Designing Pedibus Lines: a Path Based Approach

Federico Malucelli, Emanuele Tresoldi [1]

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy

Maddalena Nonato

Dipartimento di Ingegneria
Università degli Studi di Ferrara
Ferrara, Italy

Abstract

We study the problem of designing walking school bus lines (Pedibus) limiting the deviation with respect to the shortest path for each child, with the objective of minimizing the number of accompanying persons and the perceived risk of the selected trajectories. The problem is formulated using a path model and a column generation approach is proposed. Computational experiments compare the lower bounds and the solutions of the proposed approach with the arc model and a simple heuristic proposed in a previous work.

Keywords: Walking School Bus, Minimum Leaf Spanning Tree with Limited Deviations, Column Generation.

[1] Email: `emanuele.tresoldi@polimi.it`

1 Introduction

Pedibus lines are an effective tool to educate young generations to an environmental sustainable behavior and to decrease the traffic congestion and air pollution in urban areas. Helping in designing efficient Pedibus lines is of great importance to propose an attractive system able to reach the objective. The problem can be described as follows: given the school location, the children home addresses, and the distance between each pair of locations, find the minimum number of lines rooted at the school so that each location belongs to one line and the distance from school to each location along the line is below a given threshold that depends on the length of the actual shortest path. The objective function has two components: on the one hand it minimizes the number of supervising adults (one per line), on the other hand it minimizes the perceived risk by favoring the merging of lines. We refer to this problem as the Minimum Leaf Pedibus Line design Problem ($MLPLP$). Looking at the problem in an abstract way, it recalls a minimum leaf spanning tree with a limit on the distance of each node from the root.

The Pedibus line design problem is usually approached in practice with common sense solutions, which are suitable in small contexts but are not able to manage the system on a large scale. It has some similarities with the well studied School Bus Routing problem, though peculiarities such as the tight constraint on deviations and the merging of lines make it different and justify a specific study to take advantage of them. The problem in this form is proposed and studied in [5] where a brief survey on the related literature is presented. That work proposes a simple arc model that fails to solve instances with more than 50 pupils and a simple heuristic that behaves reasonably well on large instances. In this work we build upon that experience and we directly compare with those results.

We propose an alternative path-based formulation. For that formulation we describe a Column Generation approach to solve optimally the linear relaxation of the problem and generate a heuristic solution by solving the corresponding integer version of the master problem. The lower bounds and the feasible solutions obtained with this approach are compared with the solutions obtained with the simple arc model and the heuristic algorithm proposed in [5].

2 Problem definition and path-based model for $MLPLP$

The problem can be described considering a directed graph $G = (\mathcal{N}, \mathcal{A})$. The nodes in $\mathcal{N}$ represent walking-bus stops, that is children homes and the school (denoted by index 0 and called root), and the arcs in $\mathcal{A}$ correspond to shortest path connections between nodes. A coefficient c_{ij} associated with every arc $(i, j) \in \mathcal{A}$ gives the length of the shortest path from i to j, thus c_{i0} gives the length of the "ideal" path from node i to the root. Another coefficient d_{ij} gives the measure of the perceived risk traveling from i to j, or alternatively the cost to adapt the trajectory from i to j to a given standard to avoid risks. Given a coefficient $\delta > 1$, the problem consists in finding an arborescence routed in 0 and spanning all nodes in $\mathcal{N} \setminus \{0\}$ such that considering the path going from any node i to 0 in the arborescence, its length is less than or equal to δ times the length of the ideal path. The objective is to minimize the number of leaves and also the sum of coefficients d_{ij} for all arcs in the arborescence. Note that, in this representation, the leaves correspond to the accompanying persons and the paths converging to the root correspond to Pedibus lines. Lines can merge in some intermediate nodes to minimize the second term of the objective function.

The path model that we propose is based on the set $\mathcal{P}$ of all feasible paths. A path p from any node to the root 0 is feasible if it is elementary (no node is visited more than once) and if for each node i in the path the length c^p_{i0} to go from node i to 0 following the path is not longer than δc_{i0}. To complete the notation let $\mathcal{P}_i \subset \mathcal{P}$ represent the subset of all paths visiting node $i \in \mathcal{N}$, $\mathcal{P}_{ij} \subset \mathcal{P}$ the subset of all paths using arc $(i, j) \in A$ and $\bar{\mathcal{P}}_i \subset \mathcal{P}$ the subset of all paths starting from node $i \in \mathcal{N}$.

The model makes use of two sets of binary variables: x_p equal to 1 if path $p \in \mathcal{P}$ is selected and y_{ij} equal to 1 if the arc $\forall (i, j) \in \mathcal{A}$ is used. The complete formulation for the path-based MILP is given in (1-7).

The objective function (1) is made of two parts: minimization of the number of the selected paths, this corresponds to the minimization of the number of leaves, and minimization of total risk/cost multiplied by a suitable trade-off parameter ϵ.

Constraint (2) imposes that all nodes have to be visited at least once. Inequality (3) defines the activation constraint for y_{ij} variables. Constraint (4) limits the number of paths originating from a leaf to one. Finally, the last constraint (5) imposes that for each node but 0 a single outgoing arc must be selected, meaning that if two paths reach the same node they are merged together.

$$
\begin{aligned}
&\min \sum_{p \in \mathcal{P}} x_p + \epsilon \sum_{(i,j) \in \mathcal{A}} d_{ij} y_{ij} && && (1) \\
&-\sum_{p \in P_i} x_p \leq -1 && \forall i \in \mathcal{N} && (2) \\
&\sum_{p \in P_{ij}} x_p - |\mathcal{N}| y_{ij} \leq 0 && \forall (i,j) \in \mathcal{A} && (3) \\
&\sum_{i \in \bar{\mathcal{P}}_i} x_p \leq 1 && \forall i \in \mathcal{N} \setminus \{0\} && (4) \\
&\sum_{(i,j) \in \mathcal{A}} y_{ij} = 1 && \forall i \in \mathcal{N} \setminus \{0\} && (5) \\
&x_p \in \{0,1\} && \forall p \in \mathcal{P} && (6) \\
&y_{ij} \in \{0,1\} && \forall (i,j) \in \mathcal{A}. && (7)
\end{aligned}
$$

Since the number of paths in $\mathcal{P}$ is exponential in $|\mathcal{N}|$ and we cannot explicitly generate all variables x_p even for relatively small graphs, our solution approach is based on the column generation paradigm (see [1]).

The restricted master problem (RMP), given by the linear relaxation of (1-7), is initialized with either a set of $|\mathcal{N}| - 1$ dummy columns representing the shortest paths from any node in $\mathcal{N} \setminus \{0\}$ to 0 or a set of columns corresponding to the heuristic solution provided by the greedy randomized algorithm described in [5].

In the pricing sub-problem we look for the most violated dual constraint (8) corresponding to the primal variable x_p with the most negative reduced cost.

Let denote α, β and γ as non-positive vectors of dual variables associated with constraints (2), (3) and (4) then the expression of the dual constraint associated with x_p is:

$$
-\sum_{i \in \mathcal{N}_p} \alpha_i + \sum_{(i,j) \in \mathcal{A}_p} \beta_{ij} + \gamma_h \leq 1 \qquad \forall p \in \bar{\mathcal{P}}_h, \forall h \in N \setminus \{0\} \qquad (8)
$$

In practice, the pricing procedure has to find the feasible minimum cost path starting from any of the nodes in $\mathcal{N} \setminus \{0\}$ and arriving at the node 0 where the cost depends on α_i for each node $i \in \mathcal{N}$ and β_{ij} for each arc $(i,j) \in \mathcal{A}$. The total cost of a path p is then defined as $r_p = 1 + \sum_{i \in \mathcal{N}_p} \alpha_i - \sum_{(i,j) \in \mathcal{A}_p} \beta_{ij} - \gamma_h$

where h is the starting node of the path. If $r_p < 0$ then the reduced cost of the variable x_p associated with path p is negative and x_p is added to $\mathcal{P}$ in the RPM.

The problem of finding the feasible path with the most negative reduced cost can be seen as a resource constrained elementary shortest path problem (see [6] for a recent survey) in which the resource, that is monotonically consumed along the path, is the maximal allowed deviation from the shortest path (Δ).

In each iteration of the column generation procedure $|\mathcal{N}| - 1$ independent pricing sub-problems, one for each node in $\mathcal{N} \setminus \{0\}$, are solved. Two different procedures are employed to tackle this problem. The first one is heuristic, it is based on a very fast nearest neighbor algorithm and it is mainly used to rapidly populate the set $\mathcal{P}$ in the first iterations of the column generation procedure. The second procedure solves the problem exactly using a slightly modified version of the *pulse* algorithm presented in [4], the procedure is executed every time the heuristic algorithm is not able to find any path with negative reduced cost. In our version of the *pulse* procedure the resource Δ is initialized to δc_{h0} where h is the starting node of the path associated with that pricing sub-problem and it is reduced every time the path is extended. In detail, when the path is extended from node i, with resource value Δ_i, to node j the new value for the resource Δ_j is equal to $min(\Delta_i - c_{ij}, \delta c_{j0})$. Moreover, denoted with Δ_{min} and Δ_{max} the minimum and the maximum values for δc_{i0} $\forall i \in N \setminus \{0\}$, then the bounding procedure of the *pulse* algorithm is executed starting with a resource value equal to Δ_{min} and it is increased by one tenth of $\Delta_{max} - \Delta_{min}$ in each iteration until Δ_{max} is reached.

At the end of the column generation procedure an integer solution for the problem is achieved solving the MIP model (1-7) with all variables that have been included in $\mathcal{P}$. Therefore, since we do not perform a full branch and price scheme the optimality of the integer solution cannot be guaranteed.

3 Computational experiments

In order to test our approach, the column generation procedure has been implemented in Python 2.7 using Pyomo 5.0 as optimization modeling language (see [3] and [2]) and IBM CPLEX 12.7.0 as LP an MIP solver. Note that in our implementation of the *pulse* algorithm the use of multi-threading differs from the original. In particular, since for each iteration of column generation $|\mathcal{N}| - 1$ independent pricing sub-problems have to be solved, we prefer to solve all problems in parallel using a single-thread *pulse* implementation instead of solving pricing problems sequentially with a multi-thread algorithm.

To provide a fair and coherent comparison between the path-based and the arc-based models, the testing campaign has been carried out on the same set of instances and on the same machine that were used to evaluate the performances of the arc-based approach for $MLPLP$ (see [5]).

The results obtained are shown in table 1. For each instance, columns in the table report: the number of nodes $|\mathcal{N}|$ (school excluded) in G, the maximum allowed δ value, the value of the linear relaxation of the arc-based model (LP), the value of the linear relaxation computed with the column generation procedure (CG LP, * used when the optimum is reached), the value of the solution found by the greedy algorithm (Heur), the value of the best integer solution found by the arc-based model without initialization and initialized with the greedy algorithm (MIP and HMIP, * used when the optimal solution is reached), the value of the best integer solution found by the path-based procedure initialized with dummy columns (CG MIP) and with the solution found by the greedy algorithm (HCG MIP), the percentage residual gap for the MIP model (1 - 7) solved with all the variables found in the column generation procedure initialized with the results of the greedy algorithm (G% HCG) and the total computational time that is the sum of execution time of the greedy algorithm, the column generation procedure (time-limit 1 hour) and the solving time of the resulting MIP (time-limit 1 hour plus what remains from the column generation procedure).

The LP relaxation provided by the column generation procedure is greatly superior to the linear relaxation of the arc-based model. Indeed, the average gap between the CG LP and the corresponding HMIP on instances solved to the optimality is about 12% while it is more than 400% when the linear relaxation of the arc-based model (LP column) is taken into account. However, computing LP CG can be time expensive and in 1 hour we are only able to compute the optimal continuous value of the RMP for graphs with less than 200 nodes. When δ is small (1.1, 1.2) this computational time is equally split between the solution of pricing sub-problems and the linear relaxation of the RMP. On the other hand, if δ is large then longer paths have to be considered and so the pricing phase is more time expensive and can take up to 80% of the column generation procedure.

The integer solution found by the column generation procedure (CG MIP) is on average 13.76% better than the solution found by the arc-based model (MIP). This value is greatly reduced, it is about 2.14%, when both procedures are initialized with the results of the heuristic procedure (HMIP vs HCG MIP).

The greedy multi-start algorithm is very useful in particular in instances with more than 150 nodes and, when combined with the column generation procedure, provides solution that are on average 12.78% better than the solution found by the column generation procedure alone (HCG MIP vs CG MIP).

Considering only instances in which we are able to compute the optimal linear relaxation for the master problem (almost all instances with less than 200 nodes), the gap between CG LP and HCG MIP is about 13.58% while the final MIP gap, after 2 hours of computation, for the arc-based model was about twice as much on the same instances (see [5] for detailed results).

Finally, a note on the computational time required by the MIP problem solved at the end of the column generation procedure for instances with more than 150 nodes. On the one hand, when δ is big the exact pricing procedure is extremely time consuming and can be executed only a few times (2 or 3) generating a small MIP problem that can be solved in a few seconds. On the other hand, when δ is small the pricing sub-problem can be solved more than 100 times and thousands of columns are added into the RMP. This generate a very large MIP problem that usually cannot be solved within 1 hour.

References

[1] Guy Desaulniers, Jacques Desrosiers and Marius M. Solomon, *Column Generation*, Springer, Boston, MA, (2005).

[2] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson and John D. Siirola, *Pyomo–optimization modeling in python*, Springer Science & Business Media, **67** (2017).

[3] William E. Hart, Jean-Paul Watson and David L. Woodruff, *Pyomo: modeling and solving mathematical programs in Python*, Mathematical Programming Computation, **3** (2011), 219–260.

[4] Leonardo Lozano, Daniel Duque and Andrs L. Medaglia, *An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints*, Transportation Science, **50** (2016), 348-357.

[5] Federico Malucelli, Maddalena Nonato and Emanuele Tresoldi, *Optimization based planning of Pedibus lines: an arc based approach*, Transportation Research Procedia, **27** (2017), 760 - 767.

[6] Luigi Di Puglia Pugliese and Francesca Guerriero, *A Survey of Resource Constrained Shortest Path Problems: Exact Solution Approaches*, Networks, **62** (2013), 183 - 200.

Table 1
Path Model Results on Random Instances

$\|N\|$	δ	LP	CG LP	Heur	MIP	HMIP	CG MIP	HCG MIP	G% HCG	T HCG
10	11	1.267	5.053*	5.713	5.571*	5.571*	5.571	5.571	0.000	1.025
10	12	1.265	4.063*	4.693	4.693*	4.693*	4.693	4.693	0.000	0.830
10	15	1.262	3.049*	3.576	3.482*	3.482*	3.482	3.482	0.000	1.422
10	18	1.259	2.307*	3.562	3.418*	3.418*	3.418	3.418	0.000	1.524
10	20	1.257	2.050*	3.590	2.550*	2.550*	2.550	2.550	0.000	2.433
20	11	1.024	7.005*	7.124	7.114*	7.114*	7.114	7.114	0.000	1.814
20	12	1.024	6.006*	7.120	6.115*	6.115*	6.115	6.115	0.000	3.082
20	15	1.023	3.605*	5.098	4.084*	4.084*	4.086	4.086	0.000	4.969
20	18	1.023	2.823*	3.097	3.090*	3.090*	3.094	3.097	0.000	2.431
20	20	1.023	2.604*	3.102	3.071*	3.071*	3.087	3.081	0.000	2.872
30	11	1.030	9.505*	10.191	10.143*	10.143*	10.151	10.150	0.000	5.599
30	12	1.030	7.505*	9.175	8.132*	8.132*	8.137	8.137	0.000	9.461
30	15	1.029	4.360*	5.155	5.109*	5.109*	5.129	5.129	0.000	26.700
30	18	1.029	3.005*	4.151	4.097	3.146*	3.146	3.146	0.000	101.306
30	20	1.029	2.694*	3.137	3.111	3.111	3.140	3.137	0.000	7.474
50	11	1.031	12.006*	13.314	12.281*	12.281*	12.281	12.281	0.000	25.529
50	12	1.031	9.450*	11.288	10.255*	10.255*	10.279	10.279	0.000	34.825
50	15	1.030	6.783*	8.282	8.193	8.194	7.294	7.294	0.000	23.301
50	18	1.030	4.719*	7.247	7.146	7.145	6.241	6.222	0.000	1364.550
50	20	1.029	3.943*	6.243	6.148	6.150	5.234	4.261	0.000	2621.581
80	11	1.034	9.364*	12.419	10.342	11.287	10.392	10.394	0.000	125.507
80	12	1.033	7.235*	9.413	9.263	9.264	8.408	8.408	0.000	2491.820
80	15	1.032	4.848*	6.409	7.183	6.269	7.395	6.409	0.000	92.759
80	18	1.032	4.164*	5.375	8.131	5.236	7.368	5.375	0.000	137.700
80	20	1.032	5.023*	5.380	8.123	5.202	7.398	5.380	0.000	324.440
100	11	1.036	19.854*	24.590	21.488	21.486	21.522	21.533	0.000	146.327
100	12	1.036	14.179*	18.561	17.367	17.384	16.486	15.536	0.000	374.995
100	15	1.036	8.436*	11.522	15.226	10.387	11.505	10.497	11.930	7200.000
100	18	1.036	6.393*	7.559	14.217	7.446	9.509	7.559	0.000	215.956
100	20	1.036	6.105*	7.517	13.245	7.417	8.500	7.517	0.000	385.414
150	11	1.004	16.491*	21.082	21.051	20.056	19.071	19.073	10.423	7200.000
150	12	1.004	11.729	15.080	17.048	15.056	13.075	13.075	20.347	7200.000
150	15	1.004	9.676*	10.078	18.035	10.067	15.077	10.078	0.000	746.739
150	18	1.004	8.003*	8.076	21.027	8.070	16.081	8.076	0.000	1546.788
150	20	1.004	7.003*	7.077	28.029	7.076	15.079	7.077	0.000	1483.544
200	11	1.004	29.805*	38.119	33.080	33.081	31.105	31.105	0.000	843.554
200	12	1.004	21.405	29.115	28.064	27.067	25.094	25.094	11.667	7200.000
200	15	1.004	16.164	19.106	28.049	19.070	19.097	19.106	16.672	7200.000
200	18	1.004	13.207	14.105	37.038	14.100	22.097	14.104	0.000	3684.441
200	20	1.004	12.798	13.101	37.040	13.095	19.102	13.101	0.000	3632.482
250	11	1.005	31.346	43.140	39.088	38.093	36.127	36.127	0.000	7200.000
250	12	1.005	24.531	32.131	35.068	31.080	31.130	32.127	20.272	7200.000
250	15	1.005	18.742	20.130	31.065	20.114	30.125	20.129	0.000	3657.909
250	18	1.005	15.162	16.124	48.062	16.119	26.131	16.122	0.000	3688.617
250	20	1.005	13.004	13.125	48.061	13.120	25.131	13.125	0.000	3643.244
300	11	1.005	33.737	44.165	43.096	41.096	37.147	37.147	8.035	7200.000
300	12	1.005	25.848	35.156	41.075	34.107	43.149	35.156	7.500	7200.000
300	15	1.005	21.004	21.150	53.075	21.148	38.148	21.150	0.000	3671.378
300	18	1.005	14.004	14.150	61.064	14.149	36.143	14.150	0.000	3658.683
300	20	1.005	13.004	13.148	82.068	13.145	35.156	13.148	0.000	3687.936