

# Steiner tree network scheduling with opportunity cost of time

Fabien Cornillier

*Universidad del Pacifico, Graduate School of Business, Lima, Peru*

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

---

## Abstract

This paper points out the impact of opportunity cost of time (high discount rate or high rate of time preference, time-dependent profits, etc.) in designing real-world Steiner trees like electricity, gas, water, or telecommunications networks. We present the Steiner Tree Scheduling Problem which consists of finding a Steiner tree in an activity-on-arc graph that spans a set of mandatory vertices, and of scheduling each selected activity in a competition for scarce resources so as to optimize a given objective function. Project managers typically make the network design decisions and the activity scheduling decisions separately. The main contribution of this paper is in demonstrating the potential for a more efficient network planning in a context of high opportunity cost of time by making simultaneous design and scheduling decisions. Mixed integer programming formulations are also proposed, and a heuristic procedure is described.

*Keywords:* Steiner tree network design, Project scheduling, Resource-constrained scheduling

---

## 1. Introduction

Whenever significant opportunity cost of time exists (discount rate, rate of time preference, time-dependent profits, etc.) a time-cost tradeoff maybe considered in designing real-world structure like electricity, gas, water or telecommunication networks. For instance, in public procurement, contracts are most often awarded to the lowest qualified bidder complying with a predetermined delivery date. Because social welfare often depends on completion times, a competitive advantage is generally given to the contractor who proposes an aggressive schedule. Furthermore, an increasing amount of contracts nowadays includes explicit time incentives through scoring auction systems such as A+B bidding where offers are scored using both the cost (A) and the completion time (B) [9, 8, 16]. The need for a cost-time tradeoff may arise in the telecommunications industry. Telephone and cable telecommunications companies are competing in a rapidly evolving business and technological conditions. Moreover, when no regulated infrastructure monopoly exists, national regulatory authorities tend to foster infrastructure-based competition which seems to lead to a higher level of nationwide coverage and social welfare [15, 10]. In such context, telecommunications companies continuously need to optimize their network infrastructure to deploy the latest technologies on short time-to-market schedules [13]. However, expanding an existing network in a race for subscribers involves high investment costs

---

*Email address:* fyp.cornillier@up.edu.pe (Fabien Cornillier)

*Preprint submitted to*

*May 29, 2013*

while time tends to erode the expected profits of connecting new geographical areas. The time dimension becomes a key issue for an effective network modeling. In such projects the network design decisions and the activity scheduling decisions are typically made sequentially: project managers and planners determine the minimal length or cost Steiner tree network, then schedule each selected activities so as to minimize the completion time. When opportunity cost of time exists, we demonstrate that taking a more holistic perspective by intertwining both decisions can result in a more effective network design.

This paper addresses problems in which a Steiner tree network has to be designed and the selected activities conjointly scheduled while considering a limited amount of renewable resources, so as to optimize an objective function, i.e., a distance, a completion time, a cost, or a present value. These problems are referred to as *Steiner Tree Scheduling Problems* (STSPs). We propose a formal definition and two mixed-integer programming formulations of the most basic Steiner Tree Scheduling Problem where the objective is to minimize the completion time. Two variants of the STSP are also presented: a *Time-Constrained Prize-Collecting Steiner Tree Scheduling Problem*, and a *Piecewise Steiner Tree Scheduling Problem*. An example is presented through the deployment of a fiber optic network in a cable telecommunications company, and a heuristic procedure is proposed for the STSP. Without loss of generality, the primary focus of this paper is on the time dimension of the network scheduling. Adaptations to time-cost tradeoff objective functions are trivial.

## 2. The Steiner Tree Scheduling Problem

The Steiner Tree Scheduling Problem is formulated as follows. Let  $G = \{V, A\}$  be a digraph with a vertex set  $V = \{1, \dots, n_v\}$ , and an arc set  $A = \{1, \dots, n_a\}$  such that  $A \subseteq V^2$ . The vertex set  $V$  is partitioned into three subsets: the root vertex  $\{1\}$ , the set  $V_S$  of Steiner vertices, and the set  $V_T$  of terminal vertices. Let  $K$  be a set of renewable resources where each resource  $k \in K$  has a total capacity  $R_k$  for any time interval. Each arc  $i$  corresponds to an activity associated with a duration  $d_i$ , and a non-negative amount  $r_{ik}$  of required resource  $k$ . The STSP is the problem of determining a directed simple path between the root vertex and every terminal vertices, and of scheduling each activity belonging to the selected paths under the precedence and resource constraints, so as to minimize the completion time, i.e., the *makespan*. Because it is a generalization of two strongly NP-hard problems, the Resource-Constrained Project Scheduling Problem [4] and the Steiner Tree in a Graph Problem [11, 6], the STSP is strongly NP-hard.

### 2.1. Mathematical formulations

This section presents two mixed integer programming formulations for the STSP in continuous time. In both formulations, three additional dummy vertices are added to  $V$ : a source vertex  $s = 0$ , an intermediary vertex  $t' = |V| + 1$ , and a sink vertex  $t = |V| + 2$ . The additional dummy arcs  $(s, 1)$ ,  $(t', t)$ , and  $(w, t')$  for each vertex  $w \in V_T$  are also added to  $A$ . All these dummy arcs are associated with a null duration,  $r_{(s,1),k}$  and  $r_{(t',t),k}$  are equal to  $R_k$  for each resource  $k \in K$ , and  $r_{(w,t'),k} = 0$  for each vertex  $w \in V_T$  and each resource  $k \in K$ . An illustration is given in Figure 1 in which the terminal vertices appear in black, and the additional dummy vertices appear in gray. The entire set of vertices is denoted by  $V'$  and the entire set of activities by  $A'$ . We denote by  $A_p$  and  $A_s$  the sets of predecessors and successors in  $A'$ , by  $\Gamma_i^-$  the set of the immediate predecessors of  $i \in A$ , from which exactly one activity must be completed before starting its execution, and by  $\Gamma_i^+$  the set of its immediate successors.

It is worth noting that a Steiner Tree Scheduling Problem in an undirected graph can be easily transformed into a Steiner Tree Scheduling Problem in a digraph [17]. Because the directed formulation is known to be much better for the Steiner Tree Problem than the undirected formulation [7], in this paper the proposed formulations are always related to the Steiner Tree Scheduling Problem in a digraph.

## 2.2. Activity-on-node flow resource allocation

The first formulation, a flow-based continuous-time formulation, is a compact description of the STSP with a resource allocation representation based on an activity-on-node (AoN) flow network as proposed by Artigues et al. [1] for the Resource-Constrained Project Scheduling Problem. In this formulation, the definition of the Steiner arborescence is based on the classical multicommodity flow formulation proposed by Beasley [3]. The STSP is formulated as follows:

$$(STSP) \quad \min \quad c_{max} \quad (1)$$

subject to

$$h^w(\delta^+(u)) - h^w(\delta^-(u)) = \begin{cases} 1 & \text{if } u = s \\ -1 & \text{if } u \in V_T \\ 0 & \text{otherwise} \end{cases}, \quad \forall u \in V, \forall w \in V_T, \quad (2)$$

$$h_{i=(u,v)}^w + h_{j=(v,u)}^{w'} \leq y_i, \quad \forall (i, j) \in A^2, \forall (w, w') \in V_T^2, \quad (3)$$

$$y_i = 1, \quad \forall i = (u, v), v \in \{t', t\}, \quad (4)$$

$$f_{ijk} \leq x_{ij}r_{ik}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (5)$$

$$f_{ijk} \leq x_{ij}r_{jk}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (6)$$

$$\sum_{j \in A_s} f_{ijk} = y_i r_{ik}, \quad \forall i \in A_p, \forall k \in K, \quad (7)$$

$$\sum_{i \in A_p} f_{ijk} = y_j r_{jk}, \quad \forall j \in A_s, \forall k \in K, \quad (8)$$

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i, j) \in A_p \times A_s, \quad (9)$$

$$\sum_{j \in \Gamma_i^+} x_{ij} \geq y_i, \quad \forall i \in A_p, \quad (10)$$

$$\sum_{i \in \Gamma_j^-} x_{ij} \geq y_j, \quad \forall j \in A_s, \quad (11)$$

$$2x_{ij} \leq y_i + y_j, \quad \forall (i, j) \in A_p \times A_s, \quad (12)$$

$$S_j - S_i - d_i \geq M(x_{ij} - 1), \quad \forall (i, j) \in A_p \times A_s, \quad (13)$$

$$c_{max} \geq S_i + d_i, \quad \forall i \in N_T, \quad (14)$$

$$c_{max} \in \mathbb{R}^+, \quad (15)$$

$$S_i \in \mathbb{R}^+, \quad \forall i \in A', \quad (16)$$

$$f_{ijk} \in \mathbb{R}^+, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (17)$$

$$h_i^w \in \mathbb{R}^+, \quad \forall i \in A, \forall w \in V_T, \quad (18)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A_p \times A_s, \quad (19)$$

$$y_i \in \{0, 1\}, \quad \forall i \in A', \quad (20)$$

where  $c_{max}$  is the makespan,  $y_i$  is a binary variable with value 1 iff activity  $i$  is selected,  $x_{ij}$  is a binary variable with value 1 iff activity  $j$  is scheduled after the completion of activity  $i$ ,  $h_i^w$  is the amount of commodity associated to terminal vertex  $w$  and involved in activity  $i$ ,  $f_{ijk}$  is the amount of resource  $k$  transferred from activity  $i$  to activity  $j$ , and  $S_i$  is the starting time of activity  $i$ . In this formulation we denote by  $\delta^+(u)$  the set of activities outgoing from vertex  $u$ , by  $\delta^-(u)$  the set of activities incoming to vertex  $u$ , and we set  $h^w(S) = \sum_{i \in S} h_i^w$  for any subset  $S \subset A$ . Constraints (2) state that there exist a path from  $s$  to each terminal vertex  $w$  if  $y_i$  is integral and if a unit of commodity flow exists from  $s$  to  $w$ . Constraints (3) prevent two opposite commodity flows [18, 2]. By constraints (4), all the dummy activities ending at vertices  $t'$  and  $t$  have to be selected. Constraints (5) to (7) represent the resource restrictions: constraints (5) and (6) limit the amount of resources transferred from activity  $i$  to activity  $j$ , while constraints (7) ensure that all resources are transferred from each activity to other activities once completed, and constraints (8) guarantee the satisfaction of the resource requirements of each activity. Constraints (9) prevent any cycle between two activities. Constraints (11) and (10) correspond to precedence constraints, and constraints (12) state that activity  $j$  can be forced to start after the completion of activity  $i$  iff both activities are selected. By constraints (13) activity  $j$  cannot start before the completion of activity  $i$  when  $j$  is processed after  $i$ . Constraints (14) give the expression of the makespan.

*Bound on the starting times.* Since durations are non-negative values, it is possible to compute  $\theta_u$  the minimum path time in  $G$  from the root vertex  $s$  to any vertex  $u \in V_S \cup V_T$  in polynomial time, and the following constraints can be added:

$$S_i \geq \theta_u, \quad \forall u \in V_S \cup V_T, i = (u, v) \in A. \quad (21)$$

These constraints significantly improve the LP-relaxation.

### 2.3. Minimal forbidden sets

The second formulation uses minimal forbidden sets. The concept of minimal forbidden set, defined as a minimal set of activities for which the required resources exceed the availability of a least one resource, has been proposed by Radermacher [20] to represent resource restrictions. In this formulation, resource restrictions are implicitly described as a polynomial number of constraints by which at least two activities of any selected minimal forbidden set do not overlap. This is formalized by replacing constraints (5) to (8) in (STSP) with the following constraints:

$$\sum_{i,j \in F} x_{ij} \geq \sum_{i \in F} y_i - |F| + 1, \quad \forall F \in \mathcal{F}, \quad (22)$$

where  $\mathcal{F}$  denotes the power set of all the minimal forbidden sets. Because of the polynomial number of minimal forbidden sets, these constraints are in practice introduced as lazy constraints whenever a violation is observed.

### 2.4. Numerical illustration

Figures 1 and 2 depict a single-resource instance of the STSP with  $|V| = 5$ ,  $|A| = 5$ , and  $R = 10$ . The set of vertices  $V$  is partitioned into a root vertex 1, a set of Steiner vertices  $\{2, 3\}$  and a set of terminal vertices  $\{4, 5\}$ . The pair of values depicted on each arc corresponds to the duration  $d_i$  and the resource request  $r_i$ . In Figure 1a, arcs in bold correspond to the solution of the classical Steiner Minimal Tree Problem. In this solution, the set of activities  $\{4, 5\}$  is a

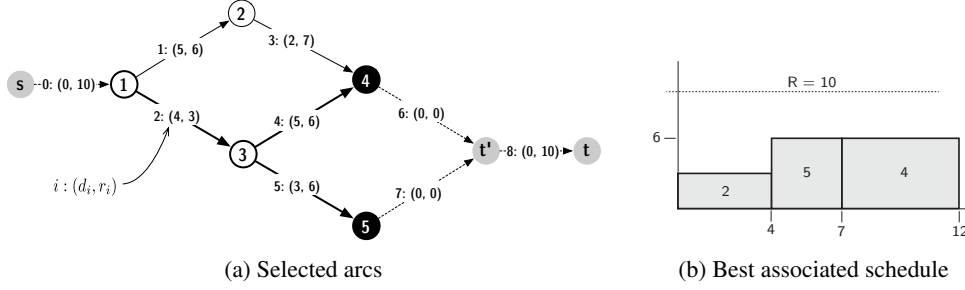


Figure 1: Optimal solution of the Steiner Tree Problem

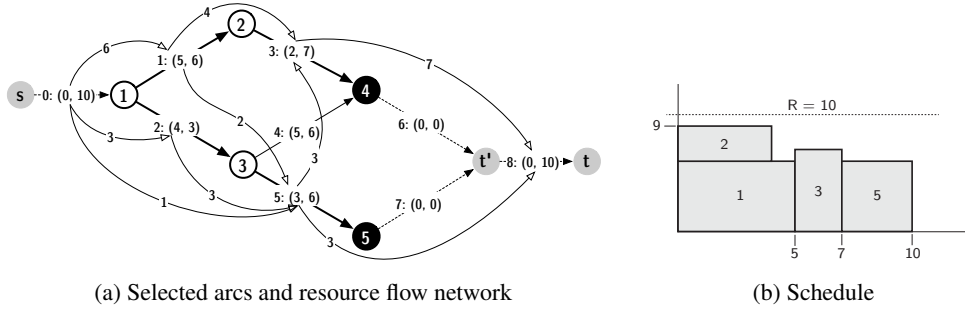


Figure 2: Optimal solution of the STSP

minimum forbidden set which prevents activities 4 and 5 to be scheduled simultaneously: both the minimum total length and the best makespan are equal to 12 (Figure 1b). Figures 2 shows that a better makespan can be found in spite of a longer Steiner tree: the total length is equal to 14, and the makespan to 10.

### 3. The Time-Constrained Prize-Collecting Steiner Tree Scheduling Problem

The Time-Constrained Prize-Collecting Steiner Tree Scheduling Problem is a generalization of the Steiner Tree Scheduling Problem where a non-negative cost  $c_i$  is associated with each activity  $i$ , and a non-negative revenue  $p_w$  is associated with each terminal vertex  $w$ . The TCPC-STSP is the problem of selecting a subset of terminal vertices, of determining a directed simple path between the root vertex and every selected terminal vertex, and of scheduling each activity belonging to the selected paths under the precedence constraints, the resource constraints, and a time limit  $T_w$  associated with each terminal vertex, so as to maximize the total profit equal to the revenue, minus the sum of activity costs. The problem is formulated as follows:

$$(TCPCSTSP) \quad \max \sum_{i=(u,w), w \in V_T} y_i p_w - \sum_{i \in A} y_i c_i \quad (23)$$

subject to

$$h^w(\delta^+(u)) - h^w(\delta^-(u)) = \begin{cases} y_i & \text{if } u = s \\ -y_i & \text{if } u \in V_T \\ 0 & \text{otherwise} \end{cases}, \quad \forall i = (u, v), v \in V, \forall w \in V_T, \quad (24)$$

$$h_{i=(u,v)}^w + h_{j=(v,u)}^{w'} \leq y_i, \quad \forall (i, j) \in A^2, \forall (w, w') \in V_T^2, \quad (25)$$

$$y_i = 1, \quad i = (t', t), \quad (26)$$

$$\sum_{i=(u,v) \in A} y_i \leq 1, \quad \forall v \in V, \quad (27)$$

$$f_{ijk} \leq x_{ij} r_{ik}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (28)$$

$$f_{ijk} \leq x_{ij} r_{jk}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (29)$$

$$\sum_{j \in A_s} f_{ijk} = y_i r_{ik}, \quad \forall i \in A_p, \forall k \in K, \quad (30)$$

$$\sum_{i \in A_p} f_{ijk} = y_j r_{jk}, \quad \forall j \in A_s, \forall k \in K, \quad (31)$$

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i, j) \in A_p \times A_s, \quad (32)$$

$$\sum_{j \in \Gamma_i^+} x_{ij} \geq y_i, \quad \forall i \in A_p, \quad (33)$$

$$\sum_{i \in \Gamma_j^-} x_{ij} \geq y_j, \quad \forall j \in A_s, \quad (34)$$

$$2x_{ij} \leq y_i + y_j, \quad \forall (i, j) \in A_p \times A_s, \quad (35)$$

$$S_j - S_i - d_i \geq M(x_{ij} - 1), \quad \forall (i, j) \in A_p \times A_s, \quad (36)$$

$$S_i + d_i - M(1 - y_i) \leq T_w, \quad \forall i = (u, w), w \in V_T, \quad (37)$$

$$S_i \in \mathbb{R}^+, \quad \forall i \in A', \quad (38)$$

$$f_{ijk} \in \mathbb{R}^+, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (39)$$

$$h_i^w \in \mathbb{R}^+, \quad \forall i \in A, \forall w \in V_T, \quad (40)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A_p \times A_s, \quad (41)$$

$$y_i \in \{0, 1\}, \quad \forall i \in A', \quad (42)$$

where constraints (24) state that there exist a path from  $s$  to each selected terminal vertex  $w$  if  $y_i$  is integral and if a unit of commodity flow exists from  $s$  to  $w$ . By constraint (26), the dummy activity ending at vertex  $t$  has to be selected. Constraints (27) limit the number of selected activities ending at any non-dummy vertex to one.

#### Illustration of the TCPCSTSP

Figure 3 depicts an instance of the Time-Constrained Prize-Collecting Steiner Tree Scheduling Problem with a time limit  $T_w = 9$  for all terminal vertices. The third value of the triplet associated with each arc corresponds to the cost, and profits are indicated above each terminal vertex. The optimal solution represented with bold arcs gives a net profit equal to 2 by reaching the single terminal vertex 4 in 9 units of time. There is no feasible solution with both terminal vertices, and 12 units of time would be needed to generate a higher profit of 4, as shown in Figure 1.

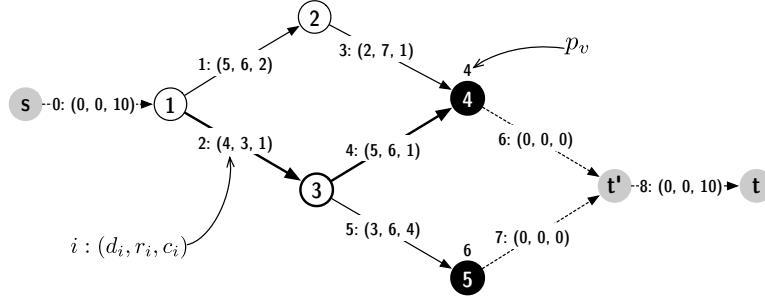


Figure 3: Optimal solution of PCSTSP with  $T_w = 9$  for all terminal vertices

#### 4. The Piecewise Steiner Tree Scheduling Problem

The Piecewise Steiner Tree Scheduling Problem corresponds to the Steiner Tree Scheduling Problem with relaxed precedence constraints. In this problem the arc subset of each intermediary solution does not need to be connected during the construction of the Steiner tree, and the profit of a terminal vertex is made only when connected to the root vertex. This section presents the formulation of a simple version of the Piecewise Steiner Tree Scheduling Problem in which the objective is to minimize the average connection time over all the terminal vertices.

$$(PSTSP) \quad \min \quad \frac{1}{|V_T|} \sum_{w \in V_T} c_w \quad (43)$$

subject to

$$h^w(\delta^+(u)) - h^w(\delta^-(u)) = \begin{cases} 1 & \text{if } u = s \\ -1 & \text{if } u \in V_T \\ 0 & \text{otherwise} \end{cases}, \quad \forall u \in V, \forall w \in V_T, \quad (44)$$

$$h_i^w \leq y_i, \quad \forall i \in A, \forall w \in V_T, \quad (45)$$

$$\sum_{w \in V_T} h_i^w \geq y_i, \quad \forall i \in A, \quad (46)$$

$$y_i = 1, \quad \forall i \in V' \setminus V, \quad (47)$$

$$f_{ijk} \leq x_{ij} r_{ik}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (48)$$

$$f_{ijk} \leq x_{ij} r_{jk}, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (49)$$

$$\sum_{j \in A_s} f_{ijk} = y_i r_{ik}, \quad \forall i \in A_p, \forall k \in K, \quad (50)$$

$$\sum_{i \in A_p} f_{ijk} = y_j r_{jk}, \quad \forall j \in A_s, \forall k \in K, \quad (51)$$

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i, j) \in A_p \times A_s, \quad (52)$$

$$\sum_{j \in A_s} x_{ij} \geq y_i, \quad \forall i \in A_p, \quad (53)$$

$$\sum_{i \in A_p} x_{ij} \geq y_j, \quad \forall j \in A_s, \quad (54)$$

$$2x_{ij} \leq y_i + y_j, \quad \forall (i, j) \in A_p \times A_s, \quad (55)$$

$$S_j - S_i - d_i \geq M(x_{ij} - 1), \quad \forall (i, j) \in A_p \times A_s, \quad (56)$$

$$S_i + d_i - M(1 - h_i^w) \leq c_w, \quad \forall w \in V_T, \forall i \in A, \quad (57)$$

$$c_w \in \mathbb{R}^+, \quad \forall w \in V_T, \quad (58)$$

$$S_i \in \mathbb{R}^+, \quad \forall i \in A', \quad (59)$$

$$f_{ijk} \in \mathbb{R}^+, \quad \forall (i, j) \in A_p \times A_s, \forall k \in K, \quad (60)$$

$$h_i^w \in \mathbb{R}^+, \quad \forall i \in A, \forall w \in V_T, \quad (61)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A_p \times A_s, \quad (62)$$

$$y_i \in \{0, 1\}, \quad \forall i \in A', \quad (63)$$

where constraints (54) and (53) state that there exist at least one incoming (resp. outgoing) resource flow to (resp. from) any selected arc, and constraints (57) give the expression of the connection time  $c_w$  of each terminal vertex  $w$ .

#### Special cases

The Piecewise Steiner Tree Scheduling Problem gives rise to two new problems. The first problem, named as *Piecewise Spanning Tree Scheduling Problem* (PSPSTSP), is the special case where all non-root vertices of  $V$  are terminal vertices ( $V_S = \emptyset$ ). The second problem, referred to as *Piecewise Path Scheduling Problem* (PPSP), is the single terminal vertex case. Because these two special cases are generalizations of the strongly NP-hard RCPSP with a single resource and no precedence constraints called the Resource-Constrained Scheduling Problem [5], the PSPSTSP and the PPSP are strongly NP-hard.

#### Illustration of the PWSTSP

Figure 4 depicts an instance of the PWSTSP. This example shows that the contiguous activities 2 and 5 are scheduled simultaneously. Terminal vertex 5 is reached at time 3, but connected to the root at time 4, i.e., at the end of activity 2, hence we have  $c_5 = 4$ . Terminal vertex 4 is reached at time 8 and before the completion of arc 2, so  $c_4 = 8$ . The minimum average connection time is equal to 6.

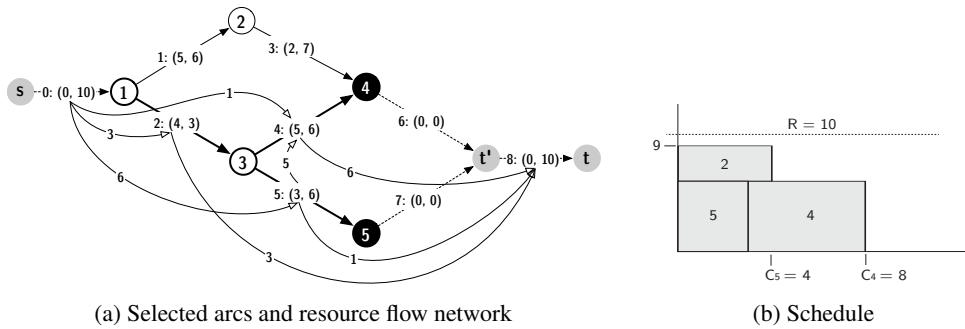


Figure 4: Optimal solution of the PWSTSP



## 5. Deployment of a fiber optic network

Suppose a telecommunications company plans to deploy fiber optic connections in a set of geographical areas. Considering its existing old copper technology network infrastructure (i.e., poles, ducts, conduits and rights-of-way) and the possible extensions, a set of intermediate node locations have been identified, and the cost, the duration, and resources requirements have been evaluated for each potential node interconnections and each type of cable pathway construction (aerial, direct-buried, underground conduits). Facing a strong competition, the company needs to plan the cable routes of this new fiber optic network given a fixed resource capacity while minimizing the expected time-to-market. In this example, we consider a single-resource instance made of 14 vertices partitioned into a root vertex (main node), 5 terminal vertices (distribution nodes), and 8 Steiner vertices (intermediate nodes), a set of 62 potential activities including 4 arcs from the root vertex, and 29 pairs of antiparallel arcs joining terminals and Steiner vertices. The capacity of the single resource is equal to 10. Figure 10 gives a complete representation of this instance, including durations in days, resource requirements, activity costs, and terminal estimated profits.

All computations of Sections 5 and 6 were performed using the C++ Gurobi 5.1 Callable Library on an Intel Core i7-920 2.67 GHz processor with 16GB of RAM under a 64-bit Linux operating system.

### 5.1. Solving a Steiner Minimal Tree Problem and scheduling the selected activities

An approximation of the optimal solution could be given by decomposing the problem into a Steiner Minimal Tree Problem and a Resource-Constrained Project Scheduling Problem, i.e., by determining the minimum duration Steiner tree then scheduling its corresponding activities. The solution is depicted in Figures 5 and 7. Solving the Steiner Minimal Tree Problem to optimality gives a selection of 9 activities for a total of 377 days. An optimal solution of the corresponding RCPSP instance gives a makespan of 329 days. Only 23 CPU seconds are necessary to solve the two subproblems. Another option is to solve the Steiner Minimal Tree Problem with weights  $d_i \cdot r_i$ . This method gives a selection of 9 activities for a total of 421 days, and a makespan of 321 days.

### 5.2. Solving the STSP instance

An exact solution can be computed by using the corresponding Steiner Tree Scheduling Problem instance. This solution is depicted in Figures 6 and 8. It is made of 9 activities for a total of 431 days, with a 14.3% increase compared to the decomposition method, and a makespan of 244 days, which evidences a decrease of 25.8%. Solving this instance to optimality requires 238 CPU seconds.

### 5.3. Solving the PWSTSP instance

The solution depicted in Figures 9 and 11 have been found after 30 minutes of computation, with an optimality gap of 51.3% after 12 hours. It is made of 9 activities for a total of 419 days, and an average connection time of 114.6 days. We also observe a makespan of 183 days, showing a decrease of 25% compared to the solution obtained for the STSP instance, and of 44.4% when compared to the solution found with the decomposition method.

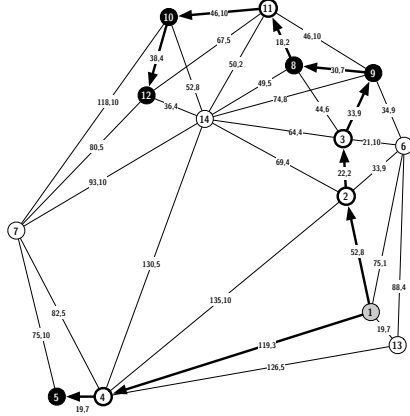


Figure 5: Optimal Steiner tree

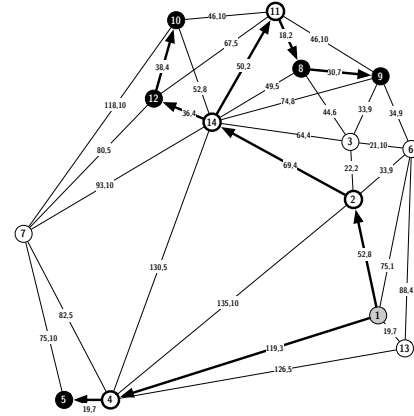


Figure 6: Optimal solution for the STSP

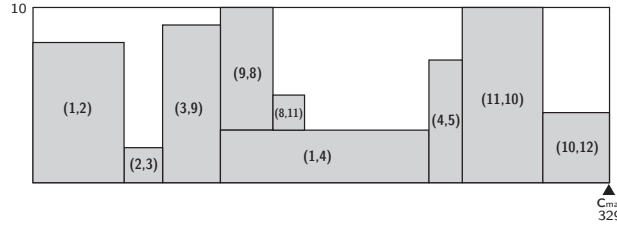


Figure 7: Optimal scheduling for the decomposition method

#### 5.4. Solving a TCPCSTSP instance

Suppose the company set up a deadline of 240 days to deploy its first fiber optic services. Because no feasible solution involving the entire set of 5 distribution nodes exists, the project manager has to choose the most profitable subset of distribution nodes which could be connected in the next 240 days. The problem becomes a TCPCSTSP with  $T_w = 240$  for each terminal  $w$ , where the objective is to maximize the profit of the company in this time window. An optimal solution is depicted in Figure 10 which gives a profit of 1686 and a makespan of 239 days by operating distribution nodes 8, 9, 10, and 12. Solving this instance to optimality requires 460 CPU seconds.

### 6. Heuristic procedure for the Steiner Tree Scheduling Problem

As the number of activities grows, the MIP formulation of the STSP becomes more difficult to solve. For larger problems we propose a simple heuristic procedure based on a priority list solution representation [19] where an activity list sorted in decreasing order of priorities represents a solution, and on an adaptation of the serial Schedule Generation Scheme (SGS) [12] employed as decoding procedure. It has been shown by Kolisch [14] that there is always a priority list from which the serial SGS generates an optimal solution for scheduling problems when regular performance measures are used [21], which is the case of the makespan and, more generally, because the serial SGS decoding procedure is a surjection to the set of active schedules, there is

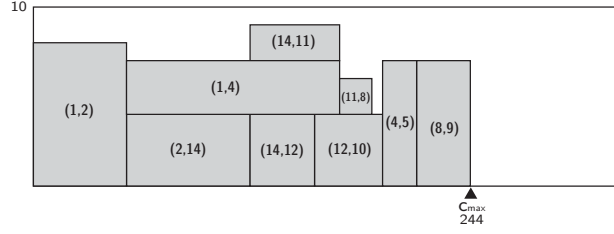


Figure 8: Optimal scheduling for the STSP

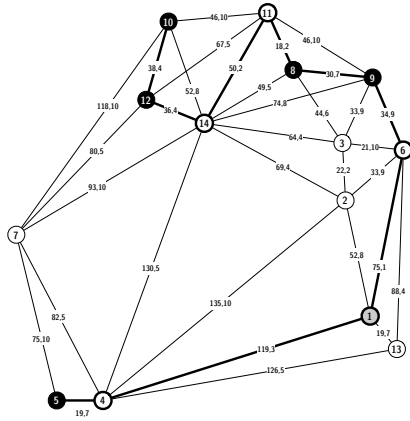


Figure 9: Solution for the PWSTSP

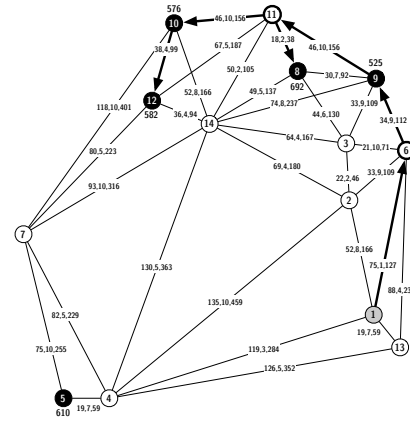


Figure 10: Optimal solution for the TCPCSTSP

always a priority list from which the serial SGS generates a schedule where no activity can be locally or globally left shifted.

### 6.1. Serial Schedule Generation Scheme

In each stage of the proposed serial SGS, the first eligible activity of the priority list is scheduled at its earliest precedence and resource feasible starting time, and all the non-scheduled activities pointing to its end node are declared as ineligible. Contrary to the Resource-Constrained Project Scheduling Problem where all the activities have to be scheduled while satisfying precedence and resource constraints, in the Steiner Tree Scheduling Problem only a subset of activities have to be selected and scheduled. Hence procedure stops when all the terminal nodes are spanned or when no eligible activity exists, and a peeling procedure is applied to the solution to remove all the arcs pointing to non-terminal leaves.

### 6.2. Variable depth search with tabu node

We propose to use a variable depth search algorithm based on 2-opt activity exchanges. The serial SGS decoding procedure is used to evaluate each intended exchange. A series of exchanges are intended sequentially on the priority list and retained each time they don't deteriorate the solution. Each time an exchange is retained the swapped activities are blocked and cannot be considered anymore for future exchanges. If no more exchange can be applied without deteriorating the current solution, the blocked activities are unblocked, the selected activities are moved

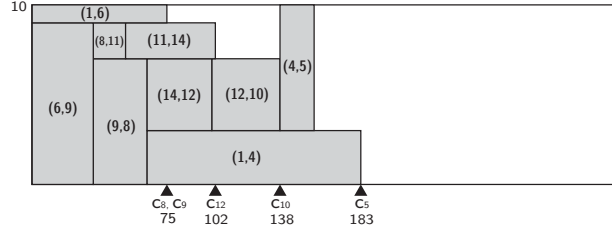


Figure 11: Scheduling for the PWSTSP

to the front of the priority list in the order of their scheduling, and a new series of exchanges is intended. Whenever no improving exchange can be found, an intermediate node of the Steiner tree solution is randomly declared as tabu, the previous tabu node is released, if any, and a new variable depth search descent is applied to the priority list while ignoring all activities incident to the tabu node.

### 6.3. Computational results

The heuristic procedure has been tested on 20 instances randomly generated and made of 20 nodes, including the root node and 5 terminal nodes. To generate these instances, the vertices coordinates are uniformly distributed in a disk of unit radius, and the edge set is built by computing the Delaunay triangulation of the vertex set, and complemented by all edges with distance  $d$  less than  $3.2/\sqrt{n_v}$ . Each edge is associated to a duration equal to  $\lceil 100 \cdot d \rceil$ , and an amount of required resources is chosen from a discrete uniform distribution  $U(1, 10)$ . The capacity of the single-resource is set to 10. For all instances, the performance of the proposed heuristic has been compared to the STSP formulation solved with the commercial solver Gurobi 5.1, within a time limit of four CPU hours. Table 1 provides the performances of the proposed heuristic and the exact method. The best integer solutions and the time to generate them are reported. For the exact method, the lower bounds ( $LB$ ) obtained after four CPU hours are also reported, and an asterisk signals when an optimal solution has been found. The solutions generated by the heuristic method were always equal or better than the solutions generated by the exact method. For eight instances solved to optimality, the heuristic procedure found the optimal solution in 0.63% of the time needed with the exact method. For all the instances for which a same objective value has been found, the time needed by the heuristic procedure represents 0.85% of the time needed by the exact method. The heuristic method also generated five improved solutions over the solutions found with the exact method (in bold).

## 7. Conclusions

The Steiner Tree Scheduling Problems represent a new class of scheduling problems in which the subset of activities to be scheduled is part of the decision variables. A basic version of the Steiner Tree Scheduling Problem have been presented, with two variants: the Time-Constrained Prize-Collecting Steiner Tree Scheduling Problem, and the Piecewise Steiner Tree Scheduling Problem. However, the proposed formulations yield to poor LP-relaxations due to the big-M constants. This problem is more acute in the piecewise version which makes its resolution even more difficult. As the Resource-Constrained Project Scheduling Problem and the Steiner Tree

Table 1: Performance comparison

Instance	A	Exact method			Heuristic method	
		Best	LB	Time (s)	Best	Time (s)
1	95	230	161	178	<b>226</b>	131
2	94	227	*	7104	227	3
3	92	158	*	1456	158	73
4	89	160	*	1805	160	25
5	92	218	209	9869	218	2
6	93	185	156	6432	185	309
7	93	221	194	1782	221	907
8	93	245	198	12490	<b>236</b>	4
9	94	240	183	670	<b>233</b>	23
10	92	222	214	8279	222	8
11	91	196	157	1028	<b>178</b>	23
12	93	216	163	5184	216	10
13	93	224	*	2485	224	399
14	92	184	127	1315	184	7
15	93	257	184	13077	257	13645
16	91	156	*	9553	156	179
17	92	232	147	286	<b>218</b>	2389
18	89	207	*	993	207	3
19	95	170	*	746	170	1
20	94	205	*	2228	205	2

Problem in a graph, both among the most intractable combinatorics problems, the Steiner Tree Scheduling Problems lead to challenging computational issues.

### Acknowledgements

This work was partially supported by the program STIC-AmSud OVIMINE. This support is gratefully acknowledged.

### References

- [1] C. Artigues, P. Michelon, and S. Reusser, *Insertion techniques for static and dynamic resource-constrained project scheduling*, European Journal of Operational Research **149** (2003), 249-267.
- [2] A. Balakrishnan, T. L. Magnanti, and R. T. Wong, *A dual-ascent procedure for large-scale uncapacitated network design*, Operations Research **37** (1989), 716-740.
- [3] J.E. Beasley, *An algorithm for the Steiner problem in graphs*, Networks **14** (1984), 147-159.
- [4] J. Blazewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Scheduling subject to resource constraints: classification and complexity*, Discrete Applied Mathematics **5** (1983), 11-24.
- [5] M. R. Garey and D. S. Johnson, *Complexity results for multiprocessor scheduling under resource constraints*, SIAM Journal on Computing **4** (1975), 397-411.
- [6] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified NP-complete graph problems*, Theoretical Computer Science **1** (1976), 237-267.
- [7] M. X. Goemans and Y. Myung, *A catalog of steiner tree formulations*, 1991.
- [8] Z. J. Herbsman, *A+B bidding method --- Hidden success story for highway construction*, Journal of Construction Engineering and Management **121** (1995), 430-437.

- [9] Z. J. Herbsman, W. T. Chen, and W. C. Epstein, *Time is money: Innovative contracting methods in highway construction*, Journal of Construction Engineering and Management **121** (1995), 273-281.
- [10] F. Jeanjean and J. Liang, *Role of access charges in the migration from copper to FTTH*, International Journal of Management and Network Economics (forthcoming).
- [11] R. M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations (R. Miller and J. Thatcher, eds.), Plenum Press, 1972, pp. 85-103.
- [12] J. E. Kelley, *The critical-path method: Resources planning and scheduling*, Industrial Scheduling (J. F. Muth and G. L. Thompson, eds.), Prentice-Hall, New-Jersey, 1963.
- [13] J. Kittl, M. Lunderborg, and E.-O. Ruhle, *Infrastructure-based versus service-based competition in telecommunications*, Communications & Strategies **64** (2006), 67-87.
- [14] R. Kolisch, *Series and parallel resource-constrained project scheduling methods revisited: Theory and computation*, European Journal of Operational Research **90** (1996), 320-333.
- [15] M. Lebourges, *Competition via investment, an efficient model for FTTH rollout*, Communications & Strategies **78** (2010), 45-66.
- [16] G. Lewis and P. Bajari, *Procurement contracting with time incentives: Theory and evidence*, Quarterly Journal of Economic Review **126** (2011), 1173-1211.
- [17] N. Maculan, *The steiner problem in graphs*, Annals of Discrete Mathematics **31** (1987), 185-212.
- [18] R. K. Martin, *A sharp polynomial size linear programming formulation of the minimum spanning tree problem*, Working paper, University of Chicago, 1986.
- [19] E. Pinson, C. Prins, and F. Rullier, *Using tabu search for solving the resource-constrained project scheduling problem*, Proceedings of the fourth international workshop on project management and scheduling, 1994, pp. 102-106.
- [20] F. Radermacher, *Scheduling of project networks*, Annals of Operations Research **4** (1985), 227-252.
- [21] A. Sprecher, R. Kolisch, and A. Drexl, *Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem*, European Journal of Operational Research **80** (1995), 94-102.