Varying Routes for the Bus Driver's Sanity Problem

Jeremy Dyal, †
Paul Hadavas, ‡
Greg Knofczynski §

Abstract

In this article, we provide results involving the bus driver's sanity problem. In particular, we explore optimal and near-optimal routes for an assortment of possible networks. The original problem was presented as a simple path wherein the weights on the edges would vary as a path is formed. We arrive at solutions for the situations pertaining to tree and cul-de-sac structures

Introduction

The bus driver's sanity problem was first presented by Will [1]. The author posed this graph theoretic problem in the context of a bus driver needing to minimize the total time to unload the kids on the bus not in minutes but in terms of kid-minutes. The main premise is that brief exposure with many kids is worse than long exposure to a few kids. Hence, the quantity to be minimized is cumulative time the kids spend on the bus. Will examined the situation when the bus travels along a single road, making stops either to the east or west of the school. With the use of dynamic programming, he developed an algorithm to find the optimal route for the bus driver. In this article we expand upon the possible routes the bus driver could take. The routes included are tree-like structures with multiple stops branching off the school's location and cul-de-sacs where stops are located off the main road.

In particular, we begin with the case of having to drop off one student per stop and one minute travel times for each edge in the path. We prove that the optimal route in such a structure is one where subtrees need to be completed before moving on to the next subtree. In the case of the cul-de-sacs, we develop a greedy heuristic to achieve a near-optimal solution

Routes with One Student Per Stop

The first result considers a route which involves a tree structure wherein initially all stops require one kid to be dropped off. Also, all travel times between stops and between the bus and all possible initial stops are one minute (see Figure 1).

Figure 1: One student per stop and one minute travel times between stops.

Theorem *Let T be a tree graph with the root node representing the bus full of kids and every other node demanding one student and each edge in the tree is* *associated with a travel time of one minute. An optimal route will be one that completes all subtrees of a node before moving to a new subtree.*

Proof: Without loss of generality, assume the root node has two subtrees. Let L and R be the left and right subtrees of T with $n(L)$ and $n(R)$, the number of kids to be dropped off in each subtree, respectively. The time required to drop off all the kids when completing the left subtree first will be

$t(L) + 2\, n(L)\, n(R) + t(R)$,

where $t(L)$ and $t(R)$ are the times required to drop off the kids in their respective subtrees and $2n(L)n(R)$ is the time that the kids belonging to the right subtree spend in the left subtree.

There are two other types of solutions to consider: those isomorphic to starting in the left subtree and those that are not. A route will be isomorphic if the path it chooses always completes a subtree before beginning down a new subtree with the difference being the order in which the subtrees are examined. For example, the time spent traversing T if subtree R had been chosen first and then subtree L would be $t(R) + 2n(L)n(R) + t(L)$, exactly the same time as before. The quantity $2n(L)n(R)$ now represents the time that the kids belonging to the left subtree spent in the right subtree. Thus, those solutions isomorphic to traversing subtrees from left to right will have the same travel time.

The other solutions to consider are when a subtree is abandoned before completely dropping off all students belonging to that subtree. In this case, someone from the left subtree spends more time on the bus than would be saved by dropping off students in the right subtree. Consider the case where only $n(L^*)$ students are dropped off in the left subtree and $n(L^{**})$ remain be dropped off on the left but the left subtree is abandoned for the right subtree. The time required would be $t(L^*) + 2\, n(R)\, n(L^*) + 2\, n(L^{**})\, n(L^*) + t(R) + 2\, n(L^{**})\, n(R) + t(L^{**})$, where $t(L^*) + 2\, n(R)n(L^*) + 2\, n(L^{**})\, n(L^*)$ is the time spent dropping off the $n(L^*)$ left subtree students, $t(R) + 2\, n(L^{**})\, n(R)$ is the time the remaining $n(L^{**}) + n(R)$ students spend in the right subtree, and $t(L^{**})$ is the time to drop off the remaining students in the left subtree once they return to the top node. Since $t(L^*) + t(L^{**}) = t(L)$ and $2\, n(R)(n(L^*)+n(L^{**})) = 2\, n(R)\, n(L)$, the time spent on this route will be greater than or equal to the time spent completing subtrees before moving on to the next subtree. Hence, the route that completes all stops in a subtree before moving on to the next will yield an optimal route.

Corollary *An optimal route for T will be the depth-first traversal of that tree*.

Routes with Stops off the Main Road

We now consider the case where the stops are located off the main road and each stop will have multiple students needed to be dropped off. Travel times are found between turning points and stops and between successive turning points. This cul-de-sac problem with all stops located off the main road is seen in Figure 2. The stops are labeled with letters and contain the number of students to be dropped off.

Figure 2: Cul-de-sac problem where all stops are located off the main road.

In this instance, the number of possible routes is 6! or 720. In general, when there are n stops, there will be $n!$ possible routes to consider. Since this grows too quickly to enumerate every possible route, we apply a greedy heuristic to achieve a good solution.

Step 0: Start with all stops unvisited and the current location of the bus is the school.

Step 1: Find the ratio of "travel time to stop" divided by "number of students being dropped off" for each unvisited stop from the current location. Then select the stop with the smallest ratio as the next visited stop in the route and, hence, also as the new current location of the bus.

Step 2: If all stops have been visited, go to Step 3 with the completed route as the incumbent solution. Otherwise, repeat Step 1. Note: in the instance that two stops tie for the smallest ratio, the stop within the same subtree as the current stop should be selected. If both stops or neither stop are within the same subtree as the current stop, select the one with the larger number of students to be dropped off.

Step 3: Switch successive pairs of stops for possible improvements in kid-minutes. If an improved route is found, update incumbent and continue with step 3. Once all possible successive pairs have been switched and no improvements have been attained, go to step 4.

Step 4: End with a good (if not optimal) solution.

For the example presented in Figure 2, the first stop in the incumbent solution is determined by computing the ratios for stops *a*, *b*, *c*, *d*, *e*, and *f* as 7/2, 7/8, 23/6, 24/4, 24/5, and 26/3, respectively. Since stop *b* has the smallest ratio, it is selected as our first stop. Next, new ratios are determined from stop *b* to the remaining stops and were found to be 10/2, 30/6, 31/4, 31/5, and 33/3 for stops *a*, *c*, *d*, *e*, and *f*, respectively. There is a tie between stops *a* and *c* but since stop *a* is in the same subtree as the current stop, *b*, it is selected as the next stop. The third set of ratios, determined from stop *a* are 30/6, 31/4, 31/5, and 33/3 for stops *c*, *d*, *e*, and *f*, respectively. Stop *c* has the smallest ratio and is thus selected as the third stop. From stop *c*, the remaining ratios for stops *d*, *e*, and *f* are 21/4, 21/5, and 23/3, respectively, thus making stop *e* the fourth stop. The last two ratios are 14/4 for stop *d* and 4/3 for stop *f*. Hence, *f* is the fifth stop and *d* is selected as the last stop. Therefore, the incumbent solution is *bacefd* with number of kid-minutes equal to (28)(7) + (20)(10) + (18)(30) + (12)(21) + (7)(6) + (4)(16) = 1294. The routes created by switching successive pairs of stops are: *abcedf, bcaefd, baecfd, bacfed,* and *bacedf.* None of these routes lead to a reduced number of kid-minutes and the algorithm stops. In this example, the incumbent solution found from the greedy heuristic is not only a good route but also the optimal route.

Also, it should be pointed out that the optimal solution drops off the fewest number of students at the second stop. The time those students spend while dropping off the other students outweighs the time the other students will spend going to the second stop.

Further Study

One area we would like to more closely study is the efficacy of the heuristic. Although our example yielded an optimal solution, we realize that will not always be the case. Comparing our good or near-optimal answers with the optimal answers for a range of problem sizes would clarify how well our algorithm performs.

We would also like to investigate other types of routes. In this paper, we examined tree and cul-de-sac situations. A loop or cycle situation can be examined to determine when the bus would drop kids off in either a complete clockwise or counter-clockwise route or if the bus should reverse itself throughout the cycle. Grid situations seem more practical for city settings. Whereas the original problem was posed as moving only in two directions, east or west, we would examine a case where all four directions (east, west, north, and south) are possible.

† *Jeremy Dyal,* Armstrong State University, Savannah, GA, USA
‡ *Paul Hadavas, Ph.D.* Armstrong State University, Savannah, GA, USA
§ *Greg Knofczynski Ph.D.* Armstrong State University, Savannah, GA, USA

Figure 1

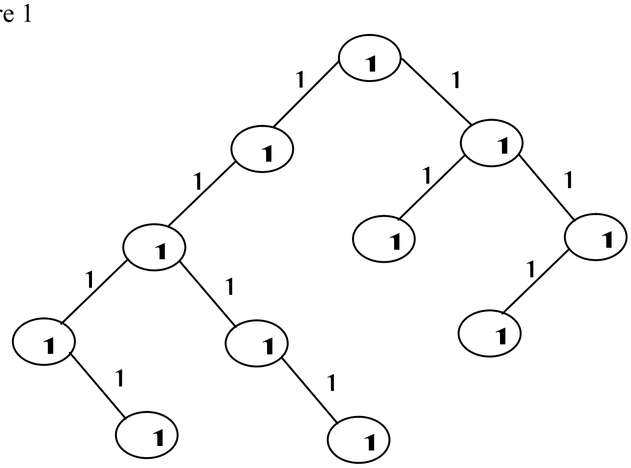
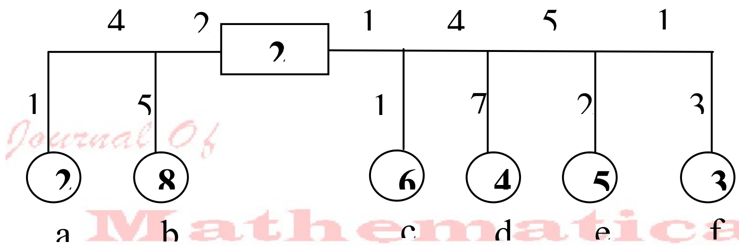


Figure 2



Reference

Will, Todd, *The Bus Driver's Sanity Problem*, College Mathematics Journal 30:3(1999), pp.187-194.