Cincinnati Public School Bus Routing and Bell Time Optimization

Dr. Shuai Wang

09/28/2019

Contents

1 Overview

1.1 Single school

We first assign the stops, which optimize the number of stops for each school and the total student walking distance. This is similar to a facility location problem.

Single school routing:

1. Generate a set of single feasible trip using randomized greedy heuristic, making sure each stop is served by a nonempty set of feasible trips.(func: greedy)
2. Run the heuristic N times, so that each stop is covered by N different feasible trips. (func: generateRoutes)
3. Find the best sub set of trips that covers each stop, by using a minimum cover problem. Some stops may be servered by more than one trip. (func: bestRoutes)
4. Trim the stops covered with multiple trips. (func: buildSolution)

The output of single school routing is set of trips that covers every stop excatly once for each school.

The senarios are the tradeoff parameters λ to obtain an array of varied routing scenarios for each school.

For each school, we end up with a set of routing scenarios, where each scenario is a complete set of trips that covers every stop exactly once.

Each senario is located in a different region of the Pareto front between two objectives, the number of buses and the average student travel time.

1.2 Multiple schools

1.2.1 Scenario selection

The input is a set of scenarios Rs of a size hs for each school S. Because what is optimal for one school may not be optimal for the entire system. We consider several several scenarios on the Pareto frontier of two objectives: numer of buses and average riding time. The trade-offs makes sense because shorter routes are more easily connected into bus schedules.

Then we first jointly select one scenario for each school in a way that favors maximal re-use of buses from school to school, by using a network flow stucture that minimized the objective across the whole district: **minimum number of buses out of Yard**.

Notes: This only tells which scenario between the school can be linked, namely some "Departure nodes" from school1 can be linked to some "Arrival nodes" from the school2. It **does not** tell which specific trips are linked to the school2. On the contrary, it might have multiple trips from same scenario of school1 can be linked to school2.

1.2.2 Bus Scheduling

Given the one routing scenario for each school, we can then solve another integer optimization problem to identify a trip-by-trip iternary for each bus on the fleet. In this final subproblem, we optimize the numeber of buses jontly in the morning and in the afternoon.

Notes: The input of the Bus selection is an array of scenario for each school. The trips in the scenario can have multiple cases where trip can be linked from the school1 and school2. Consequently, bus scheduling help to choose the minimum trips which linking the school1 and school2. E.g. scenario1 from school1 has 3 different trips {2,12,39} can be linked to school2's scenario2 trip {32}: {2,12,39} -> {32}, the bus scheduling will only keep one and only one from {2,12,39} to be linked to {32}.

2 Routing: Bi-objective Routing Decomposition (BiRD)

Decomposing the routing problem:

1. single-school
   (a) stop assignment
   (b) single-school routing
2. multi-school
   (a) scenario selection
   (b) bus scheduling

2.1 Stop Assignment

2.1.1 Set/Parameters

Identifier	Index/Set Domain	Comment
schools	S	set of schools
students	P	set of students
P_s	$s \in S$	set of students attending school s
Locations	C	set of all locations can serve as bus stops
C_p	$C_p \subseteq C$	Allowed bus stops for student $p \in P_s$
$d_{p,c}$		walking distance for student p to a stop $c \in C_p$

For student p with special needs may require a "door-to-door" pickup: the set C_p is a singleton c, where $d_{p,c} = 0$.

The objective is to minimize the number of stops for each schools and the total walking distance. This is similar to a facility location problem.

2.1.2 Variable

z_c: binary whether stop c is selected for school S.

$y_{p,c}$: binary whether student p is assigned to stop c.

2.1.3 Objective and Constraints

For each school S, minimize the total number of stops and walking distance:

$$min \sum_{c \in C} z_c + \beta \sum_{p \in P_s} \sum_{c \in C_p} d_{p,c} y_{p,c}$$

subject to:

1. student p is assigned to c only if stop c is selected for school s.

$$y_{p,c} \leq z_c, \quad \forall p \in P_s, c \in C_p$$

2. Each student is assigned to 1 stop

$$\sum_{c \in C_p} y_{p,c} = 1, \quad \forall p \in P_s$$

Where β is the trade-off of the total number of stops and walking distance.

2.2 Single school routing

After the stop assignment, C_s is the set of bus stops which serve school S. $n_{c,S}$ is number of students from school S at a stop c.

2.2.1 Bus

Identifier	index/set	comment
Bus Type	b/B	
Capacity Q_b	b	
Wheelchair spots W_b	b	fixed number of wheelchair
Bus depot Y		**stored during the night or day???**

2.2.2 Trip

1. Time

Identifier	comment
$t_{c,S}^{pick}$	length of time to pick all p for S at c; function of $n_{c,S}$
t_S^{drop}	length of time to drop all p at school S; fixed
$t_{l_1,l_2,\tau}^{drive}$	driving time from l_1 to $l_2 \in S \cup C \cup Y$ at time τ
τ_S	all buses must arrive at school S at τ_S
$t_S^{drop} + \tau_S$	bus free to leave at this time
T^{max}	maximum time that students allowed to spend on the bus

$t_{l_1,l_2,\tau}^{drive}$ are deterministic and known, and usual got from API such as Google Map. Due to reality, such as accident, and to mitigate the impact of the traffic, the drop-off times t_S^{drop} is artificially increase to say 10-15 minutes even it only takes 3 minutes.

2. Trip/Route A trip/route is a ordered sequence of stops visited or served, by a bus. For school S, a feasible trip/route:

   (a) no student spends more than T^{Max} between pickup and drop-off.

   (b) there exists a type of bus with capacity and carry all students assigned to the stops severed by the trip.

   Given a feasible route R:

   - $B_R \subseteq B$: the set of types of buses that have the necessary capacity to serve the trip.

- T_R: service time of the trip, the time between arrival at the stop to the final school
- T_S: A set of feasible trips, each stops $c \in C_s$ is served by a nonempty set of feasible trips $T_c \subset T_S$.

  For example: School S has 4 different sets of feasible trips, stop c4 can be served by two differents trips 3 and 5, so $T_c = 3, 5$.

  The heuristic returns a set of trips T what covers each stop **exactly once**. Then we run it N times to build a set of feasible trips where each stop is covered by serveral trips. The heuristic returns N different solutions and each stop will be covered by N different feasible trips.

Trips in T_S indexed by $1, ..., m_S$.

For given set of trips $T \subseteq T_S$: $I(T)$ is the subset of $\{1, ..., m_S\}$ corresponding to trips in T.

For example: School S has 4 different set of feasible trips, with 25 different trips in total. Stop c4 can be served by two differents trip 3 and 5, so $I(T_c) = I(T_4) = 3, 5 \subset 1, ...25$

2.2.3 Find best set of routes

1. Variable r_i: binary whether trip i is selected $\forall i \in \{1, ..., m_S\}$
2. Objective
$$min \quad \lambda \sum_{i=1}^{m_S} r_i + \sum_{i=1}^{m_S} r_i \Theta_i$$
3. Constraints every stop must be served by at least one trip.
$$\sum_{i \in I(T_c)} r_i \geq 1, \quad \forall c \in C_S$$

   where, for each trip i in T_s, $\Theta_i = \sum_{p:c_p \in C_i} \theta_p^{(i)}$, where C_i reprents the list of stops served by the $i-th$ trip i,

   $\theta_p^{(i)}$ corresponds to tiem spent by student p on the bus druing trip i.

   λ controls the importance of the number of trips relative to the total time students on the bus.

   In the optimal solution above, some stops may be served by more than one trip. Set of trips $T_c^{opt} \subset T_S$ serving each such stop c. For each

$i \in I(T_c^{opt})$, we compute the improvement δ_i to the objective that would be obtained by removing stop c from trip i.

2.3 Scenario Selection

scenario selection graph with $\{\mathcal{N}, \mathcal{A}\}$ where $\mathcal{N}$ consists of nodes:

2.3.1 Node

Type	Notation	comment
depot	y	
trip	$\rho_{S,h,R}$	for each school S, and routing scenario $1 \leq h \leq h_S$ and each trip $R \in \mathcal{T}_S^h$
available node	$a_{S,h}$	for each school S, and routing scenario $1 \leq h \leq h_S$

2.3.2 Arcs

The set of arcs $\mathcal{A}$ includes an arc from depot to trip; trip to corresponding available node; from available node back to the depot.

Also, an arc from each available node to each trip node where trip $R' \in \mathcal{T}_{S'}^{h'}$ is compatible with the a bus staring from school S. $\tau_S + t_S^{drop-off} + t_{S,c_{R'}^{start},\tau_S}^{drive} + \mathcal{T}_R \leq \tau_{S'}$

For a note $i \in \mathcal{N}, let\, \mathcal{I}(i) \subseteq \mathcal{N}$ be the in-neighborhood of the node i, and $\mathcal{O}(i) \subseteq \mathcal{N}$

2.3.3 Variable

1. f_i^j: integer flow variables for each arc(i,j).

2. $\mathcal{Z}_{S,h}$: whether to select scenario h for school S or not

2.3.4 Objective

Minimize the total number of buses corresponds to minimizing the total flow out of the yard node y.

$$minimize \sum_{s \in S} \sum_{h=1}^{h_S} \sum_{R \in \mathcal{T}_S^h} f_y^{\rho_{S,h,R}}$$

2.3.5 Constraints

flows along all arcs must be integral.

1.
$$f_{a_{S,h}}^{\rho_{S,h,R}} = z_{S,h}, S \in \mathcal{S}, 1 \leq h \in h_s, R \in \mathcal{T}_h^S$$

2.
$$\mathcal{Z}_{S,h} \in \{0,1\}, \quad \forall S \in S, 1 \leq h \leq h_S$$

3.
$$\sum_{h=1}^{h_S} Z_{S,h} = 1, \quad \forall S \in \mathcal{S}$$

4.
$$\sum_{i \in \mathcal{I}(i)} f_j^i = \sum_{j \in \mathcal{O}(i)} f_i^j, \quad \forall i \in \mathcal{N}$$

5.
$$f_{i,j} \in \mathbb{Z}_+ \quad \forall (i,j) \in \mathcal{A}$$

The binary variable $Z_{S,h}$ is 1 if $\mathcal{T}_S^h$ is the selected set of trips for school S.

3 exactly one set of trips/one scenario is selected for each school.
4 ensures conservation of flow f_i^j at each node,

- At the depot node y, it means that buses leaving the depot must eventually come back.
- At a trip node $\rho_{S,h,R}$, it means that a bus serving trip R must then become available at school S at time $\tau_S + t_S^{drop-off}$
- At an availability node $a_{S,h}$, it means that a bus that is available at a school after serving a trip must either return to the yard or serve another trip.

2.4 Bus Scheduling

To determine the final bus schedules given exactly one routing routing scenario for each school. We define a "bus selection graph" $(\overline{\mathcal{N}}, \overline{\mathcal{A}})$, where the set of nodes $\overline{\mathcal{N}}$ consists of:

1. a node $y_{b,\updownarrow}$ for each bus type $b \in \mathcal{B}$ and each bus depot locations $\updownarrow \in \mathcal{L}$.

2. a trip node $\rho^{AM}_{S,R,b,l}$ for each school S, each morning trip R in the selected scenario, each bus type b, and each depot location l

and set of arc $\overline{\mathcal{A}}$ consists of:

1. an arc to and from the depot node $y_{b,l}$ for each trip node $\rho^{AM}_{S,R,b,l}$ and $\rho^{PM}_{S,R,b,l}$
2. an arc from each trip node $\rho^{AM}_{S,R,b,l}$ and $\rho^{PM}_{S,R,b,l}$ where trip $R^{'}$ for school $S^{'}$ is time-compatible with a bus starting from school S.

Each arc(i,j) has a cost(i,j), and bus cost(b) for each bus type.

2.4.1 Variables

1. BusFlow f^j_i all the edges for the graph. Binary
2. YardCapacity $K_{b,l}$ positive integer

2.4.2 Objective

$$minimize \quad \gamma \sum_{b\in\mathcal{B}} \sum_{l\in\mathcal{L}} c_b K_{b,l} + (1-\gamma) \sum_{(i,j)\in\overline{\mathcal{A}}} c_{i,j} f^j_i$$

2.4.3 Constraints

1. $$\sum_{i\in\overline{\mathcal{I}}(i)} f^i_j = \sum_{j\in\overline{\mathcal{O}}(i)} f^j_i, \quad \forall i \in \overline{\mathcal{N}}$$

   enforces flow conservation, which simply means that buses must leave the yard, serve at least one trip, and then return to the yard.

2. $$\sum_{b\in\mathcal{B}} \sum_{l\in\mathcal{L}} \sum_{i\in\overline{\mathcal{I}}(\rho^{\amalg}_{\mathcal{R},\mathcal{S},\lfloor,\ddagger})} f_i^{\rho^q_{R,S,b,l}} = 1, \quad \forall s \in S, \forall R \in \mathcal{T}^q_S, q \in \{AM, PM\}$$

   enforces that each trip is served by exactly one bus, from one particular depot location and of one particular type.

3. $$\sum_{i\in\mathcal{O}^{\amalg(\dagger_{\lfloor,\ddagger})}} f_y b, l^i \leq K_{b,l}, \quad \forall b \in \mathcal{B}, l \in \mathcal{L}, q \in \{AM, PM\}$$