12-2011

Integrated Planning for Rural Rail Transportation

Jeffrey Cohan Gwaltney
University of Arkansas, Fayetteville

Integrated Planning for Rural Rail Transportation

Integrated Planning for Rural Rail Transportation

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Industrial Engineering

By

Jeff Gwaltney
University of Arkansas
Bachelor of Science in Mathematics, 2009

December 2011
University of Arkansas

Abstract

Population growth in rural areas has led to new interest in rail transportation. Planning a passenger rail system involves numerous difficult decisions, most representing a trade-off between customer service and cost. In this work, we attempt to integrate many of these planning decisions. We consider strategic decisions such as station location and vehicle procurement, as well as tactical issues that include vehicle scheduling. Our integrated model exploits the linear network structure that best suits many rural American communities, including Northwest Arkansas. Due to the intractability of the integrated rail planning problem, we have developed a customized heuristic approach to solve real world instances. In our case study, we have applied our model and solution methodology to study the possibility of implementing a passenger rail system in Northwest Arkansas. Our work represents the first steps in a passenger rail feasibility study for Northwest Arkansas, while providing new mathematical modeling and solution methodology contributions to the area of transportation research.

This thesis is approved for recommendation
to the Graduate Council.

Thesis Director:

____

Dr. Chase Rainwater

Thesis Committee:

____

Dr. Ashlea Milburn

____

Dr. Sarah Root

Thesis Duplication Release

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed ______________________________

Jeff Gwaltney

Refused ______________________________

Jeff Gwaltney

Contents

List of Figures

List of Tables

1 Introduction

Rising fuel prices (see Figure 1) and growing populations in rural areas have led to interest in rail transportation as an environmentally conscious alternative to highway expansion for the alleviation of traffic congestion. Northwest Arkansas (NWA) is a prime example of this phenomenon. In fact, NWA was the sixth-fastest growing metropolitan area from 1990-2000 with a growth rate of 47.5% [27]. Though the growth rate has decreased slightly since 2000, the NWA population could surpass 1 million within 25 years if current growth rates continue [1]. Further evidence suggesting NWA as a natural candidate for passenger rail is the advantageous distribution of the area's population. Furthermore, a study by the University of Arkansas Community Design Center [8] points out that two-thirds of all current NWA residents live within one mile of existing rail right-of-way.

Passenger rail systems of differing sizes and capabilities are available to city planners. Common amongst alternatives are Light Rail, Heavy Rail and Commuter rail systems. According to the American Public Transportation Association (APTA), Light Rail systems (also known as streetcar, tramway, or trolley systems) typically feature electrically driven vehicles with power drawn from an overhead electric line. The APTA defines Heavy Rail systems (also known as metros or subways) to be those operating on an electric railway with the capacity for heavy volume of traffic. Finally, the APTA states that Commuter Rail systems are usually located along routes of current or former freight railroad, that their trains may be electric or diesel driven, and that they typically connect a metropolitan area to its suburbs [16]. The methods we describe in this work could be applied to any of these system types. However, given the existing rail right-of-way through the heart of the area, the Commuter Rail model is most applicable to Northwest Arkansas.

In addition to rail systems differing by type and purpose, the configuration of any system plays a key role in determining its operational capabilities and challenges. Figure 2 depicts two such passenger rail configurations. Radial networks are common in urban settings, where populations are spread over vast areas, and throughout Europe and Asia, where rail systems connect cities in all directions. Because of the complexity of most urban/regional radial rail

[1] Recent growth rates based on data from the U.S. Census Bureau 2012 Statistical Abstract [6].

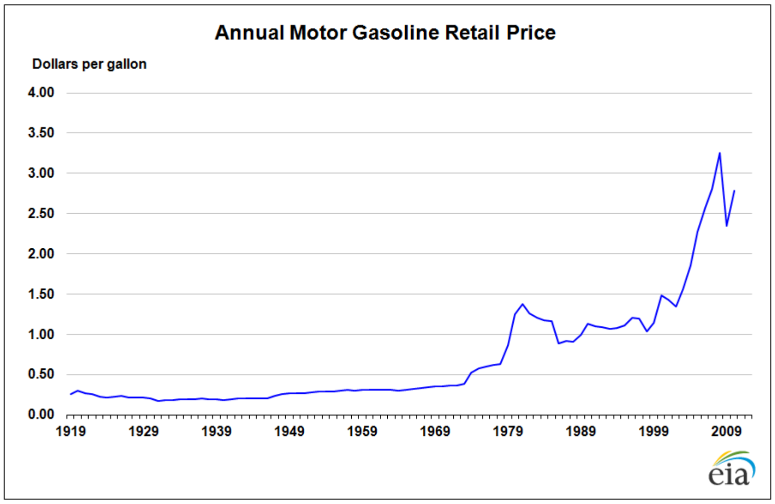
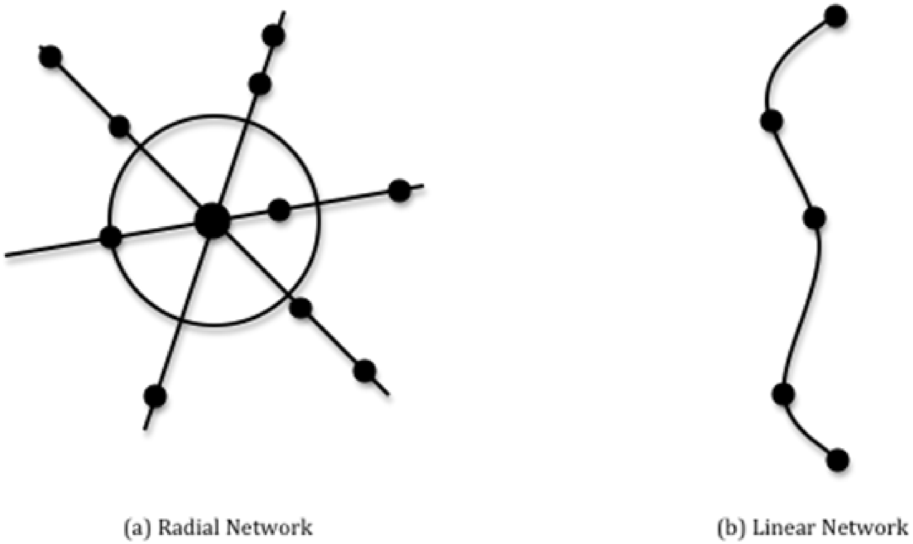


Figure 1: Historical Gasoline Prices in the U.S.
Source: U.S. Energy Information Administration [5]

networks, researchers have historically approached the rail planning problem hierarchically for the sake of tractability [11]. Any passenger rail system requires an extensive planning process that includes strategic, tactical, and operational decisions. Operational decisions typically concern day-to-day activities and schedule disruptions, tactical decisions are those with a 1-5 year impact (i.e. resource allocation), and strategic decisions are those with implications reaching beyond 5 years (i.e. resource procurement) [11]. Within each of these planning stages, numerous problems must be considered, as shown in Table 1. Authorities in the field have commented that the hierarchical planning approach fails to guarantee an optimal system, due to its inability to capture all interactions between various planning stages [10].

Some rural communities, especially those that have developed along a river, roadway, or historical rail line, lend themselves to the development of a passenger rail system that follows a single path, however. We will refer to these as linear networks (see Figure 2). Simpler networks with fewer required decisions may allow for the use of an alternative integrated planning process that simultaneously considers the set of all required decisions, yielding system-optimal solutions. Rural areas that naturally permit a linear network are prime candidates for this type of approach. In this work, we introduce a mixed integer programming model that integrates many of the strategic and tactical decisions outlined in Table 1. Since



(a) Radial Network

(b) Linear Network

Figure 2: Radial versus Linear network

Table 1: Rail Planning Decisions

Strategic	Number of Stations
	Station Locations
	Track Location
	Number of Vehicles
Tactical	Days & Times of Operation
	Vehicle Routes
	Projected Demand
	Vehicle Schedules
Operational	Crew Composition
	Crew Assignment
	Train Dispatching
	Delay Management

this integrated problem is difficult to solve using currently available computer hardware and software, we have developed a customized heuristic process to generate quality solutions for realistically-sized instances. Finally, we have applied our model and solution methods to study the possibility of implementing a passenger rail system in Northwest Arkansas.

2 Literature review

Rail planning has been studied extensively within the operations research community. However, the existing literature is limited in its applicability to rural settings. The method adopted by most researchers consists of separating rail planning decisions into subproblems and solving each individually [11]. The following well-studied subproblems result: network planning [24], line planning [10, 17], station location [25, 28], timetabling [12, 21, 22], vehicle scheduling [29], and vehicle routing [31]. Though many of these problems are studied from a deterministic standpoint, some researchers have developed models that incorporate the uncertainty involved in rail systems. For example, Kroon et al. develop train timetables that minimize the average delay associated with stochastic disturbances in [22] and [21]. In [23], List et al. consider uncertainty of future demand and operating conditions in their model meant to optimize fleet sizes. Researchers have attempted to solve these problems exactly in rare cases. In [14, 15] a modified branch-and-bound technique is employed to solve certain rail and bus scheduling problems. The authors exploit the structure of the problems LP in a way that would not extend to our integrated problem, however. Much of the literature focuses on heuristic development since these problems are often applied to very large systems. Various heuristic approaches have been applied to these problems including Lagrangian Relaxation [9, 26], Tabu Search [18, 19, 26], Neighborhood Searches [19, 26], and Genetic Algorithms [18, 19]. In these works, heuristics have been shown to be successful in generating quality solutions for many rail planning problems. No single heuristic has been applied to all of the problems that we have integrated, however. For an extended review of passenger rail research see [11, 13]. More recent research shows a continued interest in this area, but no serious work has been done to integrate these various subproblems. Instead, researchers have continued to assume that the rail planning process will follow a hierarchical structure. This hierarchical approach to rail planning is unavoidable when the network to be constructed is more complex, as is the case with most Asian and European networks. However, due to subproblem interactions, it is in the planners' best interest to integrate decisions when possible [10]. Our investigation found that integration of this type is not present in the existing literature. However, we contend that the configuration and reduced size of many rural communities, including the NWA region, open the door to a partial integration of the rail planning process. In this work, we have taken the first steps toward an improved planning tool for rural rail transportation over the methods currently found in the literature. We have integrated many of the problems outlined above by exploiting the linear network structure that is best-suited for many rural settings. Since this integrated problem is difficult to solve to optimality, we have developed a heuristic motivated by the neighborhood search concept. Neighborhood search heuristics and their variants are ubiquitous in the Operations Research literature. For a review of local search techniques, see [30].

3 Problem description

Our model is meant to assist planners as they make important strategic and tactical decisions about potential passenger rail systems. We assume that a rail right-of-way has been determined, and that a finite number of potential station locations have been identified along this right-of-way. This right-of-way features one track for each direction of travel, and forms a *linear network* as defined in Section 1. Two of the potential stations form the static endpoints for the potential system. One endpoint serves as the depot for the trains, where all trains begin and end each of their *loops* (see definition below). The opposite endpoint serves as the location where trains reverse their direction of travel.

Loop When a train departs the first station, traverses the entire track in one direction, travels the entire track in the opposite direction, and then returns to its original location, we will say that it has completed one loop.

Furthermore, we assume that origin, destination, and scheduling information is known or can be estimated for all potential customers. In this initial work, we have assumed deterministic customer demand. However, possibilities for stochastic variants of our problem are discussed in Section 7. Using this information, along with station and vehicle cost information, we have

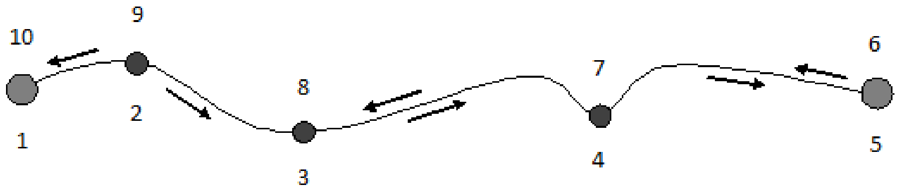


Figure 3: An Example Where $|\mathcal{L}| = 10$

developed a model to identify the station configuration, vehicle fleet size, and set of train schedules that will maximize the daily profit for the system. We focus on a single-day horizon, with time measured in minutes. To normalize costs, daily values for item procurement costs are estimated (in current dollars) using the item's purchase cost and estimated life of the item. For example, if it costs \$2,000,000 to procure a train that should remain in service for 20 years for 250 working days each year, the estimated daily cost for each train is \$400. Costs associated with installing any necessary track are not considered, because the right-of-way is assumed to exist in our scenario.

To model our problem, we consider a passenger rail network consisting of two parallel tracks, one for each direction of travel, connecting two fixed stations (endpoints of the linear network) where trains turn around and depart in the opposite direction. The set $\mathcal{L}$ contains two elements for each possible station location along the track, corresponding to the two directions of travel. Letting $L = |\mathcal{L}|$, station 1 and L both correspond to the first possible location, 2 and $L-1$ to the second possible location, and so on until the last possible station location, represented by $\frac{L}{2}$ and $\frac{L}{2}+1$. The cost of procuring a station at location $\ell \in \mathcal{L}$ is f_ℓ, where $f_\ell > 0$ for $l = 1 \ldots \frac{L}{2}$ and 0 otherwise. Figure 3 provides an illustrative example. In this example, five potential stations have been identified, including the two fixed endpoints. Therefore, $L = 10$ and the first station location is represented by 1 and 10 depending on direction of travel, the second by 2 and 9, and so on.

The set of trains that may potentially serve customers is denoted by $\mathcal{T}$, where $|\mathcal{T}| = T$. Associated with each train $\tau \in \mathcal{T}$ is a procurement cost c_τ, a per-loop operating cost v_τ, and a capacity u_τ. The time required for a train to travel from location $\ell - 1$ to ℓ is denoted

t_ℓ. The speed of a train as it travels between locations is assumed to be constant, thus t_ℓ is proportional to the length of the track between $\ell - 1$ and ℓ. In addition, we must account for the time required for a train to stop at location ℓ, denoted δ_ℓ, in the event that a station exists there. The set $\mathcal{K}$, where $|\mathcal{K}| = K$, consists of the loops that a train may take around the linear rail network. Here, K is a calculated upper bound on the possible number of loops that any train may need to take during the time horizon.

The set $\mathcal{G}$ is comprised of the set of potential passenger groups in the rail system. The number of passengers in group $g \in \mathcal{G}$ is denoted by P_g. The origin and destination for group g are denoted o_g and d_g, respectively. Each passenger group g has an arrival window $[a_g - b_g, a_g]$ associated with their destination, where a_g specifies the latest acceptable arrival time at destination d_g, and b_g specifies the maximum acceptable waiting time (where waiting occurs when a passenger arrives at their destination early). Thus, any train arriving at d_g within the arrival window of group g is eligible to serve all or some of the passengers in group g. We restrict passenger groups from being split between multiple trains. Finally, the system earns a daily revenue of r_g for serving a passenger in group g. Our system is assumed to operate H minutes per day.

The decision variables included in the model formulation representing our problem are defined in Table 2. The model follows.

Table 2: Decision Variables

Variable	Type	Description
z_ℓ	binary	1 if station constructed at ℓ, for $\ell = 1, \ldots, \frac{L}{2}$, 0 otw
$x^l_{g,\tau}$	binary	1 if group g assigned to train τ on k^{th} time τ visits o_g, 0 otw
$\rho^k_{g,\tau}$	integer	number of customers from group t served by train τ on k^{th} loop
y_τ	binary	1 if train τ is used, 0 otw
q^k_τ	binary	1 if train τ in use for k^{th} loop, 0 otw
$w^k_{l,\tau}$	continuous, ≥ 0	time at which train τ arrives at l for k^{th} visit
$n^k_{l,\tau}$	integer	number of passengers on train τ leaving location l for k^{th} time

$$\text{maximize} \sum_{k\in\mathcal{K}}\sum_{\tau\in\mathcal{T}}\sum_{g\in\mathcal{G}} r_g\rho^k_{g,\tau} - \sum_{\ell=1\dots\frac{L}{2}} f_\ell z_\ell - \sum_{\tau\in\mathcal{T}} c_\tau y_\tau - \sum_{k\in\mathcal{K}}\sum_{\tau\in\mathcal{T}} v_\tau q^k_\tau$$

subject to

$$z_\ell \leq z_1 \qquad \ell = 2\dots(L/2)-1 \tag{1}$$

$$z_\ell \leq z_{\mathrm{L}/2} \qquad \ell = 2\dots(L/2)-1 \tag{2}$$

$$x^k_{g,\tau} \leq z_{I(o_g)} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{3}$$

$$x^k_{g,\tau} \leq z_{I(d_g)} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{4}$$

$$q^k_\tau \leq y_\tau \qquad \tau\in\mathcal{T}; k\in\mathcal{K} \tag{5}$$

$$y_{\tau+1} \leq y_\tau \qquad \tau\in\mathcal{T}\setminus\{T\} \tag{6}$$

$$x^k_{g,\tau} \leq q^k_\tau \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{7}$$

$$\rho^k_{g,\tau} \leq P_g x^k_{g,\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{8}$$

$$\sum_{\tau\in\mathcal{T}}\sum_{k\in\mathcal{K}} x^k_{g,\tau} \leq 1 \qquad g\in\mathcal{G} \tag{9}$$

$$w^{k+1}_{1,\tau} = w^k_{\mathrm{L},\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}\setminus\{K\} \tag{10}$$

$$w^k_{\ell+1,\tau} = w^k_{\ell,\tau} + \delta_\ell z_{I(\ell)} + t_{\ell+1} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; \ell\in\mathcal{L}\setminus\{L\}\,. \tag{11}$$

$$(a_g - b_g)x^k_{g,\tau} \leq w^k_{\ell,\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{12}$$

$$(W - a_g)(x^k_{g,\tau} - 1) \leq a_g - w^k_{\ell,\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{13}$$

$$n^k_{\ell,\tau} - n^k_{\ell-1,\tau} = \sum_{g\in\mathcal{G};o_g=\ell} \rho^k_{g,\tau} - \sum_{g\in G;d_g=\ell} \rho^k_{g,\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; \ell\in\mathcal{L}\setminus\{1\} \tag{14}$$

$$n^k_{1,\tau} = \sum_{g\in\mathcal{G};o_g=1} \rho^k_{g,\tau} \qquad \tau\in\mathcal{T}; k\in\mathcal{K} \tag{15}$$

$$n^k_{\ell,\tau} \leq u_\tau \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; \ell\in\mathcal{L} \tag{16}$$

$$z_\ell \in \{0,1\} \qquad \ell = 1\dots L/2 \tag{17}$$

$$x^k_{g,\tau} \in \{0,1\} \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{18}$$

$$\rho^k_{g,\tau} \in \mathbb{Z}^+ \qquad \tau\in\mathcal{T}; k\in\mathcal{K}; g\in\mathcal{G} \tag{19}$$

$$y_\tau \in [0,1] \qquad \tau\in\mathcal{T} \tag{20}$$

$$q^k_\tau \in [0,1] \qquad \tau\in\mathcal{T}; k\in\mathcal{K} \tag{21}$$

$$n^k_{\ell,\tau} \geq 0 \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; \ell \in \mathcal{L} \tag{22}$$

$$w_\tau \geq 0 \qquad \tau \in \mathcal{T} \tag{23}$$

where

$$I(\ell) = \begin{cases} \ell & \text{if } \ell \leq \frac{L}{2} \\ L - \ell + 1 & \text{otherwise.} \end{cases}$$

In this formulation, constraints (1) and (2) force stations to be opened at the first and last (physical) locations if stations are opened at any other locations. This does assume that planners know where the system must begin and end. Since the endpoints of a rail system serve as depots to store, maintain and repair trains, we do not treat the locations of these two important facilities as separate decisions in this work. Constraints (3) and (4) ensure that a group cannot be served unless a station exists at its origin location and its destination location. Constraints (5) enforce the relationship between the q variables and the y variable for each train. Constraints (6) break symmetry by forcing trains to be used in order. Constraints (7) prohibit the assignment of groups to inactive trains. Constraints (8) establish the relationship between the x variables and the ρ variables for each group. Constraints (9) ensure that groups are served by at most one train, on exactly one of its loops. Constraints (10) and (11) enforce the train schedule based on the station configuration. Constraints (12) and (13) enforce destination arrival windows for customer assignments. The constant W, used in Constraints (13) and defined as $W = K\sum_{\ell\in\mathcal{L}}(\delta_\ell + t_\ell) + H$, is a logical upper bound for $w^k_{\ell,\tau}$. Constraints (14)-(16) enforce the capacity limitation for each train as it departs each location on each of its loops. Finally, (17)-(23) define the decision variables. Note that variables y, q, and n are continuous, but will take on integer values in any feasible solution due to the problem structure.

The w variables in the above formulation make it simple to understand and model the movement of trains in the system. Due to the network structure assumed above, however, it is possible to eliminate many of the w variables using a simple substitution. Once a train enters the system, its movement is implicitly controlled by the configuration of the stations and the time spent at each station. Therefore, a single variable w_τ, representing the time train τ enters the system, can replace $w^k_{\ell,\tau}$ using the following substitution:

$$w^k_{\ell,\tau} = w_\tau + k\sum_{\ell' \in \mathcal{L}}(\delta_{\ell'} z_{I(\ell')} + t_{\ell'}) - \sum_{\ell' \in \mathcal{L}:\ell' > \ell}(\delta_{\ell'} z_{I(\ell')} + t_{\ell'}) - \delta_\ell z_{I(\ell)}. \tag{24}$$

We will adopt notation to simplify this substitution. By letting

$$S(\ell, k) = k\sum_{\ell' \in \mathcal{L}}(\delta_{\ell'} z_{I(\ell')} + t_{\ell'}) - \sum_{\ell' \in \mathcal{L}:\ell' > \ell}(\delta_{\ell'} z_{I(\ell')} + t_{\ell'}) - \delta_\ell z_{I(\ell)}, \tag{25}$$

the substitution becomes

$$w^k_{\ell,\tau} = w_\tau + S(\ell, k). \tag{26}$$

This substitution eliminates the need for constraints (10) and (11). This more compact representation will be used in the computational testing discussed in Section 5. However, before considering computational issues, we explore an alternative approach for efficiently generating solutions to our problem in the following section.

4 Solution methodology

Despite the linear structure of the rail network considered in this work, experimentation has shown that our integrated problem requires a prohibitive amount of time and computer memory to solve using commercial optimization software. To illustrate this point, Table 3 reports *typical* computational times[2] for four instances generated randomly based on real data gathered during this project. Note that this table is strongly indicative of results seen across all experiments considered in this project. For more information on the construction of these instances, see Section 5.

As Table 3 shows, our problem is difficult to solve even for very small test instances and larger instances exhaust the available memory on our test machine very quickly. Furthermore, we have found the problem to be difficult to solve even when many of the decisions are fixed. For example, Table 4 shows the computational results for our problem when the station

[2] All experiments were performed using CPLEX 12 on an Apple® iMac ® computer with an Intel® Core 2 ™ 2.66 GHz processor and 4 GB of RAM.

Table 3: CPLEX Results

Instance	$\lvert\mathcal{G}\rvert$	L	T	H	K	Runtime (s)	Best Soln.	Best Bnd.	Opt. Gap
1	200	8	10	500	8	36,000	125	1305.05	944.04%
2	300	14	20	500	6	36,000	2484	5890.27	137.13%
3	1000	30	25	700	9	-*	-*	-*	-*
4	2500	44	40	900	10	-*	-*	-*	-*

* Indicates that the memory on our test machine was exhausted

procurement decisions and the vehicle procurement decisions have been fixed (i.e. we assume that we know which stations should be opened and which trains are utilized).

Table 4: CPLEX Results with Fixed Station Configuration and Train Procurement

Instance	Runtime (s)	Best Soln.	Best Bnd.	Opt. Gap
1	18000	239	1025.74	329.18%
2	18000	4172	5039	20.78%
3	-*	-*	-*	-*
4	-*	-*	-*	-*

* Indicates that the memory on our test machine was exhausted

Since CPLEX has such difficulty solving the train scheduling and customer assignment subproblems, it was apparent that any solution methodology for the problem should rely very little, if at all, on exact approaches to solve portions of the problem. With this in mind, and because none of the heuristic techniques present in the literature could be easily adapted to our model, we developed a customized heuristic solution methodology to solve the problem described in the previous section. An overview of this procedure is described in the remainder of this section. More information regarding the detailed mechanics of the heuristic can be found in the Appendix.

Our heuristic is neighborhood search-based, but features both segmented routines and a nested structure within each of these routines. To clarify, our heuristic follows the high-level process outlined in Figure 4, where portions of the best found solution are carried over between each segment. The routines included in Figure 4 are described as follows.

Initial Solution Construction Routine

In the Initial Construction and Improvement routine, a starting station configuration is generated by exhaustively considering all possible station configurations and choosing the

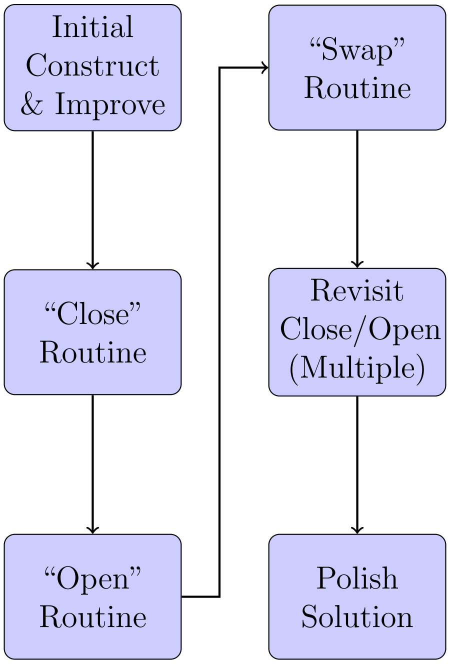


Figure 4: Heuristic Outline

station configuration that maximizes the "potential profit" associated with station costs and passenger revenue. That is, if a passenger's origin and destination locations each possess a station, then his ticket revenue will be counted towards the "potential profit." At this point, any costs associated with serving that passenger other than station construction are ignored (i.e. train procurement, operational costs). Once a configuration has been identified, a greedy method is used to construct an initial solution. In this greedy method, each train is assigned a schedule that allows it to serve the largest unserved group at the time. Other unserved groups are added if this schedule allows them to be served. After all possible groups are added, the train is checked for profitability (i.e. Does the customer revenue on that train at least cover the costs associated with the train?). If the train is profitable, it is kept, otherwise its passengers are removed and the process starts over with the next-largest unserved group as a "seed." This repeats until all trains are active and profitable or until all unserved groups have been used as a schedule "seed." This solution is used to establish a baseline fleet size. Next, a user-defined range of fleet sizes, (number of trains) around this baseline will be considered. In our implementation, a fleet size range of 5 is used. For each fleet size in this range, a solution is greedily constructed that utilizes the specified number of vehicles (i.e. vehicles are given schedules and passengers are assigned to vehicles that could feasibly serve them). Finally, we attempt to improve this constructed solution by modifying the train schedules and customer assignments through the so-called improvement routine described below.

Solution Improvement Procedure

Repeatedly during the heuristic, solutions are constructed for a particular station configuration and fleet size, and then they are passed to our Solution Improvement Procedure. In this procedure, a solution obtained via construction during one of the routines (e.g. Initial Construction, Close, Open, etc.) is taken as an input and serves as the basis for our improvement scheme. Given the current constructed solution, a user-defined number of train pairs are randomly selected, each train having an equal likelihood of being selected, for additional consideration. The number of train pairs allowed in our implementation is 200. Amongst the trains available for random selection is a dummy train to which all unserved customers are assigned. For each train pair considered, "group moves" and "group swaps" are attempted. A group move consists of moving a group served on one randomly selected train to the other randomly selected train. Similarly, a group swap consists of randomly choosing a group, where each group has an equal probability of being chosen, from each randomly chosen train and forcing each of these groups to be served by the train to which they are not currently assigned. In each of these cases, an attempt is made to serve the group or groups in question on the opposite train by allowing the number of people served within a group to be modified in order to satisfy train capacity. Next, an attempt is made to alter the train schedule so that any newly considered customers might be served. This step is done via a straightforward modification of the train arrival windows. However, it is important to note that while the train schedules can be altered to serve a new customer, no currently served customers may become unserved in this process. The sequence of group moves and swaps are repeated until a user-defined number of iterations have passed without improvement, at which point the next train pair is considered and the process is repeated. In our implementation, the limit on consecutive iterations without improvement is taken to be 25. The improvement routine stops when all randomly selected train pairs have been considered.

Close Routine

The Close routine begins with the best station configuration identified in the Initial Solution Construction. From this configuration, we explore a portion of the neighborhood of solutions defined to be those with one fewer station than the current best solution. In the first Close routine iteration, we consider all station configurations achieved by closing exactly one station that is open in the current candidate solution. For each of these new configurations, the process outlined in the previous routines is performed (Set Configuration → Construct to Find Baseline Fleet Size →Set Fleet Size → Construct Solution → Improve Solution). After this iteration, the solution with the highest objective among those considered is stored, even if it does not improve upon the overall best solution. This process is repeated with the stored solution's configuration (not necessary the configuration from the overall best solution) serving as the starting configuration for the next Close iteration. The iterations continue until no improvement has occurred for a user-defined number of Close iterations, or until all stations are closed. In our implementation the allowed number of consecutive iterations without an improvement is 2 for this routine.

Open Routine

A similar process as that found the Close routine is used to define Open routine. In this case, we partially explore the neighborhood of solutions that can be found by opening exactly one additional station in the configuration associated with the overall best solution found to this point. All configurations that feature one more open station than the current best are considered and the solution among these with the highest objective is stored as the input to the next Open iteration. This process repeats until a user-defined number of iterations have occurred with no improvement, or until all of the stations are open. In our implementation the allowed number of consecutive iterations without an improvement is 2 for this routine.

Swap Routine

In the swap routine, one currently opened station is closed, and one currently closed station is opened. We start with the configuration that produced the overall best solution up to this point. All possible pairs of stations consisting of one open and one closed station are considered and follow the same process to produce and improve solutions as detailed in the Improvement routine. After all possible pairs are considered, the best of these solutions is stored and its configuration is the starting point for the next Swap iteration. This process repeats until a user-defined number of iterations is reached, or until no improving solution is found for a defined number of consecutive iterations. In our implementation the overall limit on iterations is 15 for this routine, and we halted the routine after a single iteration with no improving solution.

Modified Close/Open Routine

In this routine, we revisit the Close and Open concept using a modified search scheme. Specifically, in an attempt to uncover complex interactions between multiple stations, we now allow up to 5 stations[3] to be opened or closed at once. Starting with the configuration from the best overall solution, we randomly decide whether to open or close stations, and how many. A user-defined number of these configurations are considered (i.e. solutions are created and improved for each configuration paired with a range of fleet sizes) and the best of these solutions is stored. Our implementation allows for 50 of these configurations to be considered. This process is repeated with the stored configuration as a starting point for the next iteration until a pre-set number of iterations are completed without improvement. For our implementation, we require improvement after at most 2 consecutive iterations in order to continue this routine.

Polishing Routine

In our final routine, we attempt to polish the best solution found using the previous sequence of routines. In this segment of the heuristic, we no longer consider configuration changes, but instead focus on the train schedules and passenger assignments. We do this by performing an extended version of the schedule improvement process outlined above and by removing passengers from trains and train loops with very light ridership. That is, if the passengers assigned to a train loop do not cover the operational cost of that loop (v_τ) or if the total ridership for a train does not cover the procurement and operational costs for that train, then the customers are removed from the loop or train, respectively. An attempt is made to serve these customers on other trains if train passenger capacity is available and train schedules allow for the customers to travel within their desired time windows. However, it is important to note that the overall system profit will increase from removing customers that

[3] The limit of 5 was put in place due to the increased computational cost incurred by considering multiple stations beyond this level.

do not generate revenue exceeding a loop cost even if these customers remain unserved.

The values we chose for each user-defined parameter stemmed from the need to solve realistically sized instances in reasonable time frames. We weighed this need against the affect these parameters had on the solution quality for smaller instances to settle on these values. Notice that each routine found in Figure 4, except the final solution polishing, follows the general steps outlined in Figure 5 in some way. The nested structure of the heuristic follows the natural hierarchy of rail planning decisions outlined above. Our heuristic, like our model, considers the interaction between these decisions in a way that is not currently present in the literature. Again, a more detailed outline of the heuristic process is given in Appendix B.

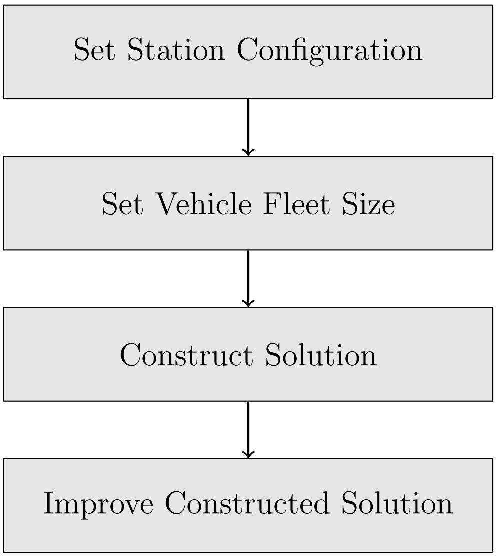


Figure 5: Nested Heuristic Structure

5 Computational study

In this section we investigate the performance of the solution methodology presented in Section 4 on a broad range of test instances. Before presenting our results, we discuss motivation and mechanics behind the scheme utilized to generate each of our random instances. Then we provide numerical results that offer insights into the capabilities of the proposed heuristic procedure.

5.1 Experimental design

Using information regarding existing rail systems and commuter patterns, we developed a procedure for randomly generating experiments intended to resemble real-world instances. The generation of our instances can be broken into: (i) determining potential station locations, (ii) assigning passenger demand time windows, (iii) identifying each potential customer's origin and destination and (iv) defining the appropriate rail system operational parameters. The following subsections describe how we handle the generation of each of these problem components.

Station Locations

The potential station locations were randomly generated in a manner consistent with the variation of available locations in Northwest Arkansas, where we hope to apply our model. That is, we assume an ordered number of possible locations along a single line. Note that the line might be representative of an existing rail bed, as is the case in Northwest Arkansas. For each consecutive pair of potential station locations, the number of miles between each location is a uniform random variable with range [0.5, β] miles, where $\beta \in [5, 15]$ miles, depending on the instance.

Passenger Time Windows

The time in which passenger demand occurs was generated in such a way that two distinct "peaks" were present in the planning horizon in order to represent "rush hour" demand caused by commuter traffic to and from work. Specifically, we generate customer demand so that 60-80% of demand occurs during two "peak" periods during the horizon. For a full-day horizon, these "peaks" may fall between 7:00-9:00 AM and 4:00-6:00 PM, for example. Note that for the instances studied in this section, the horizon length is 500 minutes, leaving the full-length horizon to be studied in Section 6. Accurately modeling peaks and valleys in customer demand throughout this horizon was accomplished by classifying our customers as one of three types: commuters, students, or others. Then, arrival deadlines were generated according to a uniform distribution bounded by the preset windows shown in Table 5 that are specific to each customer type. Recall that each customer has both an originating and return trip [4], therefore separate bounds are given for each trip. In addition,

[4] Note that we do not "link" these trips at this time.

since customers classified as "others" are assumed to travel anytime throughout the day with equal likelihood, the deadline associated with this category's outgoing trip is generated uniformly between time 0 and the end of the horizon (500 minutes). The deadline for the return trip of the "other" customers occurs with equal likelihood at anytime between the deadline of the originating trip, which we refer to as $\bar{o}$, and the end of the horizon. Passenger arrival window lengths were fixed at 15 or 30 minutes for the instances used for computational testing. For our case study, passenger arrival windows are randomly generated for each group and are equally likely to be 30 or 60 minutes. This was done simply to account for the various levels of flexibility that passengers might have.

Table 5: Customer Arrival Deadlines By Type

Customer Type/Trip Classification	Arrival Deadline Bounds (min)
Commuters Outgoing Trip	[100,175]
Commuters Return Trip	[375,450]
Students Outgoing Trip	[100,175]
Students Return Trip	[375,450]
Others Outgoing Trip	[0,500]
Others Return Trip	[$\bar{o}$,500]

Passenger Origins and Destinations

Passenger origins and destination were determined in a uniform manner (i.e. stations being equally likely to be an origin or destination) in some cases and with higher variability (i.e. stations weighted based on popularity) in others in order to model differing population and attraction distributions. To model increased variability in customer/destination location, we relied on the characterization of customer demand as being baseline, variable or more variable. In the baseline case, customers were randomly assigned an origin station[5] from the set of randomly generated potential station locations, followed by a destination randomly chosen from the locations down-track of the assigned origin station. In the variable and more variable scenarios, the likelihoods of locations being chosen as destinations were varied with increasing intensity as we moved from the variable to more variable scenarios. Specifically, in the variable instances, the overall likelihood that a station serves as a customer origin or

[5] Each origin was equally likely.

destination ranges between 7% and 23%, with the sum of the probabilities associated with each potential station is 1. For more variable instances, a wider range of 6-30% of customer demand per station is possible. Here again, the sum of the probabilities associated with each potential station must be 1. This scheme allowed us to represent the realistic situation in which one location is primarily residential and would be a likely customer origin in the morning whereas another location may be located in an industrial area where many customers work, but few live. Therefore, morning traffic would be heavy outbound from the residential area and inbound to the industrial area, while afternoon demand would follow the opposite pattern.

Rail Operations

With regard to more specific operations of the rail system, we assumed a cruising train speed of 35 miles per hour based on various existing rail systems found in the literature [1]. Furthermore, we assumed train costs between $500,000 and $2 million and 15-20 year operational lives depending on the instance. These values are consistent with the values presented in [1]. For randomly generated instances, station procurement costs were set between $2.5 million and $15 million with an assumed life of 50 years. These values are slightly lower than many station costs for existing systems [1, 2] since we assume that stations in rural settings would be cheaper due to lower construction and real estate costs.

5.2 Numerical results

In this section, we provide a comparison of the performance of the heuristic approach described in Section 4 and that of a commercial optimization solver. We also present computational results for a comprehensive set of test instances. All experiments were performed using CPLEX 12 on an Apple® iMac® computer with an Intel® Core 2™ 2.66 GHz processor and 4 GB of RAM. All instances considered in this section feature 14 locations, 20 potential trains, 8 potential loops, and were generated using the procedures and definitions presented in Section 5.1. Unless indicated otherwise, daily station costs for these instances are uniformly generated between $600 and $900 and the customer arrival window length, b, is selected to be 30 minutes. The default capacity for trains in these instances is 175 passengers unless otherwise specified in the naming convention presented in the following subsection. The revenue associated with serving a single customer, r, is \$2.75 for all of these instances.

5.2.1 Instance naming convention

To describe the characteristics of each instance tested, a four-part naming convention is adopted. The computational instances are named according to the following convention.

- *First Character of Instance Name:* All instances begin with the letter B, V, or M. These letters are used to indicate baseline, variable, or more variable customer demand variability, respectively. Each of these levels were described in Section 5.1 and reflect the likelihood that a particular station is chosen as a customer's origin/destination.
- *Second Character of Instance Name:* The second character in all instance names is either a 3 or a 5. A value of 3 (5) indicates that there were 300 (500) groups considered in that specific instance.
- *Third Character of Instance Name (optional):* In some instances, we've investigated the impact of varying certain default parameters (e.g. station cost, train capacity or passenger arrival window). For instances in which this additional consideration was made, a "special" character appears as the third character in the instance name. The optional third character may be an S, C or W, indicating a change in station cost, train capacity or passenger arrival windows, respectively. The details of the changes associated with each of these special characters is as follows:
  - S indicates that the station costs considered in that instance are more variable than those considered in a default instance. Specifically, the station costs are now generated according to the following scheme, where the percentage represents the likelihood that a single station's cost will follow the specified distribution.
    - 20% - uniform between \$300 and \$600
    - 70% - uniform between \$600 and \$900
    - 10% - uniform between \$1400 and \$2000
  - C indicates that the train capacity for that instance is reduced from the default of 175 passengers per train to 75 passengers per train.

  - W indicates that passenger arrival windows are shorter than the default of 30 minutes. For that instance, the passenger arrival windows are reduced to 15 minutes.

- *Last Characters of Instance Name:* For each combination of problem characteristics, five random instances were generated. Different instances generated using the same parameter values are differentiated by an underscore (_) followed by a replicate number.

To illustrate this naming convention, consider an instance named V5W_2. From the name, we know that the instance has variable customer demand with 500 groups and a reduced customer window length. The "_2" indicates that this is the second instance of type V5W. In the following section, we analyze the computational performance of our approach on instances identified using this naming convention.

5.2.2 Heuristic versus CPLEX

The instances presented in Table 6 were randomly generated simply to assess the value of our heuristic versus that found by CPLEX. Note that in each of these instances can be identified by the naming convention and default parameter values discussed in Section 5.2.1. In the results associated with 14 of the 15 instances shown in Table 6, the heuristic obtains a solution with an objective notably better than that obtained by CPLEX in the allotted time of 2 hours. In fact, in 14 of the 15 instances, CPLEX failed to find a solution that improved upon a plan that "does nothing" (i.e. open no stations, serve no customers). The heuristic, on the other hand, produced a profitable solution for all but 1 of the instances. The ability of our heuristic to obtain improved solutions over a commercial solver is amplified by the fact that CPLEX was given 2 hours to provide its solutions, while the heuristic, on average, required only 341 seconds. These results were very much indicative of all our attempts to use commercial optimization software to identify solutions to our rail planning problem. Consistently, the commercial optimization tool failed to obtain a solution that located any stations or served any customers. Fortunately, the heuristic approach described in Section 4 provided profitable solutions in 14 of 15 cases. However, further analysis is needed to assess the value of the proposed heuristic approach. Therefore, a broader set of tests will be considered in the next section that will assist us in understanding the effectiveness of the different phases of our heuristic in obtaining improved solutions to our problem.

Table 6: Comparison Between Heuristic and CPLEX

Instance	Heur. Runtime (s)	Final Heur. Obj.	CPLEX Runtime (s)	Best CPLEX Obj.
B3_1	70	1419.5	7200	0
B3_2	224	968.75	7200	0
B3_3	51	1259.5	7200	0
B3_4	524	455.75	7200	0
B3_5	132	1248.25	7200	0
V3_1	208	0	7200	0
V3_2	522	1198.5	7200	0
V3_3	62	1646.5	7200	0
V3_4	624	1994	7200	0
V3_5	419	2973.25	7200	0
M3_1	469	2471	7200	0
M3_2	671	2488.5	7200	154.75
M3_3	739	1393.5	7200	0
M3_4	208	2459.25	7200	0
M3_5	186	1597.25	7200	0

5.2.3 Heuristic performance

A more comprehensive set of results used to assess the performance of our heuristic is presented through the expanded set of instances shown in Tables 7 through 18 below.

To measure the effectiveness of the heuristic outlined in Section 4, four "classes" of instances have been created. Within each class, six specific instance types are considered, with five replicates for each type. Note that each of these instances complies with the naming scheme outlined earlier in this section. Tables 7 through 9 describe the heuristic results for the first class of instances, which we refer to as the base experiments. Tables 10 through 12 describe the results for the second class, which we call the capacity experiments. Tables 13 thought 15 describe the results for the third class, the station cost experiments. Finally, Tables 16 through 18 describe the fourth class of instances, the passenger arrival window experiments. Recall from Section 4 that our heuristic proceeds through three phases: (i) Construction; (ii) Improvement and (iii) Polishing. The first table for each class serves to evaluate the impact of these three phases on the objective. The second table for each class breaks down the computational time according to the three phases. Finally, solution statistics are presented for the instances in each class. The results of these experiments are summarized in Tables 19 and 20 below.

The percentage shown in the Improvement column is calculated as follows:

$$\frac{z^{Final} - z^{Construction}}{z^{Construction}}.$$

Improvement results marked by a '-' indicate that the original construction solution was no better than the "do nothing" option. Therefore, while the final polished objective has improved, the percent of improvement cannot be compared with the results in which the initial construction heuristic had a positive objective. Note that some of our heuristic solutions have a negative objective. In these cases, the "do nothing" option is clearly preferable, but our heuristic did not reach this solution. The "Groups Served" statistic presented for each instance includes any group that was partially or fully served in the final solution.

Table 7: Heuristic Results: Base Experiments

Instance	Runtime (s)	Init. Constr. Obj.	Obj. After Impr.	Obj. After Polishing	Improvement
B3_1	70	212.5	1419.5	1419.5	568%
B3_2	224	-642.5	924.75	968.75	-
B3_3	51	33	1259.5	1259.5	3717%
B3_4	524	-373	455.75	455.75	-
B3_5	132	592.25	1248.25	1248.25	111%
V3_1	208	-594.25	0	0	-
V3_2	522	677	1198.5	1198.5	77%
V3_3	62	653.5	1624.5	1646.5	152%
V3_4	624	398.5	1906	1994	400%
V3_5	419	649.25	2970.5	2973.25	358%
M3_1	469	132	2471	2471	1772%
M3_2	671	1213	2488.5	2488.5	105%
M3_3	739	462.25	1382.5	1393.5	201%
M3_4	208	1351.25	2459.25	2459.25	82%
M3_5	186	19.5	1597.25	1597.25	8091%
B5_1	702	4919.5	5880.5	5883.25	20%
B5_2	303	4592.75	6073	6103.25	33%
B5_3	44	5407.5	5795.25	5795.25	7%
B5_4	48	4652.25	6331.5	6331.5	36%
B5_5	66	4517.25	5396	5530.75	22%
V5_1	1021	3605	4912.5	5176.5	44%
V5_2	565	2735.75	4518.5	4716.5	72%
V5_3	349	2268	3883	4023.25	77%
V5_4	515	3591.5	4456	4662.25	30%
V5_5	85	4071.5	4666.75	4886.75	20%
M5_1	1474	4345.25	6211.5	6379.25	47%
M5_2	315	4678.25	5814	6009.25	28%
M5_3	228	4043.75	6627.25	6764.75	67%
M5_4	1542	4201.25	5760.75	5837.75	39%
M5_5	384	2323.25	4197.75	4951.25	113%

Table 8: Heuristic Results: Base Experiment Timing

Instance	Init. Constr. Time (s)	Improvement Time (s)	Polishing Time (s)
B3_1	<1	67	3
B3_2	<1	221	3
B3_3	<1	48	3
B3_4	<1	520	3
B3_5	<1	129	3
V3_1	<1	208	<1
V3_2	<1	519	3
V3_3	<1	59	3
V3_4	<1	620	4
V3_5	<1	415	4
M3_1	<1	466	3
M3_2	<1	668	3
M3_3	<1	735	4
M3_4	<1	206	2
M3_5	<1	183	3
B5_1	<1	697	5
B5_2	<1	297	5
B5_3	<1	40	4
B5_4	<1	44	5
B5_5	<1	59	7
V5_1	<1	1013	8
V5_2	<1	555	10
V5_3	<1	341	8
V5_4	<1	506	9
V5_5	<1	80	5
M5_1	<1	1466	8
M5_2	<1	308	7
M5_3	<1	221	7
M5_4	<1	1537	5
M5_5	<1	373	11

Table 9: Heuristic Results: Base Experiment Solutions

Instance	Stations Open	Fleet Size	Groups Served	Passengers Served
B3_1	7	2	277	2978/3171
B3_2	7	4	293	3037/3141
B3_3	7	3	295	3142/3202
B3_4	7	4	277	2825/3079
B3_5	7	3	295	3147/3208
V3_1	0	0	0	0/3063
V3_2	7	3	292	3042/3164
V3_3	7	3	283	2950/3173
V3_4	5	4	273	2988/3285
V3_5	5	4	286	3123/3278
M3_1	6	3	276	2844/3131
M3_2	4	3	229	2314/3064
M3_3	6	4	252	2686/3168
M3_4	7	2	270	2867/3215
M3_5	5	3	238	2415/3173
B5_1	7	4	470	4951/5307
B5_2	7	4	465	4923/5339
B5_3	7	3	463	4779/5133
B5_4	7	3	479	4846/5117
B5_5	7	5	473	4825/5167
V5_1	7	6	479	4966/5232
V5_2	7	8	460	4874/5328
V5_3	7	6	438	4627/5373
V5_4	7	7	470	4803/5184
V5_5	7	4	477	4701/4981
M5_1	7	6	439	4711/5434
M5_2	7	5	448	4511/5096
M5_3	7	5	450	4681/5391
M5_4	6	3	404	4041/5024
M5_5	7	8	456	4935/5449

Table 10: Heuristic Results: Capacity Experiments

Instance	Runtime (s)	Init. Constr. Obj.	Obj. After Impr.	Obj. After Polishing	Improvement
B3C_1	178	-729	0	0	-
B3C_2	271	-307.25	573	625.25	-
B3C_3	165	-934	0	0	-
B3C_4	654	140.75	531.5	561.75	299%
B3C_5	527	-932.75	-252	-199.75	-
V3C_1	182	-2275.5	0	0	-
V3C_2	1197	-1753.25	519.25	651.25	-
V3C_3	474	-2229.75	0	0	-
V3C_4	400	-1833.75	0	0	-
V3C_5	369	-1586.75	0	0	-
M3C_1	643	-488.25	458	458	-
M3C_2	858	-1194.75	1178.5	1305	-
M3C_3	773	-1766	671.25	943.5	-
M3C_4	996	-257	1281.75	1406.75	-
M3C_5	521	437.5	1546.5	1574	260%
B5C_1	206	2686.75	3289.25	3613.75	35%
B5C_2	624	1845.75	2863	3209.5	74%
B5C_3	296	2185.75	2405.25	2471.25	13%
B5C_4	959	2334.25	3017.5	3320	42%
B5C_5	1158	2038.75	3709.25	3940.25	93%
V5C_1	1624	464.25	1160.5	1193.5	157%
V5C_2	1432	-469	937.5	1223.5	-
V5C_3	580	949.25	1900.5	2932.75	209%
V5C_4	371	312.25	1700	1782.5	471%
V5C_5	1428	250.25	1195.25	1255.75	402%
M5C_1	1301	2371.5	3288	4448.5	88%
M5C_2	742	2014.75	3508.25	3593.5	78%
M5C_3	1505	1854	3277.5	3566.25	92%
M5C_4	1482	1537	1967.25	2272.5	48%
M5C_5	1796	1262.5	2242	2352	86%

Table 11: Heuristic Results: Capacity Experiment Timing

Instance	Init. Constr. Time (s)	Improvement Time (s)	Polishing Time (s)
B3C_1	<1	178	<1
B3C_2	<1	268	3
B3C_3	<1	165	<1
B3C_4	<1	651	3
B3C_5	<1	525	3
V3C_1	<1	182	<1
V3C_2	<1	1192	4
V3C_3	<1	474	<1
V3C_4	<1	400	<1
V3C_5	<1	369	<1
M3C_1	<1	640	3
M3C_2	<1	854	4
M3C_3	<1	768	5
M3C_4	<1	993	4
M3C_5	<1	517	4
B5C_1	<1	199	7
B5C_2	<1	615	9
B5C_3	<1	289	7
B5C_4	<1	950	9
B5C_5	<1	1151	7
V5C_1	<1	1615	8
V5C_2	<1	1424	8
V5C_3	<1	560	20
V5C_4	<1	363	8
V5C_5	<1	1421	7
M5C_1	<1	1290	11
M5C_2	<1	735	7
M5C_3	<1	1496	9
M5C_4	<1	1472	10
M5C_5	<1	1786	10

Table 12: Heuristic Results: Capacity Experiment Solutions

Instance	Stations Open	Fleet Size	Groups Served	Passengers Served
B3C_1	0	0	0	0/3266
B3C_2	7	3	268	2763/3260
B3C_3	0	0	0	0/2993
B3C_4	7	4	261	2917/3390
B3C_5	7	3	252	2511/3083
V3C_1	0	0	0	0/3000
V3C_2	5	5	254	2591/3186
V3C_3	0	0	0	0/3163
V3C_4	0	0	0	0/3017
V3C_5	0	0	0	0/3105
M3C_1	4	3	179	1720/2982
M3C_2	4	4	216	2144/3140
M3C_3	5	5	243	2298/2981
M3C_4	4	4	228	2265/3104
M3C_5	7	4	239	2528/3214
B5C_1	7	5	433	4417/5213
B5C_2	7	6	412	4230/5878
B5C_3	7	5	379	3671/5118
B5C_4	7	6	417	4256/5236
B5C_5	7	5	435	4355/5191
V5C_1	6	5	313	3074/5336
V5C_2	6	5	321	3218/5198
V5C_3	7	8	446	4533/5390
V5C_4	6	5	307	3254/5377
V5C_5	6	4	305	3097/5296
M5C_1	6	5	358	3526/5246
M5C_2	5	4	316	3042/5185
M5C_3	6	5	350	3643/5200
M5C_4	7	5	306	3114/5450
M5C_5	5	5	276	2952/5454

Table 13: Heuristic Results: Station Cost Experiments

Instance	Runtime (s)	Init. Constr. Obj.	Obj. After Impr.	Obj. After Polishing	Improvement
B3S_1	56	-144.75	680.5	680.5	-
B3S_2	57	-438.25	835.25	835.25	-
B3S_3	88	126.5	435.75	457.75	262%
B3S_4	488	-775.5	-28	24.25	-
B3S_5	415	439.5	600.5	600.5	37%
V3S_1	138	1271	1650.5	1716.5	35%
V3S_2	500	519.75	1729.5	1773.5	241%
V3S_3	128	-923.75	0	0	-
V3S_4	320	87.5	1504.25	1504.25	1619%
V3S_5	462	-1559.75	1059	1114	-
M3S_1	591	-382	1491.5	1552	-
M3S_2	593	1188.25	1980.5	1994.25	68%
M3S_3	701	150	2121.75	2121.75	1315%
M3S_4	198	-1169.75	0	0	-
M3S_5	563	-696.25	554.5	557.25	-
B5S_1	73	5902.25	6610.25	6662.5	13%
B5S_2	55	4524	5918.75	5951.75	32%
B5S_3	627	4029.5	5143.5	5154.5	28%
B5S_4	40	6174.5	6305.25	6341	3%
B5S_5	115	5751.25	7052.25	7121	24%
V5S_1	370	3265.75	4577.75	4668.5	43%
V5S_2	1030	3787.75	4937.752	4976.25	31%
V5S_3	818	396	2717	2934.25	641%
V5S_4	2697	733	2748.25	2778.5	279%
V5S_5	781	3382.25	3885.75	4345	28%
M5S_1	757	2820	4739.75	4743.75	68%
M5S_2	705	1933.25	5486.75	5530.75	186%
M5S_3	906	3555.25	4611	4723.75	33%
M5S_4	766	2374.5	3383.25	3842.5	62%
M5S_5	841	5111.5	6207.25	6273.25	23%

Table 14: Heuristic Results: Station Cost Experiment Timing

Instance	Init. Constr. Time (s)	Improvement Time (s)	Polishing Time (s)
B3S_1	<1	53	3
B3S_2	<1	54	3
B3S_3	<1	86	2
B3S_4	<1	484	4
B3S_5	<1	411	4
V3S_1	<1	135	3
V3S_2	<1	496	4
V3S_3	<1	128	<1
V3S_4	<1	317	3
V3S_5	<1	458	4
M3S_1	<1	587	4
M3S_2	<1	589	4
M3S_3	<1	698	3
M3S_4	<1	198	<1
M3S_5	<1	561	2
B5S_1	<1	67	6
B5S_2	<1	49	6
B5S_3	<1	621	6
B5S_4	<1	36	4
B5S_5	<1	108	7
V5S_1	<1	361	9
V5S_2	<1	1024	6
V5S_3	<1	813	5
V5S_4	<1	2689	8
V5S_5	<1	774	7
M5S_1	<1	751	6
M5S_2	<1	700	5
M5S_3	<1	899	7
M5S_4	<1	757	9
M5S_5	<1	837	4

Table 15: Heuristic Results: Station Cost Experiment Solutions

Instance	Stations Open	Fleet Size	Groups Served	Passengers Served
B3S_1	7	3	293	2990/3067
B3S_2	7	3	289	3055/3148
B3S_3	7	2	282	2969/3154
B3S_4	7	4	269	2827/3135
B3S_5	7	4	292	3050/3179
V3S_1	7	3	284	3066/3280
V3S_2	6	4	285	2942/3128
V3S_3	0	0	0	0/3193
V3S_4	6	3	251	2463/3005
V3S_5	5	4	241	2436/3115
M3S_1	5	4	241	2600/3197
M3S_2	7	4	277	2847/3186
M3S_3	5	4	285	3021/3218
M3S_4	0	0	0	0/3176
M3S_5	6	2	265	2703/3083
B5S_1	7	4	475	4978/5298
B5S_2	7	4	480	4889/5132
B5S_3	7	4	472	4970/5246
B5S_4	7	3	487	5044/5202
B5S_5	7	4	485	5036/5224
V5S_1	7	6	449	4678/5327
V5S_2	6	3	389	4131/5308
V5S_3	6	3	397	4095/5238
V5S_4	6	5	389	4002/5255
V5S_5	7	5	449	4592/5224
M5S_1	6	4	413	4225/5121
M5S_2	5	3	427	4517/5338
M5S_3	7	5	459	4645/5166
M5S_4	7	7	443	4474/5065
M5S_5	7	3	457	4711/5276

Table 16: Heuristic Results: Arrival Window Experiments

Instance	Runtime (s)	Init. Constr. Obj.	Obj. After Impr.	Obj. After Polishing	Improvement
B3W_1	1280	-1131	263	351	-
B3W_2	241	-1157	0	0	-
B3W_3	201	288	288	288	0%
B3W_4	231	-2307.75	0	0	-
B3W_5	292	-917.75	25.25	63.75	-
V3W_1	612	-2782.5	0	0	-
V3W_2	1078	-1815.25	-533.25	-489.5	-
V3W_3	293	-1502.5	0	0	-
V3W_4	987	-1210	451.25	473.25	-
V3W_5	459	-1332.5	0	0	-
M3W_1	655	-1530.5	205.5	304.5	-
M3W_2	807	120	1539.25	1544.75	1187%
M3W_3	872	872.25	2181.5	2181.5	150%
M3W_4	119	1087	1999.5	1999.5	84%
M3W_5	838	217.75	994	1112.25	411%
B5W_1	174	2725.25	3719.75	3719.75	36%
B5W_2	928	4136.75	4808	4843.75	17%
B5W_3	59	4316	6330.25	6357.75	47%
B5W_4	76	3709.25	4978.75	4978.75	34%
B5W_5	925	3282	4738.5	4738.5	44%
V5W_1	263	4950.5	5616	5657.25	14%
V5W_2	292	3233	4496.75	4555.75	41%
V5W_3	610	2320.25	3877	3910	69%
V5W_4	143	3518.75	4906.25	5038.25	43%
V5W_5	631	1763.25	2870	2937.25	67%
M5W_1	1043	3424	5385	5420.75	58%
M5W_2	496	3043.5	4684.25	4684.25	54%
M5W_3	1120	5063	6074.25	6115.5	21%
M5W_4	244	4129.75	5622.25	5664.75	37%
M5W_5	383	4572	6211.25	6302	38%

Table 17: Heuristic Results: Arrival Window Experiment Timing

Instance	Init. Constr. Time (s)	Improvement Time (s)	Polishing Time (s)
B3W_1	<1	1275	5
B3W_2	<1	241	<1
B3W_3	<1	198	3
B3W_4	<1	231	<1
B3W_5	<1	289	3
V3W_1	<1	612	<1
V3W_2	<1	1072	6
V3W_3	<1	293	<1
V3W_4	<1	982	5
V3W_5	<1	459	<1
M3W_1	<1	649	6
M3W_2	<1	802	5
M3W_3	<1	868	4
M3W_4	<1	115	4
M3W_5	<1	833	5
B5W_1	<1	162	12
B5W_2	<1	919	9
B5W_3	<1	54	5
B5W_4	<1	69	7
B5W_5	<1	917	8
V5W_1	<1	256	7
V5W_2	<1	283	9
V5W_3	<1	600	10
V5W_4	<1	135	8
V5W_5	1	611	20
M5W_1	<1	1037	6
M5W_2	1	484	12
M5W_3	<1	1114	6
M5W_4	<1	228	16
M5W_5	<1	374	9

Table 18: Heuristic Results: Arrival Window Experiment Solutions

Instance	Stations Open	Fleet Size	Groups Served	Passengers Served
B3W_1	7	6	290	3160/3284
B3W_2	0	0	0	0/3167
B3W_3	7	4	283	2984/3168
B3W_4	0	0	0	0/2997
B3W_5	7	4	279	2949/3169
V3W_1	0	0	0	0/2986
V3W_2	6	7	283	2962/3105
V3W_3	0	0	0	0/3117
V3W_4	6	6	273	2947/3169
V3W_5	0	0	0	0/3169
M3W_1	6	7	280	2934/3082
M3W_2	6	6	290	2785/2896
M3W_3	6	5	288	3086/3237
M3W_4	7	5	295	3106/3152
M3W_5	7	6	293	3063/3153
B5W_1	7	9	457	4905/5358
B5W_2	7	7	460	4921/5319
B5W_3	7	4	492	5257/5356
B5W_4	7	6	485	5029/5200
B5W_5	7	6	472	4978/5280
V5W_1	7	5	474	5163/5445
V5W_2	7	6	473	4841/5198
V5W_3	7	8	462	4868/5270
V5W_4	7	6	482	5103/5322
V5W_5	7	9	464	4775/5237
M5W_1	7	5	465	4645/5054
M5W_2	7	9	431	4607/5400
M5W_3	7	5	455	4682/5224
M5W_4	7	7	455	5637/5172
M5W_5	7	7	467	5024/5426

Profitable solutions were found for 29 out of the 30 base experiment instances. The average runtime for these instances was 425 seconds. For instances with an improvement percentage given, the average improvement from the initial constructed solution to the final solution was 603%. For the base experiments, an average of 412 seconds were spent improving upon the constructed solution. The average time spent polishing the solution was about 5 seconds. For the base experiments, 22 solutions featured 7 open stations, 3 had 6 stations, 3 had 5 stations, 1 had 4 stations, and 1 was a "do nothing" solution. The average fleet size for these instances was 4.2 trains, when customers were served.

Profitable solutions were found for 23 out of the 30 capacity experiment instances. The average runtime for these instances was 790 seconds. For instances with an improvement percentage given, the average improvement from the initial constructed solution to the final solution was 153%. For the capacity experiments, an average of 784 seconds were spent improving upon the constructed solution. The average time spent polishing the solution was 5.7 seconds. For the capacity experiments, 11 solutions featured 7 open stations, 6 had 6 stations, 4 had 5 stations, 3 had 4 stations, and 6 were "do nothing" solutions [6]. The average fleet size for these instances was 4.7 trains, when customers were served.

Profitable solutions were found for 28 out of the 30 station cost experiment instances. The average runtime for these instances was 529 seconds. For instances with an improvement percentage given, the average improvement from the initial constructed solution to the final solution was 231%. For the station cost experiments, an average of 524 seconds were spent improving upon the constructed solution. The average time spent polishing the solution was 4.6 seconds. For the station cost experiments, 17 solutions featured 7 open stations, 7 had 6 stations, 4 had 5 stations, and 2 were "do nothing" solutions. The average fleet size for these instances was 3.8 trains, when customers were served.

Profitable solutions were found for 24 out of the 30 passenger arrival window experiment instances. The average runtime for these instances was 545 seconds. For instances with an improvement percentage given, the average improvement from the initial constructed solution to the final solution was 131%. For the passenger arrival window experiments, an average

[6] Note that one of the solutions with 7 open stations had a negative final objective, meaning a "do nothing" solution would actually be preferable to the solution found by the heuristic.

of 538 seconds were spent improving upon the constructed solution. The average time spent polishing the solution was 6.3 seconds. For the passenger arrival window experiments, 20 solutions featured 7 open stations, 5 had 6 stations, and 5 were "do nothing" solutions [7]. The average fleet size for these instances was 6.2 trains, when customers were served.

It is clear from Tables 7, 10, 13, and 16 that the improvement and polishing phases have a significant impact on solution quality. On average, the percent improvement of the final objective over the initially constructed objective is over 300%. This average does not include instances for which the initial constructed solution was no better than the "do nothing" case. The heuristic achieved significant absolute improvement for many of these instances as well. All of these improved solutions were obtained in less than 45 minutes, with the average time required being 572 seconds.

Tables 8, 11, 14, and 17 show the breakdown of time spent in each phase of the heuristic for each instance. The time required to construct the initial solution was less than one second for almost every instance. The bulk of the time was spend improving that solution, with an average of 566 seconds spent per instance. The polishing routine only required about 5.5 seconds per instance, on average.

Tables 9, 12, 15, and 18 provide further insight into the actual characteristics of the final rail system plan obtained for each instance considered. We can see that in over half of these instances, all seven stations were chosen in the final solution. In about 12% of instances, the heuristic identified the "do nothing" case as the best option. Interestingly, if stations were opened, at least four stations were selected and a high percentage of customers were served. Specifically, for instances in which at least one station was opened, 86% of total customer demand is satisfied, on average. To serve these customers, only a relatively small fleet of trains was required as only 4.7 of the 20 potential trains were actually utilized in the final solutions, on average.

[7] Note that one of the solutions with 6 open stations had a negative final objective, meaning a "do nothing" solution would actually be preferable to the solution found by the heuristic.

Table 19: Heuristic Performance Summary

Instance Grp.	Heur. Runtime (s)	Init. Const. Time (s)	Impr. Time (s)	Polish Time (s)	Impr.	% Profitable
Overall	572	<1	566	5.5	308%	86.7%
Base Exp.	425	<1	412	5	603%	96.7%
Cap. Exp.	790	<1	784	5.7	153%	76.7%
Cost Exp.	529	<1	524	4.6	231%	93.3%
Win Exp.	545	<1	538	6.3	131%	80.0%
Baseline	341	<1	336	4.7	209%	90.0%
Variable	626	<1	620	5.6	216%	77.5%
More Variable	749	<1	742	6.2	469%	97.5%
300 Groups	460	<1	457	2.8	864%	73.3%
500 Groups	684	<1	675	8.2	94.4%	100.0%

Table 20: Heuristic Solutions Summary

Instance Grp.	Stations open	Active trains	% Served
Overall	6.6	4.7	86%
Base Exp.	6.6	4.2	91%
Cap. Exp.	6.0	4.7	71%
Cost Exp.	6.5	3.8	89%
Win Exp.	6.8	6.2	95%
Baseline	7	4.4	92%
Variable	6.4	5.2	85%
More Variable	6.1	4.7	83%
300 Groups	6.2	3.9	88%
500 Groups	6.7	5.3	85%

Tables 19 and 20 summarize the heuristic performance and the solution characteristics for the instances in which passengers were served. The greatest improvement over the initial solution occurred for the instances with default parameter values. Note that as the passenger demand variability increased, the heuristic runtime also increased, as did the improvement over the initial solution. The percentage of passengers served decreased as the variability increased. As expected, the heuristic took longer to find solutions for instances with 500 groups than it did for instances with 300 groups. Interestingly, the heuristic improved upon its initial solution more for instances with 300 groups than it did for instances with 500 groups. Also, more stations were open, on average, for instances with 500 groups than for instances with 300 groups.

As evidenced in these experiments, many of the problem parameters affect the heuristic performance and the solutions it produces. For example, when the train capacity or the passenger arrival window length are reduced, the heuristic has a more difficult time finding profitable solutions. Also, though the heuristic was able to produce profitable final solutions for most of the station cost experiment instances, the initial constructed objective for those instances was unprofitable in many cases. The station cost experiments indicate that the station cost structure has only a slight impact on the number of stations open in the final solution. Specifically, the average number of open stations for the base experiments was 6.5, when any customers are served, and the average number of open stations for the station cost experiments was 6.4, when customers were served. Perhaps the most striking result shown in these experiments is that the passenger arrival window strongly impacts the final fleet size. Our results indicate that systems with shorter passenger arrival windows require more trains to serve customers. The relationships outlined above provide valuable insight for planning purposes, since they allow rail system planners to better understand the implications of their assumptions and input data.

Unfortunately, we are unable to comment on the performance of our heuristic in relation to the optimal solution for any non-trivial instances since we have been unable to prove optimality for any such instance to date.

6 Case study: Northwest Arkansas

The University of Arkansas Community Design Center (CDC) publication, *Visioning Rail Transit in Northwest Arkansas* [8] is credited with the original inspiration for this work. In the study, city planners and University of Arkansas students made a case for rail transit in the Northwest Arkansas (NWA) region, which includes communities such as Fayetteville, Lowell, Springdale, Rogers, and Bentonville. The arguments presented in the report for developing a rail system in NWA were primarily qualitative in nature. Analytical planning tools were not used to select the station locations that were proposed, illustrated in Figure 6.

Northwest Arkansas is home to multiple post-secondary schools, including the University of Arkansas, attended by over 20,000 graduate and undergraduate students. Numerous companies also have their headquarters or regional offices in NWA, including Walmart Stores Inc., J.B. Hunt Transport, Inc. and Tyson Foods, Inc. Due to heavy inter- and intra-city commuting to these and other destinations, traffic congestion is an ever-present issue in the region. Rail transportation is one option available to city planners hoping to reduce congestion.

Using actual NWA commuter, geographical, and real estate data, along with information gleaned from existing rail systems in the U.S., we constructed an instance for the rail planning model that is representative of the NWA region. A formal feasibility study would require resources unavailable to us, but the instance we have generated mirrors the typical size and

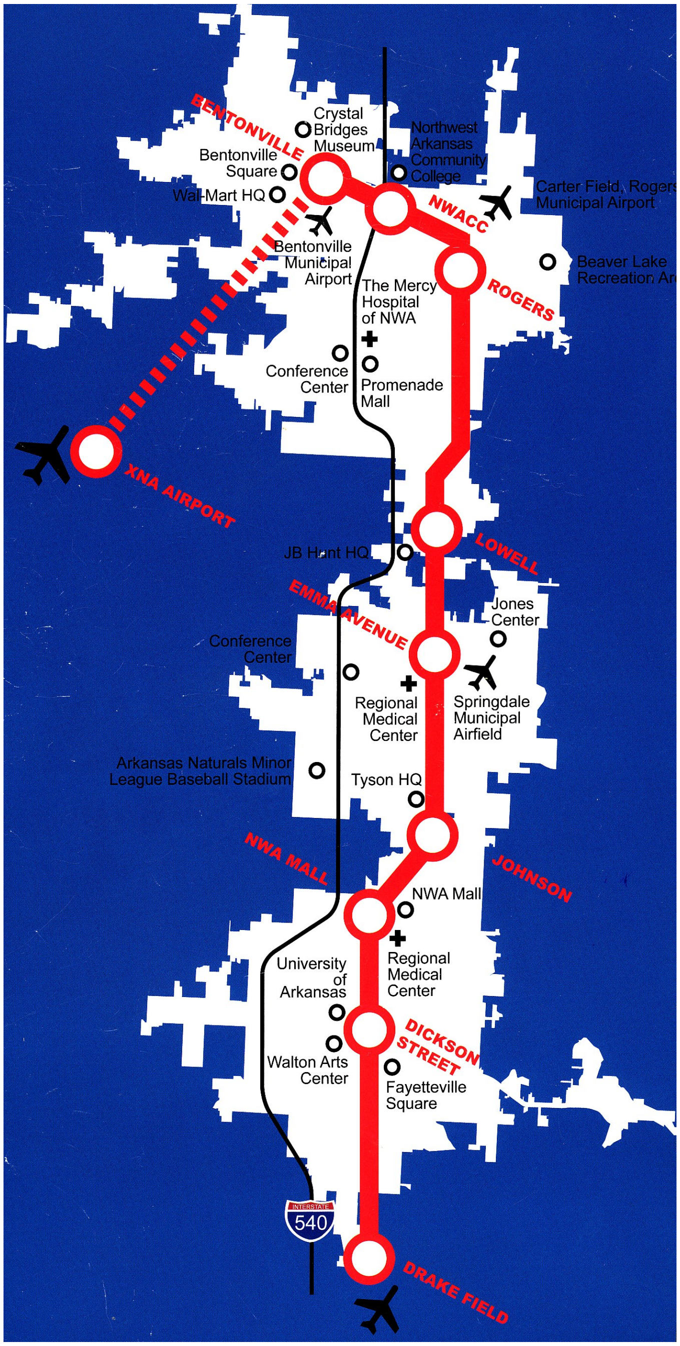


Figure 6: Map of NWA with CDC's Proposed System
Source: *Visioning Rail Transit in Northwest Arkansas* [8]

structure of any rail system that might be proposed in the area. In creating the instance, we explicitly accounted for different classes of passengers: (i) commuters, (ii) university passengers (e.g. students and teachers) and (iii) others (e.g. citizens going shopping or to visit others in the community). Instance details are given in Table 22 and described in the remainder of this section. Importantly, we show that the heuristic outlined in Section 4 is applicable to instances representative of growing rural areas and produces solutions in an acceptable amount of time, considering the integrated nature of our planning model. This is important because the NWA instance exhausts computer memory almost immediately when commercial optimization software attempts to solve it.

A list of potential station locations were generated for the NWA instance utilizing our knowledge of the region, locations of large employers, residential areas, busy highways, schools, and other attractions. These locations, which all lie along an existing rail right-of-way, are listed in Table 21. The costs of building stations at these locations were estimated using publicly available assessed property values [3, 7] and costs for existing rail systems [1, 2]. We assume station costs in NWA would be lower than those in more urban areas due to the availability of lower cost land and labor, and our estimates reflect this assumption. Train procurement costs, operational costs, speed, and capacity were estimated using information available from existing systems [1]. We assume the time required for a train to stop at a station to unload and load passengers is 2 minutes and is independent of the station and number of passengers. A revenue per customer of \$2.50 is assumed for all customers served regardless of their trip length. Trains are assumed to operate for 15 hours (900 minutes) per day, beginning at 6:00 am. To estimate zip code-to-zip code commuter volume, we used data from [4] compiled by [20] along with data from [27]. We assumed a 10% adoption level for most commuter lanes. This was reduced to 5% for passengers commuting between adjacent zip codes, and also for park-and-ride commuters. Adoption level was reduced to 1% for travel within a single zip code. Using data provided by the University of Arkansas and The Northwest Arkansas Community College (NWACC) outlining student commuter numbers by zip code, we used similar methods to estimate student demand. For each potential customer group, an original trip (usually morning) and a return trip were generated. If multiple potential stations existed in a single zip code, we assumed customers were equally likely to originate from any of these stations with two exceptions:

1. The station nearest a highway was selected as the origin for all park-and-ride customers.
2. Locations identified as light origins (e.g., Fayetteville Drake Field) were half as likely to serve as a customer origin than other locations in the same zip code.

When multiple potential stations existed within a zip code, destination selection was weighted by the size of employers located near the potential stations. For example, between the two stations in Bentonville, commuters were twice as likely to be destined for the Walmart HQ location over the alternate Bentonville location because Walmart employs a large percentage of Bentonville workers. Destinations were limited to school locations for students, and were randomly generated for other customers, though some locations were designated as attraction locations and more likely to be selected. Customer arrival deadlines were assigned uniformly within the ranges given in Table 23, and arrival window lengths were equally likely to be 30 or 60 minutes.

Table 21: NWA Instance Proposed Stations and Distances

ℓ	Station Name	$\ell - 1$ to ℓ (miles)	Zip
1	Fay - Drake Field	0.0	72701
2	Fay - 15th St	3.1	72701
3	Fay - MLKJR Blvd	0.7	72701
4	Fay - Dickson St	1.0	72701
5	Fay - Maple Ave	0.4	72701
6	Fay - Cleveland St	0.4	72701
7	Fay - North St	0.4	72703
8	Fay - Sycamore St	0.7	72703
9	Fay - Township St	0.9	72703
10	Fay - Drake St	0.6	72703
11	Fay - Wash. Reg. Med. Cntr.	0.5	72703
12	Fay - Joyce St	1.1	72704
13	Johnson	0.8	72704
14	Spring - Tyson	1.8	72762
15	Spring - Robinson Ave	1.5	72764
16	Spring - Sunset Ave	0.6	72764
17	Spring - Emma Ave	0.7	72764
18	Lowell	5.0	72745
19	Rogers - New Hope Rd	3.5	72758
20	Rogers - Walnut St	2.0	72756
21	Benton - NWACC	4.2	72712
22	Benton - Walmart HQ	2.6	72712

Table 22: NWA Instance Overview

Groups	2830
Total Potential Passengers	14368
Locations	44
Potential Trains	40
Horizon (min)	900
K	10
System Length (miles)	32.5

Table 23: NWA Parameter Details

Parameter	Values
a_g	60-180 (morning commuters) 660-780 (evening commuters) 120-360 (morning students) 420-680 (evening students) 0-900 others
b_g	30/60
c_τ	\$290
f_ℓ	\$323-1933 (based on land price and station type)
H	900
P_g	1-9
r_g	\$2.50
t_ℓ	0.7-8.6 (based on geography)
u_τ	150
v_τ	\$100

Table 24: NWA Instance Computational Results

Runtime (s)	32334
Initial Constructed Objective	-434.5
Pre-Polish Objective	2987.5
Final Objective	3645

Table 24 outlines the results of our computational testing for the NWA instance. It demonstrates that our heuristic improved upon its initial constructed solution dramatically. Importantly, given that the objective of our problem is to maximize profit, it is interesting to learn that results *from this experiment* suggest that a profitable rail system may be attainable. Table 25 provides some interesting details regarding the characteristics of a possible NWA rail system. For the instance considered, the system would use 20 of the 22 possible station locations, omitting only the locations at Martin Luther King Blvd and Drake St in Fayetteville, AR. While we allowed for up to 40 trains to be included in the system, our solution suggests that only 7 trains are needed to attain a daily profit of \$3,645. Finally, it is interesting that almost 67% of potential passenger demand would be satisfied using the plan produced by the heuristic.

Table 25: NWA Instance Final Solution Statistics

Stations Open	20/22
Fleet Size	7/40
Groups Served	1983/2830
Passengers Served	9620/14368

Table 26: NWA Instance Heuristic Solution Station Configuration

ℓ	Station Name	Status
1	Fay - Drake Field	Open
2	Fay - 15th St	Open
3	Fay - MLKJR Blvd	Closed
4	Fay - Dickson St	Open
5	Fay - Maple Ave	Open
6	Fay - Cleveland St	Open
7	Fay - North St	Open
8	Fay - Sycamore St	Open
9	Fay - Township St	Open
10	Fay - Drake St	Closed
11	Fay - Wash. Reg. Med. Cntr.	Open
12	Fay - Joyce St	Open
13	Johnson	Open
14	Spring - Tyson	Open
15	Spring - Robinson Ave	Open
16	Spring - Sunset Ave	Open
17	Spring - Emma Ave	Open
18	Lowell	Open
19	Rogers - New Hope Rd	Open
20	Rogers - Walnut St	Open
21	Benton - NWACC	Open
22	Benton - Walmart HQ	Open

The number of trains in the system throughout the planning horizon is plotted in Figure 7. As shown in the plot, the number of active trains increases steadily with demand in the beginning of the horizon and remains at 7 trains until after the second peak in demand, when trains start to exit the system. In our final solution, an average of two trains per hour stop at each station during the horizon. Despite the fact that all 7 active trains reached their ridership capacity at some point in the horizon, only 27% of total available train capacity was utilized throughout the horizon.

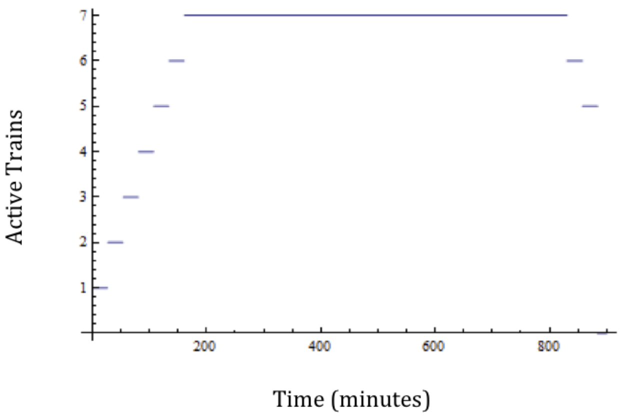


Figure 7: Active Trains Throughout the Horizon

As a comparison, our NWA solution includes 20 stations, while the solution proposed by the CDC features only 9 stations. The CDC also recommends a 10th passenger location to be connected by shuttle, however. All 9 stations proposed by the CDC were included in our final NWA solution. The CDC did not propose a specific fleet size, nor did they attempt to determine train schedules.

The strength of conclusions drawn from this experiment are limited by the strength of data available. Clearly, the approach proposed in this paper is most useful when reliable data obtained from a formal feasibility study is used to populate the model. Based on our investigation, there is evidence to suggest that a rail system is worth investigating further. In a future study conducted in partnership with city officials and financial experts, the heuristic described in this paper can effectively evaluate the feasibility of a NWA rail system.

7 Future work

Our efforts revealed a number of future research directions stemming from this work. First, we realize the importance of allowing trains to enter and leave the system to accommodate peaks in demand. This reduces costs by cutting out operational costs for some trains during periods with low demand. The model that we presented in this work does allow trains to "sit out" loops as a cost-saving measure, but the times that trains can possibly re-enter the system after leaving are mandated by the rigid train movement constraints. One resolution would be the development of an alternative model that requires a new non-negative continuous variable, Δ_τ^k, representing the delay time that train τ spends at the depot between loops $k-1$ and k (defined for $k \in \mathcal{K} \setminus \{1\}$). This allows trains to delay for any non-negative amount of time between each loop. These changes would result in the following model:

$$\text{maximize} \sum_{k\in\mathcal{K}} \sum_{\tau\in\mathcal{T}} \sum_{g\in\mathcal{G}} r_g \rho_{g,\tau}^k - \sum_{\ell\in\mathcal{L}} f_\ell z_{I(\ell)} - \sum_{\tau\in\mathcal{T}} c_\tau y_\tau - \sum_{k\in\mathcal{K}} \sum_{\tau\in\mathcal{T}} v_\tau q_\tau^k$$

subject to

$$x_{g,\tau}^k \leq z_{I(o_g)} \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{27}$$

$$x_{g,\tau}^k \leq z_{I(d_g)} \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{28}$$

$$z_\ell \leq z_1 \qquad \ell = 2\dots(L/2)-1 \tag{29}$$

$$z_\ell \leq z_{\mathrm{L}/2} \qquad \ell = 2\dots(L/2)-1 \tag{30}$$

$$q_\tau^k \leq y_\tau \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \tag{31}$$

$$y_{\tau+1} \leq y_\tau \qquad \tau \in \mathcal{T} \setminus \{T\} \tag{32}$$

$$q_\tau^{k+1} \leq q_\tau^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \setminus \{K\} \tag{33}$$

$$x_{g,\tau}^k \leq q_\tau^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{34}$$

$$\rho_{g,\tau}^k \leq P_g x_{g,\tau}^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{35}$$

$$\sum_{\tau\in\mathcal{T}} \sum_{k\in\mathcal{K}} \rho_{g,\tau}^k \leq P_g \qquad g \in \mathcal{G} \tag{36}$$

$$(a_g - b_g) x_{g,\tau}^k \leq w_\tau + \bar{S}(d_g, k, \tau) \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{37}$$

$$(W - a_p)(x_{p,\tau}^k - 1) \leq a_p - w_\tau - \bar{S}(d_g, k, \tau) \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{38}$$

$$\Delta_\tau^k \leq H q_\tau^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \backslash \{1\} \tag{39}$$

$$n_{\ell,\tau}^k - n_{\ell-1,\tau}^k = \sum_{g \in \mathcal{G}; o_g = \ell} \rho_{g,\tau}^k - \sum_{g \in G; d_g = \ell} \rho_{g,\tau}^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; \ell \in \mathcal{L} \backslash \{1\} \tag{40}$$

$$n_{1,\tau}^k = \sum_{g \in \mathcal{G}; o_g = 1} \rho_{g,\tau}^k \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \tag{41}$$

$$n_{\ell,\tau}^k \leq u_\tau \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; \ell \in \mathcal{L} \tag{42}$$

$$z_\ell \in \{0, 1\} \qquad \ell = 1 \ldots L/2 \tag{43}$$

$$x_{g,\tau}^k \in \{0, 1\} \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{44}$$

$$\rho_{g,\tau}^k \in \mathbb{Z}^+ \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; g \in \mathcal{G} \tag{45}$$

$$y_\tau \in [0, 1] \qquad \tau \in \mathcal{T} \tag{46}$$

$$q_\tau^k \in [0, 1] \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \tag{47}$$

$$n_{\ell,\tau}^k \geq 0 \qquad \tau \in \mathcal{T}; k \in \mathcal{K}; \ell \in \mathcal{L} \tag{48}$$

$$w_\tau \geq 0 \qquad \tau \in \mathcal{T} \tag{49}$$

$$\Delta_\tau^k \in [0, H] \qquad \tau \in \mathcal{T}; k \in \mathcal{K} \backslash \{1\} \tag{50}$$

where

$$\bar{S}(\ell, k, \tau) = S(\ell, k) + \sum_{2 \leq k' \leq k} \Delta_\tau^{k'}. \tag{51}$$

Another important avenue for future research involves comparing our methods to the existing hierarchical planning methods. In order to compare these two approaches, it is first necessary to adapt existing hierarchical methods to our rural setting. After reconciling any differences in assumptions between our model and the hierarchical models, it should be possible to apply both techniques for a case study such as the one included in this work. Once a solution has been constructing using hierarchical methods, its profitability could be assessed according to the measures outlined in our objective in order to provide a quantitative comparison between the two techniques.

It should be noted that certain alternate objectives for our problem could be considered using our customized solution methodology. For example, changing cost parameters to zero alters our model to maximize total passengers served, regardless of cost. To maximize passengers served, all passenger revenue values should equal one. Slight modifications to our solution methodology could then accommodate for a budget constraint while maximizing passengers served. Specifically, cost-tracking measures could be added throughout the heuristic to ensure that changes to the solution do not result in a violation of the budget constraint. It would also be relatively simple to minimize the number of trains used or the train and station costs needed to reach a prescribed service level. For this problem, our methods could be used to serve the required number of passengers by adding as many trains as necessary, and then a slight modification of the solution improvement routine could be used to ensure that this ridership level is maintained while the solution is improved. Similarly, it is possible to include additional values in the objective, such as minimizing the passenger delay time.

It would also be interesting to investigate how to allow customers to be served by multiple potential stations, while capturing their preference of one station over another. This is relevant when multiple "park & ride" facilities could serve an intermodal customer or in very densely populated areas where multiple potential station locations are being considered in a relatively small area.

Finally, the inclusion of the stochastic nature of passenger demand, train schedules, and/or future population growth is needed to assist planners in accounting for the uncertainty associated with community growth. We are especially interested in modeling the phenomenon that has been observed after many passenger rail systems are implemented in which the rail system itself causes a shift in population growth trends.

8 Conclusions

The model we have proposed is a first step at integrating many decisions faced by rail system planners. It is important to point out that we consider only a portion of the costs associated with an operational commuter rail system. In fact, we have focused on a subset of the overall process required to design a new rail system. We have, however, integrated portions of the planning process that have typically been considered separately in the planning process.

Despite the fact that we were forced to resort to a heuristic to solve our problem, our heuristic continues to exploit the characteristics of a rural rail setting, including simpler networks and train movement. Specifically, our assumption that the rail system will follow a “linear” network allows us to efficiently model train movements due to the single feasible path for vehicles to follow. This method of modeling train movements is particularly important in determining whether a train can feasibly serve a passenger group within the heuristic. These factors allow for a solution methodology that more accurately models the interactions between the problems we have integrated than traditional hierarchical methods. Our heuristic is motivated by concepts that are mature in the operations research community. However, the neighborhood considered in our heuristic is unique from those considered in single-stage planning problems. The model that we developed is complex, and the strong interaction between continuous and discrete decisions made it difficult to apply common heuristic methods without extensive customization. In real-world applications, complex side constraints and interactions are common, and heuristic techniques are often necessary to produce suitable solutions. We feel that our customized approach can serve as to assist others faced with problems not easily solved by “out of the box” heuristics.

Finally, our case study provides a glimpse into the real world applicability of integrated rail planning models. Our results for NWA indicate that a passenger rail system may, in fact, be a good option for planners in the area to pursue further. Our objective value should not be interpreted to mean such a system would operate with large daily profits since many costs were not considered here (e.g. administrative overhead, track construction). We have merely shown that such a system could potentially be an operational success with modest adoption levels.

With further development, and with technological advances in computing, integrated methods should eventually become a reality for rail planners in rural settings and later in larger systems.

This project has been supported by a grant from the Mack-Blackwell Rural Transportation Center.

Disclaimer: The views and conclusions contained in this work are those of the investigators and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Transportation.

References

[1] American Public Transit Association resource library. APTA `http://www.apta.com/resources/Pages/Default.aspx`.

[2] Approved budget fiscal year 2011. Austin Capital Metro `http://www.capmetro.org/`.

[3] Benton county tax assessor property search. `www.co.benton.ar.us/Assessor`.

[4] Census transportation planning package 2000. RITA Bureau of Transportation Statistics `http://www.transtats.bts.gov/Tables.asp`.

[5] Real energy prices. `www.eia.gov/steo/fsheets/index.cfm`.

[6] Statistical abstract of the united states: 2011. U.S. Census Bureau `http://www.census.gov/compendia/statab/overview.html`.

[7] Washington county tax assessor property search. `www.co.washington.ar.us/Assessor/Index.html`.

[8] Visioning rail transit in northwest arkansas: Lifestyles and ecologies. University of Arkansas Community Design Center, 2007. `http://uacdc.uark.edu/books/complete/8Visioning_Rail_Transit_in_Northwest_Arkansas.pdf`.

[9] U. Brännlund, P.O. Lindberg, A. Nou, and J.E. Nilsson. Railway timetabling using lagrangian relaxation. *Transportation Science*, 32:358–369, 2000.

[10] M. Bussieck, P. Kreuzer, and U.T. Zimmermann. Optimal lines for railway systems. *European Journal of Operations Research*, 96(1):54–63, 1997.

[11] M. Bussieck, T. Winter, and U.T. Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79, 1997.

[12] A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50, 2002.

[13] A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth. *Transportation: Passenger Rail Optimization*, chapter 3, pages 129–181. Elsevier, 2007.

[14] M.A. Forbes, J.N. Holt, and A.M. Watts. Exact solution of locomotive scheduling problems. *Journal of the Operations Research Society*, 42:825–831, 1991.

[15] M.A. Forbes, J.N. Holt, and A.M. Watts. An exact algorithm for multiple depot bus scheduling. *European Journal of Operations Research*, 72:115–124, 1994.

[16] Peggy Glen, editor. *Glossary of Transit Terminology*. American Public Transit Association, 1994. `http://apta.com/resources/reportsandpublications/Documents/Transit\_Glossary\_1994.pdf`.

[17] J. Goossens, S. van Hoesel, and L. Kroon. On solving multi-type railway line planning problems. *European Journal of Operations Research*, 168, 2006.

[18] M.F. Gorman. An application of genetic and tabu searches to the freight railroad operating plan problem. *Annals of Operations Research*, 78:51–69, 1998.

[19] A. Higgins, E. Kozan, and L Ferreira. Heuristic techniques for single line train scheduling. *Journal of Heuristics*, 3:43–62, 1997.

[20] Harry Kao. Commute map. `http://hairycow.name/commute_map/`.

[21] L. Kroon, G. Maroti, M.R. Helmrich, M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research*, 42:553–570, 2007.

[22] L.G. Kroon, R. Dekker, and M.J.C.M Vromans. Cyclic railway timetabling: A stochastic optimization approach. *Railway Optimization*, 2004.

[23] G.F. List, B. Wood, L.K. Nozick, M.A. Turnquist, D.A. Jones, E.A. Kjedlgaard, and C.R. Lawton. Robust optimization for fleet planning under uncertainty. *Transportation Research*, 39:209–227, 2003.

[24] T.L. Magnanti and R.T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[25] M.F. Mammana, S. Mecke, and D. Wagner. The station location problem on two intersecting lines. *Electronic Notes in Theoretical Computer Science*, 92, 2004.

[26] A. Pepin, G. Desaulniers, A. Hertz, and D. Huisman. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *J Sched*, 12:17–30, 2009.

[27] Marc J. Perry and Paul J. Mackun. Population change and distribution: Census 2000 brief. US Census Bureau `http://www.census.gov/prod/2001pubs/c2kbr01-2.pdf`, April 2001.

[28] A. Schobel. Locating stops along bus or railway lines - a bicriteria problem. *Annals of Operations Research*, 136, 2005.

[29] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2, 1989.

[30] T. Stützle. *Local search algorithms for combinatorial problems - analysis, improvements, and new applications.* PhD thesis, Darmstadt University of Technology, 1998.

[31] P.J. Zwaneveld, L.G. Kroon, H.E. Romeijn, and M Salomon. Routing trains through railway stations: Model formulation and algorithms. *Transportation Science*, 30, 1996.

9 Appendix A: Notation and decision variables

Sets

Table 27: Sets

$\mathcal{G}$	$\mathcal{G}$, indexed by g, represents the set of potential passenger groups in the rail system (members of the same passenger group share origins, destinations, and arrival deadlines)
$\mathcal{K}$	$\mathcal{K} = \{1 \dots K\}$, indexed by k, represents the number of loops that a train may make around the network where K is a calculated upper bound on the possible number of loops (A loop is defined to be leaving the first station and visiting each open station in the network before returning to the first station)
$\mathcal{L}$	$\mathcal{L} = \{1 \dots \lvert\mathcal{L}\rvert\}$, indexed by ℓ, contains all possible station locations along the track with one element of $\mathcal{L}$ corresponding to each location for each direction of travel. That is, station 1 and L both correspond to the first station, 2 and $\lvert\mathcal{L}\rvert - 1$ to the second possible station, and so on until the last station, which is represented by $\frac{\lvert\mathcal{L}\rvert}{2}$ and $\frac{\lvert\mathcal{L}\rvert}{2} + 1$.
$\mathcal{T}$	$\mathcal{T} = \{1 \dots T\}$, indexed by τ, represents the set of trains that could potentially serve customers on the network

Parameters

Table 28: Parameters

a_g	The latest possible arrival time for group g (defined for $g \in \mathcal{G}$, $0 \leq a_g \leq H$)
b_g	The maximum amount of time that passenger group g may arrive before its arrival deadline, a_g (defined for $g \in \mathcal{G}$, any train that arrives at d_g before a_g but after $a_g - b_g$ is eligible to serve all or some of the passengers in group g
c_τ	The cost of procuring train τ (defined for $\tau \in \mathcal{T}$)
d_g	The destination location for group g (defined for $g \in \mathcal{G}$, $d_g \in \mathcal{L}$)
f_ℓ	The cost of procuring a station at location $\ell \in \mathcal{L}$ (defined for $\ell = 1 \ldots L/2$)
H	The length of the service horizon, in minutes
o_g	The origin location for group g (defined for $g \in \mathcal{G}$, $o_g \in \mathcal{L}$)
P_g	The total number of passengers in group g (defined for $g \in \mathcal{G}$)
r_g	The revenue for serving one passenger from group g a single time (defined for $g \in \mathcal{G}$)
t_ℓ	The time required for any train to traverse the distance from location $\ell - 1$ to location ℓ at cruise speed (defined for $\ell \in \mathcal{L}$, for $\ell = 1$ this value should be zero or should represent the delay between arriving at location $\lvert\mathcal{L}\rvert$ on loop k and starting loop $k + 1$)
u_τ	The capacity of train τ, in customers (defined for $\tau \in \mathcal{T}$)
v_τ	The cost of operating train τ for one loop (defined for $\tau \in \mathcal{T}$)
W	$W = \lvert\mathcal{K}\rvert \sum_{\ell \in \mathcal{L}}(\delta_\ell + t_\ell) + H)$ is a logical upper bound for the $w^k_{\ell,\tau}$ variables
δ_ℓ	The additional time required to stop at location ℓ in the even that a station exists there, including the delay associated with deceleration and acceleration (defined for $\ell \in \mathcal{L}$)

Decision Variables

Table 29: Decision Variables

$n_{\ell,\tau}^k$	A non-negative, continuous variable representing the number of passengers on train τ leaving location ℓ for the k^{th} time (defined for $\ell \in \mathcal{L}; \tau \in \mathcal{T}; k \in \mathcal{K}$, may be a continuous nonnegative variable but will take on integer values due to problem structure)
q_τ^k	A continuous variable on $[0,1]$ that takes a value of 1 if train τ is active for loop k and 0 otherwise (defined for $\tau \in \mathcal{T}; k \in \mathcal{K}$, may be continuous on $[0,1]$ but will take on integer values due to problem structure)
$w_{\ell,\tau}^k$	A continuous non-negative variable that represents the time that has elapsed from the beginning of the horizon to the time that train τ arrives at location ℓ for the k^{th} time (defined for $\ell \in \mathcal{L}; \tau \in \mathcal{T}; k \in \mathcal{K}$)
w_τ	After the variable-space reduction, the nonnegative continuous variable w_τ represents the time that train τ enters the system (defined for $\tau \in \mathcal{T}$)
$x_{g,\tau}^k$	A binary variable with a value of 1 if group g is assigned to train τ on its k^{th} loop and 0 otherwise (defined for $g \in \mathcal{G}; \tau \in \mathcal{T}; k \in \mathcal{K}$)
y_τ	A continuous variable on $[0,1]$ that takes a value of 1 if train τ is active for any loop and 0 otherwise (defined for $\tau \in \mathcal{T}$, may be continuous on $[0,1]$ but will take on integer values due to problem structure)
z_ℓ	A binary variable with value 1 if a station is procured at location ℓ and 0 otherwise (defined for $\ell = 1 \ldots \|\mathcal{L}\|/2$)
Δ_τ^k	A continuous non-negative variable that represents the delay time for train τ between loops $k-1$ and k (defined for $k \in \mathcal{K} \backslash \{1\}$)

10 Appendix B: Detailed heuristic outline

The outline below describes our heuristic in moderate detail. Simple or self-explanatory functions have not been outlined completely, but some of the more complicated functions have been described below the heuristic. The file *instance.dat* contains values for all instance parameters. The file *parameters.txt* contains values for user-defined heuristic settings. At various points, solution information is saved in four different global "bins:" *cand*, *best*, *hold*, and *tsih*. These "bins" store some or all of the values for problem variables and the objective value.

```
MAIN(instance.dat, parameters.txt)
    // create necessary instance data structures and read values from instance.dat
    INITIALIZE(instance.dat)
    // read heuristic parameters from parameters.txt
    READPARAMS(parameters.txt)
    // return the station configuration with the greatest potential profit,
    // ignoring scheduling issues and train costs
    candZ = GETSTARTINGCONFIG()
    // construct fleet size and solution from scratch
    CONSTRUCTSOLUTION()
    // store new best solution
    REPLACEBESTWITHCAND()
    activeTrains = SUM(candY)
    holdObj = −10000
    for i = 0 to FLEETRANGE − 1
        fleetSize = activeTrains − FLOOR(FLEETRANGE/2) + i
        for j = 0 to 1
            // construct solution for specific fleet size using method j
            CONSTRUCTFLEETSOLUTION(fleetSize, j)
            if candObj > bestObj − TITHRESHOLD
                // improve cand solution by manipulation customer
                // assignments and train scheduling
                IMPROVECANDSOLUTION(fleetSize, TCUTOFF, GCUTOFF)
            if candObj > holdObj
                REPLACEHOLDWITHCAND()
    if holdObj > bestObj
        REPLACEBESTWITHHOLD()
```

Main Cont'd

```
$stepZ = bestZ$
$holdObj = -10000$
$noImpr = 0$
$numOpen = $ Sum($bestZ$)
for $i = 0$ to $numOpen - 3$
    $candZ = stepZ$
    for $\ell = 1$ to $L/2 - 2$
        if $candZ[\ell] == 1$
            $candZ[\ell] = 0$
            ConstructSolution()
            $activeTrains = $ Sum($candY$)
            for $i = 0$ to fleetRange
                $fleetSize = activeTrains - $ Floor(fleetRange/2) $+ i$
                for $j = 0$ to 1
                    ConstructFleetSolution($fleetSize, j$)
                    if $candObj > bestObj - $ tiThreshold
                        ImproveCandSolution($fleetSize$, tCutoff, gCutoff)
                    if $candObj > holdObj$
                        ReplaceHoldWithCand()

    $stepZ = holdZ$
    if $holdObj > bestObj$
        ReplaceBestWithHold()
        $noImpr = 0$
    else
        $noImpr+ = 1$
        if $noImpr >$ closeCutoff
            Break
if $bestObj < 0$
    ReplaceBestWithZero()
$stepZ = bestZ$
$holdObj = -10000$
$noImpr = 0$
$numClosed = L/2 - $ Sum($bestZ$)
if $numClosed == L/2$
    $stepZ[0] = 1$
    $stepZ[L/2\text{-}1] = 1$
```

MAIN CONT'D

```
for $i = 1$ to $numClosed - 3$
    $candZ = stepZ$
    for $\ell = 1$ to $L/2 - 2$
        if $candZ[\ell] == 0$
            $candZ[\ell] = 1$
            ConstructSolution()
            $activeTrains = \textsc{Sum}(candY)$
            for $i = 0$ to fleetRange $- 1$
                $fleetSize = activeTrains - \text{Floor}(\textsc{fleetRange}/2) + i$
                for $j = 0$ to $1$
                    ConstructFleetSolution($fleetSize, j$)
                    if $candObj > bestObj - \textsc{tiThreshold}$
                        ImproveCandSolution($fleetSize$, tCutoff, gCutoff)
                    if $candObj > holdObj$
                        ReplaceHoldWithCand()
    $stepZ = holdZ$
    if $holdObj > bestObj$
        ReplaceBestWithHold()
        $noImpr = 0$
    else
        $noImpr+ = 1$
        if $noImpr >$ openCutoff
            Break
$stepZ = bestZ$
$holdObj = -10000$
$noImpr = 0$
$numOpen = \textsc{Sum}(bestZ)$
$numClosed = L/2 - numOpen$
for $i = 0$ to swapSteps $- 1$
    for $j = 0$ to $numClosed - 1$
        for $m = 0$ to $numOpen - 1$
            // open $j^{th}$ closed station and close $m^{th}$ open station
            $candZ = \textsc{SwapOpenClosed}(j, m)$
            ConstructSolution()
            $activeTrains = \textsc{Sum}(candY)$
            for $i = 0$ to fleetRange $- 1$
                $fleetSize = activeTrains - \text{Floor}(\textsc{fleetRange}/2) + i$
                for $j = 0$ to $1$
                    ConstructFleetSolution($fleetSize, j$)
                    if $candObj > bestObj - \textsc{tiThreshold}$
                        ImproveCandSolution($fleetSize$, tCutoff, gCutoff)
                    if $candObj > holdObj$
                        ReplaceHoldWithCand()
```

Main Cont'd

```
        $stepZ = holdZ$
        if $holdObj > bestObj$
            ReplaceBestWithHold()
            $noImpr = 0$
        else
            $noImpr+ = 1$
            if $noImpr >$ swapCutoff
                Break
    $stepZ = bestZ$
    $holdObj = -10000$
    $noImpr = 0$
    for $i = 0$ to revisitSteps $- 1$
        for $j = 0$ to revisitStepSize $- 1$
            // randomly choose whether to open or close stations
            if RandomU() $< 0.5$ // open stations
                $numClosed = L/2 -$ Sum($StepZ$)
                $toOpen =$ RandomIntBetween($1, \min(5, numClosed)$)
                $candZ =$ OpenStations($toOpen$)
                ConstructSolution()
                $activeTrains =$ Sum($candY$)
                for $i = 0$ to fleetRange $- 1$
                    $fleetSize = activeTrains -$ Floor(fleetRange$/2$) $+ i$
                    for $j = 0$ to $1$
                        ConstructFleetSolution($fleetSize, j$)
                        if $candObj > bestObj -$ tiThreshold
                            ImproveCandSolution($fleetSize$, tCutoff, gCutoff)
                        if $candObj > holdObj$
                            ReplaceHoldWithCand()
            else // close stations
                $numOpen =$ Sum($StepZ$)
                $toClose =$ RandomIntBetween($1, \min(5, numOpen - 2)$)
                $candZ =$ CloseStations($toClose$)
                ConstructSolution()
                $activeTrains =$ Sum($candY$)
                for $i = 0$ to fleetRange $- 1$
                    $fleetSize = activeTrains -$ Floor(fleetRange$/2$) $+ i$
                    for $j = 0$ to $1$
                        ConstructFleetSolution($fleetSize, j$)
                        if $candObj > bestObj -$ tiThreshold
                            ImproveCandSolution($fleetSize$, tCutoff, gCutoff)
                        if $candObj > holdObj$
                            ReplaceHoldWithCand()
```

Main Cont'd

```
        $stepZ = holdZ$
        if $holdObj > bestObj$
            ReplaceBestWithHold()
            $noImpr = 0$
        else
            $noImpr += 1$
            if $noImpr >$ revisitCutoff
                Break
    ReplaceCandWithBest()
    $fleetSize = $ Sum($candY$)
    // try to serve any remaining unserved customers
    PolishPartOne()
    // extended version of ImpoveCandSolution()
    ImproveCandSolution($fleetSize$, polishTCutoff, polishGCutoff)
    // shut down trains that aren't even covering own cost, shut down loops that arent covering $v$
    // attempt to serve customers from these trains elsewhere
    ClearLightRunsAndLoops()
    if $candObj > bestObj$
        ReplaceBestWithCand()
```

```
ConstructSolution()
    for $\tau = 0$ to $T - 1$
    $seedGroup = -1$
    // only returns groups that have not served as a seed, if none returns -1
    $seedGroup$ = GetLargestUnservedGroup()
    if $seedGroup == -1$
        Break // all groups tried
    // return $w$ so that train $\tau$ can serve $seedGroup$ in the middle of its window
    $w_\tau$ = ScheduleTrainToServeG($seedGroup$)
    // attempt to add all unserved groups to train $\tau$ & update $x$ variable
    // train schedule can flex to serve new customers but cannot uncover others to do so
    ServeAllPossibleGroups($w_{tau}$)
    // call CPLEX to solve MIP - maximize revenue subject to capacity constraint
    // also updates $n$ and $q$ variables based on ridership
    SolveRhoMIP()
    // check if newly scheduled train is profitable
    if CheckTrainProfitability($\tau$) == 0
    // remove customers assigned to train $\tau$ reset associated variables
        ClearTrain($\tau$)
        $\tau - = 1$
    else
        $candY[\tau] = 1$

ConstructFleetSolution($fleetSize$, $method$)
    if $method == 0$ // largest group
        for $\tau = 0$ to $fleetSize - 1$
            $candY[\tau] = 1$
            $seedGroup = -1$
            $seedGroup$ = GetLargestUnservedGroup()
            if $seedGroup == -1$
                Break
        $w_\tau$ = ScheduleTrainToServeG($seedGroup$)
        ServeAllPossibleGroups($w_{tau}$)
        SolveRhoMIP()
    else // equidistant starting schedules
            $buffer$ = GetLengthOfLoop()/$fleetSize$
            for $\tau = 0$ to $fleetSize - 1$
            $w_\tau = \tau * buffer$
            ServeAllPossibleGroups($w_{tau}$)
            SolveRhoMIP()
```

```
ImproveCandSolution(fleetSize, tCutoff, gCutoff)
    for $i = 0$ to impTrainLoops $-1$
        // choose two trains using RandomIntBetween() with uniqueness check
        // train -1 serves as dummy train for unserved passengers
        train1, train2 = ChooseTrainPair()
        for $j = 0$ to impGroupLoops $- 1$
            ReplaceTsihWithCand()
            // choose one group from each train selected above
            group1, group2 = ChooseGroupFromEachTrain(train1, train2)
            temp = RandomU()
            if temp < probGroupSwap// swap two groups
                // serve group on train if possible and update variables
                // set $\rho$ as large as possible for newly served group
                serve1 = ServeGroupOnTrain(group1, train2)
                serve2 = ServeGroupOnTrain(group2, train1)
                if serve1 + serve2 == 2
                    if tsihObj > candObj
                        ReplaceCandWithTsih()
                        gNoImpr = 0
                        tNoImpr = $-1$
                    else
                        gNoImpr+ = 1
                        if gNoImpr > gCutoff
                            Break
        tNoImpr+ = 1
        if tNoImpr > tCutoff
            Break
```