Journal Pre-proof

A personalized walking bus service requiring optimized route decisions: a real case

Emanuele Tresoldi, Federico Malucelli, Maddalena Nonato

PII: S0377-2217(19)30625-3
DOI: https://doi.org/10.1016/j.ejor.2019.07.046
Reference: EOR 15947

To appear in: *European Journal of Operational Research*

Received date: 1 December 2018
Accepted date: 22 July 2019

Please cite this article as: Emanuele Tresoldi, Federico Malucelli, Maddalena Nonato, A personalized walking bus service requiring optimized route decisions: a real case, *European Journal of Operational Research* (2019), doi: https://doi.org/10.1016/j.ejor.2019.07.046

Highlights

- Definition of a new optimization problem: the Door-to-School Pedibus lines design.
- The application arises from a real problem with high social and educational impact.
- Line merging and dynamic capacity make the problem different from standard VRP.
- Multiple solution methods proposed with no clear dominance.
- Tested on both synthetic and real word scenarios.

A personalized walking bus service requiring optimized route decisions: a real case

Emanuele Tresoldi[1]

Università degli Studi di Milano - Dipartimento di Informatica

Federico Malucelli

Politecnico di Milano - Dipartimento di Elettronica, Informazione e Bioingegneria

Maddalena Nonato

Università degli Studi di Ferrara - Dipartimento di Ingegneria

Abstract

We address the design of the lines of a Walking Bus service according to a new paradigm, where children are picked up at home. The scarcity of accompanying persons together with the limit on the length of the deviations from the shortest itinerary of each child make the problem different from the traditional school bus and walking bus design. We propose an arc-based model, a path-based model tackled by column generation, and a heuristic procedure. Solution approaches are tested on a set of real and realistic instances. Real instances refer to the case study of a primary school in Italy.

Keywords: Transportation, Walking school bus, Network design, Capacitated spanning tree, Column generation

2010 MSC: 00-01, 99-00

1. Introduction

Scientific research over the last 20 years has provided increased evidence in favor of children active commuting to school (Dijk et al., 2014). In fact, any walking or biking during the trip from home to school requires some physical activity which provenly i) fights childhood obesity, which is a major health concern among parents (McDonald et al., 2011); ii) improves reaction to

[1]Corresponding author: emanuele.tresoldi@unimi.it

stress, helping children to cope with cognitive stressors often experienced during the school-day (Lambiase et al., 2010). Nevertheless, in the developed world most parents drive their children to school and school gates get congested by traffic. Data support this claim: McDonald & Aalborg (2009) reports that the percentage of elementary school pupils being driven to school in the US raised from 18% in 1969 to 55% in 2001, while the share of those walking or cycling has dramatically decreased, dropping from 41% to 13%. Direct negative consequences include environmental damage due to traffic pollution near schools, and children getting accustomed to passive travel modes. Subtler perils were recently uncovered: in Sunyer & et al. (2017) it is proved that exposure to traffic emissions is associated with children impaired cognitive development due to their impact on memory. Investigations into parents motivations (Westman et al., 2017) revealed that distance from school is only partially accountable for modal choice: even parents living within walking distance elect driving rather than walking as their preferred travel mode due to convenience and time saving and do not allow children to walk without adult supervision for fear of traffic and strangers perils.

Several projects have been carried out to reverse this trend (Chillón et al., 2011). The most effective initiative within such projects is the so-called *Walking School Bus* also known as *Pedibus*, where a group of children is led to school by an adult. Pedibus allows children to enjoy a safe walk to school at no parental burden. In addition to active travel mode benefits, walking to school with peers helps children to develop self-confidence, improves social relations, teaches them pedestrian safety rules, and increases their independent mobility (Kingham & Ussher, 2007; Mendoza et al., 2012). A Pedibus service can be realized according to a variety of paradigms, ranging from informal agreements among neighbors to formal programs sponsored by the school. Trained adults, so called Pedibus *drivers*, follow an assigned route, a so called Pedibus *line*, and make stops at specified times and locations to collect children (McDonald & Aalborg, 2009). Setting up a long term Pedibus service is a complex task involving the collaboration of the schoolteachers and management, the children families, and the municipality. A main step in the service planning is the design of the lines. These are usually proposed by the organizing committee and grounded on common sense. In the traditional service, very few stops are located along each line and parents take children to their closest stop. As lines tend to stay the same, year after year, the service remains sufficiently attractive as far as lines cross neighborhoods where demand remains high over time,

such as restricted areas with historically high density of school pupils. Other neighborhoods where demand varies along time or is scarce and diffused are likely to be out of the service reach. Children live too far from the closest stop of a fixed line, but local demand does not justify the presence of another fixed line stop. Experience from public transport suggests that sparse demand can be captured only by a personalized service, where lines adjust to demand. In this framework, this means to pick up children at home, yielding the *Door-to-School* Pedibus (D2SP) discussed in this paper. Since i) the length of the lines must be kept within walkable distance and ii) lines require drivers but drivers are scarce, the lines design step yields new challenging combinatorial optimization problems.

This study is part of an ongoing project aimed to establish the D2SP service in a medium size city in Italy. Previous works (Malucelli et al., 2017, 2018) addressed a simplified version of the line design problem, targeting the minimum number of lines. Here we take it a step further, introducing a binding feature of the real application related to the number of drivers and discuss how this issue impacts on the structure of the lines. We present two problem variants (section 2), discuss related works (sections 3), provide arc-based (section 4) and path-based mathematical models and describe how to tailor a column generation solution approach in this framework (section 5). Ad-hoc heuristics are presented in section 6. We experimentally compare the effectiveness of all presented approaches on real and realistic instances (section 7), computationally investigate the trade-off among the service quality (related to the distance walked along the lines), the service cost (related to the number of drivers) and we draw conclusions (section 8). The list of main symbols used for variables, sets, and parameters throughout the paper is reported in Tables 1 and 2 in the supplementary material available online.

2. The door to school walking bus

For safety reasons, in the traditional Pedibus services i) no more than a maximum number of children per driver is allowed on each line, ii) drivers start their service at line terminals and reach the school with the group. When designing a line, the number of drivers is fixed a priori by estimating the maximum number of children that could be collected along the line.

As mentioned, the traditional Pedibus service is suited for serving children in densely populated neighborhoods (usually within about 1.5 km from school) or those living much further who can

autonomously reach the terminal. This type of service, though, fails in capturing whole the demand uniformly distributed around school and children who cannot be brought to the Pedibus stop at the convened time. This is the case for the school tackled in our study, but it extends to many other cases where the school is located in the historical center of a city, surrounded by residential neighborhoods.

Formally, the service here proposed can be described as follows. We are given: i) a set of locations corresponding to the children homes $H = \{1, \ldots, n\}$ and the school s; ii) for each home i, the number of children to be accompanied to school q_i and the walking distance S_i from i to s along the shortest path; iii) the maximum number of children per driver ρ. The Door-to-School line design problem (D2S) arises in two different variants. In D2S_1 the problem is to find the minimum number of drivers needed to walk all children to school, provided that, for each child, the actual walked distance does not exceed δ times the shortest one ($\delta > 1$), with δ potentially varying based on the location of each i. In D2S_2 the number of drivers and their locations are given, and the problem is to design a set of lines serving all the demand so that the maximum over all children of the ratio of the actual walked distance over the shortest one is minimized.

Beside the main target, a second issue concerns safety. First of all, during preprocessing any road link not meeting a minimum safety standard for children walkability, such as for example sidewalk continuity, is discarded from the road network on which shortest paths between locations are computed. Then, for each pair of locations i and j a weight coefficient denoted as d_{ij} is devised that is inversely correlated to walkability along the shortest walking path from i to j (the highest the worst). The aim is to favour itineraries featuring safety related traits, such as wide sidewalks, high curbs, and low speed areas, by penalizing the potential occurrence of negative characteristics, such as sidewalk uneven surface or traversing congested zones. We consider such measure as additive on the arcs. Both problems D2S_1 and D2S_2 can thus include it as a secondary objective so that, among the solutions with equal primary target value, the most safety aware is selected.

Finally, the main peculiarity that distinguishes the D2S line design problem from other better known ones, such as the school bus routing problem (Riera-Ledesma & Salazar-González, 2013), is the limit ρ on the maximum ratio of children per driver together with the possibility of having multiple drivers acting together on the same line, from terminal to school, or on joint lines upon merging till destination (joint lines proceed to school as a whole), which makes capacity rise to

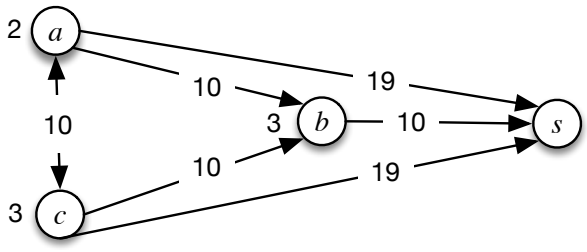

Figure 1: Demand at service points a, b, c, and travel time on the arcs. Lines $l_1 = (a, b, s)$ and $l_2 = (c, b, s)$ require one driver each and merge at b to share excess capacity and comply with the maximum number of children per driver requirement.

multiples of ρ according to the current number of drivers. Such feature helps to efficiently handle cases with $q_i \geq 2$, which frequently arise, e.g. think of children in the same family or living in the same building. As such demand cannot be split on different lines, it may turn advantageous to merge two or more lines into a single one at an intermediate stop. In fact, *line merging* can be exploited to decrease the number of required drivers, as shown in the toy example depicted in figure 1. Consider 3 homes, namely a, b, c, with $q_a = 2, q_b = 3$, and $q_c = 3$ children, respectively. The walking distances between pairs of locations is as follows: $c_{ab} = c_{ac} = c_{bc} = 10, c_{bs} = 10, c_{as} = c_{cs} = 19$. If $\rho = 4$, it easy to see that any line serving two stops requires at least 2 drivers, 2 drivers are sufficient to manage the whole demand whether it is served either along a single line or by two merged lines, whereas a solution made of 2 separate lines would require 3 drivers. In case of lines with many stops, children at or near terminal are the most affected regarding walking time. In the example, take line l^H along the shortest Hamiltonian path which starts at a, visits c and b, and ends at s. Users at terminal node a take 30 time units to reach school, thus the ratio of their traveling time over S_a is 30/19, while the total user traveling time along l^H is 150. Consider instead a solution with one driver leading line l_1, which goes from a to s passing through b, and a second driver leading line l_2 which starts at c, merges with l_1 in b, and ends at school; the whole demand is served using 2 drivers as in l^H, but the sum of the traveling time of all users is 130 and users in a, in particular, experience a ratio close to one. Thus, merging effectively allows to reduce drivers as well as walking time.

Both problem variants D2S_1 and D2S_2 are modeled on a directed graph $G = (N, A)$, where i) the set of $n + 1$ nodes $N = H \cup \{0\}$ corresponds to the home locations (client nodes) and the

school s (node 0), ii) the arc set A is complete and each arc $(i, j) \in A$ corresponds to the pedestrian connection between i and j on the street network, once it has been filtered to remove unsafe links, along the minimum traveling time path. Each arc (i, j) is thus characterized by the travel time c_{ij} of such a path and a weight d_{ij} related to its walkability, as discussed before. Each client node i is a source of q_i flow units and 0 is the only sink. Since lines that have merged are not allowed to split, it follows that a solution is made of a set of paths on G, each path going from a node i to 0 such that i) each client node belongs to at least one path, and ii) from each client node there is exactly one outgoing arc. Therefore, the service network is an arborescence centered at node 0 (not necessarily shaped as a star) whose leaves identify the line terminals and where each client's flow travels along the unique path on the tree from its source node to the sink. Since drivers are allowed to join the line only at terminal stops, their assignment to the tree leaves must be compliant with the number of children along the path. The network is capacitated, capacity intended as the maximum number of children allowed on eac arc, but capacity is a variable and not a given parameter. In addition, a solution is feasible in D2S$_1$ if for each client node i the unique path from i to 0 fulfills the distance constraint. In D2S$_2$ a solution is feasible if paths start from drivers' location and the demand collected along each path is compliant with the given number of drivers. This structure is at the core of the two mathematical models that we propose in Sections 4 and 5, one based on decisions on the arcs and the other based on decisions on the paths, respectively.

3. Related works

Few related papers stemming from telecommunication applications tackle tree shaped network design problems. We mention the most pertaining ones and underline problem differences. Mixed Integer Linear Programming (MILP) models for the minimum spanning tree with a given number of leaves are presented in Fernandes & Gouveia (1998) and Gouveia & Simonetti (2017). In terms of Pedibus, that problem corresponds to $\rho = \infty$, i.e., one adult per line terminal, with unlimited route duration and a given number of drivers, though not pre-allocated to specific nodes. The objective function is a linear function of the arcs in the tree, as in our secondary objective. Another related problem in telecommunication concerns delay-constrained minimum spanning trees (Salama et al., 1997). In Pedibus terms it corresponds to choosing the lines along the most walkable network (our secondary objective function) complying with a fixed maximum walking distance, disregarding the

number of drivers. In multi-level capacitated spanning tree problems arising in local access network design (Gamvros et al., 2006) the focus is on capacity, since nodes have traffic requirements and links can be equipped with different types of facilities, each one with its own cost and capacity. In Pedibus terms it means that drivers could join the line at any stop rather than only at the terminals.

Regarding the optimization of the home-to-school children itineraries, Porro (2015) solves the planning of traditional Pedibus lines by a three steps heuristic procedure, previously developed for the School Bus Routing Problem (SBRP). Exact approaches for the SBRP by MILP models are deeply discussed in Riera-Ledesma & Salazar-González (2013). Actually, once stops have been located and children assigned to them, the resulting problem, i.e., an Open Distance-constrained Capacitated Vehicle Routing Problem (O-D-CVRP), shares many features with D2S. Nevertheless, the two problems do not coincide. While in VRPs there is a route for each vehicle and vice versa, in D2S this one to one correspondence may not hold. In fact, a few drivers can be active on the same line, and upon merging more lines share the final part of their route. This issue and its impact on the arc-based models is further discussed in Section 5.

Two studies address the risk of home-to-school itineraries. In Japan it is quite common for children living in urban environments to walk to school unescorted. Since safety increases the more the journey is shared with peers, Tanaka et al. (2016) consider as a measure of risk the distance of the sub-path that each child walks alone. As individual shortest paths to school are likely to differ from child to child, sharing part of the itinerary leads to routes with increased distance. A bi-criteria MILP formulation is proposed which minimizes total risk and total walking distance. Tanaka et al. (2018) generalize it to allow multiple visits of the same location, a required feature if working directly on a graph that is a physical representation of the street network.

This work builds upon an arc-based model (Malucelli et al., 2017) and a path-based model (Malucelli et al., 2018) developed for the uncapacitated version, where the number of drivers per line does not depend on the number of children served along the line. In that case, minimizing the number of drivers corresponds to minimizing the leaves of the arborescence. As a whole, we argue that no previous work tackles neither $D2S_1$ nor $D2S_2$, which motivates the present study.

4. Arc-based models

The first class of formulations introduces three families of variables associated with each arc $(i,j) \in A$: i) integer flow variables w_{ij}, giving the number of children traveling directly from i to j; ii) integer variables x_{ij}, representing the number of drivers traveling from i to j; iii) binary design variables y_{ij}, equal to 1 if j is the next stop after i along a line, 0 otherwise. Variables associated with the nodes are: i) integer variables z_i $\forall i \in H$ representing the number of drivers starting their duty from i; ii) $\pi_i \geq S_i$ associated with the walking time of the itinerary along the line from i to 0. Moreover, in D2S$_1$ for each $i \in H$ the maximum ratio between π_i and S_i is given and denoted as $\delta_i > 1$, and the maximum allowed traveled distance from i to 0 is denoted as $\Delta_i = \delta_i S_i$. Both problem versions involve two criteria, therefore we introduce a trade-off parameter ϵ and merge them in a unique utility function, where ϵ is sized to yield a hierarchical objective function in which walkability is used to elect the highest quality network among those optimizing the other criterion. A MILP model for D2S$_1$, i.e., minimizing the number of drivers, is reported below.

$$\min \sum_{i \in N} z_i + \epsilon \sum_{(i,j) \in A} d_{ij} y_{ij} \tag{1}$$

$$-\sum_{(j,i) \in A} w_{ji} + \sum_{(i,j) \in A} w_{ij} = q_i \qquad \forall i \in H \tag{2}$$

$$-\sum_{(j,i) \in A} x_{ji} + \sum_{(i,j) \in A} x_{ij} = z_i \qquad \forall i \in H \tag{3}$$

$$w_{ij} - \rho x_{ij} \leq 0 \qquad \forall (i,j) \in A \tag{4}$$

$$y_{ij} - w_{ij} \leq 0 \qquad \forall (i,j) \in A \tag{5}$$

$$y_{ij} - x_{ij} \leq 0 \qquad \forall (i,j) \in A \tag{6}$$

$$x_{ij} - M y_{ij} \leq 0 \qquad \forall (i,j) \in A \tag{7}$$

$$\sum_{(i,j) \in \mathcal{A}} y_{ij} = 1 \qquad \forall i \in H \tag{8}$$

$$\pi_j - \pi_i + (\Delta_j - S_i + c_{ij})y_{ij} + (\Delta_j - S_i - c_{ji})y_{ji} \leq \Delta_j - S_i \qquad \forall (i,j) \in A : i \neq 0 \qquad (9)$$

$$\pi_i \leq \Delta_i \qquad \forall i \in H \qquad (10)$$

$$\sum_{(j,i)\in\mathcal{A}} y_{ji} + z_i \geq 1 \qquad \forall i \in H \qquad (11)$$

$$z_i - M(1 - y_{ji}) \leq 0 \qquad \forall (j,i) \in A \qquad (12)$$

$$w_{ij} \in \mathcal{Z}_0^+,\ x_{ij} \in \mathcal{Z}_0^+,\ y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (13)$$

$$z_i \in \mathcal{Z}_0^+,\ \pi_i \geq 0 \qquad \forall i \in H \qquad (14)$$

Constraints (2-3) are flow conservation constraints of children and drivers, respectively. Together with constraints (11-12) they ensure that drivers are on the line from the start as they join it only at terminals. Constraints (4), by stating the maximum number of children per driver, are actually linking the two sets of flow variables. Constraints (5, 6, 7) link the 0-1 variables to the corresponding flow variables. Constraints (8) establish the arborescence structure of the solution: a non-terminal node may have multiple inflow arcs, but from each client node there must be exactly one arc carrying flow out. Inequalities (9), defining the value of distance variables π_i, represent line continuity and subtour elimination constraints. These constraints are an extension of Miller-Tucker-Zemlin subtour elimination constraints presented by Laporte et al. (1987) for the Distance Constrained Vehicle Routing Problems (DVRP) and further studied in Kara (2010) and Bektaş & Lysgaard (2015). Inequalities (10) define the maximum deviation constraints for each child and set the range $[S_i, \Delta_i]$ for variable π_i. Constraints (11-12) relate drivers flow to the network structure, binding the sources of drivers flow to the leaves of the arborescence. Indeed, constraints (11) state that if a node is a leaf, at least one unit of driver flow must originate from there, while (12) state that if a node is not a leaf, no driver flow will originate from there. The value of M in constraints (7) and (12) can be bounded by the number of drivers in any feasible solution for D2S$_1$.

When modeling D2S$_2$, let us introduce: i) the continuous variable Θ representing the maximum, over all $i \in H$, of the ratio between the travel time from i to 0 along the line and S_i, as stated by (17); ii) an integer parameter u_i representing the number of drivers that start their duty on the line at node i, $\forall h \in H$. In particular, u_i becomes the right-hand side of the drivers flow conservation constraints (16) and allows to rewrite (11-12) as (18-19); iii) the set of terminal nodes $\bar{H} = \{h \in H : u_h > 0\}$ (in D2S$_1$ $\bar{H} = H$ as any demand node is a potential terminal); iv) values $\Theta^{lb} < \Theta^{ub}$ providing a lower and an upper bound on Θ. D2S$_2$ can be formulated as follows:

$$\min \Theta + \epsilon \sum_{(i,j)\in\mathcal{A}} d_{ij} y_{ij} \quad \text{subject to (2), (4-9), (13)} \tag{15}$$

$$-\sum_{(j,i)\in A} x_{ji} + \sum_{(i,j)\in A} x_{ij} = u_i \qquad \forall i \in H \tag{16}$$

$$\pi_i - \Theta S_i \leq 0 \qquad \forall i \in H \tag{17}$$

$$\sum_{(j,i)\in\mathcal{A}} y_{ji} \geq 1 \qquad \forall i \in H \setminus \bar{H} \tag{18}$$

$$\sum_{(j,i)\in\mathcal{A}} y_{ji} = 0 \qquad \forall i \in \bar{H} \tag{19}$$

$$\Theta^{lb} \leq \Theta \leq \Theta^{ub} \qquad \forall i \in \bar{H} \tag{20}$$

$$\pi_i \geq 0 \qquad \forall i \in H \tag{21}$$

Note that in D2S$_2$ the value of M in constants (7) can be set to $\sum_{i\in H} u_i$. Moreover, since the maximum ratio between π_j and S_j over all $j \in H$ is now a variable to be minimized, in D2S$_2$ the value Δ_i is set to $\Theta\mathcal{S}_i$ in constraints (10) now (17) and Δ_j is equal to $\Theta^{ub}\mathcal{S}_j$ in (9). Constraint (20) is not required for the correctness of the formulation but can provide a tighter linear relaxation. Lower and upper bounds on Θ can be obtained as follows. Let $\Theta_i = min_{j\in H\setminus\bar{H}}(\frac{c_{ij}+c_{j0}}{c_{i0}})$ be the minimum ratio between the travel time from i to 0 on a line starting from $i \in \bar{H}$ and serving at least one node j in $H \setminus \bar{H}$ and S_i. Since all nodes must be visited by at least one line then a feasible value for Θ^{lb} is given by $\max_{i\in\bar{H}} \Theta_i$. An upper bound on Θ is provided by any feasible solution for D2S$_2$.

Additional constraints based on the walking distance as in (see Kara, 2010) can be included in both formulations.

$$\pi_i \geq \sum_{(i,j)\in A} (S_j + c_{ij}) y_{ij} \qquad \forall i \in H \tag{22}$$

$$\pi_i \leq S_i y_{i0} + \Delta_i (1 - y_{i0}) \qquad \forall i \in H \tag{23}$$

$$\pi_j \leq \Delta_j - \Delta_j y_{ij} + min(\Delta_j, \Delta_i - c_{ij}) y_{ij} \qquad \forall (i,j) \in A \tag{24}$$

Constraints (22) reinforce the lower bound on the value of π. Inequalities (23) force $\pi_i = S_i$ when i is the last client node on the line before s, and may replace (10) in the arc model for D2S$_1$. Constraints (24) provide an upper bound on the value of π. It is worth noting that, the two-

commodity network flow approach proposed in Baldacci et al. (2004) as well as capacity bounding constraints for the Capacitated Vehicle Routing Problem (CVRP), first presented in Desrochers & Laporte (1991) and further studied in the context of Distance and Capacity Constrained Vehicle Routing Problem (DCVRP) in Kara & Derya (2011), cannot be used in our models since capacity is not an input parameter but depends on a decision variable. Finally, in both models, the linear relaxation can be further strengthened by adding i) standard subtour elimination constraints on the binary design variables y_{ij} (Applegate et al., 2006), ii) capacity cuts, as in the CVRP (Lysgaard et al., 2004) and iii) k-path inequalities (see Kohl et al., 1999).

5. Path-based models

In the path-based formulations, lines are seen as paths along which drivers walk from terminal to destination. In the models, indeed, a line is represented by a decision variable associated with the drivers active along that line. Let P be the set of all *feasible* paths from any node $i \in H$ to 0. Let us extend the notation to consider the following subsets of P: let $\mathcal{P}_i \subset P$ denote the set of paths visiting node $i \in H$, $\mathcal{P}_{ij} \subset P$ be the set of paths using arc $(i,j) \in A$, and $\bar{\mathcal{P}}_i \subset \mathcal{P}_i$ be the set of paths whose terminal node is $i \in H$. Moreover, H_p (A_p) stands for the set of nodes in H (arcs in A) covered by path p while, for each $i \in H_p$, c^p_{i0} denotes the time to destination along p.

Path p is feasible with respect to D2S_1 if i) it is elementary, ii) for each node i on the path, c^p_{i0} is no greater than Δ_i. Note that the children per driver ratio is not involved in D2S_1 path feasibility, since such requirement can be verified only once potential mergings concerning the path have been set. Variables w_{ij} and y_{ij} previously introduced are retained from the arc models with the same meaning and domain, while we introduce: i) integer variables ξ_p $\forall p \in \mathcal{P}$ representing the number of drivers traveling along path p; ii) binary variables ζ_i $\forall i \in H$, where $\zeta_i = 1$ if i is the

terminal stop of a line and 0 otherwise. A path-based formulation for D2S$_1$ is the following.

$$\min \sum_{p\in\mathcal{P}} \xi_p + \epsilon \sum_{(i,j)\in\mathcal{A}} d_{ij}y_{ij} \tag{25}$$

$$-\sum_{(j,i)\in A} w_{ji} + \sum_{(i,j)\in A} w_{ij} = q_i \qquad \forall i \in H \tag{26}$$

$$\rho \sum_{p\in P_i} \xi_p - \sum_{(i,j)\in A} w_{ij} \geq 0 \qquad \forall i \in H \tag{27}$$

$$My_{ij} - \sum_{p\in\mathcal{P}_{ij}} \xi_p \geq 0 \qquad \forall (i,j) \in A \tag{28}$$

$$M\zeta_i - \sum_{p\in\bar{\mathcal{P}}_i} \xi_p \geq 0 \qquad \forall i \in H \tag{29}$$

$$\sum_{(j,i)\in A} y_{ji} - (n-1)(1-\zeta_i) \leq 0 \qquad \forall i \in H \tag{30}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij} = 1 \qquad \forall i \in H \tag{31}$$

$$\zeta_i \in \{0,1\} \qquad \forall i \in H \tag{32}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{33}$$

$$w_{ij} \in \mathcal{Z}_0^+ \qquad \forall (i,j) \in A \tag{34}$$

$$\xi_p \in \mathcal{Z}_0^+ \qquad \forall p \in \mathcal{P} \tag{35}$$

As in the D2S$_1$ arc model, the hierarchical objective function (25) is made of the number of drivers plus a secondary term related to the service network walkability, properly weighted by ϵ. Constraints (31) and (26) are retained from the D2S$_1$ arc model, (2) and (8) respectively, and here reported for reader sake. The children per driver ratio at each node i is enforced by (27). Constraints (29) activate variable ζ_i if there is at least (one driver on) a path originating from i. Constraints (30) state that if at least one line enters i then i is not a terminal, i.e. no driver can start at i. Constraints (28) link variables ξ and y, stating that any arc traversed by at least one driver is part of the service network. Constant M in (28) and in (29) can be bounded by the total number of drivers in any feasible solution for D2S$_1$. Constraints (32-35) provide variable domains.

When dealing with D2S$_2$, the general structure of the D2S$_1$ path-based model can be retained but for few modifications to account for the different objective function and the additional requirements. In particular: i) non-negative variable Θ is retained from the D2S$_2$ arc-based model with

the same meaning and domain; ii) variables ξ_p are restricted to be binary, equal to 1 if path p is selected and 0 otherwise. Since in D2S$_2$ drivers and their allocation to terminals are given, the number of drivers on a path depends on its terminal. Then, a path p is D2S$_2$ feasible if i) it is elementary, ii) it starts from $i \in \bar{H}$, i.e., the terminal node set, iii) for each node $i \in H_p$, c^p_{i0} is no greater than ΘS_i. As in D2S$_1$, path feasibility does not involve the check for the children-per-driver ratio that depends on potential line mergings. A path-based formulation for D2S$_2$ follows:

$$\min \Theta + \epsilon \sum_{(i,j)\in\mathcal{A}} d_{ij} y_{ij} \quad \text{subject to (26), (28), (31), (33), (34)} \tag{36}$$

$$\rho \sum_{h\in\bar{H}} u_h \sum_{p\in P_i\cap\bar{\mathcal{P}}_h} \xi_p - \sum_{(i,j)\in A} w_{ij} \geq 0 \qquad \forall i \in H \tag{37}$$

$$1 - \sum_{p\in\bar{\mathcal{P}}_i} \xi_p \geq 0 \qquad \forall i \in \bar{H} \tag{38}$$

$$\Theta - \sum_{p\in\bar{\mathcal{P}}_i} \Theta_p \xi_p \geq 0 \tag{39}$$

$$\xi_p \in \{0,1\} \qquad \forall p \in \mathcal{P} \tag{40}$$

where, for each path $p \in P$ originating from $h \in \bar{H}$, $\Theta_p = max_{i\in H_p}(\frac{c^p_{i0}}{S_i})$ denotes the maximum walking time ratio over the path nodes. (36) is the hierarchical objective function of the D2S$_2$ arc-based model, section 4. Constraints (37) enforce the children-per-driver ratio at each node $i \in H$. Note that the number of drivers at i depends on the terminal node of all the paths traversing i. Inequality constraints (38) ensure that there is only one path originating from each leaf $i \in \bar{H}$. Actually, they will be satisfied as equalities due to the arborescence structure. Inequalities (39) bound Θ from below, being no less than each Θ_p for all selected path. The value of M in constraint (28) can be set to $|\bar{H}|$.

Both path-based models can be strengthened by adding the following constraints, which can be violated in the linear relaxation.

$$\sum_{p\in P_i} \xi_p \geq 1 \qquad \forall i \in H \tag{41}$$

$$1 - \zeta_i - \sum_{(j,i)\in A} y_{ji} \leq 0 \qquad \forall i \in H \tag{42}$$

Since the number of paths is exponential in n, the models are addressed by column generation. Integer solutions are provided by a primal heuristic. As the tailing off can be rather time consuming

while the application requires limited computing time (about 2 hours on a regular PC), we do not perform a full branch and price scheme but, once the root node has been solved to optimality, we execute a partial branching strategy to improve the quality of the incumbent solution. In the following we present the main building blocks of the solution approach, i.e., the pricing sub-problem, the partial branching scheme, the primal heuristic procedure and heuristic algorithms to provide an upper bound.

A Column Generation approach. In our column generation procedure the Restricted Master Problem (RMP) is given by the linear relaxation (25 - 31, 41, 42) for D2S$_1$ and (36 - 42) for D2S$_2$, and it is initialized with the columns corresponding to the solution found by the heuristic algorithms described in section 6. In the pricing sub-problem, we look for the path p associated with the variable ξ_p with minimum reduced cost, taking into account the dual representation of the RMP. Consider a path $p(h)$ originating from $h \in \bar{H}$ ($\bar{H} = H$ in D2S$_1$) and ending at 0. Let α_i, α'_i, β_i, γ_{ij}, μ_h, μ'_h and ω_h be the non-negative dual variables associated with constraints (27), (37), (41), (28), (29), (38) and (39) respectively. The reduced cost of path $p(h)$, here denoted as $\lambda_{p(h)}$, is given by (43) for D2S$_1$ and by (44) for D2S$_2$:

$$\text{(D2S}_1\text{)}\ \lambda_{p(h)} = 1 - \sum_{i \in H_{p(h)}} (\rho\alpha_i + \beta_i) + \sum_{(i,j) \in A_{p(h)}} \gamma_{ij} + \mu_h \tag{43}$$

$$\text{(D2S}_2\text{)}\ \lambda_{p(h)} = 0 - \sum_{i \in H_{p(h)}} (\rho u_h \alpha'_i + \beta_i) + \sum_{(i,j) \in A_{p(h)}} \gamma_{ij} + \mu'_h + \Theta_p \omega_h \tag{44}$$

Therefore, for both D2S$_1$ and D2S$_2$, the pricing problem is an elementary shortest path problem with resource constraints (see Irnich & Desaulniers, 2005) with respect to arc costs $\bar{\gamma}_{ij} = \gamma_{ij} - (\rho\alpha_i + \beta_i)$ for D2S$_1$ and $\bar{\gamma}_{ij} = \gamma_{ij} - (\rho u_h \alpha'_i + \beta_i)$ for D2S$_2$. At each iteration, given the current dual variables, for each $h \in \bar{H}$ we search for $p(h)$ such that $\lambda_{p(h)} < 0$ and $c^p_{i0} \leq \Delta_i \ \forall i \in H_p$ ($\Delta_i = \Theta^{ub} \mathcal{S}_i$ in D2S$_2$). Indeed, the resource monotonically consumed along the path is the time available for completing the path to destination and each arc (ij) consumes c_{ij}. In particular, the residual time available to reach destination from any internal node j is the minimum, over all the nodes i from origin to j, of Δ_i minus the time needed along the current path from i to j. This pricing sub-problem has a similar structure to the one in Malucelli et al. (2018). At each iteration of the column generation procedure, $|\bar{H}|$ independent pricing sub-problems, one for each node in $\bar{H}$, are solved. Two different procedures are employed: the pricing sub-problem is first tackled heuristically by

a simple greedy procedure based on a nearest neighbor search. This procedure is mainly used to rapidly populate the set P in the first iterations of the column generation. The second procedure solves the problem exactly using a slightly modified version of the pulse algorithm, presented in Lozano et al. (2016), whose performance heavily depends on its ability to exploit the constrained resource, i.e., time, to reduce the search space. For each $h \in \bar{H}$, the greedy procedure builds the path $p(h)$ incrementally, starting from h. The residual time in h is initialized to Δ_h and let Δ'_i denote the residual time at the current node i. Let us introduce $\mathcal{FS}(i)$ as the notation for the set of arcs originating from node i. The arc (ij) in $\mathcal{FS}(i)$ with the lowest $\bar{\gamma}_{ij}$ is chosen and the residual time is updated as $\Delta'_j = min\{\Delta'_i - c_{ij}, \Delta_j\}$; the search backtracks to i if Δ'_j is below S_j and ends when it reaches 0. If the greedy procedure fails to produce any negative cost path, then the problem is solved to optimality using the recursive *pulse* algorithm. The exact pricing algorithm is executed once for each node $h \in \bar{H}$ taking h as the starting point. Since each search is an independent process, this step can take advantage from a parallel implementation. We introduced the following modifications with respect to Malucelli et al. (2018) to speed up the solution process.

In a pre-processing phase we remove from A every arc (i, j) such that $c_{ij} + S_j > \Delta_i$ obtaining a restricted set of arcs A'. Indeed, as mentioned, Δ_i is the maximum time available to reach destination from i, while $c_{ij} + S_j$ is a lower bound when the path has to go through j on its way from i to s. Then, if $(ij) \notin A'$, not only no path will ever go directly from i to j but it will never visit j if i has already been visited, since walking distances on which coefficients c_{ij} are based, by definition satisfy triangular inequalities. Now, consider the reduced graph $G' = (N, A')$: whenever a node $i \in H$ has no incoming arcs the associated variable ζ_i is fixed to 1.

Graph G' can be further shrunk considering the value of the dual variables. Remind that all dual variables are non-negative and that in (43) and (44) α_i, α'_i and β_i are subtracted while the γ_{ij}, μ_h, μ'_h and ω_h variables are summed. Therefore, visiting a location $i \in H$ is profitable only provided that $\rho\alpha_i + \beta_i > 0$ in D2S$_1$ or $\rho u_h \alpha'_i + \beta_i > 0$ in D2S$_2$, and potentially increases $\lambda_{p(h)}$ otherwise. We can now define a new node set N' removing from H all non-profitable children. The pricing algorithms are run on the reduced graph $G'' = (N', A')$.

Now, we can bound from below the reduced cost of any path originating from $h \in \bar{H}$ in order to discard h as a starting node if that bound is non-negative. The optimistic guess is obtained by taking an upper bound of the absolute value of the contribution provided by variables α_i and β_i

and a lower bound of the one provided by the γ_{ij} variables. In detail, consider h and its neighbors $\{i : (hi) \in \mathcal{FS}(h)\}$. The reduced cost of any $p(h)$ amounts of a part which is a constant or depends just on h, i.e., $1+\mu_h-(\rho\alpha_h+\beta_h)$ in D2S$_1$ or $\mu'_h+\omega_h-(\rho u_h\alpha'_h+\beta_h)$ in D2S$_2$. An optimistic guess of the rest is given by the minimum γ_{hi} over all $(hi) \in \mathcal{FS}(h)$, plus the sum of the minimum $\bar{\gamma}_{ij}$ over all neighboring nodes $\{i : (hi) \in \mathcal{FS}(h)\}$ such that $\bar{\gamma}_{ij} < 0$ for at least one arc $(ij) \in \mathcal{FS}(i) : j \neq h$. Formally, the bound for D2S$_1$ and D2S$_2$ is:

$$\text{(D2S}_1\text{)}\ \Lambda_h = 1 + \mu_h - (\rho\alpha_h + \beta_h) + \min_{(hi)\in\mathcal{FS}(h)}\{\gamma_{hi}\} + \sum_{i:(hi)\in\mathcal{FS}(h)} min\{0, (min_{(ij)\in\mathcal{FS}(i)j\neq h}\bar{\gamma}_{ij})\}$$

$$\text{(D2S}_2\text{)}\ \Lambda_h = \mu'_h + \omega_h - (\rho u_h\alpha'_h + \beta_h) + \min_{(hi)\in\mathcal{FS}(h)}\{\gamma_{hi}\} + \sum_{i:(hi)\in\mathcal{FS}(h)} min\{0, (min_{(ij)\in\mathcal{FS}(i)j\neq h}\bar{\gamma}_{ij})\}$$

Finally, if $\Lambda_h > 0$ we do not execute pricing algorithm starting from h.

Primal heuristic procedure. At each iteration of the column generation procedure, given the values ξ^*_p of variables ξ_p in the optimal solution of the current RMP, a simple primal heuristic procedure is applied to find feasible solutions. Variables are ordered in descending order by their ξ^*_p values and the set T is initialized to $\emptyset$. Then, following this order, a variable ξ_p is selected and path p is inserted in T if no other path in T starts from the same terminal.The process ends as soon as the paths in T cover all the nodes in H. Then we check wether the paths in T define a proper tree, as well as wether rounding up the ξ^*_p values of the selected paths meets the maximum children per driver ratio. In such a case a new feasible solution is found, the incumbent T^* is updated if necessary, and, in problem D2S$_2$, considering Θ^* the associated value of variable Θ in T^*, all columns in P such that $\Theta_p > \Theta^*$ are discarded.

Partial branching scheme. At the end of the column generation process, if the final RMP solution is not integer we perform a partial branching strategy in order to find feasible integer solutions. Note that one such solution exists, as the columns of the heuristic solution belong to P. Now, consider UB as the cost of the best integer solution found during the column generation process (the incumbent solution) and LB as the floor of the final fractional RMP solution cost. Let F be, in D2S$_1$, the set of all integer numbers from UB to LB and, in D2S$_2$, the set of values obtained subtracting $\frac{UB-LB}{10}$ from UB until LB is reached. We generate $|F|$ branching nodes, one for each

element $f \in F$. In each branching node we add constraint (45) in D2S$_1$ and constraint (46) in D2S$_2$.

$$\text{(D2S}_1\text{)} \quad \sum_{p \in P} \xi_p = f \tag{45}$$

$$\text{(D2S}_2\text{)} \quad \Theta \leq f \tag{46}$$

We explore these nodes one by one starting from $f = max(F)$. At each step either we find a better integer solution, update the incumbent, and iterate on $f - 1$, or the problem cannot be solved and the search stops, returning the incumbent. In D2S$_1$,at every iteration the current problem inherits all additional columns generated so far. Additional ones are potentially generated by resuming the column generation and solving the linear relaxation taking into account constraint (45). In the pricing sub-problem the dual variable ν_p associated with (45) contributes to the reduced cost of each ξ_p. In D2S$_2$, at every iteration the current problem inherits only columns representing paths with $\Theta_p \leq f$. In both cases, no substantial modification to the pricing algorithm is required.

6. Heuristic algorithms

Two heuristic algorithms are introduced, one for each problem variant. Both algorithms are based on a three-step procedure: 1) Initial greedy solution; 2) Local search improvement; 3) MILP refinement. In particular, for a given number of times, steps 1 and 2 are executed and the information gathered during this process is forwarded to the MILP model of the associated problem variant. Each step is detailed in the following.

6.1. Heuristic algorithm for D2S$_1$

In the first step, an initial solution is found by a multi-start greedy method (GR1), as sketched in Algorithm 1. The idea is to have a set of nodes $S \subset H$, each one acting as the seed of the path it will belong to. S is randomly generated by sampling the set of nodes H, according to a uniform distribution, until n_S nodes have been selected. n_S has been calibrated as discussed in section 7.

At each execution, GR1 generates a set of disjoint paths T that eventually will compose a feasible walking bus arborescence rooted in 0. The construction phase operates in the outbound direction from school, i.e. paths originate from 0 and get reversed upon construction. At first $T = \emptyset$, then, at each iteration a new path is computed and added to T, as follows. a node $i \in S$ is sampled and removed from S to act as the first node after 0 of the current path $p = (0, i)$. Path p

is extended by appending one node at a time, if feasible with respect to constraints (10), selected (and removed) from $Q = H \setminus S$ according to a nearest neighbor criterion. When the path can no longer be extended, it is reversed, i.e., the last node in p becomes the line terminal, and added to T. The procedure iterates until S is empty. If $S = \emptyset$ and $Q \neq \emptyset$, then the next *seed* node is sampled from Q and the process iterates until each client node belongs to a path. Then T is returned. As each sampling introduces some randomness in the process, each run of the algorithm tends to yield different solution. Finally, the minimum number of drivers required to cover all paths in T is trivially computed and the incumbent T^* is updated if necessary, while $\bar{T} := \bar{T} \cup T$ collects all the paths generated so far. In the second step T is improved by a local search procedure (LS). In detail, LS operates on the terminal node i of each line and applies two different moves consisting of: i) given two lines, swap their terminal node; ii) remove the terminal node from a line (whose terminal node is updated) and append it to another line thus becoming its new terminal node.

Let $D_p = \sum_{i \in H_p} q_i$ denote the demand of all client nodes in p. Note that moves i) and ii) may reduce the number of drivers. For move ii) it happens when i is moved from path p to r, and $q_i \geq (D_p - \rho \lfloor D_p/\rho \rfloor)$, while $q_i < (\rho - (D_r - \rho \lfloor D_r/\rho \rfloor))$. In move i) the same condition must hold with respect to $q_i - q_j$, where i and j are the two swapping terminals. Indeed, such moves are performed first, if any, before looking for others that only impact on the second objective function component. The resulting neighborhood size is thus rather small, which allows to perform an exhaustive search and to run the LS procedure until convergence to a local optimum. $\bar{T}$ and T^* are updated accordingly. Finally, the $D2S_1$ MILP model (1 - 12) is solved on a restricted graph $G' = (N, A')$ where A' is made of all the arcs belonging to at least one path in $\bar{T}$, and with a time limit on the execution time (see section 7).

6.2. Heuristic algorithm for D2S$_2$

In $D2S_2$ the terminal stops are given and each path is built by iteratively adding extra stops within the two extremes of the line, i.e., the given terminal and node 0. Formally, consider the following sets: $\bar{H}$ the set of terminal nodes as described in section 4; $T = \{p(i) = (i, 0) \ \forall i \in \bar{H}\}$ as the set of the paths made of a terminal stop and node 0; $Q = H \setminus \bar{H}$ as the set of client nodes that are not yet part of a path.

Each path $p(i)$ has a residual capacity equal to $\bar{D}_{p(i)} = u_i \rho - D_{p(i)}$, and a time ratio $\Theta_{p(i)}$ given by the maximum over its nodes $j \in p(i)$ of the ratio of the walking time to destination along the

Algorithm 1: Greedy procedure for D2S$_1$.

Input : G, demand vector q, max deviation vector δ, walking time vector c, seed nodes S

Output: T, feasible solution to D2S$_1$.

$T \leftarrow \emptyset$;
$Q \leftarrow H \setminus S$; // set of non-seed nodes to be served
forall the $i \in S$ **do**
  $p \leftarrow (0, i)$, $S \leftarrow S \setminus \{i\}$; // start a new path
  if $Q = \emptyset$ **then**
    $T \leftarrow T \cup p$; // add path to T
  while $Q \neq \emptyset$ **do** // until all nodes are served
    $f \leftarrow False$; // flag
    forall the $j : (ij) \in \mathcal{FS}(i)$ **do**
      if $\Theta'_{p \cup j} \leq \Delta_j$ **then** // $\Theta'_{p \cup j}$ computes $\Theta_{p \cup j}$ for path $p \cup j$ and reversed
        $p \leftarrow append(p, j)$, $Q \leftarrow Q \setminus j$; // update p and Q
        $f \leftarrow True$, $i \leftarrow j$;
        break;
    if $f = False$ **then** // close the path
      $T \leftarrow T \cup p$; // add path to T
    if $S = \emptyset \wedge Q \neq \emptyset$ **then** // start path from non-seed
      $i \leftarrow first(Q)$; // first node in Q assigned to i
      $p \leftarrow (0, i)$; // new path
return T

path and S_j. In particular, at start $\bar{D}_{p(i)} = u_i\rho - q_i$ and $\Theta_{p(i)} = 1$. The greedy procedure GR2 iteratively inserts a client node j in Q within a path $p(i)$ right before node 0. One such pair $(j, p(i))$ is chosen randomly among those that minimally increase $\Theta = max_{i \in \bar{H}} \Theta_{p(i)}$, so that successive runs of the procedure are likely to yield different outcomes. In detail, given a path $p(i)$, the candidate node j is the first one whose insertion does not increase $\Theta_{p(i)}$ more than a randomized threshold, computed drawing samples from a uniform distribution between 1 and 1.2 (function $Rand(1, 1.2)$ in Algorithm 2), provided that $q_j \leq \bar{D}_{p(i)}$. Then, the node-path pair with the minimum increase

over all paths is selected. Each macro-iteration either yields an insertion and iterates or exits the procedure with failure if $Q \neq \emptyset$. The algorithm for D2S$_2$ is sketched in Algorithm 2 where $p(i) \cup \{j\}$ represents the path obtained by inserting node j in $p(i)$ right before 0. In the second step of the algorithm, each path in T is improved by a local search procedure whose move swaps two internal nodes of the path, i.e., any two nodes but 0 and the terminal. Step 1 and 2 are executed several times in order to generate multiple solutions whose paths are collected in $\bar{T}$. In the last step, the incumbent solution is refined by solving the MILP associated with D2S$_2$ (see 15 - 19) on a restricted graph $G' = (N, A')$ where A' is made of all the arcs in the paths in $\bar{T}$ and their reverse. Again, the solver running time is limited. It is worth mentioning that, although the constructive procedure does not contemplate line merging, on our benchmark it never failed to provide feasible solutions, as reported in section 7.

Algorithm 2: Greedy procedure for D2S$_2$.

Input : G, demand vector q, walking time vector c, terminal nodes $\bar{H}$, residual clients Q, initial solution $T = \{p(i) = (i0)\ \forall i \in \bar{H}\}$

Output: Θ, T

1 $\Theta \leftarrow 0$; // solution value

2 **while** $|Q| > 0$ **do** // all demand is served

3 $\quad \Theta^* \leftarrow \infty$, $j^* \leftarrow 0$, $i^* \leftarrow 0$;

4 $\quad$ **forall the** $i \in \bar{H}$ **do**

5 $\quad\quad$ **forall the** $j \in Q$ **do**

6 $\quad\quad\quad$ **if** $q_j \leq \bar{D}_{p(i)}$ **then**

7 $\quad\quad\quad\quad \Theta_p \leftarrow \Theta(p(i) \cup \{j\}, c)$; // compute Θ_p of $p(i) \cup \{j\}$

8 $\quad\quad\quad\quad$ **if** $\Theta_p < \Theta_{p(i)} \times Rand(1, 1.2)$ **then**

9 $\quad\quad\quad\quad\quad$ **break**;

10 $\quad\quad$ **if** $\Theta^* > \Theta_p$ **then**

11 $\quad\quad\quad j^* \leftarrow j$, $i^* \leftarrow i$, $\Theta^* \leftarrow \Theta_p$;

12 $\quad$ **if** $j^* = 0$ **then** // Manage pathological cases

13 $\quad\quad$ **return** Θ, T

14 $\quad \Theta \leftarrow max(\Theta, \Theta^*)$; $p(i^*) \leftarrow p(i^*) \cup \{j^*\}$; $Q \leftarrow Q \setminus \{j^*\}$;

15 **return** Θ, T

7. Computational results

To assess effectiveness and efficiency of our approaches we performed a test campaign on several instances. The description of the test beds and a discussion of the obtained results are reported in the following sections.

7.1. Dataset

Our dataset is made up of three different types of instances:

1. **Random instances**: 80 instances, n from 10 to 300, q_i randomly generated between 1 and 3. The topology of these instances and the weight coefficients d_{ij} are randomly generated using a uniform distribution function in the range $[0.01, 0.30]$.

2. **School Bus Problem instances**: 72 instances are from Schittekat et al. (2013). We considered all instances with n ranging from 25 to 200, q_i randomly generated from 1 to 3. We kept the topology as in the original instances but ignored all potential bus stop locations and generated weight coefficients d_{ij} following the same scheme as in the instance set 1.

3. **Real-world instances**: two real scenarios with $n = 32$ and $n = 116$, corresponding to a total of 35 and 133 children respectively. Data for both scenarios come from the elementary school Biagio Rossetti in Ferrara (Italy). For each arc $(ij) \in A$ weight coefficient d_{ij} has been assumed to be inversely proportional to the pedestrian speed along the path (ij). In details, given Ω_{ij} as the speed on arc (ij) and Ω^M as the maximal speed among all arcs in the graph then d_{ij} is equal to $1 - \frac{\Omega_{ij}}{\Omega^M}$. In these instances all children locations are real GPS positions. Distances and speeds are computed using *OpenStreetMap* (OpenStreetMap contributors, 2017) as real walking distances and speeds. Starting from these scenarios we have generated 16 instances with different input parameters.

In all instances the nodes are divided in three tiers $T1$, $T2$, $T3$, depending on their distance from school. In particular, considered S^m and S^M as the distance to school from the closest and from the furthest students and computed $S^T = (S^M - S^m)/3$, then the tiers are defined as follows:

$$T1 = \{i \in H : S^m \leq S_i \leq S^T\}.$$

$$T2 = \{i \in H : S^T < S_i \leq 2 \times S^T\}.$$

$$T3 = \{i \in H : 2 \times S^T < S_i \leq S^M\}.$$

For problem D2S$_1$ we considered two values for ρ: 5 and 10 and 4 different values δ': 0.1, 0.2, 0.5, 1.0. In order to prevent children from walking long distances, as described in section 2, we set for each node $i \in H$ the value of δ_i taking into consideration δ' and the tiers. In particular, for node i in $T1$ $\delta_i = 1 + \delta'$, for i in $T2$ $\delta_i = 1 + 0.7\delta'$ and finally for i in $T3$ $\delta_i = 1 + 0.4\delta'$.

For problem D2S$_2$ we considered the same value for ρ as in D2S$_1$. Moreover, to set the initial locations of the drivers (parameter u in the model) we computed a lower bound on the number of required drivers as $lb = \lceil \frac{\sum_{i \in H} q_i}{\rho} \rceil$ and then considering four different coefficients v: 1.1, 1.2, 1.5, 1.8 we have randomly distributed vlb drivers among all nodes in $T3$. In the real word scenarios, we also considered real locations of potential volunteers in order to define the values of u.

The trade-off parameter ϵ for D2S$_1$ is equal to 0.1 for instances with $n \leq 30$ and it is equal to 0.01 for all other instances. With such values the secondary objective function does not interfere with the primary one since its value is always lower than 1. In D2S$_2$ ϵ is equal to 10^{-5} for all instances, in this way the contribution of the secondary objective is always lower than 0.001.

7.2. Implementation Details

All algorithms presented in this paper have been implemented in Python 2.7 using Pyomo 5.5 as optimization modeling language (Hart et al., 2017) and GUROBI 8.0 as LP an MIP solver (Gurobi Optimization, 2018). All tests have been carried out using a computer equipped with Intel i7-6700K processor, 16GB of RAM and running Ubuntu Linux 16.04.

In a preliminary testing phase the best setting for the heuristic algorithms (see sections 6.1 and 6.2) have been identified. In particular, when solving D2S$_1$ we run phases one and two $1 + \lceil n/10 \rceil$ times for each instance and we use $n_s = 1 + \lceil \frac{\sum_{i \in H} q_i}{\rho}(1 - \delta') \rceil$ in the greedy procedure. The time-limit on the third phase is equal to $n/2$ seconds. The first and second phases of the heuristic algorithm for D2S$_2$ have been run twenty times, each time with a different random seed while the time-limit on the third phase is set to n seconds. The time-limit on the total computational time is 7200 seconds.

Cuts separation. In compact models described in section 4, subtour elimination constraints, capacity cuts and k-path inequalities can be included to improve the value of the linear relaxation. In our implementation, subtour elimination constraints on binary variables y_{ij} are separated using a max-flow-based procedure inspired by Fischetti & Toth (1997). Capacity cuts impose both the

connectivity of the solution and the capacity requirements. In order to separate them we start from the values x^*_{ij} of the variables x_{ij} in the optimal solution of the linear relaxation of the arc-based model. Then we generate several sets of nodes C using multiple heuristic procedures based on Naddef & Rinaldi (2002) and Lysgaard et al. (2004) and we check whether

$$\sum_{(i,j):i\in C, j\notin N\setminus C} x^*_{ij} < \left\lceil \frac{\sum_{i\in C} q_i}{\rho} \right\rceil \tag{47}$$

then we impose

$$\sum_{(i,j):i\in C, j\notin N\setminus C} x_{ij} >= \left\lceil \frac{\sum_{i\in C} q_i}{\rho} \right\rceil \tag{48}$$

Note that if $\left\lceil \frac{\sum_{i\in H} q_i}{\rho} \right\rceil = 1$ inequality (48) represents a subtour elimination constraint and can also be imposed on binary variables y_{ij}. Finally, for each set of nodes $C : |C| \leq 15$ that did not yield a capacity cut, we look for violated k-path inequalities. We solve the arc-based model (1 - 14) on a restricted graph where only nodes belonging to set $C \cup \{0\}$ are considered (using Θ^{ub} instead of Δ in D2S$_2$). Then, we use the value of the first component of the objective function (1) in the optimal solution as the right-hand side for (47) and (48). If (47) holds then (48) represents a valid cut.

In our procedure, all violated cuts found are included in the formulation. Then the new linear relaxation of the model is solved and we look for violated cuts again. We stop when no effective cut is found or when a time-limit equal to n in reached.

7.3. D2S$_1$ Results

The arc-based and the path-based approaches have been tested on all 152 instances in test sets 1 and 2 using the solution found by the heuristic algorithm (see section 6.1) as starting point for the optimization.

The arc-based model is able to achieve the optimal solution in 62 instances, 32 random instances and 30 school bus instances. Results are particularity good on small instances, indeed all instances with $n \leq 30$ are solved to optimality. However, the optimal solution is found for only 12 instances with $n \geq 80$. In 90 instances the time-limit is reached and the final MIP gap is on average equal to 8.71%, the average MIP gap being especially large (> 14%) on instances with $n \geq 100$, $\delta' \geq 0.5$ and $\rho = 10$. All such cases are characterized by an initial poor linear relaxation at the root node whose value does not improve much (less than 1% on average) with the inclusion of additional cuts.

The column generation procedure reaches its natural end on all instances with $n \leq 50$ and hits the time-limit in 58 instances, 33 random instances and 25 school bus instances. The average gap between the optimal continuous solution of the RMP and the best integer solution found is 9.18%. However, in almost half of the instances the gap is below 5%. The average number of branching nodes to explore is 3.73. It is worth noting that on instances with $n \leq 150$ the number of branching nodes is always lower than 9 and that in 44 instances this value is either 0 or 1.

Analyzing the results, it turns out that most of the computational time is spent solving the RMP or the final MILP problems while the pricing phase usually takes less than 10% of the computational time. This behavior is due to the fact that the decomposition is not particularly efficient since all requirements associated with ρ (students per driver ratio), the structure of the network, and the children flow are addressed in the RMP.

Comparing the arc-based and the path-based models, there is no ultimate winner. On the one hand, the arc-based model provides better solutions in 110 instances but the average improvement, with respect to the column generation, is about 2% and in only 14 instances the improvement is above 5%. On the other hand, the path-based model is able to find better solutions in only 12 instances but the improvement is on average larger than 7%. All such instances are characterized by $n \geq 50$, $\delta' \geq 0.5$ and $\rho = 10$ that is basically the same type of instances with large final MIP gap in the arc-based model. Comparing the optimal value of the linear relaxation at the root node in both methods we can see that the path model dominates when additional cuts are not considered in the arc-based model. Indeed, it provides a value that is on average 5.32% better and in 25 instances the difference is larger than 10%. However, the inclusion of the cuts improves, on average, the linear relaxation of the arc model by more than 3.5%, reducing the gap from the linear relaxation of the path model to less than 1%. It must be noted that, although in 57 instances cuts are very effective, increasing the linear relaxation by almost 10%, in 53 instances the inclusion of cuts does not improve the value of the linear relaxation and in 42 instances no effective cut can be found.

The behavior of both models is similar on both sets of instances and it is not possible to find any clear trend that differentiates one set from the other.

The input parameters ρ and δ' influence the solution process in similar ways. Increasing ρ always makes the problem more difficult to solve and indeed the majority of the instances with

large MIP gap have $\rho = 10$. These instances are also characterized by the largest pricing time since many pricing iterations are required for the column generation process to converge. The influence of δ' is similar but less noticeable. Higher values of δ' lead to poor linear relaxations and to pricing sub-problems that take more time to be solved in the path-based model. In details, the average pricing time per iteration increases from 9 seconds in instances with $\delta' = 0.1$ to 68 seconds in instances with $\delta' = 1.0$. On average, instances with larger δ' take longer to be solved but feasible solutions can be found sooner; this tends to increase the convergence rate of the column generation (e.g. see results on instances R079, R080, B046, B072, R047, R055, R056 in the supplementary material available online).

Finally, as regard as the quality of the heuristic algorithm (section 6.1), it provides solutions that are on average within 5% from the best solution found. On almost half of the instances the algorithm is able to generate a solution within 1% from the best found. Moreover, whether only the first component of the objective function is considered, the heuristic algorithm finds the optimal solution in 66 instances. These solutions are equally spread among small and large instances with any value for δ' and ρ. It is worth noting that the third step of the heuristic is very effective in enhancing the quality of the solutions. Indeed, it provides an average improvement of about 14% with respect to the solutions found at the end of the second step. This improvement is more noticeable, about 20%, for large instances with $n \geq 200$.

Detailed results are reported in the supplementary material available online. See tables 3, 4, 5 and 6 for results on all instances and figures 1 and 2 for a graphical comparison of the solution quality.

7.4. D2S$_2$ Results

In the testing campaign for D2S$_2$ the arc-based model was able to find the optimal solution in 77 instances with an average computational time equal to 513 seconds. In particular, all instances with $n \leq 30$ have been solved to optimality. In 75 instances the execution reaches the time-limit. Near optimal solutions (MIP gap lower than 1%) are achieved in 10 instances while in 36 instances the computation terminates with residual MIP gap larger than 20%. On average, the value of the linear relaxation is equal to 89% of the best-known solution and in 33 instances the value of the linear relaxation and the value of the optimal solution are the same. It is worth noting that, on both sets of instances, very few cuts have been found and that they do not have any impact on

the value of the linear relaxation. Since in all instances there are more adults on the lines than the minimum required, the capacity constraints are rarely violated in the linear relaxation.

Input parameters υ and ρ greatly affect the difficulty of the problem and consequently the performance of the arc-based model. On the one hand, the higher is υ the easier is the problem to solve. With high υ there are more drivers and more, shorter lines. In fact, the model is able to solve 75% of instances with $\upsilon = 1.8$ but only 30% of instances with $\upsilon = 1.1$. On the other hand, increasing ρ makes the problem more difficult since longer lines have to be taken into account and less drivers are provided (see 7.1). The arc-based model achieves optimality in 68% of the instances with $\rho = 5$ and in only 35% of the instances with $\rho = 10$. Combining the effect of υ and ρ we can identify two distinct classes of instances: easy instances with $\upsilon \geq 1.5$ and $\rho = 5$, which are solved to optimality 87% of the times, and hard instances $\upsilon \leq 1.2$ and $\rho = 10$ in which the optimal solution is reached in only 21% of the instances.

As regard as the path-based model, the column generation procedure terminates within the time-limit in 126 instances. The gap between the optimal continuous solution of the RMP at root node and the best-known integer solution is on average 1.89%. In 47 instances the value of the linear relaxation is equal to the best-known solution. Analyzing how the computational time is split between the solution of the RMP and the pricing sub-problems we notice that on average, almost 60% of the computational time is spent by the pricing. It is worth noting that this percentage is substantially higher in $D2S_2$ than in $D2S_1$. This is mainly due to three reasons, i) in the pricing algorithm of $D2S_2$ paths basically have no a-priori maximum duration and infinite capacity. We provide a limitation on the duration using the value of the heuristic solution, but this limitation can be loose depending on the instance and on the quality of the heuristic procedure. Moreover, since the value of best Θ is on average 30% larger than the average δ in $D2S_1$ the paths to be generated are 30% longer in $D2S_2$. ii) The almost flat landscape of bottleneck functions, such as the main objective function component of $D2S_2$, typically poses a challenge to column generation based solution approaches. Indeed, in several initial iterations of the column generation process many columns with negative reduced cost are found but the value of the linear relaxation does not change. iii) Removing columns with $\Theta_p > \Theta$ every time the primal heuristic improves the incumbent limits the time spent solving the RMP.

Comparing the results obtained by both models, we observed that on average, the arc-based

model achieves better solutions. The value of these solutions is about 4.61% lower than the ones obtained by solving the path-based model. However, the column generation procedure takes on average less than 50% of the computational time of the arc-based model. Moreover, in 14 instances it is able to find a better solution within the time-limit with an average improvement equal to 6.64%, and almost all of these instances belong to the hard class.

Finally, the heuristic algorithm finds solutions that are on average within 10% from the best-known one. This gap is smaller, about 5%, on instances with less than 150 nodes while it is about 17% on larger ones. In 22 instances it finds the optimal solution. In our test, the heuristic procedure is always able to find feasible solutions after the first two steps. These solutions are greatly enhanced in the last step where the MILP model is solved on a restricted graph: the average improvement from step two is 40.51%. On 22 instances the solution improves by more than 60% and only on 5 instances the third step has little effect on the solution quality (improvement below 1%). It is worth noting that, although the arc-based model is largely influenced by input parameters, the heuristic algorithm is not and indeed the quality of the solutions found is not much sensitive to changes in υ and ρ.

Detailed results on all instances are reported in the supplementary material available online. See tables 7, 8, 9 and 10 for a complete list of the results and figures 3 and 4 for a graphical comparison of the performance achieved by the heuristic algorithm, the arc-based model and the column generation procedure on test sets 1 and 2.

7.5. Results on Real-Word Scenarios

In order to test our approach in a practical context, we run our algorithms on test set number 3 whose instances come from real-world scenarios.

From a pure performance point of view the behavior of the algorithms on test set 3 is very similar to the one on test sets 1 and 2. The same can be said on the average quality of the solutions found.

Results for $D2S_1$ show that, in designing a Pedibus service network, the value of δ' is the most critical one and that in small networks changing ρ is almost not influential. Indeed, with a small δ' setting up a Pedibus network may require more than twice as many drivers as the theoretical minimum amount lb. Increasing δ' from 0.1 to 1.0 reduces the number of drivers by almost 40%. This increase in the value of δ' corresponds to an increase in the maximum allowed deviation from

the shortest path equal to 700 meters for children in T1. However, even with $\delta' = 1.0$ no student in our solutions walks more than 2.4 km.

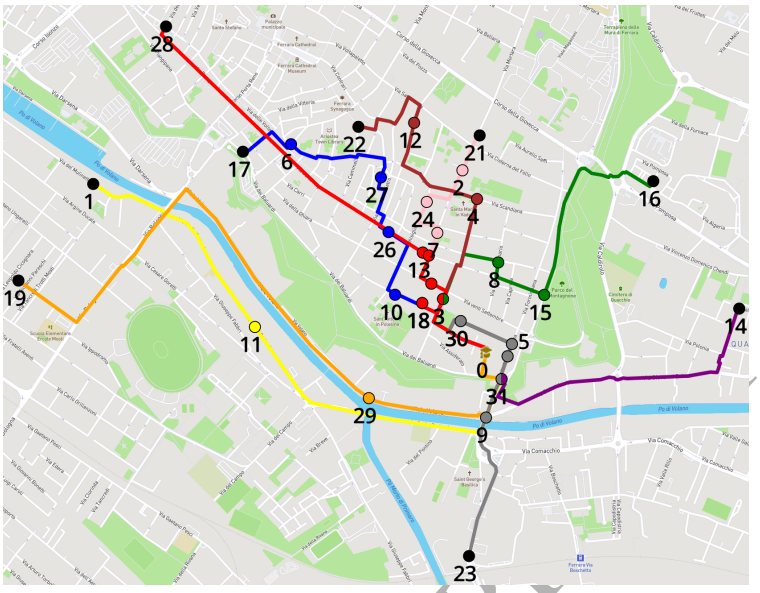

Figure 2: Instance W007, lines red and green merge at node 3, lines gray and purple merge at node 31

Figure 2 shows the solution for D2S$_1$ on a real-world instance with 32 nodes (35 children, 2 children in locations 5, 8 and 23), $\delta' = 1.0$ and $\rho = 5$. Nine drivers are used, one for each line. The initial locations of the drivers are marked by black nodes. Two line mergings are performed to reduce the second component of the objective function: the red line merges with the green one at node 3 while the gray and the purple lines join together at node 31.

For D2S$_2$ we performed two distinct batches of tests, one with drivers randomly located over the nodes in T3 and another using real locations of potential volunteers. Performance-wise differences between the two tests are minimal. On both tests the MILP model was able to reach the optimal solution in all instances with $n = 32$ and in instances with $n = 116$ and $v \geq 1.5$. The column generation procedure terminates within the time-limit in all instances but one.

Looking at the quality and structure of the solutions, the differences are more evident. The position of drivers is crucial in order to achieve a good quality solution, in particular when only few drivers are available ($v \leq 1.5$, $\rho = 10$). The improvement in the first component of the objective function Θ is on average 30% when drivers are selected in T3. This may lead to situations, like instance W004, where, when volunteers' locations are given, children walk on average 2.6 Km,

while with drivers located in T3 they walk only 1.4 Km on average.

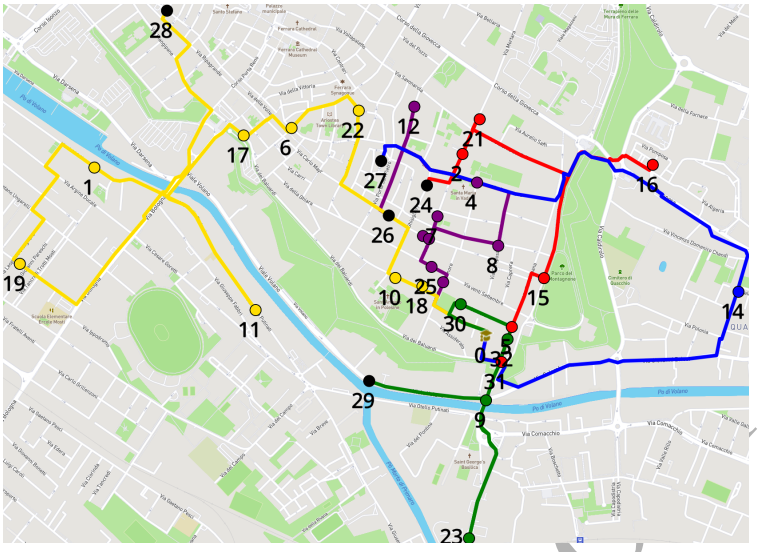

Figure 3: Instance W004: real volunteers' locations $\Theta = 3.22$.

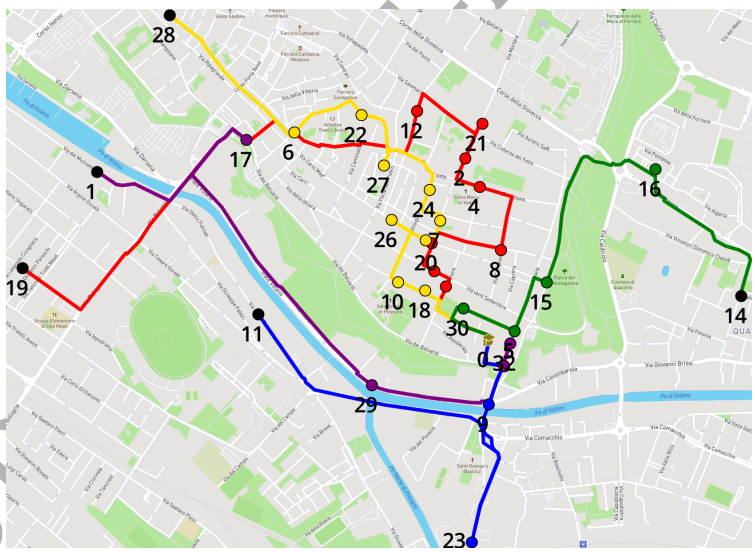

Figure 4: Instance W004: drivers selected in T3 $\Theta = 1.73$.

A comparison of the results obtained on instance W004 is depicted in Figures 3 and 4. In this instance there are 5 drivers located on the black nodes on the map, $v = 1.2$ and $\rho = 10$. Three of the volunteers live within 700 from school (T1) and so they must take a long detour from their shortest path in order to pick up children on their way to school. This, considered the min/max structure of

the objective function, has a detrimental effect on the yellow line. Indeed, as its driver lives about 1.9 km from school and $\Theta = 3.22$, the driver could walk up to 6.2 Km without any impact on the objective function. Clearly, in order to provide an acceptable solution for this instance either more drivers or different starting locations must be considered. If randomly selected drivers in T3 are used instead of volunteers, the solution (see Figure 4) is far more reasonable; indeed, as $\Theta = 1.73$, the driver of the yellow line will not walk more than 3.2 Km.

Finally, it is worth noting that heuristic solutions are on average within 10% from the best ones. This corresponds to an increase of less than 100 meters in the average walking distance and it is obtained within a fraction of the computational time required by both models.

The complete list of the results achieved on real-world scenarios for $D2S_1$ and $D2S_2$ is reported in the supplementary material available online (see tables 11, 12 and 13).

8. Conclusions

This paper studies the problem of designing walking bus lines according to a new paradigm. The walking bus analyzed here is a proposal born within a project funded by the Italian Environment Department to shift rides from private cars to more sustainable modes in the trip from home to school. The idea diverges from the classical walking bus services since it produces Door to School itineraries that can potentially attract more users. The benefit of having efficient Door to School lines is twofold. In the short term, it reduces traffic congestion in front of the school, with positive effects on safety and health. In the long term, it induces a behavioral change and raises the sustainable mobility awareness of next generation citizens.

We showed that the peculiarities of the problem make it different from previously studied problems, such as the school bus line planning. The two variants of the problem take into account specific practical settings arising from the case study. In one case the main objective is minimizing the number of drivers, which is a major concern when the cost for the accompanying persons is in charge of the school administration. Since it is a walking service, there is an explicit constraint on the maximum deviations from the walking time of the shortest itinerary for each child. The second case arises when the drivers are volunteers. Thus, the drivers and their starting locations are given and the objective is to minimize the maximum deviation from the shortest path walking time. In both variants, the feature that makes the problem interesting from a combinatorial optimization

point of view is the upper bound on the number of children per driver. Indeed, this constraint makes it profitable to merge lines at some intermediate point and join residual capacity.

In order to exploit this feature we proposed three different approaches. The first is an arc model that is fed to a commercial solver. The second one is a path model that is tackled with a column generation approach. The third one is a heuristic procedure that identifies a subgraph on which the arc model can be quickly solved. While it has been possible to apply all the three approaches to the first variant of the problem, the min-max nature of the objective function of the second one makes the pricing sub-problem particularly inefficient to be solved, as confirmed by the poor performance of the column generation based solution approach.

The outcomes of the experiments are interesting. Though there is not a clear dominance regarding solution quality, the application setting could dictate the best solution strategy to adopt according to the available computing time. Every school year the lines should be pre-planned to forecast the need for drivers. In that case running time is not that critical and a few hours can be devoted to the search for a good solution. However, the Door to School service tends to be extremely dynamic regarding children and drivers as well. It could be necessary to recompute the lines on a daily base, when real time information becomes available. In such a case the heuristic approach provides a flexible and effective tool to recompute the lines affected by the changes. Not only demand and available drivers may vary, though; updated information concerning walkability may arise from children and drivers reporting changing safety conditions along the lines, thus triggering a modification on the walkability parameters of the mathematical models. This implies that either a general solution is re-optimized to consider the changes, or new lines are recomputed from scratch. Having a variety of methods allows the planner to choose each time the most suitable one, knowing the size of the problem, the actual demand, and the available human and material resources.

References

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press. ISBN: 9780691129938.

Baldacci, R., Hadjiconstantinou, E., & Mingozzi, A. (2004). An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, *52*, 723–738.

Bektaş, T., & Lysgaard, J. (2015). Optimal vehicle routing with lower and upper bounds on route durations. *Networks*, *65*, 166–179.

Chillón, P., Evenson, K. R., Vaughn, A., & Ward, D. S. (2011). A systematic review of interventions for promoting active transportation to school. *International Journal of Behavioral Nutrition and Physical Activity*, *8*.

Desrochers, M., & Laporte, G. (1991). Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, *10*, 27 – 36.

Dijk, M. L. V., Groot, R. H. D., Acker, F. V., Savelberg, H. H., & Kirschner, P. A. (2014). Active commuting to school, cognitive performance, and academic achievement: an observational study in dutch adolescents using accelerometers. *BMC public health*, *14*.

Fernandes, L. M., & Gouveia, L. (1998). Minimal spanning trees with a constraint on the number of leaves. *European Journal of Operational Research*, *104*, 250 – 261.

Fischetti, M., & Toth, P. (1997). A polyhedral approach to the asymmetric traveling salesman problem. *Management Science*, *43*, 1520–1536.

Gamvros, I., Golden, B., & Raghavan, S. (2006). The multilevel capacitated minimum spanning tree problem. *INFORMS Journal on Computing*, *18*, 348 – 365.

Gouveia, L., & Simonetti, L. (2017). Spanning trees with a constraint on the number of leaves. a new formulation. *Computers & Operations Research*, *81*, 257 – 268.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual. Last visited 2018-10-31.

Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., & Siirola, J. D. (2017). *Pyomo–optimization modeling in python* volume 67. (2nd ed.). Springer Science & Business Media. ISBN: 9781461432265.

Irnich, S., & Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation* (pp. 33 – 65). Boston, MA: Springer US.

Kara, I. (2010). On the millertuckerzemlin based formulations for the distance constrained vehicle routing problems. *AIP Conference Proceedings*, *1309*, 551–561.

Kara, I., & Derya, T. (2011). Polynomial size formulations for the distance and capacity constrained vehicle routing problem. *AIP Conference Proceedings*, *1389*, 1713–1718.

Kingham, S., & Ussher, S. (2007). An assessment of the benefits of the walking school bus in christchurch, new zealand. *Transportation Research Part A: Policy and Practice*, *41*, 502 – 510.

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., & Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, *33*, 101–116.

Lambiase, M., Barry, H., & Roemmich, J. (2010). Effect of a simulated active commute to school on cardiovascular stress reactivity. *Medicine & Science in Sports & Exercise*, *42*, 1609 – 1616.

Laporte, G., Nobert, Y., & Taillefer, S. (1987). A branch-and-bound algorithm for the asymmetrical distance-constrained vehicle routing problem. *Mathematical Modelling*, *9*, 857 – 868.

Lozano, L., Duque, D., & Medaglia, A. L. (2016). An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, *50*, 348 – 357.

Lysgaard, J., Letchford, A., & Eglese, R. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Program. Ser. A*, *100*, 423 – 445.

Malucelli, F., Nonato, M., & Tresoldi, E. (2017). Optimization based planning of pedibus lines: an arc based

approach. *Transportation Research Procedia*, *27*, 760 – 767.

Malucelli, F., Tresoldi, E., & Nonato, M. (2018). Designing pedibus lines: a path based approach. *Electronic Notes in Discrete Mathematics*, *69*, 149 – 156.

McDonald, N. C., & Aalborg, A. E. (2009). Why parents drive children to school: Implications for safe routes to school programs. *Journal of the American Planning Association*, *75*, 331 – 342.

McDonald, N. C., Brown, A. L., Marchetti, L. M., & Pedroso, M. S. (2011). U.s. school travel, 2009: An assessment of trends. *American Journal of Preventive Medicine*, *41*, 146 – 151.

Mendoza, J. A., Watson, K., Chen, T.-A., Baranowski, T., Nicklas, T. A., Uscanga, D. K., & Hanfling, M. J. (2012). Impact of a pilot walking school bus intervention on children's pedestrian safety behaviors: A pilot study. *Health & Place*, *18*, 24 – 30. Active Living Research.

Naddef, D., & Rinaldi, G. (2002). Branch-and-cut algorithms for the capacitated vrp. In *The Vehicle Routing Problem* chapter 3. (pp. 53–84).

OpenStreetMap contributors (2017). Planet dump retrieved from https://planet.osm.org. Last visited 2018-10-31.

Porro, P. (2015). *Sviluppo di un algoritmo di ottimizzazione dei percorsi in un servizio Pedibus*. Master's thesis Politecnico di Milano Milano, Italy. Last visited 2018-10-31.

Riera-Ledesma, J., & Salazar-González, J. J. (2013). A column generation approach for a school bus routing problem with resource constraints. *Computers & Operations Research*, *40*, 566 – 583.

Salama, H. F., Reeves, D. S., & Viniotis, Y. (1997). The delay-constrained minimum spanning tree problem. In *Proceedings Second IEEE Symposium on Computer and Communications* (pp. 699 – 703).

Schittekat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., & Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, *229*, 518 – 528.

Sunyer, J., & et al. (2017). Traffic-related air pollution and attention in primary school children: Short-term association. *Epidemiology*, *28*, 181 – 189.

Tanaka, K., Miyashiro, R., & Miyamoto, Y. (2016). A bi-objective optimization model for designing safe walking routes for school children. *Geographical Analysis*, *48*, 448 – 464.

Tanaka, K., Miyashiro, R., & Miyamoto, Y. (2018). A layered network formulation for the safe walking route design problem. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *12*.

Westman, J., Friman, M., & Olsson, L. (2017). What drives them to drive? Parents' reasons for choosing the car to take their children to school. *Frontiers in psychology*, *8*.