Locomotive Scheduling Under Uncertain Demand

SCOTT SMITH AND YOSEF SHEFFI

Each day, railroads face the problem of allocating power to trains. Often, power requirements for each train are not known with certainty, and the fleet of locomotives may not be homogeneous. To deal with both of these complications, we formulate a multicommodity flow problem with convex objective function on a time-space network. The convex objective allows us to minimize expected cost under uncertainty by penalizing trip arcs likely to have too little power. Our solution heuristic sends locomotives down the shortest paths (based on marginal arc costs) in the time-space network and then attempts to improve interchanges of locomotives around cycles. Two lower bounds are also developed by relaxing the multicommodity aspect of the problem. In 19 test problems, ranging in size from 15 to 404 arcs, the heuristic performed well, with short running times and costs averaging within 3 percent of the best of the two lower bounds developed.

A problem frequently faced by transportation carriers is the allocation of a fixed supply of vehicles to a given schedule. Examples include the allocation of locomotives to freight trains, of buses to transit routes, and of airplanes to flights. These examples have the following features in common:

1. There is a published or "committed to" schedule of services that have to be carried out;
2. The supply of vehicles to trips can be represented as an integer, multicommodity minimum cost flow problem over a network of trip, layover, and storage arcs. The problem has multicommodity aspects because the vehicle fleet is not homogeneous; for example, locomotives may have different power ratings and airplanes may be of different sizes. (Naturally, however, there are some important differences between the modes. For example, in the rail mode, two or more locomotives typically are used to meet demand for a given trip, whereas only one airplane is used for a single airline trip);
3. Even though the schedule is fixed, the demand for service may vary. In the rail context, the tonnage of a given train is variable. In the bus or airline context, the number of passengers on a given trip will vary. Further, it may sometimes be desirable not to meet all the demand, for example, by having standees on buses, or refusing airline reservations, or leaving cars behind for the next freight train to pick up.

This paper formulates this allocation problem and suggests solution techniques in the context of rail. First, background information on both the formulation of the problem and past research in this area is presented. Second, the problem is formulated as a mathematical program. Third, a fast heuristic solution technique is presented. Finally, the results of the heuristic are compared with lower bounds obtained through various relaxations. The techniques presented here explicitly consider uncertainty in locomotive demand and are able to deal with locomotives of different power ratings.

Department of Civil Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Mass. 02139.

BACKGROUND

This section looks at the network representation of the locomotive scheduling problem. This formulation underlies virtually all other attempts in the literature to develop a solution for this problem. Some of that research is reviewed in the second part of this section.

Time-Space Representation

The rail scheduling problem is typically formulated as a minimum cost flow problem on a time-space network, which is a graph of locations versus time on which activities are plotted (Figure 1). Each node in this network represents a terminal (yard) at a point in time, and arcs are of the following types:

1. Trip arcs represent trains between the upstream terminal node and the downstream terminal node that the arc connects.

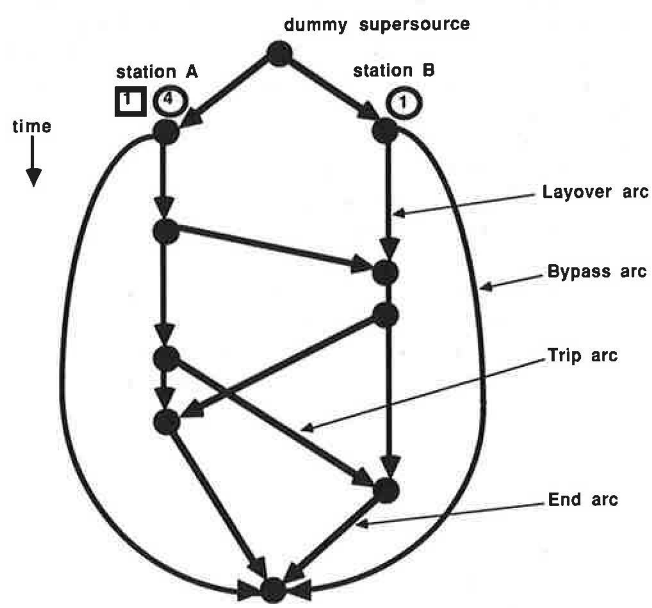

FIGURE 1 Sample time-space network.

There is a power requirement for each trip, which may be represented either by a lower bound on the arc flow of locomotives or by a penalty function that increases greatly as this flow falls below the minimum desired value;

2. Layover arcs represent short-term storage at a terminal. They have a lower bound of 0 and some fixed cost per unit;
3. Bypass arcs represent long-term storage of unneeded units and have very low cost per unit; and
4. End arcs represent locomotive requirements at each terminal at the end of the planning horizon. Any practical time-space representation has a finite planning horizon. Therefore, if this horizon is short, end effects must be considered. In the model here, we want to know how many locomotives are needed at each terminal at the end of day, week, or whatever period we are modeling. Thus, the end arcs will have either cost functions or lower bounds similar to those for trip arcs.

This is a multicommodity network flow problem with either a "bundle constraint" in the lower bound for each arc or a penalty term in the cost function that "bundles" the commodities. We flow locomotive units of various types through the network, but a minimum level of motive power must be met for each arc.

Past Work

Comprehensive reviews of rail scheduling are contained in two papers by Assad (*1,2*) and one by Peterson (*3*). Some of the earliest analytical work in locomotive assignment is that of Bartlett in 1957 (*4*), who presented a pairing algorithm to assign vehicles to a fixed schedule. Later, McGaughey et al. (*5*) described the distribution of locomotives and cabooses with a time-space network model. They used an out-of-kilter algorithm to find the optimal flow of units through a single-commodity network with a fixed lower bound on the power supplied to each arc. In 1976 Florian et al. (*6*) considered the multicommodity aspect of locomotive scheduling, with fixed lower bounds. They used Bender's decomposition to solve this multicommodity flow problem and reported good results with medium-size (about 200 train movements) problems but had less success with larger problems. In 1980, Booler (*7*) formulated the same multicommodity flow problem but obtained an integer solution using a heuristic method based on linear programming.

All of this work assumes deterministic, known lower bounds on the power flows. Furthermore, there has been only limited success with multicommodity flows, particularly with large problems, as already mentioned. As the first step to the explanation of our approach to the problem, the next section formulates the locomotive allocation problem as a mathematical program. Later we assume that the lower bound is not known with certainty, and we reformulate the problem using a penalty function.

FORMULATION

This section starts with the "traditional" mathematical programming formulation of the problem. It then incorporates the uncertainty in locomotive requirements directly into the formulation using several probability density functions of the power needs. The formulation and notation that follow relate to the time-space representation of the locomotive assignment problem.

Define the following:

i = arc number in the time-space network,
j = locomotive type,
x_{ij} = flow of locomotive type j on arc i,
H_j = horsepower rating of locomotive type j
s_i = horsepower flow on arc i ($s_i = \Sigma_j H_j x_{ij}$),
x_j = vector of locomotive flows of type j on all arcs,
$C_i(s_i)$ = general operating cost function on arc i,
c_i = operating cost per unit HP flow on arc i,
l_i = demand for power on trip arc i (this may be either a deterministic value or a random variable),
F_i = cumulative distribution function for l_i,
f_i = probability density function for l_i,
μ_i = average demand for power (expected value of l_i),
σ_i = standard deviation of demand for power,
$P_i(s_i)$ = general penalty cost function for power shortfall on link i,
p_i = penalty per unit of power shortfall on arc i,
$Z_i(s_i)$ = cost function on arc i (including operating cost and penalty),
N = node arc incidence matrix for the time-space network, and
b_j = vector of sources and sinks for locomotive type j.

We first formulate the problem with deterministic lower bounds on the power requirements, and then show how these lower bounds can be modeled as random variables. That formulation leads to the use of penalty functions in the objective of the mathematical program. In all cases, we assume that the lower bound is expressed in terms of horsepower, so that combinations of locomotive types with the same total power rating are interchangeable. The mathematical formulation is

$$\min \Sigma_i c_i \left(\Sigma_j H_j x_{ij}\right) \tag{1}$$

subject to

$$\Sigma_j H_j x_{ij} \geq l_i \text{ for all } i \tag{2}$$

$$N x_j = b_j \text{ for all } j \tag{3}$$

$$x_{ij} \text{ integer and } \geq 0 \tag{4,5}$$

This is a multicommodity minimum cost network flow problem. The objective is to assign all locomotive types j to the network, whose node-arc incidence matrix is N, at minimum cost. The various locomotive types cannot be assigned separately because they all contribute to the power on each train link. This bundling of locomotive types appears in the lower bound constraint 2.

Recall that the original problem calls for uncertainty in demand. Therefore, the fixed lower bound formulation of Equations 1 through 5 may not be realistic. This is because a fixed lower bound can be thought of as an infinite cost penalty on flows below it. Such a cost function for an arc with a lower bound of 5000 and cost per unit flow of c is shown in Figure 2. According to this cost function, 4999 HP on this train has an infinite cost whereas 5000 HP has the lowest cost, clearly an unrealistic situation in a world where the 5000-HP

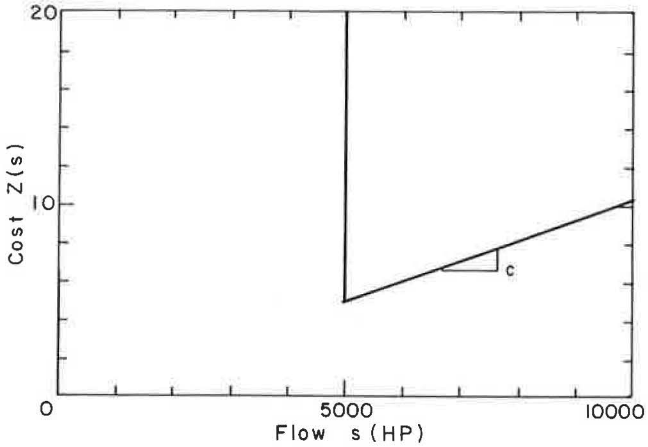

FIGURE 2 Arc cost function with deterministic lower bound.

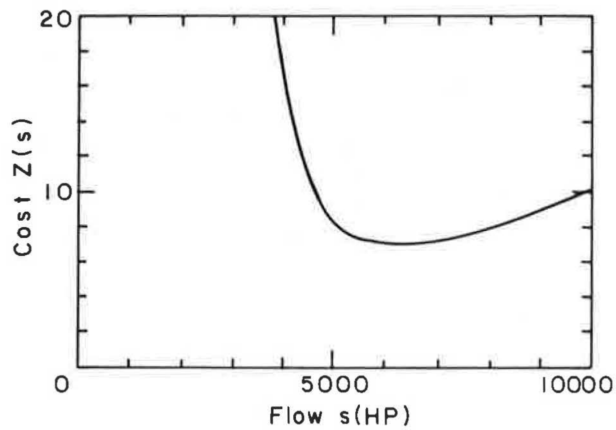

FIGURE 3 Arc cost function with uncertain power demand.

requirement is actually a random variable, which can attain values lower than 5000 HP. A more realistic cost function (Figure 3) might be obtained by reasoning as follows:

Our forecast demand is 5000 HP but we know the forecast might be off by as much as 1000 HP. Therefore, our safest assignment would be to have 6000 HP on this train; 5000 HP would probably work; 4000 HP would be unacceptable. However, we do not want a fixed lower bound of 6000 HP because 5000 HP may be all the power we have available.

To yield a cost function that looks like the one shown in Figure 3, we move constraint 2 into the objective with a penalty term. By doing so, we acknowledge that (a) power requirements may vary in a random manner and (b) the lower bound on power is not a hard-and-fast rule; rather, there is a trade-off between service quality and the amount of power supplied. This formulation provides a more realistic representation and, arguably, makes the problem easier to solve. Thus, rather than having the demand for power, l_i, as a fixed lower bound, it is modeled as a random variable.

The shape of the cost function derived this way depends on three elements:

1. The operating cost of additional power; we assume this is some function, $C_i(s_i)$ of the power supplied, s_i. This cost is assumed independent of the demand, l_i, and locomotive type;
2. The distribution of the demand for power, l_i. The probability density function for l_i is denoted by f_i and its cumulative distribution by F_i. This distribution has mean μ_i and variance σ_i^2; and
3. The magnitude and shape of the penalty, given a shortage. This will have the form $P_i(l_i - s_i)$ (recall that $s_i = \Sigma_j H_j x_{ij}$).

In the equations that follow, the subscript i is dropped to make the notation easier to read.

The cost as a function of power supply, s, is $Z(s)$. Given the probability density function of l, this cost can be expressed as follows:

$$Z(s) = C(s) + \int_s^\infty P(l - s) f(l)\, dl \tag{6}$$

As a tractable approximation to the normal distribution, we assume that l follows a logistic distribution where

$$F(l) = [1 + \exp((a/\sigma)(\mu - l))]^{-1} \tag{7}$$

$$f(l) = (a/\sigma)\exp((a/\sigma)(\mu - l)) \times [1 + \exp((a/\sigma)(\mu - l))]^{-2} \tag{8}$$

where $a \equiv \pi/\sqrt{3} \approx 1.81$

The cost function, $Z(s)$, now becomes

$$Z(s) = cs + p\int_s^\infty lf(l)\, dl - ps\int_s^\infty f(l)\, dl$$
$$= cs + p\int_s^\infty lf(l)\, dl + ps(F(s) - 1) \tag{9}$$

After integrating (by parts) the cost becomes

$$Z(s) = cs + (p\sigma/a)\log(1/F(s)) \tag{10}$$

In the logistic distribution mentioned above, negative demand is theoretically possible. However, the parameters are such that this is not a problem in practice.

Cost functions were also derived for uniform and gamma distributed demands, with some examples plotted in Figure 4. Note the following points:

1. All functions approach the $\sigma = 0$ case asymptotically as s becomes either very large or very small with respect to μ;
2. There is not much difference between the cases with logistic, uniform, and gamma distributions. This is reassuring, because it indicates that the exact shape of the distribution for demand may not matter much, and we can use the logistic distribution to form a tractable cost function. Although the gamma distribution is probably the most realistic representation of l (it never has values below zero), it does not yield a closed form for $Z(s)$ and, consequently, is difficult to work with;
3. All functions have the desired shape (as in Figure 3). In the remainder of the paper, we assume that the demand for power l is logistically distributed with mean μ and standard deviation σ.

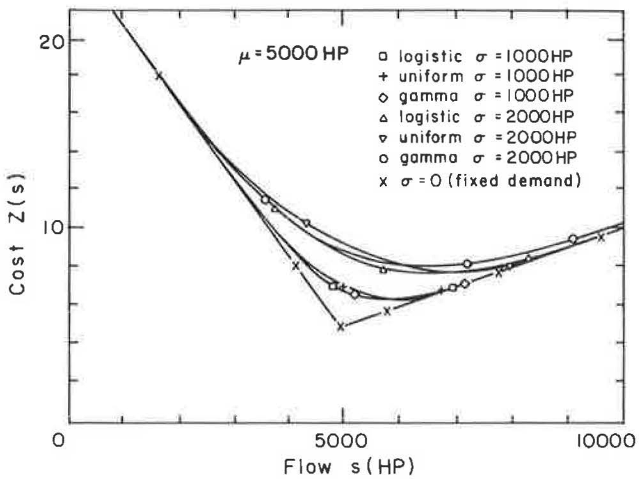

FIGURE 4 Arc cost function under various demand distributions.

The problem we are solving can now be summarized as follows: Minimize the total cost of flowing power on all arcs, subject to the following constraints:

1. Flow conservation constraints are met for each power class and
2. The number of locomotives on each arc is integer and nonnegative.

In other words:

$$\min \Sigma_i\{c_is_i + (p_i\sigma_i/a) \log(1 + \exp [(a/\mu_i)(\mu_i\text{-}s_i)])\} \quad (11)$$

$$\text{subject to } Nx_j = b_j \text{ for all } j \quad (12)$$

$$x_{ij} \text{ integer and } \geq 0 \quad (13,14)$$

(As before, s_i is defined as $\Sigma_j H_j x_{ij}$ and $a = 1.81$.)

This representation, in addition to having the advantages mentioned earlier, also lends itself well to the treatment of uncertain end effect arcs, which can be modeled like trip arcs with high σ. The next section looks at solution approaches for this problem.

SOLUTION APPROACHES

We have formulated a nonlinear, multicommodity integer network flow problem. Exact solution techniques for such a problem are likely to be neither easy nor fast. This problem, however, is similar to the (multicommodity) traffic assignment equivalent program. The traffic assignment problem deals with the assignment of an origin-destination trip matrix to a transportation network so as to minimize each user's travel time (or cost). In traffic assignment, the arcs have a fuzzy upper bound that arises from highway congestion effects, whereas our problem has a fuzzy lower bound arising from power shortfalls. Both problems can be formulated as a solution to a convex program over network flows. The heuristic used to solve the locomotive scheduling borrows from both the incremental assignment method and the Frank-Wolfe algorithm used to solve the traffic assignment problem.

Review of Incremental Assignment and Frank-Wolfe

In incremental assignment, we start with zero flow on the network and choose a number of increments, n. The algorithm, then, in each of n iterations, greedily assigns $1/n$ of the total flow along each shortest path from origin to destination. Because the cost function is nonlinear, these shortest paths may change with each iteration (*8*). Although this algorithm can be set to maintain integrality of the solution, has intuitive appeal, and is easy to implement, it has a number of shortcomings. First, it does not always work, as shown through counterexamples by Ferland et al. (*9*). Second, if it is to produce reasonable solutions, the number of increments may have to be very large, thus unduly increasing the running time.

The Frank-Wolfe algorithm (*8*) is a feasible direction method and therefore starts with an initial feasible solution and moves to improved solutions, maintaining feasibility throughout. It does this by developing linear approximations to the objective function and by solving linear subproblems to find the correct distances to move in improving directions. With cost minimization and a convex objective function, the Frank-Wolfe method does converge to the optimal solution and is easy to implement on networks. Furthermore, a lower bound to the optimal solution is provided at each iteration. Flows, however, are split between paths; thus integrality is lost. In most traffic assignment problems, convergence is rapid (about five iterations) but may be slowed if the solution is in a highly nonlinear portion of the objective function (*10*).

The Two-Commodity Heuristic Approach

The heuristic presented here obtains a feasible solution to the problem through incremental assignment, and then obtains improvements through a feasible direction method. Unlike Frank-Wolfe, it maintains integrality and exploits the integrality of the problem by moving one locomotive unit at a time, thus obviating the need for line searches in the feasible direction method.

The heuristic runs in two phases. First, it loads the network by assigning one unit at a time to shortest paths. This is referred to as the GREEDY phase. Second, after the network is loaded, it attempts improvements by sending flows around augmenting negative cycles in an INTERCHANGE phase. These augmenting cycles are similar to the augmenting paths of maximum flow algorithms in that they include both forward and reverse arcs; thus flow can be removed from an arc when going against the flow direction. Both phases are outlined in more detail below:

GREEDY Phase

Step 0. Initialization. Start with zero flow, and compute arc marginal costs at zero flow.

Step 1. Send one unit down the shortest path from any source to the supersink; update arc flows. (Note that the order in which units are selected will affect the outcome. For the experiments here, the largest units were arbitrarily selected first).

Step 2. Recompute arc marginal costs along that path.

Step 3. If all units have been sent, go to the INTERCHANGE phase, otherwise, go to Step 1.

Note that for large problems, this phase can be speeded up by sending more than one unit at first. Also, saving the shortest path tree rooted at the sink node and reoptimizing it after every assignment (rather than recomputing the shortest path at each iteration) offers another opportunity to speed up this phase of the heuristic.

INTERCHANGE Phase

Step 0. Identify arcs that are candidates for improvement. In the present implementation these are arcs with large negative marginal cost (i.e., trip arcs with insufficient power).

Step 1. Search for a flow-augmenting negative cycle involving some candidate arc. If no negative cycle can be found in the network, stop. Otherwise, go to Step 2.

Step 2. Interchange flows around this cycle and update arc marginal costs. Go back to Step 1.

The interchanges performed in the second phase are generally more complicated than simply sending one unit of flow around the cycle. This is because the interchanges often involve minor HP changes and thus may involve the exchange of two locomotive types. For example, if our two locomotives types have 2000 HP and 3000 HP, respectively, two interchanges that would produce a small horsepower change would be:

1. Add one high-power and remove one low-power unit on the arc that needs additional power (net change of 1000 HP) or
2. Add two low-power and remove one high-power unit (net change of 1000 HP).

Within the heuristic, these interchanges are performed in the following manner:

1. Create an ordered list of arcs that will benefit from more power.
2. Attempt to find an improving interchange involving one of the arcs on the ordered list. This is done as follows:
   2a. Select the first interchange type.
   2b. REPEAT.
      Select the first arc.
      REPEAT.
         Try to find an improving interchange (flow-augmenting negative cycle) with this arc and interchange type. If one is found, go to Step 3. Otherwise, select the next arc
      UNTIL all arcs examined.
      Consider the next interchange type.
   UNTIL all interchange types have been considered.
3. If we have found an interchange
   Perform the interchange and update arc marginal costs.
   Update the ordered list of arcs, go back to Step 2.
   Otherwise, we terminate, because no interchange can be found.

Example

Consider a two-node network with two trip arcs and one bypass arc (Figure 5). The arc costs are shown in Table 1. The locomotive supply includes one high-powered (3000 HP) and two low-powered (2000 HP) locomotives. The greedy phase of the heuristic performs as follows:

1. Send the high-powered unit down arc 1.
2. Send a low-powered unit down arc 2.
3. Send a low-powered unit down arc 1.

We now have 5000 HP on arc 1 and 2000 HP on arc 2. Arc 2 is short of power.

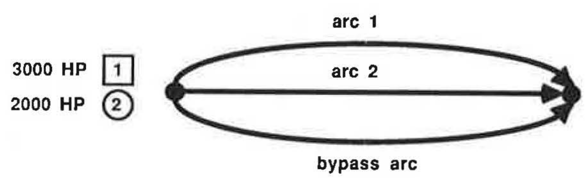

FIGURE 5 Sample network.

TABLE 1 COSTS FOR THE EXAMPLE NETWORK

Parameters	arc 1	arc 2	bypass
μ	4200	3000	0
σ	1000	1000	0
c	1	1	0
p	10	10	0
	Costs		
Flow (HP)	arc 1	arc 2	bypass
0	42002	30024	0
1000	33016	21146	0
2000	24102	12837	0
3000	15596	6829.	0
4000	8919.	4837.	0
5000	6166.	5146.	0
6000	6208.	6024.	0
7000	7034.	7003.	0
8000	8005.	8000.	0

Moving to the interchange phase, we see that the high-powered unit on arc 1 can be exchanged with the low-powered unit on arc 2. After this exchange is performed, we are finished, as no more improving interchanges can be seen.

The progress of the heuristic in terms of the arc flows and the objective function is plotted in Figure 6. The method works by first moving in big jumps (whole units) toward the optimal solution, then refining the solution by making smaller jumps (interchanges).

Advantages and Disadvantages of the Heuristic Approach

This double-phase heuristic has several advantages. First, it maintains feasibility throughout. Second, by always moving in an improving direction, the method is intuitively appealing. Therefore, it may lend itself well to interactive use. Third, it is easy to incorporate other side constraints into the framework of this heuristic. Some of these are the following:

1. Prohibition of certain locomotive types from certain sections of track,
2. Assigning newer, more reliable, locomotives to high-priority trains, and
3. Sending locomotives to home shops for scheduled maintenance.

Finally, the heuristic is also quite fast and produces close to optimal results in several test problems.

The disadvantages of this method are, first, its heuristic nature: optimality is not guaranteed. In addition, the complexity of the interchange phase increases as the number of the commodities is increased beyond two. This was not a problem in the case study reported later but may present difficulty in other applications.

LOWER BOUNDS

Several lower bounds were derived to test the performance of the heuristic. A lower bound may be (a) an optimal solution to a relaxed version of the primal problem, (b) a dual feasible solution, or (c) some combination of the foregoing, such as a dual feasible solution to a relaxed version of the primal. Two lower bounds were derived for the problem discussed here.

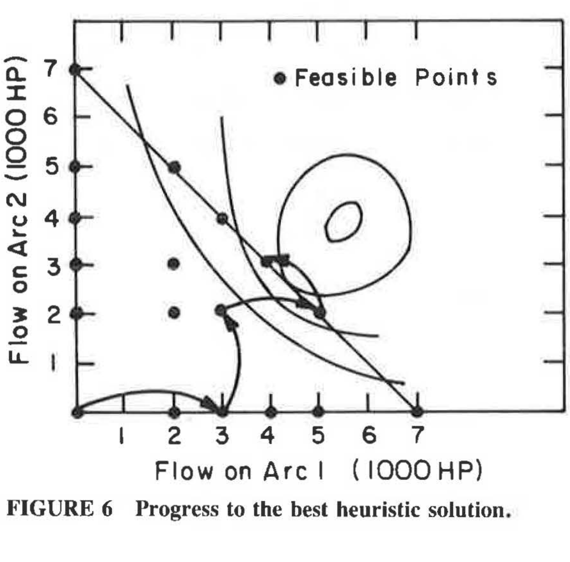

FIGURE 6 Progress to the best heuristic solution.

Frank-Wolfe Relaxation

By relaxing the integrality constraint in the original problem, we obtain a single-commodity (horsepower) network flow problem with convex objective function. This can be solved with the Frank-Wolfe (convex combinations) method already reviewed. The Frank-Wolfe method provides both a feasible solution and lower bound on the relaxed problem at every iteration. This lower bound on the relaxed problem will, naturally, also provide a lower bound on the original minimization problem in the following manner:

heuristic solution $\geq$ optimal solution

$\geq$ optimal solution to relaxed problem

$\geq$ lower bound to relaxed problem

Unfortunately, a complete relaxation of the integrality constraint in this manner may lead to a large gap between the optimal solution and the optimal solution to the relaxed problem. Such a gap makes it difficult to evaluate the performance of the heuristic.

Greatest Common Factor (GCF) Relaxation

This relaxation is based on the following observation: Any feasible solution will have a horsepower flow in each arc that is a multiple of the greatest common factor (GCF) of the horsepower ratings. For example, if there are two locomotive types rated at 2000 HP and 3000 HP, the flow on each arc will be a multiple of 1000 HP. If there are three locomotive types with ratings of 1750 HP, 2000 HP, and 3000 HP, the flow on each arc must be a multiple of 250 HP. Any other horsepower flow is infeasible because it cannot possibly be produced as a combination of locomotive flows.

We can use this observation to transform the original network problem with convex nonlinear cost function to a conventional linear network flow problem with integral upper bounds on the arc flows. This latter problem is easy to solve. The steps in the transformation are

1. Let b = GCF of the locomotive power ratings.
2. Transform the cost function by making it piecewise linear with breakpoints at multiples of b. See Figure 7. Note that the cost function remains convex and we change its value only at points that cannot be generated by any combination of locomotives.
3. Create n arcs (one for each division in the cost function) for each original arc in the network (Figure 8). (We do not need to add additional constraints because the cost function is still convex, thus the arcs will be loaded in the correct order).
4. Because our sources, sinks, and bounds are all multiples of b, we can scale flows down by a factor of b without losing integrality. The solution to this problem, when scaled back up, will be a multiple of b.

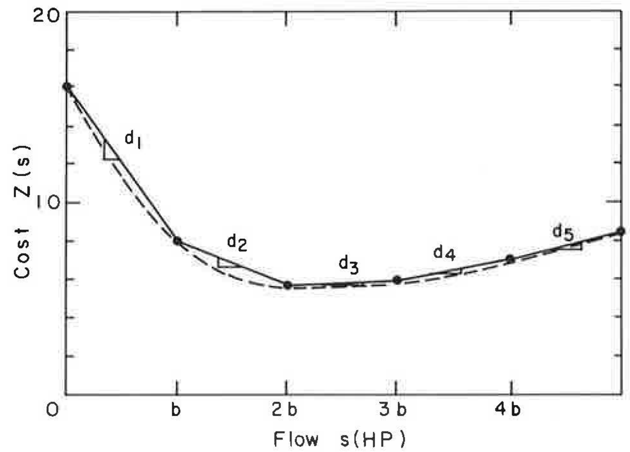

FIGURE 7 Piecewise linearization of the arc cost function.

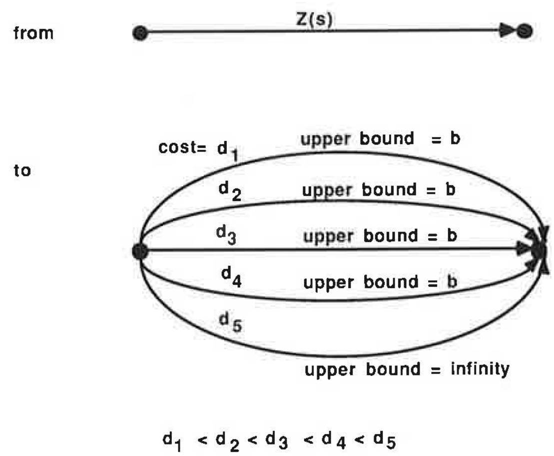

FIGURE 8 Transformation to a piecewise linear cost function.

5. We now have a conventional minimum cost network flow problem that will provide a valid lower bound on the original problem because (a) all feasible solutions to the original problem are feasible in this problem and (b) the cost function was changed only at infeasible points in the original problem. Note, however, that a feasible solution in this problem may not be feasible in the original problem. An example would be an arc flow that is greater than zero but less than the power rating of the smallest locomotive.

TEST PROBLEMS AND RESULTS

The heuristic was implemented in FORTRAN on a MicroVax II running MicroVMS 4.5 and tested on 19 problems of four time-space network configurations (Table 2). The smallest of these networks is shown in Figure 1, and Problems L1–L8 were drawn from an actual 3-day train schedule for the Grand Trunk Western Railroad.

The logistic form of the cost function was used in all cases. The power requirements varied from 3000 HP to 15,000 HP on the trip arcs, and locomotive supplies were fixed to be "barely adequate." The basic networks had $p/c = 10$, $c = 0.5$, $\sigma/\mu = 0.20$, but these were systematically varied in some of the test problems. Table 2 shows these parameters and results for the various networks. The heuristic-optimal values of the objective function behaved reasonably, with the following configurations having increased costs over the baseline (Problems S1, L1):

- Higher penalty term: Increasing the penalty term tenfold approximately doubled the objective function (Problems S3, L3).
- Higher standard deviation: Increasing σ/μ from 0.1 to 1 also increased the cost substantially (Problems S5, L5).
- Lower power supply: Because the initial power supply was "barely adequate," reducing it increased costs somewhat as more trains were underpowered (Problems S6, L6).

We would expect average running times to be a function of both the number of locomotives supplied and of the size of the network. In the cases here, as the networks became larger, running times seemed to be $O(nt)$ where n is the number of nodes and t the number of locomotives. They were reasonable in all cases, ranging from 0.5 sec for the smallest network to 63 sec for the largest. This is acceptable because it is envisioned that in an operating environment, the model will be run about once per 8-hr shift rather than continuously.

The numerical results were normalized to the best lower bound found, which was the GCF lower bound. The results of the heuristic were, on average, within 3 percent of this bound. These normalized results are shown in Table 2. Problems with a flat objective function (low p/c and high σ/μ – problems S2, S5, L2, L5) tended to perform better with results, on average, within about 1 percent of the lower bound. Conversely, problems with a highly nonlinear objective (S3, S4, L3, L4) give results that were on average only within 6–7 percent of the lower bound. The Frank-Wolfe algorithm also tended to have poor convergence on these problems.

FURTHER WORK

We have developed a model that deals explicitly with the uncertainty in power requirements. Moreover, the heuristic used to solve this model is promising because it is both fast and fairly accurate. Further research should focus on improvements to the heuristic and incorporation of schedule variability.

The present implementation of the heuristic does not optimize speed. Some improvements, mentioned earlier, include sending more than one unit at a time in the early stages of the heuristic, and keeping and reoptimizing a shortest path tree rooted at the supersink, rather than recalculating shortest paths for each iteration. Another improvement to the heuristic would be the incorporation of additional side constraints and provision for more than two commodities.

Of possibly greater interest is the incorporation of schedule variability. Although the model now assumes a fixed schedule, one way to do this would be to incorporate the heuristic into an interactive system that displays where and when shortages of locomotives are likely to occur, and then allowing the user to adjust the schedule accordingly before running the heuristic

TABLE 2 TEST RESULTS

Trial	Trips	arcs	nodes	kHP	p/c	σ/μ	GCF Cost	Normalized Costs greedy	int	FW	FWLB	GCF
T1	3	15	10	13	10	0.1	18.8	1.017	1.017	0.998	0.951	1
T2	3	15	10	13	10	0.1	18.6	1.069	1.028	0.991	0.981	1
S1	9	42	25	52	10	0.1	123	1.025	1.018	0.995	0.984	1
S2	9	42	25	52	5	0.1	105	1.016	1.013	0.999	0.991	1
S3	9	42	25	52	50	0.1	249	1.066	1.044	0.988	0.978	1
S4	9	42	25	52	10	0.05	105	1.053	1.047	0.994	0.881	1
S5	9	42	25	52	10	1	254	1.002	1.000	1.000	0.996	1
S6	9	42	25	37	10	0.1	184	1.038	1.000	0.999	0.995	1
S7	9	42	25	67	10	0.1	110	1.065	1.008	0.996	0.979	1
M1	35	153	88	137	10	0.1	1156	1.034	1.011	1.013	0.976	1
M2	35	153	88	168	10	0.1	1085	1.035	1.031	1.004	0.976	1
L1	102	404	239	283	10	0.1	1531	1.065	1.035	1.020	0.933	1
L2	102	404	239	283	5	0.1	1191	1.027	1.021	1.004	0.968	1
L3	102	404	239	283	50	0.1	3598	1.133	1.104	1.039	0.914	1
L4	102	404	239	283	10	0.05	1414	1.117	1.067	1.034	0.886	1
L5	102	404	239	283	10	1	2475	1.014	1.010	1.005	0.992	1
L6	102	404	239	253	10	0.1	1577	1.062	1.048	1.017	0.960	1
L7	102	404	239	309	10	0.1	1484	1.068	1.046	1.015	0.927	1
L8	102	404	239	406	10	0.1	1454	1.041	1.020	1.009	0.919	1
							average	1.050	1.030	1.006	0.957	1

Trips = number of trips in this network

kHP = total horsepower supply (thousands HP)

p/c = ratio of penalty to cost term

σ/μ = coefficient of variation for power demand

GCF Cost = Total cost (thousands \$) for GCF relaxation

greedy = total cost after GREEDY phase / GCF Cost

int = total cost after INTERCHANGE phase / GCF Cost

FW = total cost of Frank-Wolfe solution / GCF Cost

FWLB = total cost of Frank-Wolfe lower bound / GCF Cost

GCF = total cost of GCF relaxation / GCF Cost

run time = total running time for the heuristic, excluding input/output and computation time for relaxations.

again. However, to adjust schedules within the algorithm will require consideration of systemwide train scheduling and customer demand, both of which are very difficult to quantify.

ACKNOWLEDGMENT

This research was funded by the Grand Trunk Western Railroad.

REFERENCES

1. A. A. Assad. Models for Rail Transportation. *Transportation Research*, Vol. 14A, 1980, pp. 205–220.
2. A. A. Assad. Analytical Models in Rail Transportation: An Annotated Bibliography. *INFOR*, Vol. 19, No. 1, 1981, pp. 59–80.
3. E. R. Peterson. Operations Research and Rail Transportation. In *Scientific Management of Transport Systems* (N. K. Jaiswal, ed.), Elsevier North-Holland, New York, 1981, pp. 37–51.
4. T. E. Bartlett. An Algorithm for the Minimum Number of Transport Units to Maintain a Fixed Schedule. *Naval Research Logistics Quarterly*, Vol. 4, No. 2, 1957, pp. 139–149.
5. R. S. McGaughey, K. W. Gohring, and R. N. McBrayer. Planning Locomotive and Caboose Distribution. *Rail International*, Vol. 4, 1973, pp. 1213–1218.
6. M. Florian, G. Bushnell, J. Ferland, G. Guerin, and L. Nastansky. The Engine Scheduling Problem in a Railway Network. *INFOR*, Vol. 14, No. 2, 1976, pp. 121–138.
7. J. M. P. Booler. The Solution of a Railway Locomotive Scheduling Problem. *Journal of the Operational Research Society*, Vol. 31, 1980, pp. 943–948.
8. Y. Sheffi. *Urban Transportation Networks*. Prentice-Hall, Englewood Cliffs, N.J., 1985.
9. J. A. Ferland, M. Florian, and C. Achim. On Incremental Methods for Traffic Assignment. *Transportation Research*, Vol. 9, 1975, pp. 237–239.
10. S. E. Mimas. *Equilibrium Traffic Assignment Models for Urban Networks*. Master's thesis. Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, 1984.