Modeling and Optimization of Mobility Analysis: Optimal Requirement Studies[1]

Fan Yang,
Ervin Y. Rodin,
S. Massoud Amin

Center for Optimization and Semantic Control[2]
Department of Systems Science and Mathematics
Washington University

ABSTRACT

In this report, we describe our work in developing models, methodologies and simulations for network optimization problems in the planning, analyzing and optimizing of large scale (air) transportation networks with time window constrained routing and scheduling. Our research is motivated by certain problems encountered in the United States military's strategic mobility analysis, in general, and specifically in Mobility Analysis Support System (MASS) of the USAF's Air Mobility Command (AMC).

This work is performed within the framework of Semantic Control paradigm, a three-layer supervisory hierarchical structure. In this context a new mathematical programming model, called Network Optimization Mobility Analysis (NETO), for the mobility analysis system is formulated as a pickup-delivery vehicle routing and scheduling problem with time-window constraints (PDPTW). In order to cope with the computational complexity inherent in the PDPTW formulation, we have developed and implemented a novel algorithm called SP-CGCE (set-partitioning formulation, column generation and column elimination). The computational results indicate a promising and robust performance by this solution algorithm. The problems tested/solved here involve many more nodes than similar problems previously attempted. The test results indicate that the SP-CGCE algorithm is at least twice as fast as currently available column generation-branch and bound schemes; this increase in speed is due to the effectiveness of the column elimination process used after the completion of the linear programming phase to obtain integer solutions.

In particular, the focus of this report is the optimal *requirement studies problem*, where the following question is addressed: *"How many of what types of transportation assets are necessary to move cargo to the specified destinations, satisfying a particular desired closure schedule?"*

1. INTRODUCTION

The Center for Optimization and Semantic Control at Washington University in St. Louis has been conducting research jointly with the Air Mobility Command of the United States Air Force with respect to the Mobility Analysis Support System. In the past, we have solved several large-scale, time-dependent, mixed variable, uncertain and complex problems encountered in aerospace and decision support domains [1-5,8,9] using the Semantic Control paradigm (see below). The Center researchers approach the solution of such problems using a judicious combination of classical mathematical methodologies (mathematical programming, computational geometry, control theory, game theory, stochastic, etc.), together with Artificial Intelligence paradigms such as Planning, Search, Fuzzy System Theory, Neural Networks, Rule Based Systems, and Logic Programming [8-9]. Our approach is based on the Semantic Control paradigm—a three-level hierarchical structure (Figure 1-1) consisting of:

- an Identifier, which processes the list of requirements, known as the time-phased force deployment data/document (TPFDDs[3]), and interprets the available information;
- a Goal Selector, which generates and evaluates several plans; and
- an Adapter, which implements the optimal plan.

For example, the Identifier module consists of neural networks for processing, pattern recognition and optimization of TPFDDs. Once trained, neural networks identify requirements and consequently recommends assignment and allocation of aircraft in order to deliver those requirements. Currently, given a requirement containing:

i) a commodity code (such as outsize, oversize, bulk, passengers),
ii) an onload-offload region, and
iii) the percent of the total requirement to be moved,

the neural network recommends the appropriate assignment and allocation of aircraft to deliver that requirement.[4] The neural network module serves as a "pattern recognizer" in order to reduce the complexity; in addition to this, we are currently developing a "fuzzy model" of the air transportation which incorporates leeways in the constraints and goal. This should prove very useful since several quantities such as MOG (maximum on

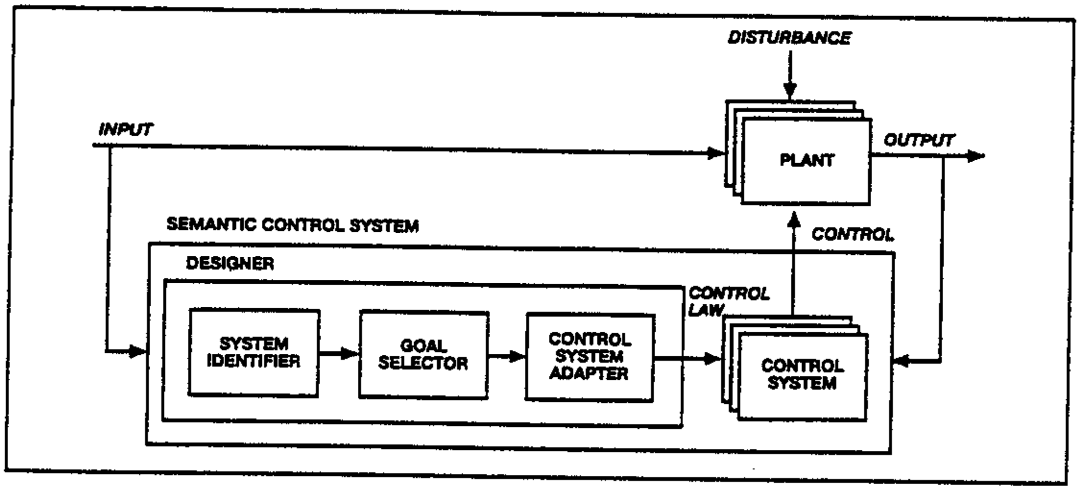

Figure 1-1: A semantic control system consists of a System Identifier, a Goal Selector, a Control System Adapter, and one or more control systems/laws.

ground) are not crisp variables. This approach admits such uncertainties as part of the model, thus reducing labor-intensive post-optimality sensitivity analysis. These issues will not be discussed further in this report. We refer the interested reader to [8,9]. This report deals mainly with algorithm development and simulation of exact mathematical programming and optimization methodologies (cf. [6] and [10]) for the Goal Selector module of the Semantic Controller. More specifically our objectives in this report are:

1) reviewing existing mobility analysis models and addressing their various limitations,
2) presenting the new mobility analysis model NETO formulated as a PDPTW problem,
3) discussing our solution algorithm (SP-CGCE) and comparing its performance to other published results,
4) providing a brief overview of the system implementation and related issues,
5) giving an example of the optimal requirements studies problem, and
6) concluding with a discussion of other relevant problems addressed by this approach as well as related open problems.

This report is divided into six sections:

- Section 1 and subsections 1.1 through 1.3 present background information on the strategic mobility analysis and limitations of current simulation and mathematical models.
- Section 2 discusses the system architecture and the components of our model NETO.
- Section 3 focuses on the mathematical formulation, algorithmic details, and performance analysis. The mathematical model for our formulation is given in more detail in Appendix B.
- Section 4 presents system implementation and gives an example of the optimal requirements studies problem.
- Section 5 discusses other related problems and defines future work.
- Section 6 concludes with a brief summary.

1.1. STRATEGIC MOBILITY ANALYSIS: BACKGROUND [7]

Various objectives of strategic mobility analysis are grouped into three broad planning categories:

- Resource Planning: long-range deployment planning and programming.

- Deliberate Planning: mid-range deployment planning that encompasses the development and analysis of operational plans.
- Execution Planning: including both the short-range crisis action planning before an engagement begins and the continuing planning and replanning as execution proceeds.

There are two fundamental questions involved in the above planning categories as well as in all other planning activities:

1) How to accomplish the objectives (and to what degree) given the resources?
2) Given the objectives, what are the minimum resources required to accomplish them and how to do so?

In particular *Resource Planning* encompasses program development and related policy research that is conducted in the planning, programming, and budgeting system (PPBS). Although mobility studies in resource planning may presume specific theater scenarios, the analyses are collectively meant to conduct coordinated long-range resource planning for total forces. These studies are generally of two types: capability assessments, which determine the force closure that can be supported by a given set of lift assets, and requirements studies, which estimate the lift assets necessary to support a given force closure.

In *capability assessments* (the forward problem), a strategic mobility model is used to assess how soon a particular set of transportation assets can effect theater closure of a particular set of forces, support resource, and resupply, given the constraints of scenario and cargo priorities. Although capability assessments theoretically are one-shot uses of the model, more runs are almost always needed to assess the implications of uncertainty. To explore degrees of risk with a given force structure and operational objectives, the model may be exercised numerous times with different versions of scenario assumptions.

In *requirements studies* (the backward problem) the following question is asked, "How many of what types of transportation assets are necessary to move cargo to the specified destinations, satisfying a particular desired closure schedule?" The results of the analysis describe a set of transportation assets , or perhaps the required increments to a baseline set of assets. Conducting this type of study with the currently available mobility models is necessarily a tedious iterative process. At the Joint Staff, an important recent example of a requirements study is the RIMS (Revised Intertheater Mobility Studies), which required over 400 MIDAS runs (Model for Intertheater Deployment by Air and Sea) between October 1986 and April 1989.

1.2. ANALYSIS PROCESS OF THE CURRENT MOBILITY MODELS

The models that are currently being used in the defense communities [7], such as MIDAS (Model for Intertheater Deployment by Air and Sea, a Joint Deployment System model, 1980), RAPIDSIM (Rapid Intertheather Deployment Simulator, 1974), TFE (Transportation Feasibility Estimator, a Joint Operation Planning System), FLOGEN (Flow Generator, an Air Mobility Command model), SEACOP (Strategic Sealift Contingency Planning System, a Military Sealift Command model), MASS (Mobility Analysis Support System, an Air Mobility Command model, 1980's), etc., all process data in a similar way. Each model uses several inputs in the form of data files; all use similar algorithms to simulate the transportation system, and all produce similar outputs, e.g. delivery dates, utilization rates, and delays/queues.

Typically, four files provide input for the simulation models: a requirement file, a PREPO (prepositioning) file, a transportation resources file, and a scenario file. The model then assigns cargoes to transportation assets according to certain rules, and simulates cargo movement through the transportation system. All of the current models use the same solu-

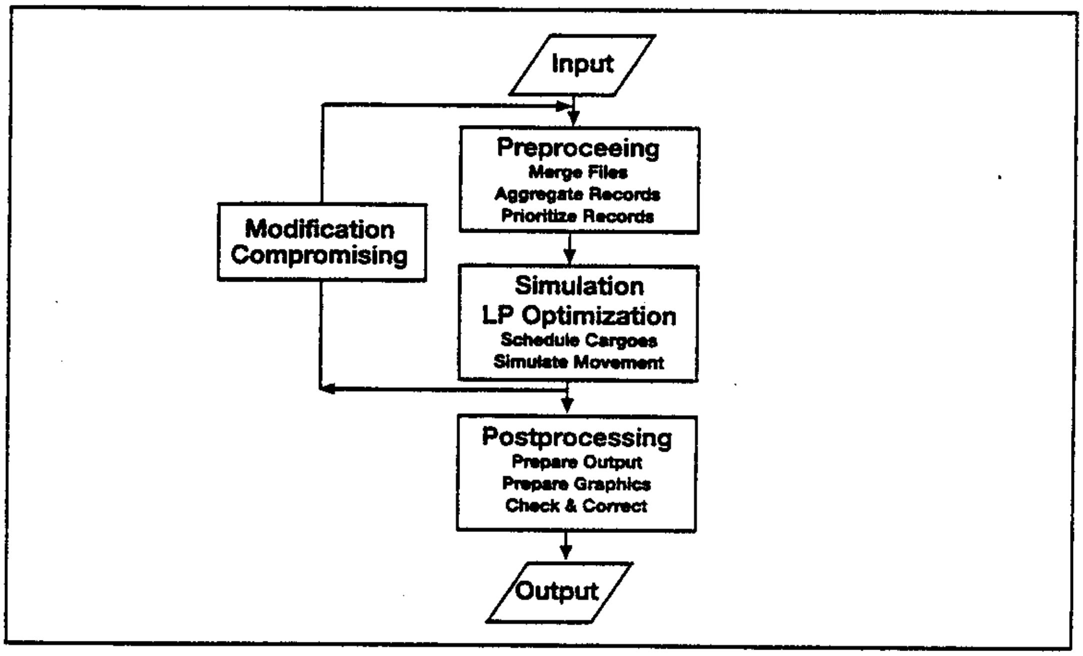

Figure 1-2. The Analysis Process of the Current Mobility Models

tion technique (deterministic simulation) and basically follow the same steps to arrive at a delivery profile. A model may undergo some or all of the following analysis tasks (Figure 1-2): merge files; aggregate records (by ports, route, ships, or cargo); prioritize records; select models (i.e., air or sea for those with no chosen mode); schedule cargoes; simulate movement; prepare textual output; prepare graphical output; check and correct.

As mentioned earlier, we were motivated by problems encountered in the Mobility Analysis Support System (MASS). MASS is a family of analytical tools developed originally by the Command Analysis Group at the headquarters, Military Airlift Command, Scott AFB, Illinois, from the mid-1980's to the early 1990's. The Command Analysis Group (Studies and Analysis Flight now) is presently working under the Plans and Analysis Directorate at the headquarters, Air Mobility Command. The MASS family consists of a variety of models to aid in the analyses of the full spectrum of airlift operations from daily peacetime cargo movement to full-scale global wartime movements such as Desert Shield/Storm.

MASS is a deterministic simulation model which directs aircraft through a network of onload, enroute, offload, and recovery bases in order to deliver a set of requirements needed to achieve some predefined scenario goal. MASS is capable of handling many diverse scenarios. An enroute base is an intermediate stop, normally for fuel or to change crews, between an offload and onload base. A recovery base is visited after the offload for fuel and/or crew change, in order to relieve congestion at the offload base. The recovery base is where aircraft wait to be scheduled for their next mission.

The cargo requirements (TPFDD) contain cargo information or requirements such as onload(origin), offload(destination), available date, required delivery date, size, weight and nature of the cargo, etc. The set of cargo requirements given by TPFDD are taken as input, based on the availability of aircraft, airfields, parking space, crew members and routes etc., MASS works through the entire airlifting operation, simulating onloading, offloading, scheduling, routing, refueling, crew changing processes, generating a multitude of step by step aircraft activities, cargo movement and delivery information. MASS also simulates the impacts made by some anticipated/unanticipated changes in the airlifting system, such as increased number of aircraft, base closure, etc. It is an effective tool in that it offers a feasible solution to the airlift problem; however, it does not address the backward problem directly nor does it guarantee an optimal solution. In subsection 1.3 we describe the limitations of current models in more detail.

1.3. LIMITATIONS OF CURRENT MODELS

Schank et al. ([7], pp. 39-50) from the RAND corporation provide a very comprehensive review of strategic mobility models and analyses. The following limitations of the current models are identified:

- All work in only one direction, accepting similar types of input data and producing the same general information.
- None are optimal.

Current Models All work in One Direction: All major current mobility models are simulations or have major simulation components. They accept data on what has to be moved (cargoes), what is prepositioned (PREPO), what transportation assets are available, and what the assumptions are regarding timing and available infrastructure. They then assign cargoes to transportation assets according to specific rules and simulate their movement through the transportation system. Finally, all produce estimates of when units are delivered into the theater and utilization rates of the transportation assets and facilities. All existing models use this process, regardless of the decisions and objectives being addressed. All models basically provide the closure profile for these forces, support units and resupply given these transportation assets (forward problem).

This question may be appropriate for deliberate planning or execution planning analysis, but it does not directly address the concerns of how many transportation assets are required (important for resource requirement studies).

Strategic mobility analysis that addresses transportation asset requirements seeks the best mix of transportation assets for achieving a desired closure profile for a given set of forces, support units and resupply (backward problem). The unknown values in requirements determination are required inputs to existing models.

At present, therefore, analysis cannot directly answer the question of how many of each type of transportation assets are required. They can only obtain an approximate solution by multiple runs and trial and error.

The Solution Is Not Optimal: This laborious process, more art than science, certainly does not provide "optimal" answers. In fact, a good deal of expertise is typically needed to develop even a "good" answer to transportation force structure issues. This suggests that a different modeling approach is warranted, one that moves away from simulations, or at least from current simulation methods, to an approach that directly addresses force requirements questions.

Our mobility analysis system, NETO, accepts the same input file information as the above models; however, it differs from them and other newer approaches (such as ADANS) in its optimization and analysis capabilities. For example, all other models are geared toward addressing the capability assessment (the forward problem) while unable to solve the optimal requirements studies (the backward problem); NETO is capable of solving both forward and backward problems. In the remainder of this report we focus precisely on the commonly ignored optimal requirement studies problem. In what follows, we discuss our approach to address the above limitations and to solve the optimal requirement studies problem without the need for repeated runs.

2. NETWORK OPTIMIZATION MOBILITY ANALYSIS SYSTEM

NETO consists of two interrelated components: a network optimization engine with time window-constrained routing and scheduling based on integer and combinatorial optimization methodology; and an analysis system with a information management system built upon RDBMS and multimedia technology. RDBMS is not presented here; it falls outside of the focus of this report. In this section, we will give an overview of the NETO system architecture, describe the underlying labeled digraph and PDPTW. More detailed issues as well as the forward problem, selection of cutting plane, column generation, column elimination, numerical results and comparisons are reported in [6].

NETO SYSTEM ARCHITECTURE

The diagram and system hierarchy of NETO system architecture are shown in Figure 2-1. The input information is the same as the original input information (cf. Figure 1-1), except that it might come from a database system. We will substitute the original simulation process with our optimization system.

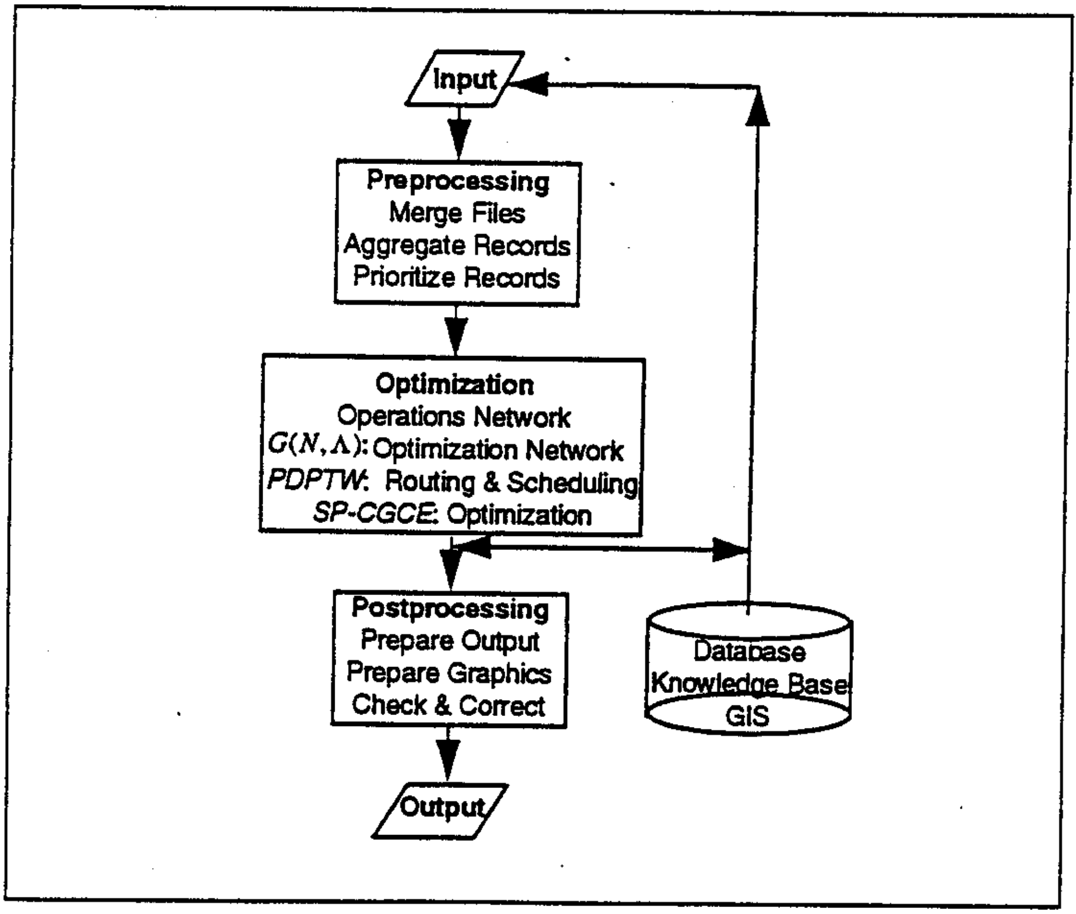

Figure 2-1. NETO System Architecture

NETO SYSTEM COMPONENTS

The functions of the various components are described as follows:

Operations Network: The Operations Network is the original mobility operation information represented in the form of transportation network consisting of all relevant data, such as air bases, seaports, air routes, sea routes, onloads, offloads, enroutes, cargoes, transportation vehicles, weather, scenario, movement requirements, logistics factors, and so forth. The operations network is more than just a geographic network such as a map; it is a model of the concept of operations, a database of the operation information.

Optimization Network $G(N,\Lambda)$: The Optimization Network is the Operations Network represented in the form of a *labeled digraph* suitable for mathematical optimization purposes. It has four types of nodes: starting nodes S, terminating nodes T, pickup nodes $P^+$and delivery nodes P^-. P^+ and P^- forms a *complete digraph* $P^+ \times P^-$. For S arcs only go from S to P^+. For T arcs only go from P^- to T. Denote $P=P^+ \cup P^-$, $N=S \cup P \cup T$, and $\Lambda = S x P^+ \cup P^+ x P^- \cup P^- x T$. Then we can write the digraph as $G(N,\Lambda)$.

In the mobility analysis system, the set of nodes S could be the home depots. The set of nodes T could correspond to the recovery bases. P^+ depicts the onload bases of requirements and P^- describes their offload bases. A node in the optimization network may correspond to many physical nodes in the operations network or vice versa; an arc in the optimization network may correspond to several arcs/paths in the operations network and vice versa. The labels contain various relevant information derived from the operations network. Among these data are the cost of arc (i,j), time window constraints $[a_i,b_i]$,$[a_j,b_j]$ (time intervals during which service is required; "service" meaning either pickup or delivery), the physical nodes that make up the arc (i,j), etc. Some of the variables used in the optimization network are:

$\overline{d}_i$:	load vector(volume, weight...) of cargo i at node i
$[a_i,b_i]$:	pickup time window at node i for movement/cargo i
$[a_0,b_0]$:	time window for vehicle leaving the depot S
$[a_{2n+1},b_{2n+1}]$:	time window for vehicle returning to the depot T
$\overline{D}$:	capacity of vehicle (load weight limit, volume,...)
t_{ij}:	travel time from node $i \in N$ to node $j \in N$
s_i:	service time(pickup time or delivery time) at node $i \in N$
$\overline{Y}_i$:	the total load on the vehicle just after it leaves node $i \in N$
T_i:	time of start service at node $i \in N$
T_0:	arrival time at node i or time vehicle leaves the depot S
T_{2n+1}:	time vehicle returns to the depot T
RT:	feasible route defining formulation

A complete list of terminology, definitions, notation, and symbols is given in Appendix A at the end of this report. If the labels which store the transformed mobility information are oriented, all mobility analysis will result in the same kind of optimization network. Therefore if labels are not considered, the optimization network $G(N,\Lambda)$ is a topological representation of the mobility system. An operations network is converted to an optimization network through *Network Construction.*

Network Construction: This transforms an operations network into an optimization network, taking into consideration such factors as routes, enroutes, cargoes, fuel, and other information in association with the operations network. There are basically two tasks: building up the digraph topology G and computing the labels. For example, to construct the arc from a pickup node $i \in P^+$ to a delivery node $n+i \in P^-$, we may select the shortest path P with the maximum length of any segment in P not exceeding a certain quantity in the operations network. This could mean that a certain type of aircraft can make a sustained flight with supported available refueling along the route.

Reduced Optimization Network: The Reduced Optimization Network is an Optimization Network reconfigured by tightening some excessively wide time windows and by eliminating as many as possible inadmissible arcs. By excessively wide time windows, we mean those time windows that can be narrowed without changing the problem under consideration. By an inadmissible arc we mean arcs which violate the constraints imposed upon the mobility system, such as time window and vehicle capacity, etc. PDPTW [16,17,20,21] is the underlying model for the optimization. The PDPTW model represents a vehicle routing and scheduling problem where cargoes are to be picked up in specified origins (sources) within given pickup time periods and to be delivered to desired destinations (sinks) within given delivery time periods.

The PDPTW was first formulated based on a vehicle flow/multicommodity flow-based nonlinear model (cf. [6]) and then reformulated into set-partitioning formulation and solved by the Column Generation Column Elimination Algorithm (SP-CGCE). The Column Generation Technique is based on the primal-simplex method to efficiently solve LP problems with a very large number of columns. It decomposes the original LP program into a master problem and a subproblem. In our research we have decomposed the linear relaxation of the set partitioning formulation of the PDPTW into a shortest-path subproblem with constraints. After the LP optimal is obtained by the column generation process, the Column Elimination Technique has been employed to obtain integer optimality.

The optimization result has been utilized for various output analysis purposes, according to the specific needs of the operation. In particular, the output data have been stored in the database system for further analysis.

3. SP-CGCE SOLUTION ALGORITHM AND PERFORMANCE

The use of a set-partitioning formulation, with the column generation scheme for solving vehicle routing problem (VRP) and PDPTW problems has recently become more frequent [11-21]. This is mainly because good alternative formulations for PDPTW problems are not known and the linear programming relaxation of the set partitioning formulation often yields a strong bound.

Other algorithms for solving PDPTW problems found in the literature [17,20] generally take the following *set-partitioning formulation, column generation, branch-and-bound* approach: these algorithms use a set-partitioning formulation and solve the relaxed set-partitioning problem by column generation, where columns are generated when necessary by solving a constrained Shortest Path Problem. Often the linear optimal solution is also an integer solution. If it is not, the linear optimal solution offers a good lower bound for the original set-partitioning problem, especially if some heuristic cutting planes are used. Then this scheme resorts to branch-and-bound to find the integer optimal. Since the original integer formulation is a set-partitioning formulation, branch-and-bound can not take place on the decision variables directly, but rather, on the arcs/paths in the network. This creates a tremendous number of subproblems/new nodes in the branch-and-bound process and each of them corresponds to a subgraph of the original graph $G(N,A)$. Again, column generation with the shortest path problem can be used to solve the subproblem on the subgraph to linear optimality in order to get the lower bound for a further branch-and-bound process.

In set-partitioning literature, the concept of identifying columns that would not contribute to the optimal solution and thus be excluded from the optimizing process was first mentioned in 1963 by Balinski [22]. Agarwal [23] applied this in 1989 for a VRP problem based on a well known result in combinatorial optimization by Pierce in 1973, [24]; we call this general concept column elimination.

In this work, the column generation technique is used to solve the linear relaxation of the SP to its linear optimal. The generating algorithm of the column generation is a constrained shortest path problem which is solved by dynamic programming. Based on the information of the reduced costs of the SP linear relaxation and its linear optimal value and an integer optimal upper bound, a column elimination technique is developed to eliminate many non-promising columns, thus reducing the size of the SP. The reduced SP then can be solved directly. By combining column generation with column elimination, we developed a solution algorithm for NETO; furthermore, it is mathematically guaranteed that the reduced SP will yield the integer optimal solution for the original problem (cf. section 3 of chapter 4 in [6]). The set-partitioning formulation of NETO is provided in Appendix B of this report. In subsection 3.1 we present a performance comparison of our SP-CGCE algorithm and previously published results.

COMPARISON OF PERFORMANCE BETWEEN SP-CGCE AND OTHER ALGORITHMS

The computational experiment was conducted on 99 different test problems; the problem size varied from ten pickup nodes to 120 pickup nodes, the number of feasible arcs ranged from 180 to 23,198, the feasible routes range from 16 to 32,375. Numerical results [6] indicate robust performance of the algorithm, especially the column elimination technique which generally reduces the SP problem size by an order of 2. The test results indicate an at least 100% speed increase over currently available column generation, branch-and-bound scheme; this is due to the effectiveness of the column elimination process. Additionally, in return for the sacrifice of some optimality , larger and more difficult problems can be solved several times faster. The gap between the LP bound and the integer optimum for the 99 problems tested range from 0% to 3.7% with an average of 0.1%.

As discussed earlier, most other algorithms for solving the PDPTW problem solve the LP to optimality and then utilize a branch-and-bound scheme to find the integer optimum. Since some of the subproblems on the subgraph can be almost as difficult as the original graph, solving one such subproblem might as well double the total solution time, and solving two might triple the time. This is evident in the numerical results given by Dumas[17] as recompiled here in Table 3-1.

Table 3-1. Time Required to Solve LP and ILP to Optimality [17]

Problem	A19	A30	B30	C20	C30	D40	D50	D55
Z(LP) cpu time(sec)	92	47	112	28	111	66	95	204
Z(ILP) cpu time(sec)	95	51	114	51	169	172	215	313

In the above table, Z(LP) cpu time is the time required to solve LP relaxation and to obtain the LP optimal solution Z(LP) via column generation. Z(ILP) cpu time is the time required to solve the Integer Linear Program optimal solution Z(ILP) by branch-and-bound using the LP optimal as a lower bound. The average ratio of Z(ILP) cpu time over Z(LP) cpu time is 1.56.

The *SP-CGCE* algorithm developed for NETO, however, does not use branch-and-bound to solve the problem to integer optimality after the LP optimal is obtained. It uses column elimination. From the computational results presented previously in this report and in [6], some of which are recompiled here in Table 3-2, we can see that the column elimination (*TCE*) is a fraction of the time required to solve the LP optimal (*TRts+TCG*). The only overhead involved is *Tzu*, the time required to find an upper bound of the integer problem (*ASP*), which is also a fraction of the time needed to solve the LP optimal.

Table 3-2. Time Required to Solve LP and ILP Optimal(SP-CGCE)

Problem	A36	B45	D60	D70	E40	E45
Z(LP) cpu time(sec)	257	1420	43	182	80	404
Z(ILP) cpu time(sec)	0	1	2	1	6	3

Z(LP) cpu time= *TRts+TCG;* (ILP) cpu time=TZu+TCE

In conclusion, the numerical experiments show that the column generation/column elimination algorithm is indeed a powerful, flexible, stable and efficient one. The column elimination procedure, in particular, is remarkably efficient. For more details on performance evaluation and theoretical issues such as network reduction (time-window tightening, arc elimination) and SP-CGCE algorithm development as well as an up-to-date literature review of PDPTW and related subjects, we refer the reader to [6].

4. NETO SYSTEM IMPLEMENTATION AND DEMONSTRATION

The first part of this section discusses the implementation of NETO, and the second part provides an example for solution of the backward problem.

4.1 SYSTEM IMPLEMENTATION

To test the SP-CGCE algorithm and demonstrate the new model *NETO*, a prototype system is implemented on a SunSparc Server 670MP workstation in a total 8181 lines of source code in C. The Linear Programming and Integer Programming solver for the *SP-CGCE* algorithm is built upon Cplex 3.0 Callable Library.• The user interface is implemented using Xt Tool Kit Intrinsics and Xlib. Technically speaking, the GUI interface class hierarchy based on object oriented programming is as in Figure 4.1.

In Figure 4.1 the "inputButton" controls the user input interface; the "outputButton" takes care of the output of optimization results and statistics; the "generateButton" generates a test problem; the "optimizeButton" activates the column generation column elimination program to solve the problem; the mapBox Widget Class allows a programmer to draw geographic maps in an X window. It is designed to give the application programmer the ability to work entirely in world (latitude, longitude) coordinates and frees him/her from thinking about the projection, scale, and display of the data. It has a 'zoom box' built in. The user can drag out a zoom box with the first mouse button (changeable through added translations). He/she can then zoom by clicking the first button within the zoom box, or cancel it by clicking outside its boundaries.

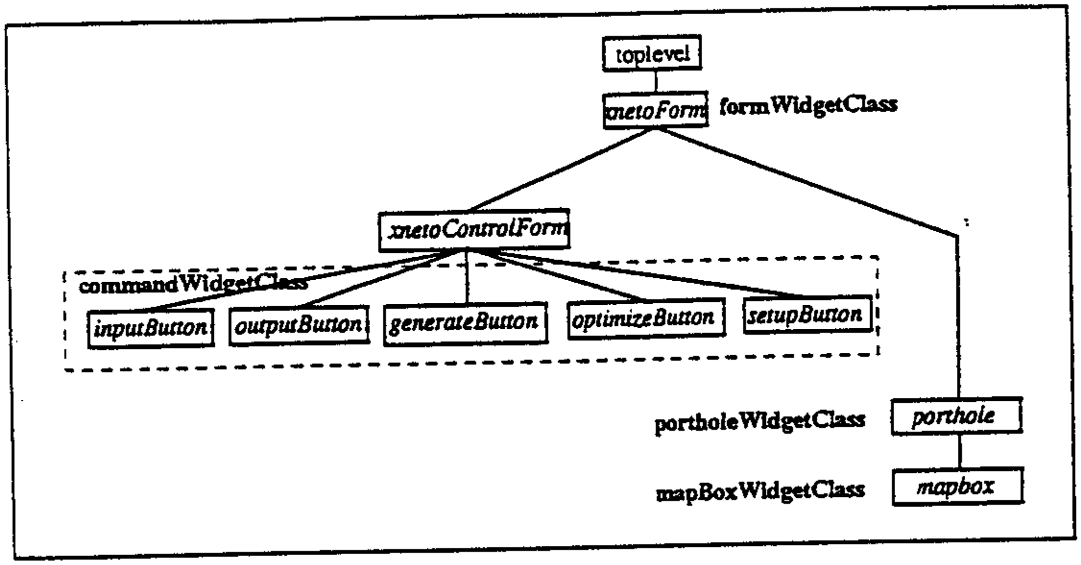

Figure 4.1 X GUI Interface Class Hierarchy

4.2 SYSTEM DEMONSTRATION

An example of the optimal requirement studies (the backward problem) is given in this sub-section. Five additional examples are given in [6] as demonstrations of the NETO and the SP-CGCE algorithm. In [6], the first example is a backward requirement analysis type problem; the second is a forward capability analysis-type problem; the third is a full-load problem; the fourth is a multidepot problem; and the last is a split type of problem.

4.2.1 EXAMPLE: OPTIMAL REQUIREMENTS STUDIES (BACKWARD PROBLEM)

In this example the problem is to find how many aircraft are necessary to move cargoes to the specified destination, while satisfying the closure schedule specified by the TPFDD. An illustration is provided of how the system and the algorithm function.

4.2.1.1 INPUT AND PREPROCESSING

Raw input information stored in the database system will first be preprocessed by the tasks listed previously in this report, such as by merging files and aggregating records into a correct and efficient form. Here we start from a regular and simplified TPFDD format and proceed to the optimization process described below.

4.2.1.2 OPTIMIZATION

Operations Network: The operations network constitutes the original mobility operation information represented in a form of transportation network consisting of all relevant data, such as air bases, seaports, air routes, sea routes, onloads, offloads, enroutes, cargoes, transportation vehicles, weather, scenarios, movement requirements, logistics factors, etc. The operations network in this example is outlined in terms of cargoes, transportation resource and operation scenarios:

Cargo: Cargo information described by the simplified TPFDD as movement requirement is specified in Table 4-1. For a graphical representation of the TPFDD, please refer to Figure 4-2.

Table 4-1 TPFDD for the Backward Problem

APOE	APOD	EAD(min)	LAD(min)	TONNAGE
ALLEGHENY CO	MYRTLE BEACH AFB	303	1685	256
HAWTHORNE MUNI	GADSDEN MUNI AFB	144	1825	267
DULUTH INT	MYRTLE BEACH AFB	636	2851	138

Transportation Resource:

- Aircraft Capacity: 500 tons
- Aircraft Speed: 120 mph.
- Aircraft Berth:
  - Starting Depot: SAN FRANCISCO INTL.
  - Returning Depot: SAN FRANCISCO INTL.

For simplicity, cargo loading/unloading time is converted into vehicle speed thus the service time s is set to zero. Aircraft capacities and cargoes are modeled here with only one dimension (weight); other dimensions such as load size and passenger/cargo type can also be incorporated.

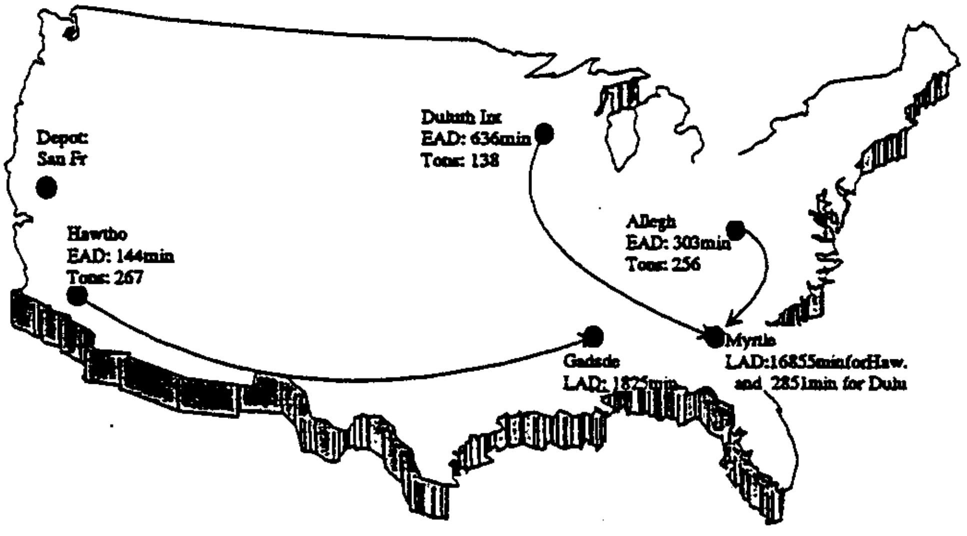

Figure 4.2 Graphical Representation of the TPFDD in the Backward Problem

Table 4-2 Operations Network Information

a. Distance Among Bases: Great Circle Distance

AIR BASES	SAN FRANC ISCO INT	ALLEGHE NYCO	HAWTHOR NE MUNI	DULUT H INT	MYRTLE BEACH AFB	GADSDEN MUNI AFB
SAN FRANCISCO INT	0	2282	347	1657	2451	2038
ALLEGHENYCO	2282	0	2165	764	473	581
HAWTHORNE MUNI	347	2165	0	1632	2269	1842
DULUTH INT	1657	764	1632	0	1151	947
MYRTLE BEACH AFB	2451	473	2269	1151	0	431
GADSDEN MUNI AFB	2038	581	1842	947	431	0

b. Flying Time Among Bases: Using t[i,j]= GCD[i,j]/AifcraftSpeed

AIR BASES	SAN FRANC ISCO INT	ALLEGHE NYCO	HAWTHOR NE MUNI	DULUT H INT	MYRTLE BEACH AFB	GADSDEN MUNI AFB
SAN FRANCISCO INT	0	1141	173	828	1270	1019
ALLEGHENYCO	1141	0	1082	382	236	290
HAWTHORNE MUNI	173	1082	0	816	1134	921
DULUTH INT	828	382	816	0	575	473
MYRTLE BEACH AFB	1270	NA	1134	575	0	215
GADSDEN MUNI AFB	1019	290	921	473	215	0

Operation Scenarios:

- Operation Time: Starting at 00.00 hr, ending at 48:00 hr (i.e. Time Duration = 48.00 hrs)
- Infrastructure:
  - Availability of Aircraft: to be decided optimally
  - Transportation Network /Operations Network: Table 4-2.

Network Construction & Optimization Network: The network construction transforms the above operations network into the optimization network , taking into consideration factors such as routes, enroutes, cargoes, fuel and other information in association with the operations network. TPFDD and transportation network information is transformed into the labeled digraph $G(N,\Lambda)$. After the network construction process, the graph and labels information are shown in Table 4-3.

Table 4-3. Optimization Network Information

a. Node *N, labels* a[i], b[i] and d[i] in $G(N,\Lambda)$

Node N	0	1	2	3	-1	-2	-3	7
Air Base	SAN FR	ALLEGH	HAWTHO	DULUTH	MYRTLE	GADSDE	MYRTLE	SAN FR
a[i]	0	303	144	636	0	0	0	0
b[i]	2880	2880	2880	2880	1685	1825	2851	2880
d[i]	0	256	267	138	-256	-267	-138	0

b. Cost c_{ij} for $G(N,\Lambda)$

Node	0	1	2	3	-1	-2	-3	7
0	0	3722	1787	3097	NA	NA	NA	NA
1	NA	0	2165	764	473	581	473	NA
2	NA	2165	0	1632	2269	1842	2269	NA
3	NA	764	1632	0	1151	947	1151	NA
-1	NA	NA	2269	1151	0	431	0	2451
-2	NA	581	NA	947	431	0	431	2038
-3	NA	473	2269	NA	0	431	0	2451
7	NA	NA	NA	NA	NA	NA	NA	0

c: Flying Time t_{ij} for $G(N,\Lambda)$

Node	0	1	2	3	-1	-2	-3	7
0	0	1141	173	828	NA	NA	NA	NA
1	NA	0	1082	382	236	290	236	NA
2	NA	1082	0	816	1134	921	1134	NA
3	NA	382	816	0	575	473	575	NA
-1	NA	NA	1134	575	0	215	0	1225
-2	NA	290	NA	473	215	0	215	1019
-3	NA	236	1134	NA	0	215	0	1225
7	NA	NA	NA	NA	NA	NA	NA	0

Please note that *-i* is equivalent to *n+i*. So either *(i,n+i)* or *(i,-i)* denote the same pickup-delivery pair. NA means not applicable.

To construct the optimization network, two tasks arise: building up the digraph topology *G*, and computing the labels. For simplicity, we take the direct physical route *(i,j)* as the arc *(i,j)* of G(N,Λ). With the arcs available, other parts of the network can be built very easily. In particular, the cost of arc *(i,j)* is defined as

$$c_{ij} = \begin{cases} GCD(i,j), & \textit{if } i \neq 0 \\ K + GCD(i,j), & \textit{if } i = 0 \end{cases}$$

, **and *K*=1441, here *K* represents the "fixed cost" of** utilizing a vehicle. In reality, arc construction is a complicated procedure which can be done in various ways according to the actual operational situation. Information concerning crew scheduling, traffic congestion, aircraft mechanical limitations, weather situation, closed air bases, hostile regions and so forth might all be included in the optimization network construction process. For example, to construct the arc from a pickup node $i \in P^{+}$ to a delivery node $n+i \in P^{-}$, we may select the shortest path *P* with the maximum length of any segment in path *P* not exceeding certain number in the operations network, which may mean that a certain type of aircraft can make a sustained flight with supported available refueling along the route.

Network Reduction & Reduced Optimization Network: The above optimization network can be further reconfigured by tightening some time windows and some inadmissible arcs through the process of network reduction, which reduces the size of the problem. There are nine rules for time window tightening and inadmissible arc elimination (cf. Chapter 4 of [6]); these rules identify infeasible/inadmissible arcs and reduce the network size. Table 4-4 shows the result of network reduction.

Table 4-4 Reduced Optimization Network Information

a: Node, labels a[i], b[i] and d[i]

Node	0	1	2	3	-1	-2	-3	7
Air Base	SAN FR	ALLEGH	HAWTHO	DULUTH	MYRTLE	GADSDE	MYRTLE	SAN FR
a[i]	0	1141	173	828	1377	1094	1403	0
b[i]	2880	1419	904	1080	1655	1825	1655	2880
d[i]	0	256	267	138	-256	-267	-138	0

b. Cost c_{ij} for $G(N,\Lambda)$

Node	S	1	2	3	-1	-2	-3	T
S	0	3722	1787	3097	NA	NA	NA	NA
1	NA	0	-	-	473	581	473	NA
2	NA	2165	0	1632	-	1842	-	NA
3	NA	764	-	0	-	947	1151	NA
-1	NA	NA	-	-	0	431	0	2451
-2	NA	581	NA	-	431	0	431	2038
-3	NA	-	-	NA	0	431	0	2451
T	NA	NA	NA	NA	NA	NA	NA	0

Note: "-" entry in above table means the arc is eliminated.

Comparing 4-3(a) with 4-4(a), it can be seen that 6 out of 8 time windows are tightened, e.g. the original time-window in Table 4-3(a) for node 1 was [303, 2880], in the reduced version it has been tightened to [1141, 1419]; Comparing 4-3(b) with 4-4(b), it can be seen that 11 out of 33 arcs in the original optimization network are eliminated, e.g. in the entry for the arc connecting node 1 to node 2 has been eliminated. These window tightening and arc reductions result in a reduced network with less computational complexity.

PDPTW & SP-CGCE OPTIMIZATION:

With G(N,A) available, the *SP* formulation for the PDPTW problem can be carried out as discussed in Appendix B. The *SP* formulation is an implicit one, because it offers the structure, but does not explicitly express the parameter values. These values, such as the cost coefficients and columns, will be generated along with the solution of the formulation. The Column Generation part of the SP-CGCE algorithm solves the LP relaxation of the SP formulation to LP optimal after generating 4 columns (Table 4-5). During the first iteration of column generation, the column which is generated corresponds to the feasible route of (0,3,1,-3,-1,7) (also see Table 4-3):

1) the vehicle leaves its home base at node 0 (referring to SAN FR),
2) picks up cargo at node 3 (referring to DULUTH, picking up 138 tons there),
3) the vehicle then goes to node 1 (referring to ALLEGH, picking up 256 tons),
4) the vehicle delivers the cargo from DULUTH (labeled node "3") to node "-3" which is the drop-off at MYRTLE,

5) the vehicle then delivers the cargo from ALLEGH (labeled node "1") to node "-1" which here also refers to MYRTLE.

The reduced cost corresponding to this column/route is -6560. By column generation technique, we know it is the minimum reduced cost among all other columns which are not in the base of the simplex algorithm. Since this is a negative value, the LP solution is not yet an optimum one; therefore, more column generations are needed. As shown in Table 4-5, after the 4th column generation iteration, the optimal solution is obtained.

Table 4-5 Column Generation Process

Column Generations	min Reduced Cost	Corresponding Route	LP Optimal?	Add This Column?
1st	-6560	(0.3.1.-3.-1.7)	N	Y
2nd	-5179	(0.2.-2.1.-1.7)	N	Y
3rd	-3946	(0.2.3.-3.-2.7)	N	Y
4th	0	All 7 Feasible Routes	Y	N

The Column Elimination part of the algorithm solves the SP problem to integer optimal using 5 out 7 total feasible columns/routes (Table 4-6), i.e. two columns are eliminated.

Table 4-6 Column Elimination Process

Route	Reduced Cost	Zupper -Zlower	Eliminated?	The Path
1	3206	1973	Y	0(T:0), 1(T:1141), -1(T:1377), 7(T:26 02)
2	1973	1973	N	0(T:0), 2(T:173), -2(T:1094), 7(T:211 3)
3	3354	1973	Y	0(T:0), 3(T:828), -3(T:1403), 7(T:262 8)
4	50000000	1973	N*	0(T:0), 2(T:173), -2(T:1094), 1(T :1384), -1(T:1620), 7(T:2845)
5	0	1973	N	0(T:0), 3(T:828), 1(T:1210), -3(T:1446,-1(T:1446), 7(T:2671)
6	0	1973	N	0(T:0), 2(T:173), 3(T:989), -3(T:1564), -2(T:1779), 7(T:2798)
7	0	1973	N	0(T:0), 3(T:828), 1(T:1210), -1(T:1446), -3 (T:1446), 7(T:2671)
8	not PD PTW feasible			0(T:0), 2(T:173), 3(T:989), 1(T:1371), -1(T:1607), -3(T:1607), -2(T:1822), 7(T:2841)
9				0(T:0), 2(T:173), 1(T:1255), -1(T:1491), -2(T:1706), 7(T:2725)
10				0(T:0), 2(T:173), 3(T:989), 1(T:1371), -3(T:1607), -1(T:1607), -2(T:1822), 7(T:2841)

*The 50000000 value is an internal flag of the implementation for code optimization

The optimization statistics and optimal solution are shown in Table 4-7, from which we know that the minimum number of vehicles used is 2 and the optimal routes and schedule are 0(T: 0), 2(T:173), -2(T:1094), 7(T:2113) and 0(T: 0), 3(T: 828), 1(T:1210), -1(T:1446), -3(T:1446), 7(T:2671).

Table 4-7 Optimization Results

a. Optimization Statistics

Optimal Values	Vehicles Needed	Pickup-Delivery Pair	Nodes	Arcs	F_Arcs
12452	2	3	7	33	22
FRts	ColumGens	LPOptimal	Zupper	RtsElimd	Ttotal
7	4	10479	12452	2	0

b. Optimal Routing & Scheduling

Binary Variable	Route Cost	Routing & Scheduling Information Format: Node(T:Arrival/Departure Time)
x1	5667	0(T: 0), 2(T:173), -2(T:1094), 7(T:2113)
x6	6785	0(T: 0), 3(T: 828), 1(T:1210), -1(T:1446), -3(T:1446), 7(T:2671)

4.2.1.3 POSTPROCESSING AND OUTPUT

Depending on the situation, various postprocessings could be done and user-friendly output could be generated. Here, we will give the GUI display of the optimal routes in Figure 4.3, the throughput in Figure 4.4, and the Vehicle-in-Use Information in Figure 4.5.

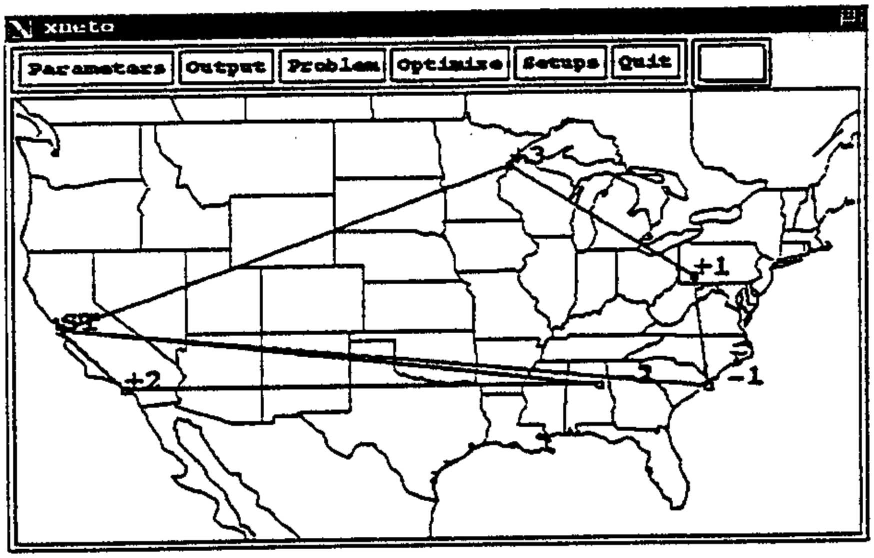

Figure 4.3 Optimal Routing in the Requirements Studies Example

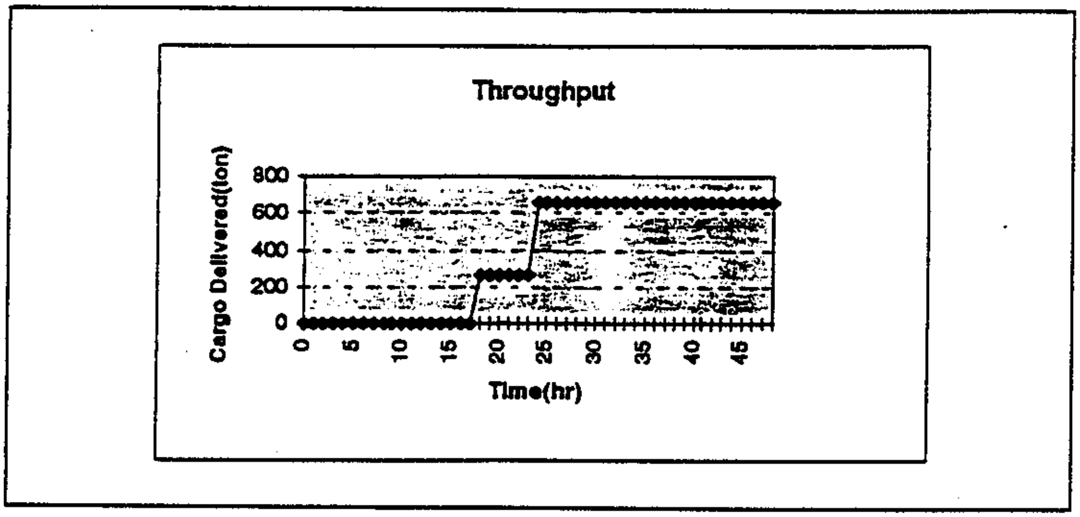

Figure 4.4 Throughput

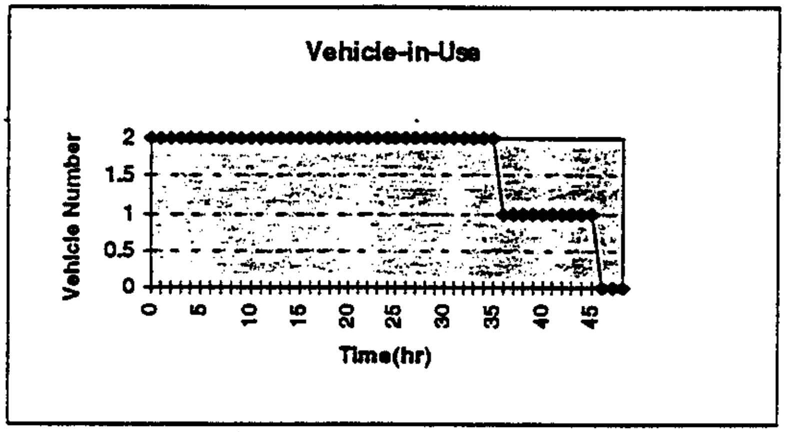

Figure 4.4 Vehicles-in-Use

Figure 4.5 Vehicles-in-Use

5. DISCUSSION OF RELATED ISSUES AND FUTURE WORK

Although the focus of this report has been our solution to the requirement studies problem, the above implemented model and solution scheme is powerful, flexible and extendible in dealing with many other real world issues. Here we mention, with some modification, those problems addressed by this approach and additional issues for further work.

1) *Vehicle Numbers:* Consideration of the number of vehicles is easily incorporated in this approach. For the minimum number of vehicles problem, what is needed is to take.

$c_{ij} = \begin{cases} K & if\ i = 0 \\ 0 & if\ i \neq 0 \end{cases}$. For problems that are concerned with an exact number m of vehicles, the constraint: $\sum_{r\in\Omega} x_r = m$ must be included. For problems with a maximum number m vehicles the constraint: $\sum_{r\in\Omega} x_r \leq m$ must be included.

2) ***Multi-depot, Nonhomogenous-vehicle:*** This approach can easily be extended to a multi-depot, nonhomogenous-vehicle situation in which the feasible routes would be obtained by applying the *Constrained Shortest Path* Algorithm to different depots and types of vehicles. The computation time/complexity increases linearly.

3) ***General VRP Problems:*** This scheme can solve general *VRP* problems, pickup only problem, delivery only problem, split, full load, with or without time windows. The only major modification is the constrained shortest path problem.

4) ***Forward Problem:*** The approach can also address the aforementioned forward problem. The forward problem can be formulated similarly by incorporating the penalty term T_{m+1}-a_i, $i \in P^+$ into the cost coefficient c_r.

5) ***Soft Time Windows:*** The approach can also address the so called soft time window problem. The penalties will be incorporated into the cost coefficient c_r to include a route with violated time windows as a feasible route.

6) ***Larger Problems:*** The algorithm can be tailored to solve even larger problems by sacrificing optimality and settling for a sub-optimal solution. One way of doing this is to first divide the original problem into subproblems using the concept of clustering, and then use this scheme to solve each subproblem to optimality. It is also possible to not require optimality in the constrained shortest path and column elimination algorithm. From another point of view, a semantic control paradigm can be employed to deal with larger problems in which the higher and intelligent layer of the system will identify the problem situation and transfer control accordingly to the lower and actuating layer of the system which in our case would be the PDPTW algorithm.

7) ***Full Load and Split Problem:*** By adding the full load requirement into the constrained shortest path problem, the algorithm solves the full load problem; by assigning different nodes to the split loads, it solves the split problem. Also, the algorithm is easily adapted to deal with regular routing problems, pickup problems, delivery problems and TSP problems. Of course, it performs better with problems which are more tightly constrained.

The objective function is flexible, i.e. various objective functions can be included. As mentioned above, an objective function to address the forward problem, the backward problem, and soft time windows can be included in the scheme. Other factors, such as travel distance, travel time, vehicle utilization considerations etc., can also be easily incorporated.

The mobility system is a large-scale and complicated system; in order to address more realistic and larger problems in mobility analysis and other large scale transportation systems, much more research is needed in addition to the work presented in this report. The following outlines several open problems that need to be addressed:

1) ***Crew Scheduling:*** There are regulations/constraints on the working hours of crew members. The crew scheduling issues can also be incorporated into the scheme. One way is to do (vehicle) routing first and (crew) scheduling second, in which case crew scheduling will take place after the vehicle routes are settled. Crew scheduling could also be done along with feasible route generation, in which case each feasible route needs to meet crew scheduling constraints.

2) ***More efficient parallel algorithm development for the constrained SPP problem:*** The column generation-column elimination algorithm is efficient in finding the integer optimal when the relaxed LP optimal is achieved. Unfortunately, solving the constrained SPP problem for the column generation process to obtain the LP optimal is very time-consuming and computer memory-intensive. It is the major bottleneck of the algorithm; therefore a more efficient algorithm for the constrained SPP problem is desired. In recognition of the ever-increasing use of parallel computing, parallel algorithm development may be a worthwhile pursuit. The use of parallel algorithms will

be achievable in the near future since the process for solving the constrained SPP problem is based on dynamic programming and has very strong parallelism.

3) *Vehicle Concurrence Issues:* Vehicles might compete for common resources such as routes, crew members, air fields etc., which could affect operations. The effects of one vehicle on another make the system a time-dependent and vehicle-dependent dynamic system; these issues are not considered here. The set-partitioning formulation, column-generation solution approach might be inherently weak in modeling these factors. One possible way of handling this is by a route-first and concurrence-check-second approach. Since the optimal solution is often not unique, the check can be first conducted for all optimal solutions. If none satisfies the concurrence check criteria, certain modifications need to be performed.

4) *Nonlinear Loading Algorithm:* In this report, a multidimensional linear loading algorithm is used (constraint B-5 for load progression in the formulation given in Appendix B). In more complicated cases, nonlinear loading may be involved, and issues concerning nonlinear loading algorithms coupled with the optimality analysis should be explored. Generally, any loading algorithm could replace the existing loading in the NETO as long as it can be incorporated in the constrained shortest path problem.

5) *Dynamic Routing and Scheduling:* The situation studied in this paper is basically a static routing and scheduling problem with time windows and capacity constraints. The movement requirement is known in advance. In some situations, the movement requirement is dynamic and the routing and scheduling should be performed continuously.

6) *Probabilistic Considerations:* In this report we have assumed that the parameters of the operation, including the network, resources, etc. are all deterministic. But in real situations, various uncertainties could be involved in many aspects of the problem; therefore probabilistic studies are useful in addressing more realistic scenarios.

6. CONCLUSION

This report is based on a doctoral dissertation by the first author [6]; it is the first attempt to use column generation-column elimination scheme to solve VRP problems in general and VRPTW and PDPTW problems in particular; it is also the first attempt to model and solve the mobility analysis system problems using network optimization with time window constrained routing and scheduling.

The new model (NETO) not only offers optimal solutions but also solves both the forward problem and the backward problem. Above all, it is flexible and can be extended to include many additional practical and operational constraints and considerations. The SP-CGCE algorithm is an efficient and competitive approach to solve practical vehicle routing and scheduling problems. The computational results presented briefly in section 3.3 of this report indicate robust performance for the algorithm. All these characteristics discussed above make NETO, with the SP-CGCE algorithm powerful, flexible and practical.

In summary a new mobility analysis model named NETO [6] is proposed to address various limitations of the existing ones. The new model consists of a network optimization engine with time window constrained routing and scheduling that is based on integer and combinatorial optimization methodology, and an analysis system with a management information system built upon RDBMS and multimedia technology. It is our belief that NETO with the SP-CGCE algorithm can be, should be and will be utilized to solve practical mobility analysis problems as well as other transportation system related problems.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the AFOSR for funding this research, to anonymous referees for their insightful suggestions, to Lt. Col. Anthony Waisanen of AMC for many fruitful discussions and his knowledge of military transportation problems and to Col. Craig M. Northrup of AMC for his support of and interest in this work.

APPENDIX A: NOTATIONS AND DEFINITIONS

AMC:	Air Mobility Command
APOE:	Aerial Port of Embarkation
APOD:	Aerial Port of Debarkation
CINC:	Commander-in-chief
COA:	Courses of Action
CONUS:	Continental United States
EAD:	Earliest Available Date
FLOGEN:	Flow Generator
GCD:	Great Circle Distance
GUI:	Graphical User Interface
IP:	Integer Programming
LAD:	Latest Arrival Date
LP:	Linear Programming
MASS:	Mobility Analysis Support System
MASS:	Mobility Analysis Simulation System
MIDAS:	Model for Intertheater Deployment By Air and Sea
MIP:	Mixed Integer Programming
MIS	Management Information System
MSC:	Military Sealift Command
NETO:	Network Optimization Mobility Analysis System
OSD:	Office of the Secretary of Defense
PDPTW:	Pickup and Delivery Problem with Time Window Constraint
RDBMS	Relational Database Management System
RIMS:	Revised Intertheater Mobility Study
SP-CGCE:	Set-partitioning Formulation , Column Generation Column Elimination
SPP:	Shortest Path Problem
TPFDD:	Time Phased Force Deployment Data
TSP:	Traveling Salesman Problem
VRP:	Vehicle Routing and Scheduling Problem
VRPTW:	Vehicle Routing and Scheduling Problem with Time Window Constraint
R^n_+:	*n* dimensional real vector space

Z_+^n: a set of non-negative integral n-dimensional vector space

B_+^n: a set of non-negative binary n-dimensional vector space

P^+: pickup node set, $P^+ = \{1,2,\ldots,n\}$. The corresponding delivery node to $i \in P^+$ is $n+i$, also referred to as *-i*

P^-: delivery node set, $P^- = \{n+1, n+2, \ldots, 2n\}$*=(-1, -2, ... -n}*

P: operation node set, $P \equiv P^+ \cup P^-$. P includes all pickup and delivery nodes

S: Starting node set from which vehicles departure. For single depot case, $S = \{0\}$. S is also used for the space $\{0,1\}^\Omega$

T: Terminating node set to which vehicles return. For single depot case $T = \{2n+1\}$

N: all nodes of the optimization network, $N = S \cup P \cup T$ *=(0, 1, ..., n, n+1, ..., 2n, 2N=1)*

Λ: all arcs of the optimization network, $\Lambda = S \times P^+ \cup P^+ \times P^- \cup P^- \times T$

$G(N,\Lambda)$: the original underlying graph of the *optimization network*

ρ_r: a feasible route in $G(N,\Lambda)$

Ω: or $\Omega(N,\Lambda)$, the set of all feasible routes in $G(N,\Lambda)$. $\Omega = \{\rho_r\}$

$|\Omega|$: the cardinality of Ω

X_B: the basic variables in the simplex method

X_N: the non-basic variables

c_B: the cost coefficient corresponding to X_B

c_N: the coefficient corresponding to X_N

B: the basis in the simplex method

N: the non-basic columns.

Ω_B: i.e. $\{\rho_r : \delta_r \in B\}$, the set of the feasible routes that correspond to the columns in the feasible base B

δ_r: column coefficient of the set partitioning formulation, corresponding to feasible route ρ_r, where $\delta_{ir} = \begin{cases} 0 & \text{if node } i \text{ is not on route } r \\ 1 & \text{if node } i \text{ is on route } r \end{cases}$ $i \in P^+$, $r \in \{ s \mid \rho_s \in \Omega\}$

x_r: $x_r = \begin{cases} 0 & \text{if feasible route } r \text{ is not selected in the solution} \\ 1 & \text{if feasible route } r \text{ is selected in the solution} \end{cases}$ $r = \{r \mid \rho_r \in \Omega\}$

binary variable

SP: the original master problem with set partition formulation

RSP: the linear relaxation of SP

ASP: the augmented set partition master problem

$RASP$: the linear relaxation of ASP

$\tilde{x}_r^R$: the optimal solution variable for RSP or $RASP$

$\tilde{x}_i$: a feasible solution for SP (integer solution)

x_{ij}: x_{ij} where $(i,j) \in \Lambda$ is the vehicle flow variables of the feasible route $\rho_r \in \Omega$.

$$x_{ij} = \begin{cases} 1 & \text{if feasible route } \rho_r \text{ goes directly from } i \text{ to } j \\ 0 & \text{if feasible route } \rho_r \text{ does not go directly form } i \text{ to } j \end{cases}$$

c_{ij}: the cost of arc (i,j)

$\bar{c}_{ij}$: artificial cost of arc (i,j) for the shortest path problem

c_r: the cost of route ρ_r: $c_r = \sum_{(i,j)\in\Lambda} c_{ij} x_{ij}$

$\bar{c}_r$: reduced cost $\bar{c}_r = c_r - \pi\delta_r$, where π is the dual variables vector/simplex multiplier

S: the original problem space, $S = \{0,1\}^{|\Omega|}$. S is also used for the starting nodes

S_R: i.e. S_{RSP}, or S_{RASP}, the problem space of RSP or $RASP$, $S_R \in R_+^{|\Omega|}$ and is the linear relaxation of S

$\bar{d}_i$: load vector(volume, weight...) of cargo i at node i

$[a_i, b_i]$: pickup time window at node i for movement/cargo i

$[a_0, b_0]$: time window for vehicle leaving the depot S

$[a_{2n+1}, b_{2n+1}]$: time window for vehicle returning to the depot T

$\bar{D}$: capacity of vehicle(load weight limit, volume,...)

t_{ij}: travel time from node $i \in N$ to node $j \in N$

s_i: service time(pickup time or delivery time) at node $i \in N$

$\bar{Y}_i$: the total load on the vehicle just after it leaves node $i \in N$

T_i: time of start service at node $i \in N$

T_0: arrival time at node i or time vehicle leaves the depot S

T_{2n+1}: time vehicle returns to the depot T

RT: feasible route defining formulation

RTL: columns to be generated each time for the column generation process

$\delta^{-}(j)$: the set of nodes that are connected to node j

$\rho_{\alpha}^{k}(j)$: the α^{th} route in all routes that starts at S and ends at j with k arcs

$P^{k}(j)$: the set of all routes that start at S and end at j with k arcs, i.e. $P^{k}(j) = \bigcup_{\alpha} \rho_{\alpha}^{k}(j)$

$h_{\alpha}^{k}(j)$: the cost of route $\rho_{\alpha}^{k}(j)$

$T_{\alpha}^{k}(j)$: arrival time at node j of route $\rho_{\alpha}^{k}(j)$

$\overline{Y}_{\alpha}^{k}(j)$: vehicle load at node j along route $\rho_{\alpha}^{k}(j)$

APPENDIX B: SET PARTITIONING FORMULATION OF THE PDPTW

B.1 Route Defining Formulation

The PDPTW problem is formulated as a set-partitioning model; the formulation is based on the concept of feasible routes:

A ***feasible route*** $\rho_r$in $G(N,\Lambda)$ is a non-cyclic path that originates from S and terminates at T, while satisfying pairing constraints, precedence constraints, capacity constraints and time window constraints. Introducing binary route flow variable x_{ij} as

$$x_{ij} = \begin{cases} 1 & \text{if the feasible route } r \text{ goes directly from } i \text{ to } j \\ 0 & \text{if the feasible route } r \text{ does not go directly form } i \text{ to } j \end{cases} \quad (i,j) \in \Lambda.$$

Then ρ_r can be defined as follows:

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{j,n+i} = 0, \; i \in P^{+} \qquad \text{(B-1) (pairing constraints)}$$

$$T_i + s_i + t_{i,n+i} \le T_{n+i}, \; i \in P^{+}) \qquad \text{(B-2) (precedence constraints)}$$

$$\textit{Route:} \left.\begin{aligned} & x_{ij} = 1 \Rightarrow T_i + s_i + t_{ij} \le T_j, \; i,j \in P \\ & x_{0j} = 1 \Rightarrow T_0 + t_0 \le T_j, \; j \in P^{+} \\ & x_{i,2n+1} = 1 \Rightarrow T_i + s_i + t_{i,2n+1} \le T_{2n+1}, \; i,j \in P \end{aligned}\right\} \text{(B-3) (time progression)}$$

$$\left.\begin{aligned} & a_i \le T_i \le b_i, \; i \in P \\ & a_0 \le T_0 \le b_0 \\ & a_{2n+1} \le T_{2n+1} \le b_{2n+1} \end{aligned}\right\} \text{(B-4) (time window constraints)}$$

$$\left.\begin{aligned} & x_{ij} = 1 \Rightarrow \overline{Y}_i + \overline{d}_j = \overline{Y}_j, \; i \in P, \; j \in P^{+} \\ & x_{ij} = 1 \Rightarrow \overline{Y}_i - \overline{d}_j = \overline{Y}_j, \; i \in P, \; j \in P^{-} \\ & x_{0j} = 1 \Rightarrow \overline{Y}_0 + \overline{d}_j = \overline{Y}_j, \; j \in P^{+} \end{aligned}\right\} \text{(B-5) (load progression)}$$

$$0 \le \overline{Y}_i \le \overline{D}, \; i \in P^{+} \qquad \text{(B-6) (capacity constraint)}$$

In the above formulation, equation (B-1) ensures that both the pickup node i and its correponding delivery node $n+i$ are on the same route ρ_r; (B-2) ensures that on the route ρ_r, pick-up is performed before delivery; (B-3) represents the time progression in the network, while (B-4) are time window constraints. Contraints (B-5) express the compatibility requirements between routes and vehicle loads, while constraints (B-6) are the capacity constraints.

B-2 Set Partitioning Formulation:
In set-partitioning formulation, the column coefficients δ_r in the constraint matrix are defined by the *feasible route* ρ_r in $G(N,\Lambda)$ in the following way:

B-2 Set Partitioning Formulation:
In set-partitioning formulation, the column coefficients δ_r in the constraint matrix are defined by the *feasible route* ρ_r in $G(N,\Lambda)$ in the following way:

$$\delta_r = [\delta_{ir}]_{n\times 1}, \text{ where } \quad \delta_{ir} = \begin{cases} 0 & \text{if node } i \text{ is not on route } \rho_r \\ 1 & \text{if node } i \text{ is on route } \rho_r \end{cases} \quad i \in P^+.$$

Next let's introduce the binary decision variable x_r:

$$x_r = \begin{cases} 0 & \text{if the feasible route } \rho_r \text{ is not selected in the solution} \\ 1 & \text{if the feasible route } \rho_r \text{ is selected in the solution} \end{cases}, \quad r \in \{r \mid \rho_r \in \Omega\}$$

and the cost coefficient c_r associated with x_r or within the feasible route ρ_r. Note that x_r is defined through ρ_r, and ρ_r is defined through x_{ij}. Then we are ready to give the set-partitioning formulation for the *PDPTW* problem:

$$z = \min \sum_{r\in\Omega} c_r x_r$$

$$SP: \quad st. \quad \sum_{r\in\Omega} \delta_{ir} x_r = 1, \quad i \in P^+$$

$$x_r \in \{0,1\}, \quad r \in \Omega \text{ ie. } X \in S = \{0,1\}^{|\Omega|}$$

In the above *SP* formulation, the column coefficients δ_r and the cost coefficient c_r are not explicitly available. They need to be obtained through corresponding feasible route ρ_r, which were defined previously. A flexible objective function can be obtained by varying the exact formulation of the c_r's. In particular to solve the requirement studies problem, if $c_{ij} = \begin{cases} K \text{ if } i = 0 \\ 0 \text{ if } i \neq 0 \end{cases}$ then the objective is to minimize the number of vehicles used. In general, the number of feasible routes ρ_r and the number of columns δ_r in $|\Omega|$ is huge, making it computationally prohibitive to enumerate all feasible routes/columns and solve the *SP* problem to integer optimality.

REFERENCES

[1] E.Y. Rodin, "Semantic Control Theory," Appl. Math. Lett. Vol. 1, No., 1 (1988), pp. 73-78

[2] Y. Lirov, Artificial Intelligence Methods in Decision and Control Systems, doctoral dissertation, Washington University, St. Louis, MO, (August 1987)

[3] S.M. Amin, Intelligent Prediction Methodologies in the Navigation of Autonomous Vehicles, doctoral dissertation, Washington University, St. Louis, MO, (January 1990)

[4] R.D. Weil, AI Methods in Utilizing Low Dimensional Models of Differential Games, doctoral dissertation, Washington University, St. Louis, MO, (September 1990)

[5] D. Geist, Semantic Control in Continuous Systems Applications to Aerospace Problems, doctoral dissertation, Washington University, St. Louis, MO, (December 1990)

[6] F. Yang, Network Optimization with Time Window Constrained Routing and Scheduling, doctoral dissertation, Washington University, St. Louis, MO , (Aug. 1995)

[7] J. Schank, M. Mattock,G. Sumner, I . Greenberg, J. Rothenberg, and J. P. Stucker, "A review of Strategic Mobility Models and Analysis," RAND National defense research institute, report # R-3926-JS, (1991)

[8] E.Y. Rodin, et al, "Intelligent Aircraft Routing & scheduling Project," *Center for Optimization and Semantic Control*, Washington University, St. Louis, MO, (1993)

[9] S.M. Amin, "Artificial intelligence methodologies in airlift transportation networks", *Center of Optimization and Semantic Control*, Washington University, St. Louis, MO (1992)

[10] K. Ruland,"Polyhedral solution to Pick-up and Delivery problems", doctoral dissertation, Systems Science and Mathematics, Washington University,St. Louis,MO, (August 1995)

[11] S. Arunapuram "Vehicle routing and scheduling with full loads", doctoral dissertation, *Case Western Reserve University*, (1993)

[12] M. Desrochers, J. Lenstra, M. Savelsbergh, F. Soumis, Vehicle Routing with time windows: optimization and approximation. In *Vehicle routing: Methods and Studies*. B. Golden and A. Assad(eds.). Norht-Holland, Amsterdam, 65-84, (1988)

[13] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh. "A classification scheme for vehicle routing and scheduling problems," European. J. of Opers. Res., Vol. 46, 322-332, (1990)

[14] M. Desrochers, J. Desrosiers, M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research*, Vol. 40, No. 2, (March-April 1992)

[15] J. Desrosiers, Soumis,F., and Desrochers,M., "Routing with time windows by column generation," *Networks*, Vol.14, 545-565, (1984)

[16] Y. Dumas, " Confection d'itineraire de vehicules en vue du transport de plusieurs origines a plusieurs destinations, " Publication 434, centre de techerche sur les transports, Universite de Montreal, (1985)

[17] Y. Dumas, J. Desrosiers and F. Soumis, "The pickup and delivery problem with time windows," *European Journal of Operational Research*, Vol. 54, 7-22, (1991)

[18] Laporte, "The Vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational /research*, Vol. 9, 345-358, (1992)

[19] C.C. Ribeiro and Soumis, F., "A Column Generation Approach to the Multiple-depot vehicle Scheduling Problem", *Operations Research* Vol 42, No. 1, (January-February 1994)

[20] M. Sol and Savelsbergh, M.W.P "A Branch-and-Price algorithm for the Pickup and Delivery Problem with time windows", Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, (1995)

[21] M. Solomon and Desrosiers,J., "Time window constrained routing and scheduling problems," *Transportation Science*, Vol. 22, 1-13, (1988)

[22] M.L. Balinski and Quandt, R.E., "On an integer program for a delivery problem," Oper. Res. Vol. 12, 300-304, (1964)

[23] Y. Agarwal, Mathur, K., and Salkin,H.M., "A set-partitioning-based exact algorithmfor the vehicle routing problem", Networks, Vol.19, 731-749, (1989)

[24] J.F. Pierce and J. S. Lasky, J.S., "Improved combinatorial programming algorithms for a class of all zero-one integer programming problems," Management Sci., Vol. 19, No. 5, 528-543, (1973)

ENDNOTES

[1] Submitted August 1995; In final form November 1995

[2] This work was supported in part by AFOSR under grant number 890158.

[3] A complete list of terminology, definitions, notation, and symbols is given in the Appendix A at the end of this report.

[4] The networks were trained and validated on a declassified TPFDD file from Operation Just Cause in Panama.