MODELLING TRACK MAINTENANCE AND ITS EFFECTS ON THE RELIABILITY OF A SINGLE TRACK RAILROAD LINE

by

John F. Romps III

Bachelor of Civil Engineering
Catholic University of America, 1991

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in Transportation

at the
Massachusetts Institute of Technology

May 1993

Signature of Author ______________________
Department of Civil and Environmental Engineering
May 13, 1993

Certified by ______________________
Carl D. Martland
Senior Research Associate
Department of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ______________________
Eduardo Kausel
Chairman, Departmental Committee on Graduate Studies
Department of Civil and Environmental Engineering

MODELLING TRACK MAINTENANCE AND ITS EFFECTS ON THE RELIABILITY OF A SINGLE TRACK RAILROAD LINE

by

John F. Romps III

Submitted to the Department of Civil and Environmental
Engineering on May 13, 1993, in partial
fulfillment of the requirements for the degree of
Master of Science in Transportation

Abstract

Track maintenance is necessary for the operation of a safe, reliable line-haul. When maintenance is not performed track condition deteriorates. This leads to track failures and the imposition of slow orders, both of which result in train delay. The decision as to when to replace track components is based in the pocketbook. Rail, ties or ballast should be replaced when the cost of train delays and spot maintenance exceeds the costs of renewal maintenance. The action of taking a portion of a line out of service for any maintenance activity in order to effect repairs also leads to train delay which effects line-haul productivity.

In this thesis, a methodology is developed for evaluating the effect of track maintenance on train reliability. A maintenance planning model is used to predict the annual project maintenance requirements and track defect rates for a given maintenance policy and traffic level. A line simulation model was modified to accept track failure rates and maintenance activities as inputs. The result is a means of predicting track maintenance requirements that arise from different traffic levels and the reciprocal effects that these maintenance requirements have on the traffic.

A study was performed in which maintenance requirements on a line were predicted for normal and heavy axle loads under moderate and heavy traffic conditions. It was found that largest factor in determining the extent of train delay is traffic density. Average train delay was twice as much for trains operating on the high density line as for those on the low density line. The effect of heavy axle loads on track is an increase in required maintenance, and a subsequent reduction in line capacity. However, fewer trains are required to transport the same net tonnage of cargo. The increase in train scheduling flexibility is offset by the decrease in line capacity due to increased maintenance, and no net change in service level results.

Thesis Supervisor: Mr. Carl D. Martland
Title: Senior Research Associate

ACKNOWLEDGEMENTS

The work described in this thesis was made possible through the generous grants provided by the Association of American Railroads in their ongoing support for the MIT Affiliated Laboratory at the Center for Transportation Studies.

To Carl Martland, his continued support and patience were just as important for the completion of this thesis as the knowledge and experience he shared. He is also one of the few people in my personal experience who has been a truly suffering Red Sox fan longer than myself.

To Nan, for putting up with the continual delays and late night phone calls.

To my dungeon-mates and all those other rail-group wanna-bees: Bis, Edub, The good doctor, the humble scribe, kai-CHIANG!!!, Rammer, Nik, the Soeldmeister, Emily, Cammille, and Big Dan Turk.

This thesis is dedicated to my parents. They have supported me in every endeavor, and encouraged me to perform my best at whatever I do. I have them to thank more than anyone else for making me the master's recipient I will be in two weeks time. I am proud to be their son.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Objectives

Track maintenance is a fact of life for a productive railroad. Whether it is the repair of a defect on the line, or the replacement of old, failure prone materials, before defects occur, track maintenance has to be performed or eventually the railroad will not be able to perform its function. A standard definition of transportation is "the safe, efficient, economical movement of goods (and people) from one place to another". When track is in need of repair it is not safe; trains will not be able to travel at full speed over the poorer sections, so it is inefficient; and delays to freight mean lost money, so it is not economical. The objective of this thesis is to develop a means of evaluating the effects that maintenance activities have on the trip time and reliability of trains. This is accomplished by using a track component deterioration model to predict maintenance requirements and track failure rates. This information is used as inputs for a line-haul simulation model which determines what effect it will have on traffic.

In order to run trains along a single track route in a reliable fashion it is important for a railroad to be able to predict possible delays that the trains could encounter. Trains are held for operations and for engineering reasons. Engineering delays result from equipment failures and maintenance of way. There are two types of equipment failure which can interfere with train operation on a route: failure of a train or failure of the infrastructure. The occupation of a portion of a line by maintenance of way crews will have a disruptive effect on traffic, although the disruptive effects can sometimes by reduced by scheduling the work between trains. Operations delays are primarily dispatch decisions in which two trains meet on a single track line and one must be held while the other continues, or when a faster train overtakes another train and the slower train is held on a siding so it can be passed. Delays due to failures can propagate down the line in a cascade effect resulting in an increase in train meets, which lead to still more delays which are perceived to be operations delays even though they resulted from an equipment or track failure.

There are no set guidelines for determining operations costs of maintenance projects beyond the direct costs of acquiring the necessary materials and performing the work. Life cycle costing when combined with deterioration models can predict these costs. A simulation model can be used to predict costs incurred by trains that are delayed by scheduled maintenance work being performed on the line and by track failures which occur (randomly). The cost of train delays which result from different maintenance of way decisions can be used for a more thorough cost\benefit analysis of different maintenance policies. The combination of life cycle costing and simulation modelling can contribute to the important decision of when maintenance projects should be scheduled.

The approach taken in this thesis consists of modifying a line haul simulation model to include both planned and unplanned track maintenance.

Another model (TRACS, Total Right-of-way Analysis and Costing System) which combines engineering deterioration and life cycle costing for track components will be used to determine the maintenance requirements and ensuing costs for the line in question. It is entirely feasible that a railroad which keeps detailed enough records could simply empirically determine replacement cycles and failure rates for its infrastructure and use them in the simulation, or that another model or combination of models could be used instead of TRACS. The reasons for using TRACS will be detailed in chapter 3.

The characteristics of the line, including the composition of the traffic and the quality of the track components, in addition to a given maintenance policy will greatly influence the deterioration rate of the track components as well as the rate at which failures occur. The models will be used to determine the relationship between planned and unplanned maintenance and the trip time reliability of a single track line. Figure 1.1 depicts the proposed methodology combining the maintenance model and the simulation model to come up with statistics reflecting the reliability of the line.

1.2 Background

1.2.1 Service Reliability

Possible definitions of line reliability include the following: (Martland et al., 1991)

a. The ability to meet a particular train schedule

b. The variability in train arrival times

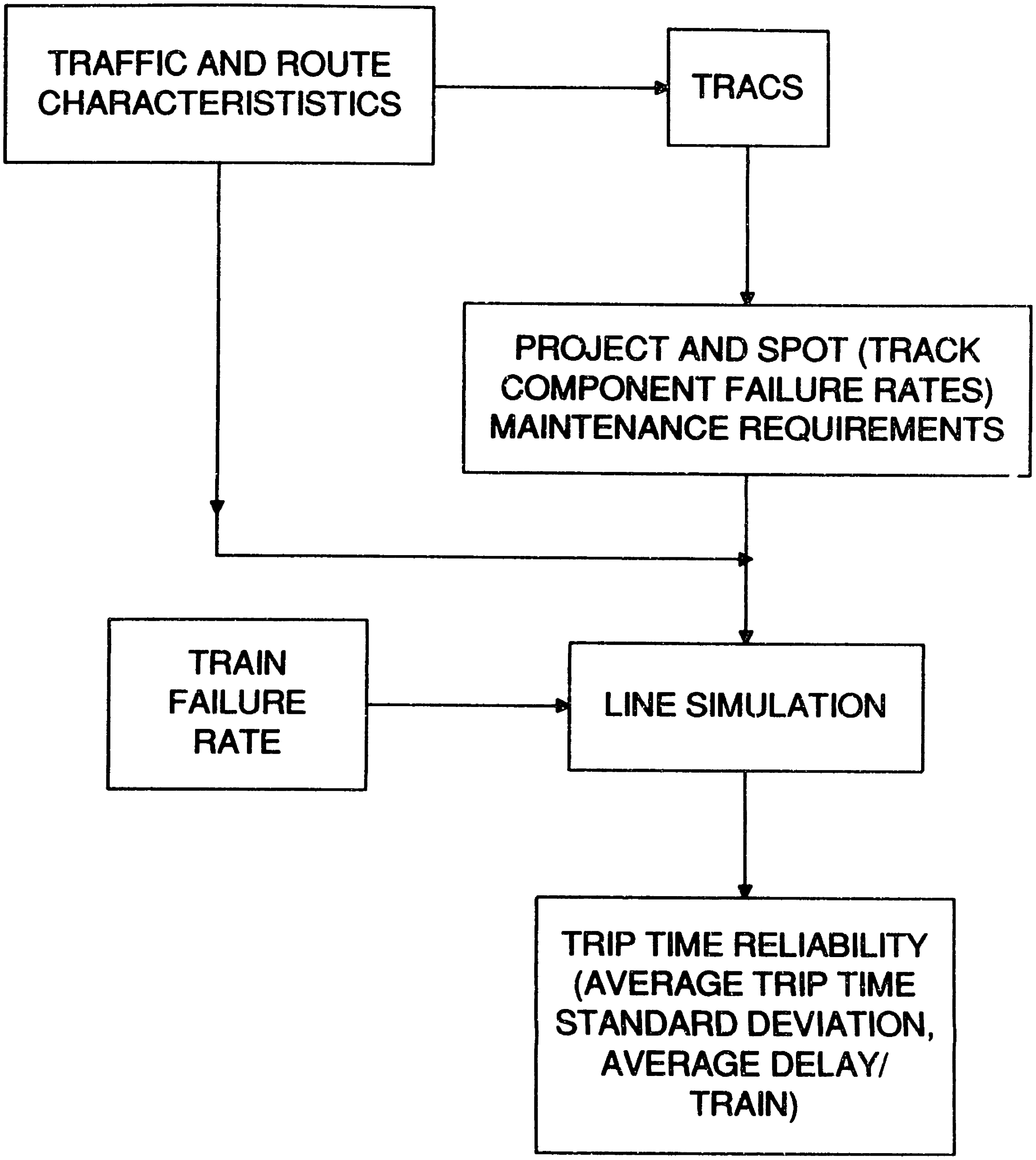


Figure 1.1: Methodology for combining TRACS and a line-haul simulation model

c. The variability in train travel times

While not exactly the same, they are clearly related. Adherence to a schedule means arriving on time, and in order to set up the schedule in the first place the train's expected travel time must be known. Variability in travel time translates into variability in arrival, and unless there is enough slack built into the schedule, this variability will translate into late arrivals.

When it comes to attracting customers, railroads have one major advantage over the trucking industry; railroads can move large quantities of goods long distances much more cheaply than a trucking company can. The advantage that trucks have is the speed and reliability of their service. Manufacturers, seeking to reduce inventories, are no longer satisfied with 12-hour delivery windows and need transportation service that is closely coordinated with production schedules. The rail industry therefore recognizes that cost-based competition will be inadequate for the long run, that railroads must meet and exceed the competition in order to survive and to prosper (Martland et al., 1991). If a shipper is has a large enough safety stock, a few extra days in transit for most non-perishable goods will not mean a large increase in cost. It is only when goods are not delivered when expected that costs begin to rise rapidly. Even within the railroads themselves reliability of travel time is perceived as more important than length of travel time. A train that is five hours early can sometimes cause as many problems in a congested yard as a train five hours late.

1.2.2 Engineering Reliability

In-transit train failures and track failures are a significant cause of increased delay experienced by trains. This is due to the fact that it is not only the train that experiences the failure firsthand that is affected. On a congested line, the delay experienced by one train could "cascade" down the line to other trains, which might not encounter the defect firsthand. In addition to the cascade effect, during the time that a train is delayed on a line due to a failure of some kind, there is more time for other trains to enter the line, and for the other trains that are already on the line to change their position. All of this will lead to an increase in meets and passes, and the necessity for dispatchers to create new meet/pass plans. Dontula (1991) found in a simulation case study of train failure rates on a generic line that "...on lines operating close to capacity, it should be realized that the operations delays such as meet/pass delays increase because of occurrence of engineering failures. For the lines operating not so close to capacity, the engineering delays are about 50% of the operating delays and about one third for the total delays."

Train failures include occurrences such as locomotive breakdowns, brake failures, and coupler failures. Track failure can result in either the shut-down of the track in the immediate vicinity, or the imposition of a slow order (a mandatory restriction in speed over a section of track in order to avoid further damaging the track) and derailments when the damaged train travels across the damaged portion of track. Examples of track failures are the development of rail defects, track geometry defects, a clustering of broken ties, fouled ballast, and signal failures. Usually only a broken rail or a signal failure will result in halting traffic on a line. Other problems will allow travel at reduced speed, although it is important to get a spot maintenance gang out to the problem area as fast as possible in order to clear up the problem and get traffic moving at full speed again.

1.2.3 Maintenance Of Way (MOW)

The engineering department of a railroad is responsible for maintaining and upgrading the infrastructure. Budget constraints will often necessitate trade-offs when scheduling maintenance. One of these trade-offs is spot vs. planned maintenance. Proper planning of MOW work can actually prevent a large percentage of track failures from ever occurring. MOW work is ideally scheduled whenever it becomes more cost effective to replace the component than it would be to keep performing spot repairs whenever a defect is detected. This is not an ideal world, and there are a number of reasons, ranging from work crew conflicts to lack of funds, which will result in the postponement of maintenance work. The second trade-off is planned maintenance vs. line operations. This is the area in which this thesis makes a contribution.

The ability to plan ahead for MOW projects is important for capital budgeting. For example, if it is known that the rail over an important section of track is expected to reach wear limits in five years, the railroad should make sure that it will have the money on hand five years from now to purchase the new rail.

When up to 15 percent of every dollar spent at most major railroads is spent on maintaining the railroad (Samuels, 1993), it is vital that the railroad manage its capital expenditures as well as possible.

MOW projects also have an impact on train trip times. An MOW gang can be on the line anywhere from 4 to 10 hours at a stretch. The large size of these projects makes scheduling of utmost importance. The time between trains when a crew can get out on the line is referred to as a maintenance window. It is up to dispatcher to give a maintenance crew clearance to move out onto the line. On highly congested lines it is sometimes impossible to schedule maintenance when it will not have any effect on traffic. In cases such as this certain trade-offs must be considered.

> The main difference between trains and work gangs is that the activities of the latter are much less time sensitive; i.e., the value of the output of track maintenance gangs often remains constant when the completion time of the activity is shifted a few hours later or in advance, while the same is not true for trains. The only direct costs associated with track maintenance that are time sensitive are those connected to labor and equipment utilization, and any extra labor costs for overtime and nighttime work. These costs are often not high enough to justify the hours of lateness caused to high priority trains, but they may be high enough to give higher priority to the track maintenance gang than to a low priority coal-hauling train. Thus train dispatchers should have the final say regarding the exact time windows assigned to the track maintenance gangs, and they should be able to assess the effects that this track outage will have on the performance of the trains transiting the line in the vicinity of the maintenance gangs (Harker and Jovanovic, 1990).

The most common types of MOW projects, and how they are planned and the costing concerns that go along with this will be discussed further in chapter 3.

1.3 Motivation

The two reasons for developing this research are the increasing importance being placed on trip time reliability and the need for a means of evaluating the cost of maintenance plans beyond the obvious capital costs for equipment, machinery and labor. The significance of trip time reliability is related in the following quote:

> Customers demand standards for delivery times,, they expect performance to meet these standards, and they seek carriers who can provide the desired level of service. The trucking industry's ability to make deliveries on schedule explains their continued inroads into rail traffic and puts much of the rail industry's general merchandise and intermodal traffic at risk (Martland et al., 1991).

Maintenance budgets for railroads amount to about 15% of every dollar spent, which in a 30 billion dollar industry amounts to a lot of spending potential. It is important for the rail industry to get the most out of the money spent. In the 1970's when money got tight, the first department to have its budget cut was maintenance. Now that the rail infrastructure has been brought back up to standard, it is important to keep it in good repair. It is also important that when performing the necessary preventive maintenance and upgrading that the railroad endeavor to balance the costs of maintenance activities , both planned and spot, with train delay costs, including effects on trip time reliability. British Railways has also noted a need for development of a model for the study of line-haul reliability. "The link between the reliability of individual components and sub-systems, and their effect on the performance of the train service operating on a route, however, is an area in which we identified a gap in understanding." (Dunkerley, 1992)

In the increasingly competitive freight transportation industry, there is always the opportunity to handle more traffic from customers. Increased traffic means either more trains on the line or heavier trains. Either of these will necessitate more maintenance on the infrastructure. In order to stay competitive, the railroad industry will have to provide reliable service. The methodology combining track deterioration and line-haul simulation models developed in this thesis will provide a valuable tool for railroad management in making decisions which will have an impact on future maintenance policy.

1.4 Research Approach

1.4.1 Definition of Problem

This section describes the desired characteristics that a good modelling system should have:

1. The maintenance planning model will ideally predict the amount of planned and unplanned maintenance that will be necessary over a line in a given year. The characteristics of the line such as traffic, initial conditions of materials, and a maintenance policy should be easy to adjust. The ability to change the maintenance policy will allow the user to defer or move up maintenance projects.

2. The line-haul simulation model should not depend on which maintenance planning model is being used. As long as a project maintenance plan and a train schedule are available, the model should allow for easy transfer of the data into a format that can be used for simulation inputs. It is also desirable for easy conversion of unplanned maintenance events to probabilities of failure.
3. The simulation should be flexible. It should make it easy for changes to be made in the project maintenance plan, as well as in the probabilities for failure of the different track components which will generate the spot maintenance events. The user will be able to make length and time of individual events without effecting the scheduling of other events. It should also accept train schedules and allow for different travelling speeds of trains and different maximum speeds along segments.
4. The final output, from the simulation model, should provide the user with average trip times and the variation of trip times for different priorities of trains. It should also return the average delay to trains of different priorities. It is important to consider that since trains can have different travelling speeds, the average delay will not necessarily increase linearly with the average trip time, if all trains are considered.

5. The model should also allow the user to determine the effects of individual planned maintenance events.
6. A cost for a given project will be part of the output of both models. The maintenance model would ideally determine the cost of materials, and labor. The simulation model gives the user the opportunity to attach a cost to delays experienced by trains as a result of the maintenance activities. The additional time that a train is made to wait due to maintenance events can be assigned a dollar value. The total cost of a project plan can be used to compare it to other plans. This will allow the user to pick the best plan.

1.4.2 Approach to Solution

This section describes the direction taken in modifying the simulation model and the role played by TRACS.

1. The track maintenance life-cycle costing model TRACS was chosen to determine maintenance requirements. It has separate input files for route, maintenance policy, and traffic. This will make the performance of case studies much easier.
2. TRACS predicts spot maintenance requirements as well as project maintenance. The spot maintenance events will be used to incorporate the element of random track failures into the simulation model.

3. In his thesis, Dontula (1991) developed a line-haul simulation model for the purpose of studying trip time reliability of a single track line. Among other outputs, the model can determine the average delay per train, and standard deviation of trip time. The model has been modified to allow the placement of planned and unplanned track maintenance activities on the line.
4. The simulation model calculates average trip time and delay data for all trains after steady state has been achieved. In order to determine the effects of individual maintenance activities, the simulation model will also calculate this data for trains that arrive at their destinations within a given window of when the maintenance event took place.

Chapter 2: Maintenance Practices and Policies

2.1 Reasons for MOW: Goals and Objectives

The reason for performing maintenance and upgrading is to keep the track in good working condition. This statement begs the question of what is "track in good condition?" A simple answer is "track that allows safe and efficient travel by trains with an acceptable amount of delay". Knowing this, the railroads must determine when track does not meet this definition.

There are three approaches to evaluating the condition of track. The first is to use a set of rules or equations to assign what is known as a quality index to the track components. Some times, individual quality indices for rail, ties, and ballast will be combined to return one general quality index. The other approach is to set physical limits for the track components. For example, whenever the rail head is worn down 0.65 inches, it is time to replace that rail. The third method is to use a time interval as a quality standard (Love, 1981). This is probably the least reliable of the three. It assumes that after a set period of time, the component will be deteriorated enough to warrant replacing it. Whatever method is used, the track is inspected, and a decision is made on whether or not to take immediate action on that segment of track.

There is more to deciding when to perform maintenance than determining the condition of the track components and deciding which components need attention in the near future. There is a much larger picture to be looked at. There are economies of scale that can be achieved if the future condition of all track components is determined. For example, it might be advantageous to resurface the line a year early if the ties need to be replaced during the current year. Knowing what actions will become necessary in future years will allow the railroad to make sure that it will have the necessary materials on hand. A large part of having the materials is having the money to buy the materials. Capital budgeting is a major impetus for performing deterioration analysis.

Maintenance needs can only be estimated by considering the physical condition of the track structure and of the expected condition following maintenance work. An important consideration is the impact of the maintenance on trip quality. If performing the maintenance immediately will only allow an increase in train speed of a few miles per hour, it might not be cost effective to perform that maintenance action as a first priority. It is also possible that a section of track could be in such poor condition, that delaying maintenance will only cost the railroad money in the long run, due to delays and lengthy trip times.

All track maintenance and construction work falls under the responsibility of a railroad's engineering department. Maintenance crews generally get onto the line in one of two ways, either through large scale production work or basic day to day maintenance. The day to day crews are responsible for such activities as replacing bolts, anchors, and joint bars, performing spot rail and tie replacement, repairing derailment damage, clearing roadway, brush and ditches, and spot surfacing, lining, and gaging track. Production gangs limit their activities to tie replacement, rail replacement, and surfacing, but perform them in much larger quantities. (Love, 1981)

2.2 Types of Maintenance and Work Gangs

"Loads from passing trains, weather, age and environmental forces damage track components and the track condition worsens from its as-constructed state as the damage accumulates. Rail gets worn, bent, and fatigued, ties split and rot, ballast gets crushed, and track geometry deviates from normal. We define the sum of all accumulated damage to date as deterioration."(Love, 1981) The following sections describe typical maintenance activities. The track components of interest are rail, ties, and ballast.. These sections are included to provide the reader with a better idea of what is involved in typical maintenance projects.

2.2.1 Rail

2.2.1.1 Jointed Rail

Rail can be replaced at or prior to failure. The first step in replacing rail is distributing new rail to the work sites. This will be done anywhere from several weeks to several days before the project is to be performed. A jointed rail is removed by taking the track out of service, pulling spikes, unbolting the joint bars at each end and lifting the rail out. Replacement is the opposite procedure. Production gangs on the Class 1 Railroads use a crane to distribute the rail from a work train. Other equipment, of a major relay gang, includes spike pullers, material pickup carts, speed swings, adzers, diggers, cribbers, tampers, spikers, anchor applicators, pickup trucks, fuel trucks, dump trucks, and crew transport trucks with a gang of 30 men. Other gangs may use other types of machinery as well. Spot maintenance gangs use much less equipment which is generally less mechanized. With hand tools (spike pullers, spiking maul, etc.), they change out single defective rails.

2.2.1.2 Continuously Welded Rail

Jointed rail is used less and less as the years go by. "Most rail wear occurs at the joint and is, in turn, the forerunner of many other track and joint ailments--worn joint bars, increased tie wear, loosened ballast and pumping joints, and possible general track deterioration." (Hay, 1982) Continuously welded rail (cwr) is the choice of railroads today. Jointed rail would be fabricated in lengths of about 39 feet, just short enough to fit inside a boxcar for transport. Lengths of cwr are typically 1/4 of a mile and must be transported to the project site on special trains. The long ribbons of cwr allow for the use of super-machines when laying rail. The P811 actually lifts out the old rail and places the strips of new rail on the track for installment. This machine can actually relay ties while moving over the track. A welding crew, spiking crew, and tamping crew will follow the machine, just like with jointed rail.

The typical field welding process, which eliminates the use of joints, used by Class 1 railroads is called thermit welding. This process involves clamping the two rails 1/4-3/4 inches apart; heating the ends of the rail with a flame jet until red hot; and using the chemical reaction between iron oxide and metallic aluminum oxide to generate enough heat to melt steel into the gap. A skilled foreman with 12-18 helpers can make 30-50 fusion welds per 8-hr day. Other methods of welding continuous welded rail include electric flash welding which uses electrical heating for fusion of the rail ends, and oxyacetylene pressure process (Hay, 1982).

When performing spot maintenance on cwr, conditions sometimes will not allow the welding of the new rail in place. These conditions are cold weather and a tighter schedule due to the increase in defects that accompanies the colder temperatures. When a defect is discovered in the winter, the defective rail is cut out, a "plug" is situated and held in place with a temporary joint or "patch". In the summer the temperature will be more amenable for performing welds, and spot crews will have the time to go out to complete the winter patch jobs.

2.2.1.3 Grinding

Grinding is performed to remove excess metal after welding, to remove corrugations when they occur in rail, and to maintain a usable rail profile.

The Mt. Newman Mining Company employed [in 1982] two 28 stone grinders year round to control rail surface distortion in 253 miles of mainline track. "As a result, plastic flow distortion, contact fatigue, and the [problem] of sinusoidal wear [were] controlled to a high degree and [were] rarely the cause of rail renewal." (Roney, et al., 1982)

Grinding away corrugations and irregularities before they have a chance to grow effectively reduces the number of defects which develop into operating problems. This is why railroads which run trains with heavier axle loads, with the increase in stresses in rail, implement more aggressive grinding policies. An increase in grinding effectively increases the head wear rate in the rail being ground, but this does not necessarily reduce track life. It is the ability of the rail surface to support traffic, and not the amount of rail which determines when it should be replaced. On one railroad, it was found that with proper "profile grinding...condemnable wear limits can be increased" (Roney et. al, 1982). Another benefit of grinding is fewer rail defects. This translates to fewer traffic disruptions such as down time while a spot crew is repairing a defect on the line, or slow orders that are set until the crew can get out to the problem section.

2.2.1.4 Other Reasons for Rail Maintenance

> Rail maintenance is also performed to replace broken or badly worn joint bars and to tighten bolts at joints. Another rail operation is transposing rail on curves to increase its useful life. Transposing rail involves exchanging the high rail on the outside of the curve with the low rail on the inside. The procedure for doing this is the same as that for replacing a rail except that the material is already on hand and there is no scrap rail to remove (Love, 1981).

It is sometimes advantageous for a railroad to remove rail before it wears out or develops too many defects. This "good" rail would then be "cascaded" to another section of the railroad where the traffic is not as heavy, thereby adding several years to the life of the rail. This allows the railroad to use new rail on mainlines rather than on branch lines or in terminals. The economies of scale that are realized make the project economically feasible even though it involves using relay crews more often than usual. Two arguments for the viability of cascading rail are given below:

> First,...there is still a variety of lines to be maintained. These include shortlines, branchlines, and yard mileage.
>
> A second reason for cascading is that rail manufacturing technology is still improving. [It might be economically feasible to place the latest high quality rail on the main lines, thereby releasing the rail that was previously on the main line for use on other track (Staplin and Wells, 1987).

2.2.2 Ties

Determining when to replace ties is not the same as when to relay rail. It is possible for track to provide good service with a number of failed ties still in service. A single failed tie, unlike a rail defect, does not necessarily require a repair crew to be dispatched as soon as possible. Failed ties only become a problem when clustering occurs. A cluster is generally defined as three or more ties immediately next to each other. The impact of failed ties on surrounding ties must also be considered when deciding which ties to replace. When a tie fails, part of the workload that it used to support is transferred to the ties immediately next to it. This makes these ties more likely to fail than other ties in the system. The greater workload will make them wear out and deteriorate at a faster than normal rate. Hence clusters tend to form. This effect must be taken into consideration whenever a maintenance policy is determined, and the thresholds at which ties will be replaced are set. "...a single bad tie or missing tie does not materially affect lateral track strength...It is the clusters of three or more failed ties which pose the greater risk of derailment and track, equipment or lading damage." (Davis and Chow, 1989) For this reason a railroad may put off replacing single failed ties until there are enough to allow the activity to achieve economies of scale, by performing the renewals in groups. One measure of when to send out a crew is to predict how many failed ties in a row will restrict travel at normal speeds. Whenever this threshold is reached it is necessary to perform spot replacements of the cluster of failed ties. Love (1981) described another policy of when to replace ties as whenever the system average, of number of sound ties, falls below a certain percentage. For example, whenever one hundred or more ties per mile are bad, all failed ties must be replaced.

The process of replacing ties is traditionally executed by large gangs using several specialized machines and operators and additional laborers. The use of supermachines, such as the P811, to perform these same tasks is becoming more common. Smaller gangs are also used for jobs in which there are not many ties to be replaced, or when the failed ties are spread out over large distances. No matter what machinery is used or how big the job is, the steps to be followed are the same.

> Spike pullers remove spikes from ties to be renewed. [The ties are removed from the track]. Other follow behind, cleaning out space i the ballast shoulder to make entrance room for the new tie and inserting it. Tie plates are replaced and respiked. Another group follows behind, tamping any loose ties.

2.2.2.1 Tie Disposal

"A small number of ties removed from track for reasons other than complete wear may offer economical reuse in secondary, yard, and side tracks, but the majority face disposal." (Hay, 1982) It used to be common practice to cut the old ties into pieces for easy removal. The preservatives used in the ties precludes using them for firewood, and there is limited use for ties that have been cut to facilitate removal. Previously the only methods for disposing of ties was landfilling or incinerators. The latest development in tie disposal is to send whole ties to cogeneration plants. However, recycling plants won't take ties that have been cut. In today's environmentally conscious world, it is becoming common to recycle the ties rather than cutting them up and sending them off to a landfill. At least one major North American Railroad is under contract to supply used ties to such a plant.

2.2.3 Ballast

> Ballast performs a number of functions as part of the track system. It serves to distribute the load from the super-structure to the sub-grade, provides lateral and longitudinal stability, absorbs the impact of passing loads, provides a means by which surface and alignment can be corrected and maintained, protects the super-structure from the elements and provides drainage for the super-structure...For ballast to perform these functions, it must remain free draining (Matthews, 1990).

"The continual repetitive application and release of wheel loads causes the degradation of ballast materials, permanent settlement, and loss of surface and line." (Hay, 1982) Degradation of materials refers to the accumulation of dirt and crushed ballast which can impede drainage and affect stability. Surface and line are the longitudinal and lateral support, respectively, provided by the ballast.

Restoration of surface is achieved by physically raising the track and inserting ballast wherever it is needed. The ballast is then "tamped" to compact the ballast thereby increasing its density, and increasing its strength.

> An alternative to tamping [in surfacing operations], developed by British Rail is stone injection. [It] is the process of surfacing by raising certain "low" ties and placing a measured amount of 1/2 to 3/4-inch stone under them by blowing the stones through a tube...Stone injection has been shown to last three times as long as tamped track before returning to its pre-maintenance profile roughness (Chrismer, 1991).

The advantage of tamping over stone injection is that it can be performed much faster using either machines which tamp as they pass over the track, or hand tampers for small jobs. Tamping does not require stopping and inserting a tube into the ballast.

Lining can be performed by combination tamping/lining machines, undercutting/lining machines which lift the rail to facilitate cleaning of ballast, and large tamping machines. It can also be performed by gangs using plows. (Hay, 1982)

The usual method of cleaning ballast is through undercutting.

> "Undercutting cleans the entire ballast section to some predetermined depth. An endless chain or belt is 'dug in' transverse to the track at the required depth. Buckets on the chain scoop the ballast and lift it to vibrating screens or grates as the machine advances. Clean ballast is restored to the track and dirt is wasted on the shoulder or carried by conveyors to dirt cars (Hay, 1982).

Tamping is required to provide compaction of the ballast after any cleaning operation.

2.2.3.1 Surfacing and Slow Orders

Surfacing of track is also performed after tie insertions and rail relays. The only problem with this is that a slow order must be implemented until the ballast settles. One alternative to lengthy slow orders was reported by Tuzik ("Looking at...", 1991) from and interview with, then vice-president of engineering at Conrail, Richard Pyson.

> All of Conrail's surfacing gangs are equipped with CAT (Continuous Action) tampers and stabilizers. The primary reason for using the stabilizer is that it dramatically reduces the amount of time that the track is under a slow order. [Conrail] previously accomplished the stabilizing effect by running a certain number of trains over the track at varying speeds until [they] moved a given amount of tonnage. On a mainline, six or eight trains can accomplish this...the stabilizer and good record-keeping allow trains to run with fewer delays, which helps provide better customer service (Tuzik, "Looking at...", 1991).

2.2.4 Combination of Projects

There are times when two major components of a track section will require replacing within a year or so of each other. When this occurs, the economies of scale that can be achieved by combining the two jobs will often outweigh the savings of leaving one of the components in the track for the extra time. Mishalani (1989) describes two projects that could potentially be combined:

> Consider the following two activities: rail replacement and tie replacement. very briefly, the replacement activities include removing the spikes, moving the rails to the sides of the track, placing the new rails - which are already transported from the plant and are placed on the sides of the track - onto the ties, spiking the ties, and re-surfacing the ballast. Concerning tie replacement, the components include removing the spikes, moving the rails to the sides of the track, removing the ties, placing in new ties, moving the rails back to their position on the new ties, spiking the ties, and re-surfacing the ballast.

It is easy to see that there are several activities common to both projects. If the opportunity arises to replace both ties and rail at the same time, all of these common activities would only need to be performed once. Fewer tasks to perform not only means less time on the track, but also fewer crew members to be paid while on the track. It is also true that the supermachines in use today can be used for more than one task. The rental costs of these machines is high, and opportunities to use them for combination jobs should not be passed by.

2.3 Scheduling of MOW

As mentioned earlier, maintenance activities can be divided into project or spot maintenance. Maintenance projects are scheduled to take place at a set time. The purpose for projects is to sustain a safe travelling surface free of defects. The decision of whether or not to schedule a maintenance project has a lot to do with the number of defects that can be expected to develop if the maintenance is not effected. While planned maintenance is preventive in nature, spot maintenance is reactive. Spot maintenance crews tend to be smaller and have to ability to get the job done without occupying the line for long periods of time. This section deals with the scheduling of maintenance projects. Although it is possible for a railroad to be able to estimate the number of spot maintenance activities that will be necessary during a given period, the only scheduling that is performed ahead of time, is to have the necessary repair crews, equipment, and materials on hand. As will be seen, there are several factors that go into the scheduling of maintenance projects.

2.3.1 Determining Need

Before the scheduling process can begin, the engineering department of a railroad must first determine what sections of track will be approaching their deterioration thresholds in the near future. There are two sources of information from which track condition can be ascertained. The first is records of the materials that are in place and the tonnage which has passed over the track since the installation of these materials. Other information which is relevant to deterioration of track materials are: "train speeds, axle loads, subgrade and ballast support, grades, curvature, weather and climate, and the level of prior maintenance." (Hay, 1982) Using these records in deterioration models will result in estimates of the amount of wear the track components have experienced. This approach is discussed in chapter 3. The other method of determining track condition is to inspect it. Deterioration models can only return generalities. They cannot predict the exact location of rail defects, broken ties, compacted or crushed ballast, and rail alignment problems.

> Track inspection takes many forms. On-the-ground inspection will be performed by section, subdivision, and division supervisors. whether a walking inspection or one with a motorcar (making frequent stops), the track level, track gage, pocket rule, taper gage, and straightedge should be used to check curve alignments. Cross-sectional profiles taken with a rail-profiling device give a measure of rail wear. the recommendations of local work forces should receive consideration.l Engine crews can always be counted on to point out locations of rough track. Their reports can be verified by on-the-ground inspection and by supervisory personnel riding with the enginemen. Detection of rail defects by detector cars [is another option] (Hay, 1982).

One answer to the need for an objective method of evaluating several key rail parameters is the track geometry car. "These cars evaluate the following: cross level, superelevation, profile, warp, gage, and curvature. Departures from the geometric ideal or some other established norm give evidence of the need for maintenance effort." (Hay, 1982) Rail defect detection cars have also been developed. One type of detection car uses electric induction to locate distortions in the rail. Another car magnetizes the rails, and then locates fissures by the magnetic fields they generate. The last type of automatic detection car reported by Hay (1982) passes ultrasonic sound waves into the rail which are reflected back to a receiver. Defects are indicated, as in the other cases, by departure from the usual pattern.

Once the needs for a region, over a set time period, have been determined, the work programs must be scheduled. Determining which projects to undertake depends on a variety of factors, not the least of which is the location of the project. Large North American Railroads are divided into regions. Each region will have its own maintenance needs, and each deserves its share of the money spent on maintenance annually. A major part of this scheduling process is assigning priorities to each project, and determining the costs of the projects. Put simply, this consists of deciding which projects need to be scheduled during the current maintenance cycle in order to avoid operations problems on the line. In reality, this is an iterative process which passes from the smaller divisions of a railroad to upper management and back, until a balance is achieved among priority jobs, budgets, and regional preferences.

Constraints that must be considered are budget, time, labor, technology, and availability of materials. It is easy to decide that ties need to be replaced over a three mile section of track. The hard part is getting work crew, machinery, and ties to the site, and paying for all of it. The logistics of getting these elements there simultaneously will be discussed in the following sections. It is important that before the maintenance season begins, the railroad have the necessary equipment, personnel and repair materials on hand, or at the least know that it can acquire them before the projects are to be implemented. One reason that planning several years ahead is important is making sure that enough rail, ties and ballast are produced for the railroad. For example "the process of tie production--from felling the tree in the forest to insertion of the treated tie in the track--is a one-and-one-half-to-two-year-cycle. When tie producers are made aware of railroad tie replacement programs two years out, they adjust to meet the projected demand" ("Why deferring...", 1991).

Not performing enough maintenance will result in an inadequate travel surface. Defects and slow orders due to unsafe track conditions will impair the ability of the railroad to provide an acceptable level of service to its customers. On the other hand, a maintenance policy that repairs every problem as soon as (or before) it arises will prove overly expensive, and without proper coordination with dispatchers, could prove costly to line operations as well. Too many maintenance crews on the line at one time will result in an increase in delays on the line, which affects the reliability of trains and leads to dissatisfied customers. Balance is the key to scheduling maintenance activities.

When scheduling maintenance operations, the effects of the operation on the traffic should be considered. The length of delays to trains will not merely be the length of time that the maintenance crews occupy the line. Some maintenance operations require the imposition of temporary slow orders after the crew is finished its work. This, of course, will add to the running times of trains on the line. Depending on the density of the traffic at the time of the maintenance operation, a delay to one train can have a cascading effect to other trains on the line. On a busy day these delays could even extend into a yard where trains are held because they physically cannot get onto the line or because they are waiting for important traffic that is still on the line. The importance of proper scheduling is emphasized in the following statement: "A delay of two hours which causes a missed connection could result in a full day's delay in delivery of a car to its destination. The true cost of that delay is far more than the ownership value of the rail car and the interest on the lading value; poor service equates to lost customers" (Aspebakken et al., 1991).

Figure 2.1 depicts the iterative nature of the selection of maintenance projects for an upcoming year.

2.3.2 Logistics

Once the maintenance projects have been decided upon, each region must make sure that it can assemble simultaneously the required materials, equipment, and labor at each project. Prioritizing maintenance projects is also important. The order in which projects are to be performed will impact the availability of machinery and crew members. It might become necessary to experiment with different crew mixes.

Railroads to combine projects which are to be performed over the same stretches of track. Whenever more than one maintenance operation is to be performed over the same stretch of track it is important that they be sequenced properly, so as to achieve economies of scale by not performing the same action more than once. Hay (1982) describes this concept in great detail:

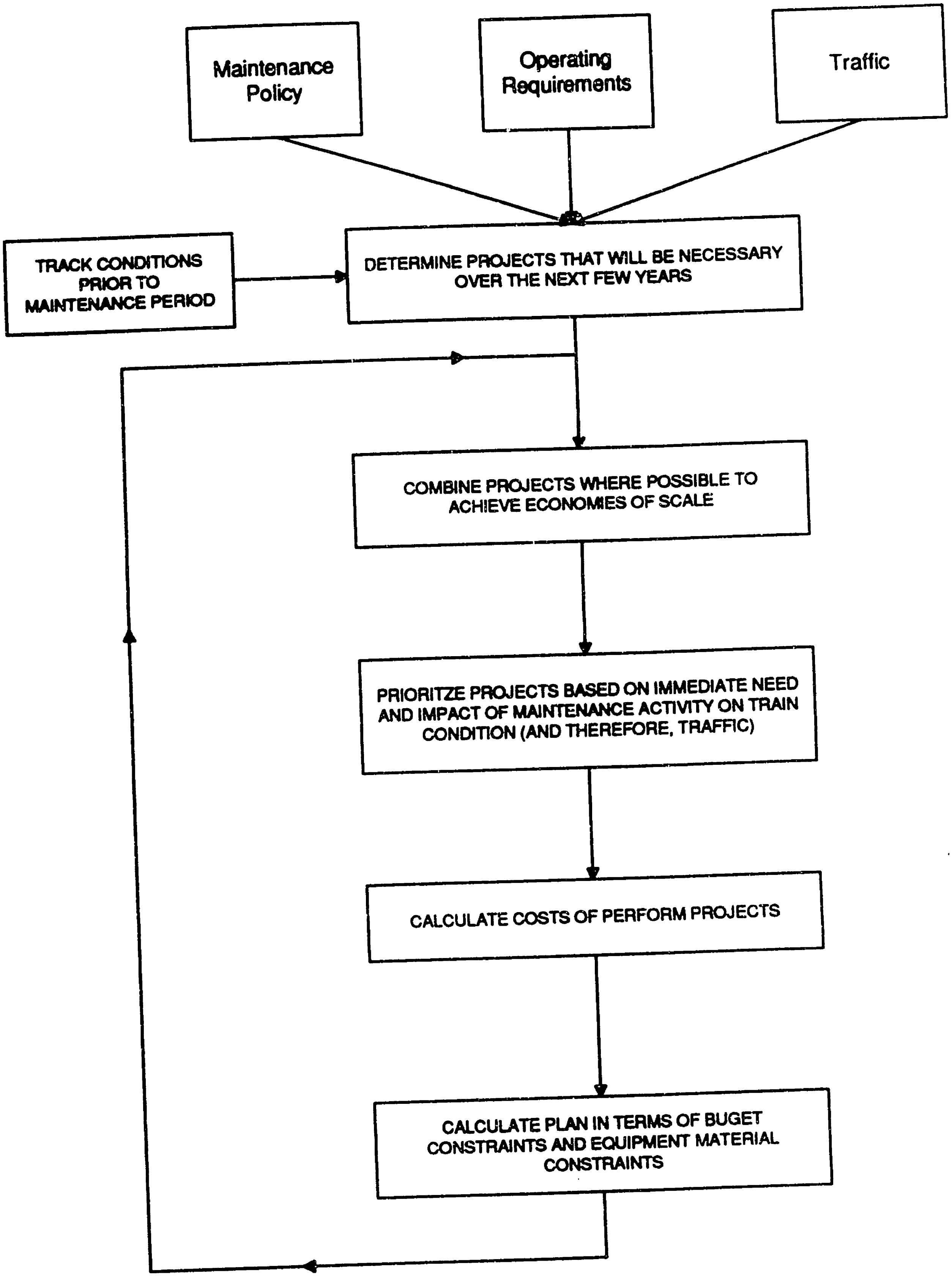


Figure 2.1: Selection of maintenance projects

> It is desirable to have the track in good line and surface before relaying rail. In any event, rough spots of line and surface should be removed by a spotting gang. The rail program may begin early in the spring, too early for ballasting and raising. Relay must then come first, ballasting afterward. Before ballasting and raising, any necessary bank restoration and widening or roadbed stabilization should be performed to prevent disturbing the surface at a later date. Tie renewals are begun as early in the season as possible. When rail relay, tie plate renewals, or gaging are contemplated, ties are installed afterward to prevent the excessive redriving of spikes into the new wood. Ties should be straightened and respaced in conjunction with heavy surfacing or ballasting. Rail-end welding and joint-bar renewals are made only when track is in good line and surface.

Strategically it is important to know what maintenance projects are going to be performed each year. Tactically the goal is scheduling individual projects within the timeframe of a year. There are two considerations when scheduling maintenance projects. The first is the availability of work crew and equipment. The second is finding time windows during which a work gang can get out onto a line with a minimal disruption to traffic. This will be discussed after the section on productivity.

Manpower and machinery are limited resources. If cost were no object, it would be feasible to buy or rent as many machines as there are jobs on a railroad, and to hire and train extra crews made up of temporary workers. All of these extra resources could then be used whenever a time window presents itself with virtually no disruptions to traffic whatsoever. Unfortunately, operating budgets exist. If there is more than one job that can use the same machine, it is usually more cost effective to space the jobs in such a way as to minimize the number of that machine which will have to be bought or rented. An extreme example of the wrong way to manage equipment would be to buy three identical machines for three separate jobs to be performed simultaneously on different parts of the line, and then not to use any of the machines again until the following year. It would make much more sense to purchase one machine and schedule the jobs such that the machine could be used at each one. The same principle can be applied to manpower.

2.3.3 Productivity

Predicting the production rates of various work gangs is important for scheduling and for costing purposes. The amount of time a crew will be on the line is important when trying to fit it into a work window when it will disrupt line activity the least. The length of time that the crew will have to be paid for as well as the rental costs of the machinery are cost questions. There is also the cost to trains which will be affected by the maintenance activity. Without production rates it would not be possible to perform any kind of cost benefit analysis for scheduling a particular crew at a particular location at a certain time.

> The rate [of production] will depend on the technology involved and can be estimated from several sources. The most accurate source for information is extensive past data kept for the purpose of recording production. Ideally, the production rate will be based on work performed in the same location, under the same conditions, and with the same technology as the current work and will have outside factors such as weather delays, travel time and train delays separated from the actual work time" (Love, 1981).

As this quote indicates, determining production rates is generally an empirical process. The speed at which a machine can run and the speed at which it is operated under real conditions are generally not the same. The same can be said about the human members of the crew. One major North American railroad (Conrail, 1992) actually has a set of tables for different work crews which predict productivity rates based on the size of the job and the percentage of curved track in the job site. The methodology tacks on extra time for travel and traffic on each day to determine the total number of days required for the job.

Not all of the time that a crew is on the clock is spent working on the track. There is a portion of each work day lost for travel to the site; setting up the equipment; and breaking it down at the end of the day. The production rate also depends on whether or not the crew is to hold the track when traffic approaches, or whether it is to clear the track and allow trains to pass through. A final consideration is the length of the work day. Union work rules place a limit on the number of hours that a crew can be on the job, including travel to and from the site. Strict, on-track, production rates must be tempered by all of these considerations when calculating the amount of time a job will take.

A list of typical production rates is provided in Appendix A.

2.3.3.1 The Role of Technology

The two categories into which on-track machinery can be divided are machines that can be removed, or setoff, from the track and machines that must be cleared on a side track. The former include small tamping and lining machines, spike pullers and spike drivers, adzers, rail anchor applicators, track cars, track wrenches, tie spacers, and others. Machines which cannot be removed include rail grinding and contouring equipment, undercutters and ballast cleaners, ballast regulators, cranes of various types and high-capacity, heavy-duty versions of tamper-liners, tie remover-inserters, and others. Examples of off-track machinery are bulldozers, front-end loaders, and cranes. There are also a number of hand held tools that are used. These include rail saws, power drills, rail grinders, and one-man tamping units (Hay, 1982).

Production equipment that can be set-off the track quickly is seeing more use on high traffic lines with short work windows. The obvious advantage of so-called quick removal equipment (QRE) or rapid on-off equipment (ROO) is that trains do not need to be held up until the job is completed or while the crew moves to the next siding. Another advantage is that the equipment can be trucked to the work site, where work can begin almost immediately rather than having to be put on the track at a siding. Tie crews can be selective when using ROO equipment. With super-machines all ties, new or old, will be replaced during the project. The trade-off when using ROO equipment is in productivity. An ROO tie gang will install about 15-20% the number of ties that a super-machine could install over the course of a day. "It must be remembered that ROO gangs are smaller, and the disruptions to traffic are much less, whereas high-production gangs require a minimum four-hour work block (eight to ten hour blocks would be better)" (Tuzik "Fast-on..", 1991).

2.3.4 Work Windows

Knowing that sometimes it is not possible to plan maintenance activities so that they have no effect at all on traffic, it is important that the railroad endeavor to keep its customers happy by informing them, in advance, of possible delays. As noted earlier, the shifting of a project by as little as fifteen minutes can result in major time savings for a train. Richard Pyson, vice president-engineering ar Conrail, describes what his railroad does when confronted with this problem: "Every week we have a meeting with the Maintenance, Transportation and Customer Service Departments. The group examines where we're going to work, and what type of track occupancy we can expect. The customer service people are than able to advise customers who will be affected by our plans. I won't say it works 100% of the time, but it's better than operating by the seat of your pants" (Tuzik, "Looking at..", 1991).

Deciding when to move a work crew and machinery out onto the line is the last step in scheduling maintenance. This can be a problem on high traffic lines, such as passenger lines, with short headways between trains. Although computerized dispatching algorithms are being developed, string line diagrams are still in use throughout the industry. Production rates are used to calculate the number of hours that will be required to complete the job. The number of days the crew will work is then determined based on daily hour restrictions. "Windows" of time between trains are then located in the daily schedule, and the crew is given clearance to work at these times. This is generally not much of a problem for areas with low density traffic. When work windows are not easy to come by, railroads might consider options such as smaller gangs utilizing quick removal equipment.

Precision scheduling is not enough to ensure the safety of the work crew and the trains on the line around the work site. There are a number of precautions that can be taken to provide adequate warning of approaching trains. "Restricting speed signs are set at specified distances beyond the relay limits, and intelligent, well-qualified flagmen placed at each end of the relay. In addition, permission to take and open the track must be secured daily from the dispatcher--and returned to the dispatcher at the end of the work by the same person who secured it. Lookout orders should be given to all trains approaching or passing the work on any track. Watchmen to warn the men of the approach of a train on another track are also desirable" (Hay, 1982). There have been recent advancements made in the production of electronic flagmen. These are basically portable motion detectors which sound an alarm to the work crew when a train passes it. The obvious advantage of this equipment is that it frees up two members of a crew for other work. The disadvantages are the loss of work time due to false detections, and the necessary maintenance of machinery in which a failure places the safety of the crew at risk.

Actually getting a work crew onto the line is an iterative process combining logistics and productivity methods. Figure 2.2 depicts this combined methodology.

2.3.5 Seasonality

Weather conditions have a major impact on the production rate of a maintenance. Scheduling work during the rain season or when the climate is going to be cold will mean longer work times. Not only do the men in a crew work slower under inclimate conditions, but the ability to work with the track materials tens to decrease as the temperature drops. "When the ground freezes tie removal, tamping, or digging of the subgrade is very difficult if not impossible." (Love, 1981) Rail is more pliable when it is warm. It is not a coincidence that most rail defects occur during the winter months. Cold whether places the rail in tension making it more susceptible to failure. In fact "Over 64 percent of all defects occur in the coldest six months of the year. The effect on service failures [defects which are not detected by inspection methods] is even greater; with over 75 percent occurring in the same period" (Davis, 1992). The effect of this increased failure rate can be significant.

> Sudden temperature drops often produce several service failures on the same line at the same time which can cause severe service disruptions to the line. The number of failures can overwhelm the available repair resources, and repairs may require more time than average (Davis, 1992).

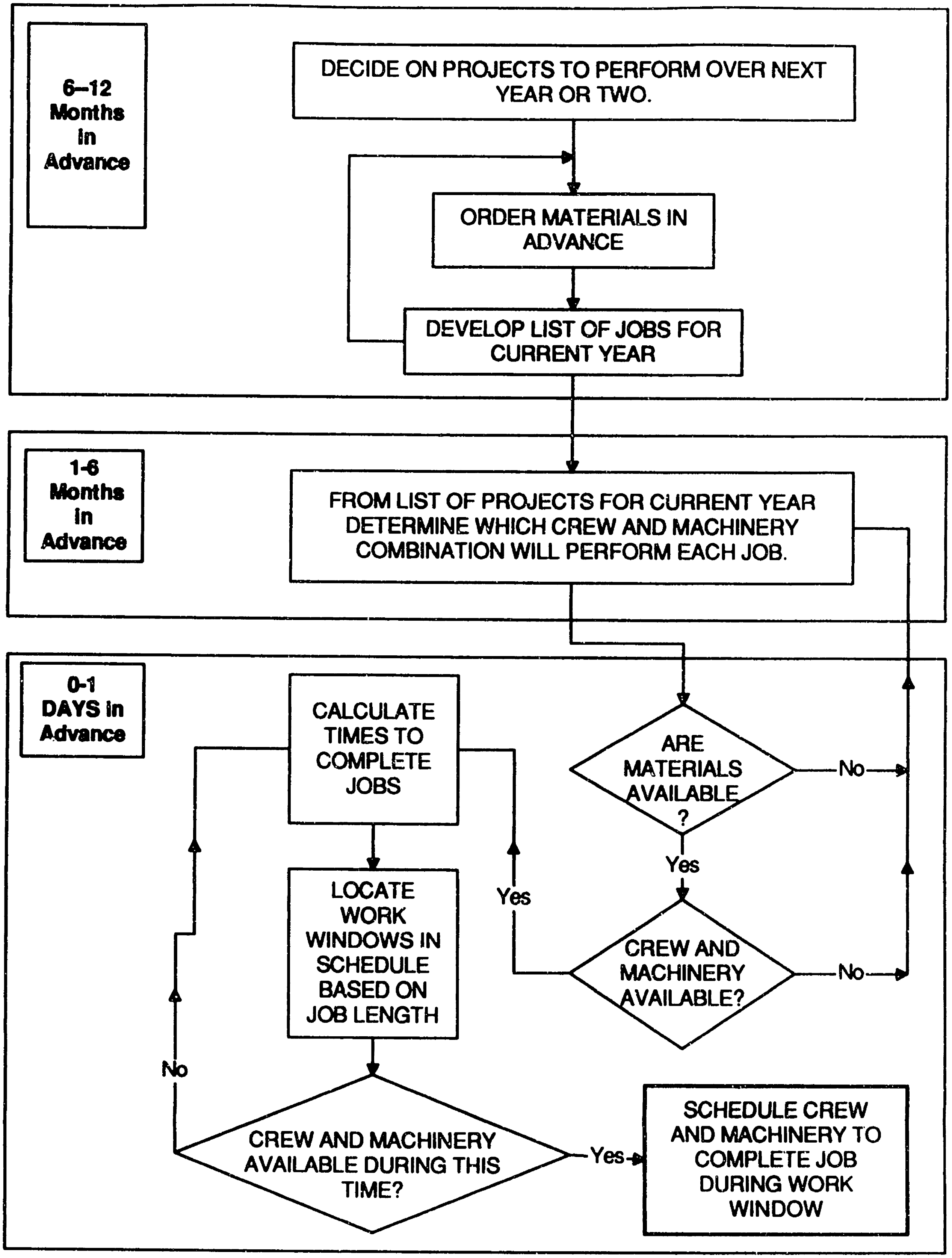


Figure 2.2: Scheduling of maintenance projects

For this reason project work is normally scheduled during the warmest months of the year. Cold weather is not conducive to welding. In fact, it is not uncommon for spot maintenance crews to use temporary joints when "patching" rail in the winter. When the weather gets warmer, the crews revisit the defects and properly weld the new rail in place.

2.4 Maintenance Costs

2.4.1 Direct Costs

Calculating the cost of a particular maintenance job is a very straightforward process. Once the job to be performed is decided upon, it is necessary to procure the necessary materials for the project as well as the work crew and any machinery the crew will need to install the materials.

There are three variables that must be considered when determining the cost of the materials. First, is the quantity of materials. This is determined by the size of the job. Second is the quality of material to be used. The easiest example of this is rail. As recently as November of 1992, the standard price of carbon rail was $460/ton while premium-steel rail sold for $610/ton (Burnes, 1992). The quality of materials selected is generally a function of the traffic and curvature as well as the environmental conditions in the area where the rail or ties or ballast is to be installed. "[Conrail's] idea is that the head is more substantial on 136-pound rail, and with the trend toward more grinding as a rail maintenance technique, [they] want more steel on the head of the rail" (Tuzik, "Looking at...",1991). Other factors include the cost of transporting a particular grade of material, and the budget of the engineering department of the railroad. Budget considerations aside, there are still reasons for using lower quality materials, even though higher quality materials last longer. It may be the case that on lines which do not experience a lot of traffic (i.e. branchlines) the deterioration rate of low quality track components might not be much higher than what a higher quality component might experience. There is even a practice known as cascading rail, in which rail taken from mainlines is used to replace rail on lines with light density traffic. There are many economies of scale that must be considered when purchasing materials.

The third consideration when determining the cost of materials for a job is the transportation cost of the materials. Moving the materials from wherever they are produced to the job site can become a major portion of the cost of the materials. This is especially true for ballast. It is conceivable that the cost of transporting a low quality ballast from its quarry to the project site could be so much more than the cost of transporting a higher quality ballast, that the ballast of higher quality could actually be less expensive.

The crew costs and machinery costs depend on the size of the job and the productivity of the crew and machinery. For a particular job, it is possible to estimate how long it will take a work gang, with or without machinery, to complete a job. This is accomplished by dividing a productivity number of the form "units placed per unit of time" into the number of units of material to place. (Units of materiai are usually expressed in miles for rail and ballast. For ties it is a matter of how closely the ties to be replaced are clustered as well as the number of ties to be placed.) Once the estimated length of the job is determined, a gang and the necessary machinery is assigned to the job. The wages for the gang and the rental cost of the machinery per hour or day will determine the cost of the job. Economies of scale can also be achieved by longer work cycles due to the cost of transporting heavy machinery (ex. grinders) to work sites. The fewer the number of times the equipment has to be moved, the lower expenses will be.

2.4.2 Hidden Costs

The hidden costs of maintenance are train delay costs and losses in expected levels of service reliability. The first cost is borne by the railroad itself. It must pay for the amount of time that cars and locomotives occupy its lines and terminals. Based on an estimate of the time value of the equipment and contents fo the train, a value of $180/train/hour was used by Dontula (1991). He estimated that on a line with 40 trains per day capacity, operating at 70% of this capacity, a defect repair of 2 hours would add up to approximately $2500 in train delay costs. The second cost is not as easily quantified. Unreliable service leads to dissatisfied customers. While it is hard to predict what a few late trains will mean for a shippers perception of service; customers who are consistently disappointed with the service rendered will take their business elsewhere.

MOW work, track defects, and train breakdowns have the effect of reducing the capacity of the line. Reduction in capacity is what leads to increases in meets and passes and variabilities in trip times. For the railroad, the question is often whether to reduce capacity knowingly, for MOW, or to risk delays from slow orders and line closings due to defects which arise from deferred maintenance. At least with planned maintenance, the railroads have the option of informing their customers of possible delays ahead of time.

The various elements which play a role in determining the total cost of a maintenance activity are depicted in Figure 2.3.

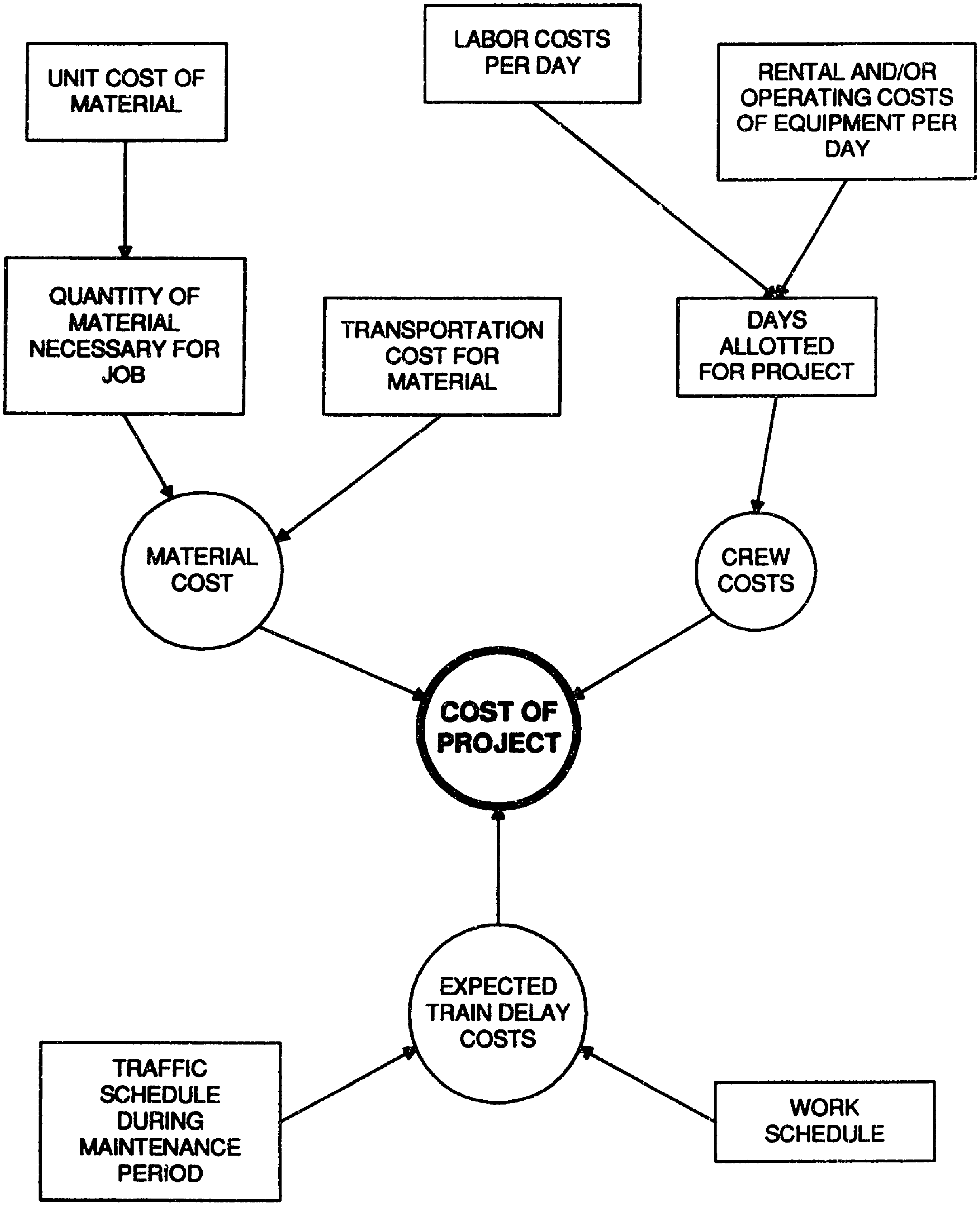


Figure 2.3: Determining the cost of a track maintenance activity

Chapter 3 Maintenance Planning Models

There are many factors that go into planning maintenance activities. A railroad's engineering department uses inspections and information from databases (past inspections, records of past maintenance activities, and traffic records) to determine the present condition of the track materials. Maintenance activities are projected and prioritized over the next several years, and based on manpower, equipment, and budgetary constraints, the year in which each activity is to take place is determined. There are many intermediate steps in this process, as evidenced in chapter 2. Track maintenance planning models are one means by which railroads are making several of these steps easier and faster to perform. Whether they are used to predict life cycles of materials, expected numbers of defects, or to actually schedule projects, they all improve maintenance planning.

3.1 Deterioration Modelling

By definition, a maintenance *planning* model needs to be able to estimate a time for track maintenance activities. This information can be determined by deterioration models. The complexity of deterioration models can range anywhere from "rules of thumb" used by experienced roadmasters, to a complex analytical model with a large number of inputs. While subjective predictions, when made by experienced personnel, can be very accurate, the sheer size of railroads along with the large amounts of money spent on maintenance activities encourages the use of objective analytical models. Objective models can standardize maintenance decisions over an entire railroad. They are also able to take advantage of the large databases of information that railroads have available on track condition.

3.1.1 Wear Modelling

Wear models can be developed for any track component, be it rail, ties, ballast, turnouts, signals, etc. The models take as inputs the initial condition of the component in question, the type and amount of traffic which will pass over it, and wear parameters to be used in a (sometimes highly complex) mathematical relationship relating the inputs which yields a deterioration rate of the component. This can be used to determine the amount of time until the component wears out (i.e. the amount of wear has reached predetermined limits for safe efficient operation), and therefore needs to be replaced. The most important initial conditions are the condition and quality (i.e. characteristics) of the material of which the component in question is composed. Complex models have been developed which take into account such information as the curvature and superelevation of the track, the environment which surrounds it (temperatures, weather conditions). Initial conditions can be obtained from records, and from inspection as mentioned in section 2.3.

A wear model will be able to predict when a component will no longer meet standards which allow it to remain in service. Advances in materials and operating regulations almost make it a pre-requisite for a model to be able to adapt to different standards. For example, the limit on head wear of track might be 0.5 inches for 132 pound rail, but for 136 pound rail, the limit might be 0.65 inches. Another reason flexibility is important is the value of performing "what if?" analyses. It could be important to know how much more damage will be incurred if maintenance is deferred for an extra year or two.

An example of a wear model for rail is RAILWEAR, the wear model used in TRACS (Martland and Auzmendi, 1990). This model relies on such information as the metallurgy of the rail, the degree of curvature, annual amounts of lubrication and grinding, and the annual traffic, in MGT. It calculate yearly wear to the high rail, low rail and gauge face. When the cumulative wear exceeds the wear limit, the model recommends a relay.

3.1.2 Fatigue (Defect) Modelling

Some track components will fatigue and give out or fail before they reach their expected wear limits. This is not unexpected, but predicting how often and where defects crop up can be a problem. Just as knowing the approximate time that a replacement project will be necessary allows the railroad to acquire the necessary materials and equipment ahead of time; knowing the expected number of failed rails and ties (for example) will give the railroad the opportunity to make preparations to effect repairs. Having an idea where these problems could arise also gives the railroad an idea of where inspections will be likely to turn up potential or actual defects.

Since defects are essentially random occurrences, fatigue models are probability based. A relationship is determined, usually between MGT and the occurrence of failures, that can be fit to a probability distribution. An example of this the one used in the RAIL PERFORMANCE MODEL, (developed by Wells and Gudiness of the AAR, 1981) which uses a cumulative Weibull distribution of the form:

$$F(MGT)=1-e^{-(\frac{MGT}{\beta})^{\alpha}} \tag{3.1}$$

Where F(MGT) = Cumulative probability that a rail will have a defect before reaching a specified level of MGT.

Alpha is the shape parameter. It indicates how rapidly the slope of the curve changes. Beta is the characteristic life of the rail, the MGT by which 62% of the rails are expected to have a defect. (Wells, 1983)

A rail defect model which takes a different approach is the Phoenix model, also developed by the AAR (Steel and Joerms, 1988) to simulate the mechanical processes that lead to fatigue defects. To do this, the Phoenix model calculates the stress levels within the rail head, based on elements such as wheel loads, rail metallurgy, and the interaction between rail and wheel at point of contact. It then uses a set of curves (S/N curves) developed from laboratory experiments which relate the stress levels to the number of repeated loadings (based on the mgt) the rail can experience before cracks begin to form. Shyr (1993) has developed a system of equations calibrated to Phoenix output, for use in TRACS.

The predictive value of defect models is of particular use when simulating line activity. Unplanned defects are a fact of everyday operation of a railroad line. When simulating line operations, it makes sense to include them as random factors. Defect models are useful when performing cost/benefit analysis of a maintenance policy. When maintenance is deferred on a component, the probability of that component failing goes up. Defect modelling allows a comparison of the costs of replacing components before defects occur, or not replacing the component but having to repair the additional defects which arise instead.

To summarize, a deterioration model takes as input the initial state of a track component, and determines that state of that component over a planning horizon of one or more years. How this information is used will be described in the following sections.

3.2 Life Cycle Costing Models

It is not enough to know when a component is going to wear out, or when it will probably have to be replaced due to defects. Ultimately, cost is what determines which repairs are made and which are put off until a future date. Life cycle costing models are used for financial planning as well as the capital cost planning that is essential when purchasing specialized equipment and large quantities of materials. "Class I capital expenditures for 1990 were $2.644 billion for roadway and structures, $996,000 for equipment. Maintenance expenses were $6.350 billion for equipment and $4.278 billion for maintenance of roadway and structures" ("Review of...", 1992). This statement reveals the value of a model which can take deterioration information over a several year planning horizon and translate it into a net present value (NPV), or an Equivalent Uniform Annual Cost (EUAC) for that horizon.

The Total Right-of-Way Analysis and Costing System (TRACS) developed by the Association of American Railroads and its affiliated laboratory at M.I.T. calculates the costs of performing maintenance on rail, ties, ballast, and turnouts over a 1-25 year period. It requires four major sets of inputs. These are the traffic file, route/track file, cost file, and a knowledge base which defines components, maintenance policies and default values. The traffic, route/track, and knowledge base files are used by the deterioration models to determine the number of and time of maintenance events. The cost file is used in conjunction with crew consist and productivity information to calculate the costs of performing the maintenance for each year. Discount rates are then used to express the total cost as a NPV or EUAC. The user has the option to view both cost and maintenance (both project and spot) information in a set of reports. Figure 3.1 exhibits the structure of

TRACS (Hargrove and Martland, 1989). The input requirements and outputs of TRACS will be elaborated on in section 3.4.

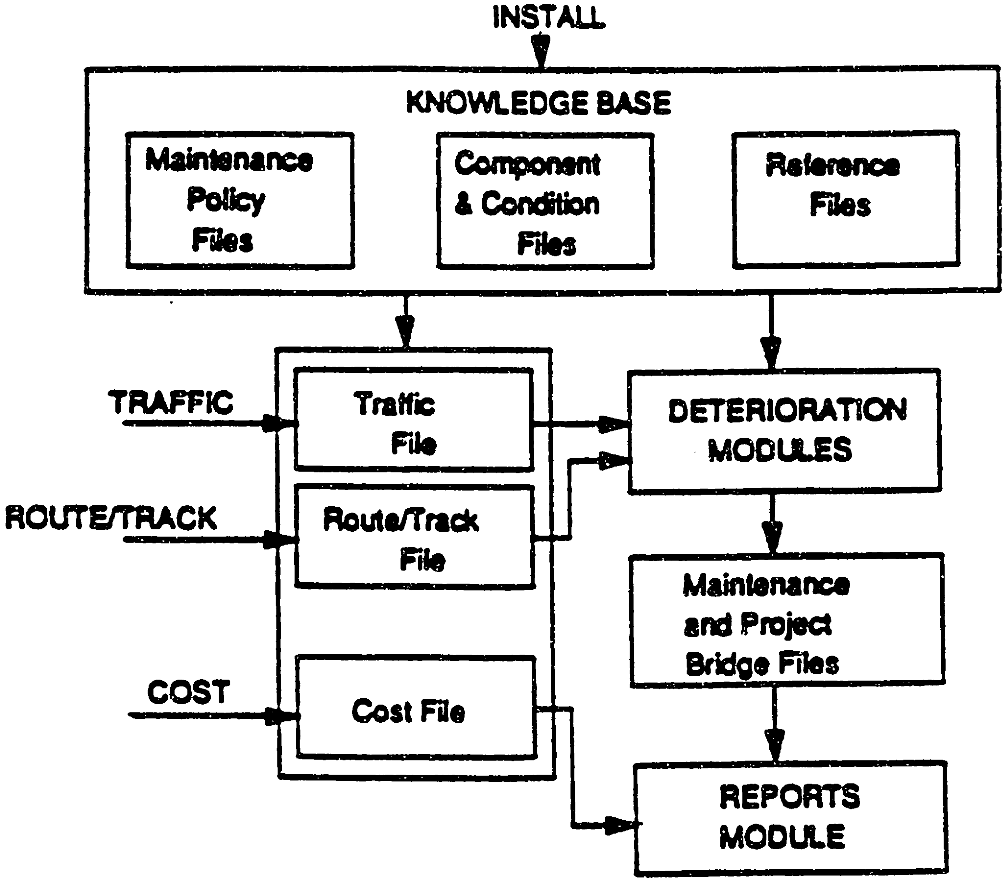


Figure 3.1: Structure of TRACS

The ability to plan for the future, and to perform what-if analyses, provides options for railroads. Testing the effects of different traffic patterns, or different track materials on future maintenance costs can aid in decisions about whether to expand service or to keep the company on the same path for the time being. Knowing maintenance requirements ahead of time will make capital budgeting decisions much easier.

3.3 Maintenance Scheduling Models

Models such as TRACS provide the user with valuable cost information, but the costs are only generated for what the deterioration models determined to be necessary maintenance operations. Opportunities to achieve economies of scale, either by combining maintenance projects or by including sections of track in projects a year or two early to take advantage of a project on adjacent sections, must be discovered by the user. Models which make decisions like this, ultimately aid play a role in the scheduling of projects, and are therefore included in this section.

One such model is the Rail Performance Model (RPM). It was developed by the AAR's Track Maintenance Research Committee. The RPM calculates optimum rail life by using expected future maintenance, relay, and other pertinent costs and reducing to a net present value. It basically determines the point at which it is more economical to replace the rail instead of fixing defective and broken rails (Staplin and Wells, 1987). This is a different approach than that taken by TRACS, which schedules rail to be removed based upon a maintenance policy. In other words, the RPM makes suggestions for replacement based on the future costs of maintenance; TRACS makes suggestions on what it will cost, based upon a policy that specifies limits for wear and for defects/year.

REPOMAN takes the RPM and TRACS several steps further. It is an expert system which uses heuristics, rules of thumb, economic analysis, maintenance policies, and experience in developing a relay program. The steps it follows in developing a program are contained in six modules which are described below (Martland et al., 1990)

1. Data Compression: This module takes segment information from railroad databases and creates files that can be used by the expert system.
2. Categorization: This module is a knowledge based expert system. It determines wear rates, and calculates the market value of the rail in each segment. It uses the projected wear rates to place each segment in a categories of OK, MAYBE, SHOULD, and MUST. In addition to this the module indicates possible actions to be taken for rail in each category.
3. Field inspection: The rail must still be inspected to catch errors in the databases and to locate any defects or developing defects that were missed in prior inspections. This process is made easier by the recommendations of the expert system.
4. Recategorization: The information gathered in the field inspection is used to update the database. This updated information is run through the expert system to yield more accurate categorization.
5. Preliminary Relay Program: A preliminary rail relay program is prepared assigning an action for every MUST segment. In addition, the expert may recommend other segments for relay for reasons such as reduction of maintenance costs, generation of second-hand rail, taking advantage of

economies of scale in relaying rail, or smoothing year to year peaks in workload.

6. Report Generation: This module presents the information in user friendly formats.

3.4 Use of TRACS in this thesis

In order to perform the case study in chapter 5, TRACS was used to generate maintenance requirements, for different traffic consists, over a line approximately 300 miles long. By using an existing maintenance model, with default maintenance practices, it was not necessary to develop a new model or to rely on the past maintenance practices of a particular railroad. TRACS was developed by the A.A.R. affiliated laboratory at M.I.T. in cooperation with an industry task force.

There are several reasons why TRACS was chosen to determine the maintenance need for the case study in chapter 5. The primary reason is that TRACS contains deterioration and fatigue models for the major track components: rail, ties, ballast, and turnouts. For each component, TRACS reports the project (replacement) maintenance requirements. The outputs are translated into steady state replacement cycles which can be used in a simulation model as typical yearly projects.

TRACS will also report the expected number of spot maintenance events to be performed each year over each section of track. The measure of a maintenance policy is how well it controls operating disruptions due to track failures. Different maintenance policies mean different numbers of failure.

As part of the initial conditions of a particular run, TRACS has the capability to specify the axle load distribution based upon train consists. The user can select from 4 different locomotives and up to eight different cars. There are several characteristics which distinguish different cars (and locomotives) from one another. The characteristic which is most important to the case study is the axle load imparted onto the rail.

TRACS allows the user to change the deterioration limits which determine when rail is to be replaced. It also gives the user direct access to the grinding program to be implemented. Grinding is usually increased, when heavy axle loads are run across the line, to control the growth of corrugations in the rail. TRACS determines when rail or ballast or turnouts or ties *have to be replaced* under a set maintenance policy. The maintenance policy can be altered to determine the lives of rail under different wear and defect limits. The flexibility in changing the maintenance policy will allow a check of what deferring maintenance will have on the development of rail defects.

Table 3.1 depicts average quantities of maintenance generated from three runs of TRACS. Table 3.2 reports the typical number of crew-hours spent performing this maintenance. Figure 3.2 is a graphic representation of these results. All three runs used a route of 292 miles with the same initial condition of track materials. The difference in each case is the axle load distribution of the traffic with the loaded cars in each case of 263,000, 286,000, and 315,000 pounds. These loaded cars imparted axle loads of approximately 33, 36, and 39 tons respectively. The quantities of maintenance reported for each year are average (steady state) values over a 25 year planning horizon.

Further details concerning these runs, as well as a set of runs for a 30 mgt line are discussed in chapter 5.

	Axle Load (tons)		
TRACK COMPONENT	33	36	39
RAIL			
miles relayed/year	20.7	20.6	25.0
defects/mile/year	0.8	0.9	1.2
BALLAST			
miles renewed/year	57.3	57.3	67.6
miles surfaced/year	139.9	155.9	176.6
TIES			
project ties / year	31745.	32188.	33147.
spot replacements/mile/year	7.91	8.14	8.20
TURNOUTS			
Turnouts replaced/year	3.2	4.8	6.4
Components replaced/year	20.8	19.2	40.0

Table 3.1: Yearly Avg. Maintenance Requirements: 80 MGT, 292 mile line

	Axle Load (tons)		
TRACK COMPONENT	33	36	39
RAIL			
relays	206.7	206.1	250.3
defects	466.4	524.7	699.6
BALLAST			
renewal	573.0	573.0	676.2
surface	699.6	779.5	882.8
TIES			
renewal	158.7	160.9	165.7
spot	11.5	11.9	12.0
TURNOUTS			
Turnouts replacement	22.4	33.6	44.8
Component replacement	62.4	57.6	120.0

Table 3.2: Avg. hours/year spent on Maintenance: 80 MGT, 292 mile line

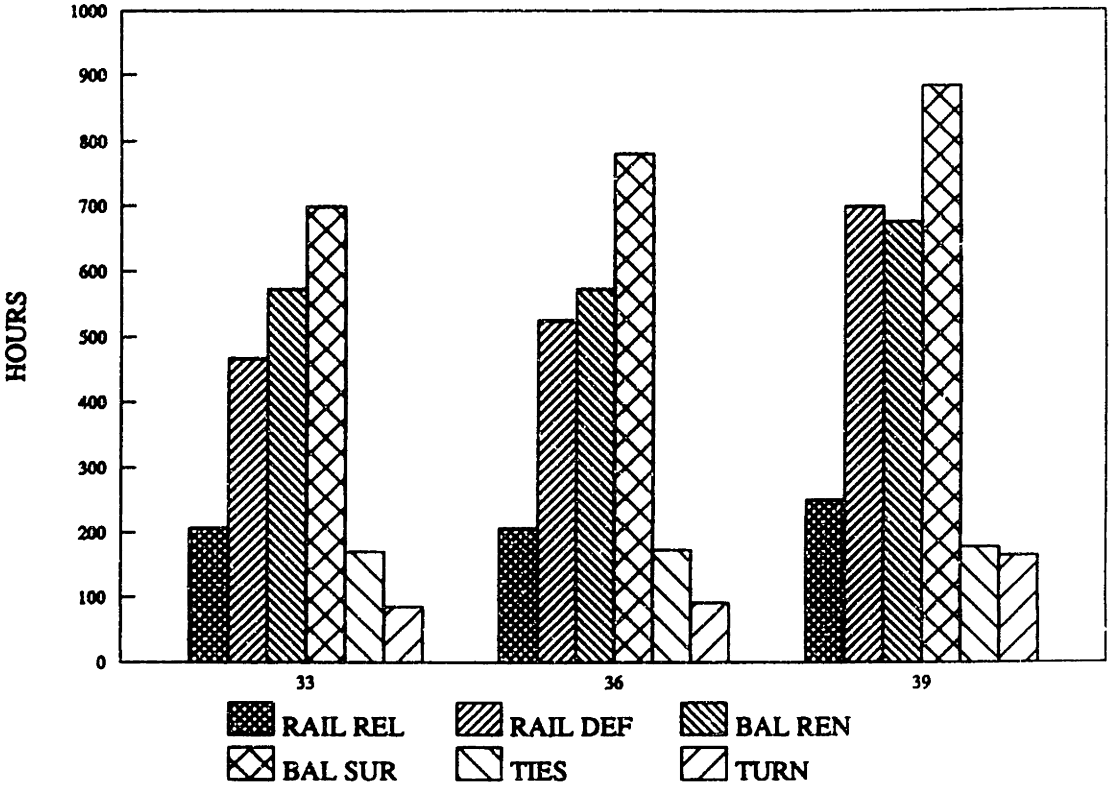


Figure 3.2 Avg hours of maintence per year on an 80 mgt line

Chapter 4 Modifying a Simulation Model to Include Track Maintenance

4.1 Introduction

4.1.1 Why Use a Line Haul Model?

In the section on costs in chapter 2, the notion was put forth that there is more to determining the cost of maintenance activities (both planned and unplanned) than adding up the material, labor and equipment costs. "For heavy tonnage lines the costs due to train delay or lost service can be substantially more than the actual cost of repairing/replacing the defect." (Davis and Staplin, 1991) Different maintenance plans will call for varying amounts of project maintenance which in turn will result in different amounts of spot maintenance that will have to be performed for track segments on which maintenance is deferred. There are also options when selecting crew consists for particular jobs. A larger crew or a crew with more advance equipment can complete jobs faster, but usually at greater cost. A model with which a railroad can predict expected average delays and variabilities of trip times for different maintenance plans would allow them to compare these plans on more than just their straightforward costs.

4.1.2 Why use a Simulation Model?

Analytical models have been developed which can estimate delays due to MOW activity. Dontula (1991) developed such a model using deterministic queuing theory relationships. The problem with most such models is that they "simply compute the mean delay (or travel time). However, the reliability and hence, the quality of freight rail service, must include the variance term." (Harker and Wong, 1990) Another difficulty with analytical models is that they are either overly complex like the model developed by Harker & Wong (1990) in that they require large quantities of data, or the simplifying assumptions that have to be made call the results into question. The state of a railroad line is constantly changing. The speed of trains is not a constant, as they have to slow down for meets, or trains which have broken down in front of them. Maintenance of Way activities which include section shut-downs and slow orders will cause changes in train speeds and delays that can be estimated with queuing theory, but queuing theory brings in its own abundant assumptions. At the time of its creation, the model developed by Harker and Wong, was "the only analytical approach for dealing directly with the scheduled train issue." (Harker and Wong, 1990)

A simulation model can handle events such as train meets as they occur. It is not necessary to determine beforehand the probability that two such trains will meet on the line, and the further probability, given they do meet, of which one will be delayed. Basically, a simulation model is better suited for handling the detailed events which can occur along a line-haul. A simulation model can also calculate average trip time and standard deviation of trip time without delving into probability theory. A trip time distribution, from which these values can be calculated, can be part of the output of a simulation model.

4.1.3 Structure of This Chapter

The introduction gave some reasons for using a simulation model when studying line reliability. Section 4.2 presents the background on the simulation model which was modified, and describes the structure of the model including changes and additions that were made to the model for the purpose of this thesis. Section 4.3 describes the methodology used in placing track maintenance events on the line. Section 4.4 details the calculation of travel times, and the dispatching logic. Section 4.5 provides descriptions of the major functions and data structures.

4.2 Structure of the Model

The line-haul simulation used in this thesis is a modified version of a discrete-event based simulation model created by Dontula (1991). This section contains a description of the basic structure of the model as developed by Dontula. It also contains the changes and additions that were made to the original model in order to better simulate track maintenance activities. The additions made it possible to place track maintenance events on the line at pre-determined times in the simulation, and to create random track failures which resulted in track maintenance events. These additions necessitated changes in the calculation of trip times across segments, in the dispatching policy, and in the scheduling methodology.

4.2.1 Inputs to the Model

The inputs that must be supplied by the user in order to run this model are track configuration and the characteristics of each segment, train schedules and characteristics, a maintenance plan detailing when and where maintenance projects are to take place, and train and track failure patterns. The structures of the various input files, and how the information they contain is converted for use, are described below.

Track Configuration and Characteristics

The line is made up of an odd number of single track segments separated by segments with adjacent passing sidings. At each end of the line is a station (or terminal) from which trains are dispatched onto the line. The model treats these stations as segments with sidings. If there are total of M segments between stations, the outbound station would be the 0th segment, and the inbound station would be the (M+1)th segment. Single track segments are assigned odd identification numbers (from 1 to M), and segments with sidings are assigned even identification numbers (from 2 to M-1). The track configuration input file contains the segment id number, the segment length, and the maximum allowable speed on that segment. A uniform siding speed is used throughout the line and can be changed from one run to the next of the program. The siding speed is separate from the segment maximum speed in order to allow the program to simulate the reduced travelling speeds through passing sidings.

Other information that is pertinent about a segment, while the simulation is running, include its free time, the id number of the train on its segment and/or siding and the time at which it is due to depart the segment, and whether or not a slow order is being imposed and the location of that slow order. All of the variables used to keep track of this information are described in section 4.5.

Train Schedules

In the original model, trains were dispatched at regular intervals (offsets), alternating between the outbound and inbound yards. This effectively scheduled the same number of trains for departure for each day of the simulation. The approach taken in the modified model is to allow different numbers of trains to be dispatched depending on the day of the and the direction of the trains. Separate input files are used for outbound and inbound trains, although the formats of the two files are identical. For each day of the simulation, the input file contains: the day, the time of departure of the first train of the day, the id number of the first train of the day, and the time between successive departures (offset). Two successive lines in an outbound (ob) train schedule might look like the following:

8	**150.000**	**157**	**95**
9	**50.000**	**183**	**95**

On the 8th day the first ob train scheduled to depart is train #157, at 150 minutes after the day begins (2:30 a.m.). Train #159 is scheduled to depart at 150+95 = 245 minutes after the start of day three, and ob trains will continue to be dispatched onto line every 95 minutes on day 8 until train #181, the last outbound train scheduled for departure on day 8, is dispatched.

It is possible that a train might not be able to move onto the line at its scheduled time of departure. This would happen if the first segment were occupied by a train or maintenance crew, or if a line blockage problem could result if dispatched. Whenever a train is delayed, its dispatch time is pushed back until the problem is resolved. It then attempts to depart again. Trains can be delayed until the next departure time is pushed past the first departure of the following day. When this happens, all remaining trains on the current day are canceled, and any cars left in the station are assumed to be added to trains throughout the following day. In the sample schedule, if train # 181 had been delayed until after 50 minutes had passed on day 9, the it would be removed from the schedule, and train #183 would be the next train to depart from the outbound station.

Train Characteristics:

Aside from the train id number and its direction, there are several other pieces of information that are important when it is on the line and after it has arrived. Information that is important when on the line includes location, priority, length, current speed, and its maximum travelling speed. The length, maximum travelling speed, and priority are all constants for each direction of travel. A more detailed scheduling procedure, possibly one that reads trains in from a file, could be used to impart more variety to the trains on the line. Once a train arrives at its destination, its departure time and arrival time are required to calculate its trip time. The total amounts of delay experienced in the dispatching yard, and on the line are also important. Other information that is used in the dispatching is described in the section on dispatching and in section 4.5.

Maintenance Plan:

Planned maintenance events can be read in from a file created by the user. The details required for each event are the segment on which the event is to take place, the day of the event, the hour at which it is to start on that day, the length of time for which the track is to be closed, the length of the slow order to follow the track closure, and the location of the slow order within the segment. If the user wants to implement a slow order without first closing the segment, all he has to do is set the length of time the track is closed to zero. The starting and ending positions of the slow orders are entered as the distance from the outbound end of the line.

Train and Track Failure Patterns

Train failures are generated the same way as in the original model. The probability of failure was entered as the mean train miles between failure. The mean repair time was also entered. This data was used to induce random train break-downs on the line of varying length. The probability of failure was a composite of locomotive and car probabilities of failure.

The train failure information is contained in the same input file as the track failure information. The user can enter defect rates and repair times for up to four track components. For each component, the user can list the following information:

- a defect rate in defects/mile/year
- the probability (given a defect occurs) that it results in track closure
- the average down time (given track closure)
- the variance of track down time
- the average slow order length (given the track is not closed)
- the variance of slow order length

The option of implementing a slow order if the track is not closed, is designed to simulate the impact of the defect on passing traffic until a repair crew can get out to the repair site. The variance of down times and slow order lengths are used to represent variation in travel and repair times. In the model's present form, when added and subtracted to the average down time and slow order length, the variance will yield upper and lower bounds for those variables.

4.2.2 Events

The following events can occur in the simulation:

Departure of a train from a station

Departure of a train from a segment

Creation of Track Maintenance Events

Placement of Planned Maintenance on the Line

Placement of a Random Track Failure on the Line

> Whenever a train departs from the origin we term that an event. If we cannot dispatch a train because such s move might create infeasibility problems such as line blockage, we delay that train at the origin until the infeasibility is resolved. Whenever a train departs from its origin, another train departure event is created. The departure from a segment, on the other hand, refers to the movement of a train from one segment to another, and finally to the destination yard if the present segment is M for an outbound train and 1 for an inbound train. (Dontula, 1991)

Descriptions of the train depart station and train depart segment functions are contained in section 4.5.

Three functions have been written to handle track maintenance. Each of these functions is an event in and of itself. At the beginning of each day (i.e. every 24 hours) in the simulation, the create track maintenance function is called. This function determines which planned maintenance events are to occur on that day, and places them in the event list (to be described in the next section). The function also uses the track failure information to generate random track failures that are to occur throughout the day, and places them in the event list as well. Both of the functions which handle placement of track maintenance on the line can shut down segments by increasing their free times, and they can place slow order on segments. The difference between the functions is how they determine the characteristics of the maintenance event. The planned maintenance information is obtained from an array which is created from the information in the planned maintenance input file. The length of track closures and slow orders for unplanned maintenance events are generated using random numbers and probabilities from the track failure input file. Descriptions of these functions are contained in section 4.5.

4.2.3 Execution of the Discrete Event Simulation

The following description of a the discrete event simulation was obtained from Dontula (1991)

> In any discrete event simulation, the order of execution of events is of paramount importance. The simulation clock is always updated to the next immediate event that is to occur. All the events that are generated as the simulation proceeds are put in a priority queue. The weight of the event for prioritization is the time occurrence of the event. The recursive procedure of priority queuing is as follows:

> If Current Event is the first event, create a priority queue.
> whenever an event is generated, search the priority queue and place event such that the weights of the events are in an ascending order.
> If there are several events that are scheduled to occur at the same time, the order of execution of these events is FIFO (First in first out).
>
> The event that is to be executed next is picked up from the top of the priority queue thus generated. The events can be one of the [five] events listed in the section 4.2.2....The execution of the program continues until one of two conditions are satisfied. One is that there are no more events to execute in the event list, and the other is that the total period of simulation has elapsed.

4.3 Placing Maintenance Events on the Line

Track closings and slow orders are simulated as track maintenance events in the model. Each maintenance event consists of five components:

- time of event
- length of time track is closed (maint_length)
- length of time slow order is imposed (so_length)
- starting position of the slow order relative to the outbound end of the line
- end position of the slow order relative to the outbound end of the line

At the beginning of each day of operations, an event calls the function *create_track_maint_events*. This function generates unplanned track maintenance events for the following day, and it determines which planned maintenance events are to take place during that day as well. The function places an event containing the segment number, time, and type of event in the event list. When the event reaches the head of the queue it will call one of two functions. If the maintenance is unplanned, the function called is *track_maint_failure*; otherwise the function called is *track_maint_planned.* Both functions determine the length of track closure time and slow orders, as well as the position of the slow order within the segment. The difference is that *track_maint_failure* generates these values based on probabilities entered by the user, whereas *track_maint_planned* takes these values from the planned maintenance input file via a matrix titled *planned_list.*

Whichever function is called, if the maint_length is greater than zero, the segment's free time will be set equal to the current time plus the maint_length. If the segment is current occupied by a train (and hence its free time is greater than the current time) the free time is incremented by maint_length. In a real situation, the repair crew would have to wait for the train to clear the line anyway, so this is not a problem. The slow order length is added to the segment free time to get the slow order end time, and the segment's so_end_time is updated accordingly.

4.3.1 Generating Unplanned Maintenance Events

The generation of train failures is handled the same way as in the original model. Whenever a train enters a segment a random number is generated; if that number is less than the quotient of the segment length divided by the train failure rate (in miles between failure), a failure occurs. The time that the train is delayed is determined using an exponential distribution, a random number generator and a mean repair time.

The creation of unplanned maintenance due to track failures is handled differently.

At the beginning of each day of operations, an event calls the function *create_track_maint_events*. This function scrolls through the list of segments, and generates unplanned maintenance events for the following day. For each segment it determines whether or not a failure occurs for each component, and if so, the time at which it occurs, and the length of down time or slow order. The steps followed for a segment (*i*) and one of the track components (*A*), at the start of a typical day of operation, are listed below. It should be noted that all times are in minutes.

```
PROB_FAIL_A_on_i = FRA * LENGTH_i / 365

Generate random number RA1

if( RA1 < PROB_FAIL_A_on_i)
  FAILURE_OCCURS

if( FAILURE_OCCURS)
 {
  Generate random number R_TIME

  TIME_OF_DEFECT = TIME AT START OF DAY. + 1440. * R_TIME

  Generate random number RA2

  if(RA2 < PROB_TRACK_CLOSURE_A)
   {
    Generate random number RA3

    DOWN_TIME_i = AVG_DOWN_TIME_A + (-0.5 * RA3)*(DT_VAR_A)
    ADD_SO_LENGTH_i = NUM_TRAINS_SO/AVG_OFFSET
   }
```

```
else (track is not closed)
  {
   Generate random number RA4
   $SO\_LENGTH\_i = AVG\_SO\_LENGTH\_A + (-0.5 * RA4)*(SO\_VAR\_A)$
  }
}
```

All of the random number generators used are designed to return uniformly distributed numbers between 0 and 1. The addition slow order length generated when track closure occurs is to simulate the "seasoning" period that is required after ballast has been tamped as part of a maintenance activity. The length of this slow order is dependent on how long it takes NUM_TRAINS_SO to pass over that segment. The AVG_OFFSET is used as an indicator of the number of trains crossing the line per day, and is expressed in units of trains per minute. Dividing the number of trains required for "seasoning" by the average offset yields the approximate number of minutes until enough trains have passed over the recently repaired section. Since rail repairs do not generally require tamping of the ballast upon completion, this additional slow order is not added when a rail defect closes down the line.

If either a track closure, a slow order, or both are to be implemented, a track maintenance event is placed in the event list (priority queue) based on the time it is to begin. Contained in this event are the segment number, time of occurrence, length of track closure, length of time beyond track closure time that a slow order is in effect, and the location of the slow order. For unplanned maintenance events, slow orders are placed in the middle of the segment.

4.4 Dispatching Algorithm

In the original program, when a train reached the end of a segment, if the next segment was free, and if moving the train would not result in line blockage, the train would be dispatched at full speed. The condition of the segments beyond the next two segments, would not influence the decision of whether or not to dispatch the train, and it would also not affect the speed of the train when it reached the end of the next segment. The inclusion of maintenance activities, and slow orders as well as the change to allow a train to travel through a turnout at reduced speed, required a change in the dispatching policy, and in a way that travel times across segments are calculated.

4.4.1 Calculation of Travel Times

The calculation of travel times is key to the determination of dispatch times from segments and sidings and, therefore, segment free times. These two elements are what drive the dispatching algorithms. Previously, the time to cross a segment was calculated by dividing the length of the segment by the speed of the train. If a train was held at the end of a segment, it was immediately decelerated to zero. Upon dispatch, 8 minutes was added to travel time across the next segment to approximate acceleration back up to speed. There are three problems with this approach. First, the amount of time the train spent on the segment on which it was halted should have been increased as well. Second, 8 minutes is only adequate for an acceleration rate of 360 miles/hr^2 and a travelling velocity of 45 mph, therefore limiting the possibility of having different segments with different maximum travelling speeds. The third problem is that it is not possible to model slow orders within a segment using this method.

4.4.1.1 Equations of Motion

Travel times are now calculated using the universal equations of motion. In the following equations, Vi is initial velocity, Vf is final velocity, a represents the rate of acceleration/deceleration, s is the distance travelled, and t is the travel time.

$$Vf = Vi + a*t$$

$$s = Vi*t + a*t^2/2$$

$$Vf^2 - Vi^2 = 2*a*s$$

Through the use of these equations, when given a trains initial speed when entering a segment, the desired final speed, the maximum allowable speed on that segment, and the acceleration and deceleration rates, the optimal travel time across that stretch of track is calculated. Optimal travel time means that even if Vf is less than Vi, the train will be accelerated (if there is enough room and Vi is less than the maximum allowable speed) to a cruising speed and then decelerated so that its speed is Vf just as it reaches the end of the segment.

An example will be used to illustrate this method. Given a train which enters a 10 mile segment with initial velocity (Vi) equal to 40 mph, and it is desired that this train be travelling at 20 mph at the end of the segment (probably because it will have to stop at the end of the next, shorter segment). It is also given, that the maximum allowable speed on the segment is 50 mph, and the acceleration and deceleration rates are 360 mph_2 and -600 mph_2 respectively. One way to get the train across the segment is to travel at 40 mph for 9 miles and to decelerate, to 20 mph, over the last mile. This would take approximately 15.5 minutes. The method used in the simulation would have the train accelerate to 50 mph over the first 1.25 miles; travel at 50 mph for the next 6.75 miles; and decelerate to 20 mph over the last 1.75 miles. This trip would take approximately 12.8 minutes. While it is not necessarily realistic, using the latter method for all cases at least provides consistency in determination of trip times, and programming it is a lot easier than deciding which trains to optimize and which trains to send along at minimum possible speed.

4.4.1.2 Slow Orders

Slow orders are an integral part of modelling track maintenance and track defects. Slow orders are placed over a section of track for a variety of reasons. Poor geometry, a rail defect, a cluster of broken ties, or recently surfaced or renewed ballast are several of these reasons. In any case, the simulation would not be complete without the ability to model slow orders. When a train encounters a slow order, it must slow down to the reduced speed; travel at this speed until the tail end of the train clears the slow order zone; and then accelerate back up to normal speed. Using the equations of motion, the simulation calculates the slow down distance, slow order distance, acceleration distance, and the time to cross this entire distance. For a train of length 1 mile, with a normal travelling speed of 40 mph, crossing a 2 mile, 10 mph slow order zone would require the following steps: deceleration to 10 mph over 1.25 miles; travel at 10 mph for 3 miles (the 2 mile slow order zone + 1 mile until the tail of the train clears this zone); and acceleration back up to 40 mph over 2.08 miles. The total time to cross the 6.33 miles is 26 minutes. If the train had crossed this 6.33 miles at full speed, it would have taken 9.5 minutes. This extra 16.5 minutes are caught using the new method of calculating travel times.

4.4.2 Preventing Line Blockage

When a train reaches the end of a segment, a decision has to be made whether to dispatch that train onto the next segment, or to hold it at its current position. If the train is at the end of a double track segment, and the next (single track) segment is not yet free, the train must be held. A check must still be performed to ensure that moving the train "does not result in a situation where no trains on the line can move further unless some train (or trains) reverse" (Dontula, 1991). Such a situation is known as line blockage. In a simulation model which has no provisions for backing trains up, line blockages must be avoided. The line blockage check used by Dontula consisted of checking the next two segments beyond the current position of the train. The only situation in which a train is held at the end of a double track segment in order to prevent line blockage is depicted in figure 4.1 for an outbound train. As seen in the figure, the train on segment (or siding) i will only be held if the segment or siding on i+2 is occupied by a train in the same direction. A train at the end of a single track segment is never held to prevent line blockage, since holding it would prevent any trains from passing it, thereby resulting in line blockage!

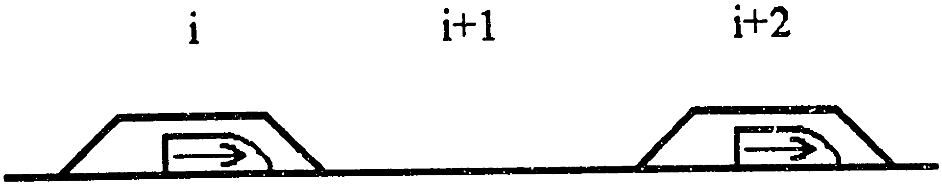


Figure 4.1: Preventing line blockage

4.4.3 Dispatching a Train From the End of a Segment with a Siding

When a train reaches the end of a double track segment (i), and it has been confirmed that the next segment is free (occupied by neither a train nor a maintenance crew) and that line blockage will not occur if it is dispatched, the dispatch choice algorithm is entered. On a real line, there would be signals positioned approximately every 2.5 miles (Thelen and Tse, 1990). These signals are basically traffic lights. The signal green when the track is clear for several blocks, yellow when a train should begin slowing down in order to avoid a collision, and red when movement into the next signal block is prohibited. Since in the simulation, there is no limit on how long segments between sidings can be, the dispatching algorithm looks ahead at the next 5 segments (i+1 through i+5, for an outbound train) in order to determine the speed that the train should have when it reaches the end of the next segment (i+1). In the look ahead the algorithm looks for trains, train directions, train priorities and the dispatch times of any trains occupying these segments and sidings. It determines whether or not the train will be held for a higher priority train. It also checks whether or not maintenance crews or slow orders are in place on i+1 and i+2, and the end times of these activities. Using this information, the algorithm determines the probable speed at which the train on i will be travelling when it reaches the end of i+2, and whether or not it will probably be dispatched onto the siding of i+2. From this information, the algorithm determines the maximum speed at which the train can be travelling when it reaches the end of segment i+1. If the train is not held for a higher priority train, the algorithm returns this speed. If the train is held, the algorithm returns the time at which the higher priority train should clear segment i+1, thereby allowing the train currently on i to move onto it. The Figures 4.2a and 4.2b describe the process of dispatching a train from a segment with a siding. The dispatch choice algorithm for this case is described in Appendix B.

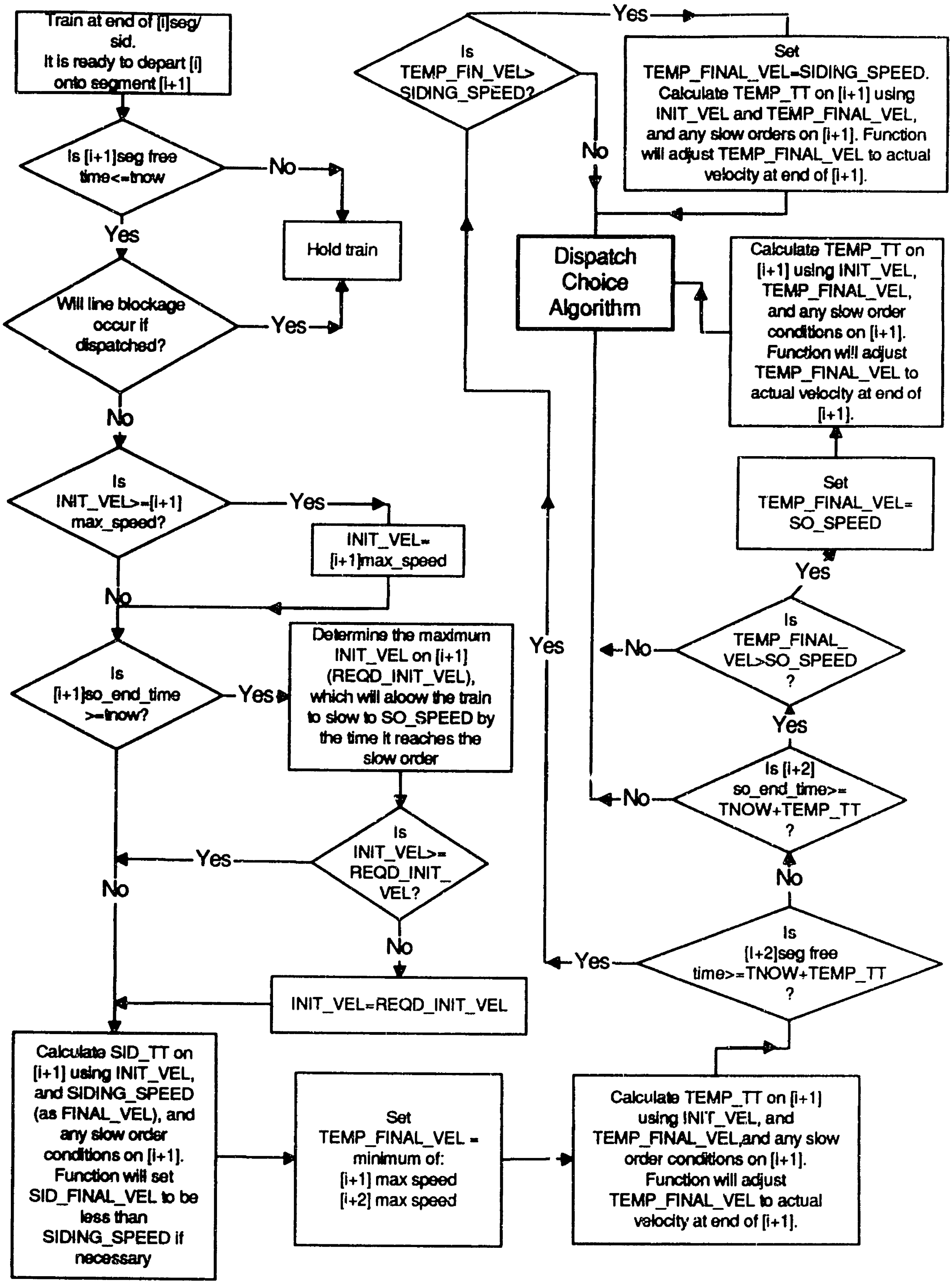


Figure 4.2a: Dispatching a Train From a Segment With a Siding

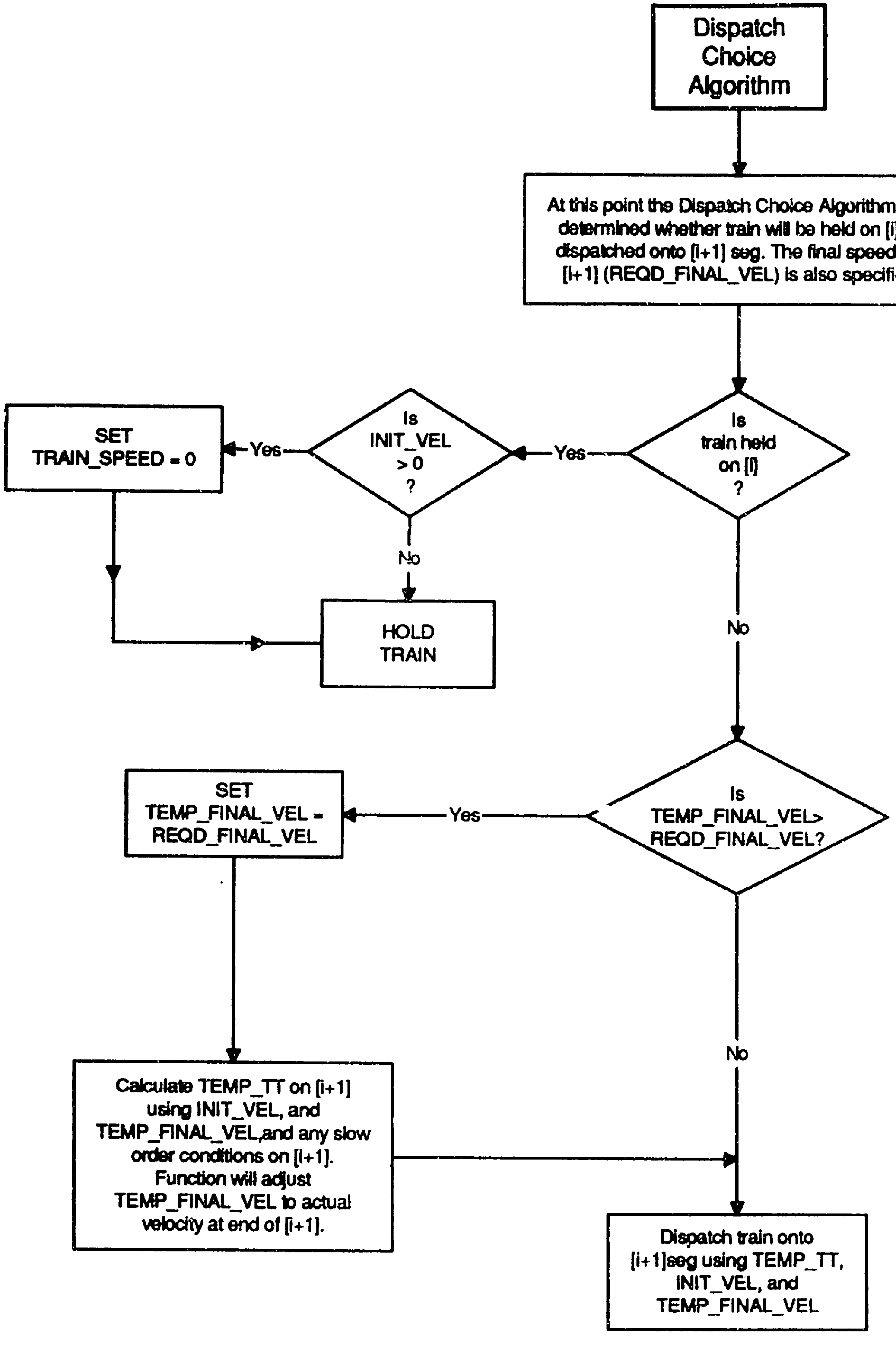


Figure 4.2b: Dispatching a Train From a Segment With a Siding

4.4.3.1 Prioritizing Trains

The addition of the dispatching algorithm to the model provided an opportunity to include a limited check ahead for the purpose of giving higher priority trains precedence when determining which train should wait when two trains meet on the line. When a train reaches the end of a segment with a siding (i), and if it is clear to be dispatched onto i+1, the dispatching algorithm will check the segment and siding of i+2 to see if either one is occupied by a train in the opposite direction with a higher priority than the train on i. If such a train exists, even if it has only just entered i+2, the program will hold the train on i until the higher priority train has travelled across i+1 and has moved onto the adjacent track on i. It would be possible, with some modifications to the dispatching algorithm to check three segments ahead for higher priority trains. If segment i+2 is unoccupied, the algorithm could check to see if there are any higher priority trains on i+3 which are due to move onto i+2 within a set time threshold, say 10 minutes. This would be closer to what a dispatcher would actually do than the current limited check.

4.4.4 Dispatching a Train From the End of a Single Track Segment

When a train reaches the end of a single track segment (i) it will be dispatched onto either the siding or segment of (i+1). The line blockage check is not necessary for reason detailed in section 4.2. A dispatch check algorithm, similar to that for a train at the end of a double track segment, is entered. The algorithm looks ahead at segments i+1 through i+4 (when the train on i is outbound). After collecting the same information as the other dispatching algorithm, two decisions are made. The first is whether or not to dispatch the train onto the siding or segment. If either is occupied, the train is dispatched onto whichever is unoccupied. If neither is occupied, the decision depends on the existence of trains in the opposite direction on segments i+2 through i+4, and on the priorities of these trains. Usually, if there is a train in the opposite direction on i+2 or i+3, the train currently on i will be held at the end of the siding of i+1. The other decision is the required speed of the train at the end of i+1. This is usually dependent upon whether the train is dispatched onto the siding or segment. Figures 4.3a and 4.3b describe the process of dispatching a train from a single track segment. Appendix B. contains a description of the dispatch choice algorithm.

4.4.4.1 Redispatching a Train (Defacto Signalling)

It is always possible that by the time a train reaches the end of a single track segment (i), changing conditions on the next four segments while it is in transit, will result in its final speed on i (the initial speed on i+1) being too high. Either the train has been instructed to move onto the siding, and its speed is greater than the siding speed, or its speed is too high for it to decelerate to the required final speed at the end of the next segment or siding, within the length of i+1. This could be due to the imposition of a maintenance crew or slow order on i+2 or i+3, or the unexpected breakdown of a train, or simply because the dispatch choice

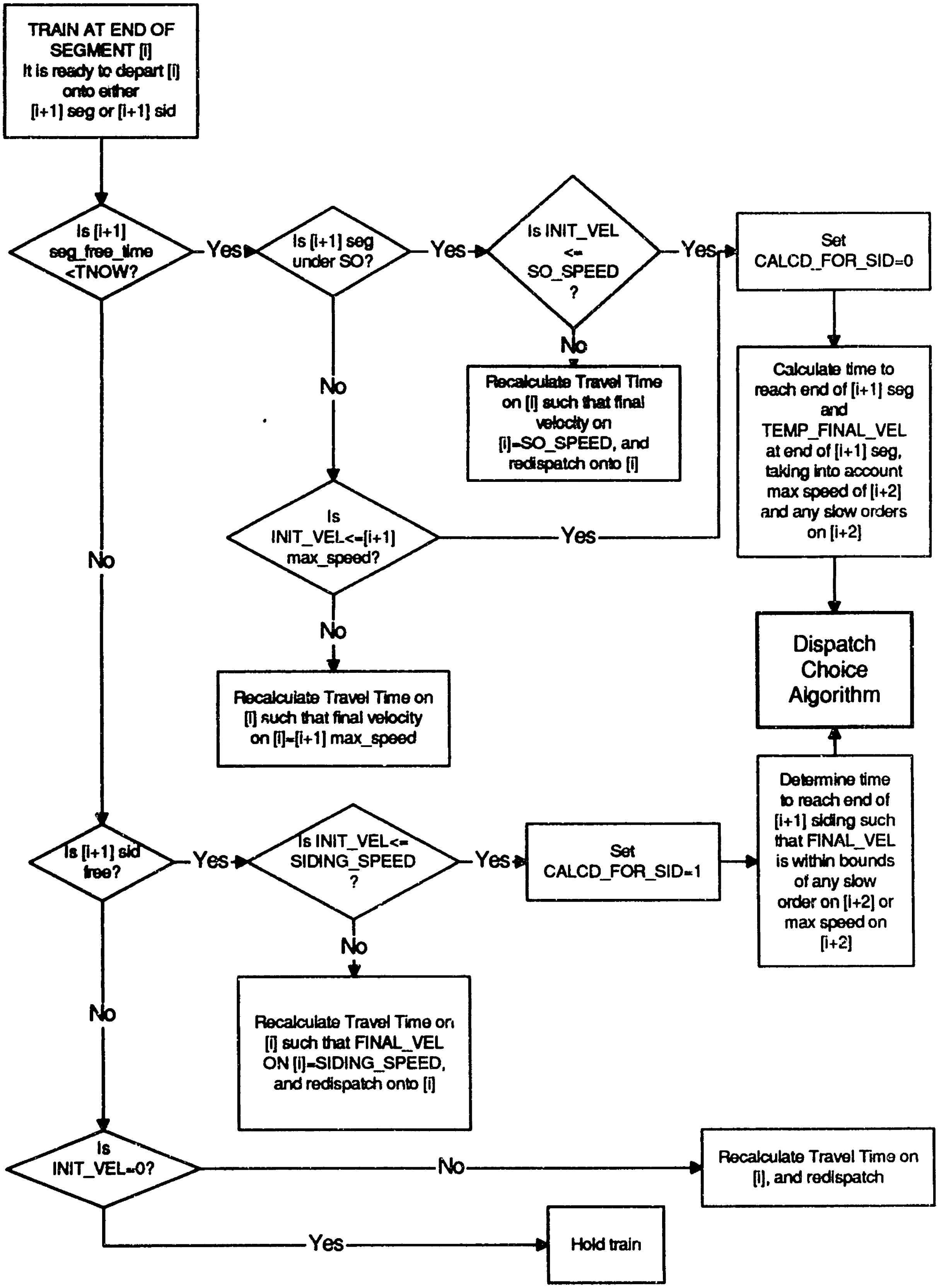


Figure 4.3a: Dispatching a Train From a Single Track Segment

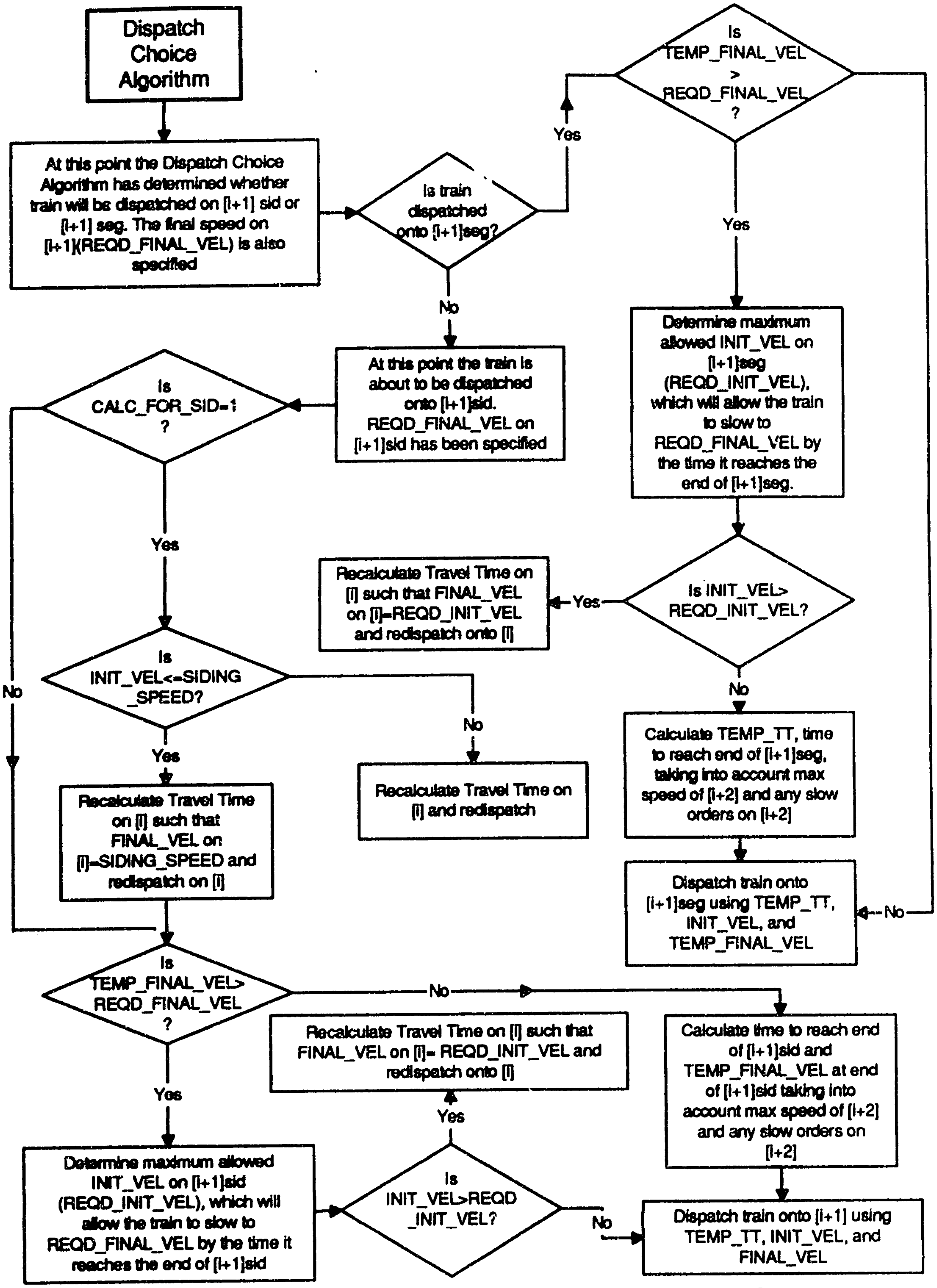


Figure 4.3b: Dispatching a Train From a Single Track Segment

algorithm predicted the wrong final speed when it was dispatched onto i. The lack of signals every couple of miles makes this a possibility. In order to handle this case when it occurs, a function has been written which will recalculate the time the train would take to cross the segment if its final speed is corrected. The result is what the author likes to refer to as "defacto signalling". This function basically makes up for the lack of a formalized signal system. The end result is the same, and fewer checks performed than if a signal was placed every 2.5 miles.

4.4.4.2 Avoiding Line Blockage due to Maintenance Events on Segments With Sidings

A potential line blockage will arise whenever a maintenance event requires the closing of one half of a double track segment. For example, a siding i+1 could be occupied by an inbound train which is waiting for an outbound train on i to move onto i+1's main segment. If a maintenance event closes down the main segment of i+1 before the outbound train reaches it neither the outbound train nor the inbound train would be able to move off of their respective segments. In reality, the maintenance crew would wait for the outbound train to pass onto i+2 before moving onto the line, or it would clear off in order to let the outbound train pass. The model approximates this by moving the outbound train across i+1's main segment at a slow order, and subsequently increasing the maintenance completion time (i+1's free time) by the travel time of the outbound train across the segment.

4.5 Program Description

The simulation model design detailed in sections 4.2, 4.3, and 4.4 was implemented in the C programming language using a Turbo C compiler. This section contains a description of the three major data structure types, and all of the major functions including the control function.

4.5.1 Data Structures

The main data structures that are used in this program are a doubly linked list, and two arrays of structures. The structures defined in the program are TRAIN, SEGMENT and PLANNED_EVENT. The TRAIN and SEGMENT structures were included in the original program. Variables that were added to these structures are italicized.

Structure TRAIN

```
int     id_number;   /* The id number of a train if available */
int     direction;   /* denote inbound or outbound of a train */
int     location;    /* indicate the segment number */
int     fail_count;  /* counts the # of failure occurred to a train */
double  train_length; /* in terms of miles */
int     priority;    /* priority attached to train, if any */
double  speed;        /* speed of a train on a segment, can be different
                         from the regular speed (defined below) if speed
```

```
                    restrictions are imposed on a segment */
double   norm_speed;   /* normal speed train would follow if not
                    restricted to follow lower speed   */
double   dep_time;     /* time train departs station*/
double   arr_time;     /* time train arrives at destination yard */
double   cum_delay_ops; /* cumulative delay due to operations such
                    as meet/pass delays */
double  cum_delay_eng; /* cumulative delay due to engineering failures */
double  cum_delay_dep; /* cumulative delay incurred by train when
                    attempting to depart a station */
double failure;        /* time train was delayed on current seg or sid
                    due to train failure */
double last_init_vel;  /* initial velocity of train when it was dispatched
                    onto its current segment */
double last_disp_time; /* time at which train was dispatched onto current
                    segment */
int    prev_on_sid;    /* 1 if train was previously on a siding */
                  /* 0 if train was previously on a segment */
int    prev_so;        /* 1 if train was affect by a SO on the segment
                    it occupied previously */
                  /* 0 if not */
```

```
int    so_at_last_disp;     /* 1 if SO was in effect when the train was
                               dispatched onto the segment it occupies */
                            /* 0 if not */
double ideal_tt;         /* amount of time it would take train to cross
                            the line without slowing down or stopping */
int    priority_del;   /* IF the train is held for priority reasons,
                          the priority of the train for which it was held*/
int    maint;          /* 1 if train failed while travelling on current
                          segment; 0 if not */
struct train *next; /* pointer to next train in the list of trains */
struct train *prev; /* pointer to previous train in train list */
```

Structure SEGMENT

```
int      id_number;
int      train_id1;  /* id # of train on the segment  */
int      train_id2;  /* id # of train on the siding    */
double   seg_length; /* length of the segment in miles */
double   max_speed;  /* maximum speed allowable on a segment */
double   segment_free_time; /* time when the segment will be free */
double   siding_free_time;  /* time when the siding will be free
                               (only if it exists) */
```

```
int     seg_status;      /*0 if free 1 if it is occupied or busy */

int     siding;          /*0 if it exists 1 if it does not  */

int     siding_status;  /*0 if free 1 if it is occupied or busy */

double seg_dispatch;    /* when a train is dispatched onto a segment  */

double sid_dispatch;    /* or a siding, the time the train will reach */

                /* the end of the seg or sid will be tnext.   */

                /* ____dispatch is the time of the next        */

                /* attempted dispatch that seg or sid          */

                /* (i.e. it is set equal to tnext             */

double so_end_time; /*indicates the time current/last slow order end/ended*/

int maint;          /* 0 if no maintenance on seg */

            /* 1 if maintenance is taking place on seg */

            /* its purpose is to allow a train on a siding to check

              if the seg parallel to it is out of commission when

              deciding whether or not to dispatch it      */

double so_ob, so_ib; /* endpoints of slow order on the seg, relative to the

                 outbound and inbound ends of the line respectively

                 a so 3 miles long starting 5 miles from the ob end

                 would have so_ob = 5, so_ib = 8 */
```

The following structure was created to hold and provide easy access to the data obtained from the planned maintenance event input file.

Structure PLANNED_EVENT

```
int    segment_index;  /* index number of segment on which
                          the maintenance event is to occur */
double event_time;     /* time the event is to occur  */
double maint_length;   /* length of time (in minutes) the maintenance
                          activity is to last */
double  so_length;     /* length (in minutes) slow order will be in
                          in effect after maintenance ends */
double so_ob, so_ib; /* endpoints of slow order on the seg, relative to the
                        outbound and inbound ends of the line respectively
                        a so 3 miles long starting 5 miles from the ob end
                       would have so_ob = 5, so_ib = 8 */
```

The following description of the manner in which the structures were manipulated is provided by Dontula (1991).

> A double linked list is used to keep track of trains on the line-haul as well as the events that are to be processed. The trains are linked together in a train list. Whenever a train is generated by a train departure from the origin event, the train is attached to the existing double linked list by manipulating the pointers. Whenever a train arrives at its destination yard the train is detached from the doubly linked list again by manipulating tea pointers. the details regarding the manipulation of pointers can be found in any text book on the C programming language. Similarly, all the existing events are linked together as a double linked list using an event list. The events are created whenever a train departs from tea origin, departs from the

segment, or a track maintenance activity takes place. Whenever a new event is created it is attached to the event list by manipulating the pointers. The main difference between this list and the train list is that the event list is a priority queue. The weight used in the prioritization is the time of occurrence of the event. The event to be processed is picked up from the front of the priority queue. Once the event is processed, the event is removed from the event list. An array of structures is utilized to store the segment information. The size of the array is set to the number of segments on the line-haul [including the stations at both ends]. (Dontula, 1991)

4.5.2 Description of Functions

The remainder of this chapter contains descriptions of some of the important functions in the simulation.

simulate()

This function controls the flow of the program. After the completion of each event, simulate() picks up the event in the front of the priority queue and calls the function which corresponds to the event. The functions it calls are trn_depart_stn(), trn_depart_segment(), create_track_maint_events(), track_maint_failure(), track_maint_planned(), and remove_maint_indicator(). Every time simulate() picks up an event, it compares the time the event is to occur with the defined simulation duration. When an event time is greater than the duration, the simulation ends.

trn_depart_stn()

This function creates trains and places them on the line. When called, this function creates a train structure, intitializes its variables, and places the train in the train list. The function then calls dep_sid_seg() (a station is treated as a segment with infinite sidings) to determine whether or not the train will be able to move out onto the line. If the train is successfully dispatched, trn_depart_stn() determine the time of the next departure from that end of the line from the information obtained from the schedule input files (see section 4.2.1), and a train depart station event is placed in the event list according to the time of departure. If the train is held in the station, it will be removed from the train list, and the time of the next departure attempt is set to the time at which dep_sid_seg() predicted the problem would clear up. A train depart station event is then placed in the event list according to this time.

trn_depart_segment()

This function is called whenever a train, which is already out on the line, attempts to depart from either a single or double track segment. The procedure is described for an outbound train. When trn_depart_segment() is called, it first checks to see if the next segment is M+1 (for an M segment line, M+1 is the inbound station). If the train is at the end of the line, it is removed from the train list, and all of the pertinent information stored in its structure is gathered into a set of arrays. If the train is not at the end of the line, the function then calls move_trn(), which determines whether the train currently occupies a single or double track segment, and takes appropriate action.

move_trn()

This function was written as a means of condensing code that was previously contained in trn_depart_segment(). This function is called by trn_depart_segment() whenever a train is attempting to depart a segment, and it has not reached the end of the line. The first action taken by move_trn is to determine whether the train is currently on a single or double track section. If on a single track segment and both the next siding and next segment are not free, the train is held until one of them is clear. If either (or both) is free, the function dep_segment() is called. If on a double track segment and the next segment is not free, the train is held at its current location. If the next segment is free, the function dep_sid_seg() is called.

dep_segment()

This function is called whenever a train is attempting to depart from a single track segment, and either the next siding, the next segment or both is free. dep_segment() calculates the fastest possible trip time across the next segment if it is free, or across the siding if the segment is occupied. The function disp_choice_fr_segment() is called to determine whether the train is to be moved onto the next siding or next segment, and the train's final speed when it reaches the end. The time at which it reaches the end of the next siding or segment is then used to enter a train-depart_segment event in the event list priority queue.

dep_sid_seg()

This function is called whenever a train is attempting to depart from a double track segment, and next segment is free. dep_sid_seg() then calculates the fastest possible trip time across the next segment for both the case where it moves onto the following siding or it moves onto the following segment. The function disp_choice_fr_sid_seg() is called to determine whether the train is to be dispatched onto the next segment or held in its current position. Reasons that the trains could be held are that moving it would result in line blockage, or for a higher priority train. If the train is dispatched, disp_choice_fr_sid_seg() returns train's final speed when it reaches the end of the single track segment. The time at which it reaches the end of the next segment is then used to enter a train-depart_segment event in the event list priority queue.

disp_choice_fr_segment()

This function is called by dep_segment to determine whether to move a train at the end of a single track segment onto the next siding or the next segment. The dispatch choice algorithm used is described in Appendix B. If the train is dispatched, this function returns where it moves, the final speed at the end of the next siding or segment and the time at which it reaches the end.

disp_choice_fr_sid_seg()

This function is called by dep_sid_seg to determine whether to move a train at the end of a double track segment onto the next segment, or to hold its position. The dispatch choice algorithm used is described in Appendix B. If the train is dispatched, this function returns the final speed at the end of the next segment and the time at which it reaches the end. If the train is held, this function returns the time at which it will next attempt to depart.

redisp()

When a attempts to depart from a single track segment, the possibility exists that its final speed (as determined by disp_choice_fr_sid_seg() when it was last dispatched) is too high. Possible reasons for this are described in section 4.4.4.1. This function obtains information about the train's last departure from its structure, and together with the new required final speed, determines the time at which it will arrive at the end of its current segment if it had been dispatched properly the first time.

create_track_maint_events()

This function is called at the beginning of each day (every 1440. minutes) for which the simulation is running. It uses the data from the random maintenance input file to generate track failures and the times of these failures. It also determines which events from the planned maintenance input file are to take place during the next day. create_track_maint_events then places all of these events into the event list according to the time they occur.

track_maint_failure

This function is called whenever a randomly generated track failure occurs. The maint-failure-event contains the time and type of failure that occurred. The probability of track closure, and the average length of the down time or slow order depends on the type of failure. Once this information is decided upon, the procedure described in section 4.3.1 is used to determine the length of down time of the track and the length of any slow order which follows it. Slow order are always placed in the middle of the line. The segment free time is incremented by the calculated down time, and the slow order end time is set equal to the sum of the segment free time and the slow order length. The segment structure variable *maint* is incremented by one, and a remove maint indicator event is entered in the event list according to the maintenance end time (the segment free time).

track_maint_planned()

This function is called whenever a planned track maintenance event occurs. The maint-planned-event contains the time of the event and the index number of the planned_event array at which the activity information is stored. The segment on which the activity occurs, the length of the down time and slow order, and the position of the slow order within the segment are contained at this address. Once this information is obtained, the segment free time is incremented by the down time, and the slow order end time is set equal to the sum of the segment free time and the slow order length. The segment structure variable *maint* is incremented by one, and a remove maint indicator event is entered in the event list according to the maintenance end time (the segment free time).

remove_maint_indicator()

The generation of random track failures means that it is possible for a segment to be undergoing two different maintenance activities at the same time. This makes using a boolean (0 or 1) variable impractical since the program would have no way of knowing how many activities are going on, and once the first one ends, the boolean would be set to zero. Instead, the variable *maint* in each segment structure, is incremented by one whenever maintenance begins on a segment. The function remove_maint_indicator is called when maintenance activities are predicted to end. This function reduces *maint* by one for the segment contained in the event structure.

rem_trck_from_elist()

This function removes a track maintenance event from the event list.

The descriptions of the remaining functions are reproduced from Dontula (1991).

put_in_eventlist()

This function takes as input the event number, train id number, and the time of occurrence of the event. It creates a new event structure. Using pointer manipulation techniques, the event is inserted at the appropriate location. As specified earlier, the weight used in the process of insertion of the event is the time of occurrence of the event.

remove_from_eventlist()

This function removes an event that has already been processed from the event list. Since this event is always located at the top of the event list priority queue, it is easy to remove the event.

remove_from_trainlist()

When a train reaches its origin or a train cannot be dispatched from the originating yard we remove the train from the train list. So its location in the doubly linked list is searched and once the appropriate structure is found using pointer manipulation the train is removed from the train list by pointer manipulation.

create_train_failure()

This function along with a function that creates uniformly distributed random variate checks to see if a train failure occurs on a segment. If a failure does occur, it creates the repair time by using an exponential distribution. The repair time distribution though can be changed quite easily.

Chapter 5 Case Study

5.1 Introduction

In this chapter, the methodology described in chapter 1 is used to evaluate the effects that increasing axle loads of trains will have on yearly maintenance requirements of a generic line, and what this means for the traffic crossing the line. Section 5.2 describes the traffic consist and the initial route conditions that were used as inputs for TRACS, and how this information was translated to schedule and route input files for the simulation model. In section 5.3, the outputs from the TRACS analyses are discussed, and the defect rate input files for the simulation are developed. Section 5.4 is comprised of a few studies using the simulation model. Section 5.5 is a summary of results.

5.2 Inputs to the Models

The traffic consist, number of trains per week and the number of cars per train, was determined using information developed for the A.A.R.'s economic analysis of Heavy Axle Loads (Martland, 1991). The initial condition of the track as well as the breakdown of the route by curvature were obtained from the same source. The use of this information is described in the following sections.

5.2.1 Traffic Characteristics

5.2.1.1 TRACS

The major inputs required by TRACS were the annual amount of traffic expressed in MGT (million gross tons), and a percentage breakdown of this tonnage by axle load. In other words TRACS requires the annual tonnage carried by each axle load used. The lines looked at were coal lines, with loaded trains travelling in one direction and empty trains returning in the opposite direction. Analysis was performed for high density (80 mgt) and low density (30 mgt) traffic. At each traffic level, runs were performed in which the loaded trains used cars with 33 ton (the base case), 36 ton, and 39 ton axle loads. It was decided to carry the same annual net tonnage of cargo for each case. For this study the number of cars per train was held relatively constant, thereby allowing the heavy axle load cases to carry the same net tonnage of cargo with fewer annual trains. The other option would have been to run the same number of trains while reducing the number of cars per train, thereby achieving the same annual net tonnage.

Table 5.1 provides the methodology used to obtain the number of trains per week, and the actual annual mgt for each case. The mgt/year was set at 80 and 30 mgt for the base cases. The values for the other cases were obtained using the following formula:

$$MGT_i = MGT_b * \frac{NAPT_b}{NAPT_i} \qquad (5.1)$$

Where MGT_b and MGT_i are the mgt/year for the base case and case i respectively, and $NAPT_b$ and $NAPT_i$ are the net tonnage as a percentage of total tonnage for the base case and case i, respectively.

	80 MGT			30 MGT		
Axle Load	33 ton	36 ton	39 ton	33 ton	36 ton	39 ton
Net tons/train	10774	11994	13086	10774	11994	13086
Trailing tons (Wt of loaded cars)	13939	15158	16695	13939	15158	16695
Wt of empty cars	3165	3164	3609	3165	3164	3609
# locomotives	5	5	6	4	4	5
tons / locomotive	180	180	180	180	180	180
loco tons / train	900	900	1080	720	720	900
Wt of loaded train	14839	16508	17775	14659	15878	17595
Wt of empty train	4065	4064	4689	3885	3884	4509
Tot. tons/roundtrip	18909	20122	22464	18544	19762	22104
Net as % of total	0.570	0.596	0.583	0.581	0.607	0.592
MGT/year	80.00	76.44	78.28	30.00	28.72	29.44
# loaded trains/week	81.38	73.05	67.01	31.11	27.95	25.61
# empty trains/week	81.38	73.05	67.01	31.11	27.95	25.61

Table 5.1: Determination of number of trains per week

It should be noted that the annual mgt is lower for the HAL cases than in the base case. This is because fewer cars were used to carry the same net weight. For the 39 ton axle load cases a bigger wheelset was necessary to support the heavier load, resulting in a heavier empty car, and an annual mgt between the 33 and 36 ton axle load cases. The number of trains per year in each direction was determined by dividing the annual gross tonnage by the tonnage per round trip. This was divided by 52 (weeks/year) to obtain the number of roundtrips per week. Table 5.2 shows the percent of total mgt carried by each axle load. This was determined by dividing the total weight of that type of car (be it empty, loaded or a locomotive) by the total tonnage in a roundtrip. Empty cars have axle loads of 8 tons, and the locomotives used imparted axle loads of 30 tons. The axle load from the loaded cars depended on the case.

Weight of Car (tons)	Axle Load (tons)	80 MGT			30 MGT		
		33	36	39	33	36	39
32	8	16.74%	15.72%	16.07%	17.07%	16.01%	16.33%
180	30	9.52%	8.95%	9.62%	7.77%	7.29%	8.14%
132	33	73.74%	0%	0%	75.17%	0%	0%
144	36	0%	75.33%	0%	0%	76.70%	0%
157	39	0%	0%	74.32%	0%	0%	75.53%

Table 5.2: Percent of total mgt carried by each axle load

5.2.1.2 Simulation

Creating schedules for the simulation model required the number of trains that traveled in each direction each week. These values are shown in Table 5.1, and the method of calculation was described in the previous section. Rather than dispatching the same number of trains per day over the three weeks that the simulation was to run, it was decided to take advantage of the option to dispatch trains at different offsets for each day of the week. (See section 4.3.1 for details). After reviewing a study which included typical peak and off peak traffic loads for a Major North American Railroad, it was decided to schedule 67% of the weekly traffic over 5 off peak days, with the remaining 33% of traffic being dispatched over the other two days of the week (Martland, 1976). For example, in the 80 mgt, 30 ton axle load case, there were 162 trains per week. This was separated into 108 off peak trains, and 54 peak trains. On Sunday through Wednesday, 22 trains were scheduled per day (11 in each direction). On Thursdays and Fridays, the peak days, 26 and 28 trains were dispatched respectively. Saturdays saw 20 trains dispatched. The number of trains dispatched daily for each case is detailed in Table 5.3.

5.2.2 Route Characteristics

5.2.2.1 TRACS

The route characteristics required for the creation of TRACS route input files, which are pertinent to this study, are: the length of the line, the percentage

CASE	S	M	T	W	Th	F	S	TOTAL
80 33	22	22	22	22	26	28	20	162
36	20	20	20	20	24	24	18	146
39	18	18	18	18	22	22	18	134
30 33	8	8	8	8	10	12	8	62
36	6	8	8	8	10	10	6	56
39	6	6	6	8	10	10	6	52

Table 5.3: Scheduled number of trains per day

Curvature & Metallurgy	80 MGT (miles)	30 MGT (miles)
Tangent 270 BHN	24.24	145.42
Tangent 300 BHN	218.12	96.94
2 degree 270 BHN	0.0	23.36
2 degree 300 BHN	46.72	23.36
4 degree 340 BHN	2.92	2.92

Table 5.4: Length of track by metallurgy and curvature

of the total length at different curvatures, the quality of each track component (for example 270 BHN or 300 BHN rail), and the condition of the components (for example the amount of wear already existing on the rail before the analysis is to begin). The line created for use was 292 miles long. The percentages of different curvatures and metallurgies of rail for the different cases were obtained from the Martland memo, and applied to the 292 mile line. Table 5.4 lists the length of rail at each curvature and metallurgy for the 80 mgt and 30 mgt cases. It should be noted that the lower density line uses a smaller percentage of higher grade rail on tangent and 2 degree curves. As will be shown in section 5.3, the lower annual tonnage allows the use of lower quality materials without an increase in required maintenance when compared to the 80 mgt case. The ties, ballast and turnout types selected were in keeping with those detailed in the HAL economic analysis.

5.2.2.2 Simulation

As mentioned in the previous section the line used was 292 miles long. For the purpose of simulation it was divided into 21 single track segments of length 12 miles separated by 20 segments, with sidings, of length 2 miles. A graphic representation is provided in Figure 5.1. The maximum speed on the mainline was set at 40 mph. The normal operating speed of all trains was also set to 40 mph. The maximum speed on sidings was set at 30 mph, a typical maximum speed for travel through turnouts (Hay, 1982). The slow order speed was set at 10 mph.

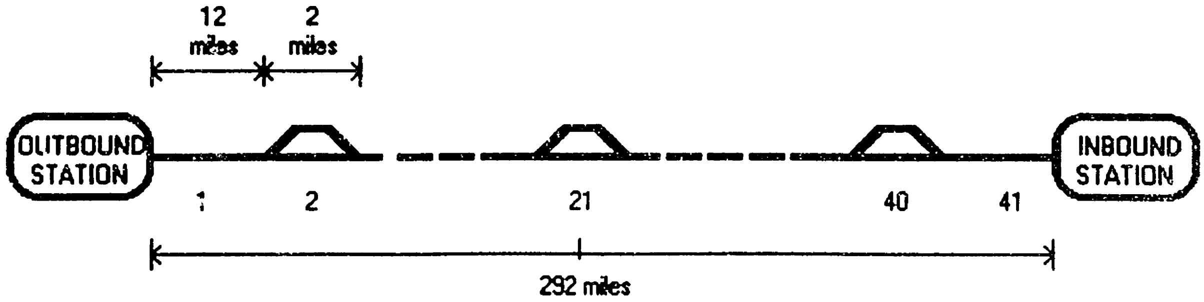


Figure 5.1: Line used in simulation

5.3 TRACS Output

The outputs of interest from TRACS were the average quantities of material replaced per year. These results are listed in Table 5.5. Multiplying these

	80 MGT			30 MGT		
Axle Load	33 ton	36 ton	39 ton	33 ton	36 ton	39 ton
RAIL						
avg mi. relayed/year	20.67	20.61	25.03	9.35	12.23	14.67
defects/mile/year	0.8	0.9	1.2	0.7	0.9	0.8
BALLAST						
avg mi. renewed/year	57.3	57.3	67.6	14.4	15.8	17.11
avg mi. surfaced/year	139.9	155.9	176.6	141.8	140.4	156.6
TIES						
avg. project ties/year	31745	32188	33147	33387	33411	33901
avg. spot ties/year	2310	2377	2540	2076	2059	2105
TURNOUTS						
avg. turnouts/year	3.2	4.8	6.4	1.6	1.6	1.6
avg. components/year	20.8	19.2	40	6.4	8	11.2

Table 5.5: Average yearly maintenance requirements for a 292 mile line

	HOURS OF MAINTENANCE					
	80 MGT			30 MGT		
Axle Load	33 ton	36 ton	39 ton	33 ton	36 ton	39 ton
RAIL						
relay	206.7	206.1	250.3	93.5	122.3	146.7
defects	466.4	524.7	699.6	408.1	524.7	466.4
BALLAST						
renewal	573.0	573.0	676.2	144.0	157.5	171.1
surface	699.6	779.5	882.8	708.8	702.0	783.1
TIES						
project ties	158.7	160.9	165.7	166.9	167.1	169.5
spot tie	11.5	11.9	12.0	10.4	10.3	10.5
TOTAL	170.3	172.8	177.7	177.3	177.3	180.0
TURNOUTS						
turnouts	22.4	33.6	44.8	11.2	11.2	11.2
components	62.4	57.6	120.0	19.2	24.0	33.6
TOTAL	84.8	91.2	164.8	30.4	35.2	44.8

Table 5.6: Average hours spent on maintenance per year for a 292 mile line

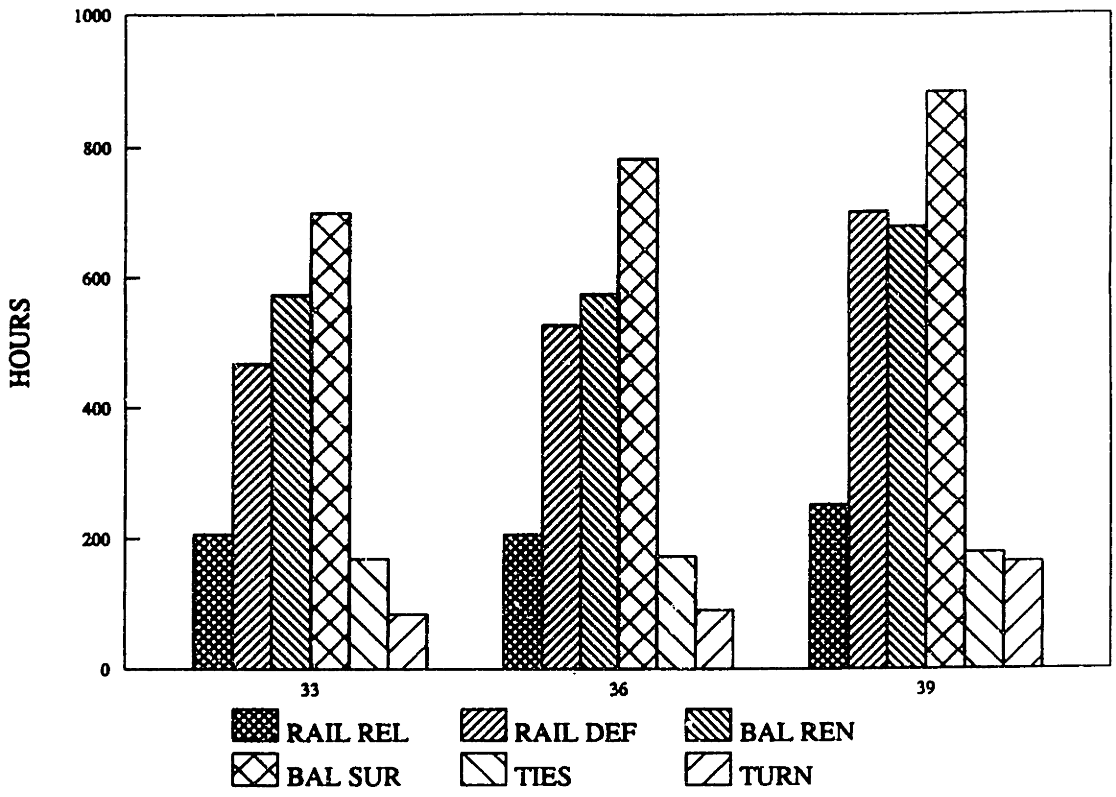


Figure: 5.2 Avg. hours spent on maintenance per year: 80 mgt line

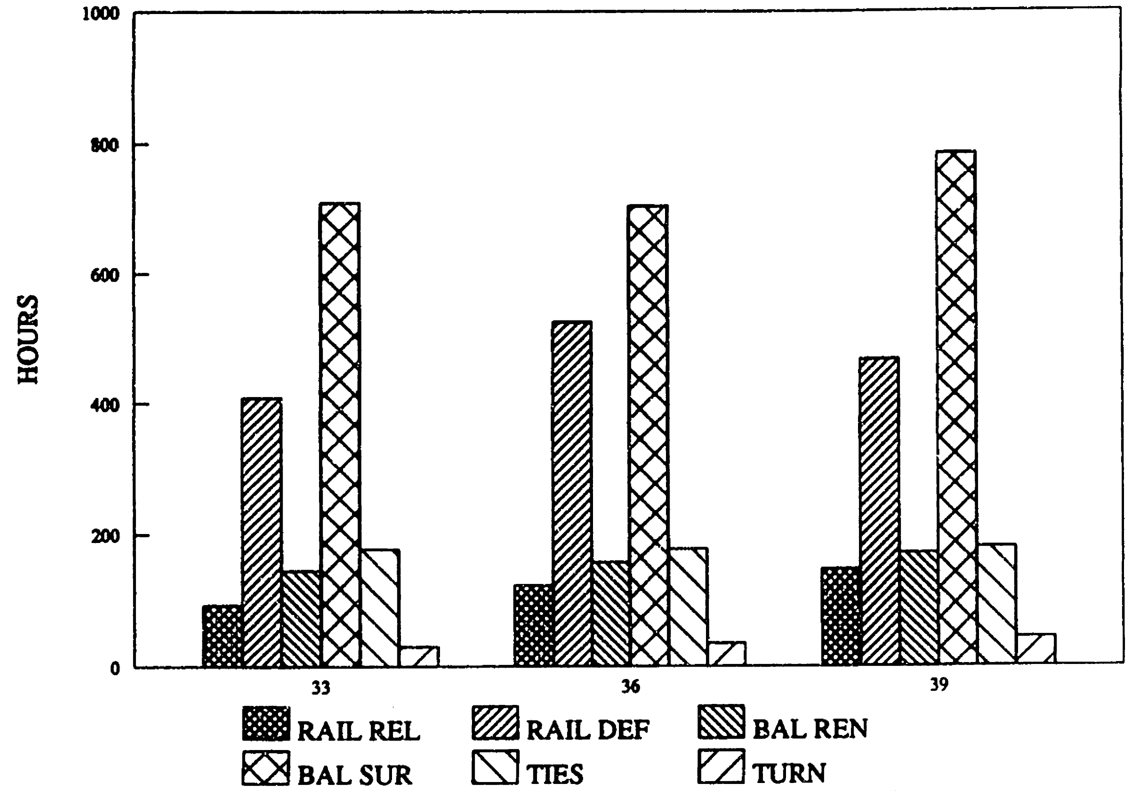


Figure 5.3: Avg. hours spent on maintenance per year: 30 mgt line

quantities by the productivity rates listed in Appendix A. yields the average hours spent on maintenance for rail, ballast, ties and turnouts. These quantities are listed in Table 5.6 and are represented in Figures 5.2. and 5.3.

The use of heavy axle loads definitely makes a difference in the amount of time required to maintain the line. Although, for the 80 mgt cases, the difference between the 33 ton axle loads and 39 ton axle load cases overshadows the difference between the 33 and 36 ton axle load cases. One other striking aspect of the graphs is the dominance of ballast work over all other work. This is due to the higher frequency of project maintenance that is performed (approximately half of the line is surfaced every year) in comparison to the other track components.

The use of this information in the simulation runs is described and demonstrated in the next section.

5.4 Simulation Model Analysis

5.4.1 Unscheduled (Spot) Maintenance

5.4.1.1 Determination of Defect Rates

The simulation of unplanned maintenance events on a line requires yearly average "defect" rates. Included in the output from TRACS are average numbers of rail defects per year, and the average number of spot tie replacements per year. The tie maintenance policy used in TRACS required a cluster of three failed ties before a spot gang would replace any of them, so the number of ties spot replaced per year was divided by three achieve the defect rate used in the simulation runs.

It has been reported (Davis, 1992) that over 75 percent of all rail service failures occur during the six coldest months of the year. To simulate the effect of seasonality, summer and winter defect rates were calculated by decreasing and increasing, respectively, the average defect rate by 50 percent. The resulting defect rates used for all cases are shown in Table 5.7. The tie defect rates were treated the same way. The resulting lower and higher defect rates can also be looked at as examples of lines in above and below average condition.

	Summer		Average		Winter	
80 MGT	RAIL	TIES	RAIL	TIES	RAIL	TIES
33	0.40	1.319	0.80	2.637	1.20	3.956
36	0.45	1.357	0.90	2.713	1.35	4.070
39	0.60	1.367	1.20	2.733	1.80	4.100
30 MGT						
33	0.35	1.185	0.70	2.370	1.05	3.555
36	0.45	1.175	0.90	2.350	1.35	3.525
39	0.40	1.202	0.80	2.403	1.20	3.605

Table 5.7: Spot maintenance events/mile/year

Not every defect occurrence will result in a line shutdown. Most of the time, traffic will still be able to pass over the damaged portion of track, albeit at reduced speed. For the purpose of this study, 40 percent of rail defects and 10 percent of tie defects resulted in track closure. In a comprehensive study of rail defects and their effects on maintenance, Davis reported on a track segment of predominantly unit trains which averaged 50-60 mgt/year over a total length of 225 miles. This line reported 72 service defects over the course of a year. Taking the average defect rate from the 80 mgt, 33 ton axle load case (0.8 defects/mile/year) multiplying by 225 miles and taking 40% of this product yields 72 service failures/year.

Given that the track is closed, the average down time for rail defects is four hours. For tie failures the average of track closure time is three hours. The down time represents a combination of travel time to the site and the time required to complete the repair. This time was varied by plus or minus 1 1/2 hours, using a uniform probability distribution, to represent different travel times. If a maintenance operation requires surfacing, such as tie replacements (AREA Railway manual) it will take a while, after completion of work for the track to return normal condition. Upon completion of the tie repairs, a slow order was placed over the defective section for a length of time which would allow a approximately ten trains to pass over the section. If the either type of defect does not result in a track closure, a slow order of average length 24 hours, plus or minus 4 hours, is imposed on the damaged section of track to simulate the time it takes to get a crew out for a "non-emergency" repair. In these cases, the repair crew will usually perform the repairs between trains, or will clear off of the track to let the trains pass through at reduce speed. For the purpose of modelling slow orders, all defects are placed in the middle of the section in which it is located, and the slow order is imposed one mile on both sides of the defect. The slow order speed was 10 miles per hour for all maintenance events.

A train failure rate of 25,000 mean miles between failure, with a mean repair time of 65 minutes was used in all cases except for the zero defects cases. Both the probability of defect and the train repair times were exponentially distributed.

5.4.1.2 Effects of Different Defect Rates

A set of five runs were performed for each axle load case using different random defect files. These runs were: zero defects, only train defects, summer (below average) defect rates, average defect rates, and winter (above average) defect rates. The information of interest resulting from each run were the average train delay, average trip time, and standard deviation of trip time. The results depicted in Figures 5.4 through 5.6.

The average delay includes delays experienced in the yard while attempting to get onto the line; delays due to train meets, and delays due to train failures and track defects. One element that is readily apparent is the influence of traffic density on the average delay, even when no defects occur. It is also apparent that the delays increase as the defect rate increases. For the 80 mgt line, delays due to track failure average 40 minutes longer for the combination of the low track failure rate and train failures than in the cases where only train failures were generated. This is an increase of approximately 60%. The additional delay experience in the average and high track defect cases were approximately 72 minutes (a 107%

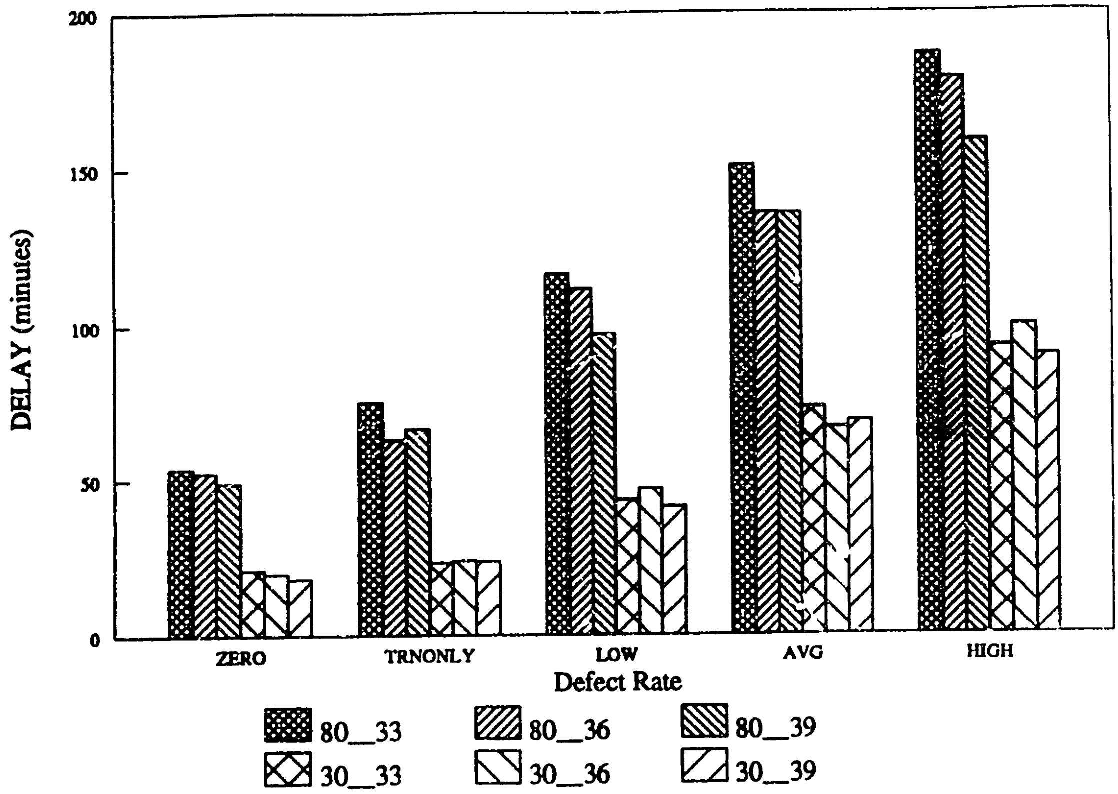


Figure 5.4: Average delay at different defect rates

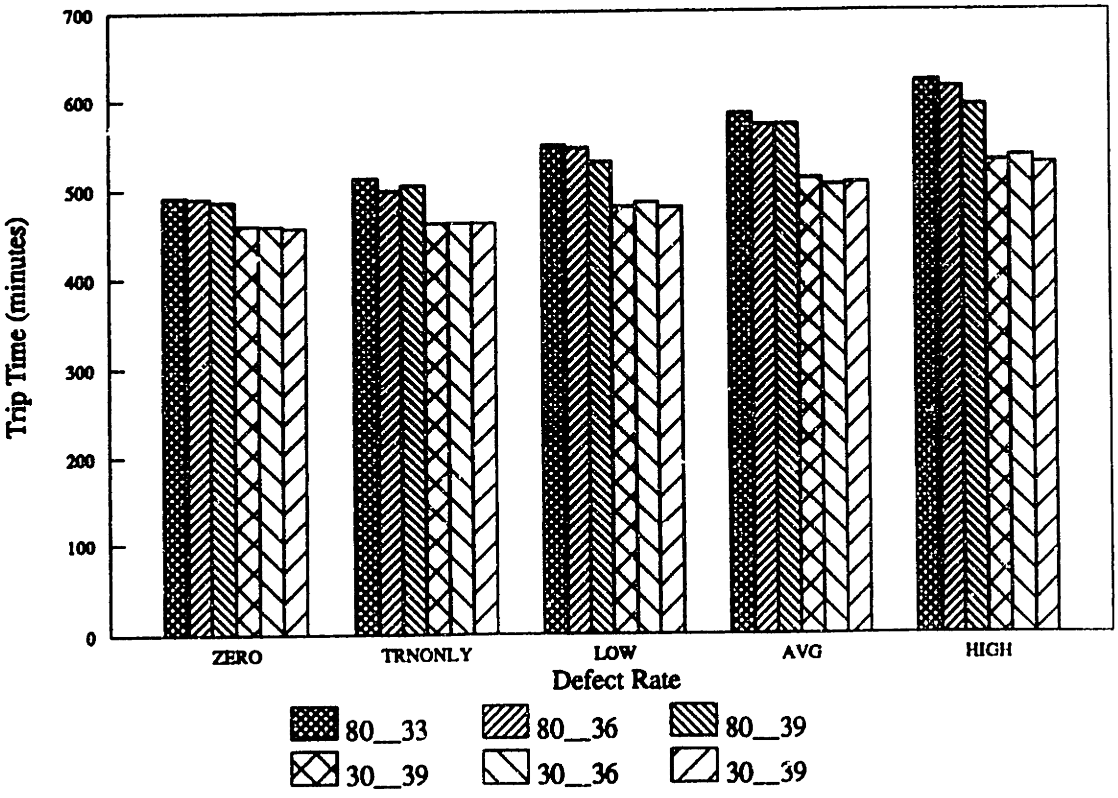


Figure 5.5: Average trip time at different defect rates

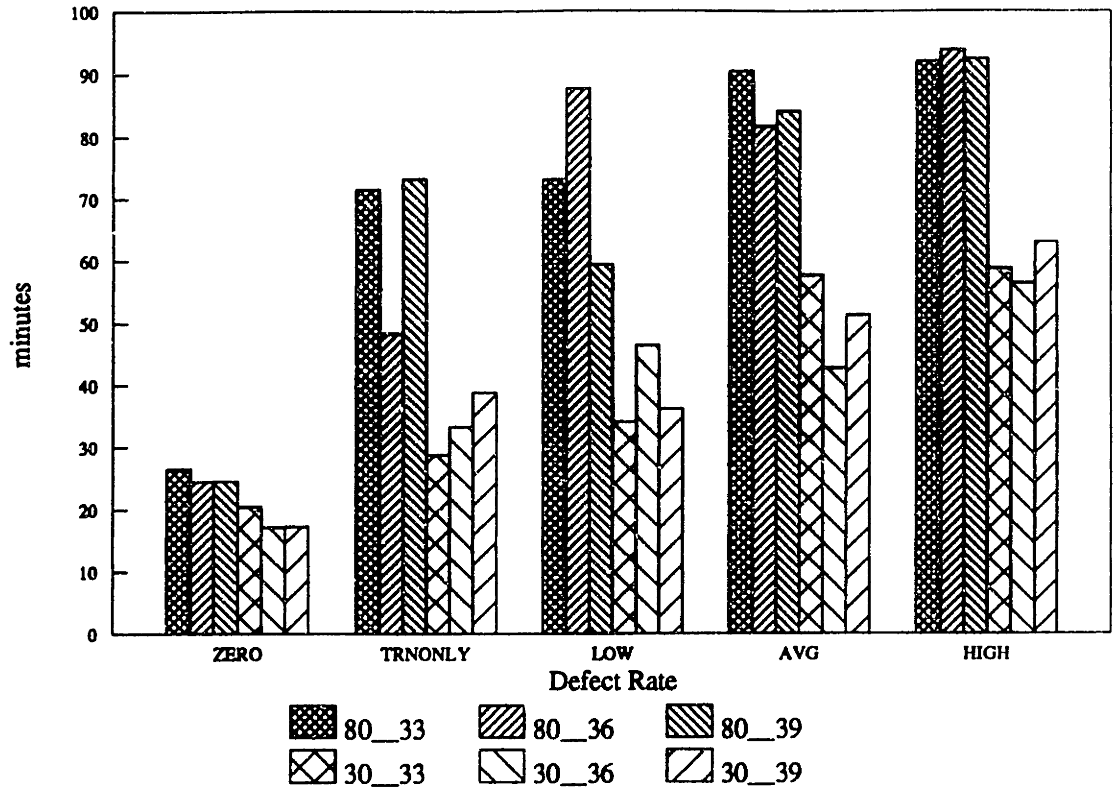


Figure 5.6: Standard Deviation of trip time at different defect rates

increase), and 107 minutes (a 157% increase over the train failure only cases). The 33 ton axle load case incurred greater delays than either the 36 or 39 ton axle load cases, even though its track defect rates were lower. In fact, whenever track failures were included, the average delays experience by the 33 ton axle load trains were consistently higher than those experienced by the 39 ton axle load trains. Apparently, the increase in defects due to heavy axle loads is offset by the flexibility gained in the schedule from the decrease in the number of trains required to carry the same net tonnage of freight.

If looked at independently from the 80 mgt results, the increases in delay due to track defects in the 30 mgt runs would appear to be more pronounced than they actually are. As they stand, delays for each case with track failures resulted in additional delays, over the train failures only cases, of approximately 20 minutes 45 minutes, and 70 minutes. This means that in an average case an extra three quarters of an hour is spent on the line, and in the worst case, and almost an extra hour and a quarter is spent on the line. For the 80 mgt cases, these extra times are one and a half, and almost two additional hours.

These delays must be kept in perspective with the average travel times achieved when unplanned track failures do not occur. In the 30 mgt cases, this trip time is 7.7 hours. The 80 mgt cases averaged a trip time of 8.4 hours when track failures did not occur. This means that for average track failure rates, 6% of the trip time on a 30 mgt line is time lost to track failures. For the 80 mgt line, 15% of travel times consists of additional delays due to failures.

A interesting result for both 80 mgt and 30 mgt, is that heavy axle load traffic did not have a significant effect on average total train delay. The increase in average yearly spot maintenance activities, and the ensuing track occupancy time for maintenance, was offset by the slack built into the heavy axle load schedules due to the lower number of trains required to carry the same net tonnage of freight.

Since trip time is the combination of ideal trip time and delays, the depiction of the average trip times are just a reflection of the average delays. Again, there is positive correlation between trip time and defect rate. For the average defect rates, trip times increased by approximately 70 minutes (14% of the train failures only case) and 45 minutes (10%) on the 80 mgt and 30 mgt lines respectively. For the 80 mgt 33 axle load case, trip time increased an average of 1.8 hours (21%) over the train failure only case when winter (above average) defect rates were used.

Average trip time and average delay are two important measures of how well a line is running. However, they do not give any indication of how reliable service is. For example, a line that offers 8 hour average trip times could have half of its trains crossing the line in 5 hours and the other half crossing the line in 11 hours. The standard deviation of trip time is one way to gauge the how reliable service is. Figure 5.6 exhibits the standard deviation of trip time for the cases. The effects of increasing defect rate are not as pronounced as they were for the delay and trip time, but there is a correlation. The results were not as clear cut as when comparing the average delays and travel times. With fewer trains on the line, the timing of a failure is more important when it comes to affecting the travel times of individual trains. This appears to be the case for the 30 mgt cases. Standard deviation of travel times did increase by approximately 100 percent for the average and above average failure rates, but the effect of the heavy axle loads is not immediately apparent except that they did not have a negative impact on reliability when compared to the 33 ton axle load case.

One way of looking at defects is that they limit access to portions of the track. This not only decreases the number of trains that can physically occupy the track, it also decreases the number of trains that can occupy the track before excessive delays due to train meets result. Essentially, track (and train) failures, decrease the capacity of the line. The closer the amount of traffic travelling across that line is to the capacity, the greater the impact of any defects will be.

5.4.1.3 The effects of a Cold Snap

One reason for the difference for the greater number of defects in the winter than in the summer, is the susceptibility of rail to sudden changes in temperature. As noted in section 2.3.4, an overnight drop in temperature, or a "cold snap" can result in several rails breaking almost simultaneously across a line. To simulate the effects of such an occurrence, 5 maintenance events were "scheduled" to take place across the line within one hour of each other. Four runs were performed for each case. One on a typical off-peak traffic day, and one where the cold snap took place on the morning of the day of heaviest traffic. The other two runs looked at the same days, but without the cold snaps. The same segments and repair times were used for all cases, although the 30 mgt line experienced longer slow orders because it would take longer for the necessary number of trains to pass over the freshly worked on track before it returned to "normal". The average delay, and standard deviation of trip time were plotted only for those trains which were effected directly (OP DIR and P DIR), which means they were delayed by a closed track, or by the ensuing slow order (OP DIR&SO, and P DIR&SO). Base information was calculated only for the trains that were affected by the cold snap.

Figures 5.7 and 5.8 show the average delay for the 30 and 80 mgt cases respectively. On the 30 mgt line the average delay for trains that were directly affected was higher than for all trains affected, even with a long slow order. The opposite was true for the 80 mgt case. This is probably due to the fact that the cascading of delays due to track closure is more pronounced on congested lines.

To illustrate the impact of a sudden onslaught of defects, the numbers for the 33 ton axle load cases will be given. For the 30 mgt case, on the off peak and peak days respectively, 8 and 15 trains were impacted both directly (the track closure) and by the ensuing slow order. The average increase in delay, per train, for an off peak day was 136 minutes. A cold snap on a peak traffic increased delay per train affected by 178 minutes. This translates to 18 and 44 additional train-hours of delay for the off peak and peak days respectively. For the 80 mgt case, the total number of trains impacted on the off peak and peak days were 22 and 28 respectively. When multiplied by average increases in delay of 300 and 395 minutes, the total train hours of delay are 110 and 184 on off peak and peak days, respectively. For a light density line it might be acceptable to risk incurring 20 or 40 train-hours of delay. On a high density line, the cost of 200 train-hours, at $180/train-hour, totals $36,000! This does not even include the costs of the maintenance activities themselves, or the hidden cost of unsatisfied customers.

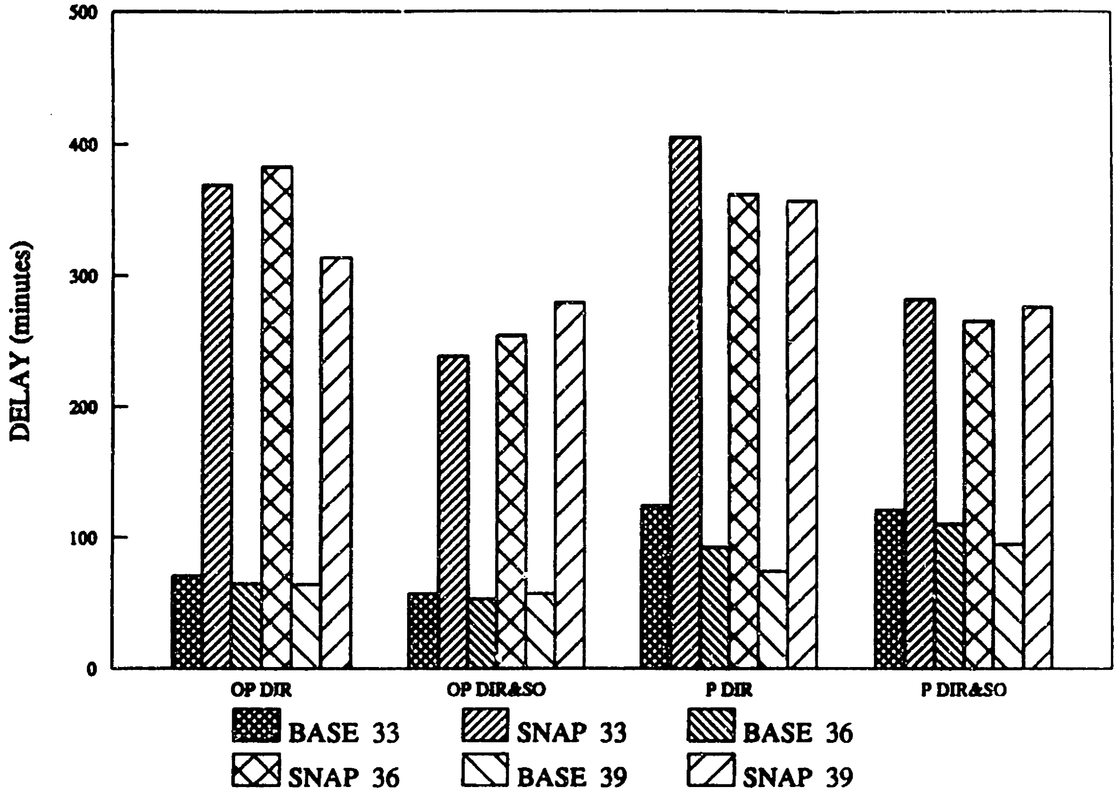


Figure 5.7: Average Delays due to a cold snap: 30 mgt line

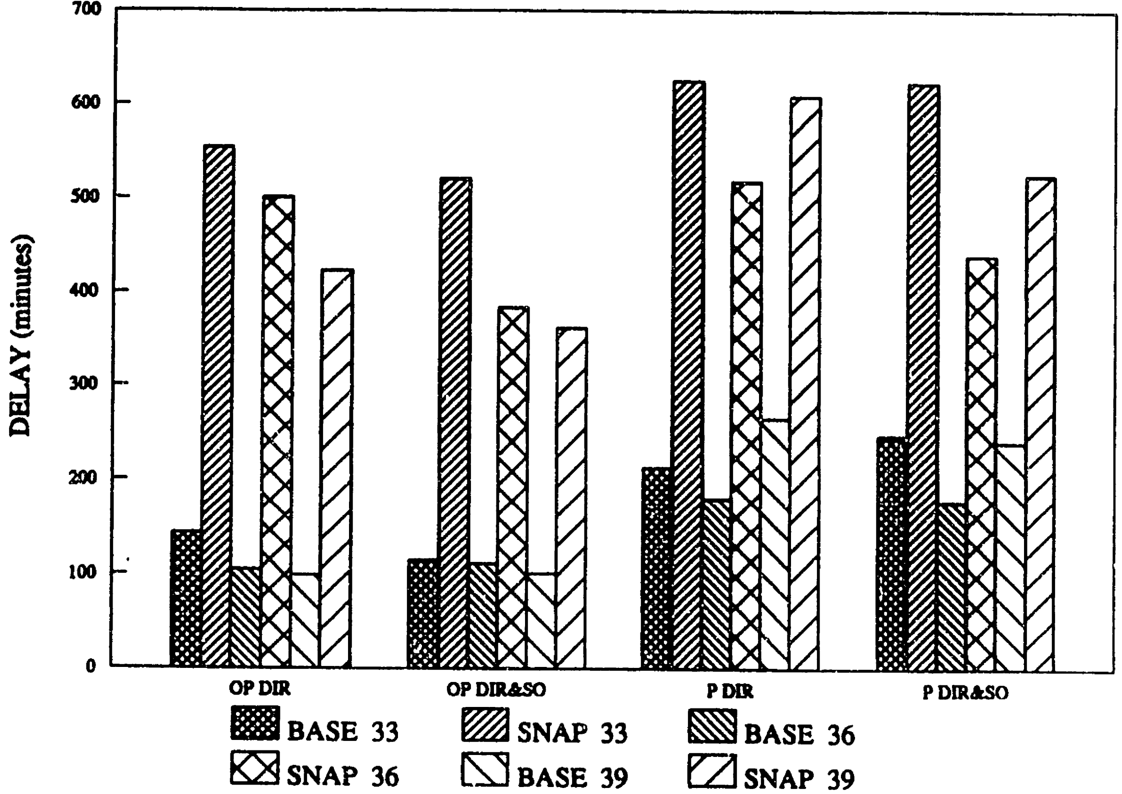


Figure 5.8: Average Delays due to a cold snap: 80 mgt line

5.4.2 Planned Maintenance

In the previous section it was shown that unplanned maintenance events will result in an increase in trip time and trip time standard deviation. This effect was more pronounced on the 80 mgt line than on the 30 mgt line. In this section, the effects of planned maintenance events are discussed. The major difference between these situations is that, when maintenance is planned, it is possible to build maintenance windows into the schedule to reduce the number of trains which are waiting on the line while the maintenance is being performed. The only drawback to maintenance windows is that trains will either have to be scheduled closer together on both sides of the maintenance activity, or a number of trains will have to be canceled. The former approach was taken for this study.

The simulation runs trains across the line for a three week period, so the maintenance activities were scheduled to take place over the five non-peak days during the middle week. The schedules were altered during this week by inserting a gap between departures in the middle of the day. The gap was approximately 6 hours in the 80 mgt case and 9 hours in the 30 mgt case. The planned maintenance was simulated by closing the segment of track of track in the middle of the line, segment 21, for a set length of time each day. An adjacent siding was reserved at the start of the maintenance activity by closing down the segment on the outbound side, segment 20, of the maintenance activity. This segment's free time was increased for one hour after the maintenance was scheduled to end. This ensured that segment 20 could only be occupied by, at most, one train which would wait until the maintenance was complete for the day, on segment 21. A slow order was then placed over a two mile section of segment 21. This slow order was advanced a mile along section 21 every two days to simulate progress by the gang. Runs were performed in which segment 21 was occupied for 0, 4, 6, and 8 hours. The maintenance was scheduled to begin as soon as the last inbound train that departed before the gap in station departures, cleared segment 21. This ensured that the next train along, barring unexpected track or train failures, would be the outbound train which departed either 6 or 9 hours later, depending on the traffic density. Since most project maintenance activities are scheduled for the summer, the summer defect rates (including train failures) were used for these cases.

Statistics were kept over two time periods. The first time period begins with the first train to encounter the maintenance, through the last train on the last day of the scheduled maintenance. The second time period covers the entire week in which the maintenance takes place. That is to say, it begins with the first train to depart a station on the first day of maintenance (the first non-peak day of the second week), and ends with the last train on the second peak day of the week.

The average delay and standard deviation of trip time are depicted in figures 5.9 through 5.12. These figures indicate that for low density lines, maintenance activities of four hours or less can be *scheduled* with no effect on average delay, while longer activities will produce average increases of 30 minutes on the

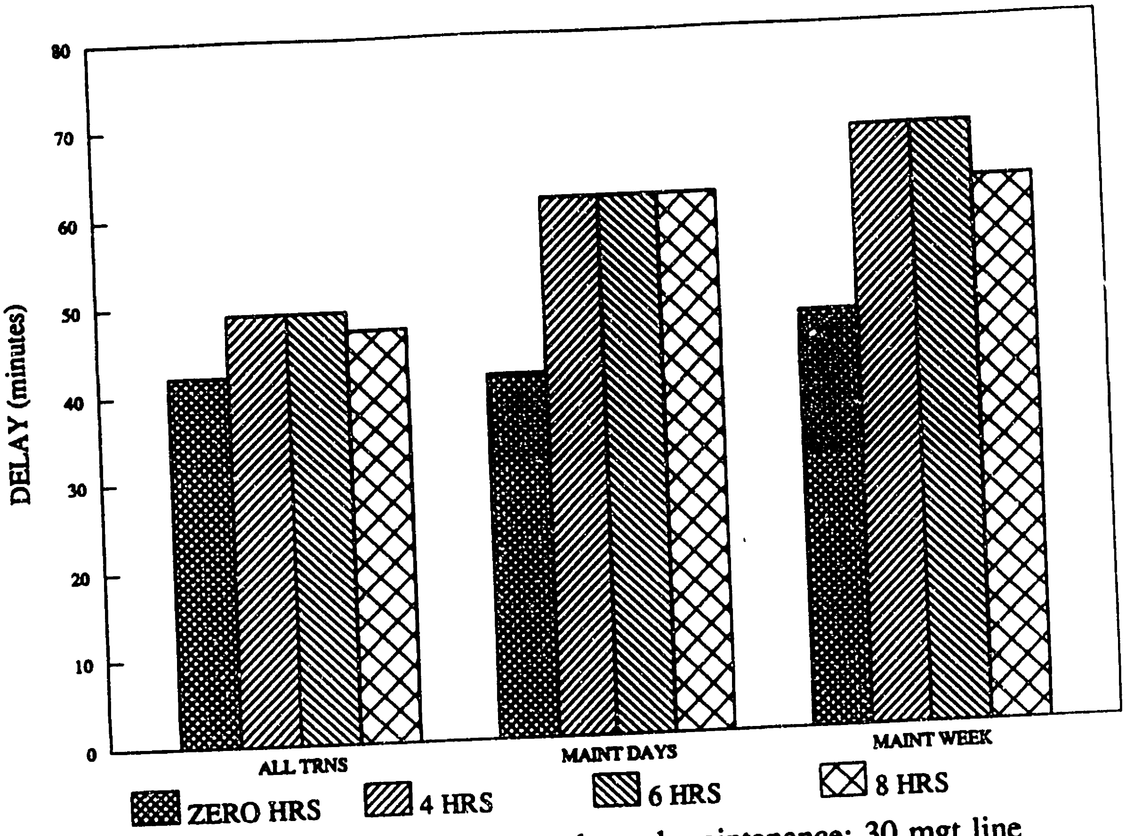


Figure 5.9: Average delay due to planned maintenance: 30 mgt line

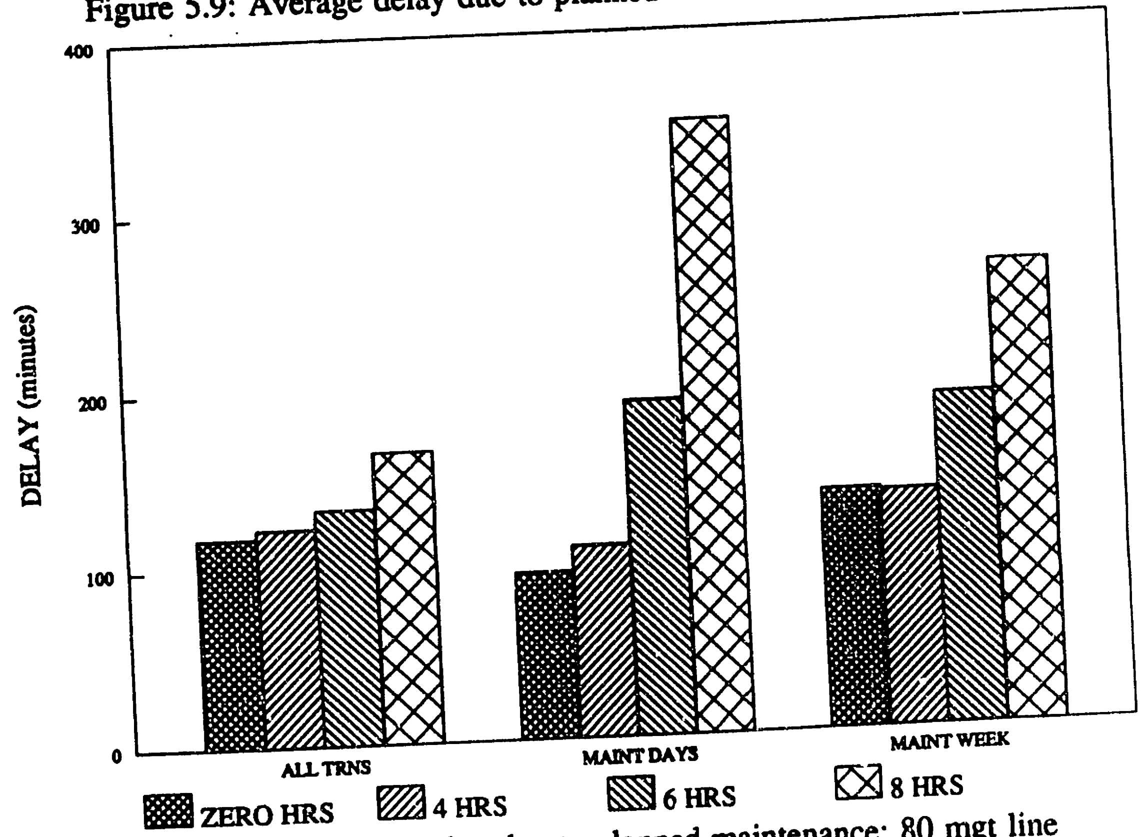


Figure 5.10: Average delay due to planned maintenance: 80 mgt line

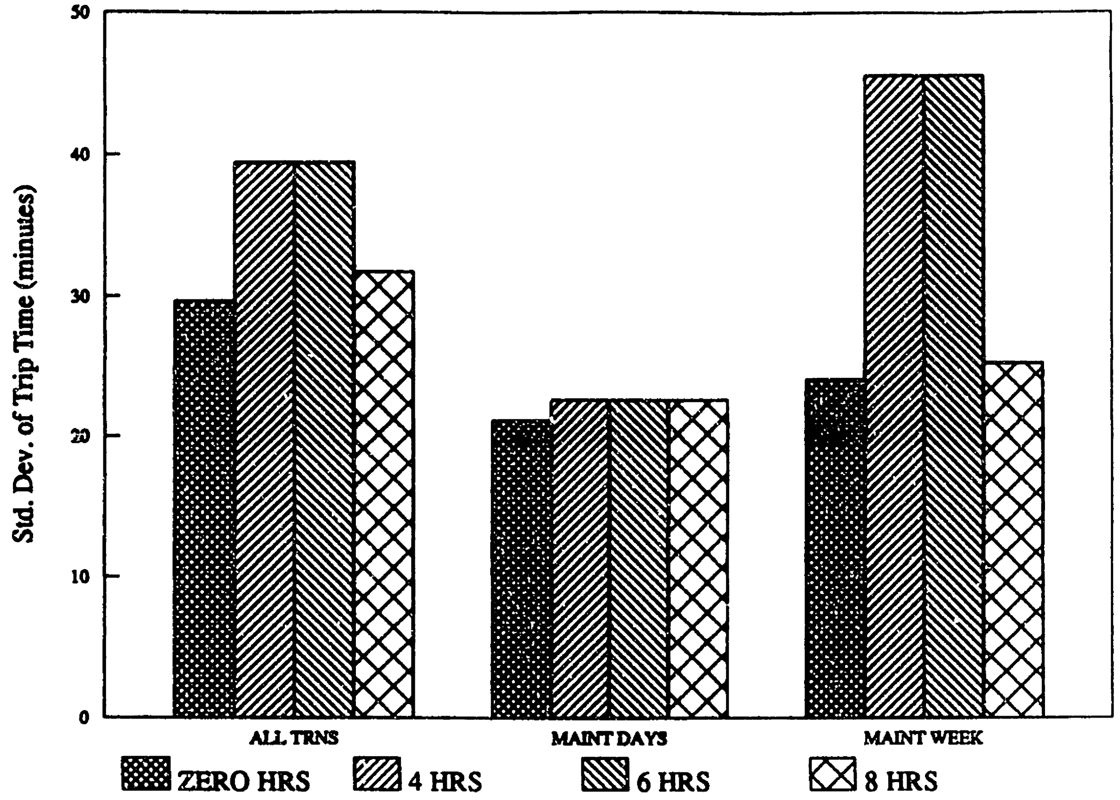


Figure 5.11: Standard Deviation of trip time with planned maintenance: 30 mgt line

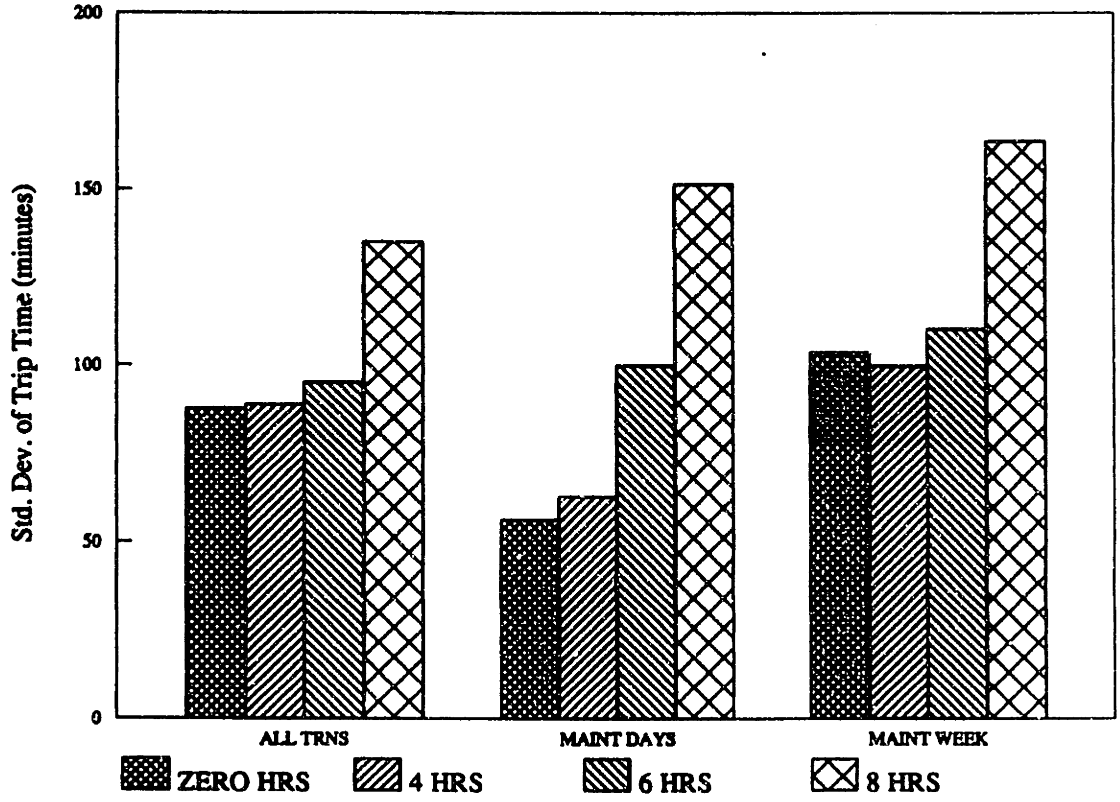


Figure 5.12: Standard Deviation of trip time with planned maintenance: 80 mgt line

maintenance days, and 20 minutes over the entire week. The effect on standard deviation of trip time was only significant when track occupancy time reach 8 hours. The 80 mgt traffic did not fare as well with increases in average delay of maintenance day traffic of 1 1/2 hours and 4 hours for the 6 and 8 hour projects respectively. For all trains during the week, average delays were increased by 40 minutes and 130 minutes for the 6 and 8 hour track occupancy runs. Increases in standard deviation ranged from 30 to one hundred minutes.

Table 5.6 lists the required amounts of maintenance for the 30 and 80 mgt lines with the different prevailing axle loads. On average, approximately 600 more hours are required on the high density line, for project maintenance for rail, ties and ballast. In the 80 mgt, 33 ton axle load case, approximately 1600 hrs are required for project maintenance (including travel on an 10 hour day). This translates to 160 gang-days of 8 hours track occupancy. If all project work is performed during the warmest three or four months of the year, there would be at least one gang on the line every day, and half of these days would see two or more gangs operating full time. Gangs which occupy the track for 6 hours would require 240 days to complete the work while 4-hr gangs would require 320 days. By comparison, the 30 mgt 33 ton case would require about 1100 hours of project maintenance which translates to 110, 165, and 220 days for crews which occupy the line for 8, 6, and 4 hours respectively.

It is clear from the figures that 4 hour maintenance activities are preferable to 6 and 8 hour activities. Unfortunately the amount of maintenance that needs to be performed does not change. In fact, if crews only occupy the track for 4 hours at a time, a greater percentage of the maintenance is spent on activities with fixed times every day, such as travel and set up and break down of equipment. A 4 hour gang will take, twice as many gang-days to complete a job as an 8 hour gang. The extra 2 or so hours for these extra days are paid time for the crew. This could run into serious money. For a 45-man tie crew (Tuzik, 1991), at an average of $12.50/hour/man, a five day job at 10 hours per day (including travel and track access time) will cost $28,125 for labor. A 10 day job at 6 hours per day will cost $33,750 for labor. This does not even include machinery costs, which are usually calculated by the day rather than the hour.

Assuming that the crew has the time to work 10 days, and is not needed on another job for those extra five days, the question that has to be answered is whether or not this extra $5,625 being spent on labor is offset by the savings in reduced train delays. In the 30 mgt case there was no difference in average train delay for the 4 and 8 hour maintenance blocks. In the 80 mgt case, when no maintenance was performed, the average train delay was 94 minutes. On days when maintenance was performed the delay per train was 108 minutes (an increase of 14) when 4 hour work blocks were used, 189 minutes (an increase of 95 minutes) when 6 hour work blocks were used, and 348 minutes (an increase of

254) when 8 hour maintenance blocks were used. The average number of trains per day was 22 for all cases. Over a ten day work period, the 4 hour work blocks would result in an average of 51 train-hours of delay. Over a seven day work period, the 6 hour block would result in an average of 244 train-hours of delay. Over a five day work period, the 8 hour work blocks would result in an average of 465 train-hours of delay. At $180/train-hour, the 4 hour work day would save approximately $34,750 over the 6 hr case, and $53,500 over the 8 hour case. It is important to realize that shorter work blocks will limit the amount of work that a single crew can accomplish during the summer. This may necessitate having more crews on the line simultaneously, which decreases the capacity of the line, thereby increasing train delays. The extent of these delays has not been looked at, and would make a good topic for future study using the simulation model.

It should be noted that even though the trains were scheduled closer together at the beginning and end of the day, thereby increasing the chances that a train scheduled to depart at the end of one day will be delayed beyond the first departure of the following day (thereby resulting the cancellation of the late train by the simulator) not one train was canceled in either the 30 or the 80 mgt case. This bit of information implies that even a high density line can handle large maintenance projects as long as it is willing to absorb the extra delay. One way to handle this is by planning for the trains to arrive late and putting slack into the schedules.

This will preserve the image that the trains are arriving "on time", even though the trip times will necessarily have to be increased.

5.5 Comments

Traffic and route characteristics from a heavy axle load (HAL) study were used as inputs to TRACS. The deterioration models contained in TRACS were used to determine predict average annual maintenance requirements for traffic consists using different axle loads on an 80 mgt and a 30 mgt line. The axle load cases were 33 tons, 36 tons, and 39 tons. The results (depicted in Table 5.5) were used, along with the productivity rates from Appendix A, to determine the average number of hours spent yearly on maintenance of major track components (Table 5.6). For both the 80 and 30 mgt lines, the HAL cases required more yearly gang-hours than the 33 ton cases. On the 80 mgt line, the average hours of maintenance per year increased by approximately 7% and 30% when the axle loads were 36 and 39 tons respectively. On the 30 mgt line, the average hours increased by 10% and 15% for the 36 and 39 ton axle loads, respectively.

The average rates of occurrence of rail defects and spot tie replacements were used to create above average, average, and below average failure rate inputs to the simulation model from chapter 4 (Table 5.7). The HAL traffic information was used to develop train schedules for the different cases, and a generic line was set up with evenly spaced sidings for use in the simulation model. A series of runs, using the various defect rates, were performed for each axle load case at both annual tonnages (80 mgt/year and 30 mgt/year can be interpreted as high density and moderate density lines, respectively). Traffic density played an important role in determining average train delay, even when there were no track failures. Average delays with no track failures were 68 minutes for the high density line and 24 minutes for the moderate density line. The effects continued to be apparent when track failures were added to the runs. For average defect rates, average trip time increased by approximately 70 minutes and 45 minutes for the 80 and 30 mgt lines respectively.

Heavy axle loads did not have a significant effect on average total train delay, even though HAL lines experienced more track failures per year. With heavier axle loads, a railroad car transports the same net tonnage using fewer trains. In this study, it was assumed that train length was held constant, so that the maximum reduction in trains was scheduled. The extra capacity created offsets any additional track outages that arise from the extra failures. For this reason, trains in the HAL runs for both the 80 and 30 mgt lines did not experience any significant increases in average delay or standard deviation of trip time.

Seasonal effects on track failure rate were significant. Above average failure rates resulted in long average train delays, and increases in standard deviation of trip time. A series of runs were also performed which simulated the occurrence of 5 defects simultaneously, the result of a "cold snap", a sudden overnight drop in temperature. Two runs were performed for each axle load case. The first run "scheduled" the snap to occur prior to an off peak traffic day, and the second had the snap occurring on the morning of the peak day of traffic for the week. The average delay was determined for only those trains affected by the snap. On the moderate density (30 mgt) line the snap resulted in 22 train-hours of delay when occurring on an off-peak day, and 28 train-hours of delay when it occurred on a peak traffic day. On the high density line, the off-peak and peak results were 110 and 184 train-hours of delay respectively. The need for supplementary preventive maintenance during the winter months is further emphasized when the train-hours of delay are converted to a tangible cost (@ $180/train-hour). The cost in lost time for the peak day on the high density line was approximately $33,000. This does not include the effects felt at stations further down the line, or the cost of disgruntled customers. The effects of planned track maintenance activities on a line depend on the traffic density and the length of the maintenance activity. On a moderate density line, where long maintenance windows could be easily scheduled, work blocks of 4 and 6 hours had little effect on average delay. Even when 8 hour work blocks were scheduled, the average delay only went up by 30 minutes for trains travelling on the days the maintenance was performed. For the high density line, with its shorter work windows, the average delay went up by 1 1/2 and 4 hours for the 6 and 8 hour projects respectively. Also for the 80 mgt line, increases in standard deviation of trip time ranged from 30 to 100 minutes.

The total cost of train delay on a high density line over a five day period for a gang which occupies the line for 8 hours was compared with the train delay cost over a ten day period of 4 hour work blocks per day. Even though it was for only five days, the 8 hour work blocks resulted in 465 train-hours of delay while the 4 hour work blocks resulted in only 51 train-hours of delay. This explains the attention that the new quick removal equipment gangs, are getting.

Chapter 6 Summary and Future Research

6.1 Summary

6.1.1 Background

The objective of this thesis is to develop a methodology for evaluating the effect of track maintenance on train reliability. The delays experienced by a train on the line can be attributed to one of two sources. Operations delays are the end result of dispatching decisions. They occur when two trains meet on a single track line and one is held on a passing siding while the other continues, or when a one train overtakes another train and the slower train is moved to a siding so it can be passed. Engineering delays result from train or infrastructure failures and maintenance of way. Occupation of a segment by a broken down train or a maintenance crew will prevent other trains from entering that segment. If that segment has no passing siding, trains will be delayed on both sides of the closed segment until repairs are effected or the maintenance is completed. If the traffic density is high enough or the down time is long enough, the train delays will propagate down the line in a cascade effect. The result is an increase in train meets which is essentially an increase in operations delays.

Train performance can be described by the average trip time and the variability of the trip time. Customers are not satisfied with large delivery windows. They want to know that their goods will be delivered when they are expected. While trip time is important, getting it there *on time* is more important than getting it there fast. This is especially true in the rail industry where most goods transported are not perishable.

Track maintenance is performed to keep the track in good condition. Old and poorly maintained track components are more liable to fail, resulting in slower travel speeds, train delays while repairs are effected, or possibly even derailments. The decision as to when a component needs to be replaced is based on a trade off in costs. A track component is replaced either when it becomes more expensive to maintain the component than to replace it, or when leaving it in the track results in excessive train-delay costs compared to the cost of replacement. The scheduling of component replacement activities is an iterative process which begins a year or more before the actual activity takes place. Candidates for replacement are decided based on condition and the needs of particular areas. The possible jobs are prioritized, the costs are predicted, and the next year's activities are decided upon. Decisions are made early to ensure that the necessary material can be purchased and transported to the work site ahead of time. For the list of projects for the year, the crew and machinery assignments are made and project dates are decided. Care must be taken at this stage to avoid conflicts in the schedule. When the time to perform the maintenance arrives, work windows must be found or created in the train schedule so as to minimize the impact of the maintenance on trains and vice versa.

The straight-forward calculation of the costs of maintenance activities does not include the cost of the train delays or the effects of these activities on train reliability. The monetary cost of a maintenance can be calculated by adding the cost of materials and transport of materials to the crew and machinery costs. To achieve the true cost of the project, the expected train delay cost should also be added in. The impact of delays and any variations in trip times on the perception that customers have of the service being provided should also be considered.

Track maintenance planning models are used to forecast track maintenance requirements several years in advance. Deterioration models use current track conditions, maintenance policies and traffic consists to predict when track components such as rail, ties, ballast and turnouts will wear out or require replacement due to failure. Life cycle costing models use the forecasted replacement information to calculate the net present value of performing future maintenance projects. Maintenance scheduling models make use of a set of rules to determine the best time (year) to perform component replacements.

In order to model the effects of planned and unplanned track maintenance activities on line reliability, an existing line simulation model was modified to include these events. The original model is a discrete event simulation model. Inputs the to model are train schedules, line configuration, planned track maintenance activities, and train and track failure patterns including repair times. Events are the departure of a train from a station, departure of a train from a segment, creation of track maintenance, placement of planned maintenance on the line, and placement of random track failures on the line.

6.1.2 Case Study

Using the simulation model and TRACS, the methodology developed in chapter 1 is used to model the effects of track maintenance on train reliability for a generic line running traffic at different axle loads. A 30 mgt and an 80 mgt line are studied. Three cases are looked at for each line. The difference between the cases are the axle loads imparted by the loaded train. These axle loads are 33, 36 and 39 tons. TRACS is used to determine maintenance requirements and average defect rates. A line simulation model is used to study the effect of randomly generated defects on the average trip time and standard deviation of trip time of trains for each case. Another set of simulation runs modelled a simulated "cold snap" with 5 defects occurring almost simultaneously. A third set of runs examined the effect on traffic, of scheduled maintenance activities of length 4, 6 and 8 hours.

6.2 Conclusions

The following conclusions are reached from the study performed in chapter 5.

1. The combined use of a maintenance planning model and a line simulation model makes it possible to estimate the effects of maintenance on the average trip times and reliability of freight trains.

2. The traffic density of a line is a major factor in determining the effects that track failures will have on train operations. The average delay due to unplanned track failures experienced by trains on an 80 mgt line were twice those of trains on an identical line running only 30 mgt of traffic annually.

3. On high density lines, work windows of longer than 4 hours should only be implemented when other alternatives are not available, or if lengthy delays are not a problem. The cost of delays incurred by trains when work blocks of 6 or 8 hours are implemented overshadows the additional cost of sending a crew out more times for shorter work blocks. Quick removal equipment should see more use on high density lines with limited numbers of work window available.

The additional capacity afforded by low density lines plays a role in keeping the standard deviation of trip times from increasing when large blocks of maintenance are placed on the line. With proper scheduling of trains, the effects of project maintenance can be limited to an increase in trip times. The only case scenario which experienced a major increase in standard deviation of trip times was the 80 mgt line running trains with 33 ton axle loads, and this was only when the maintenance crew occupied the line for 8 hour blocks

4. Heavy axle loads do not have a significant effect on average total train delay, even though they lead to an increase in the number of track failures per year. The ability to run fewer trains while transporting the same net tonnage of freight leads to an increase in capacity which offsets any delays due to the increase in yearly spot maintenance activities (and the ensuing track occupancy time by the repair crew).

5. Preventive maintenance is important during the winter months when defect rates are higher. On high density lines, a cold snap which induces several rail defects within a short period of time will result in excessive levels of train delay. Therefore, it is important for railroads which operate in cold climates to make the most of the summer months by maintaining section of track that are most likely to be trouble when winter comes. During the winter, these railroads should schedule more frequent inspections, and contingency plans must be made for incidents such as the cold snap in the case study.

6.3 Suggestions for Future Research

Following are some suggestions for further work that can be done using the simulation model and in the area of track maintenance planning.

1. Perform a HAL study in which the train length and number of trains are kept at pre-HAL levels. This would result in an even greater increase in yearly maintenance requirements, but it would not be offset by an increase the increase in line capacity that results when the number of trains is reduced.
2. Perform a set of runs with more than one project gang on the line at a time. The study of planned maintenance performed in chapter 5 only placed one crew on the line, and the work period was sandwiched between two weeks in which no planned maintenance was performed anywhere on the line. The average yearly number of crew hours required for project maintenance on the 80 mgt line would require that at least two crews be working somewhere on the line for at least half of the summer work period.
3. Perform runs of the model with varying response times of repair crews. Response time could be treated as a function of manpower levels. How many repair crews are necessary to keep a line running smoothly. Rather than simply changing values in the input file, the simulation model could be modified to make the repair times for unplanned maintenance a function of the time of day and location of the failure. This would better reflect travel time from nearest maintenance station.

4. Modify the simulation model so that each segment to has its own defect/failure rate. This would allow further insight into the effects of deferred (by increasing the probability of defect maintenance with age) maintenance on line operations/reliability.
5. Modify the simulation model to allow the user to enter real train schedules. This would make it possible to assign different values of key train characteristics to train travelling in the same direction. Presently trains in the same direction are all assigned the same priority, and normal travelling speed.
6. The productivity rates of maintenance crew could be varied to see what effects this has on line reliability. One example would be a study of rapid on-off equipment. Are the gains which result from lower track occupancy times worth the additional cost of specialized equipment?
7. Validation of the model using empirical data on average train delays at different traffic densities would lend credence to any future studies performed with this model.

References

A.R.E.A., 1990 Manual for Railway Engineering.

Aspebakken, J.I., G.L. Glanen, R.E. Stroot, "Service Maintenance Planning no High Density, Heavy Haul Traffic Routes." Paper presented at the Heavy Haul Workshop, Maintenance of Way and Train Operations, Balancing the Conflicts. Vancouver, BC, Canada, June 1991.

"The big freeze: Getting ready for winter", *Railway Track & Structures*, October 1990.

Burnes, D., "Exploring the world market for m/w", *Railway Track & Structures*, November 1992.

Chrismer, S.M., "Increased ballast maintenance life through stone injection", *Railway Track & Structures*, May 1991.

Conrail visit, Joint Technical Conference: AAR/BRR, Philadelphia, PA, December 1992.

Davis, D.D., P. Chow, "Tie Condition Inspection: A Case Study of Tie Failure Rate, Modes, and Clustering", Association of American Railroads Technical Center, Chicago, IL, Report No. R-714, July, 1989.

Davis, D.D., D.E. Staplin, "Rail Defects and Their Effects on Service Reliability." Paper presented at the Heavy Haul Workshop, Maintenance of Way and Train Operations, Balancing the Conflicts. Vancouver, BC, Canada, June 1991.

Davis, D.D. "Rail Defects and Their Effects on Rail Maintenance Planning", Association of American Railroads Technical Center, Chicago, IL, Report No. WP-154, April 1992.

Dontula, R.V., "A Model For Evaluating Railroad Line-Haul Performance", MST thesis, Department of Civil Engineering, MIT, 1991.

Dunkerley, P.L., "Evaluating Train Service Reliability", BRR/AAR Conference, 1992.

"Examining Maintenance Needs: Current and Future Demands", *Railway Track & Structures*, November 1992.

Federal Railroad Administration (FRA), "Task Force Report: Rail Failure Evaluation," U.S. Department of Transportation, May 1984.

Hargrove, M.B., C.D. Martland, "TRACS: Total Right-of-Way Analysis and Costing Systems", Computer Applications in Railway Planning and Management, Vol. 1, Computational Mechanics Inc., Billerica, Massachusetts, 1990.

Hargrove, M.B., A.R. Auzmendi, "The Effect of Traffic Density on Maintenance of Way Productivity: An Empirical Assessment of North American Experience." Paper presented at the Heavy Haul Workshop, Maintenance of Way and Train Operations, Balancing the Conflicts. Vancouver, BC, Canada, June 1991.

Harker, P.T., D. Jovanovic, "A decision Support System For Train Dispatching: An Optimization-Based Methodology", Transportation Research Forum, Vol. XXXI, Num. 1., 1990.

Harker, P.T., S. Hong, "Two Moments Estimation of the Delay On A Partially Double-Track Rail Line With Scheduled Traffic", Transportation Research Forum, Vol. XXXI, Num. 1., 1990.

Hay, W.W., Railroad Engineering, 2d ed., John Wiley & Sons, 1982.

Hill, Stephen J., Mark C. Walbrun, "Roadmasters and B&B Report", *Railway Track & Structures*, December 1991.

"Looking beneath the surface", *Railway Track & Structures*, March 1990.

Love, J.J., "A Demand-Responsive Approach to Railroad Track Maintenance Management", MSCE thesis, Department of Civil Engineering, MIT, 1981.

"Maintenance planning for freight and transit systems", *Railway Track & Structures*, December 1991.

Martland, C.D., "Incremental Per Diem: Studies in Railroad Operations and Economics Volume 19", prepared for the Task Force on Reliability Studies of the Freight Car Utilization Research/Demonstration Program, MIT Report No. CTS 76-4, July, 1976.

Martland, C.D., A.R. Auzmendi, "A Simplified Model of Rail Wear for Use in Track Maintenance Planning and Costing", Association of American Railroads Research and Test Department, Washington, D.C., Report No. R-742, February, 1990.

Martland, C.D., McNeil, S., Acharya, D., Mishalani, R. and Eshelby, J.E., "Applications of Expert Systems in Railroad Maintenance: Scheduling Rail Relays", Transportation Research Part A, 1990.

Martland, C.D., P. Little, O.K. Kwon, R. Dontula, "Background on Railroad Reliability", Center for Transportation Studies, MIT, 1991.

Martland, C.D., Internal A.A.R. memo concerning HAL Track Analysis, April 5, 1991.

Martland, C.D., R.V. Dontula, "The Effect of Engineering and Mechanical Reliability on the Line Haul Performance of a Single Track Railroad", presented at the Canadian Transportation Research Forum, June 1992.

Matthews, R., "Adequate drainage provides the key to a stable roadbed", *Railway Track & Structures*, March 1990.

Mishalani, R.G., "Planning Rail Replacements,: Satisfying Multiple Criteria Within a Knowledge System Environment", MST thesis, Department of Civil Engineering, MIT, 1989.

"Rail Grinding Accelerates", *Progressive Railroading*, November 1992.

"Review of maintenance-of-way programs an outlook for 1992", *Railway Track & Structures*, January 1992.

Roney, M.D., S.T. Lamson, M.B. Baggott, "Determining the Economical Timing for the Grinding and Renewal of Rail." Paper presented at the Heavy Haul Workshop, 1982.

Roney, M.D., E.R. McIlveen, "Predicting the Future Needs for Programmed Track Maintenance on CP Rail", Paper presented at the Heavy Haul Workshop, Maintenance of Way and Train Operations, Balancing the Conflicts. Vancouver, BC, Canada, June 1991.

Samuels, J.M., "TQM at Conrail", lecture sponsored by MIT International Shipping Club and MIT AAR Affiliated Lab Seminar, March 9, 1991.

SATO, Yashihiko, "Composition of a System for Mechanical Track Maintenance of Railway.", Paper presented at the 6th World Conference on Transport Research, Lyon, France, June 1992.

Shyr, F., "Combining Laboratory and Field Data in Rail Fatigue Analysis", Doctoral Dissertation, Department of Civil Engineering, MIT, 1993.

Smith, E.W., C.D. Martland, unpublished paper reporting the results from a survey 19 North American railroads, concerning maintenance practices in regards to turnouts, 1993.

Staplin, D.E., T.R. Wells, "Branchline Rail Replacement Strategies." Paper presented at the Transportation Research Board Workshop on "Rail Replacement Strategies and Maintenance Management", University of Illinois Champaign-Urbana, June 16, 1987.

Steel, R.K., M.W. Joerms, "A Fatigure Analysis of the Effects of Wheel Load on Rail Life," Association of American Railroads Technical Center, Chicago, IL, Report No. R-689, September, 1988.

"Super Machines vs Quick on-off", *Progressive Railroading*, September 1992.

Thelen, G.A., Y.H. Tse, "An Automatic Speed Enforcement System for Heavy Freight Trains." Paper presented at the 1990 ASME/IEEE Joint Railroad Conference, April 17-19, 1990.

Tuzik, R.E., "Looking at Conrail's track programs with Richard Pyson", *Railway Track & Structures*, December 1991.

Tuzik, R.E., "Fast-on, fast-off machines", *Railway Age*, September 1991.

Webb, H.G. "Track Maintenance Policy and Planning", Transportation Research Record 1029.

Wells, T.T., "Rail Performance Model: Evaluating Rail," American Railway Engineering Association, 1983 Annual Convention, 1983.

"Why deferring tie maintenance can be costly", *Progressive Railroading*, September 1991.

Zarembski, A.M., "Track replacement needs as a function of traffic densities", *Railway Track & Structures*, May 1990.

Appendix A Productivity Rates

The concept of productivity rates was discussed in chapter 2. When the size of job is known, the productivity rates can be used to to calculate how long it will take to complete it. This is useful for scheduling crews and equipment and for determining approximate costs ahead of time. These rates are not intended to represent standard practice across the railroad industry, but they should give an indication of typical average work times. The rates reported include time spent accessing the track, and moving off of the track when the work is completed.

RAIL

Relay gang

TRACS default value is 0.1 miles/hour

A value of 1 mile/day is reported by the A.R.E.A., 1990 Manual for Railway Engineering. For a 10 hour work day, including 8 hours of on track time, the rate is 0.1 miles/hour.

Spot maintenance gang

TRACS default value is 0.5 defects/hour (2 hours / defect)

TIES

Source (Tuzik, 1991)

Two Class I railroads reported tie replacement rates of approximately 200 ties/hour for both small (Fast-On Fast-Off) gangs and large production gangs.

BALLAST

Source (AAR/BRR Joint Conference, 1992)

A Major North American railroad reported that, including delays over a ten hour gang day, surfacing rates average about 0.2 miles/hour.

Renewal rates are taken to be half the surfacing rate; 0.1 miles/hour.

TURNOUTS

Source (unpublished paper, Smith and Martland, 1993)

Replacement of a turnout averages 7 hours. (0.143 turnouts/hour)

Spot maintenance of turnout components is half that rate; 3.5 hours. (0.33 turnouts/hour).

Appendix B

This appendix contains the source code, written in C, for the functions disp_choice_fr_segment and disp_choice_fr_side_seg. The cases referred to in the code are illustrated on pages 187-195.

```
/****************************************************************************/
/***                                                  ***/
/*** Given:  - Train at end of sid/seg                        ***/
/***        - tentative trip time accross segment              ***/
/***                                                  ***/
/*** Find:  - Dispatch decision (hold or move onto segment)        ***/
/***                                                  ***/
/***        - If held: the time of next dispatch attempt           ***/
/***                                                  ***/
/***        - If dispatched: required speed at end of segment       ***/
/***                    (this will depend on where it is        ***/
/***                    predicted to go next, either sid or seg)***/
/***                                                  ***/
/****************************************************************************/


/****************************************************************************/
/***                                                  ***/
/*** Segs are O1, A, B1, C, D1, and E;   Sids are O2, B2, and D2.    ***/
/***                                                  ***/
/*** Train to be dispatched is on either O1 or O2              ***/
/***                                                  ***/
/***    O     A     B     C     D     E       ***/
/***    _____       _____       _____            ***/
/***  /______\__________/______\__________/______\__________      ***/
/***                                                  ***/
/****************************************************************************/

   /******************************************************************/
   /**                                              **/
   /** Information needed to make dispatch decision:             **/
   /**                                              **/
   /**    Direction of trains on A, B1, B2, C, D1. D2 and E     **/
   /**    relative to train on O1/O2.                    **/
   /**                                              **/
   /**    Arrival time (of O1/O2) at end of A.              **/
   /**                                              **/
   /**    If train on O1: IF there is a train on O2, whether     **/
   /**                 or not it was held for priority,  **/
   /**                 & if so, degree of train it was   **/
   /**                 held for.               **/
   /**                                              **/
   /**    If train on O2: IF there is a train on O1, whether     **/
   /**                 or not it was held for priority,  **/
   /**                 & if so, degree of train it was   **/
   /**                 held for.               **/
   /**                                              **/
   /**              IF not a train on O1, whether or not  **/
   /**                 O1 is undergoing maintenance.   **/
   /**                                              **/
   /**    use generic variables for O1/O2.  O and OTHER        **/
   /**                                              **/
   /**    FINAL SPEEDS for:  Halt at end of siding            **/
   /**                 Halt at end of segment        **/
   /**                 Disp onto siding full speed     **/
   /**                 Disp onto segment full speed     **/
```

```
    /**                                         **/
    /**    Free times of A-C                         **/
    /**                                         **/
    /**    Dispatch times of trains on A-E               **/
    /**                                         **/
    /**    Priorities of trains on O-E (IF THEY ARE IN THE      **/
    /**                      OPPOSITE DIRECTION)     **/
    /**                                         **/
    /*****************************************************************/


int disp_choice_fr_sid_seg(TRAIN *t_train, double temp_tt, double fin_vel,
              double *reqd_num, double sid_tt, double sid_fin_vel)
{
    /* if the dispatch function holds the train, reqd_num will be    */
    /* the time at which the next departure attempt will be made     */

    /* if the dispatch function decides to move the train:          */
    /* reqd_num will be the required final velocity determined by    */
    /* the dispatch function for the end of the seg/sid            */

    /* sid_tt and sid_fin_vel are the travel time and final velocity */
    /* the train would have crossing A, if it moves onto the        */
    /* siding (B2) without stopping                           */



 TRAIN *info_train;

 int loc, dir_val, id_num;

 double restr_l, restr_v;

 double Vstp_end_of_sid, Vstp_end_of_seg, Vfull_speed_sid, Vfull_speed_seg;
 double init_vel, Vmax, temp_fin_vel;
 double clr_time = 0.;

 int disp_code;   /* 0 if held       */
                  /* 1 if dispatched */

 int HOLD_TRAIN = 0;
 int DISPATCH_TRAIN = 1;

 int O_priority=0, OTHER_maint=0, OTHER_priority=0;
 double OTHER_maint_end_time = 0.0;
 double O_arr_at_B1, O_arr_at_B2;
 double O_arr_at_CviaB1, O_arr_at_CviaB2;
 double B_disp_time, B_arr_O = INFINITY;
 int B_dir = 0, B_priority = 0;
 double B1_free_time, B2_free_time;
 int B1_dir = 0, B2_dir = 0;
 double C_free_time, C_disp_time, Carr_at_EviaD1=0.0, Carr_at_EviaD2=0.0;
 int C_dir = 0;
 double D1_disp_time=0.0, D1_arr_B = INFINITY, D1_free_time = 0.0;
 int  D1_dir = 0, D1_priority = 0;
 double D2_disp_time=0.0, D2_arr_B = INFINITY, D2_free_time = 0.0;
 int  D2_dir = 0, D2_priority = 0;
 double E_disp_time=0.0, E_arr_at_C, E_free_time = 0.0;
 int  E_dir = 0, E_priority = 0;
 double E_Vmax, E_temp_fin_vel, E_tnext, E_clr_time = 0.;


 loc = t_train->location;
 dir_val = t_train->direction;
 id_num = t_train->id_number;
```

```
  O_priority = t_train->priority;

  O_arr_at_B1  = tnow + temp_tt;
  O_arr_at_B2  = tnow + sid_tt;



/* Determine values of OTHER_priority, OTHER_maint, OTHER_maint_end_time */
/** REMEMBER!!! SIDINGS ARE NOT ASSIGNED TRACK MAINTENANCE **/
/** ====> if train is on the segment, OTHER_maint must be 0 **/


  if( segment_list[loc].train_id1 == t_train->id_number)
   {
    if(segment_list[loc].siding_status == 1)
     {
      info_train = obtain_from_trainlist(segment_list[loc].train_id2);
      if(info_train->priority_del >0)
        priority_check(info_train);
      OTHER_priority = info_train->priority_del;
     }
    OTHER_maint = 0;
   }
  else if( segment_list[loc].train_id2 == t_train->id_number)
   {
    if(segment_list[loc].seg_status == 1)
     {
      info_train = obtain_from_trainlist(segment_list[loc].train_id1);
      if(info_train->priority_del >0)
        priority_check(info_train);
      OTHER_priority = info_train->priority_del;
     }
    OTHER_maint = segment_list[loc].maint;
    OTHER_maint_end_time = segment_list[loc].segment_free_time;
   }
  else
   {
    printf("\n\ntrain is nowhere.   EXITING");
    exit(0);
   }



/** IF one segment stands between the train and the station         **/
/** (i.e. the station is located at B):                **/
/**                                                    **/
/**  if the priority of the next train due to depart the station  **/
/**    is greater than 1 PLUS THE PRIORITY OF TRAIN ON O          **/
/**    AND if the train in the station is due to depart within    **/
/**    1/2 of the trip time from O to station:          **/
/**    HOLD THE TRAIN                                   **/
/**  else                                      **/
/**    dispatch train at max speed                      **/
```

```
/**                                         **/
/**  EITHER WAY                                    **/
/**  RETURN WITHOUT GOING THROUGH THE REST OF THE ALGORITHM        **/

/** The requirements for holding the train are arbitrary, and can   **/
/** be altered as the user sees fit.  The assumption is that the    **/
/** next train from the station would have to be much more important**/
/** than the incoming train in order to justify holding it only one **/
/** segment short of the station.                        **/
  if( (dir_val == OUTBOUND) && (loc == (num_segments-1)) )
   {
   if( (next_ib_train->priority > (O_priority+1) ) &&
      (next_ib_train->dep_time < (tnow + temp_tt/2) ) &&
      (OTHER_maint == 0) &&
      ((OTHER_priority == 0) || (OTHER_priority < next_ib_train->priority)) )
    {
    t_train->priority_del = next_ib_train->priority;
    *reqd_num = next_ib_train->dep_time;
    disp_code = HOLD_TRAIN;
    }
   else
    {
    *reqd_num = minimum(t_train->norm_speed,SIDING_SPEED);
    disp_code = DISPATCH_TRAIN;
    }
   return(disp_code);
   }
  else if( (dir_val == INBOUND) && (loc == 2) )
   {
   if( (next_ob_train->priority > (O_priority+1) ) &&
      (next_ob_train->dep_time < (tnow + temp_tt/2)) &&
      (OTHER_maint == 0) &&
      ((OTHER_priority == 0) || (OTHER_priority < next_ob_train->priority)) )
    {
    t_train->priority_del = next_ob_train->priority;
    *reqd_num = next_ob_train->dep_time;
    disp_code = HOLD_TRAIN;
    }
   else
    {
    *reqd_num = minimum(t_train->norm_speed,SIDING_SPEED);
    disp_code = DISPATCH_TRAIN;
    }
   return(disp_code);
   }


/** If train was not two away from the station, the function now
   goes through the dispatch decision algorithm            **/



/** Calculate possible required speeds at end of segment **/
  Vstp_end_of_seg = Vi_from_s_Vf(segment_list[loc+(2*dir_val)].seg_length,
                      0.0, DECEL);
  Vstp_end_of_sid = minimum(Vstp_end_of_seg, SIDING_SPEED);
  Vmax = minimum(t_train->norm_speed, segment_list[loc+dir_val].max_speed);
  Vfull_speed_seg = minimum(Vmax, segment_list[loc+(2*dir_val)].max_speed);
```

```
  if( segment_list[loc+(2*dir_val)].so_end_time >= O_arr_at_B1 )
   {
   Vstp_end_of_seg = minimum(Vstp_end_of_seg, SO_SPEED);
   Vfull_speed_seg = minimum(Vfull_speed_seg, SO_SPEED);
   }
  Vfull_speed_sid = minimum(Vmax, SIDING_SPEED);
 /** Finished calculating speed options **/

/** Determine free times for B1, B2, and C  **/
  B1_free_time = segment_list[loc+(2*dir_val)].segment_free_time;
  B2_free_time = segment_list[loc+(2*dir_val)].siding_free_time;
  C_free_time = segment_list[loc+(3*dir_val)].segment_free_time;

/***** Note: B CANNOT BE OCCUPIED BY MORE THAN ONE TRAIN   *****/
/*****                                           *****/
/***** FURTHER: If either B1 or B2 is occupied, that train *****/
/*****        MUST be in the OPPOSITE direction of the  *****/
/*****        train on O.                        *****/
/*****                                           *****/
/***** If B contained any trains in the same direction as *****/
/***** the train on O, this function would never have been called */


/** Determine values for B **/
  if(segment_list[loc+(2*dir_val)].seg_status == 1)
   {
   /*train MUST be in opposite direction of train on O*/
   B_disp_time = segment_list[loc+(2*dir_val)].seg_dispatch;
   info_train =
         obtain_from_trainlist(segment_list[loc+(2*dir_val)].train_id1);
   B1_dir = -1;
   B_dir = -1;
   B_priority = info_train->priority;
   /**** Determine B1 arrival time at O if dispatched A.S.A.P. ****/
   init_vel = info_train->speed;
   Vmax = minimum(info_train->norm_speed,
            segment_list[loc+dir_val].max_speed);
   temp_fin_vel = Vmax;
   find_restrictions((loc+(2*dir_val)), 0, 0, info_train->train_length,
                  (-dir_val), &restr_v, &restr_l, B_disp_time,
                  info_train->so_at_last_disp);
   if(init_vel > restr_v)
    init_vel = restr_v;
   if(segment_list[loc+dir_val].so_end_time >= B_disp_time )
     B_arr_O = tt_X_segment_with_SO(init_vel, &temp_fin_vel, info_train,
                      (loc+dir_val), restr_v, restr_l, &clr_time);
   else
     B_arr_O = tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, info_train,
                      (loc+dir_val), restr_v, restr_l, &clr_time);
   B_arr_O += B_disp_time;
```

```
  }
 /*B1 is not occupied by a train*/
  else if(segment_list[loc+(2*dir_val)].siding_status == 1)
   {
   B_disp_time = segment_list[loc+(2*dir_val)].sid_dispatch;

   info_train =
          obtain_from_trainlist(segment_list[loc+(2*dir_val)].train_id2);

   B2_dir = -1;

   B_dir = -1;
   B_priority = info_train->priority;

   /**** Determine B2 arrival time at O if dispatched A.S.A.P. ****/

   init_vel = info_train->speed;
   Vmax = minimum(info_train->norm_speed,
            segment_list[loc+dir_val].max_speed);
   temp_fin_vel = Vmax;

   find_restrictions((loc+(2*dir_val)), 1, 0, info_train->train_length,
               (-dir_val), &restr_v, &restr_l, B_disp_time,0);

   if(init_vel > restr_v)
    init_vel = restr_v;

   if(segment_list[loc+dir_val].so_end_time >= B_disp_time)
    B_arr_O = tt_X_segment_with_SO(init_vel, &temp_fin_vel, info_train,
                      (loc+dir_val), restr_v, restr_l, &clr_time);
   else
    B_arr_O = tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, info_train,
                      (loc+dir_val), restr_v, restr_l, &clr_time);

   B_arr_O += B_disp_time;

   }
  else /*B2 is not occupied by a train*/
   B_dir = 0;



 /****************************************************/
 /****CHECK FOR PRIORITY HOLD DUE TO TRAIN ON B****/
 /****************************************************/
  if( (B_dir == -1)&&(B_priority > O_priority)&&
    ((OTHER_maint==0)||(B_arr_O < OTHER_maint_end_time)) )
   {
   if ((OTHER_priority == 0) || (OTHER_priority < B_priority))
    {
    *reqd_num = B_disp_time;
    t_train->priority_del = B_priority;
    return(HOLD_TRAIN);
    }
   }


 /**************************************/
 /****CHECK FOR MAINTENANCE HOLD  ****/
 /**************************************/
```

```
  if( (B1_dir == -1)&&(B2_free_time >= O_arr_at_B2 ) )
   {
   fprintf(fdiag,"This should not happen #1 @  %lf\n", tnow);
   /*** This print statement will be sent if for some reason
       B2 is not occupied, but it will not be free before
       the train on O reaches it. ***/
   *reqd_num = minimum(B1_free_time, B2_free_time);
   return(HOLD_TRAIN);
   }
  if( (B2_dir == -1)&&(B1_free_time >= O_arr_at_B1 ) )
   {
   *reqd_num = minimum(B1_free_time, B2_free_time);
   return(HOLD_TRAIN);
   }


 /** Determine arrival times at C of train on O by way **/
 /** of seg B1, and sid B2                             **/
 /** if B1 is occupied: will not need O_arr_at_Cvia1        **/
 /** likewise, if B2 is occupied will not need O_arr_at_CviaB2 **/


  if(B1_dir == 0)
   {
    if(segment_list[loc+(2*dir_val)].so_end_time >= O_arr_at_B1)
      Vmax = minimum(t_train->norm_speed, SO_SPEED);
    else
      Vmax = minimum(t_train->norm_speed,
                segment_list[loc+(2*dir_val)].max_speed);
    temp_fin_vel = Vmax;
    find_restrictions((loc+dir_val), 0, 0, t_train->train_length, dir_val,
                 &restr_v, &restr_l, O_arr_at_B1,1);
    if(fin_vel > restr_v)
     fin_vel = restr_v;
    O_arr_at_CviaB1 = tt_X_segment_no_SO(fin_vel, &temp_fin_vel, Vmax,
                 t_train,(loc+(2*dir_val)),restr_v, restr_l,&clr_time);
    O_arr_at_CviaB1 += O_arr_at_B1;
   }

  if(B2_dir == 0)
   {
    Vmax = minimum(t_train->norm_speed, SIDING_SPEED);
    temp_fin_vel = Vmax;
    find_restrictions((loc+dir_val), 0, 1, t_train->train_length, dir_val,
                 &restr_v, &restr_l, O_arr_at_B2,1);
    if(sid_fin_vel > restr_v)
     sid_fin_vel = restr_v;
    O_arr_at_CviaB2 = tt_X_segment_no_SO(sid_fin_vel, &temp_fin_vel, Vmax,
                 t_train,(loc+(2*dir_val)),restr_v,restr_l,&clr_time);
    O_arr_at_CviaB2 += O_arr_at_B2;
   }
```

```
/****************************************************/
/***  COMPARE ARRIVAL AT C WITH C_FREE_TIME    ***/
/***  WHEN B1 or B2 is OCCUPIED            ***/
/****************************************************/
/***If B1 or B2 is occupied, there cannot be a  ***/
/***train on C in the opposite direction of the ***/
/***train on O.  This allows a preliminary     ***/
/***comparison of C's free time with the arrival***/
/***time of the train on O at C to determine    ***/
/***whether or not the train will be halted at  ***/
/***the end of B.         ***********************/
/****************************************************/
  if( (B1_dir == -1)&&(C_free_time >= O_arr_at_CviaB2) )
   {
   *reqd_num = Vstp_end_of_sid;
   return(DISPATCH_TRAIN);
   }
  if( (B2_dir == -1)&&(C_free_time >= O_arr_at_CviaB1) )
   {
   *reqd_num = Vstp_end_of_seg;
   return(DISPATCH_TRAIN);
   }



/** Determine values for C **/


  if(segment_list[loc+(3*dir_val)].seg_status == 1)
   {
   C_disp_time = segment_list[loc+(3*dir_val)].seg_dispatch;
   if( (segment_list[loc+(3*dir_val)].train_id1%2) != (id_num%2) )
    {
    /*trains are in opposite direction*/
    C_dir = -1;
    }
   else  /*trains are in same direction*/
    {
    C_dir = 1;
    }
   }
  else /*C is not occupied by a train*/
   C_dir = 0;

/******************************************************************/
/***                                        ***/
/*** Performing the following check here will allow the use of ***/
/*** one set of checks for all possible remaining combinations ***/
/*** of trains on B-E.                           ***/
/***                                        ***/
/*** If the function makes it past this check, then the train  ***/
/*** is going to be dispatched no matter what, AND it will not ***/
/*** be impeded by either maintenance or trains on C.       ***/
/***                                        ***/
/*** In other words, if the function gets past this check, the ***/
/*** only possible reasons for stopping the train at the end of***/
/*** B will be for trains on D and E which will either reach   ***/
```

```
/*** C before O does, or which have higher priority than O.   ***/
/***                                                          ***/
/*****************************************************************/

 if(B_dir == 0)
  {
  if(C_dir != -1)
   {
   if( (B1_free_time >= O_arr_at_B1) &&
      (B2_free_time >= O_arr_at_B2) )
    {
    *reqd_num = 0.0;
    return(DISPATCH_TRAIN);
    }
   else if( (B1_free_time >= O_arr_at_B1 ) &&
         (C_free_time >= O_arr_at_CviaB2) )
    {
    *reqd_num = Vstp_end_of_sid;
    return(DISPATCH_TRAIN);
    }
   else if( (B2_free_time >= O_arr_at_B2 ) &&
         (C_free_time >= O_arr_at_CviaB1) )
    {
    *reqd_num = Vstp_end_of_seg;
    return(DISPATCH_TRAIN);
    }
   else if(C_free_time >= O_arr_at_CviaB1)
    {
    *reqd_num = Vstp_end_of_sid;
    return(DISPATCH_TRAIN);
    }
   }
  else /* C_dir == -1 */
   {
   /** B1 and B2 are empty of trains **/
   /** B1 might have maintenance going on **/
   /** B2 will not have maintenance without a train on it **/
   if(B1_free_time >= O_arr_at_B1 )  /* O will arrive at B before */
                                     /* the maintenance ends      */
    {
    if(B1_free_time >= C_disp_time) /* C will arrive at B before */
     {                              /* the maintenance ends      */
     *reqd_num = C_disp_time;
     return(HOLD_TRAIN);
     }

   /* C will arrive at B after the maintenance ends */
   /* O will arrive at B before the maintenance ends */
    else
     {
     *reqd_num = Vstp_end_of_sid;
     return(DISPATCH_TRAIN);
     }
    }
 /*maintenance will not affect O */
   else if(C_disp_time >= O_arr_at_B1) /* C will arrive at B after O */
    {
    *reqd_num = Vstp_end_of_sid;
```

```
    }
   else if((C_disp_time < O_arr_at_B1 )&&   /* C will arrive at B before O */
         (C_free_time >= O_arr_at_CviaB1) ) /* but will not clear C in   */
    {                             /* time to allow O to travel */
                                  /* through without stopping  */
    *reqd_num = Vstp_end_of_seg;
    }
   }
  }


/*** It has already been established that both B1 and B2 cannot be ***/
/*** occupied by trains at the same time.                ***/
/***                                          ***/
/*** A check has already been performed to see if there is maint   ***/
/*** on B1/B2 and a train on B2/B1. This would have resulted in    ***/
/*** HOLDING the train on O.                       ***/
/***                                          ***/
/*** IF THE FUNCTION HAS REACHED THIS POINT, THE TRAIN ON O IS      ***/
/*** GOING TO BE DISPATCHED ONTO A.  The only question that remains***/
/*** is the velocity it will have at the end of A.  This will be   ***/
/*** determined by whether it moves onto B1 or B2.           ***/
/***                                          ***/
/*** It is still possible for the train on O to reach the end of A ***/
/*** and for either B1 or B2 to not be free (either due to maint   ***/
/*** or because it is occupied).  If this is the case, then O will ***/
/*** (obviously) move onto the other segment which is free.       ***/
/***                                          ***/
/*** Once it has been established that the train is going to be    ***/
/*** dispatched, the only values that reqd_num will be assigned     ***/
/*** are final velocities (at the end of A).              ***/
/***                                          ***/
/*** The following if statements will make sure that if O reaches  ***/
/*** the end of A and cannot move onto B1 or B2, the variables      ***/
/*** that deal with speeds and travel times across that seg/sid    ***/
/*** are assigned the values for speed and travel across the other ***/
/*** (sid/seg).                            ***/
/***                                          ***/
/*** After these if statements, we will not have to be concerned   ***/
/*** with whether or not B1, B2 or both are free.            ***/
/*** the correct assignments will be made to reqd_num by default   ***/


  if( (B1_free_time >= O_arr_at_B1) || (B1_dir == -1) )
   {
   Vstp_end_of_seg = Vstp_end_of_sid;
   Vfull_speed_seg = Vfull_speed_sid;
   O_arr_at_CviaB1 = O_arr_at_CviaB2;
   }
/*else  do not change V...seg */


  if( (B2_free_time >= O_arr_at_B2) || (B2_dir == -1) )
   {
   Vstp_end_of_sid = Vstp_end_of_seg;
   Vfull_speed_sid = Vfull_speed_seg;
   O_arr_at_CviaB2 = O_arr_at_CviaB1;
   }
/*else  do not change V...sid */
```

```
/*********************************************************************/
/** IF there are three segments between the train and the station**/
/** (i.e. the station is located at D):                **/
/**                                           **/
/** D1_dir = -1;                                  **/
/** D1_priority = next_ib_train->priority or                 **/
/**          next_ob_train->priority                **/
/**                                           **/
/** D2's variables will all be zeroes                    **/
/**                                           **/
/** E's variables will all be zeroes                     **/
/**                                           **/
/**                                           **/
/**                                           **/


 if( (dir_val == OUTBOUND) && (loc == (num_segments-3)) )
  { /*there are 3 segments between the train and the IB station */

  D1_dir = -1;
  D1_priority = next_ib_train->priority;
  D1_disp_time = maximum(next_ib_train->dep_time, (C_free_time+DEP_BUFFER));
  D2_dir = 0;
  E_dir = 0;

  /**** Determine D1 arrival time at B if dispatched A.S.A.P. ****/

  init_vel = next_ib_train->speed;
  Vmax = minimum(next_ib_train->norm_speed,
           segment_list[num_segments].max_speed);
  temp_fin_vel = Vmax;

  restr_v = next_ib_train->norm_speed;
  restr_l = next_ib_train->train_length;

  if(segment_list[num_segments].so_end_time >= D1_disp_time )
    D1_arr_B = tt_X_segment_with_SO(init_vel, &temp_fin_vel, next_ib_train,
                  (num_segments), restr_v, restr_l, &clr_time);
  else
    D1_arr_B = tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, next_ib_train,
                  (num_segments), restr_v, restr_l, &clr_time);

  D1_arr_B += D1_disp_time;

  D1_free_time = INFINITY;
  }
 else if( (dir_val == INBOUND) && (loc == 4) )
  { /*there are 3 segments between train and the OB station */
  D1_dir = -1;
  D1_priority = next_ob_train->priority;
  D1_disp_time = maximum(next_ob_train->dep_time, (C_free_time+DEP_BUFFER));
  D2_dir = 0;
  E_dir = 0;

  /**** Determine D1 arrival time at B if dispatched A.S.A.P. ****/

  init_vel = next_ob_train->speed;
  Vmax = minimum(next_ob_train->norm_speed, segment_list[0].max_speed);
  temp_fin_vel = Vmax;

  restr_v = next_ib_train->norm_speed;
  restr_l = next_ob_train->train_length;
```

```
  if(segment_list[0].so_end_time >= D1_disp_time )
   D1_arr_B = tt_X_segment_with_SO(init_vel, &temp_fin_vel, next_ob_train,
                        0, restr_v, restr_l, &clr_time);
  else
   D1_arr_B =  tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, next_ob_train,
                        0, restr_v, restr_l, &clr_time);
  D1_arr_B += D1_disp_time;
  D1_free_time= INFINITY;
 }
else
 {
  /* train is in middle of line somewhere */


  /* E_free_time is needed if D1_dir or D2_dir == 1 */
  /* so it is determined before finding values for  */
  /* D1 and D2                                       */
  E_free_time = segment_list[loc+(5*dir_val)].segment_free_time;



/** Determine values for D1 **/
  D1_free_time = segment_list[loc+(4*dir_val)].segment_free_time;
  if(segment_list[loc+(4*dir_val)].seg_status == 1)
   {
   if( (segment_list[loc+(4*dir_val)].train_id1 % 2) != (id_num%2) )
    {
    /*trains are in opposite direction*/
    D1_disp_time = maximum(segment_list[loc+(4*dir_val)].seg_dispatch,
                   (C_free_time+DEP_BUFFER));
    info_train =
        obtain_from_trainlist(segment_list[loc+(4*dir_val)].train_id1);
    D1_dir = -1;
    D1_priority = info_train->priority;
    /**** Determine D1 arrival time at B if dispatched A.S.A.P. ****/
    init_vel = info_train->speed;
    Vmax = minimum(info_train->norm_speed,
             segment_list[loc+(3*dir_val)].max_speed);
    temp_fin_vel = Vmax;
    find_restrictions((loc+(4*dir_val)), 0, 0, info_train->train_length,
              (-dir_val), &restr_v, &restr_l, D1_disp_time,
               info_train->so_at_last_disp);
    if(init_vel > restr_v)
     init_vel = restr_v;
    if(segment_list[loc+(3*dir_val)].so_end_time >= D1_disp_time )
     D1_arr_B = tt_X_segment_with_SO(init_vel, &temp_fin_vel, info_train,
                 (loc+(3*dir_val)), restr_v, restr_l, &clr_time);
    else
     D1_arr_B = tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, info_train,
                 (loc+(3*dir_val)), restr_v, restr_l, &clr_time);
```

```
    D1_arr_B += D1_disp_time;
   }
  else  /*trains are in same direction*/
   {
    D1_dir = 1;
    D1_disp_time = maximum(segment_list[loc+(4*dir_val)].seg_dispatch,
                    (E_free_time+DEP_BUFFER));
   }
  }
 else /*D1 is not occupied by a train*/
  D1_dir = 0;

/** Determine values for D2 **/
 D2_free_time = segment_list[loc+(4*dir_val)].siding_free_time;
 if(segment_list[loc+(4*dir_val)].siding_status == 1)
  {
   if( (segment_list[loc+(4*dir_val)].train_id2 % 2) != (id_num%2) )
    {
     /*trains are in opposite direction*/
     D2_disp_time = maximum(segment_list[loc+(4*dir_val)].sid_dispatch,
                    (C_free_time+DEP_BUFFER));
     info_train =
        obtain_from_trainlist(segment_list[loc+(4*dir_val)].train_id2);
     D2_dir = -1;
     D2_priority = info_train->priority;
   /**** Determine D2 arrival time at B if dispatched A.S.A.P. ****/
     init_vel = info_train->speed;
     Vmax = minimum(info_train->norm_speed,
             segment_list[loc+(3*dir_val)].max_speed);
     temp_fin_vel = Vmax;
     find_restrictions((loc+(4*dir_val)), 1, 0, info_train->train_length,
              (-dir_val), &restr_v, &restr_l, D2_disp_time,0);
     if(init_vel > restr_v)
      init_vel = restr_v;
     if(segment_list[loc+(3*dir_val)].so_end_time >= D2_disp_time)
       D2_arr_B = tt_X_segment_with_SO(init_vel, &temp_fin_vel, info_train,
                (loc+(3*dir_val)), restr_v, restr_l, &clr_time);
     else
       D2_arr_B = tt_X_segment_no_SO(init_vel, &temp_fin_vel,  Vmax, info_train,
                (loc+(3*dir_val)), restr_v, restr_l, &clr_time);
     D2_arr_B += D2_disp_time;
    }
  else  /*trains are in same direction*/
   {
    D2_dir = 1;
    D2_disp_time = maximum(segment_list[loc+(4*dir_val)].sid_dispatch,
                    (E_free_time+DEP_BUFFER));
   }
  }
 else /*D2 is not occupied by a train*/
  D2_dir = 0;

/** Determine values for E **/
```

```
if(segment_list[loc+(5*dir_val)].seg_status == 1)
{
 if( (segment_list[loc+(5*dir_val)].train_id1 % 2) != (id_num%2) )
  {
  /*trains are in opposite direction*/
  info_train =
      obtain_from_trainlist(segment_list[loc+(5*dir_val)].train_id1);
  E_dir = -1;
  E_priority = info_train->priority;
  E_disp_time = segment_list[loc+(5*dir_val)].seg_dispatch;


 /*** The variable E_arr_at_C is never used in cases where
    C is occupied by a train in the opposite direction of
    the train on A.  It is also never used when
    D1 or D2 are occupied by trains in the opposite direction
    of the train on A.  This information can be used to cut
    down on the # of times the variable is calculated      ***/
 /*** Actually, if D1_dir or D2_dir == -1, there could not be
    a train of E such that E_dir == -1.
    ==> DO NOT NEED TO CHECK IF D1 == -1 or D2 == -1     ***/


   if(C_dir != -1)
    {
    if( (D1_dir == 1) || ((D2_dir==0)&&(D1_free_time>E_disp_time)) ||
       ((C_dir == 1)&&(C_disp_time < E_disp_time)) )
     {
     E_Vmax = minimum(info_train->norm_speed, SIDING_SPEED);
     E_temp_fin_vel = E_Vmax;
     E_tnext = maximum(E_disp_time, (D2_free_time+DEP_BUFFER));
     find_restrictions((loc+(5*dir_val)), 0, 1,info_train->train_length,
                 (-dir_val), &restr_v, &restr_l, E_tnext,
                 info_train->so_at_last_disp);
     }
    else
     {
     E_tnext = maximum(E_disp_time, (D1_free_time+DEP_BUFFER));
     if(segment_list[loc+(4*dir_val)].so_end_time >= E_tnext)
      E_Vmax = minimum(info_train->norm_speed, SO_SPEED);
     else
      E_Vmax = minimum(info_train->norm_speed,
                 segment_list[loc+(4*dir_val)].max_speed);
     E_temp_fin_vel = E_Vmax;
     find_restrictions((loc+(5*dir_val)), 0, 0,info_train->train_length,
                 (-dir_val), &restr_v, &restr_l, E_tnext,
                 info_train->so_at_last_disp);
     }
    init_vel = info_train->speed;
    if(init_vel > E_Vmax)
     init_vel = E_Vmax;
    E_arr_at_C = tt_X_segment_no_SO(init_vel, &E_temp_fin_vel, E_Vmax,
                       info_train, (loc+(4*dir_val)),
                       restr_v, restr_l, &E_clr_time);
```

```
      E_arr_at_C += E_tnext;
      }
     }
    else   /*trains are in same direction*/
     {
     E_dir = 1;
     }
   }
  else /*E is not occupied by a train*/
   E_dir = 0;
  }
/** IF C_dir == 1   **/
/** Determine arrival times at E of train on C by way **/
/** of seg D1, and sid D2                           **/
/** if D1 is occupied: will not need Carr_at_EviaD1        **/
/** likewise, if D2 is occupied will not need Carr_at_EviaD2 **/
/** E could be undergoing maintenance or occupied ==>       **/
/** trip time (ie dispatch time) is set equal to max(tt,E_free_time) **/
/** if E is occupied: train will stop at end of D    **/
 if(C_dir == 1)
  {
  info_train =
       obtain_from_trainlist(segment_list[loc+(3*dir_val)].train_id1);
  if(D1_dir == 0)
   {
   if(segment_list[loc+(4*dir_val)].so_end_time >= C_disp_time)
    Vmax = minimum(info_train->norm_speed, SO_SPEED);
   else
    Vmax = minimum(info_train->norm_speed,
              segment_list[loc+(4*dir_val)].max_speed);
   temp_fin_vel = Vmax;
   find_restrictions((loc+(3*dir_val)), 0, 0, info_train->train_length,
                dir_val, &restr_v, &restr_l, C_disp_time,
                info_train->so_at_last_disp);
   init_vel = info_train->speed;
   if(init_vel > restr_v)
    init_vel = restr_v;
   else if(init_vel >Vmax)
    init_vel = Vmax;
   Carr_at_EviaD1 = tt_X_segment_no_SO(init_vel, &temp_fin_vel, Vmax,
                           info_train, (loc+(4*dir_val)),
                           restr_v, restr_l, &clr_time);
   Carr_at_EviaD1 += C_disp_time;
   if( (E_free_time > Carr_at_EviaD1)&&(temp_fin_vel > 0.0) )
    {
    temp_fin_vel = 0.0;

    Carr_at_EviaD1 = tt_X_segment_no_SO(init_vel, &temp_fin_vel, Vmax,
                            info_train, (loc+(4*dir_val)),
                            restr_v, restr_l, &clr_time);
    Carr_at_EviaD1 += C_disp_time;
    }
```

```
    Carr_at_EviaD1 = maximum(Carr_at_EviaD1, (E_free_time+DEP_BUFFER));
   }

  if(D2_dir == 0)
   {
    Vmax = minimum(t_train->norm_speed, SIDING_SPEED);

    temp_fin_vel = Vmax;

    if( (E_dir != 0)&&(temp_fin_vel > 0.0) )
     temp_fin_vel = 0.0;

    find_restrictions((loc+(3*dir_val)), 0, 1, info_train->train_length,
                 dir_val, &restr_v, &restr_l, C_disp_time,
                 info_train->so_at_last_disp);

    init_vel = info_train->speed;

    if(init_vel > restr_v)
     init_vel = restr_v;
    else if(init_vel >Vmax)
     init_vel = Vmax;

    Carr_at_EviaD2 =  tt_X_segment_no_SO(init_vel, &temp_fin_vel, Vmax,
                          info_train, (loc+(4*dir_val)),
                          restr_v, restr_l, &clr_time);

    Carr_at_EviaD2 += C_disp_time;

    if( (E_free_time > Carr_at_EviaD2)&&(temp_fin_vel > 0.0) )
     {
      temp_fin_vel = 0.0;

      Carr_at_EviaD2 =  tt_X_segment_no_SO(init_vel, &temp_fin_vel, Vmax,
                            info_train, (loc+(4*dir_val)),
                            restr_v, restr_l, &clr_time);

      Carr_at_EviaD2 += C_disp_time;
     }
/** if the train is 4 segments out from its destination station, segment
   E does not exist, and there are essentially many D2's.

   For this reason, the train on C will only affect the train on O
   while it is occupying segment C.  Once it leaves C, it leaves the
   line.
***/
    if( ((dir_val == OUTBOUND) && (loc == (num_segments-4))) ||
       ((dir_val == INBOUND) && (loc == 4)) )
     Carr_at_EviaD2 = clr_time;

    Carr_at_EviaD2 = maximum(Carr_at_EviaD2, (E_free_time+DEP_BUFFER));
   }

  /**D1 could be under maintenance **/

  if( (D1_free_time >= C_disp_time)&&(D1_dir == 0) )
   {
    if(D2_dir == 0)
     {
      if( (E_dir == -1)&&(E_disp_time < C_disp_time) )
       Carr_at_EviaD1 += D1_free_time - C_disp_time;
      else
       Carr_at_EviaD1 = Carr_at_EviaD2;
     }
```

```
     else /*D2_dir == -1 */
      Carr_at_EviaD1 += D1_free_time - C_disp_time;
     }
  /** If E_dir == -1, whether or not C moves onto D1 or D2 depends
      on whether or not E will reach D first.  Whichever gets there
      first will move onto the siding.
  **/
   if(E_dir == -1)
    {
     if(E_disp_time < C_disp_time)   /*E will move onto the siding */
      Carr_at_EviaD2 = Carr_at_EviaD1;
     else if(E_disp_time > C_disp_time) /*C will move onto the siding */
      Carr_at_EviaD1 = Carr_at_EviaD2;
    }
   }




/*******************************************************/
/**                                       **/
/**  Use segment information to determine dispatch     **/
/**  action (hold or dispatch), and final speed.      **/
/**                                       **/
/*******************************************************/


/*** Once the function has reached this point, the train is  ***/
/*** guaranteed of being dispatched.                 ***/
/*** This allows us to make the following assignment which   ***/
/*** will be in effect for the rest of the function.        ***/


  disp_code = DISPATCH_TRAIN;


  if(C_dir == 1)
   {
  /*D1_dir cannot be 1; D2_dir cannot be 1*/
    if(D1_dir == -1)
     {               /* 4-11 */  /* 3C-3 */
                     /* 3BC-19 */ /* 2C-1 */
      if( (Carr_at_EviaD2 >= O_arr_at_CviaB1)||
         (D1_priority >= O_priority) )
       *reqd_num = Vstp_end_of_sid;
      else
       *reqd_num = Vfull_speed_seg;
     }
    else if(D2_dir == -1)
     {               /* 4-11 */  /* 3C-4 */
                     /* 3BC-20 */ /* 2C-2 */
      if( (Carr_at_EviaD1 >= O_arr_at_CviaB1)||
         (D2_priority >= O_priority) )
       *reqd_num = Vstp_end_of_sid;
      else
       *reqd_num = Vfull_speed_seg;
     }
    else if(E_dir == 1)  /* 3BC-24 */ /* 2C-6 */
```

```
    {
    if(Carr_at_EviaD1 >= O_arr_at_CviaB1)
     *reqd_num = Vstp_end_of_seg;
    else
     *reqd_num = Vfull_speed_seg;
    }
   else if(E_dir == -1)  /* 3BC-23 */  /* 2C-5 */
    {
    if( (Carr_at_EviaD1 >= O_arr_at_CviaB1)||
        ((E_priority > O_priority)&&(E_disp_time < O_arr_at_CviaB1)) ||
        ((E_priority == O_priority)&&(E_arr_at_C< O_arr_at_CviaB1)) )
     *reqd_num = Vstp_end_of_sid;
    else
     *reqd_num = Vfull_speed_seg;
    }
   else /* D and E are empty*/  /* 2B-10 */  /* 1-6 */
    {
    if(Carr_at_EviaD1 >= O_arr_at_CviaB1)
     *reqd_num = Vstp_end_of_sid;
    else
     *reqd_num = Vfull_speed_seg;
    }
  }
 else
  {
  /** The final velocity, at the end of A, will depend on whichever
     (B1, B2, or Both) is free, and whether or not the train is to be
     halted at the end of B or run all of the way through.     ***/

 /**** If program reaches this point, then either B1, B2 or  ****/
 /**** both will be free when the train on O reaches B.      ****/
 /**** AND                                                   ****/
 /**** Train on C will clear C before train on O reaches C   ****/
 /****                                                       ****/

  if(D1_dir == 1)
   {
   if(D2_dir == -1)
    {                /* 4BDE-4 */  /* 3CD-2 */ /* 3D-2 */
                     /* 3BD0-4 */          /* 2D0-2 */
    if( (D1_disp_time >= O_arr_at_CviaB1)||
        (D2_priority >= O_priority) )
     *reqd_num = Vstp_end_of_sid;
    else
     *reqd_num = Vfull_speed_seg;
    }
   else if(E_dir == -1)  /* 3BD-13 */ /* 3C-13 */ /* 2D-1 */
    {
    if( (D1_disp_time >= O_arr_at_CviaB1) ||
        ((E_priority > O_priority)&&(E_disp_time < O_arr_at_CviaB1)) ||
        ((E_priority == O_priority)&&(E_arr_at_C< O_arr_at_CviaB1)) )
     *reqd_num = Vstp_end_of_sid;
    else
     *reqd_num = Vfull_speed_seg;
    }
   else /*E_dir==1 OR E is empty */  /* 3BD-15 */ /* 3C-15 */ /* 2D-3 */
    {                        /* 2B-13 */  /* 2C-9 */  /* 1-9 */
    if(D1_disp_time >= O_arr_at_CviaB1)
     *reqd_num = Vstp_end_of_sid;
    else
     *reqd_num = Vfull_speed_seg;
```

```
   }
  }
 else if(D1_dir == -1)
  {
  if(D2_dir == 1)
   {              /* 4BDE-2 */ /* 3CD-1 */ /* 3D-4 */
                  /* 3BD0-3 */          /* 2D0-1 */
   if( (D2_disp_time >= O_arr_at_CviaB1)||
      (D1_priority >= O_priority) )
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
   }
  else /*E_dir == 1 OR E is empty */ /* 3BD-11 *//* 3C-11 *//* 2D-7 */
   {                          /* 2B-11  *//* 2C-7 */ /* 1-7 */
   if(D1_priority >= O_priority)
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
   }
  }
 else if(D2_dir == 1)
  {
  if(E_dir == 1)       /* 3BD-16 */ /* 3C-16 */ /* 2D-4 */
   {
   if(D2_disp_time >= O_arr_at_CviaB1)
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
   }
  else if(E_dir == -1)  /* 3BD-14 */ /* 3C-14 */ /* 2D-2 */
   {
   if( (D2_disp_time >= O_arr_at_CviaB1) ||
     ((E_priority > O_priority)&&(E_disp_time < O_arr_at_CviaB1)) ||
     ((E_priority == O_priority)&&(E_arr_at_C< O_arr_at_CviaB1)) )
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
   }
  else /*E is empty*/   /* 2B-14 */ /* 2C-10 */ /* 1-10 */
   {
   if(D2_disp_time >= O_arr_at_CviaB1)
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
   }
  }
 else if(D2_dir == -1)
  {
   /* E_dir == 1 OR E is empty */    /* 3BD-12 *//* 3C-12 *//* 2D-8 */
                             /* 2B-12  *//* 2C-8 */ /* 1-8 */
   if(D2_priority >= O_priority)
    *reqd_num = Vstp_end_of_sid;
   else
    *reqd_num = Vfull_speed_seg;
  }
 else if(E_dir == 1)    /* 2B-16 */ /* 2C-12 */ /* 1-12 */
  {
   *reqd_num = Vfull_speed_sid;
  }
```

```
  else if(E_dir == -1)    /* 2B-15 */ /* 2C-11 */ /* 1-11 */
   {
    if(((E_priority > O_priority)&&(E_disp_time < O_arr_at_CviaB1)) ||
      ((E_priority == O_priority)&&(E_arr_at_C< O_arr_at_CviaB1)) )
      *reqd_num = Vstp_end_of_seg;
    else
      *reqd_num = Vfull_speed_seg;
    }
  else /*C, D, and E are empty*/  /* 1-1 */ /* 1-5 */
   {                      /* 1-2 */
    *reqd_num = Vfull_speed_seg;
    }
  }
 return(disp_code);
}
```

```
/*********************************************************************************/
/***                                                          ***/
/*** Given:  - Train at end of segment                        ***/
/***        - tentative trip time accross seg/sid             ***/
/***             (i.e. arrival time at next segment)          ***/
/***                                                          ***/
/*** Find:   - Dispatch decision (move onto seg, move onto sid)       ***/
/***                                                          ***/
/***        - If held: the time of next dispatch attempt      ***/
/***                                                          ***/
/***        - If dispatched: required speed at end of seg/sid     ***/
/***                                                          ***/
/*********************************************************************************/


/*************************************************************************/
/***                                                     ***/
/*** Segs are A, B1, C, D1, and E;   Sids are B2, and D2.      ***/
/***                                                     ***/
/*** Train to be dispatched is on A.                     ***/
/***                                                     ***/
/***       A      B      C      D      E         ***/
/***              _____         _____             ***/
/***    ___________/______\___________/______\___________       ***/
/***                                                     ***/
/*************************************************************************/


     /*********************************************************************/
     /**                                                       **/
     /** Information needed to make dispatch decision:         **/
     /**                                                       **/
     /**    Direction of trains on B1, B2, C, D1, D2 and E      **/
     /**    relative to train on A.                            **/
     /**                                                       **/
     /**    Arrival time (of train on A) at end of B1/B2       **/
     /**                                                       **/
     /**                                                       **/
     /**    FINAL SPEEDS for:  Halt at end of siding           **/
     /**                       Halt at end of segment          **/
     /**                       Disp onto siding full speed     **/
     /**                       Disp onto segment full speed    **/
     /**                                                       **/
     /**    Free times of B1, B2, C                            **/
     /**                                                       **/
     /**    Dispatch times of trains on D,E                    **/
     /**                                                       **/
     /**    Priorities of trains on D,E (IF THEY ARE IN THE    **/
     /**                         OPPOSITE DIRECTION)      **/
     /**                                                       **/
     /*********************************************************************/


int disp_choice_fr_segment(TRAIN *t_train, double temp_tt, double *reqd_num,
                int unexpected_maint)
 {
    /* if the dispatch function holds the train reqd_number will be  */
    /* the time at which the next departure attempt will be made     */
```

```
   /* if the dispatch function decides to move the train:         */
   /* reqd_number will be the required final velocity determined by */
   /* the dispatch function for the end of the seg/sid             */

   /* unexpected_maint will be == 1 if the train departing the segment */
   /* ran into a maintenance activity on the next seg, AND the sid is  */
   /* not free (i.e. occupied).  It is used to make sure that the train*/
   /* is given access to B1, so that it will be dispatched from its    */
   /* current segment, thereby avoiding a line blockage. The seg free  */
   /* time will be increased by the length of the trip accross it, to  */
   /* simulate the delay to the maintenance crew while it waits for    */
   /* the train to clear the seg. (this last part is handled by        */
   /* disp_from_segment)                                               */

  TRAIN *info_train;

  int loc, dir_val, id_num;

  double arr_time_at_C;

  double Vstop_at_end, Vfull_speed_sid, Vfull_speed_seg, Vcautious;
  double Vmax, Vcautious_sid;
  double init_vel;

  int disp_code;   /* 1 if dispatched onto seg    */
                   /* 2 if dispatched onto sid    */

  int DISP_ONTO_SEG = 1;
  int DISP_ONTO_SID = 2;


  int A_priority=0;
/*   int B_dir = 0; */
  double C_free_time = 0.0;
  int C_dir = 0;
  double D1_disp_time = INFINITY, D1_free_time = 0.0;
  int  D1_dir = 0, D1_priority = 0, D1_trn_mnt=0;
  double D2_disp_time = INFINITY, D2_free_time = 0.0;
  int  D2_dir = 0, D2_priority = 0, D2_trn_mnt=0;
  double E_disp_time = INFINITY, E_free_time = 0.0, E_arr_at_C;
  int  E_dir = 0, E_priority = 0;
  double E_Vmax, E_temp_fin_vel, E_restr_v, E_restr_l, E_tnext, E_clr_time=0.;

  loc = t_train->location;
  dir_val = t_train->direction;
  id_num = t_train->id_number;

  A_priority = t_train->priority;

  arr_time_at_C = tnow + temp_tt;


 /** Calculate possible required speeds at end of segment **/

  Vstop_at_end = 0.0;

  Vmax = minimum(t_train->norm_speed, segment_list[loc+dir_val].max_speed);

  Vfull_speed_seg = minimum(Vmax, segment_list[loc+(2*dir_val)].max_speed);

  if(segment_list[loc+(2*dir_val)].so_end_time >= arr_time_at_C)
   {
   Vfull_speed_seg = max_Vi_so(dir_val, (loc+(2*dir_val)) );
   Vfull_speed_seg = minimum(Vfull_speed_seg, Vmax);
   }

  Vfull_speed_sid = minimum(Vfull_speed_seg, SIDING_SPEED);
```

```
  Vcautious = Vfull_speed_seg/2.0;
  Vcautious_sid = Vfull_speed_sid/2.0;
 /** Finished calculating speed options **/

/** Trains on B must be in opposite direction of train on A      **/
/** if they were in the same direction, the train on A would     **/
/** have been held on the previous sid/seg by the blockage check **/

/** Even if B1 or B2 may not have trains on them, either one might **/
/** still not be free, due to maintenance or a train that has not  **/
/** yet completely cleared it.                       **/


/** It is important to note that either B1 or B2 will be free, or  **/
/** B1 is under maintenance (and unexpected_maint == 1).         **/
/** if this were not the case, the program would never have called **/
/** this function.                                **/
/**                                               **/
/** For all intents and purposes, if unexpected_maint == 1, B1 will**/
/** be considered to be free at tnow.                  **/

/** The dispatch choice algorithm assumes that if B1 and B2 are    **/
/** unoccupied they are also free for travel.             **/

/** For these reasons:                            **/
/**                                               **/
/** If B1 (is occupied OR unoccupied but not yet free)AND         **/
/**     (unexpected_maint != 1)                    **/
/** Then B2 must be free for travel:               **/
/**    The final velocities for and dispatch codes for travel     **/
/**    accross the segment will be set equal to the fin velocities **/
/**    and dispatch codes for travel accross the siding.        **/
/**      Vfull_speed_seg = Vfull_speed_sid;              **/
/**      DISP_ONTO_SEG   = DISP_ONTO_SID;                     **/
/**                                               **/
/** If B2 is occupied OR unoccupied but not yet free:             **/
/** Then B1 must be free for travel:               **/
/**    The final velocities for and dispatch codes for travel     **/
/**    accross the siding will be set equal to the fin velocities **/
/**    and dispatch codes for travel accross the segment         **/
/**      Vfull_speed_sid = Vfull_speed_seg;              **/
/**      DISP_ONTO_SID   = DISP_ONTO_SEG;                     **/

/** Performing the above actions allows the use of only one check  **/
/** for occupancy of B1 or B2. (if B1/B2 is occupied, B2/B1 is not)**/
/** (i.e.  B1_dir ==  0 and B2_dir ==  0  ===> B_dir ==  0        **/
/**      or B1_dir == -1 and B2_dir ==  0  ===> B_dir == -1        **/
/**      or B1_dir ==  0 and B2_dir == -1 ===> B_dir == -1        **/

/** That is to say, disp_code can be set equal to DISP_ONTO_SEG    **/
/** and if seg is not free, disp_code will be assigned the value   **/
/** for DISP_ONTO_SID.                            **/


/** Determine B_dir **/
```

```
/*  if( (segment_list[loc+dir_val].seg_status == 1) ||
      (segment_list[loc+(2*dir_val)].siding_status == 1) )
    B_dir = -1;
   else
    B_dir = 0;
*/
   if( (segment_list[loc+dir_val].segment_free_time >= tnow)&&
      (unexpected_maint == 0) )
    { /* B1 not free */
     Vfull_speed_seg = Vfull_speed_sid;
     DISP_ONTO_SEG   = DISP_ONTO_SID;
     Vcautious = Vfull_speed_sid/2.;
    }
   else if(segment_list[loc+dir_val].siding_free_time>= tnow)
    { /* B2 not free */
     Vfull_speed_sid = Vfull_speed_seg;
     DISP_ONTO_SID   = DISP_ONTO_SEG;
     Vcautious = Vfull_speed_seg/2.;
    }

/** Determine values for C **/
   C_free_time = segment_list[loc+(2*dir_val)].segment_free_time;
   if(segment_list[loc+(2*dir_val)].seg_status == 1)
    {
     if( (segment_list[loc+(2*dir_val)].train_id1 % 2) != (id_num%2) )
      {
       /*trains are in opposite direction*/
       C_dir = -1;
      }
     else  /*trains are in same direction*/
      {
       C_dir = 1;
      }
    }
   else /*C is not occupied by a train*/
    C_dir = 0;

/*****************************************************/
/***  COMPARE ARRIVAL AT C WITH C_FREE_TIME     ***/
/*****************************************************/
/*** If C is not free when the train arrives at ***/
/*** the end of B, it will be told to wait at   ***/
/*** the end of the siding, if it is free.      ***/
/*** If the siding is not free, DISP_ONTO_SID   ***/
/*** will have been set equal to DISP_ONTO_SEG  ***/
/*** anyway, so the train is told to wait at the***/
/*** end of whichever is free (by default).     ***/
/*****************************************************/


   if(C_free_time >= arr_time_at_C)
    {
     *reqd_num = Vstop_at_end;
     return(DISP_ONTO_SID);
    }
```

```
/***!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!***/
/***                                          ***/
/***  If C_dir == 1, the dispatch check will never use any      ***/
/***  information on D1, D2, or E.                        ***/
/***                                          ***/
/***  ===> only calcuiate these values if C_dir == 0          ***/
/***                                          ***/
/***!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!***/


if(C_dir != 1)
 {

/*******************************************************************/
/** IF there are two segments between the train and the station **/
/** (i.e. the station is located at D):                  **/
/**                                           **/
/** D1_dir = -1;                                  **/
/** D1_priority = next_ib_train->priority or              **/
/**           next_ob_train->priority              **/
/**                                           **/
/**                                           **/
/** D2 and E's variables will not be changed from their defaults **/
/** which means that:                              **/
/**                                           **/
/**      D2's variables will all be zeroes                **/
/**                                           **/
/**      E's variables will all be zeroes               **/
/**                                           **/
/**      D2 and E's dispatch times will be INFINITY          **/
/**                                           **/

  if( (dir_val == OUTBOUND) && (loc == (num_segments-2)) )
   { /*there are 2 segments between train and IB station */
    D1_dir = -1;
    D1_priority = next_ib_train->priority;
    D1_trn_mnt = 0;
    D1_disp_time = maximum(next_ib_train->dep_time,(C_free_time+DEP_BUFFER));
    D1_free_time = INFINITY;
   }
  else if( (dir_val == INBOUND) && (loc == 3) )
   { /*train is 2 segments between train and OB station */
    D1_dir = -1;
    D1_priority = next_ob_train->priority;
    D1_trn_mnt = 0;
    D1_disp_time = maximum(next_ob_train->dep_time,(C_free_time+DEP_BUFFER));
    D1_free_time = INFINITY;
   }
  else
   {
    /* train is in middle of line somewhere */
    /* Determine D1_dir, D2_dir, E_dir.    */

   /** Determine values for D1 **/

    D1_free_time = segment_list[loc+(3*dir_val)].segment_free_time;
```

```
if(segment_list[loc+(3*dir_val)].seg_status == 1)
 {
  if( (segment_list[loc+(3*dir_val)].train_id1 % 2) != (id_num%2) )
   {
   /*trains are in opposite direction*/
   D1_disp_time = maximum(segment_list[loc+(3*dir_val)].seg_dispatch,
                   (C_free_time+DEP_BUFFER));
   info_train =
       obtain_from_trainlist(segment_list[loc+(3*dir_val)].train_id1);
   D1_dir = -1;
   D1_priority = info_train->priority;
   if(info_train->failure > 0.0)
    D1_trn_mnt = 1;
   }
  else  /*trains are in same direction*/
   {
   D1_dir = 1;
   D1_disp_time = maximum(segment_list[loc+(3*dir_val)].seg_dispatch,
                   (E_free_time+DEP_BUFFER));
   }
 }
else /*D1 is not occupied by a train*/
 D1_dir = 0;

/** Determine values for D2 **/
D2_free_time = segment_list[loc+(3*dir_val)].siding_free_time;
if(segment_list[loc+(3*dir_val)].siding_status == 1)
 {
  if( (segment_list[loc+(3*dir_val)].train_id2 % 2) != (id_num%2) )
   {
   /*trains are in opposite direction*/
   D2_disp_time = maximum(segment_list[loc+(3*dir_val)].sid_dispatch,
                   (C_free_time+DEP_BUFFER));
   info_train =
       obtain_from_trainlist(segment_list[loc+(3*dir_val)].train_id2);
   D2_dir = -1;
   D2_priority = info_train->priority;
   if(info_train->failure > 0.0)
    D2_trn_mnt = 1;
   }
  else  /*trains are in same direction*/
   {
   D2_dir = 1;
   D2_disp_time = maximum(segment_list[loc+(3*dir_val)].sid_dispatch,
                   (E_free_time+DEP_BUFFER));
   }
 }
else /*D2 is not occupied by a train*/
 D2_dir = 0;

/** Determine values for E **/
E_free_time = segment_list[loc+(4*dir_val)].segment_free_time;
```

```
if(segment_list[loc+(4*dir_val)].seg_status == 1)
 {
  if( (segment_list[loc+(4*dir_val)].train_id1 % 2) != (id_num%2) )
    {
    /*trains are in opposite direction*/

     info_train =
            obtain_from_trainlist(segment_list[loc+(4*dir_val)].train_id1);

     E_dir = -1;

     E_disp_time = segment_list[loc+(4*dir_val)].seg_dispatch;

   /*** The variable E_arr_at_C is never used in cases where
         C_dir == 1.  It is also never used when
         D1 or D2 are occupied by trains in the opposite direction
         of the train on A.  This information can be used to cut
         down on the # of times the variable is calculated ***/

  /*** Actually, if D1_dir or D2_dir == -1, there could not be
        a train of E such that E_dir == -1.
        ==> DO NOT NEED TO CHECK IF D1 == -1 or D2 == -1     ***/

     if(C_dir != 1)
       {
       if( (D1_dir == 1) || ((D2_dir==0)&&(D1_free_time>E_disp_time)) )
         {
         E_Vmax = minimum(info_train->norm_speed, SIDING_SPEED);

         E_temp_fin_vel = E_Vmax;

         E_tnext = maximum(E_disp_time, (D2_free_time+DEP_BUFFER));

         find_restrictions((loc+(4*dir_val)), 0, 1,info_train->train_length,
                               (-dir_val), &E_restr_v, &E_restr_l, E_tnext,
                               info_train->so_at_last_disp);
       }
       else
         {
         E_tnext = maximum(E_disp_time, (D1_free_time+DEP_BUFFER));

         if(segment_list[loc+(3*dir_val)].so_end_time >= E_tnext)
          E_Vmax = minimum(info_train->norm_speed, SO_SPEED);
         else
          E_Vmax = minimum(info_train->norm_speed,
                               segment_list[loc+(3*dir_val)].max_speed);

         E_temp_fin_vel = E_Vmax;

         find_restrictions((loc+(4*dir_val)), 0, 0, info_train->train_length,
                               (-dir_val), &E_restr_v, &E_restr_l, E_tnext,
                               info_train->so_at_last_disp);
         }

       init_vel = info_train->speed;
       if(init_vel > E_Vmax)
        init_vel = E_Vmax;

       E_arr_at_C = tt_X_segment_no_SO(init_vel, &E_temp_fin_vel, E_Vmax,
                                     info_train, (loc+(3*dir_val)),
                                     E_restr_v, E_restr_l, &E_clr_time);

       E_arr_at_C += E_tnext;
       }
```

```
      }
     else  /*trains are in same direction*/
      {
      E_dir = 1;
      }
    }
   else /*E is not occupied by a train*/
    E_dir = 0;
   }
  }
 /*else information on D1, D2, and E was not calculated **/

  /*******************************************************/
  /**                                       **/
  /**  Use segment information to determine dispatch    **/
  /**  action (hold or dispatch), and final speed.     **/
  /**                                       **/
  /*******************************************************/


  /** If train is to be dispatched onto the siding disp_code will be changed,
     to DISP_ONTO_SID. If train is to be dispatched onto the segment,
     nothing will be done to disp_code.                    **/
  /** If train is to be dispatched at other than Vfull_speed_seg, *reqd_num
     will be changed appropriately. If train is to be dispatched at
     Vfull_speed_seg nothing will be done to disp_code.          **/


   disp_code = DISP_ONTO_SEG;
   *reqd_num = Vfull_speed_seg;


    /* train will always be dispatched */
    /* onto whichever (B1, B2) is free */
    /* only difference is final speed  */

    /* Cases where C is occupied by a train in the same direction */
    /* always result with the assignment: */
    /*    disp_code = DISP_ONTO_SID
         *reqd_num = Vstop_at_end    */
    /* This includes                       */
    /* Cases 4-11, 3BC-19, 3BC-20, 3BC-23, 3BC-24, and 2B-10  */
    /* also                                */
    /* 1-6, 2C-1,2  2C-5,6              */
    /* 3C-3,4                       */


   if(C_dir == 1)
    {
     disp_code = DISP_ONTO_SID;
     *reqd_num = Vstop_at_end;
    }
   else if(C_dir == -1)
    {
     /** B must be empty!!!!!**/
    /* it has already been established that C will be free when
       the train on A arrives at the end of B. If this were not true,
       the check immediately after the determination of C's values
       would have returned DISP_ONTO_SID, Vstop_at_end, already.
```

```
===> It can be assumed that the train on C will PROBABLY clear
completely onto B1 (B1 and B2 must be devoid of trains if C_dir == -1)
before the train on A reaches C via B2.

 If the conditions on D1, D2, and E allow dispatching the train
 straight through to C (via B2) it should be done at Vcautious_sid.
 Otherwise the train will be held.
*/
/* train will always be dispatched onto the siding */
disp_code = DISP_ONTO_SID;
/* if conditions allow; the train will be dispatched at
  Vcautious_sid, OTHERWISE *reqd_num will be CHANGED
  to Vstop_at_end*/
*reqd_num = Vcautious_sid;


if(D1_dir == 1)
 {
 if(D2_dir == -1)     /* 3CD-2 */
  {
  if( ( D1_disp_time >= arr_time_at_C) ||
     ( (D2_disp_time < arr_time_at_C) && (D2_priority >= A_priority) )||
     ( (D2_priority > A_priority) && (D2_trn_mnt == 0) ) )
   {
   *reqd_num = Vstop_at_end;
   }
  }
 else if(E_dir == 1)   /* 3C-15 */
  {
  *reqd_num = Vstop_at_end;
  }
 else if(E_dir == -1)  /* 3C-13 */
  {
  if( D1_disp_time < arr_time_at_C)
   {
   if(A_priority > E_priority)
    {
    }
   else if( ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))
         || (E_arr_at_C < arr_time_at_C) )
    {
    *reqd_num = Vstop_at_end;
    }
   }
  else
   {
   *reqd_num = Vstop_at_end;
   }
  }
 else              /* 2C-9 */
  {
  if( D1_disp_time >= arr_time_at_C )
   {
   *reqd_num = Vstop_at_end;
   }
  }
 }
else if(D1_dir == -1)
 {
```

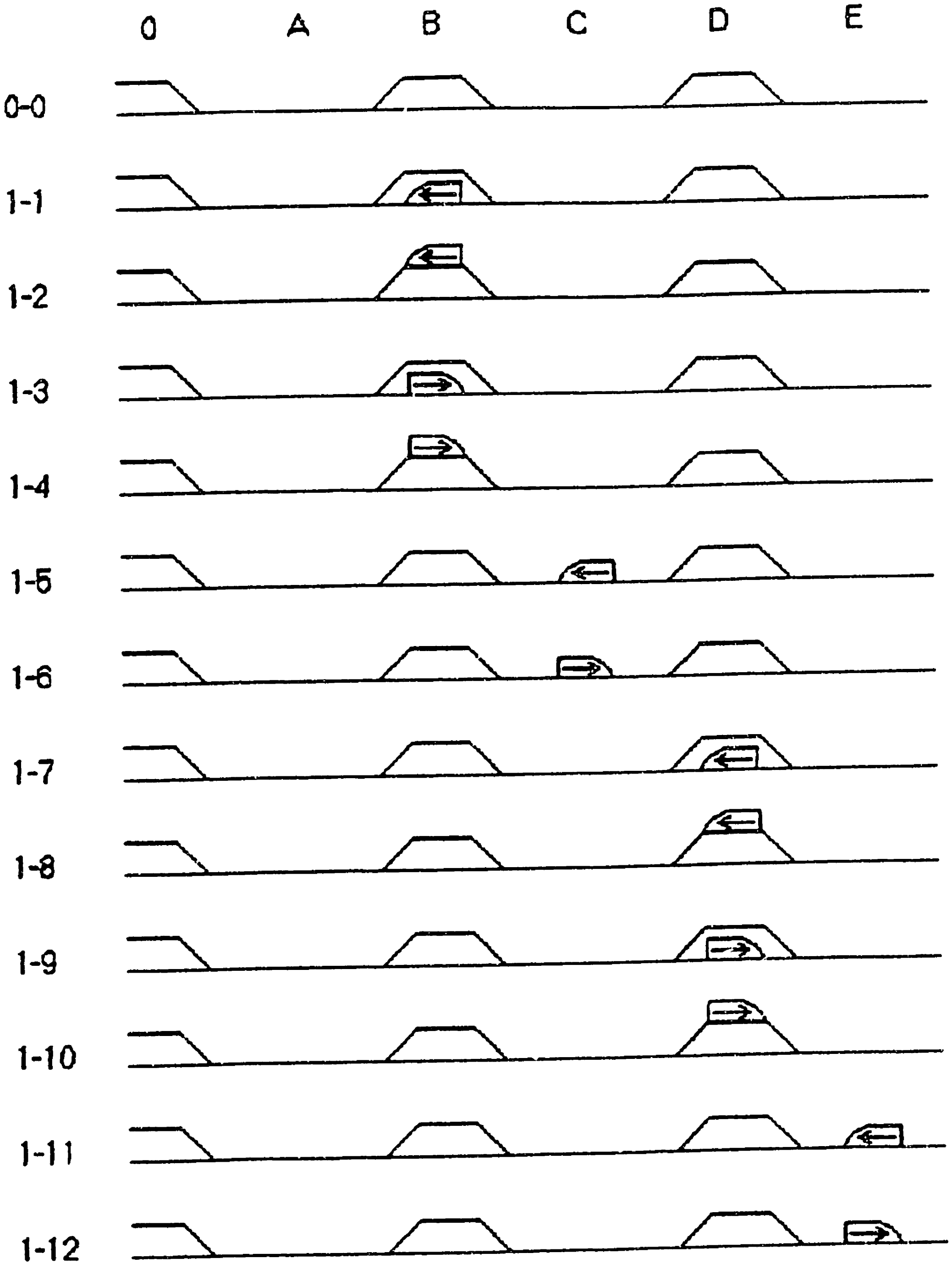
```
    if(D2_dir == 1)       /* 3CD-1 */
      {
       if( ( D2_disp_time >= arr_time_at_C) ||
             ( (D1_disp_time < arr_time_at_C) && (D1_priority >= A_priority) )||
             ( (D1_priority > A_priority) && (D1_trn_mnt == 0) ) )
         {
          *reqd_num = Vstop_at_end;
         }
      }
    else /* E_dir == 1 OR E is unoccupied */ /* 3C-11 */
      {                                   /* 2C-7 */
       if((D2_free_time >= arr_time_at_C)||
           ((D1_disp_time < arr_time_at_C) && (D1_priority >= A_priority))||
           ((D1_priority > A_priority) && (D1_trn_mnt == 0)) )
         {
          *reqd_num = Vstop_at_end;
         }
      }
   }
  /* D1 is unoccupied */
 else if(D2_dir == 1)
   {
    if(E_dir == 1)        /* 3C-16 */
      {
       *reqd_num = Vstop_at_end;
      }
    else if(E_dir == -1)  /* 3C-14 */
      {
       if( D2_disp_time < arr_time_at_C)
         {
         if(A_priority > E_priority)
            {
            }
         else if( ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))
                      || (E_arr_at_C < arr_time_at_C) )
            {
             *reqd_num = Vstop_at_end;
            }
         }
       else
         {
          *reqd_num = Vstop_at_end;
         }
      }
    else                  /* 2C-10 */
      {
       if( D2_disp_time >= arr_time_at_C )
         {
          *reqd_num = Vstop_at_end;
         }
      }
   }
 else if(D2_dir == -1)
   {
     /* E_dir == 1 OR E is unoccupied */ /* 3C-12 */
                                       /* 2C-8  */
    if((D1_free_time >= arr_time_at_C)||
       ((D2_disp_time < arr_time_at_C) && (D2_priority >= A_priority))||
       ((D2_priority > A_priority) && (D2_trn_mnt == 0)) )
      {
```

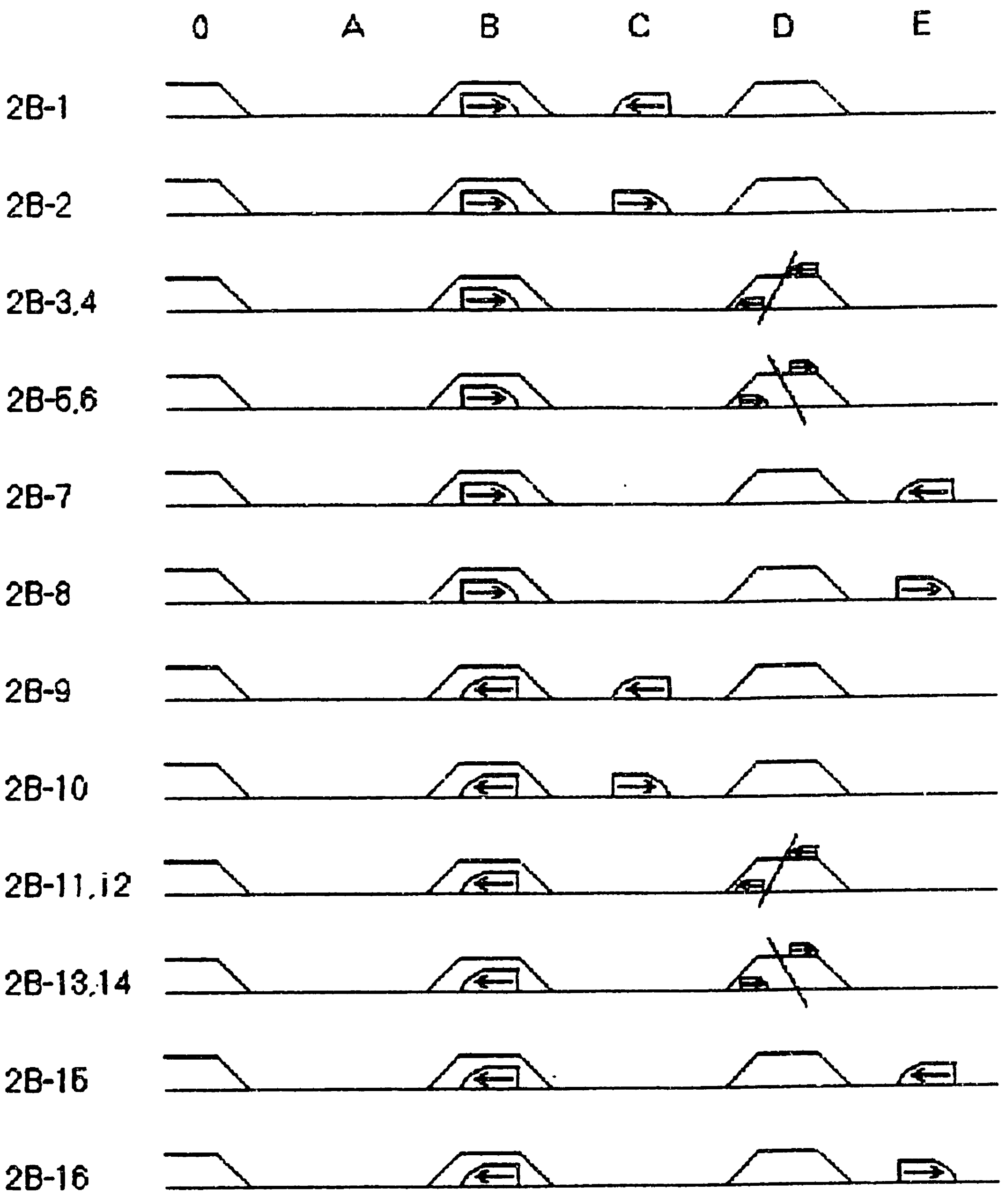
```
      *reqd_num = Vstop_at_end;
     }
    }
  /* D2 is unoccupied */
   else if(E_dir == 1)     /* 2C-12 */
    {
    }
   else if(E_dir ==-1)     /* 2C-11 */
    {
     if((D1_free_time >= arr_time_at_C)||
       (D2_free_time >= arr_time_at_C)||
       ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))||
       ((E_priority >= A_priority) &&(E_arr_at_C < arr_time_at_C)) )
      {
      *reqd_num = Vstop_at_end;
      }
    }
   /* else E is unoccupied */ /* 1-5 */

   }
  /* C is unoccupied */


  /* As long as C is unoccupied, the remaining cases are treated as if
     B1 and B2 are free.  If either B1 or B2 is not free the proper
     variables will have been changed as explained above.
     ( for example: if B1 is not free then DISP_ONTO_SEG will have been
       set equal to DISP_ONTO_SID.)
    This way, no matter what assignments are made, if B1 or B2 happens
    to be unavailable, disp_code and reqd_num will still be assigned
    to correct values  */
   else if(D1_dir == 1)
    {
     if(D2_dir == -1)
      {
       if(E_dir == 1)       /* 3D-2 */ /* 4BDE-4 */
        {
         disp_code = DISP_ONTO_SID;
         *reqd_num = Vstop_at_end;
        }
       else                 /* 2D0-2 */ /* 3BD0-4 */
        {
         if( ( D1_disp_time >= arr_time_at_C) ||
            ( (D2_disp_time < arr_time_at_C) && (D2_priority >= A_priority) )||
            ( (D2_priority > A_priority) && (D2_trn_mnt == 0) ) )
          {
          disp_code = DISP_ONTO_SID;
          *reqd_num = Vstop_at_end;
          }
        }
      }
     else if(E_dir == 1)    /* 2D-3 */ /* 3BD-15 */
      {
        disp_code = DISP_ONTO_SID;
        *reqd_num = Vstop_at_end;
      }
     else if(E_dir == -1)   /* 2D-1 */ /* 3BD-13 */
      {
       if( D1_disp_time < arr_time_at_C)
        {
```

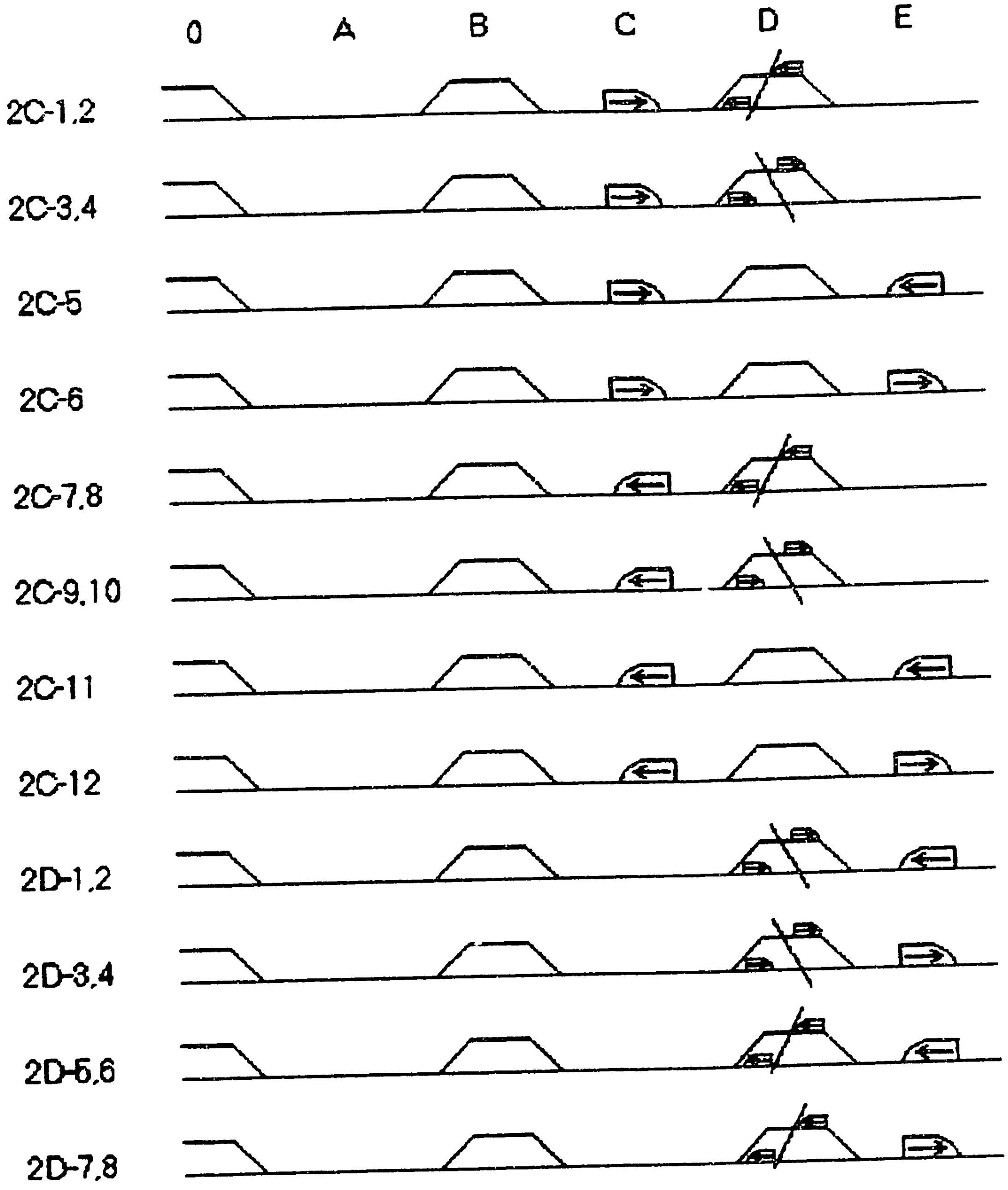
```
      if(A_priority > E_priority)
       {
       }
      else if( ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))
            || (E_arr_at_C < arr_time_at_C) )
       {
        disp_code = DISP_ONTO_SID;
        *reqd_num = Vstop_at_end;
       }
      else
       {
        *reqd_num = Vcautious;
       }
     }
    else
     {
      disp_code = DISP_ONTO_SID;
      *reqd_num = Vstop_at_end;
     }
   }
  else              /* 1-9 */ /* 2B-13 */
   {
    if( D1_disp_time >= arr_time_at_C )
     {
      disp_code = DISP_ONTO_SID;
      *reqd_num = Vstop_at_end;
     }
    else
     {
/*?*/   *reqd_num = Vcautious; /* TRY A RUN WITH Vfull_speed_seg */
     }
   }
 }
else if(D1_dir == -1)
 {
  if(D2_dir == 1)
   {
    if(E_dir == 1)       /* 3D-4 */ /* 4BDE-2 */
     {
       disp_code = DISP_ONTO_SID;
       *reqd_num = Vstop_at_end;
     }
    else              /* 2D0-1 */ /* 2BD0-3 */
     {
      if( ( D2_disp_time >= arr_time_at_C) ||
         ( (D1_disp_time < arr_time_at_C) && (D1_priority >= A_priority) )||
         ( (D1_priority > A_priority) && (D1_trn_mnt == 0) ) )
       {
        disp_code = DISP_ONTO_SID;
        *reqd_num = Vstop_at_end;
       }
     }
   }
  else /* E_dir == 1 OR E is unoccupied */ /* 2D-7 */ /* 3BD-11 */
   {                             /* 1-7 */ /* 2B-11 */
    if((D2_free_time >= arr_time_at_C)||
      ((D1_disp_time < arr_time_at_C) && (D1_priority >= A_priority))||
      ((D1_priority > A_priority) && (D1_trn_mnt == 0)) )
     {
      disp_code =DISP_ONTO_SID;
```

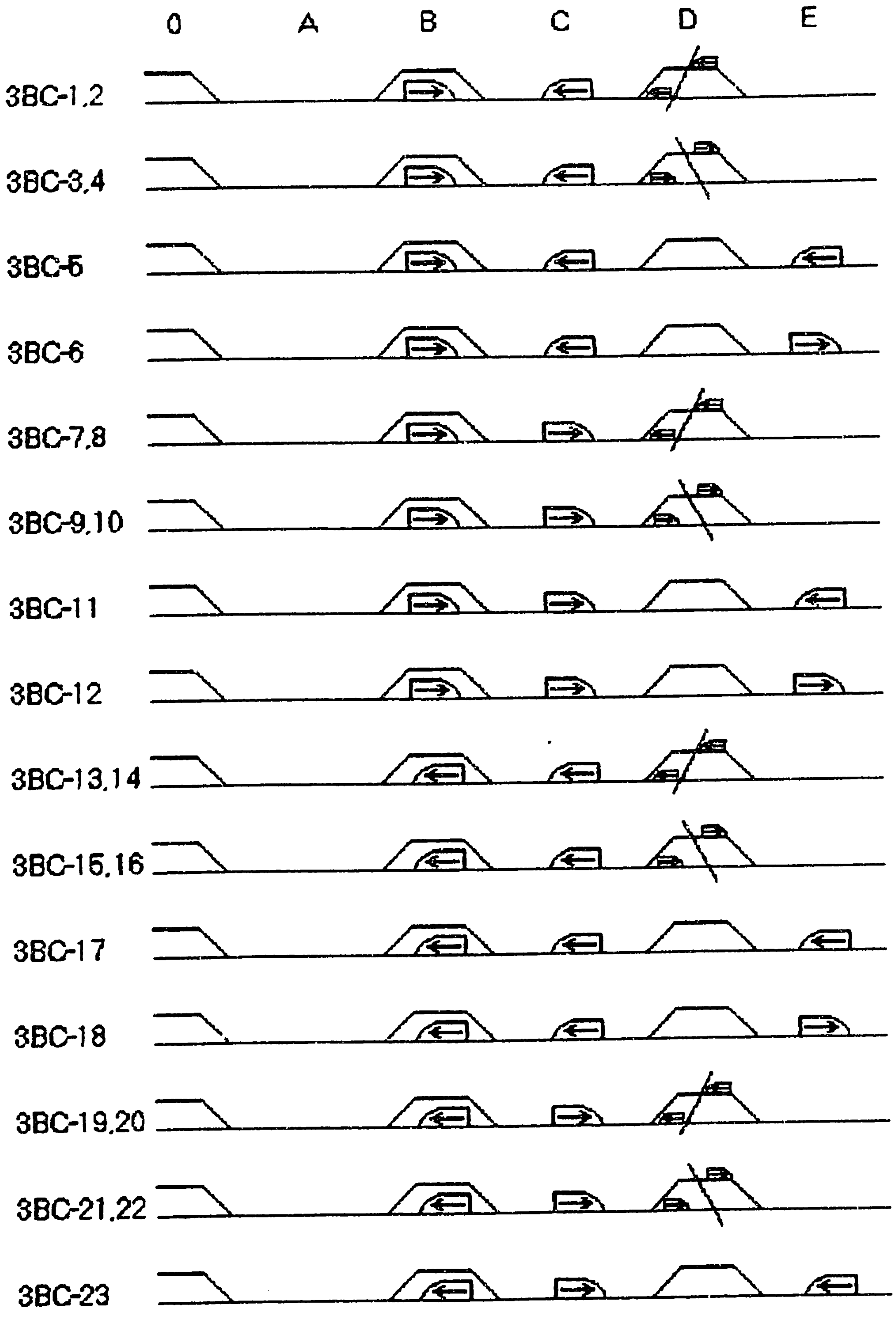
```
            *reqd_num = Vstop_at_end;
          }
        }
      }
/* D1 is unoccupied */
   else if(D2_dir == 1)
     {
      if(E_dir == 1)        /* 2D-4 */ /* 3BD-16 */
        {
         disp_code =DISP_ONTO_SID;
         *reqd_num = Vstop_at_end;
        }
      else if(E_dir == -1)    /* 2D-2 *//* 3BD-14 */
        {
         if( D2_disp_time < arr_time_at_C)
           {
            if(A_priority > E_priority)
              {
              }
            else if( ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))
                         || (E_arr_at_C < arr_time_at_C) )
              {
               disp_code = DISP_ONTO_SID;
               *reqd_num = Vstop_at_end;
              }
            else
              {
               *reqd_num = Vcautious;
              }
           }
         else
           {
            disp_code = DISP_ONTO_SID;
            *reqd_num = Vstop_at_end;
           }
        }
      else                /* 1-10 */ /* 2B-14 */
        {
         if( D2_disp_time >= arr_time_at_C )
           {
            disp_code = DISP_ONTO_SID;
            *reqd_num = Vstop_at_end;
           }
         else
           {
/*?*/            *reqd_num = Vcautious; /* TRY A RUN WITH Vfull_speed_seg */
           }
        }
      }
    else if(D2_dir == -1)
      {
      /* E_dir == 1 OR E is unoccupied */  /* 2D-8 */  /* 3BD-12 */
                                                             /* 1-8 */  /* 2B-12 */
       if((D1_free_time >= arr_time_at_C)||
          ((D2_disp_time < arr_time_at_C) && (D2_priority >= A_priority))||
          ((D2_priority > A_priority) && (D2_trn_mnt == 0)) )
          {
          disp_code =DISP_ONTO_SID;
          *reqd_num = Vstop_at_end;
          }
```

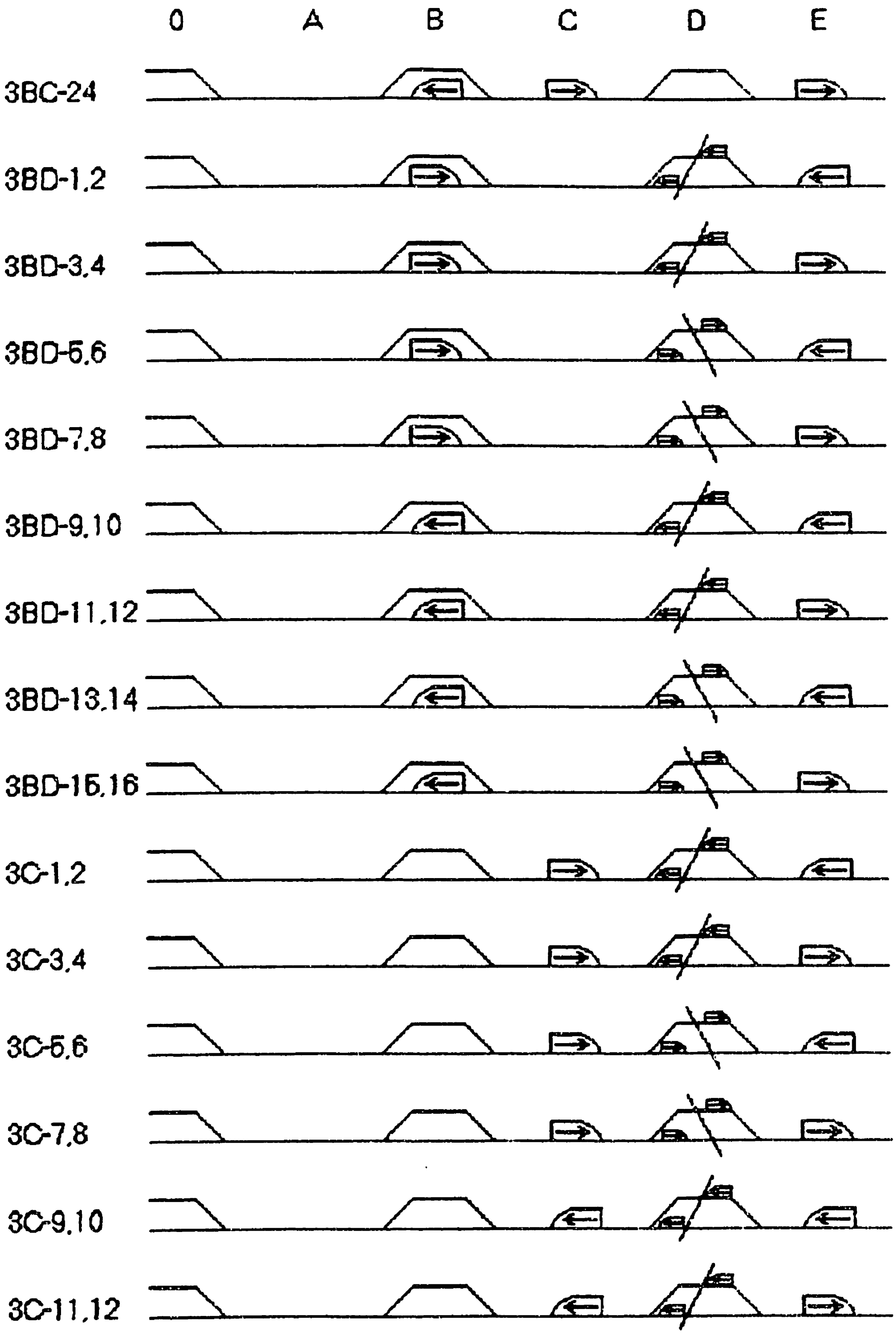
```
   }
/* D2 is unoccupied */
 else if(E_dir == 1)       /* 1-12 */ /* 2B-16 */
  {
  }
 else if(E_dir == -1)    /* 1-11 */ /* 2B-15 */
  {
   if((D1_free_time >= arr_time_at_C)||
     (D2_free_time >= arr_time_at_C)||
    ((E_priority > A_priority) && (E_disp_time < arr_time_at_C))||
    ((E_priority >= A_priority) &&(E_arr_at_C < arr_time_at_C)) )
    {
    disp_code = DISP_ONTO_SID;
    *reqd_num = Vstop_at_end;
    }
  }
 /* else  C, D, and E are empty */  /* 1-1 */
                                   /* 1-2 */

 return(disp_code);
 }
```

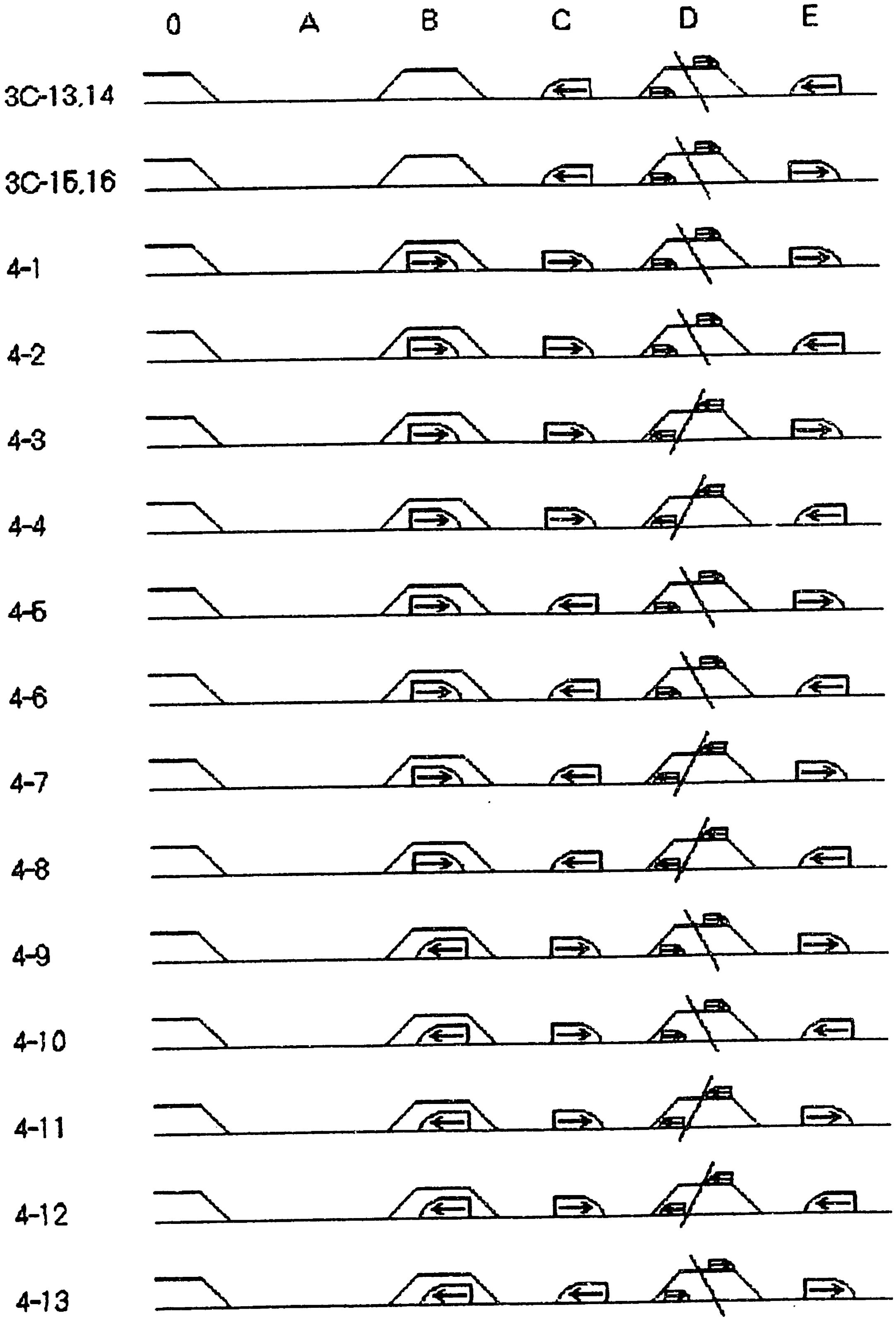

0
A
B
C
D
E
0-0
1-1
1-2
1-3
1-4
1-5
1-6
1-7
1-8
1-9
1-10
1-11
1-12

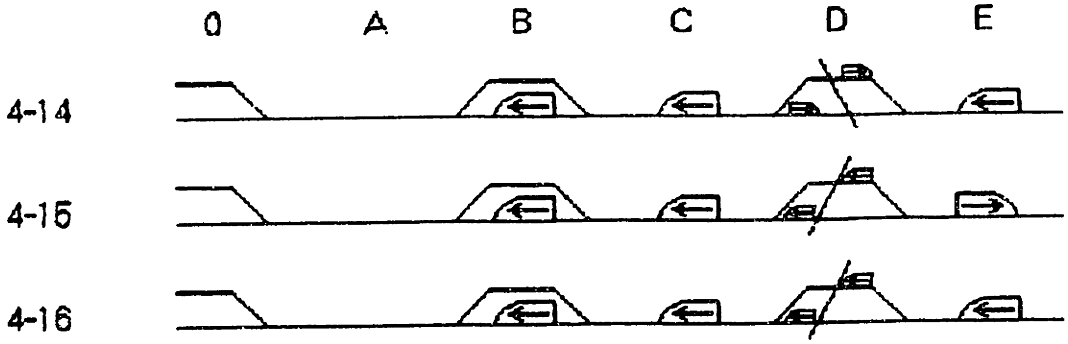

0
A
B
C
D
E
2B-1
2B-2
2B-3,4
2B-5,6
2B-7
2B-8
2B-9
2B-10
2B-11,12
2B-13,14
2B-15
2B-16

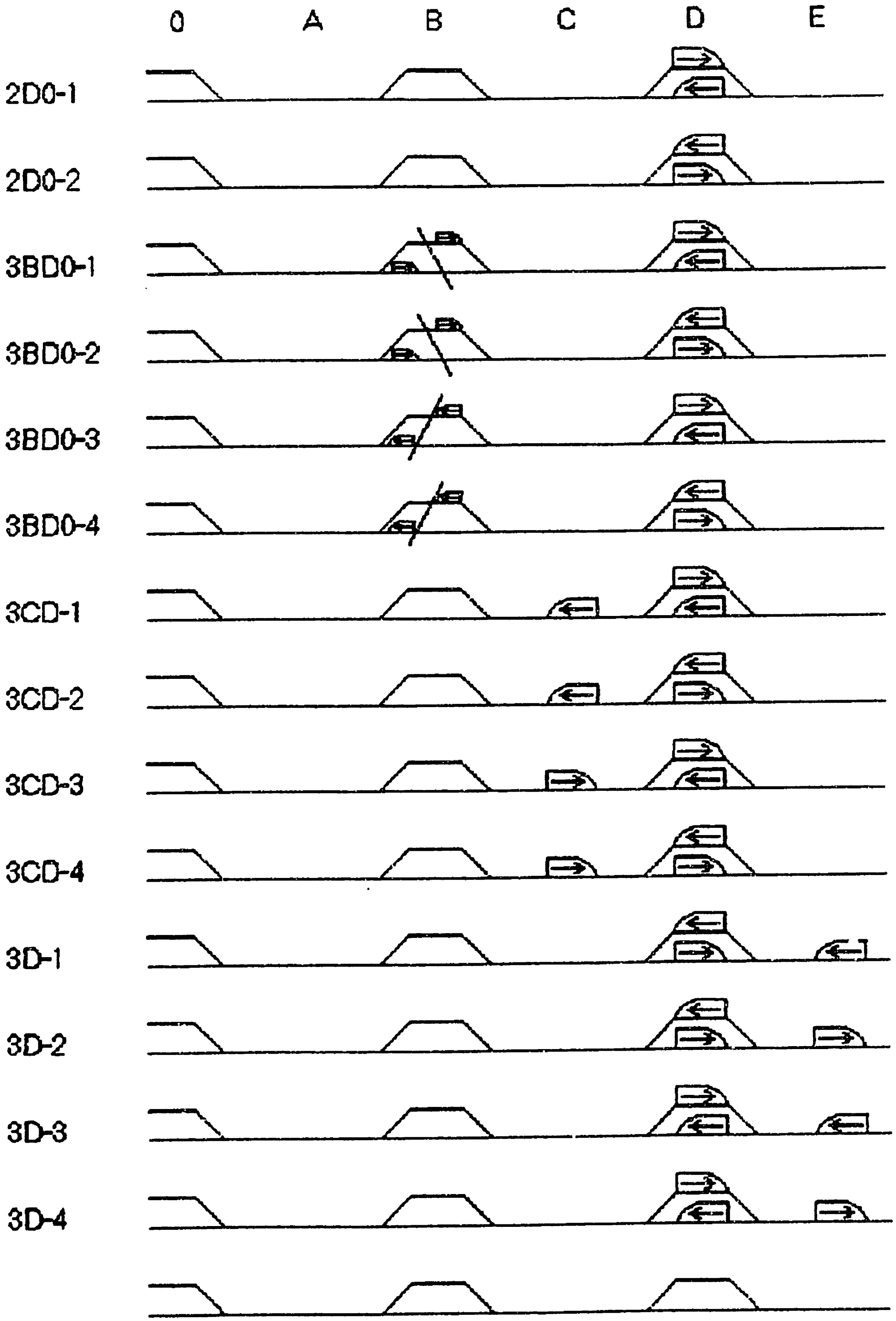

0
A
B
C
D
E
2C-1,2
2C-3,4
2C-5
2C-6
2C-7,8
2C-9,10
2C-11
2C-12
2D-1,2
2D-3,4
2D-5,6
2D-7,8

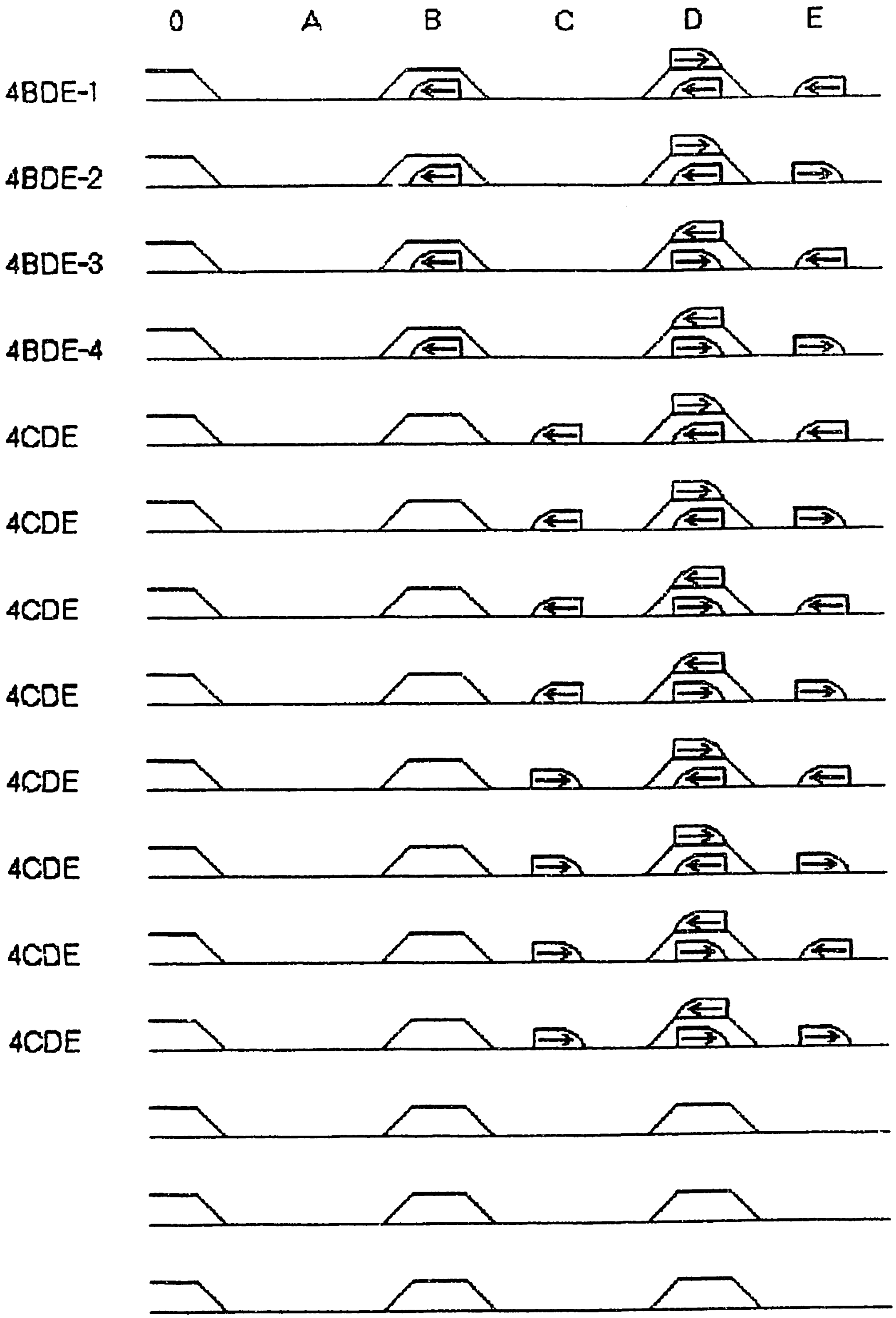

0
A
B
C
D
E
3BC-1,2
3BC-3,4
3BC-5
3BC-6
3BC-7,8
3BC-9,10
3BC-11
3BC-12
3BC-13,14
3BC-15,16
3BC-17
3BC-18
3BC-19,20
3BC-21,22
3BC-23


0
A
B
C
D
E
3BC-24
3BD-1,2
3BD-3,4
3BD-5,6
3BD-7,8
3BD-9,10
3BD-11,12
3BD-13,14
3BD-15,16
3C-1,2
3C-3,4
3C-5,6
3C-7,8
3C-9,10
3C-11,12


0
A
B
C
D
E
3C-13,14
3C-15,16
4-1
4-2
4-3
4-4
4-5
4-6
4-7
4-8
4-9
4-10
4-11
4-12
4-13


0
A
B
C
D
E
4-14
4-15
4-16


0
A
B
C
D
E
2D0-1
2D0-2
3BD0-1
3BD0-2
3BD0-3
3BD0-4
3CD-1
3CD-2
3CD-3
3CD-4
3D-1
3D-2
3D-3
3D-4


0
A
B
C
D
E
4BDE-1
4BDE-2
4BDE-3
4BDE-4
4CDE
4CDE
4CDE
4CDE
4CDE
4CDE
4CDE
4CDE

Appendix C

This appendix contains the data from which the figures in chapter 5 were generated. Pages 197-198 are the data from runs with no planned maintenance and increasing defect rate. Pages 199-200 are the data from runs with increasing planned maintenance loads. The defect file used in these two pages was the below average (summer) defect rate file. Pages 201-204 are the data from the "Cold-Snap" runs.

80 mgt line

track defect file	act'l t/d	avg engr delay	avg ops delay	avg dep delay	avg tot delay	avg trip time	trip time std dev
33 tons							
ZERO	23.1	0	53.34	0.45	53.79	491.34	26.52
TRNONLY	23.1	2.94	71.29	0.92	75.15	512.23	71.48
LOW	23.1	2.97	107.29	5.74	116	548.26	73.06
AVG	23.1	2.3	143.52	5.25	151.08	583.83	90.55
HIGH	23.1	2.46	179.42	4.89	186.78	619.88	92.01
36 tons							
ZERO	20.9	0	51.64	0.75	52.39	489.64	24.37
TRNONLY	20.9	3.28	57.1	2.51	62.88	498.37	48.26
A	20.9	3.69	104.46	3.27	111.42	546.15	87.72
B	20.9	2.72	130.09	3.03	135.84	570.82	81.7
C	20.9	3	171.54	4.19	178.72	612.53	93.79
39 tons							
ZERO	19.1	0	48.24	0.69	48.93	486.24	24.51
TRNONLY	19.1	3.23	62.87	0.45	66.55	504.1	73.14
A	19.1	2.51	89.57	4.68	96.76	530.08	59.45
B	19.1	2.48	130.46	2.54	135.48	570.94	84.18
C	19.1	4.04	150.28	4.23	158.55	592.32	92.34

30 mgt line

track defect file	act'l t/d	avg engr delay	avg ops delay	avg dep delay	avg tot delay	avg trip time	trip time std dev
33 tons							
ZERO	8.9	0	20.73	0.2	20.93	458.73	20.47
TRNONLY	8.9	1.11	21.96	0.22	23.29	461.07	28.71
A	8.9	2.35	39.33	1.78	43.46	479.68	34.14
B	8.9	2.35	70.08	0.42	72.84	510.43	57.76
C	8.9	2.39	89.91	0.21	92.51	530.3	58.9
36 tons							
ZERO	8	0	19.81	0	19.81	457.81	17.17
TRNONLY	8	2.65	21.35	0	23.99	461.99	33.32
A	8	2.65	44.14	0.03	46.81	484.79	46.38
B	8	0	65.49	1.02	66.5	503.49	42.62
C	8	2.65	94.89	1.71	99.24	535.54	56.57
39 tons							
ZERO	7.4	0	18.09	0	18.09	456.09	17.23
TRNONLY	7.4	4.15	19.61	0	23.76	461.76	38.78
A	7.4	2.82	38.39	0.21	41.42	479.21	36.14
B	7.4	2.87	65.09	0.55	68.51	505.96	51.31
C	7.4	2.87	86.23	0.68	89.78	527.1	63.15

```
PLANNED MAINTENANCE STUDY FOR 80 MGT LINE 33 TON AXLE LOAD CASE

MW33A.OUT          ALL RUNS USED   W33A.RND

                                                                   trip
ALL TRNS                  avg      avg      avg      avg      avg   time
MAINT DAY    act'l       engr      ops      dep      tot     trip    std
MAINT WEE      t/d      delay    delay    delay    delay     time    dev

NO PLANNED MAINTENANCE
ALL TRNS      23.1       3.49   112.56     1.76   117.81   554.05   87.35
211-318       22.2       2.02    90.33     1.39    93.74   530.35   55.94
211-372         23       3.96   128.75     2.21   134.92   570.71  103.31

4 HOUR TRACK MAINTENANCE
ALL TRNS      23.1       3.12   114.24     5.17   122.52   555.35   88.56
211-318       22.2          0   107.38     0.86   108.24   545.38   62.57
211-372       23.2       1.58   130.16     1.66    133.4   569.74   99.57

6 HOUR TRACK MAINTENANCE
ALL TRNS      23.1       3.13   128.02     1.87   133.02   569.15   94.73
211-318       22.5       2.02   185.37     2.08   189.47   625.39    99.8
211-372       23.1       2.81   181.71     2.55   187.07   622.53  110.43

8 HOUR TRACK MAINTENANCE
ALL TRNS      23.1       2.61   157.11     5.26   164.98   597.72   135.1
211-318       22.6       2.04   344.01      1.6   347.66   784.05  151.68
211-372         23       1.24   257.87     1.97   261.08   697.11  163.53
```

```
PLANNED MAINTENANCE STUDY FOR 30 MGT LINE 33 TON AXLE LOAD CASE
                                                                        trip
ME33A.OUT          ALL RUNS USED   E33A.RND
                                                                        time
ALL TRNS                  avg       avg       avg       avg       avg    std
MAINT DAY     act'l      engr       ops       dep       tot      trip    dev
MAINT WEE       t/d     delay     delay     delay     delay      time

NO PLANNED MAINTENANCE
ALL TRNS        8.9      1.07      39.3       1.7     42.07    478.37  29.58
83-118          8.3         0     41.32         0     41.32    479.32  21.09
79-140            9         0      46.9      0.43     47.33     484.9  23.87

4 HOUR TRACK MAINTENANCE
ALL TRNS        8.9      2.35     44.75      1.91     49.01     485.1  39.43
83-118            8         0     61.21      0.15     61.36    499.21  22.51
79-140            9      3.85     63.35      1.05     68.26     505.2   45.6

6 HOUR TRACK MAINTENANCE
ALL TRNS        8.9      2.35     44.75      1.91     49.01     485.1  39.43
83-118            8         0     61.21      0.15     61.36    499.21  22.51
79-140            9      3.85     63.35      1.05     68.26     505.2   45.6

8 HOUR TRACK MAINTENANCE
ALL TRNS        8.9      1.07     44.01      1.82      46.9    483.09  31.65
83-118            8         0     61.21      0.15     61.36    499.21  22.51
79-140            9         0     61.12      0.77     61.89    499.12  25.11
```

80 MGT   OFF PEAK   COLD SNAP

	IMPACT DIRECTLY			IMPACTED DRCTLY AND BY SO		
263	253	258	6	253	274	22
286	226	230	5	226	245	20
315	210	214	5	210	225	16

	MMBF	avg engr delay	avg ops delay	avg dep delay	avg tot delay	avg trip time	time std dev
	263						
ALL TRNS	25000	3.38	211	7.62	222.01	652.38	127.13
DIRECTLY	25000	0	548.48	5.7	554.18	986.48	46.24
DIR & SO	25000	0	498.07	22.96	521.03	936.07	115.7
	286						
ALL TRNS	25000	2.83	186.01	5.14	193.98	626.85	107.21
DIRECTLY	25000	0	496.74	3.73	500.47	934.74	45.79
DIR & SO	25000	0	362.35	21.54	383.88	800.35	117.27
	315						
ALL TRNS	25000	3.14	185.23	3.53	191.9	626.37	118.26
DIRECTLY	25000	0	419.2	3.34	422.53	857.2	63.1
DIR & SO	25000	0	348.43	12.61	361.04	786.43	126.11
SAME TRAINS:	NO COLD SNAP						
	263						
ALL TRNS	25000	2.46	179.42	4.89	186.78	619.88	92.01
DIRECTLY	25000	0	136.53	5.7	142.24	574.53	36.1
DIR & SO	25000	0	109.83	4.19	114.02	547.83	34.16
	286						
ALL TRNS	25000	3	171.54	4.19	178.72	612.53	93.79
DIRECTLY	25000	0	100.07	3.73	103.8	538.07	29.77
DIR & SO	25000	0	105.98	4.5	110.47	543.98	31.99
	315						
ALL TRNS	25000	4.14	163.47	3.38	170.99	605.61	87.66
DIRECTLY	25000	0	92.17	6.24	98.42	530.17	14.32
DIR & SO	25000	0	97.78	2.82	100.6	535.78	34.7

80 MGT PEAK COLD SNAP

		IMPACT DIRECTLY			IMPACTED DRCTLY AND BY SO		
	263	345	357	13	345	372	28
	286	308	318	11	308	332	25
	315	283	294	12	283	308	26
	act'l t/d	avg engr delay	avg ops delay	avg dep delay	avg tot delay	avg trip time	trip time std dev
ALL TRNS	23.1	2.05	217.05	8.28	227.38	657.1	149.04
DIRECTLY	18.7	0	597.81	27.26	625.07	1035.81	73.17
DIR & SO	24.5	0	601.78	20.91	622.69	1039.78	116.82
ALL TRNS	20.9	2.43	176.97	4.55	183.95	617.4	101.42
DIRECTLY	27.3	0	499.05	19.07	518.12	937.05	68.32
DIR & SO	22.8	0	415.88	22.39	438.27	853.88	105.16
ALL TRNS	19.1	2.56	175.88	4.65	183.09	616.44	118.03
DIRECTLY	22.4	0	587.6	20.76	608.36	1025.6	96.87
DIR & SO	22	0	493.38	32.05	525.43	931.38	121.76

SAME TRAINS: NO COLD SNAP

ALL TRNS	23.1	2.46	179.42	4.89	186.78	619.88	92.01
DIRECTLY	32	0	212.95	0.46	213.4	650.95	32.82
DIR & SO	24.3	0	246.28	0.87	247.15	684.28	71.61
ALL TRNS	20.9	3	171.54	4.19	178.72	612.53	93.79
DIRECTLY	24.8	0	178.07	2.65	180.72	616.07	29.86
DIR & SO	21.9	0	175.03	1.84	176.87	613.03	27.05
ALL TRNS	19.1	4.14	163.47	3.38	170.99	605.61	87.66
DIRECTLY	23.9	0	263.43	1.9	265.33	701.43	79.99
DIR & SO	19.9	0	237.47	2.65	240.12	675.47	67.64

30 MGT OFF PEAK COLD SNAP

		IMPACT DIRECTLY			IMPACTED DRCTLY AND BY SO		
	263	95	98	4	95	102	8
	286	85	88	4	85	92	8
	315	79	83	5	79	84	6
	act'l t/d	avg engr delay	avg ops delay	avg dep delay	avg tot delay	avg trip time	trip time std dev
ALL TRNS	8.9	2.39	98.63	0.31	101.33	539.02	73.1
DIRECTLY	51.4	0	365.78	3.27	369.04	803.78	94.35
DIR & SO	7.4	0	234.28	3.58	237.86	672.28	132.4
ALL TRNS	8	2.65	104.13	1.71	108.49	544.78	70.76
DIRECTLY	11.2	0	382.7	0	382.7	820.7	36.67
DIR & SO	7.4	0	253.28	0	253.28	691.28	132.93
ALL TRNS	7.4	2.87	94.87	0.78	98.52	535.74	76.71
DIRECTLY	4.4	0	310.52	3.03	313.55	748.52	120.98
DIR & SO	5.6	0	276.52	2.52	279.04	714.52	134.09
SAME TRAINS: NO COLD SNAP							
ALL TRNS	8.9	2.39	89.91	0.21	92.51	530.3	58.9
DIRECTLY	10.5	0	69.09	1.63	70.73	507.09	22.96
DIR & SO	6.7	0	56.07	0.93	57	494.07	25.05
ALL TRNS	8	2.65	94.89	1.71	99.24	535.54	56.57
DIRECTLY	0	0	64.9	0	64.9	502.9	21.7
DIR & SO	8.5	0	53.27	0	53.27	491.27	18.51
ALL TRNS	7.4	2.87	86.23	0.68	89.78	527.1	63.15
DIRECTLY	5.9	0	64.34	0	64.34	502.34	21.36
DIR & SO	7.1	0	56.98	0	56.98	494.98	20.24

```
30 MGT       PEAK         COLD SNAP


                                                          IMPACTED
                        IMPACT  DIRECTLY                DRCTLY AND BY SO
                263       128       134         7       128       142        15
                286       119       122         4       119       127         9
                315       109       112         4       109       118        10
                                                                           trip
                          avg       avg       avg       avg       avg      time
              'act'l     engr       ops       dep       tot      trip       std
                t/d     delay     delay     delay     delay      time       dev



ALL TRNS        8.9      2.39     100.6       0.3    103.29    540.99     80.91
DIRECTLY       12.2         0    404.87         0    404.87    842.87     104.9
DIR & SO       10.8     13.88    267.86         0    281.75    719.75    140.33


ALL TRNS          8         0     103.5      1.65    105.16     541.5     71.83
DIRECTLY        8.2         0    361.63         0    361.63    799.63     132.2
DIR & SO        8.9         0    264.85         0    264.85    702.85    146.48


ALL TRNS        7.4      2.87     97.97      0.81    101.65    538.84     82.87
DIRECTLY        9.9         0    356.83         0    356.83    794.83    122.33
DIR & SO        6.8         0     273.8      2.24    276.04     711.8    122.01

SAME TRAINS:  NO COLD SNAP


ALL TRNS        8.9      2.39     89.91      0.21     92.51     530.3      58.9
DIRECTLY       10.6         0    124.21         0    124.21    562.21     98.09
DIR & SO       10.5     14.88    106.25         0    121.12    559.12     90.15


ALL TRNS          8      2.65     94.89      1.71     99.24    535.54     56.57
DIRECTLY       12.5         0     93.12         0     93.12    531.12     44.46
DIR & SO        8.5     26.53      84.2         0    110.73    548.73     90.28


ALL TRNS        7.4      2.87     86.23      0.68     89.78     527.1     63.15
DIRECTLY       11.6         0     74.32         0     74.32    512.32      6.58
DIR & SO        9.9         0     95.08      0.47     95.55    533.08      42.9
```