Special Need Students School Bus Routing: Consideration for Mixed Load and Heterogeneous Fleet

Hernan Caceres[a,b], Rajan Batta[a,*], Qing He[a,c]

[a] *Department of Industrial and System Engineering, University at Buffalo, New York, USA*
[b] *Departamento de Ingeniería Industrial, Universidad Católica del Norte, Antofagasta, Chile*
[c] *Departments of Civil, Structural, and Environmental Engineering, University at Buffalo, New York, USA*

Abstract

The problem of routing special-education students differs in many aspects with that of routing regular students. A bus can be configured to also support wheelchairs. Students may be served differently depending on their disability, and they need to be picked up and dropped off in their homes. In our study we modeled a mixed integer program that accounts for these and other characteristics. We use column generation to find approximate solutions for real instances and the benchmark set of instances from Park et al. (2012). Our motivation and examples are drawn from a large suburban school district in Western New York, United States.

Keywords: School Bus Routing, Column Generation, Special Education

1. Introduction

The school bus routing problem (SBRP) for special needs students is an extension of the regular SBRP where there is a bigger concern with service level. Particularly in New York State, challenged students need to be picked up at their door step and dropped off to the school where their special-education program runs. They may need special equipment and they are to be supervised at times by someone other than the driver, among others. Because the number of students per special education program is very small in relation to the regular students per school, the school districts often allow a mixed load configuration for these buses, where a bus can serve students from different schools (Park and Kim, 2010).

A major difference between routing special education students as opposed to regular students is the diversity of the students and the programs they attend. In addition to special restrictions in travel time and equipment, special education students do not necessarily live close to their programs whereas in the most common SBRP for regular students it is expected that a large number of students live relatively close to their school. Moreover, the number of students per program is dramatically lower than for regular schools. In regards to the programs (or schools), these are geographically dispersed and have different start and end times. The location of the program is particularly troubling because many times they are located beyond the limits of the school district, making the routes very long, which then permits the assignment of only a few students.

The most recent review of the SBRP is from Park and Kim (2010) and even though the research has been extensive, they state that only a few papers have considered the problem of routing special education students. Russell and Morrel (1986) were the first to address the special education routing problem. They modified the savings algorithm from Clarke and Wright (1964) to build an initial solution that they later improve with the 3-opt and M-TOUR algorithm. Because of the diversity of the students; routes would often involve multiple schools resulting in longer travel time for the students. They addressed this issue and implemented a shuttle system where students are dropped off at one of two stations and then assigned to buses that have at most two schools as a destination. Braca et al. (1997) also discussed routing for special education students in their work with the school system at New York City. However,

[*] Corresponding author
Email addresses: hernanan@buffalo.edu (Hernan Caceres), batta@buffalo.edu (Rajan Batta), qinghe@buffalo.edu (Qing He)

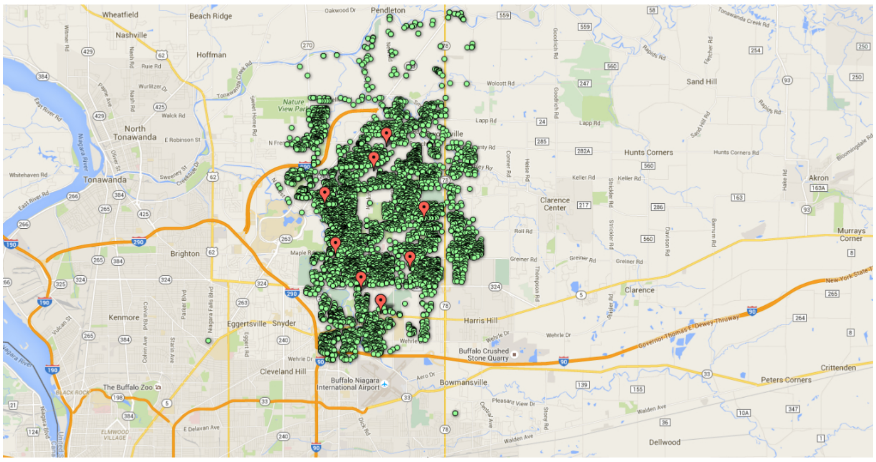

Figure 1: Location of homes and schools for regular students

their main focus was on regular students. The work of Ripplinger (2005) focused in the rural case of the SBRP and briefly discussed considerations for special education students, particularly it suggests the idea of keeping track of vehicle occupancy classified by seat-type, a feature that we modeled in our problem formulation.

Kamali et al. (2013) examined at the busing problem in a broader way, by first assigning students to school and then solving the routing problem. Provided that the students' need can be satisfied by more than one school, it becomes relevant then to combine assignment and routing in a single problem. They developed a mathematical model that was then solved approximately with a heuristic.

A review of Park and Kim (2010) about SBRP describes the variants of the problem and as one of their conclusions they noticed that there is a lack of work in the mixed ride of special-education and general students. Continuing their work, Park et al. (2012) focused in the mixed ride of general students and developed a post-improvement procedure that aims to minimize the number of buses needed to serves all students in a particular region.

In our work we study SBRP for special-education students in a suburban area, addressing both morning and afternoon problems and allowing mixed load with heterogeneous fleet where we aim to minimize the total number of buses needed. We also assume that stop locations and bell time for the schools are given. Among our constraints, we consider vehicle capacity, maximum ride time and school time windows.

2. Problem description

The problem studied in this research is based on the operation of the Transportation Department at Williamsville Central School District (WCSD) in New York, and focuses on the transportation of students enrolled in Special Education programs. This problem might seem similar to the typical SBRP. However, there are important differences that need careful considerations.

As in many school districts, WCSD uses a contractor that provides full transportation for these students. But is the school district that designs the routes and decides the number of buses to be utilized. This is done within a month prior to the beginning of each school year.

At WCSD, Special Education students are picked up at their doorstep whereas regular students are required to walk to a certain stop location, that could be assigned to multiple students. Additionally, every bus is required to have an aide on board to care for the students and assist the most challenged kids; a nurse may also be required depending on the student's needs. If needed, the buses have to be specially equipped, for example, they need to be able to handle safe transportation for students in wheelchairs, which reduces the capacity of the bus.

The number of students in special education programs is significantly small, about 4% of the total student population. However, the number of schools involved is much greater than regular programs. They have different bell times and their locations are more disperse (having several out of the district's boundaries) in comparison to regular

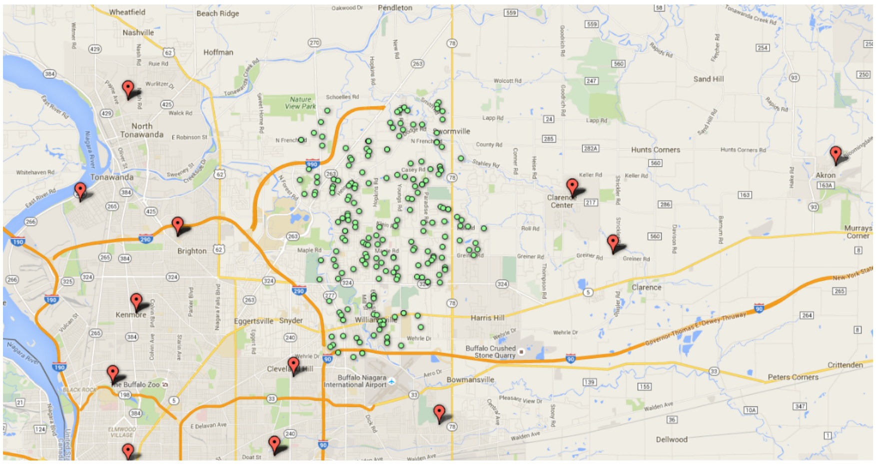

Figure 2: Location of homes and schools for special education students

schools. From Figures 1 and 2 we can see the significant differences between the locations of homes (in green circles) and schools (in red pins) for the regular case of SBRP as opposed to for the special education program. These particular features tend to make routes significantly longer (Russell and Morrel, 1986) and with less use of bus capacity. Consequently, the operation of special education students is very expensive, reaching 40% of the annual transportation budget of WCSD. In other words, 40% of the transportation budget is spent on special education children, who account for just 4% of the total number of students. Thus, careful planning of this operation is needed to identify saving opportunities while guaranteeing the quality of service.

Accordingly, the purpose of this research is to propose a procedure to plan routes for the special education buses where we account for the following considerations. Each school has a defined time window for drop off in the morning and pick up in the afternoon, and a particular waiting time or delay for both drop-off and pick-up. Each student has a personal waiting time or delay at pick-up and drop-off depending on their need, a personal maximum ride time, a personal number of seats to occupy (if unable to share a seat with other students or/and a nurse is needed). The students may request to ride alone in the bus (in this case the number of seats equals the capacity of the bus), may request not to be in the bus with a particular group of students, and needs to be picked up and dropped off at his/hers doorstep. The capacity of the buses is 18 students. However, the district established that each bus must hold no more than 10 students at any moment. Additionally, the buses can be configured to hold up to three wheelchairs, having the remainder space setup with regular seats. A bus may also carry students from different schools at the same time, and it cannot make U-turns.

The model in the following section considers all of the above and supports school bus routing for both morning and afternoon runs. The objective of the model is to minimize the total number of buses used in the morning and afternoon runs. Later, we provide a solution strategy based on column generation. Even though our methods and testing is based on the Williamsville School District scenario, we believe that the methods can be readily modified for other school districts and the results are widely applicable.

3. Mathematical model

In order for our model to support both AM and PM runs, we define θ as a binary parameter equal to 1 if routes are for the AM runs and 0 for the PM runs. In addition we define $\phi = 2\left(\theta - \frac{1}{2}\right)$. These parameters enable us to formulate a unique model for both cases.

Consider the set A of all stops for students, the set S of all schools and the sets D_1 and D_2 corresponding to the start and end depots respectively. The set of all location is L, having $L = D_1 \cup A \cup S \cup D_2$. Let the function $\delta(i)$ represent the school of the students in stop $i \in A$ (this means $\delta(i) \in S$). Let t_i be the fixed waiting time at node $i \in A \cup S$ (time per stop), $\dot{t}_i$ be the variable waiting time at node $i \in A \cup S$ (time per student), t_{ij} be the travel time from node

i to node j, and τ_i the maximum travel time for student $i \in A$. Let a_i and b_i represent the earliest and latest time of arrival to location $i \in L$ (time window). For $i \in S$ the time windows of arrival is provided as $[a_i, b_i]$, for the rest the assignment is as follows:

$$a_i = \begin{cases} a_j - \theta \max\{t_i + \tau_i, t_i + t_{ij}\} + (1-\theta)\left(t_j + t_{ji}\right) : j = \delta(i) & \text{if } i \in A \\ \theta \min_{j \in A}\{a_j - t_{ij}\} + (1-\theta) \min_{j \in S}\{a_j - t_{ij}\} & \text{if } i \in D_1 \\ \theta \min_{j \in S}\{a_j + t_j + t_{ji}\} + (1-\theta) \min_{j \in A}\{a_j + t_j + t_{ji}\} & \text{if } i \in D_2 \end{cases}$$

$$b_i = \begin{cases} b_j - \theta\left(t_i + t_{ij}\right) + (1-\theta) \max\{t_j + \tau_i, t_j + t_{ji}\} : j = \delta(i) & \text{if } i \in A \\ \theta \max_{j \in A}\{b_j - t_{ij}\} + (1-\theta) \max_{j \in S}\{b_j - t_{ij}\} & \text{if } i \in D_1 \\ \theta \max_{j \in S}\{b_j + t_j + t_{ji}\} + (1-\theta) \max_{j \in A}\{b_j + t_j + t_{ji}\} & \text{if } i \in D_2 \end{cases}$$

Let $G = (L, E)$ be a directed graph. The set of edges is $E = E_{DA} \cup E_{AA} \cup E_{AS} \cup E_{SA} \cup E_{SS} \cup E_{SD}$ where $E_{DA} = \{(i, j) \in D_1 \times A\}$ is the set of edges connecting the depot to the students, $E_{AA} = \{(i, j) \in A^2 \mid i \neq j, t_{ij} + t_j + t_{j,\delta(j)} \leq \max\{\tau_i, t_{i,\delta(i)}\}, a_i + t_i + t_{ij} \leq b_j\}$ is the set of feasible links between students, $E_{AS} = \{(i, j) \in A \times S \mid j \neq \delta(i), t_{ij} + t_j + t_{j,\delta(i)} \leq \max\{\tau_i, t_{i,\delta(i)}\}, a_i + t_i + t_{ij} \leq b_j\} \cup \{(i, \delta(i)) \mid i \in A\}$ is the set of feasible links from students to schools, $E_{SA} = \{(i, j) \in S \times A \mid i \neq \delta(j), a_i + t_i + t_{ij} \leq b_j\}$ is the set of feasible links from schools to students, $E_{SS} = \{(i, j) \in S \times A \mid i \neq j, a_i + t_i + t_{ij} \leq b_j\}$ is the set of feasible links between schools, and $E_{SD} = \{(i, j) \in S \times D\}$ is the set of edges connecting schools to the depot.

Additionally, let $A_j = \{i \in A : j = \delta(i)\}$ be the set of students attending school $j \in S$. Notice that the sets A_j are mutually exclusive.

Regarding the attributes of the vehicles we consider in our model we define B to be the set of buses and Q to be the set of seat types that the buses have (e.g. regular seats, wheelchair spaces). Let d_{ijk} be a binary parameter equal to 1 if bus $k \in B$ starts in depot $i \in D_1$ and finished $j \in D_2$, s_i^q be the number of seat types $q \in Q$ used by student $i \in A$, and c_k^q capacity of bus $k \in B$ in regards of seat-type $q \in Q$.

We now define the decision variables of our model. Let z_k be a binary variable indicating if bus $k \in B$ is used, and x_{ijk} be a binary variable indicating if bus $k \in B$ goes from node $i \in L$ to node $j \in L$. Let u_{ik} be the time of arrival of bus $k \in B$ at node $i \in L$ and v_{ik} the load of bus $k \in B$ upon arrival at node $i \in L$. Let ε be the inverse of an upper bound on the total travel time for al vehicles. Thus, the formulation of our problem reads as follows:

$$\min \quad \sum_{k \in B} z_k + \varepsilon \sum_{k \in B} \sum_{(i,j) \in E} t_{ij} \boldsymbol{x}_{ijk} \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in B} \sum_{j \in L, j \neq i} \boldsymbol{x}_{ijk} = 1 \qquad i \in A \tag{2}$$

$$\sum_{(i,j) \in E} \boldsymbol{x}_{ijk} \leq z_k \qquad k \in B \tag{3}$$

$$\theta \sum_{j \in A} \boldsymbol{x}_{ijk} + (1-\theta) \sum_{j \in S} \boldsymbol{x}_{ijk} \leq \sum_{j \in D_2} d_{ij} \qquad i \in D_1, k \in B \tag{4}$$

$$\theta \sum_{i \in S} \boldsymbol{x}_{ijk} + (1-\theta) \sum_{i \in A} \boldsymbol{x}_{ijk} \leq \sum_{i \in D_1} d_{ij} \qquad j \in D_2, k \in B \tag{5}$$

$$\sum_{k \in B} \sum_{(i,j) \in L^2 \setminus E} \boldsymbol{x}_{ijk} = 0 \tag{6}$$

$$\sum_{i \in L} \boldsymbol{x}_{ijk} = \sum_{i \in L} \boldsymbol{x}_{jik} \qquad j \in A \cup S, k \in B \tag{7}$$

$$\sum_{g \in L} \boldsymbol{x}_{gik} \leq \sum_{g \in L} \boldsymbol{x}_{g,\delta(i),k} \qquad i \in A, k \in B \tag{8}$$

$$\boldsymbol{u}_{ik} + t_i + \dot{t}_i \sum_{q \in Q} s_i^q + t_{ij} \leq \boldsymbol{u}_{jk} + M_1\left(1 - \boldsymbol{x}_{ijk}\right) \qquad i \in L \setminus S, j \in L, k \in B \tag{9}$$

$$\boldsymbol{u}_{ik} + t_i + \dot{t}_i \sum_{q \in Q} \sum_{e \in A_i} \sum_{g \in L} s_e^q \boldsymbol{x}_{gek} + t_{ij} \leq \boldsymbol{u}_{jk} + M_1\left(1 - \boldsymbol{x}_{ijk}\right) \qquad i \in S, j \in L, k \in B \tag{10}$$

$$0 \leq \phi\left(\boldsymbol{u}_{\delta(j),k} - \boldsymbol{u}_{ik}\right) \leq \max\{\tau_i, \theta t_{i,\delta(j)} + (1-\theta)\, t_{\delta(j),i}\} \qquad i \in A, k \in B \tag{11}$$

$$a_i \leq \boldsymbol{u}_{ik} \leq b_i \qquad i \in L, k \in B \tag{12}$$

$$\boldsymbol{v}^q_{ik} + \phi s^q_i \leq \boldsymbol{v}^q_{jk} + M_2\left(1 - \boldsymbol{x}_{ijk}\right) \qquad i \in L \backslash S, j \in L, k \in B, q \in Q \tag{13}$$

$$\boldsymbol{v}^q_{ik} - \phi \sum_{e\in A_i}\sum_{g\in L} s^q_e \boldsymbol{x}_{gek} \leq \boldsymbol{v}^q_{jk} + M_2\left(1 - \boldsymbol{x}_{ijk}\right) \qquad i \in S, j \in L, k \in B, q \in Q \tag{14}$$

$$0 \leq \boldsymbol{v}^q_{ik} \leq c^q_k \qquad i \in L, k \in B, q \in Q \tag{15}$$

$$\boldsymbol{x}_{ijk} \in \{0, 1\} \tag{16}$$

where (2) ensures that all students are pick up, (3) each bus leaves the depot at most once, (4) and (5) verifies that all buses start and return to their respective depot, (6) are the prohibited moves, (7) is the flow conservation, (8) ensures the bus visits the corresponding schools, (9) and (10) define the time of arrival at each stop, (11) ensures a school is visited after picking up the corresponding students for the AM run or the opposite in the PM run and the travel time is lower than the maximum allowed for each student, (12) is the time windows for the arrival to school, (13) and (14) define the variation of the number of used seats in the bus, and (15) verifies the capacity of the bus is not violated.

4. Solution strategy

Because of the complexity of the routing problem (1)-(16) from the previous section, CPLEX is not able to solve realistic size instances. Thus, we elect to solve the problem approximately. Our approach is based on a column generation decomposition and a procedure tailored for our problem's special characteristics. In the work of Caceres et al. (2014) a similar decomposition was successfully applied for the regular bus routing problem.

4.1. Problem decomposition by column generation

4.1.1. Master problem

Let P_k be the set of feasible paths for bus $k \in B$, where $p \in P_k$ is an elementary path. Let x^p_{ijk} be equal to 1 if edge $(i, j) \in L^2$ is covered by bus $k \in B$ when using path $p \in P_k$, $\theta^p_k = \sum_{i\in D}\sum_{j\in A} x^p_{ijk} + \varepsilon \sum_{i\in L}\sum_{j\in L} t_{ij} x^p_{ijk}$ be the cost of using path $p \in P_k$ with vehicle $k \in B$ and $v^p_{ik} = \sum_{j\in A\cup S} x^p_{ijk}$ be equal to 1 if stop $i \in A$ is visited by bus $k \in B$ when using path $p \in P_k$ and 0 otherwise. Let y^p_k be the binary decision variables that are equal to 1 if path $p \in P_k$ is used by bus $k \in B$ and 0 otherwise. Then, the master problem reads as follows:

$$\text{Min} \quad \sum_{k\in B}\sum_{p\in P_k} \theta^p_k y^p_k \tag{17}$$

$$\text{s.t.} \quad \sum_{k\in B}\sum_{p\in P_k} v^p_{ik} y^p_k = 1, \quad i \in A \tag{18}$$

$$\sum_{p\in P_k} y^p_k \leq 1, \quad k \in B \tag{19}$$

$$y^p_k \text{ binary} \tag{20}$$

Given that the are different seats configuration for the buses, we define a certain number of available bus for each configuration. Therefore, let W define the set of unique bus classes, where each element $w \in W$ represents a bus class with distinct seats configuration, and let K_w be the number of available buses for each class. Then, the new master problem reads as follows:

$$\text{Min} \quad \sum_{w\in W}\sum_{p\in P_w} \theta^p_w y^p_w \tag{21}$$

$$\text{s.t.} \quad \sum_{w\in W}\sum_{p\in P_w} v^p_{iw} y^p_w = 1, \quad i \in A \tag{22}$$

$$\sum_{p\in P_w} y^p_w \leq K_w, \quad w \in W \tag{23}$$

$$y^p_w \text{ binary} \tag{24}$$

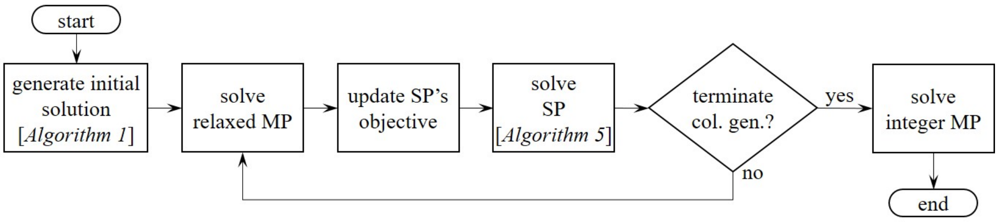

Figure 3: Column generation procedure.

4.1.2. Sub problem

Since the buses different configuration, one subproblem must be solved for each bus class. Thus, there will be $|W|$ subproblems to solve separately.

Let π_i represent the dual variables associated with constraints (22) and ρ_w represent the dual variables associated with constraints (23). Then, for a given bus the subproblem minimizes the reduced cost $\theta_w^p - \left(\sum_{i\in A} \pi_i v_{iw}^p + \rho_w\right)$. Thus, the subproblem for class $w \in W$ reads as follows:

$$\min \quad 1 - \rho_w + \sum_{(i,j)\in E} \left[\varepsilon t_{ij} - \pi_i\right] \boldsymbol{x}_{ij} \tag{25}$$

$$\text{s.t.} \quad \sum_{j\in L, j\neq i} \boldsymbol{x}_{ij} \leq 1 \qquad i \in A \tag{26}$$

$$\theta \sum_{j\in A} \boldsymbol{x}_{ij} + (1-\theta) \sum_{j\in S} \boldsymbol{x}_{ij} = \sum_{j\in D_2} d_{ij} \qquad i \in D_1 \tag{27}$$

$$\theta \sum_{i\in S} \boldsymbol{x}_{ij} + (1-\theta) \sum_{i\in A} \boldsymbol{x}_{ij} = \sum_{i\in D_1} d_{ij} \qquad j \in D_2 \tag{28}$$

$$\sum_{(i,j)\in L^2\setminus E} \boldsymbol{x}_{ij} = 0 \tag{29}$$

$$\sum_{i\in L} \boldsymbol{x}_{ij} = \sum_{i\in L} \boldsymbol{x}_{ji} \qquad j \in A \cup S \tag{30}$$

$$\sum_{g\in L} \boldsymbol{x}_{gi} \leq \sum_{g\in L} \boldsymbol{x}_{g,\delta(i)} \qquad i \in A \tag{31}$$

$$\boldsymbol{u}_i + t_i + \dot{t}_i \sum_{q\in Q} s_i^q + t_{ij} \leq \boldsymbol{u}_j + M_1\left(1 - \boldsymbol{x}_{ij}\right) \qquad i \in L\setminus S, j \in L \tag{32}$$

$$\boldsymbol{u}_i + t_i + \dot{t}_i \sum_{q\in Q} \sum_{e\in A_i} \sum_{g\in L} s_e^q \boldsymbol{x}_{ge} + t_{ij} \leq \boldsymbol{u}_j + M_1\left(1 - \boldsymbol{x}_{ij}\right) \qquad i \in S, j \in L \tag{33}$$

$$0 \leq \phi\left(\boldsymbol{u}_{\delta(j)} - \boldsymbol{u}_i\right) \leq \max\{\tau_i, \theta t_{i,\delta(j)} + (1-\theta)\, t_{\delta(j),i}\} \qquad i \in A \tag{34}$$

$$a_i \leq \boldsymbol{u}_i \leq b_i \qquad i \in L \tag{35}$$

$$\boldsymbol{v}_i^q + \phi s_i^q \leq \boldsymbol{v}_j^q + M_2\left(1 - \boldsymbol{x}_{ij}\right) \qquad i \in L\setminus S, j \in L, q \in Q \tag{36}$$

$$\boldsymbol{v}_i^q - \phi \sum_{e\in A_i} \sum_{g\in L} s_e^q \boldsymbol{x}_{ge} \leq \boldsymbol{v}_j^q + M_2\left(1 - \boldsymbol{x}_{ij}\right) \qquad i \in S, j \in L, q \in Q \tag{37}$$

$$0 \leq \boldsymbol{v}_i^q \leq c^q \qquad i \in L, q \in Q \tag{38}$$

$$\boldsymbol{x}_{ij} \in \{0, 1\} \tag{39}$$

4.2. Column generation procedure

Our column generation procedure follows the structure of Figure 3. In a general sense the procedure contemplates the following four steps.

Step 1 (generate initial solution). Using Algorithm 1, we construct an initial solution that is used to begin the column generation process as for establishing the value of ε as the inverse of total travel time, needed in the objective functions (1) and (25).

Step 2 (solve a relaxation the master problem). Solve the linear relaxation of the master problem with all the available columns. Update the coefficient of the objective function of the subproblem with the new values of the dual variables.

Step 3 (solve subproblems). Using Algorithm 5, we generate new columns. If at least one of the columns found in any of the subproblems has a negative cost, then go to *Step 2*.

Step 4 (solve the master problem as IP). Solve the master problem as an integer program using regular branch and bound.

In the following section, we present further explanation on how the algorithm used here works.

4.2.1. Generating an initial solution

The procedure to generate the initial solution is detailed in Algorithm 1. In order to create a route, the algorithm first selects the bus with the most availability of the seat type with least demand among the students. Next, we filter the stops that have positive demand for the selected seat type. To choose which stop to add into the new route, we calculate for each stop the new cost to the route resulting from adding such a stop.

To calculate the new cost for the route, Algorithm 2 is invoked, where the best insertion (the cheapest) of a given stop is found, and we use it to calculate the cost. In Algorithm 2 all possible insertions are tried, and depending upon whether or not the corresponding school of the stop is already in the route, this is added and tried on every possible position in the route. To check the feasibility of the insertion Algorithm 3 is invoked, where first we check if all edges are part of the set E, then the capacity of the bus is checked, then the time windows and finally the maximum riding time for each specific student. If no insertions are possible, a cost equal to ∞ is assigned to the new cost of adding the stop.

Once we know the new cost for all possible insertions, in Algorithm 1 line 16 for each of the stops we calculate a special saving cost that is the product of the number of seats needed by the stop and the saving of having this stop inserted in the route rather than have it in an individual bus. The purpose of combining the saving and the number of seats is to prioritize the selection of stops with higher demand for seats; stops with lower demand are easily accommodated when a route can no longer permit big bulks of demand. Algorithm 1 can be invoked under two modes: deterministic or random. In the deterministic mode, the next stop to add is selected by choosing the one with the maximum saving, whereas in the random mode we choose the next stop to be added invoking Algorithm 4. In Algorithm 4 we select the stop randomly considering that the probability of selecting a particular stop i is proportional to $e^{\gamma S[i]}$ where $\gamma = 0.1$ and $S[i]$ represents the saving calculated within Algorithm 1.

Continuing with Algorithm 1, after selecting the next stop to be added we continue adding stops in the same manner until no stop can be added. Then, a new route is created following the same criteria previously used. This algorithm returns a set of routes that corresponds to a feasible solution for our problem.

In order to create diversity in the set of initial solutions, in our implementation Algorithm 1 is run once in deterministic mode and 50 times in random mode. All the set of solutions generated is then passed to the master problem to continue with the column generation procedure.

4.2.2. Approximate solution for the subproblem

The subproblem is NP-hard. Therefore finding an optimal solution for it is computational intensive. Thus, we chose to solve the subproblem approximately by means of a heuristic. The heuristic is described in Algorithm 5. This algorithm is also based on the choosing of stops on the basis of savings generated from including a stop in a route as opposed to having it a particular route. The algorithm creates routes for the given bus class. In order to decide which stop to add, for every stop we calculate the new cost of the route if such a stop is added using Algorithm 2. Similarly, this new cost is used to calculate the saving that is later used to select the stop that will be added to the route. Such selection is performed once in deterministic mode by selecting the stop with maximum saving or using random mode with Algorithm 4. After selecting the new stop to be added, we repeat the procedure until no more stops can be added.

There are as many subproblems as bus classes, and we use Algorithm 5 to solve them to obtain the new columns before going back to the master problem for the next iteration. In order to accelerate the column generation, in our implementation Algorithm 5 is run once in deterministic mode and 50 times in random mode. All the solutions

Algorithm 1 Generate initial solution for the master problem

```
1: function GetInitialSolution(L, B)                        ▷ L: set of stops, B: set of buses
2:     for i = 1 to n do                                     ▷ n is the number of student stops
3:         t ← travel time from stop i to its corresponding school
4:         T[i] ← 1 + ε × t
5:     end for
6:     q ← index of seat type with least demand
7:     w ← index of the bus-class with the most availability of seat type q
8:     U ← set of stops with demand of seat type q
9:     r ← new route with start and end depot using bus-class w
10:    n_b[w] = n_b[w] − 1                                   ▷ n_b[w] is the number of available buses in class w
11:    while U.size > 0 and n_b[w] ≥ 0 do
12:        c ← r.cost
13:        for each student i in U do
14:            c′ ← TryAddStop(r, i)                         ▷ Get new cost of r if stop i is added
15:            m ← number of seats requested in stop i
16:            S[i] ← m × (T[i] − (c′ − c))                  ▷ Calculate the saving
17:        end for
18:        if count(S[i] > −∞) > 0 then
19:            if mode = deterministic then
20:                i* ← arg max{S[i]}
21:            else if mode = random then
22:                i* ← SelectStop(S, L)
23:            end if
24:            r.addStop(i*)
25:            U.remove(i*)
26:        else
27:            update q, w, U                                ▷ See lines 6 to 8
28:            r ← new route with start and end depot using bus-class w
29:            n_b[w] = n_b[w] − 1
30:        end if
31:    end while
32:    return r                                              ▷ Return the corresponding shortest path
33: end function
```

Algorithm 2 Try to add a stop to a route

```
function TryAddStop(r, s)                                   ▷ r is a route and s a stop
    c ← ∞
    n ← number of stops in route r
    if r contains the school of student s then
        j ← position in r of the school of stop i
        i ← j
        while i > 1 do
            insert stop s in position i of route r
            if RouteIsFeasible(r) then
                c ← min{c, r.cost}
            end if
            remove stop s from route r
            i ← i − 1
        end while
    else
        for j = 2 to n do
            insert the school of stop s in position j of r
            i ← j
            while i > 1 do
                insert stop s in position i of route r
                if RouteIsFeasible(r) then
                    c ← min{c, r.cost}
                end if
                remove stop s from route r
                i ← i − 1
            end while
            remove the school stop s from route r
        end for
    end if
    return c                                                ▷ Cost of r for best insertion of i
end function
```

Algorithm 3 Check feasibility of a route

```
function RouteIsFeasible(r)                                 ▷ r is a route
    if all edges are valid then
        if capacity is not violated then
            if time windows are not violated then
                if maximum ride times are not violated then
                    return true
                end if
            end if
        end if
    end if
    return false
end function
```

Algorithm 4 Selection of stop

function SelectStop(S, L) ▷ S: array of savings
    $n \leftarrow$ length of array S
    $\gamma \leftarrow 0.1$
    sort(S, L) ▷ Ascending by $S[i]$
    $F[1] \leftarrow e^{\gamma S[1]}$
    for $i = 2$ to n **do**
        $F[i] \leftarrow e^{\gamma S[i]} + F[i-1]$
    end for
    $r \leftarrow$ random value from uniform distribution $U[0, 1]$
    $r \leftarrow r \times F[n]$
    $i \leftarrow 1$
    while $F(i) < r$ **do**
        $i \leftarrow i + 1$
    end while
    return $L[i]$ ▷ Return the corresponding student
end function

generated with negative objective value are then passed to the master problem to continue with the column generation procedure.

5. Computational experiments

The computational experiment consists of two parts. First, we perform experiments using the set of random instance from the work of Park et al. (2012). Second, we consider four real-world instances from WCSD, corresponding to the school years 2013-2014 and 2014-2015

Let us begin with the set of instances from Park et al. (2012). Even though in their work the focus was on routing of regular students, the similarities allowed us to use their instances as a benchmark. Table 1, 2 and 3 show the experience for these instances, where for each we show the name, maximum ride time allowed in seconds and the number of schools stops and students. Under column *Edges* we show the total number of edges and the number of valid edges after discarding the edges that do not conform to the set of edges defined as a function of the time windows with the procedure described in Section 3. Under *Edges* we show the number of vehicles in the solution for each one the methods utilized. Solutions are illustrated under column *Vehicles*. Columns *Braca* and *Park* show the results obtained by applying the work of Braca et al. (1997) and Park et al. (2012), respectively. Column *Start* presents the number of vehicles found with Algorithm 1 that finds the initial solution for the column generation procedure. Under column *CG*, we show the final result for our procedure, meaning at the end of the column generation. Finally, under *CPU time* we show the computational time needed to find the initial solution and the time needed to run the column generation procedure (both times are in minutes).

Table 1 shows the computational experience for 16 instances of type *RSRB*. These instances have randomly distributed schools and bus stops. The maximum riding time is set to either 2700 or 5400 seconds. Note that the first half of the instances is geographically identical to the second half having only the maximum riding time as a difference. At the bottom of Table 1, we can see how our results compare to previous work. For this set of instances, we were able to improve or maintain unchanged 100% of the solutions, provided by Park et al. (2012). For many we can see that our initial solution is good enough to surpass previous work, however after column generation some improvement is achieved.

Similarly, Tables 2 and 3 show the computational experience for 16 instances of type *CSCB*($m/4$,$m/2$) and 16 instances type *CSCB*($m/2$,m) respectively. *CSCB*($m/4$,$m/2$) implies that there are $m/4$ cluster centers for the schools and $m/2$ centers for the bus stops, where m denotes the number of schools. Similarly, *CSCB*($m/2$,m) has $m/2$ and m cluster centers for the schools and bus stops each (see Park et al. (2012) for a detailed description of the instances). For the first type, 62% were improved or maintained unchanged and for the second type 69% were improved or maintained unchanged.

Algorithm 5 Approximate solution for the subproblem

```
function GetShortestPath(w, ρ, π)          ▷ w: bus-class; ρ, π: dual values of master problem
    U ← set of students that can be accommodated in bus-class w
    for i = 1 to n do
        t ← travel time from stop i to its corresponding school
        T[i] ← 1 − ρ[w] + ε × t − π[i]
    end for
    r ← new route containing start and end depot
    while U.size > 0 do
        c ← r.cost
        for each student i in U do
            c′ ← TryAddStop(r, i)          ▷ Get new cost of r if stop i is added
            m ← number of seats requested in stop i
            S[i] ← m × (T[i] − (c′ − c))          ▷ Calculate the saving
        end for
        if mode = deterministic then
            i* ← arg max{S[i]}
        else if mode = random then
            i* ← SelectStop(S, L)
        end if
        r.add(i*)
        U.remove(i*)
        if count(S[i] > −∞) = 0 then
            break
        end if
    end while
    return r          ▷ Returns the corresponding shortest path
end function
```

Table 1: Computational results for instances type RSRB

Instance	Max ride time	Schools	Stops	Students	Edges		Vehicles				CPU time	
					Total	Valid	Braca	Park	Initial	CG	Initial	CG
RSRB01	2700	6	250	3409	66564	7846	36	30	29	26	0.3	2.4
RSRB02	2700	12	250	3670	69696	6136	34	29	28	25	0.5	2.1
RSRB03	2700	12	500	6794	264196	17880	77	56	53	53	0.7	7.8
RSRB04	2700	25	500	6805	277729	18141	78	59	57	55	0.8	3.2
RSRB05	2700	25	1000	13765	1054729	65206	118	98	94	94	2.9	18.4
RSRB06	2700	50	1000	12201	1106704	58822	106	89	83	83	2.5	10.7
RSRB07	2700	50	2000	26912	4210704	191296	201	154	153	153	5.4	35.3
RSRB08	2700	100	2000	31939	4418404	217929	212	157	155	155	10.4	33.2
RSRB01	5400	6	250	3409	66564	16422	31	27	25	25	0.2	1.7
RSRB02	5400	12	250	3670	69696	12648	29	23	24	23	0.4	4.2
RSRB03	5400	12	500	6794	264196	52667	61	47	46	46	0.9	6.3
RSRB04	5400	25	500	6805	277729	48255	60	46	44	44	1.1	4.7
RSRB05	5400	25	1000	13765	1054729	179930	86	79	78	78	3.5	38.2
RSRB06	5400	50	1000	12201	1106704	167578	81	72	66	66	4.3	28.4
RSRB07	5400	50	2000	26912	4210704	600950	163	134	132	132	11.7	107.1
RSRB08	5400	100	2000	31939	4418404	697900	186	142	137	137	16.6	39.1
% Worsened instances:									6%	0%		
% Unchanged instances:									0%	6%		
% Improved instances:									94%	94%		

Table 2: Computational results for instances type CSCB($m/4$,$m/2$)

Instance	Max ride time	Schools	Stops	Students	Edges		Vehicles				CPU time	
					Total	Valid	Braca	Park	Initial	CG	Initial	CG
CSCB01	2700	6	250	3907	66564	12192	39	30	30	27	0.4	6.8
CSCB02	2700	12	250	3204	69696	11443	38	30	27	26	0.5	6.5
CSCB03	2700	12	500	6813	264196	25887	69	55	54	54	1.6	14.6
CSCB04	2700	25	500	7541	277729	19996	84	62	58	58	0.6	3.2
CSCB05	2700	25	1000	16996	1054729	77842	157	116	117	117	2.2	13.5
CSCB06	2700	50	1000	18232	1106704	66231	138	120	117	117	3.8	18.2
CSCB07	2700	50	2000	27594	4210704	211285	229	175	173	173	12.5	81.9
CSCB08	2700	100	2000	27945	4418404	213903	226	175	173	173	15.5	30.7
CSCB01	5400	6	250	3907	66564	34660	33	24	25	25	0.2	10.8
CSCB02	5400	12	250	3204	69696	29461	26	22	23	23	0.8	5.5
CSCB03	5400	12	500	6813	264196	88767	50	41	43	43	2.3	29.0
CSCB04	5400	25	500	7541	277729	59236	67	43	43	43	1.3	4.8
CSCB05	5400	25	1000	16996	1054729	277938	134	97	99	99	4.5	18.0
CSCB06	5400	50	1000	18232	1106704	198223	110	102	100	100	3.7	28.7
CSCB07	5400	50	2000	27594	4210704	764237	179	133	140	140	16.0	98.4
CSCB08	5400	100	2000	27945	4418404	722483	174	141	139	139	18.1	57.0
% Worsened instances:									38%	38%		
% Unchanged instances:									13%	6%		
% Improved instances:									50%	56%		

Notice that there is a significant difference between the performance of our work over the instances type *RSRB* and *CSCB*. Recall from Park et al. (2012) that instances of type *RSRB* have randomly distributed schools and bus stops, whereas instances of type *CSCB* have schools and bus stops gathered together in several clusters. Thus, we can infer that our procedure is more appropriate for cases where the dispersion of the location of students and school is very high. This allows to take great advantage of the mixed ride model. On the contrary, clustered students that attend the same school does not follow the same logic, and a single load model may be more appropriate. Despite the type of instance, as the number of students increases it becomes more computationally expensive to generate an initial solution and to improve it. Particularly, we can see how it becomes difficult to improve our initial solution in bigger instances.

For the real instance involving our work with WCSD, we run our procedure over 4 instances with the geographical dispersion seen in Figure 2. The results are shown in Table 4. Our results here are compared with the current situation at the school district for the corresponding school year. We can see that in all instances our solution outperforms the current practice, having a minimal 10% improvement (up to 20%)on all 4 of them and in a reasonable computation time. Recall that out approach takes great advantage of the mixed load strategy for building the initial solution and for the improvement via column generation.

6. Conclusion and further research

In this work, we modeled and presented a solution scheme for the mixed load routing problem for special education students. Given the nature of dispersed locations of the stops and schools for this class of problem, we took advantage of the mixed ride strategy that allow a bus carries students for different schools, and modeled the problem accordingly.

Special education students may require specially equipped busing for wheelchairs. Therefore, our model supports the use of a heterogeneous fleet of buses. In order to find the appropriate amount of regular and specially equipped buses to be leased, we start with a overestimated number of buses available to solve the problem and let the optimization choose the appropriate amount for each class of buses.

In addition, our formulation responds to both morning and afternoon problem in a uniform way. Provided that for this problem we allow mixed ride and that the schools have different start and end times, it is not appropriate to use the same sequence of stops for the morning in the afternoon as suggested in previous work. See Park and Kim (2010) for a discussion in morning versus afternoon problems.

Table 3: Computational results for instances type CSCB($m/2$,m)

Instance	Max ride time	Schools	Stops	Students	Edges		Vehicles				CPU time	
					Total	Valid	Braca	Park	Initial	CG	Initial	CG
CSCB09	2700	6	250	4148	66564	15223	32	36	25	24	0.7	14.0
CSCB10	2700	12	250	4217	69696	6217	33	39	30	28	0.2	1.8
CSCB11	2700	12	500	5996	264196	22773	58	68	57	53	1.1	6.5
CSCB12	2700	25	500	6098	277729	19095	77	79	65	62	0.7	4.1
CSCB13	2700	25	1000	14163	1054729	58120	165	158	127	124	1.8	22.4
CSCB14	2700	50	1000	16294	1106704	64553	137	133	106	106	3.6	16.4
CSCB15	2700	50	2000	27428	4210704	214164	247	253	200	200	9.5	65.8
CSCB16	2700	100	2000	24051	4418404	237488	208	213	175	175	11.9	31.8
CSCB09	5400	6	250	4148	66564	34148	26	30	23	23	0.7	7.3
CSCB10	5400	12	250	4217	69696	18794	30	32	25	24	0.4	1.7
CSCB11	5400	12	500	5996	264196	73050	50	50	42	42	1.0	7.4
CSCB12	5400	25	500	6098	277729	56640	55	57	48	48	1.1	15.0
CSCB13	5400	25	1000	14163	1054729	193375	135	124	105	105	2.9	35.1
CSCB14	5400	50	1000	16294	1106704	187565	117	109	92	92	4.0	28.6
CSCB15	5400	50	2000	27428	4210704	750416	215	199	164	164	12.8	128.3
CSCB16	5400	100	2000	24051	4418404	862815	158	148	133	133	22.0	73.4
% Worsened instances:									31%	31%		
% Unchanged instances:									19%	13%		
% Improved instances:									50%	56%		

Table 4: Computational results for real instances of WCSD

Instance	Max ride time	Schools	Stops	Bus classes	Edges		Vehicles			Buses saved	CPU time	
					Total	Valid	Current	Initial	CG		Initial	CG
2013-2014 AM	3600	39	227	2	71824	49801	41	36	32	9	0.4	4.2
2013-2014 PM	3600	39	225	2	70756	51157	36	33	31	5	0.3	9.6
2014-2015 AM	3600	33	189	2	50176	35818	32	26	26	6	0.3	17.8
2014-2015 PM	3600	34	200	2	55696	40871	29	27	26	3	0.2	3.3

By using a customized column generation procedure, we solved approximately a set of randomly generated instances made available by Park et al. (2012) and a set of instances from a real school district in Western New York. We found that our approach responds better to instances where the dispersion and the dispersion and variation of the locations of the students and the schools is high. One reason for this is that our approach builds the routes allowing mixed ride from the beginning, whereas Park et al. (2012) approach is to improve a set of route initially built with homogeneous load. Validated by instances in a real school district, our proposed approach could save $10 \backsim 20\%$ buses for special education students, as compared to the existing bus operations.

References

Braca, J., Bramel, J., Posner, B., Simchi-Levi, D., 1997. A Computerized Approach to the New York City School Bus Routing Problem. IIE Transactions, 693–702.

Caceres, H., Batta, R., He, Q., 2014. School bus routing with stochastic demand and duration constraints. submitted to Transportation Science.

Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12 (4), 568–581.

Kamali, B., Mason, S., Pohl, E., 2013. An Analysis of Special Needs Student Busing. Journal of Public Transportation, 21–45.

Park, J., Kim, B.-I., Apr. 2010. The school bus routing problem: A review. European Journal of Operational Research 202 (2), 311–319.

Park, J., Tae, H., Kim, B.-I., 2012. A post-improvement procedure for the mixed load school bus routing problem. European Journal of Operational Research 217 (1), 204–213.

Ripplinger, D., 2005. Rural school vehicle routing problem. Transportation Research Record: Journal of the Transportation Research Board (1922), 105–110.

Russell, R., Morrel, R., 1986. Routing special-education school buses. Interfaces 1986 (October), 56–64.