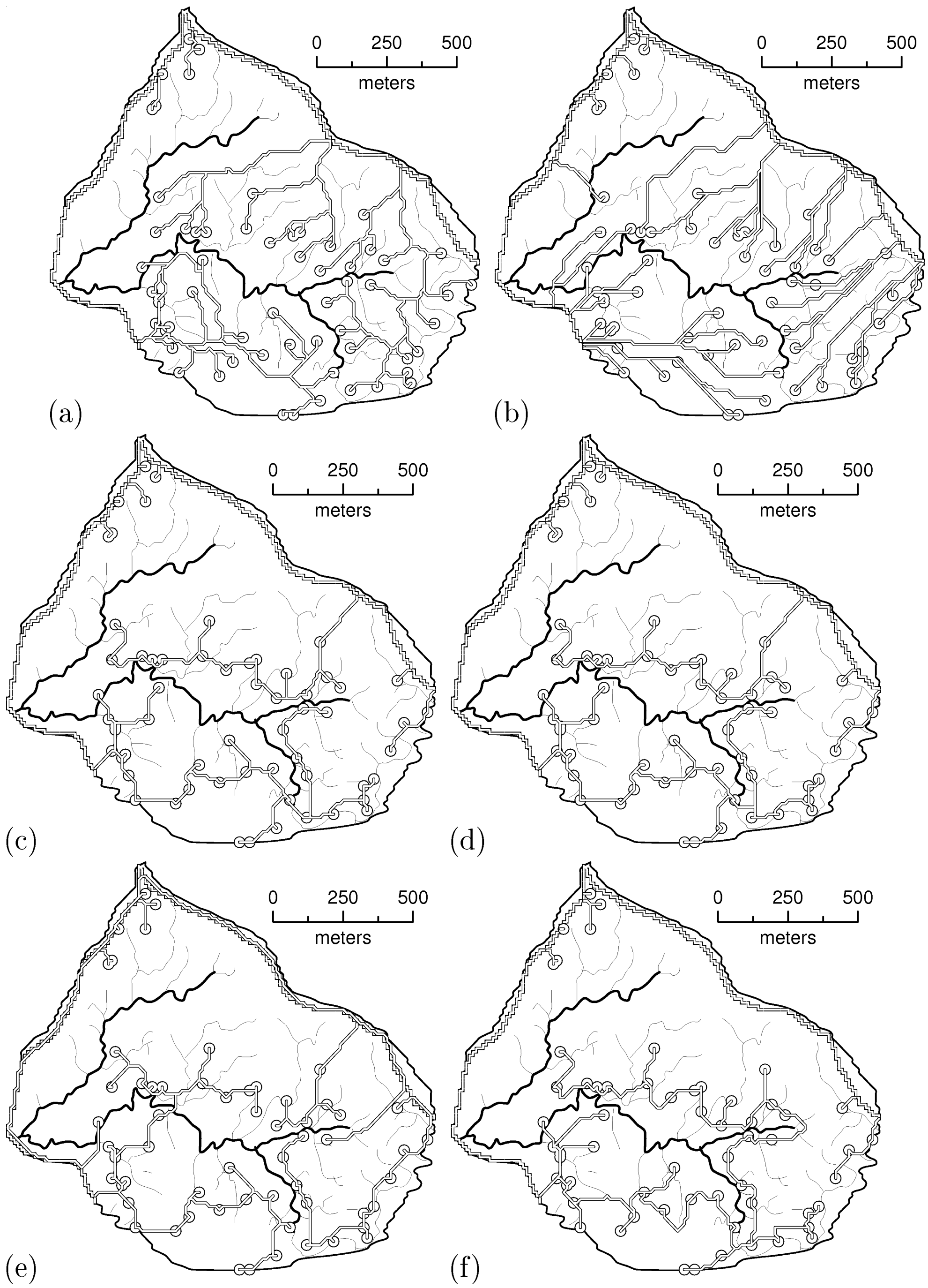

Figure 9. Road network at Bulungan to access the 60 target trees: (a) hand-made layout; (b) solution with the independent paths with reduction method; (c) solution with the source-to-closest-target method; (d) solution with hierarchical ($n = 2$) method; (e) solution with hierarchical ($n = 3$) method; (f) solution with the method based on minimum spanning tree. White dots are target trees, primary streams are indicated by black lines and secondary streams are indicated by thin black lines.

DISCUSSION

Building Cost

The lowest building cost is given by the branch evaluation method, then by the hierarchical method with $n = 2$ or by the source-to-closest-target method, then by the hierarchical method with $n = 3$ or by the method based on the minimum spanning tree, and finally by the independent paths with reduction method. This order was obtained both with the Monte Carlo evaluation of MTAP solution methods and with the practical case-study at Bulungan, with the reservation that the branch evaluation method could not be used at Bulungan due to the number of targets.

The comparison between the hand-made solution at Bulungan and numerical solutions must be cautious, since the forest engineers may have used to derive their solution some professional experience and intuition that is not taken into account in the cost grid. Designing a realistic and feasible road network in an automated way can actually be split into two nested problems: 1. What cost grid should be used? 2. Given the cost grid, which algorithm should be used? The former question is a challenging issue by itself, and was not the focus of this paper. Nevertheless, the comparison between the hand-made solution and numerical solutions can be used to assess the relevance of the cost grid. The former question may also be the basis for defining other criteria of assessment of the numerical methods, namely: Is the numerical method sensitive to small changes of the unit costs? Sensitivity analyses would thus be required to select robust methods that can deal with uncertainty in cost estimates [1]. As concerns the second question, the idea is not to replace hand-made solutions by numerical solutions, but rather to use numerical methods to identify better strategies for hand-made layout, in the same philosophy as computer assisted design. The human component will still be required to rebut, accept or modify numerical solutions in an interactive way. At Bulungan for instance, forest engineers have focused on crests and streams, but apparently have attached less importance to the length of the road network. Comparing Figure 9a to Figures 9c–f suggests that road segments could be saved.

The results of this study were based on six heuristics. Other methods could be imagined to get lower costs. For instance the source-to-closest-target method is known to build suboptimal branching points. This method could thus be improved by identifying a posteriori all branching points and re-computing their position as indicated before (see "Preliminary algorithms"). The method based on the minimum spanning tree could also be improved by increasing the number of nodes of the intermediary graph. This number could be increased until all the nodes of the MTAP are included in the intermediary graph. However computations in this case would be long.

Computing Time

The computing time can be a limiting factor to the size of the MTAP that can be solved numerically. The Monte Carlo evaluation of MTAP solution methods classified the methods into two groups: the branch evaluation method in one hand (slow method), the five other methods on the other hand (fast methods). However the analysis of variance was limited to small-sized MTAP. As the size of the MTAP increases, the branch evaluation method has to be disregarded; differences in execution time between the five remaining methods then appear. Most of the computing time is spent running Dijkstra's algorithm. The methods can then be ranked according to the number of times they call Dijkstra's algorithm.

The independent paths with reduction method is the fastest, as it calls Dijkstra's algorithm once. Let T_N be the execution time of the independent paths with reduction method for a MTAP with nodes. It is almost independent of the number m of targets. The ranking of the methods by execution speed then is: the source-to-closest-target method (execution time . mT_N); the minimum spanning tree method (execution time . $(m+1)T_N$); the hierarchical method with $n = 3$ (execution time . $[(3m+2) \backslash 2]T_N$); and lastly the hierarchical method with $n = 2$ (execution time . $(2m+1)T_N$). As an example, Figure 10 shows the execution time for the five methods for the MTAP at Bulungan, when m varies from 3 to 60 targets. The approximate estimations of the execution times are reliable, except for the hierarchical method with $n = 3$ when m is high. In that case, the computing time that is spent outside Dijkstra's algorithm is no longer negligible.

Nevertheless, we did not optimize for speed the algorithms that we implemented. Thus, the two hierarchical methods and the minimum spanning tree method could be made to process faster by storing in a data structure intermediary computations. These computations are independent of Dijkstra's algorithm, so the gain would be small unless the number m of targets is high. More importantly since most of the computing time is spent running Dijkstra's algorithm, we used a basic implementation of Dijkstra's algorithm that requires $O(N^2)$ operations (where N is the number of nodes and O is Landau asymptotic symbol, meaning that the number of operations is less than some constant multiple of N^2). By using a more complex implementation, this could be reduced to $O(N \ln N)$ [4]. Notice also that some of the methods are embedded in the others. Thus, the hierarchical

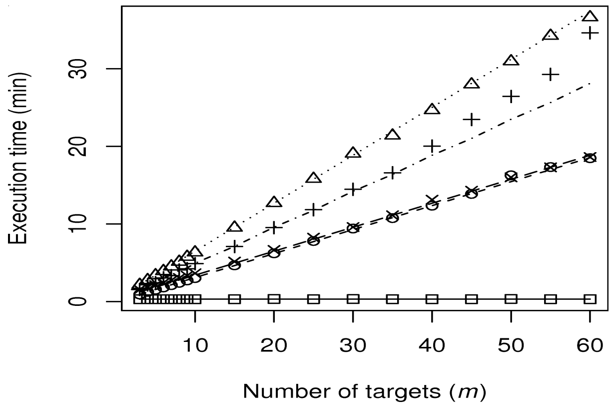

Figure 10. Execution time as a function of the number *m* of targets for the MTAP at Bulungan, for the five MTAP solution methods (the branch evaluation method). Symbols are the actual execution times, lines are a priori approximate estimations. Plain line and squares: independent paths with reduction method; dashed line and circles: source-to-closest-target method; long-dashed line and crosses: minimum spanning tree method; dot-dashed line and pluses: hierarchical method ($n = 3$); dotted line and triangles: hierarchical method ($n = 2$).

methods can give the solution of the source-to-closest-target method or of the minimum spanning tree method with little additional computation. Similarly, the solution of the independent paths with reduction method can be given with little additional computation by any other method.

CONCLUSION

The branch evaluation method cannot be used when the number of targets of the MTAP increases (typically over 6), which prevents from using it in practical situations. The independent paths with reduction method is the fastest of the remaining method, but it behaves very poorly with respect to the building cost. The lowest building cost is provided, if we disregard the branch evaluation method, by the source-to-closest-target method and by the hierarchical method with $n = 2$. Moreover the source-to-closest-target method is about twice as fast as the hierarchical method with $n = 2$. The minimum spanning tree method is about as fast as the source-to-closest-target method, but it yields higher building costs. As a conclusion, the source-to-closest-target method provides the best trade-off between building cost and computing time for large MTAP. Other heuristics and integer optimization methods would have to be investigated to locate even better road design methods.

ACKNOWLEDGEMENTS

We thank Dr. P. Sist for useful information on the Bulungan forest. We are also grateful to two anonymous reviewers for helpful remarks.

AUTHOR CONTACT

Nicolas Picard can be contacted by email at --
nicolas.picard@cirad.fr

REFERENCES

[1] Anderson, A. E. and J. Nelson. Projecting vector-based road networks with a shortest path algorithm. Can. J. For. Res., 34(7):1444–1457, 2004.

[2] Chung, W. and J. Sessions. Designing a forest road network using heuristic optimization techniques. In J. F. McNeel, editor, Appalachian Hardwood: Managing Change, Corvallis, Oregon, 2001. Council on Forest Engineering.

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

[4] Cormen, T. H., C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. MIT Electrical Engineering and Computer Science. MIT Press, Cambridge, MA, 1990.

[5] Coulter, E., W. Chung, A. Akay, and J. Sessions. Forest road layout using a high resolution digital terrain model generated from LIDAR data. In Proceedings of the First International Precision Forestry Symposium, June 2001, Seattle, Seattle, Washington, 2001. University of Washington, College of Forest Resources.

[6] Dean, D. J. Finding optimal routes for networks of harvest site access roads using GIS-based techniques. Can. J. For. Res., 27:11–22, 1997.

[7] Dijkstra, E. W. A note on two problems in connection with graphs. Numer. Math., 1:269–271, 1959.

[8] FAO. Cost control in forest harvesting and road construction. Number 99 in FAO Forestry Paper. FAO, Rome, Italy, 1992.

[9] Freycon, V. GIS, a tool for drawing a network of forest roads: methods and results. Number 13 in Série FORAFRI. CIRAD, Montpellier, France, 1998.

[10] Freycon, V. GIS methodological assessment in designing an optimal forest road network. In Y. Laumonier, B. King, C. Legg, and K. Rennolls, editors, Data Management and Modelling Using Remote Sensing and GIS for Tropical Forest Land Inventory, pages 351–360, Rodeo International Publishers, Jakarta, Indonesia, 1999.

[11] Gold, C. and S. André. A simplified approach to forest decision support systems. In Adams/GIS World, editor, Pathways to Knowledge Integration: Proceedings of GIS '98/RT '98, pages 125–127, GIS World Inc., Fort Collins, Colorado, 1998.

[12] F. E. Greulich. Road network design: optimal economic connection of three horizontal control points on flat, uniform terrain. J. For. Eng., 7(1):73–82, 1995.

[13] Greulich, F. E. The barycentric coordinates solution to the optimal road junction problem. J. For. Eng., 10(1):111–114, 1999.

[14] Heralt, L. Using the ROADENG system to design an optimum forest road variant aimed at the minimization of negative impacts on the natural environment. J. For. Sci., 48(8):361–365, 2002.

[15] Kanzaki, K. A planning of the forest road network by the theory of graphs. J. Japan. Forest. Soc., 48:365–371, 1966.

[16] Kruskal, J. B. On the shortest spanning tree of graph and the traveling salesman problem. P. Am. Math. Soc., 7:48–50, 1956.

[17] Liu, K. and J. Sessions. Preliminary planning of road systems using digital terrain models. J. For. Eng., 4(2):27–32, 1993.

[18] Morris, J. Data structures and algorithms. University of Western Australia, Crawley, Australia, 1998.

[19] Murray, A. T. Route planning for harvest site access. Can. J. For. Res., 28(7):1084–1087, 1998.

[20] Murray, A. T. and S. Snyder. Spatial modeling in forest management and natural resource planning. For. Sci., 46(2):153–156, 2000.

[21] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2003.

[22] Richards, E. W. and E. A. Gunn. A model and tabu search method to optimize stand harvest and road construction schedules. For. Sci., 46(2):188–203, 2000.

[23] Sakai, T. Two algorithms for forest road planning using digital terrain map (DTM). J. Jap. For. Eng. Assoc., 10(3):225–231, 1995.

[24] Sessions, J. Solving for habitat connections as a Steiner network problem. For. Sci., 38(1):203–207, 1992.

[25] Sist, P., D. Sheil, K. Kartawinata, and H. Priyadi. Reduced-impact logging in Indonesian Borneo: some results confirming the need for new silvicultural prescriptions. For. Ecol. Manag., 179:415–427, 2003.

[26] Sloane, N. J. A. The on-line encyclopedia of integer sequences, 2004. AT&T Integer Sequences Research. http://www.research.att.com/~njas/sequences/

[27] Tan, J. Locating forest roads by a spatial and heuristic procedure using microcomputer. J. For. Eng., 10(2):91–100, 1999.

[28] Tellier, L. N. The Weber problem: solution and interpretation. Geogr. Anal., 4(3):215–233, 1972.

[29] Weintraub, A. and A. Cholaky. A hierarchical approach to forest planning. For. Sci., 37(2):439–460, 1991.

[30] Winkler, N. Environmentally sound road construction in mountainous terrain. Applying advanced operating methods and tools. Number 10 in Forest Harvesting Case-Study. FAO, Rome, Italy, 1998.

[31] Yoshimura, T. and K. Kanzaki. Fuzzy expert system laying out forest roads based on the risk assessment. In FAO, editor, Proceedings of the Seminar on Environmentally Sound Forest Roads and Wood Transport, Sinaia, Romania 17-22 June 1996, Rome, Italy, 1998. FAO.

APPENDIX

Number of Branching Patterns

This appendix aims to provide the number P_m of branching patterns for the MTAP with m targets. For instance, the diagrams in Figure 4 show the eight branching patterns for the MTAP with three targets, as defined by Dean [6]. It seems that some branching patterns were forgotten by Dean [6], such as the ones shown in Figure 11 for $m = 3$. One may argue that the branching patterns in Figure11 are degenerate cases of the branching patterns in Figure 4 when the length of some appropriate segments are set to zero, but in that case one should also consider, for instance, that the branching pattern shown in Figure 4f is a degenerate case of that shown in Figure 4e. Hereafter, we consider only the branching patterns that were considered by Dean [6], to keep consistency with the literature.

The computation of P_m is based on a recursion formula. Given a branching pattern with s segments and b branching points, there are $s + b + 1$ ways to connect an additional target: the target can be connected to an existing branching point, which gives b branching patterns with $s + 1$ segments and b branching points; the target can be connected to an existing segment, which gives s branching patterns with $s + 2$ segments and $b + 1$ branching points; or the target can be connected by an independent path, which gives one branching pattern with $s + 1$ segments and b branching points. For $m = 1$ target, there is $P_1 = 1$ branching pattern with one segment and zero branching point. Using this initial condition and the recursion formula, one can readily check that any branching pattern with b branching points and s segments that reaches m targets verifies $s – b = m$. Moreover, s varies between m and $2m - 1$, and b varies between zero and $m - 1$.

Hence the number of branching patterns to reach m targets can be counted by a vector $(a_{0m}, ..., a_{m-1,m})$ such that $P_m = \Sigma_{b=0}^{m-1} a_{bm}$ where a_{bm} is the number of branching patterns with b branching points to reach m targets. The recursion formula states that the a_{bm} branching patterns to reach m targets generate $(b + 1)a_{bm}$ branching patterns with b branching points to reach $m + 1$targets, and $(b + m)a_{bm}$branching patterns with $b + 1$ branching points to reach $m + 1$targets. Initially, $a_{01} = 1$. The sequence then is: $P_1 = 1$, $P_2 = 2$, $P_3 = 8$, $P_4 = 52$, $P_5 = 472$, ..., $P_{14} =$ 363,581,406,419,456,etc. The integer sequence (P_m) is known as the number of series-parallel networks with m labeled edges, and is labeled as sequence A006351 in Sloane [26]. The progression of P_m is quicker than an exponential relationship.

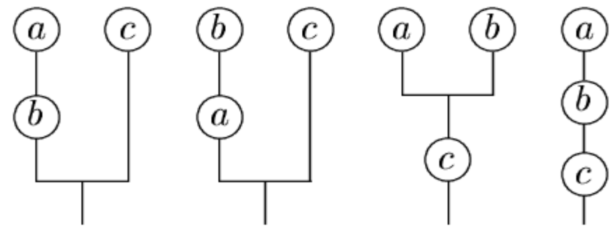

Figure 11. Examples of branching patterns for the MTAP with $m = 3$ targets that were not considered by [6].

Oregon State University

Department of Forest Engineering

Advancing safe, economical, and environmentally sound forest operations through outstanding teaching and innovative leadership in research and outreach education

The Department of Forest Engineering was established in 1913 to educate logging engineers for the developing forest industry. Research and outreach education missions were added 1968 and 1972 respectively. Today 16 faculty members, including engineers, hydrologists, and forest scientists, support undergraduate and graduate education, research, extension, and continuing education programs.

The university is ideally located on the west coast of North America among diverse and productive forests. The forest sector is a major element of the region's economy. Steep terrain commonly requires well engineered roads and harvest systems. Societal concerns over forestry practices have increased regulations and costs, which add challenges in global markets.

We offer a four-year forest engineering undergraduate degree that is accredited by both the Accreditation Board of Engineering Technology and the Society of American Foresters. A similarly accredited five-year program provides Bachelor of Science degrees in Forest Engineering and in Civil Engineering. Graduates have excellent employment opportunities. In 2006 we anticipate enrolling students in a new degree in Forest Operations Management. We supplement classroom learning through a logging training program that operates on our nearby 5000 ha College of Forestry forest and a required 6-month work experience.

Graduate education is offered through research-focused PhD and Master of Science degrees, and a non-thesis Master of Forestry. Program concentrations are available in forest engineering, forest hydrology, forest operations, forest soil science, and harvesting. A joint PhD program in harvesting and silviculture is available.

Faculty research interests include: harvesting process engineering, transportation system design, logistics, production planning to maximize stand and log value, development of landscape planning models, applications of information technologies, forestry workforce issues, understanding and mitigating environment impacts of forestry activities, and basic hydrological sciences. Faculty members have developed several decision-support software programs that are widely used in designing forest operations.

Research results are actively conveyed to forest operations professionals, the public, and policy makers, and contribute to the scientific foundation for forest management and regulation. In addition to many kinds of publications, we offer extension and continuing education programs that support life-long learning. A new International Forest Engineering Institute was established in 2005 and we are preparing to host the 13th International Mountain Logging and Skyline Logging Symposium in April 2007.

The department actively supports the International Union of Forestry Research Organizations (IUFRO), the Council on Forest Engineering, and this journal. Six faculty currently or have previously served as IUFRO officers.

Our faculty approaches the 21st century with excitement about the relevance of forest engineering in balancing society's demands for wood products, clean water, and other resources from sustainable forests around the world.

For further information, please contact Steve Tesch, PhD, Professor and Head of Forest Engineering at:

Forest Engineering Department
Oregon State University
Corvallis, OR 97331-5706
Phone: 541-737-4952
Fax: 541-737-4316
E-mail: Steve.Tesch@oregonstate.edu