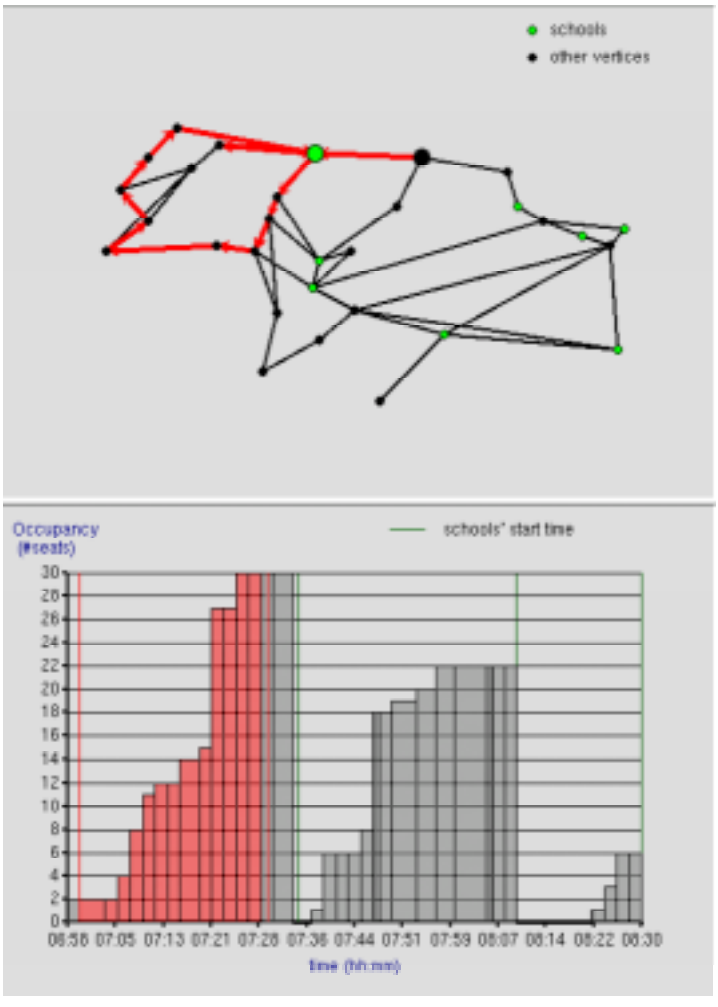


Decision-aid Methodology for the School Bus Routing and Scheduling Problem

Michela Spada, ROSO-IMA-EPFL
Michel Bierlaire, ROSO-IMA-EPFL
Thomas Liebling, ROSO-IMA-EPFL

Conference paper STRC 2003
Session Model & Scheduling

STRC

3rd Swiss Transport Research Conference

Monte Verità / Ascona, March 19-21, 2003

Decision-aid Methodology for the School Bus Routing and Scheduling Problem

Michela Spada
Institut de Mathématiques
École Polytechnique Fédérale de Lausanne

Phone: +41 21 693 81 00
Fax: +41 21 693 55 70
e-Mail: michela.spada@epfl.ch

Michel Bierlaire
Institut de Mathématiques
École Polytechnique Fédérale de Lausanne

Phone: +41 21 693 25 37
Fax: +41 21 693 55 70
e-Mail: michel.bierlaire@epfl.ch

Thomas Liebling
Institut de Mathématiques
École Polytechnique Fédérale de Lausanne

Phone: +41 21 693 25 03
Fax: +41 21 693 55 70
e-Mail: thomas.liebling@epfl.ch

Abstract

We consider the school bus routing and scheduling problem, where transportation demand is known and bus scheduling can be planned in advance. We present a comprehensive methodology designed to support the decision of practitioners. We first propose a modeling framework where the focus is on optimizing the level of service for a given number of buses. Then, we describe an automatic procedure generating a solution to the problem. It first builds a feasible solution, which is subsequently improved using a heuristic.

We analyze two important issues associated with this methodology. On the one hand, we analyze the performance of three types of heuristics both on real and synthetic data. We recommend the use of a simulated annealing technique exploring infeasible solutions, which performs slightly better than all others. More importantly, we find that the performance of all heuristics is not globally affected by the choice of the parameters. This is important from a practitioner viewpoint, as the fine tuning of algorithm parameters is not critical for its performance. On the other hand, we propose an interactive tool allowing the practitioner to visualize the proposed solution, to test its robustness, and to dynamically rebuild new solutions if the data of the original problem are modified.

Keywords

Vehicle Routing - Scheduling - Heuristics - 3rd Swiss Transport Research Conference - STRC 2003 - Monte Verità

1 Introduction

Busing systems are common in countries where school assignment is imposed by home location, as in Switzerland, France, and Italy for instance. Since the transportation demand is known in advance, bus schedules can in principle be planned efficiently, however, this results in an intricate combinatorial optimization problem.

In this paper, we present a decision-aid methodology, combining automatic solution generation and the expertise of practitioners. We first propose a modeling framework, where the concept of level of service is captured by two different objective functions. We obtain a nonlinear integer programming problem which cannot be solved exactly in reasonable time for problems with realistic size (Section 3). Therefore, we propose and analyze automatic procedures to build a solution, all of which start by generating an initial feasible solution (see Section 4). In most cases, that solution is not satisfactory and must be improved, thus in Section 5, we propose three automatic procedures to improve the quality of the solution: a tabu search heuristic, and two variants of simulated annealing.

An important drawback of heuristics for practitioners is the choice of a specific heuristic and the identification of appropriate parameters. We have tested several instances of each proposed heuristic on a set of 30 problems. The results, presented in Section 6, suggest that the simulated annealing heuristic exploring infeasible solutions is the most efficient out of the three, irrespectively of the choice of parameters.

Finally, we present in Section 7 an interactive decision-aid tool. Its main objective is to analyze the relevance of the solution produced by the automatic procedure and possibly modify it manually. This is specifically important when the conditions for the bus operations are modified. Indeed, in that case, it is preferable to manually create a new solution from the original one, instead of relaunching the whole process on the modified problem.

2 Literature review

Bodin et al. (1983) provide a general description of the vehicle routing and scheduling problem. For the special case of school buses, there are various approaches described in the literature. They differ in the way of the problem decomposition, in the modeling assumptions and in the solution algorithms.

With respect to the problem decomposition, we can distinguish a school-based approach and a home-based approach. In the school-based approach, a separate problem is solved for each school, and no mixed loads are allowed, in the sense that children attending different schools are not allowed to travel in the same bus at the same time. The school-based approach is preferred by many authors, such as Bodin and Berman (1979), Angel et al. (1972), Bennett and Gazis (1972), Desrosiers et al. (1981), Newton and Thomas (1969), Clarke and Wright (1964), Rosenkrantz et al. (1974) and Gavish and Shlifer (1979). The home-based approach aims at solving the problem for each child at a time. It is more flexible as mixed loads are considered. However, the update of the solution when a child is included is more complicated. This approach has been proposed by Braca et al. (1994). In this paper, we prefer the school-based approach, as it greatly simplifies the problem, but we follow Braca et al. (1994) in allowing mixed loads.

From the modeling viewpoint, most authors aim at minimizing the costs, that is the number of buses (Clarke and Wright, 1964, Bodin and Berman, 1979, Desrosiers et al., 1981, Swersey and Ballard, 1984, Braca et al., 1994), or a combination of the number of buses and the total travel time (Gavish and Shlifer, 1979). In this paper, we prefer to explicitly optimize the level of service provided by the bus operator. Such an approach is not usual in the literature, where the level of service is often captured in the constraints. The only reference we have found is a paper by Bennett and Gazis (1972) who include the total travel time spent by all children in their objective function. Bodin and Berman (1979), Gavish and Shlifer (1979), Desrosiers et al. (1981), and Braca et al. (1994) add an upper bound on the travel time for each child. Desrosiers et al. (1981), Swersey and Ballard (1984), and Braca et al. (1994) add a time window on the arrival at school. Desrosiers et al. (1981) add also an upper bound on the waiting time between school arrival time and school starting time and uppers bounds on the number of students at stops and on routes. Braca et al. (1994) impose an earliest pick-up time for children. In addition, they require a minimal number of children to create a route.

Various solution algorithms have been proposed. For school-based approaches, a routing problem is solved first for each school. Bennett and Gazis (1972) use a method proposed by Clarke and Wright (1964), assigning a direct depot-stop-school route to each stop, and then merging those routes to comply with the problem constraints. Finally, a Lin-3-opt procedure (Lin, 1965) tries to improve the solution. Bodin and Berman (1979) use a method proposed by Newton and Thomas (1969). They start by solving a Traveling Salesman Problem (TSP) with a Lin-3-opt, and then split the solution to comply with the problem constraints. Desrosiers et al. (1981) use adaptations of the methods by Newton and Thomas (1969), Clarke and Wright (1964) and the insertion technique proposed by Rosenkrantz et al., 1974, followed by a Lin-2-opt. Gavish and Shlifer (1979) use a different approach, solving the problem for each school by applying a branch-and-bound procedure in which a sequence of assignment problems are solved.

Once feasible tours have been generated, they must be merged in order to optimize the objective function. Bodin and Berman (1979) use a Lin-2-opt. Desrosiers et al. (1981) propose a heuristic based on solving a sequence of transportation problems. Gavish and Shlifer (1979) formulate this problem as a modified assignment problem. Swersey and Ballard (1984) use a linear integer program solved using linear or Lagrangian relaxations.

Braca et al. (1994), preferring a home-based approach, solve the whole problem in one stage. They initially construct a route between a randomly selected home and associated school. Then, they insert home–school pairs in a greedy procedure, choosing first those pairs that minimize the total route length, and making sure that all constraints are satisfied. That solution is not revised, except if it contains buses with very few children.

3 Formulation

In this section we describe notations and give a formulation to school bus routing and scheduling problem as mathematical program.

3.1 Notations

The transportation network is given by $G = (\mathcal{V}, \mathcal{E}, \ell)$ where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ a set of edges and ℓ a weight function associating a non-negative weight to each edge (typically, the travel time necessary to traverse the edge). Based on ℓ, we denote by $d(i, j)$ the travel time between nodes i and j, obtained as the travel time of the shortest path (measured in travel time) in G between i and j. We denote by $\mathcal{H} = \{h_1, \ldots, h_H\} \subseteq \mathcal{V}$ the set of nodes, called homes, where children get on the bus. Similarly, we denote by $\mathcal{S} = \{s_1, \ldots, s_S\} \subseteq \mathcal{V}$ the set of nodes where schools are located. Class at school s_i begins at time o_{s_i}.

There are $|\mathcal{C}|$ children commuting from home to school. Child $c \in \mathcal{C}$ has home node $h(c)$ and school node $s(c)$. Each child in a bus takes up some space, denoted by $q(c)$. Note that a "child" in the model may represent in reality a group of several children with the same characteristics, that is the same home location and the same school. This simplifies the formulation with a moderate loss of flexibility, in that children with similar characteristics are sometimes required to make exactly the same journey on the same bus. We denote by $N_{\mathcal{C}}$ ($\geq |\mathcal{C}|$) the number of individual children. We also assume that each child performs exactly one trip (without transfer).

For each school $s_i \in \mathcal{S}$, we denote by $\mathcal{C}(s_i)$ the set $\{c \in \mathcal{C} \mid s(c) = s_i\}$ of children attending school s_i. We denote by $\mathcal{H}(s_i) \subset \mathcal{H}$ the set of nodes $\cup_{c \in \mathcal{C}(s_i)} h(c)$, that is the set of home nodes of children attending school s_i. There are B buses to transport the children. Bus $b \in \mathcal{B} = \{1, \ldots, B\}$ has a capacity Q_b (measured as the same unit as the children's space requirement $q(c)$) and is typically a number of seats. Note that $q(c)$ may be fractional, as some buses may load more children than the actual number of seats, for example when the bus has benches rather than individual seats and young children are carried.

A solution to the school bus routing and scheduling problem consists in specifying for each bus a tour, that is a list of stops, and for each stop, a list of children to pick-up and/or drop. Bus b performs n_b stops denoted by a^b_α, with $\alpha = 1, \ldots, n_b$. Each a^b_α is a quadruplet

$$a^b_\alpha = (v^b_\alpha, t^b_\alpha, \overline{\mathcal{C}}^b_\alpha, \underline{\mathcal{C}}^b_\alpha), \tag{1}$$

where $v^b_\alpha \in \mathcal{V}$ is the node corresponding to the stop, t^b_α is the time at which the bus is at the stop, $\overline{\mathcal{C}}^b_\alpha$ is the set of children picked-up by bus b at stop α and $\underline{\mathcal{C}}^b_\alpha$ is the set of children dropped by the bus at the stop. Note that for each stop either $\overline{\mathcal{C}}^b_\alpha$ or $\underline{\mathcal{C}}^b_\alpha$ must be non empty. We denote the tour performed by bus b as $\mathcal{T}_b = \left(a^b_\alpha\right)_{\alpha=1}^{n_b}$. A bus tour schedule $\mathcal{P}$ is therefore defined by a set $\{\mathcal{T}_b\}_{b=1}^{B}$ of bus tours, and $\bar{\mathcal{P}}$ is the set of all possible bus tour schedules.

We assume, for safety reasons, that each child travels on a single bus, i.e. no bus transfer is allowed. Consequently, for each bus b and each stop $(v^b_\alpha, t^b_\alpha, \overline{\mathcal{C}}^b_\alpha, \underline{\mathcal{C}}^b_\alpha)$, we impose that children get on bus b only at home, and get off only at school, that is

$$v^b_\alpha = h(c), \quad \forall c \in \overline{\mathcal{C}}^b_\alpha, \tag{2}$$

and

$$v^b_\alpha = s(c), \quad \forall c \in \underline{\mathcal{C}}^b_\alpha. \tag{3}$$

The bus carrying child c in tour schedule $\mathcal{P}$ is denoted by $\beta^{\mathcal{P}}(c)$.

3.2 Performance measures

Ideally, with one bus per child, each child could be picked-up from home and driven directly to school, this is the *taxi solution*. Due to the limited number of buses, each child may experience a delay due to the fact that the bus must pick-up other children on its way to the school. For each child, we define the *delay* as the difference between its actual journey time and the shortest possible time between home and school. Also, as buses may have to perform several tours, some children will be dropped at school early, before the actual class starting time. The time spent by the child waiting at school for the class to begin is referred to as the *waiting time*. The sum of the delay and the waiting time is called the *time loss* of the child. We will consider here two related performance measures for the bus tour schedules, leading to two different objective functions to be minimized. The first one is *total* time loss summed over all children or equivalently, the mean time loss over all children. The second measure is the *maximum* time loss over all children. Minimizing this measure aims at balancing time losses.

For stop a_α^b and child c, we define

$$\overline{\delta}_{c\alpha}^b = \begin{cases} 1 & \text{if bus } b \text{ picks child } c \text{ during stop } a_\alpha^b, \text{ i.e. } c \in \overline{\mathcal{C}}_\alpha^b \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

and

$$\underline{\delta}_{c\alpha}^b = \begin{cases} 1 & \text{if bus } b \text{ drops child } c \text{ during stop } a_\alpha^b, \text{ i.e. } c \in \underline{\mathcal{C}}_\alpha^b \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The departure time σ_c of child c is given by

$$\sigma_c = \sum_{b=1}^{B} \sum_{\alpha=1}^{n_b} \overline{\delta}_{c\alpha}^b t_\alpha^b. \tag{6}$$

Note that only one term of the sum will be nonzero, since each child performs exactly one trip. Similarly, the arrival time τ_c of child c is defined as

$$\tau_c = \sum_{b=1}^{B} \sum_{\alpha=1}^{n_b} \underline{\delta}_{c\alpha}^b t_\alpha^b. \tag{7}$$

Therefore, the time loss $f(c)$ of child c is given by

$$f(c) \;\; = \;\; \left(o_{s(c)} - \sigma_c\right) - d\left(h(c), s(c)\right), \tag{8}$$

where $o_{s(c)}$ is the starting time of school $s(c)$. Equation (8) is the difference between actual travel plus waiting time and the shortest possible travel time. Finally, we get for the first objective function $\mathcal{F}_1$, i.e. total time loss, to be minimized over all tour schedules $\bar{\mathcal{P}}$:

$$\mathcal{F}_1 = \sum_{c \in \mathcal{C}} f(c)\, q(c), \tag{9}$$

and the second objective function $\mathcal{F}_2$, i.e. maximum time loss, to be minimized over all tour schedules $\bar{\mathcal{P}}$:

$$\mathcal{F}_2 = \max_{c \in \mathcal{C}} f(c). \tag{10}$$

3.3 Constraints

The following constraints must be verified in the problem. First, each child should arrive at school on time, that is

$$\tau_c \leq o_{s(c)}, \quad \forall c \in \mathcal{C}. \tag{11}$$

Second, we impose that each child is carried by exactly one bus, that is

$$\sum_{b=1}^{B} \sum_{\alpha=1}^{n_b} \overline{\delta}_{c\alpha}^{b} = 1, \quad \forall c \in \mathcal{C}, \tag{12}$$

and

$$\sum_{b=1}^{B} \sum_{\alpha=1}^{n_b} \underline{\delta}_{c\alpha}^{b} = 1, \quad \forall c \in \mathcal{C}, \tag{13}$$

and that it steps out of the same bus it boarded, that is

$$\sum_{\alpha=1}^{n_b} \overline{\delta}_{c\alpha}^{b} - \underline{\delta}_{c\alpha}^{b} = 0 \quad \forall b \in \mathcal{B}, \ \ \forall c \in \mathcal{C}. \tag{14}$$

Also, departure time from home must obviously precede arrival time at school, that is

$$\sigma_c \leq \tau_c, \quad \forall c \in \mathcal{C}. \tag{15}$$

Then, no bus can carry more children than its capacity allows for. The occupancy of bus b at stop a_α^b is given by

$$\text{occ}(b, \alpha) = \sum_{k=1}^{\alpha} \sum_{c=1}^{|\mathcal{C}|} q(c) \left[\overline{\delta}_{ck}^{b} - \underline{\delta}_{ck}^{b} \right]. \tag{16}$$

Consequently, we have for each bus b the following set of constraints.

$$\text{occ}(b, \alpha) \leq Q_b, \quad \forall \alpha = 1, \ldots, n_b. \tag{17}$$

For each tour, the schedule must be such that the bus can complete the trip within time, that is

$$t_{\alpha+1}^{b} - t_{\alpha}^{b} \geq d(v_{\alpha}^{b}, v_{\alpha+1}^{b}). \tag{18}$$

Note that the optimization problem defined above is a nonlinear mixed integer programming problem, with decision variables v_α^b, t_α^b, $\overline{\delta}_{c\alpha}^{b}$ and $\underline{\delta}_{c\alpha}^{b}$. The nonlinearity is due to (4) and (5). It can be transformed into a linear integer program using standard techniques, at the expense of an increased number of variables and constraints.

4 Construction of a feasible solution

We describe here a heuristic to construct a feasible solution. The idea is to first ignore the number of available buses B, and use as many of them as needed to verify all other constraints. We denote by Q the capacity of these virtual buses such that $Q = \min_{b \in \mathcal{B}} Q_b$. Schools are considered in increasing order of their starting times. For each school s, the associated home nodes in

$\{\mathcal{U}_b\}_{b=1}^{B^+}$ =**BUILDCHAINS**

Init $b = 0$

School loop For each $s \in \mathcal{S}$

Init $\hat{\mathcal{H}} = \texttt{sort}\,(\mathcal{H}(s))$

Home loop While $\hat{\mathcal{H}} \neq \varnothing$,

Create bus $b = b + 1$, $\hat{s}(b) = s$, $\text{occ}_b = 0$, $\mathcal{U}_b = \varnothing$

Candidates loop For each $h \in \hat{\mathcal{H}}$

Check capacity Let $c \in \mathcal{C}$ such that $h(c) = h$ and $s(c) = s$. If $(\text{occ}_b + q(c)) \leq Q$ then

Insert $\mathcal{U}_b = \texttt{INSERTNODE}(\mathcal{U}_b, h)$

Update $\hat{\mathcal{H}} = \hat{\mathcal{H}} \setminus h$, $\text{occ}_b = \text{occ}_b + q(c)$

Figure 1: Building chains for initial tours with many buses

$\mathcal{U}$ =**INSERTNODE(**$\mathcal{U}$,v^***)** with $\mathcal{U} = \{v_1, \ldots, v_{p-1}\}$

Empty chain If $\mathcal{U} = \varnothing$ then `return` $\{v^*\}$

Insert before $\mathcal{U}_0 = \{v^*, v_1, \ldots, v_{p-1}\}$

Positions loop For $i = 1, \ldots, p-1$

Insert after $\mathcal{U}_i = \{v_1, \ldots, v_i, v^*, v_{i+1}, \ldots, v_{p-1}\}$

Shortest chain `return` $\mathcal{U}_i$ such that $i = \text{argmin}_{j=0,\ldots,p} L(\mathcal{U}_j)$

Figure 2: Node insertion in chain

$\mathcal{H}(s)$ are sorted in decreasing order of their distance from s. The procedure BUILDCHAINS, described in Figure 1, builds chains $\mathcal{U}_b = \{v_1^b, \ldots, v_{p-1}^b\}$ of length $L(\mathcal{U}_b) = \sum_{i=1}^{p-2} d(v_i^b, v_{i+1}^b)$, describing the sequence of houses visited by "bus" b. Each virtual bus b serves a unique school $\hat{s}(b)$.

The insertion of a node in the chain is based on a minimal chain length increase principle, as described in Figure 2.

Finally, for each virtual bus b, we insert the node corresponding to school $\hat{s}(b)$ in the chain. As both chain traversals are possible, we consider inserting its school at either one of its extremities, preferring that resulting in the shortest chain. If needed, the chain is reversed such that the school appears last ($v_p^b = \hat{s}(b)$). Finally, we define the tour $\mathcal{T}_b$ associated with the chain $\mathcal{U}_b = \{v_1^b, \ldots, v_p^b\}$ as $\left(a_\alpha^b\right)_{\alpha=1}^{p}$, with a_α^b defined by (1) and

- $t_p^b = o_{\hat{s}(b)}$ (the bus is scheduled to arrive exactly when class starts),

- $t^b_\alpha = t^b_{\alpha+1} - d(v^b_\alpha, v^b_{\alpha+1})$, $\alpha = p-1, \ldots, 1$ (the bus uses the shortest route),
- $\overline{\mathcal{C}}^b_\alpha = \{c(v^b_\alpha, \hat{s}(b))\}$, $\alpha = 1, \ldots, p-1$ (only children going to school $\hat{s}(b)$ step in the bus),
- $\overline{\mathcal{C}}^b_p = \varnothing$ (no pick-up at school),
- $\underline{\mathcal{C}}^b_\alpha = \varnothing$, $\alpha = 1, \ldots, p-1$ (no drop off at home),
- $\underline{\mathcal{C}}^b_p = \cup_{\alpha=1}^{p-1} \overline{\mathcal{C}}^b_\alpha$ (all children that were picked-up are dropped off).

If the solution obtained by applying this procedure is composed of B^+ buses, it is feasible only if $B^+ \leq B$, where B is the number of buses actually available. If it is infeasible, tours are merged pairwise $(B^+ - B)$ times to obtain exactly B buses. Merging tour $\mathcal{T}_k = \left(a^k_\alpha\right)_{\alpha=1}^{n_k}$ performed by bus k with tour $\mathcal{T}_j$ results in a new tour $\mathcal{T}^+$, as described in Figure 4. The impact of merging tours $\mathcal{T}_j$ and $\mathcal{T}_k$ is measured by w_{jk}, defined as the starting time of $\mathcal{T}_k$ minus the arrival time of tour $\mathcal{T}_j$ plus the minimum travel time between last node of $\mathcal{T}_j$ and first node of $\mathcal{T}_k$. If w_{jk} is positive, it means that the starting time of the bus j is not affected by the merge. In that case, w_{jk} represents the waiting time between the two merged tours (see Figure 3(a)). If w_{jk} is negative, it means that bus j had to advance its departure time to be able to perform the merged tour on time. In that case, $-w_{jk}$ represents the amount of the shift (see Figure 3(b)). We assume that tour $\mathcal{T}_k$ is an initial "artificial" tour as built by the procedure described in Figure 1, while tour $\mathcal{T}_j$ already corresponds to a "physical" bus. Consequently, the prior objective in merging is to avoid modifying starting time of bus j (and subsidiary to minimize the driver's waiting time between two chains). If this is not possible, we prefer the merger that minimizes the shift in departure time. Considering a chain j, if there is at least one non-negative w_{jk}, chain j is merged to the tour k^* such that

$$k^* = \underset{k=1,\ldots,B,\ w_{jk} \geq 0}{\operatorname{argmin}} w_{jk},\ k^* \neq j. \tag{19}$$

Otherwise, it is merged to the tour k^* such that

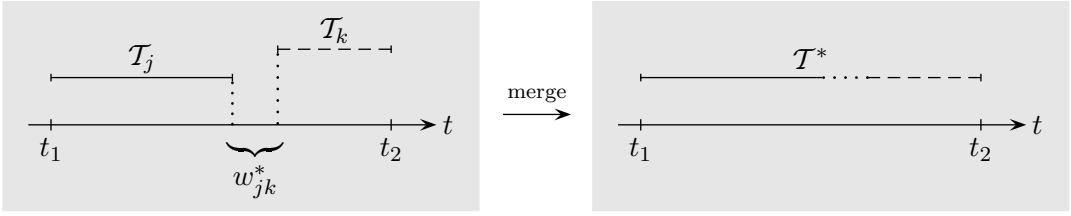
$$k^* = \underset{k=1,\ldots,B,\ w_{jk} < 0}{\operatorname{argmax}} w_{jk},\ k^* \neq j. \tag{20}$$

At each merger, the pair (j,k) of tours chosen minimizes w_{jk}, if there exists $w_{jk} \geq 0$, or maximizes w_{jk} (if $w_{jk} < 0$) for all tours $1 \leq j \neq k \leq B^+$. Ties are broken arbitrarily.

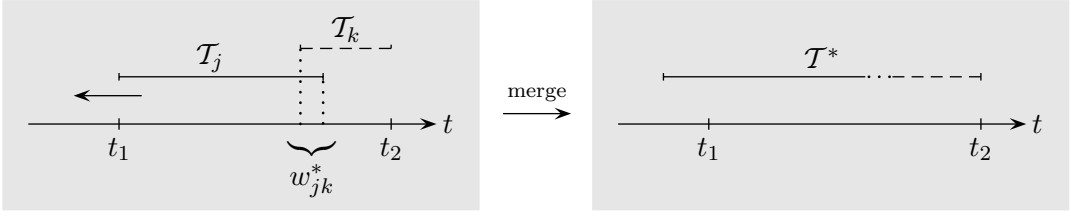
5 Algorithmic improvement

We now describe an approach to improve an initial solution found, e.g. by the above procedure, i.e. we try to achieve a better value of the objective function ($\mathcal{F}_1$ or $\mathcal{F}_2$) while keeping feasibility. This is the optimization consisting in minimizing objective function (9) or (10) under the constraints (2) to (8) and (11) to (18). This is a nonlinear integer program.

A formulation of the problem as an integer linear program can be obtained using standard techniques (see Watters, 1967). Unfortunately, its size becomes too large in terms of variables and constraints for an exact resolution with current software and computers. Instead, we have implemented and compared several local search heuristics to handle the improvement step: simulated annealing, feasible and infeasible version, described in Section 5.2 and tabu search, described in Section 5.3. The performance of these methods is analyzed and discussed in Section 6.



(a) $w_{jk} \geq 0$



(b) $w_{jk} < 0$

Figure 3: Two possibilities to merge tour $\mathcal{T}_k$ with tour $\mathcal{T}_j$. w^*_{jk} is w_{jk} minus the minimum travel time between the last node of $\mathcal{T}_j$ and the first node of $\mathcal{T}_k$.

$\mathcal{T}^+ =$ **MERGE(**$\mathcal{T}_j, \mathcal{T}_k$**)** with $\mathcal{T}_j = (a^j_\alpha)_{\alpha=1}^{n_j}$ and $\mathcal{T}_k = \left(a^k_\alpha\right)_{\alpha=1}^{n_k}$

Shift $w_{jk} = t^k_1 - t^j_{n_j}$

Delay first tour If $w_{jk} < 0$, then $t^j_\alpha = t^j_\alpha + w_{jk}$ for each $\alpha = 1, \ldots, n_j$

Merge $\mathcal{T}^+ = (a^+_\alpha)_{\alpha=1}^{n_+}$, where $n^+ = n_j + n_k$, $a^+_\alpha = a^j_\alpha$ for $\alpha = 1, \ldots, n_j$ and $a^+_\alpha = a^k_{\alpha - n_j}$ for $\alpha = n_j + 1, \ldots, n^+$.

Figure 4: Merging two tours

5.1 Local search

Local search is a standard approach in combinatorial optimization. It is based on a neighborhood structure with which the set of solutions is endowed resulting in a directed graph with the solutions as its node set and each node being linked to its neighbors by arcs of the graph. Applying a local search heuristic corresponds to searching the nodes in this graph according to some strategy. Here we will propose a neighborhood structure and two such heuristics. First we describe a variant of simulated annealing (see Kirkpatrick et al., 1983 and Rossier et al., 1986) with cyclic reheating and then simple tabu search (see Glover, 1977, Hansen, 1986 and Hansen and Jaumard, 1987).

5.2 Simulated annealing

Recall that simulated annealing can be viewed as a biased random walk on the neighborhood graph. It starts at a given solution $\mathcal{P}$, selects a neighbor solution $\mathcal{P}'$ at random and moves there with probability

$$p = \min\left\{1, \exp\left(-\frac{\mathcal{F}(\mathcal{P}') - \mathcal{F}(\mathcal{P})}{T}\right)\right\},$$

where $T > 0$ is a parameter called "temperature" and stays put with probability $(1 - p)$. The higher T the more likely it is that a candidate solution be it better or worse than the incumbent is accepted. The iterations start with a high temperature T_0 allowing the heuristic to escape from local optima. Then the temperature is gradually decreased (each μ iterations) to intensify the search. Once a given criterion is met, T is reheated to the initial temperature T_0. Thus T oscillates, successively decreasing and increasing the probability of accepting a candidate solution. This way phases of local improvement and diversification that escape of local optima alternate each other.

We consider here two neighborhoods: feasible neighborhood $\mathcal{N}_f$ generates only feasible solutions, while infeasible neighborhood $\mathcal{N}_i$ enlarges the set of candidate solutions and may generate solutions violating the capacity constraint (17). In both neighborhoods solutions are modified by removing one child from a bus and trying to place it in an other one. A child c, which is to be moved, is randomly selected with probability

$$\frac{f(c)^\lambda}{\sum_{c\in\mathcal{C}} f(c)^\lambda} \tag{21}$$

where $f(c)$ is the time loss of child c, defined by (8), and λ is a parameter such that $0 \leq \lambda \leq 1$. A value of $\lambda = 0$ means that we assign the same probability to all children. A bus b' is also randomly selected, with equal probabilities among the buses not carrying child c in the current solution. The feasible neighborhood $\mathcal{N}_f$ solution consists to insert child c in bus b' resulting in the largest improvement (or the least deterioration) of the solution, in sense of the objective function chosen $\mathcal{F}_f \in \{\mathcal{F}_1, \mathcal{F}_2\}$.

The other proposed neighborhood $\mathcal{N}_i$ modifies a solution in a similar way as $\mathcal{N}_f$, except that capacity satisfaction is not enforced when child c is inserted in bus b'. Therefore, $\mathcal{N}_i$ may create infeasible solutions. In order to restore feasibility, we augment the objective function value $\mathcal{F}_f$ by a penalty on the capacity violation:

$$\mathcal{F}_i = \mathcal{F}_f + \kappa \sum_{b=1}^{B} \max\{0, \text{occ}^b_{\max} - Q_b\} \tag{22}$$

where $\kappa > 1$ is the penalty parameter and

$$\text{occ}^b_{\max} = \max_{\alpha=1,\ldots n_b} \text{occ}(b, \alpha). \tag{23}$$

The value of the penalty parameter is first initialized to $\kappa_0 > 1$. Then, each time an infeasible solution is produced, κ is multiplied by κ_0. When a feasible solution is generated, the value of κ is reset to κ_0.

The general description of this heuristic is shown in Figure 5. The input parameters are: the initial temperature T_0 to start the heuristic, ω the decrease coefficient of the temperature, the

number of consecutive iterations μ with the same temperature, a penalty coefficient κ_0 of objective function $\mathcal{F}_i$, the maximum number γ_{max} of consecutive iterations during which the current solution may remain without any change (i.e. simulated annealing stagnation), and the maximal number η_{max} of consecutive iterations without improvement of the record solution, that is the best solution met. The heuristic is stopped when η_{max} is reached. Figure 5 captures the two versions of our simulated annealing heuristic, one for each type of neighborhood (i.e. $\mathcal{N} \in \{\mathcal{N}_f, \mathcal{N}_i\}$). Results on testing these heuristics are described in Section 6.

5.3 Tabu search

Tabu search is an alternate local neighborhood search method. Rather than performing a random walk, as in simulated annealing, in tabu search the solution space is fathomed by moving from a given solution $\mathcal{P}$ to its best neighbor $\mathcal{P}'$ (which sometimes can be worse than $\mathcal{P}$) in the neighborhood structure. Note that, for the sake of efficiency, we do not necessarily consider the entire neighborhood of $\mathcal{P}$ to determine $\mathcal{P}'$, but only a subset of a given size denoted by ν. Care has to be taken to prevent cycling and this is achieved by keeping and updating a fixed length list of forbidden moves, the tabu list, thus avoiding to return to the same solution after a small number of iterations.

Here we consider the feasible neighborhood N_f (same as described for simulated annealing) which consists in removing one child c, randomly selected with probability (21), from its bus b and trying to place it in an other randomly selected one $b' \neq b$, respecting all constraints of the problem. We say that "child c moves from b to b'".This is done repeatedly to find a path to improve the solution.

To avoid cycling, the following moves are stored in the tabu list: if a child is moved from bus b to bus b', the opposite move from bus b' to bus b is stored. The tabu list moves are forbidden for ζ iterations. Note that if a forbidden move gives a neighbor solution better than the best solution found so far, this move is carried out in spite of its tabu status.

The heuristic is stopped once a maximal number of neighbor solutions ρ_{tot} has been visited, i.e. a maximum total iteration number. Tabu search is described in Figure 6. Test results for this heuristic are described in Section 6.

6 Case studies

In order to validate the heuristic approaches, these were tested on both real world and synthetic data sets.

The real data comes from two Swiss towns, Savigny and Forel. The network is represented in Figure 7. It contains 34 nodes, including 12 schools: 4 *kindergartens* starting at 8.35, 6 *primary schools* starting at 8.15 and 2 *high schools* starting at 7.40. The maximum travel time between two nodes is 18 minutes. For the school year 1997–1998, four buses were used to transport 274 children (79 for kindergarten, 162 for primary school and 33 for high school). The capacity of each bus is 210 units, and each high school child takes up 10 units in the bus, while kindergarten

$\{\mathcal{T}_b^*\}_{b=1}^B$ =SIMULATEDANNEALING ($\{\mathcal{T}_b\}_{b=1}^B$, T_0, ω, μ, κ_0, γ_{max}, η_{max})

Init $T = T_0$, $\eta = 0$, $\gamma = 0$, $\kappa = \kappa_0$, $n = 0$, $\mathcal{P} = \{\mathcal{T}_b\}_{b=1}^B$, $\mathcal{P}^* = \mathcal{P}$

Loop While $\eta < \eta_{max}$ do

Update iterations total number $n = n + 1$

Choose neighbor solution in $\mathcal{N}$ Choose randomly a child c and a bus $b' \neq \beta^{\mathcal{P}}(c)$,

Insert c in $\mathcal{T}_{b'} \rightarrow \mathcal{P}' = \{\mathcal{T}_b'\}_{b=1}^B$ a solution.

Penalty If $\mathcal{P}'$ is infeasible then $\kappa = \kappa \cdot \kappa_0$ else $\kappa = \kappa_0$

Acceptance solution Generate uniformly $x \sim U[0, 1)$

Test 1 If $\mathcal{F}_i(\mathcal{P}') < \mathcal{F}_i(\mathcal{P})$ then

Update $\mathcal{P} = \mathcal{P}'$, $\gamma = 0$

Test 2 Else if $\exp\left(-\dfrac{\mathcal{F}_i(\mathcal{P}') - \mathcal{F}_i(\mathcal{P})}{T}\right) > x$ then

Update $\mathcal{P} = \mathcal{P}'$, $\gamma = 0$

Else $\gamma = \gamma + 1$

Test 3 If $\mathcal{F}_i(\mathcal{P}') < \mathcal{F}_i(\mathcal{P}^*)$ then

Update $\mathcal{P}^* = \mathcal{P}'$, $\eta = 0$

Else $\eta = \eta + 1$

Update If $(n \mod \mu = 0)$ then $T = \omega \cdot T$

If $(\gamma > \gamma_{max})$ then $T = T_0$, $\gamma = 0$

Best Solution return $\{\mathcal{T}_b^*\}_{b=1}^B = \mathcal{P}^*$

Figure 5: Simulated annealing

$\{\mathcal{T}_b^*\}_{b=1}^B$ =**TABUSEARCH (**$\{\mathcal{T}_b\}_{b=1}^B$**,** ν**,** ζ**,** ρ_{tot}**)**

Init $\rho = 0$, $\mathcal{L} = \varnothing$, $\mathcal{P} = \{\mathcal{T}_k\}_{k=1}^U$, $\mathcal{P}^* = \mathcal{P}$

Loop While $\rho < \rho_{tot}$ do

Neighbor solution of $\mathcal{P}$ in $\mathcal{N}_f$ For $i = 1, \ldots, \nu$

Choose randomly a child c_i and a bus $b_i' \neq \beta^{\mathcal{P}}(c_i)$.

Insert c_i in $\mathcal{T}_{b_i'} \rightarrow \mathcal{P}_i' = \{\mathcal{T}_b'\}_{b=1}^B$ a feasible solution.

Best neighbor $\mathcal{P}' = \min_{1 \leq i \leq \nu}\{\mathcal{F}_f(\mathcal{P}_i')\}$

Best solution If $\mathcal{F}_f(\mathcal{P}') < \mathcal{F}_f(\mathcal{P}^*)$ then $\mathcal{P}^* = \mathcal{P}'$, $\mathcal{P} = \mathcal{P}'$

Next solution If $(c, \beta^{\mathcal{P}}(c)) \notin \mathcal{L}$ then

Update solution $\mathcal{P} = \mathcal{P}'$

Update tabu list $\mathcal{L} = \mathcal{L} \cup \{(c, \beta^{\mathcal{P}}(c))\}$

If $|\mathcal{L}| > \zeta$ then delete oldest element of $\mathcal{L}$

Update counter $\rho = \rho + 1$

Best solution `return` $\{\mathcal{T}_b^*\}_{b=1}^B = \mathcal{P}^*$

Figure 6: Tabu Search

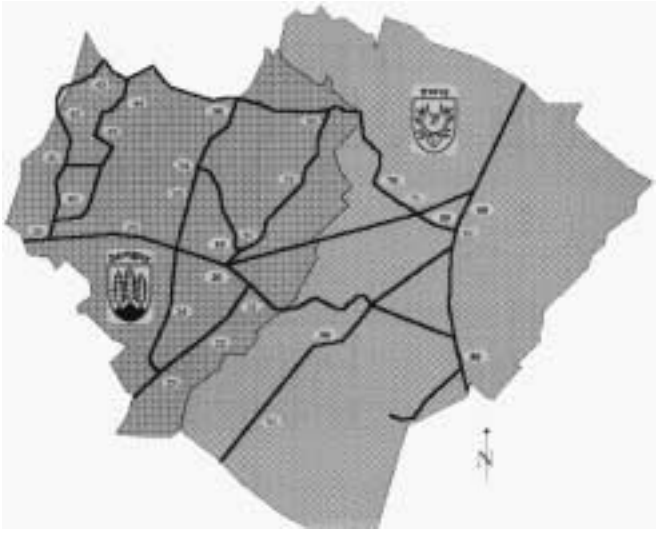


Figure 7: Savigny and Forel's network

and primary school children take up 7 units.

The synthetic data has been generated in order to analyze the relative efficiency of the methods as a function of the network size. For a given number of nodes randomly distributed in a square, the network is obtained from a Delaunay triangulation, from which links are removed such as to obtain a realistic degree for each node, while maintaining the connectivity. The distributions of homes, schools and trips have been chosen to resemble the proportions in the real data set.

The experimental design is based on four variables: the network, the number of buses, the objective function and the heuristic. In addition to the Savigny–Forel network, we have generated four artificial networks with 10, 20, 30 and 50 nodes, respectively. For each of the five networks, we have run several scenarios for the number of buses (see Table 1), and the two objective functions (9) and (10), to obtain a total of thirty problems. We have considered three types of heuristics, that is tabu search and simulated annealing with feasible neighborhood and with infeasible neighborhood. In each case, simulated annealing has been run using four different initial temperatures, and five different seeds for the random number generator. The tabu search algorithm has been run using two different neighborhood sizes (14 and 18 neighbor solutions visited, resp.), and ten different seeds for the random number generator. Thus, twenty instances of each type of heuristic, totaling 1800 runs were carried out on a Pentium III 499MHz. On this machine an instance with 50 nodes and 9 buses took about 10 hours to complete 50'000 solution evaluations, that is about 0.7 sec. per evaluation. We summarize now the main observations from these runs.

Clearly, the quality of the solution depends on the specific heuristics used and the choice of its parameters. In order to compare the overall quality of the heuristics, we adopt a variant of the performance profiles analysis proposed by Dolan and Moré (2002).

If $f_{p,a,i}$ is the performance index of instance i of algorithm a solving problem p, then the *performance ratio* is defined by

$$r_{p,a,i} = \frac{f_{p,a,i}}{\min_{a,i}\{f_{p,a,i}\}}. \tag{24}$$

Network	# buses
Savigny-Forel	4 5 6
Artificial $\|\mathcal{V}\| = 10$	2 3 4 5
Artificial $\|\mathcal{V}\| = 20$	2 3 4 5
Artificial $\|\mathcal{V}\| = 30$	3 6
Artificial $\|\mathcal{V}\| = 50$	9 13

Table 1: Number of buses considered for each network

For any given threshold π, the overall performance of algorithm a is given by

$$\rho_a(\pi) = \frac{1}{n(a)n_p}\Phi_a(\pi) \tag{25}$$

where n_p is the number of problems considered, $n(a)$ is the number of instances of algorithm a which have been run, and $\Phi_a(\pi)$ is the number of problems and instances of algorithm a for which $r_{p,a,i} \leq \pi$. We refer the reader to Dolan and Moré (2002) for more details about the benchmarking method.

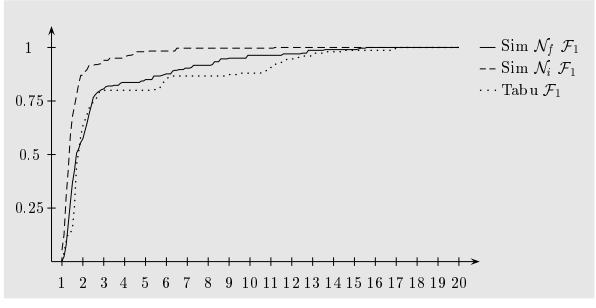
The performance indices we have selected are the relative improvements of the objective function (that is the ratio between the final and initial values of the objective function) after a given number of candidate evaluations (the *computation budget*). Interestingly, the profiles are qualitatively similar for different values of the computation budget. These results are encouraging, as they indicate that the proposed approach is rather robust and does not depend too heavily on the tuning of the heuristics parameters. We present here the results for two indices based on a fixed budget (100 and 500, resp.) and an index where the budget depends on the problem size ($100|\mathcal{V}|$ evaluations).

We compare tabu search, and the two versions of simulated annealing for the problem with objective function $\mathcal{F}_1$ defined by (9) and $\mathcal{F}_2$ defined by (10), for a fixed budget of 100 evaluations (Figure 8), 500 evaluations (Figure 9) and $100|\mathcal{V}|$ evaluations (Figure 10). In each figure, the plots in the bottom are just zooms of the top plots, in order to emphasize the profile for small values of π. For the sake of clarity, the plots for $\mathcal{F}_1$ and $\mathcal{F}_2$ have been separated, although the performance profile analysis has been performed over all methods.

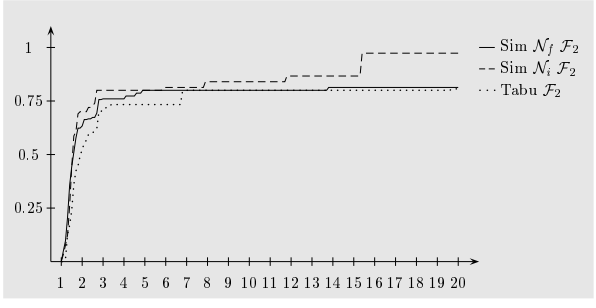
The most noticeable performance is achieved by the infeasible version of simulated annealing when objective function $\mathcal{F}_1$ is considered (see, Figures 8(c), 9(c) and 10(c)).

The slight superiority of simulated annealing over tabu search has been observed for other problems, such as task scheduling (Glardon et al., 1992) and sequencing (Kurbel, 1998). However, this may not be generalized, as the contrary has also been observed (see, for instance, Hao and Pannier, 1998, for constraint solving problems).

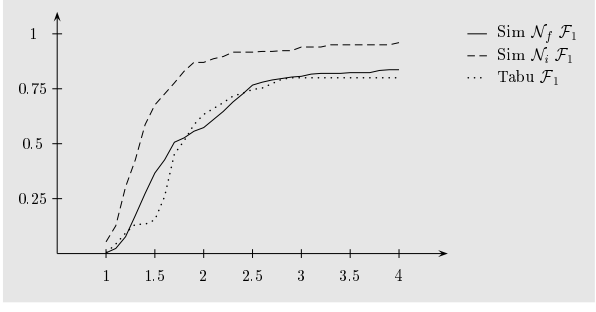
It is interesting to note the slope of the profiles for each objective function (compare, for instance, Figures 10(c) and 10(d)). The profile is steeper with objective function $\mathcal{F}_1$, illustrating a



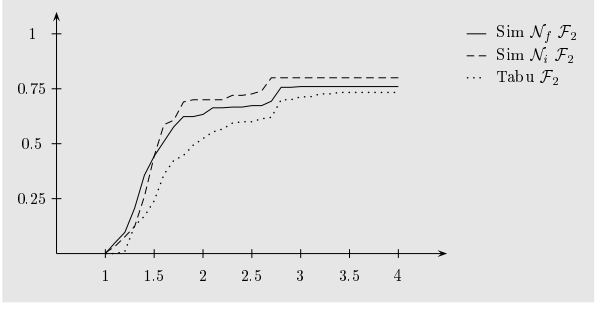
(a) $\mathcal{F}_1$



(b) $\mathcal{F}_2$

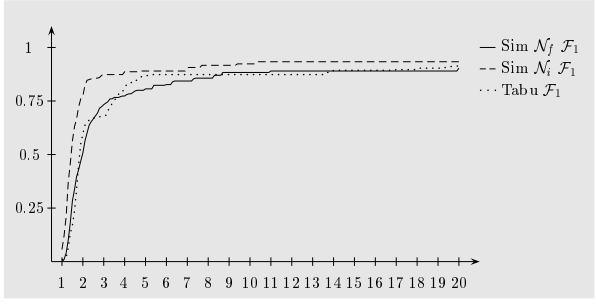


(c) $\mathcal{F}_1$

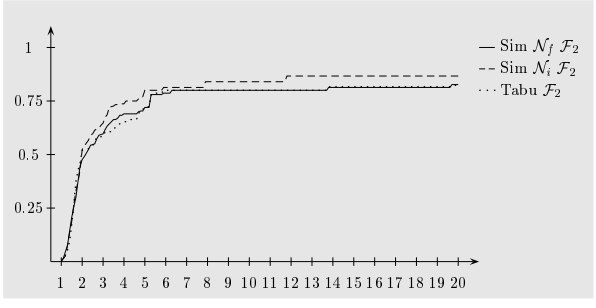


(d) $\mathcal{F}_2$

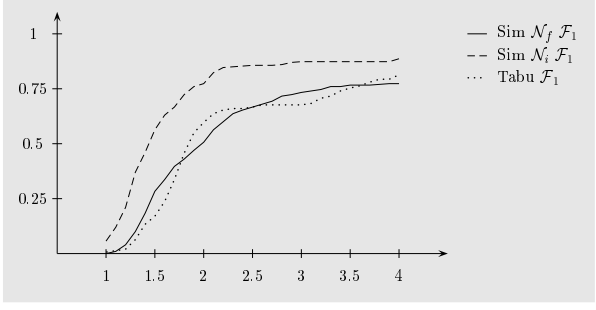
Figure 8: Performance profiles for budget=100



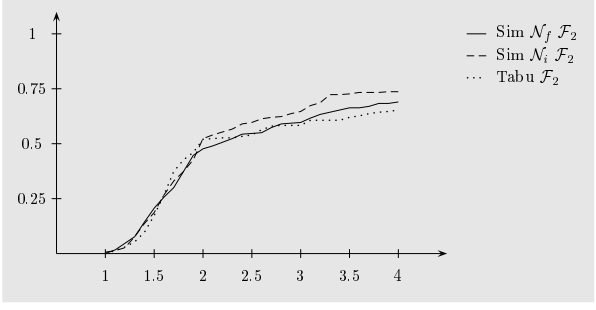
(a) $\mathcal{F}_1$



(b) $\mathcal{F}_2$

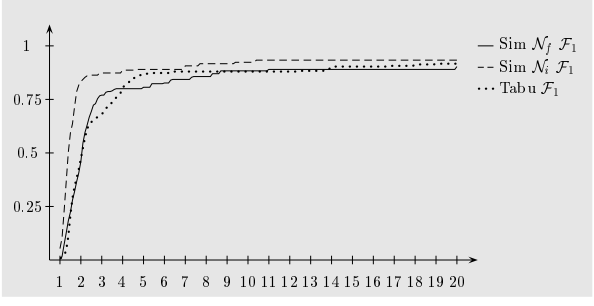


(c) $\mathcal{F}_1$

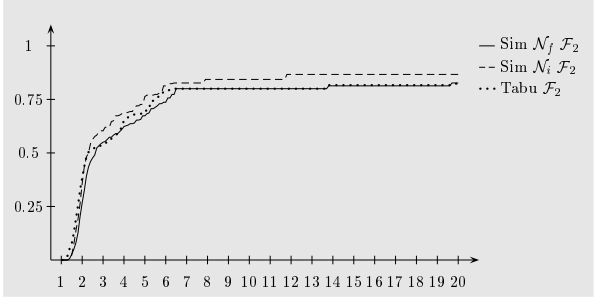


(d) $\mathcal{F}_2$

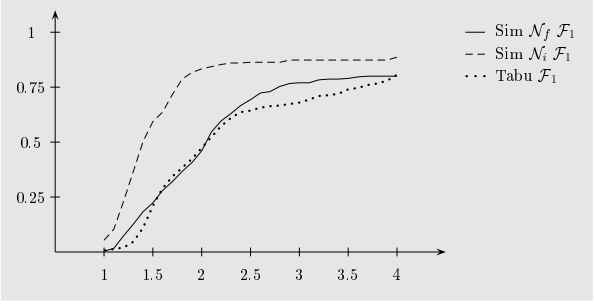
Figure 9: Performance profiles for budget=500



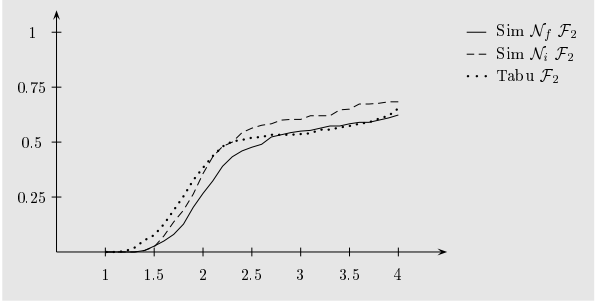
(a) $\mathcal{F}_1$



(b) $\mathcal{F}_2$



(c) $\mathcal{F}_1$



(d) $\mathcal{F}_2$

Figure 10: Performance profiles for budget=$100|\mathcal{V}|$

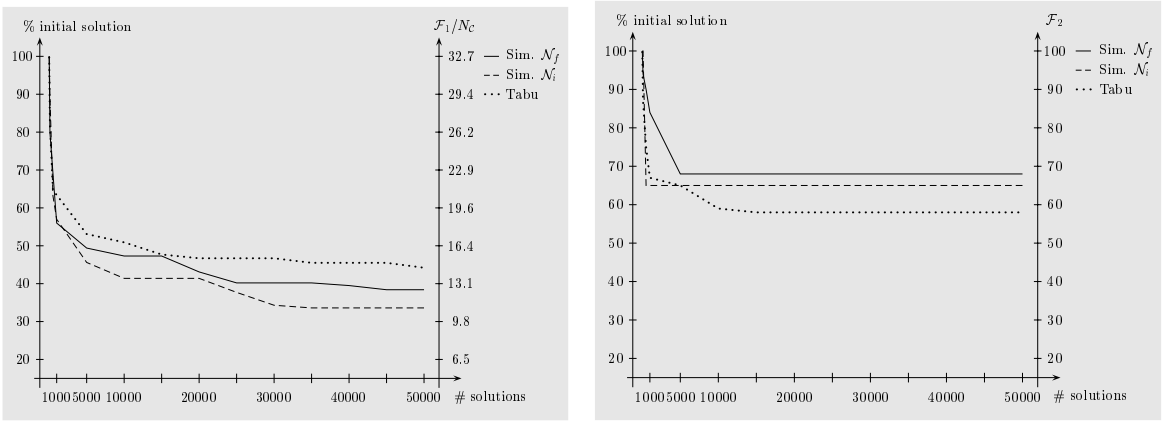


Figure 11: Heuristic improvements with $|\mathcal{V}| = 50$ and $B = 9$

more robust behavior of the heuristics when $\mathcal{F}_1$ is preferred over $\mathcal{F}_2$. This is due to the fact that the same value of the objective (10) may be achieved for a large number of different solutions, as only the largest time loss is considered. Therefore, the heuristics are unable to discriminate among a large class of solutions with the same objective, and are more likely to cycle randomly among them.

All heuristics exhibit the same type of behavior: a significant improvement of the objective function in the early iterations, and a slower reduction rate in subsequent iterations. This is illustrated by Figure 11 where the x-axis represents the number of evaluations of candidate solutions, and the y-axis is the ratio of the objective function for the solution computed by each heuristic and the objective function of the initial solution. Results for $\mathcal{F}_1/N_\mathcal{C}$ are represented on the left figure, and results for $\mathcal{F}_2$ on the right.

After comparing the behavior of various heuristics, we focus now on analyzing the quality of the produced solutions. The procedure described in Section 4 is designed to generate feasible solutions, without focusing on any objective function. Consequently, the quality of those solutions is pretty poor, requiring the subsequent use of the heuristics. We illustrate this with Table 2, where each column corresponds to an instance of a problem, the last three columns corresponding to the real case. The first row provides the number of nodes, the second the number of buses. The quantity $\mathcal{F}_1/N_\mathcal{C}$ is the mean of time losses for all children, and $\mathcal{F}_2$ is the maximum time loss among all children.

For the sake of comparison, we present the solutions obtained after 5'000 (Table 3) and 50'000 (Table 4) solution evaluations with the tabu search heuristics.

We compare now our better solutions found with the solution actually implemented by Savigny and Forel in 1997-1998, where the "no transfer" constraint (see (12), (13) and (14) in our model) is violated for three children. The significant improvement obtained by the heuristics over the actual solution is reported in Table 5. The distribution of time loss is reported in Figure 12, where the black bars represent the Savigny-Forel solution, the dark gray bars represent the solution obtained when minimizing $\mathcal{F}_1$ and the light gray for $\mathcal{F}_2$.

We conclude this result analysis by looking at the sensitivity of the solution with regards to the number of buses. Indeed, it is an important economical information, as it allows to assess the

	Synthetic						Real		
$\|\mathcal{V}\|$	20	20	20	20	50	50	34	34	34
B	2	3	4	5	9	13	4	5	6
$\mathcal{F}_1/N_\mathcal{C}$	46.5	28.7	18.2	14.1	32.8	28.5	41.8	33.9	27.7
$\mathcal{F}_2$	148	126	79	37	100	181	129	129	80

Table 2: Values of initial solutions, in minutes.

	Synthetic						Real		
$\|\mathcal{V}\|$	20	20	20	20	50	50	34	34	34
B	2	3	4	5	9	13	4	5	6
$\mathcal{F}_1/N_\mathcal{C}$	13.3	6.2	4.1	2.9	16.4	8.3	10.3	10.6	9.2
$\mathcal{F}_2$	37	26	25	22	59	46	47	44	36

Table 3: Values of final solutions (in minutes) after 5'000 evaluations.

	Synthetic						Real		
$\|\mathcal{V}\|$	20	20	20	20	50	50	34	34	34
B	2	3	4	5	9	13	4	5	6
$\mathcal{F}_1/N_\mathcal{C}$	6.8	4.5	2.4	1.1	11.8	6.1	8.9	6.4	5.5
$\mathcal{F}_2$	27	22	17	15	39	35	35	27	22

Table 4: Values of final solutions (in minutes) after 50'000 evaluations.

	Savigny and Forel	Optimization with $\mathcal{F}_1$	Optimization with $\mathcal{F}_2$
$\mathcal{F}_1/N_C$ = mean of time loss	16.9	9.2	13.3
$\mathcal{F}_2$ = max of time loss	42	40	30
Earliest departure	6.49	7.12	6.56
# children attaining the min of time loss	22	54	32
# children attaining the max of time loss	1	3	7

Table 5: Comparison of Savigny and Forel with heuristics best solutions.

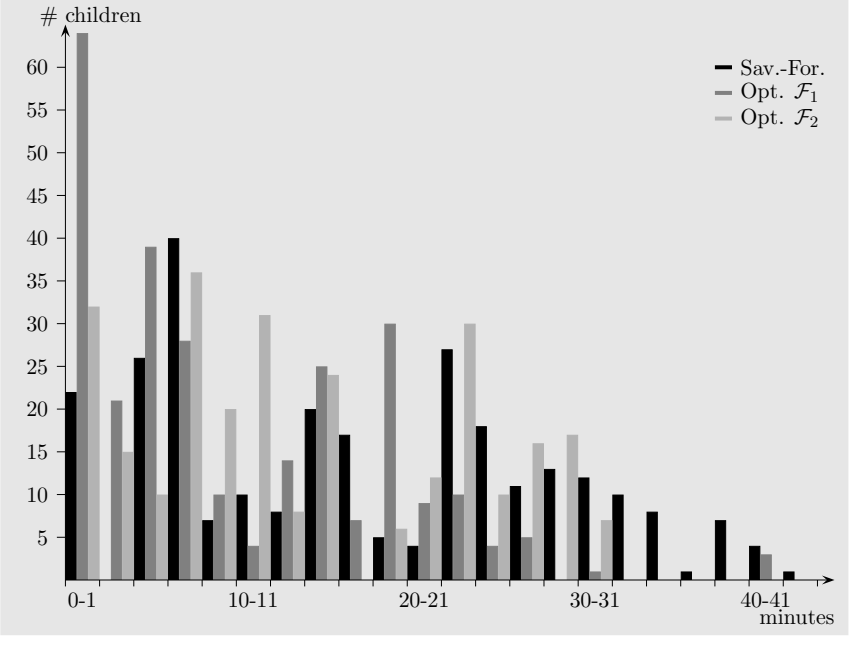
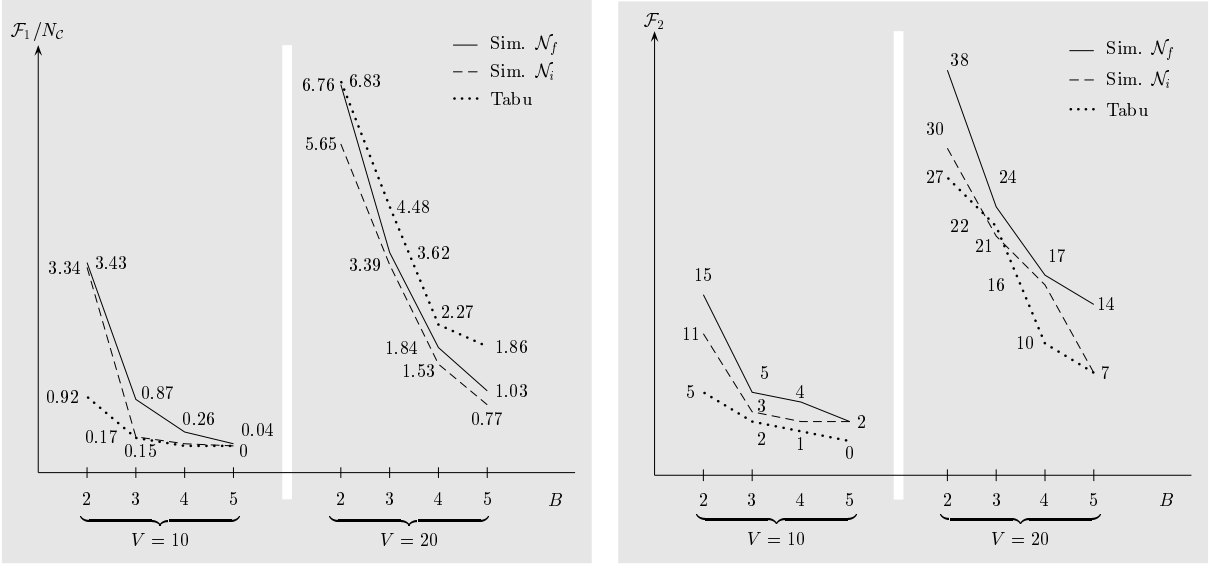


Figure 12: Time loss distribution of three characteristic solutions.



(a) $\mathcal{F}_1/N_C$ (b) $\mathcal{F}2$

Figure 13: Best results found for $|\mathcal{V}| = 10$ and $|\mathcal{V}| = 20$ with simulated annealing, and tabu search.

benefits in terms of level of service for the investment in operating an additional bus. Here, we have allowed each heuristic to examine 50000 candidates before stopping the iterations. For the tests we have performed, the marginal benefit of adding one bus in the system decreases with the number of buses, as expected. This is illustrated by Figure 13.

7 Empirical improvement

The solution obtained by the automatic procedure may not always be completely satisfactory from a practitioner point of view. Indeed, it is common that additional constraints not included in the model, have to be satisfied. Therefore, it is important for the practitioner to have a tool to visualize the proposed solution and possibly to be able to modify it. Moreover, the tool can be used to adjust the solution when the original conditions are modified (a bus is unavailable for maintenance, a school is closed, some children do not attend school, etc.) Indeed, a manual modification can be much faster than running the overall process from the beginning.

We describe here a decision-aid tool (implemented in Python, see Lutz and Ascher, 1999) designed to help practitioners improving the solution. It has two main functionalities: solution display and edition.

The display contains (i) a graph representing the network, where nodes corresponding to schools are differentiated and the path of a bus can be dynamically displayed (see Figure 14 top), (ii) a diagram representing the tour of a bus which consists of an x-axis for time and a y-axis for the

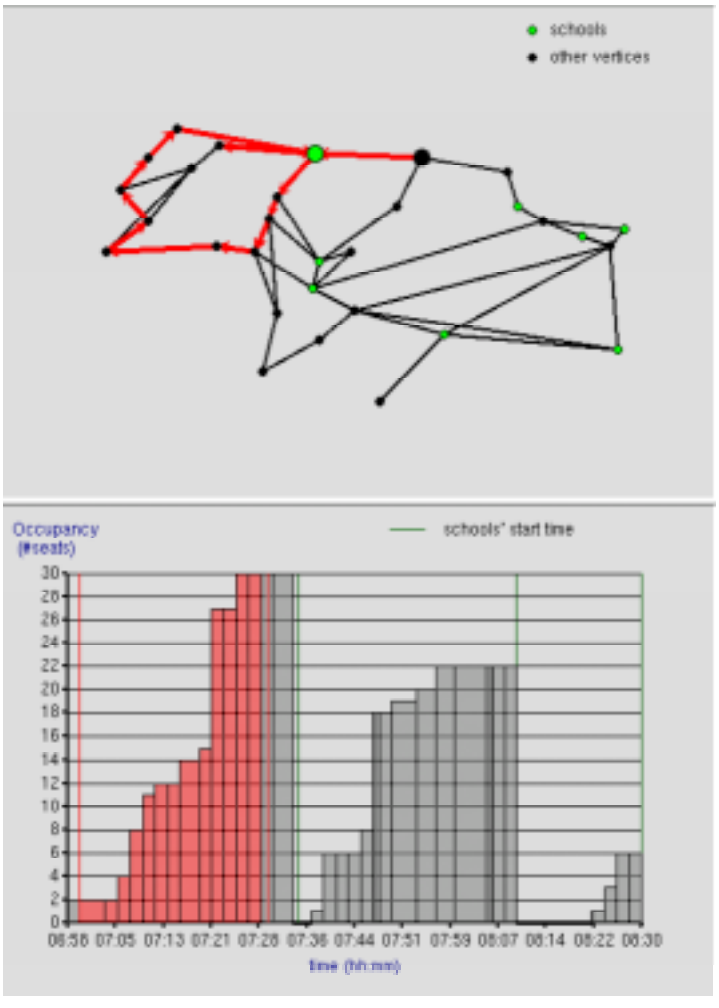


Figure 14: Evolution of a bus capacity and course bus.

occupancy of the bus (see Figure 14 bottom) and (iii) the description of each bus, its average, minimum and maximum occupancy, average, minimum and maximum travel times, waiting times and late arrival of children using it (see Figure 15). Finally, the exact occupancy of a bus at a given time is also available, for each child travel time and waiting time are reported.

The edition capabilities of the tool enable the user to modify the journey of one child, and to evaluate the impact of this modification on the solution. For the selected child c, the tool proposes within each bus b the best pick-up time for c, that is the pick-up minimizing time loss (defined by (8)) of all children carried by bus b. The impact of such a change on the objective function is reported, that is the total and the maximum time loss of all children ($\mathcal{F}_1$ and $\mathcal{F}_2$ defined by (9) and (10)). The user may then decide to implement one of the proposed modifications or keep the current solution.

Providing a child-oriented edition capability is motivated by the structure of the feasible solution described in Section 4. Indeed, the procedure mainly focuses on buses, irrespectively of the children's journey. Therefore, it is common for this procedure to generate solutions which are unsatisfactory for the children, like leaving very early in the morning, or spending a long time in the bus. The child oriented edition tool helps to fix such aberrations.

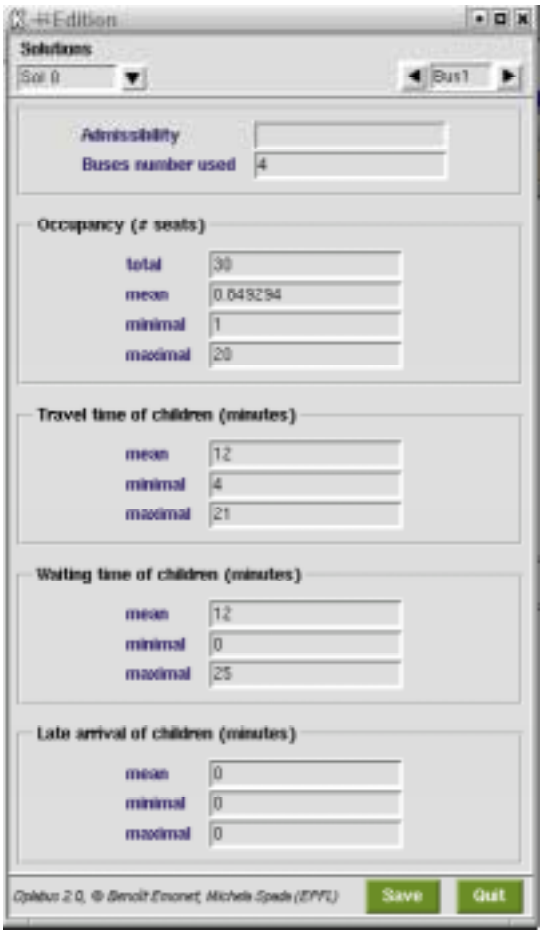


Figure 15: Solution characteristics for a given bus.

To illustrate the tool, we show how to further improve a solution found with tabu search. We consider the real problem with 4 buses. The optimum value of the solution given by tabu search minimizing objective function (10), is $\mathcal{F}_2 = 33$ minutes. The value of the other objective function (9) for the same solution is $\mathcal{F}_1/N_{\mathcal{C}} = 17.1$ minutes. We were able to improve the tabu solution as illustrated by Figure 16. We decide to move child 246, currently getting in bus 0 at 7.13 AM. The decision-aid tool proposes us to move it to take bus 1 at 7.18 AM. Departing 5 minutes later, the child's time loss $f(246)$ comes down from 20 to 15 minutes. This modification has no impact on the maximum time loss $\mathcal{F}_2$, because the maximum is reached for another child. The mean time loss $\mathcal{F}_1/N_{\mathcal{C}}$ decreases from 17.1 down to 15.9. The gain of 1.2 minutes is due to the fact that bus 0 can now start its tour at 7.13 AM instead of 6.58 AM, which has an impact on several children.

8 Conclusions and future developments

We have proposed a new modeling approach for the school bus routing and scheduling problem, and analyzed several heuristics to solve the problem. The model focuses on optimizing the service level of the bus system, as it aims at minimizing the children time losses either within the bus or at school before class starts. Contrarily to previous models proposed in the literature, we consider the number of buses as given, as it seems to be more consistent with real applications.

We have designed a procedure to generate a feasible solution of the problem. In general, that solution appears to be unrealistic, and not operational. Therefore, it is required to apply some heuristics in order to optimize it. We have compared two versions of simulated annealing (with feasible and infeasible neighborhoods), and tabu search. The infeasible version of simulated annealing appears to be the best performing heuristic, but the others are also performing well. Most importantly, it appeared from our tests that the behavior of all heuristics seems robust with respect to their design parameters. This is critical in a real context, where practitioner cannot afford to tune the parameters of a given heuristics in order to obtain a satisfactory solution. Also, it appeared that all heuristics perform better when the level of service is computed based on an average value, instead of an extreme value. Actually, the use of an extreme value is relevant from an application point of view, but requires an additional discrimination criteria to allow the heuristics to perform efficiently.

Finally, we have implemented a decision-aid tool designed to evaluate local improvements suggested by a human expert. This tool turns out to be effective not only for evaluation, but also to discover better solutions that heuristics may not have found.

Several improvements must be investigated in the future. Firstly, the model can be refined by adding further characteristics, such as allowing some transfers, allowing children with same origin and destination to use different buses, or weighting differently the time loss as a function of the children's age. Also, the optimization criteria can be extended, and combined. They may include the number of buses, or the total time spent by the drivers. In some cases, it is also conceivable to adjust the school starting time on the bus schedule. A multi-objective approach could then be considered, and filter methods would be good candidate approaches in this context. Secondly, the methods to identify the solutions can be improved. We strongly advocate a better cooperation between systematic heuristics and human-driven tools, in order to keep

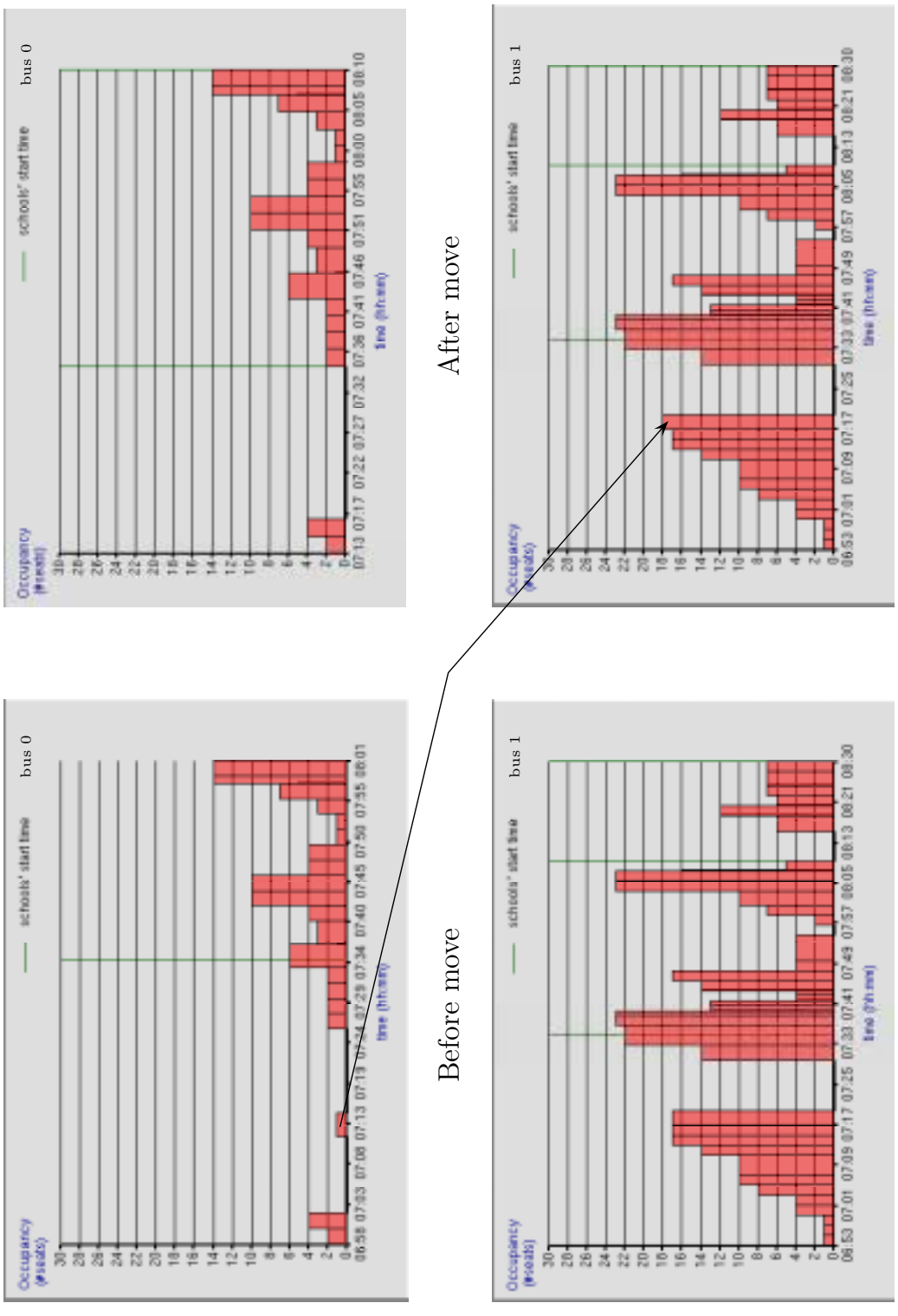


Figure 16: Moving a child from a bus 0 to bus 1

control on the solution generation. Also, procedures based on infeasible solutions deserve more investigation, as they would allow to skip the algorithm described in Section 4. Stress must then be put on feasibility restoration techniques, based for example on Lagrangian approaches. Finally, a strong collaboration with practitioners would be likely to initiate new ideas for efficient heuristics dedicated for this specific problem.

9 Acknowledgments

We would like to thank B. Emonet for implementation of the interactive graphical tool (see Emonet, 2000), R. Berger who helped us to develop the algorithms (see Berger, 2001), and the towns of Savigny and Forel who originated this study and provided the real data.

References

Angel, R., Caudle, W., Noonan, R. and Whinston, A. (1972). Computer-assisted school bus scheduling, *Management Science B* **18**: 279–288.

Bennett, B. and Gazis, D. (1972). School bus routing by computer, *Transportation Research* **6**: 317–326.

Berger, R. (2001). Planification de tournées de véhicules scolaires, *Travail pratique de diplôme*, EPFL, Switzerland.

Bodin, L. and Berman, L. (1979). Routing and scheduling of school buses by computer, *Transportation Science* **13**: 113–129.

Bodin, L., Golden, B., Assad, A. and Ball, M. (1983). Routing and scheduling of vehicles and crews. the state of the art, *Computers and Operations Research* **10**: 63–211.

Braca, J., Bramel, J., Poser, B. and Simchi-Levi, D. (1994). A computerized approach to the new york city school bus routing problem, *Technical report*, Graduate School of Business, Columbia University, NY.

Clarke, G. and Wright, J. W. (1964). Scheduling of vehicle from a central depot to a number of delivery points, *Operations Research* **12**: 568–581.

Desrosiers, J., Ferland, J., Rousseau, J.-M., Lapalme, G. and Chapleau, L. (1981). An overview of a school busing system, *in* N. Jaiswal (ed.), *Scientific Managment of Transport Systems*, Vol. IX, International Conference on Transportation, New Delhi, November 26-28, 1980, pp. 235–243.

Dolan, E. and Moré, J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming, Serie A* **91**: 201–213.

Emonet, B. (2000). Interface graphique pour la confection d'horaires et de tournées de véhicules scolaires, *Projet de semestre*, EPFL, Switzerland.

Gavish, B. and Shlifer, E. (1979). An approach for solving a class of transportation scheduling problems, *EJOR* **3**(2): 122–134.

Glardon, C., Delaloye, V. and Liebling, T. M. (1992). Referee assignment for volleyball championsips: a competition between three local search heuristics, *Investigación Operativa* **2**(3): 199–219.

Glover, F. (1977). Heuristic for integer programming using surrogate constraints, *Decision Sciences* **8**: 156–166.

Hansen, P. (1986). The steepest ascent mildest descent heuristic for combinatorial programming, *Technical report*, Congress on Numerical Methods in Combinatorial Optimization, Capri,Italy.

Hansen, P. and Jaumard, B. (1987). Algorithms for the maximum satisfiability problem, *Technical report*, Rutgers University.

Hao, J.-K. and Pannier, J. (1998). Simulated annealing and tabu search for constraint solving, *Proceedings of the fifth international symposium on artificial intelligence and mathematics*.

Kirkpatrick, S., Gelatt, C. D. J. and Vecchi, M. P. (1983). Optimization by simulated annealing, *Science* **220**: 671–680.

Kurbel, K. (1998). The trade-off between solution quality and computing times of intelligent algorithms - a computational study on the role of parameters and time budgets, *in* Kasabov, N. et al. (ed.), *Progress in Connectionist-Based Information Systems Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Systems*, Vol. 1, pp. 604–607.

Lin, S. (1965). Computer solutions of the traveling salesman problem, *Bell System Tech. Journal* **44**: 2245–2269.

Lutz, M. and Ascher, D. (1999). *Learning Python*, O'Reilly & Associates.

Newton, R. M. and Thomas, W. H. (1969). Design of school bus routes by computer, *Socio-Economic Planning Science* **3**: 75–85.

Rosenkrantz, D., Stearns, R. and Lewis, P. (1974). Approximate algorithms for the traveling-salesperson problem, *Proceedings of the 15th Annual IEEE Symposium on Switching and Automata Theory*, pp. 33–42.

Rossier, Y., Troyon, M. and Liebling, T. M. (1986). Probabilistic exchange algorithms and Euclidean traveling salesman problems, *OR Spektrum* **8**(3): 151–164.

Swersey, A. J. and Ballard, W. (1984). Scheduling school buses, *Management Science* **30**(7): 844–853.

Watters, L. J. (1967). Reduction of integer polynomial programming problems to zero-one linear programming problems, *Operations Research* **15**: 1171–1174.