PROCEEDINGS

Twenty-seventh Annual Meeting

Volume XXVII • Number 1 1986

TRANSPORTATION RESEARCH FORUM

PROCEEDINGS—

Twenty-seventh Annual Meeting

September 22-24, 1986
Seattle, Washington

Volume XXVII • Number 1 1986

TRANSPORTATION RESEARCH FORUM
In conjunction with

CANADIAN TRANSPORTATION
RESEARCH FORUM

The Automated Blocking Model: A Practical Approach to Freight Railroad Blocking Plan Development

*By Carl Van Dyke**

ABSTRACT

A railroad blocking or classification plan is the set of rules by which a railroad groups its cars to be moved by trains from location to location to their ultimate destination. This paper reviews past methods for computerized railroad blocking plan development, and how these techniques were adapted in the creation of the new Automated Blocking Model.

I. INTRODUCTION

The gathering, grouping, movement, and distribution of railroad freight cars is an extremely complex problem. Traditionally, this has been a manual process, where individuals make incremental changes to an already existing plan. As railroads have continued to merge and differentiate their service, this problem has become increasingly complex. Today, the thought of trying to manually create a coherent operating plan for one of America's mega-railroads is extremely daunting.

To address these types of problems railroads have turned to computers. However, the railroad operating plan development problem is considered too large and complex to be solved in its entirety using computerized optimization techniques. Furthermore, methods that take the railroad planner out of the process have not been well received in the industry, and have often poorly represented reality.

Traditionally, the problem has been broken into three components: 1) traffic gathering and distribution, 2) blocking, and 3) train scheduling. "Traffic gathering" is the movement of individual cars between various points of origination or termination and central locations (yards) for processing. "Blocking" is the grouping of cars for joint movement to either their destination or another location for further processing. "Train scheduling" is the development of routes and timetables for trains, and the assignment of blocks to those trains.

The Service Planning Model (SPM) was developed in the late 1970's to assist railroads in the evaluation of one or more operating plans. However, the creation of these operating plans is left to manual methods or ad hoc methods developed by individual railroads. To strengthen the planning process a number of railroads using the SPM asked the Association of American Railroads to fund the creation of railroad blocking plan software compatible with the SPM. In response the Federal Railroad Administration and the Association of American Railroads have jointly funded the development of the Automated Blocking Model (ABM).

The ABM is designed to be the first of two software packages to assist a railroad in developing its operating plan. The ABM primarily addresses blocking and partially addresses gathering and distribution. Train scheduling is expected to be addressed by a follow-on program to be developed in the future. Both packages are intended to be compatible with the SPM. They allow the SPM to be used in the resulting plan's evaluation.

Numerous attempts have been made at developing computerized methods of developing blocking plans. Among them, the ABM is unique for the following reasons:

1) It takes a practical, results oriented approach.
2) Its development was at the request, and under the supervision of several railroad operations planners.
3) It is very flexible and offers a wide variety of approaches to developing blocking plans.
4) It is microcomputer based allowing easy setup and use by interested railroads.

The ABM is designed to assist the planner, not replace him. While it does have the ability to automatically create and refine blocking plans, these plans are not optimal, and must be further reviewed and refined by the planner. The ABM includes capabilities to:

- evaluate an existing blocking plan
- automatically create a blocking plan
- iteratively improve a blocking plan
- allow interactive editing of a blocking plan
- complete a partial blocking plan

This paper will explore 1) previous attempts at developing blocking plan software, 2) the design and implementation of the ABM, 3) the results of the railroad use of the ABM, and 4) expected future development work.

II. SPECIFIC DESIGN OPTIONS

A literature review reveals that a number of approaches are possible in designing computer assisted blocking plan software including:

1. Automated Plan Evaluation & Costing
2. Mathematical Optimization
3. Single Pass Heuristics
4. Iterative Heuristics
5. Interactive Plan Development

Each of these approaches is explored in detail below.

A. Automated Plan Evaluation & Costing

An automated blocking plan evaluation and costing program is the simplest of all the possible approaches. In this approach the user supplies a partial or complete blocking plan, a network, and a traffic base. The software assigns traffic to the blocks, and computes a number of costs and statistics that can be used to evaluate the plan including the car volumes for each yard, link, and block. In addition, derived statistics such as switch-engine-hours and gross-ton-miles may be generated. If the blocking plan is incomplete then a list of cars not moved can also be produced. If constraints are provided then an exception report can be produced whenever a constraint is violated.

This approach can be used as part of an iterative blocking plan development approach in which the analyst would develop a blocking plan, evaluate it, and refine it repeatedly until it met his or her needs. The computer would automate the most tedious aspects of blocking plan evaluation and would provide general indications of where the plan could be improved, but would not assist directly in creating or revising a blocking plan. One example of this approach is the blocking model used by the Norfolk Southern [16]. The advantages of this approach are its simplicity, modest data requirements, low level of computational intensity, and ease with which the user can understand the system. In addition, the user will be thoroughly familiar with the end product of the process and should be comfortable with the result and be able to defend it to others. Obvious disadvantages are the amount of work that must be performed by the user, and the lack of any guarantee of optimality.

B. Mathematical Optimization

This is the most complex of all the possible approaches. In this approach a mathematical model that describes the problem is created. Generally the model consists of constraints and an objective function. The program's goal is to maximize or minimize the objective function while obeying the constraints.

The ideal formulation would be an objective function which converted a particular blocking plan into profit. The constraints would not only impose limits on such things as cars handled at each yard and minimum block size, but would also take into consideration less easily quantified constraints. In general it is very difficult to relate a blocking plan to profit. As a result the objective function is usually restated in terms of cost and the problem becomes one of cost minimization. This problem can be defined on paper in a way that takes into account most of the important considerations and if solved would yield a very good result. Unfortunately, the best theoretical formulations have very complex non-linear objective functions, and huge numbers of constraints, which are essentially unsolvable with today's computational methods.

Several attempts have been made at simpler formulations. In general, the blocking problem can be restated, without a significant loss in accuracy, as a multicommodity network transshipment problem. In this formulation the network's physical links are replaced with links representing all possible blocks. Traffic is then flowed over the network in such a way as to minimize total cost and obey a set of capacity constraints. The optimal blocking plan consists of all blocks (or links) with non-zero volumes. This approach allows the use of well established linear optimization techniques.

A fairly complete formulation of this problem was done by Bodin et al [6]. In this formulation the constraints 1) limit the maximum and minimum block size, 2) limit the maximum number of blocks that can be made at each yard, 3) force all cars leaving the same yard with the same destination to use the same block, and 4) limit the total number of cars that can be classified at each yard. The objective function attempts to take into consideration the cost of handling a car at each yard, the cost of placing a car into a particular block, and the delay costs associated with classifying of a car at a particular yard for a particular outbound block as a function of block volume. In a test case a 33 yard problem based on the Norfolk and Western Railroad was solved. This problem proved to be too large for the MPSX/370 IBM 370 based mathematical programming package they were using. To solve the problem they were forced to go through a complex process of eliminating many of the constraints, and fixing the values of many of the variables. The eliminated and frozen variables were then adjusted on a piecemeal basis to solve the problem.

The attempt by Bodin et al is the only known attempt to solve the full blown problem. At least two other formulations (Suzuki [20], and Kornhauser [11]) have been found that incorporate a multicommodity transshipment problem approach. However, both of these are highly restrictive cases and are essentially enhanced heuristics. These will be discussed in later in this paper.

The obvious advantage of the full optimization approach is the potential for high quality results. The formulations can even deal with such tricky considerations as backhauls, multiple routes, and differentiation among traffic types. However, the full problem is clearly too large for a microcomputer. More limited problems may be solvable on a microcomputer but only by restricting the problem greatly. As constraints are eliminated, and the objective function simplified, the resulting solution will start to diverge from the optimal one. As a result, it may be possible to more accurately and efficiently solve the problem using a heuristic. Other potential problems with this type of approach are the complexity of the algorithm, the high potential for improper use by the user, and the lack of familiarity and trust in the solution produced due to its "black-box" generation.

C. Single Pass Heuristics

A heuristic is an algorithm that attempts to find a good solution to a problem, but can not guarantee the end product will be the best possible solution (no guarantee of optimality). These types of approaches fall into two classes: 1) single pass and 2) iterative. The later approach is discussed in Section II.D.

A single pass heuristic attempts to create a blocking plan based on a set of rules designed to do a reasonable job. Once the blocking plan is complete no attempt is made to modify it further. This type of approach can be used by itself, or as a first step of a more complex approach. The literature contains many examples of single pass heuristics being used by themselves or as part of a larger solution (Suzuki [20], Kornhauser [11], and Siddiqee [19]).

In general, these approaches attempt to lay down a set of sensible rules and then build a blocking plan using these rules. The approaches tend to do two things: 1) divide all of the yards into several distinct types, and 2) use what is called the block early/ block late principle.

In the algorithm's simplest form all yards are divided into transit and non-transit yards. A non-transit yard can only receive traffic terminating at itself, and can only make blocks for nearby transit and non-transit yards. A transit yard can receive cars destined to any other point in the system, and can make blocks for any yard in the system. Further refinements of this approach define several types of non-transit and transit yards, and give each transit yard a "sphere of influence" consisting of a number of transit yards that are dependent on it.

The basic approach is to first create outbound blocks from each non-transit yard. Blocks are defined based on a minimum block size specification, and a maximum number of blocks that can be made at each non-transit yard. Whenever possible, direct blocks are made to nearby non-transit yards. Generally, the farthest point for which a non-transit yard can make a block is the first transit yard that is encountered in the logical direction of movement. Next each transit yard is blocked. Whenever possible, blocks are made directly to each car's destination (block early). If this is not possible then cars are put in blocks that go as close to their destination as possible (block late). Some iteration may be required to account for all the traffic.

The advantages of this type of approach are several:

- A systematic approach is used to create the blocking plan with overall "goodness" in mind.
- The approach is relatively simple and does not require an excessive amount of computer time.
- Constraints and cost objectives can be accounted for.

There are also a number of disadvantages:

- The plan is not guaranteed to be the best one: large improvements may be possible through additional refinements.
- It is a "black box" solution and the user may have problems understanding it and defending it to others.
- It may not deal with complex constraints such as multiple routes, backhauls, and traffic priorities.

To overcome some of these disadvantages the approach is often combined with an iterative heuristic. Such approaches are detailed in the next section.

D. Iterative Heuristics

An iterative heuristic attempts to improve a blocking plan using a series of incremental changes. Generally, these approaches start with a blocking plan supplied by the user, or generated using a single pass type heuristic such as that detailed in the previous section. Using a set of sensible decision rules a number of changes are made to the blocking plan and the traffic is reassigned. The revised blocking plan is then evaluated. If the direction of change was favorable then the changes are retained, otherwise the old plan is restored. This process is repeated until no changes further improvements can be made. The bulk of the attempts at computer based blocking plan development have used this type of approach (Duvalyan [8], Kornhauser [11], Koutoukova [12], Martens [15], and Suzuki [20]).

These approaches generally have three steps:

1) Identification of Possible Improvements
2) Reassignment of Cars to Blocks
3) Evaluation of Results of Changes

In the simplest of these approaches a "by-pass" decision rule is used. In this approach each yard is examined and the possibility of building a direct block for traffic currently going through an intermediate yard is examined. If the objective function improves with the addition of the "by-pass" block then the new block is added, otherwise it is not. This continues until no by-pass blocks can be built that will improve the objective function. These approaches often start with a blocking plan consisting of each yard building blocks only for its nearest neighbors. The by-pass decision rules often exclude consideration of "non-transit" yards (see Section II.C. for discussion of transit and non-transit yards).

This type of approach has most of the same advantages and disadvantages of the single pass heuristics. It is better than the single pass approach in that it tries to improve the plan as much as possible, within the confines of the decision rules. The basic problem with the incremental approach is that you are always examining incremental variations on your starting blocking plan, making the end result a function of the quality of your original plan, and making it difficult to find improvements resulting from simultaneous changes to many blocks. In addition, the backhaul, multiple route, and traffic priority issues remain unaddressed. Most of these problems are unresolvable.

An interesting approach implemented by Suzuki [20] and Kornhauser [11] is to imbed a restricted multicommodity transshipment problem within the heuristic. In this approach a number of changes are made to a blocking plan based on a set of decision rules. Instead of assigning the traffic to the new blocks based on past assignments, the traffic is assigned to the blocks using a multicommodity transshipment algorithm. Because the number of blocks is restricted and some of the variables fixed by the decision rules this problem is more tractable than the mathematical optimization approach detailed in Section II.B. This approach has the advantage of allowing multiple routes, multiple traffic priorities, and backhauls. It should be noted that because only a subset of the feasible blocks are being examined there continues to be no guarantee of optimality.

The above approach is a distinct improvement over simpler assignment approaches. However, it may be too computationally intensive, and data intensive, for a microcomputer. This may remain the case even if clever (and complex) methods are used to restrict the number of decision variables.

E. Interactive Plan Development

The process of developing, evaluating, and modifying a blocking plan by hand can be very tedious. The approach detailed in Section II.A. helps to streamline the evaluation process, but does nothing to help in plan development. The approach detailed here helps to speed the development and plan refinement process. The basic approach is to allow the user to interactively enter blocks into the computer, and have the computer recompute the block and yard volumes, constraint violations, and objective function, as each block is entered. The only known example of this approach is the CANAT model used by Canadian National. This approach can be used to create plans from scratch, complete partial plans, and modify existing plans.

In this approach the computer tracks the number of cars to be moved to each destination, and indicates what cars have not yet been assigned to a block at each yard. These cars can either be originating at that yard, or arriving on blocks from other yards. As each block is added, the affected cars are moved through the system based on the block definitions, and the volumes at all the affected yards are updated. If a block is eliminated, then the traffic is removed from the affected yards including downstream yards. This allows the user to interactively create a blocking plan, observing all of the constraints and objectives that a heurisitic would consider, plus those that are not quantifiable.

This approach has most of the same advantages and disadvantages as the evaluation method detailed in Section II.A., with the added benefit of partially automating the development and refinement process.

III. GENERAL CHARACTERISTICS OF THE ABM DESIGN

The ABM is designed to be a flexible tool for the development and refinement of railroad blocking plans using several of the methods described above. The ABM was designed with three functions in mind:

1) automated blocking plan development,
2) evaluation of blocking plans, and
3) interactive plan definition and refinement.

Due to the limits of time on this project, and the restrictions imposed on microcomputer based approaches, the use of multicommodity transshipment optimization routines was avoided. The ABM attempts to perform all three functions in a logical and easy to use manner. It also attempts to keep the data requirements manageable, while having a flexible structure of constraints and costs. There are a total of eight routines making up the ABM. The two key routines are called Auto-Block and Manu-Block. The Auto-Block routine is an iterative heuristic that attempts to improve a blocking plan by using a modification of the methods described in Section II.D. This iterative heuristic method relies on a shortest path algorithm to find the least cost path across a set of predefined blocks. The Manu-Block routine is an interactive blocking plan development routine of the type described in Section II.E. These two routines are the heart of the ABM, with all the other routines functioning in supporting roles. There are a total of six support routines: one control program, two preprocessors, two coprocessors, and one post-processor.

The control program, called the Automated Blocking Model Master Control Program (ABM) supervises the use of the other programs making up the ABM. It also maintains status information such as the names of data files, and the current state of the blocking plan.

The Blocking Model Data Conversion Routine (BDC) is a preprocessor that converts the ABM input files into formats easily used by the rest of the ABM, and does simple error checking. The Blocking Model Preprocessor (BMP) does further error checking, and restates the input data in a format that is easily and efficiently used by the rest of the ABM.

The Blocking Plan Development Routine (BPD) is a single pass heuristic coprocessor that will create a complete blocking plan for use by the Auto-Block, Manu-Block, and Blocking Plan Evaluation routines. It is based on the types of approaches described in Section II.C. The routine can either start from scratch, or with a partial blocking plan. The other co-processor is the Blocking Plan Evaluator (BPE), which produces a number of reports detailing a blocking plan. It functions in a manner similar to the approach described in Section II.A.

The Blocking Model Data Feed Program (BDF) is a post processor used to create an output blocking file for use by the ABM or SPM.

The ABM relies heavily on a shortest path routine to route both blocks and flows. All blocks will be routed on the lowest cost path from their origin to their destination. This routing will be a function of the block's traffic mix. Each flow will be "routed" across the available blocks using the lowest cost combination of blocks subject to capacity constraints.

Figure 1 presents a diagramatic representation of the ABM system.

IV. THE ABM DATA BASE AND USING THE ABM

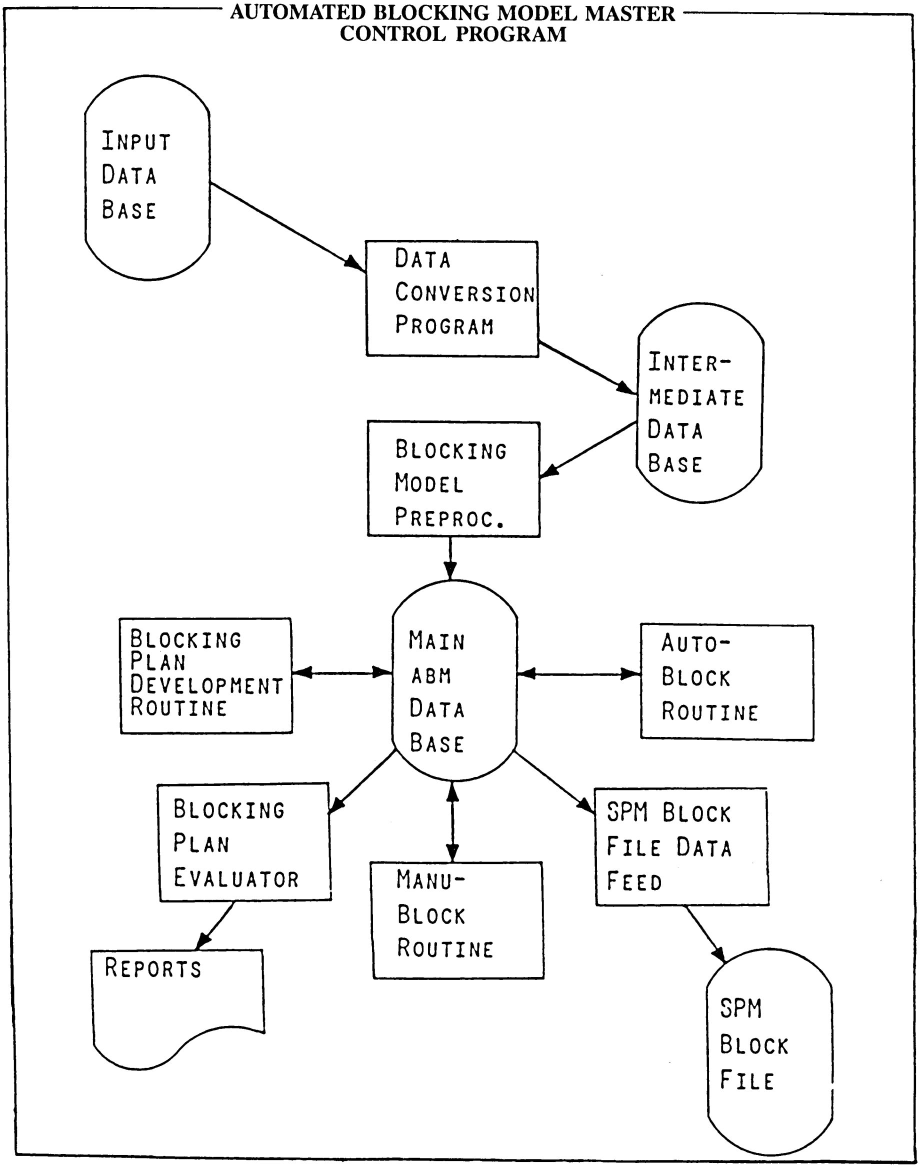
The key to the ABM is a central data base that can be accessed by six of the ABM programs. This central data base contains information on the current blocking plan including how cars are assigned, traffic loadings at each yard, and the associated costs. Each time one of the main ABM programs is run the central data base is updated. Thus the Blocking Model Preprocessor can be used to add a user supplied blocking plan to the central data base. This plan could then be completed using the Blocking Plan Development Routine and improved upon by the Auto-Block Routine. It can then be refined by the Manu-Block Routine, detailed by the Blocking Plan Evaluation Program, and sent to the Service Planning Model using the Blocking Plan Data Feed Routine.

The idea behind the ABM is to give the user a variety of tools for developing a blocking plan. By maintaining all of the plan related information in a central data base, each of these tools can be used in a flexible manner to suit the user's particular needs.

Given this flexibility and range of tools the ABM can be used in a variety of ways. These range from simply evaluating an existing blocking plan, to modifying a blocking plan, to completing or adding to a blocking plan, to developing a totally new plan. The user can let the software do all of the work, or can make every change by hand. How the ABM is to be used is largely up to the individual, and what suits his or her style and needs.

One should keep in mind in any ABM application that the processing time and problem complexity is directly related to the problem size. One should keep the problem as small as possible. While the ABM can in theory handle problems of up to 2000 blocks, thousands of flows, dozens of traffic priorities, and 250 yards, problems of this size will take hours (if not days) to process on a microcomputer. It is probably best to limit problems to no more that 1000 flows, 60-80 yards, a few hundred blocks, and less than 6 traffic priorities.

Figure 1



Diagramatic Representation of the ABM System

V. CURRENT STATUS AND FUTURE PLANS

This paper has reviewed the basic approaches that can be used to computerize the development of blocking plans, what approach the ABM takes, and in general terms how one can apply the ABM. The ABM is currently being tested at the Burlington Northern Railroad and the Grand Trunk Western Railroad. While the tests are not complete, the results appear to be basically favorable. The software's blocking plan generation capabilities are producing reasonable solutions. The interactive editing capabilities are being well received. The major significant drawback is long processing times for large problems, which may cause a parallel mainframe version to be developed. It is expected that the ABM will be used at several railroads in the coming year, and that a companion train scheduling model will be developed.

BIBLIOGRAPHY

1. Achermann, J., "Model and Calculation Programme Relating to the Optimum Formation of Trunk Haul Freight Trains," *Monthly Bulletin of the International Railway Congress Association, Cybernetics and Electronics on the Railways*, vol. 6, no. 5, May 1969, pp. 181-226.
2. Assad, A.A., "Analytical Models in Rail Transportation: An Annotated Bibliography," *INFOR*, vol. 19, no. 1, February 1981, pp. 59-80.
3. Assad, A.A., "Modelling of Rail Networks: Toward a Routing/Makeup Model," *Transportation Research*, vol. 14B, 1980, pp. 101-114.
4. Assad, A.A., "Models for Rail Transportation," *Transportation Research*, vol. 14A, 1980, pp. 205-220.
5. Assad, A.A., *Supply Modelling of Rail Networks: Toward a Routing/Makeup Model*, Operations Research Center Working Paper, Massachusetts Institute of Technology, December 1977.
6. Bodin, L.D., et al, "A Model for the Blocking of Trains," *Transportation Research*, vol. 14B, 1980, pp. 115-120.
7. Charnes, A. and M.H. Miller, "A Model for the Optimal Programming of Railway Freight Train Movements," *Management Science*, vol. 3, no. 1, October 1956, pp. 74-92.
8. Duvalyan, S.V., "Working Out Optimal Freight Make-up Plan on Electronic Computer and Distribution of Classification Work on Large Railway Network," *Rail International*, vol. 4, no. 10, October 1973, pp. 1069-1071.
9. Folk, J.F., "A Brief Review of Various Network Models," *Studies in Railroad Operations and Economics*, vol. 7, Center for Transportation Studies, Massachusetts Institute of Technology, June 1972.
10. Holt, J., "The Planning of Train Services Using Computer Methods," *Rail International*, vol. 6, no. 3, March 1975, pp. 177-187.
11. Kornhauser, A.L. and P. Mayewski, *A Heuristic Approach to the Generation of Network-Oriented Blocking Patterns for Railroad Operations*, Princeton University, undated, 21 pages.
12. Koutoukova, G.A. and L.K. Markov, "Calculation of Train Formation Diagrams by Electronic Computers," *Proceedings of the Symposium on the Use of Cybernetics on the Railways*, The International Railway Congress Association, 1963, pp. 166-173.
13. Levin, A., "Scheduling and Fleet Routing Models for Transportation Systems," *Transportation Science*, vol. 5, no. 3, August 1971, pp. 232-255.
14. Magnanti, T.L. and R.T. Wong, "Network Design and Transportation Planning: Models and Algorithms," *Transportation Science*, vol. 18, no. 1, February 1984, pp. 1-55.
15. Martens, A., "Mathematical Models for the Problem of Routing Wagons Between Marshalling Yards," *Monthly Bulletin of the International Railway Congress Association, Cybernetics and Electronics on the Railways*, vol. 4, no. 4, April 1967, pp. 145-155.
16. Norfolk Southern Corporation, *Blocking Model User's Guide*, 17 pages.
17. Sato, A. et al, "A Simulation Model of Freight Transport System," *Railway Technical Research Institute Quarterly Report*, vol. 16, no. 4, December 1975, pp. 173-176.
18. Siddiqee, W. and D.A. D'Esopo, "Computer Methods in Blocking and Train Operations Strategies," *Transportation Research Record No. 656*, Transportation Research Board, 1977, pp. 45-52.
19. SRI International, "Blocking Strategy Analysis and Development Process," Report Extract: SRI/USRA Report on Computer Models Used in Conrail Formation Studies, Source and Title Unknown, SRI International, pp. 61-64.
20. Suzuki, S., "A Method of Planning Yard Pass Trains on a General Network," *Operations Research, 1972*, (M. Ross, ed.), North Holland Publishing Co., 1973, pp. 353-361.
21. Thomet, M.A., "A User-Oriented Freight Railroad Operating Policy," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-1, no. 4, October 1971, pp. 349-356.
22. Truskolaski, A., "Applications of Digital Computer to the Optimization of Freight Train Formation Plans," *Monthly Bulletin of the International Railway Congress Association, Cybernetics and Electronics on the Railways*, vol. 5, no. 5, May 1968, pp. 211-226.
23. Truskolaski, A., "Comments on the use of Optimization Techniques in Planning Freight Transportation Operations," *Rail International*, vol. 4, no. 11/12, Nov.-Dec. 1973, pp. 1219-1223.

ENDNOTE

* Project Manager, ALK Associates, Inc.