Discrete time formulation for the assignment problem applied in cross docking facilities

Saharidis G.K.D.
Department of Mechanical Engineering, University of Thessaly, Leoforos Athinon, Pedion Areos, 38834 Volos, Greece and Kathikas Institute of Research & Technology, Phafos, Cyprus Email: saharidis@gmail.com

Golias M.M. (Corresponding author)
Department of Civil Engineering, University of Memphis, Memphis, 3815 Central Ave Memphis, TN 38152, USA, Phone: 908-202-5479, Fax: 901-678-3026, Email: mgkolias@memphis.edu

Zhang T.
Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Ernest Cockrell, Jr. Hall, University Station C1761, Austin, TX 78712, Email: ti.zhang@austin.utexas.edu

Word Count: 5 089
Figure and Tables Count: 2 x 250 = 500
Total Words: 5 589

Submitted for presentation at the 91st Transportation Research Board Annual Meeting
Abstract: Assignment problems arise in different situations in transportation and are a well-studied topic in combinatorial optimization. Despite this fact, few works have been published on the use of a discrete time formulation. In this paper two new mathematical formulations are presented for the scheduling of inbound trucks to doors at a cross-docking facility, and compared to the classical machine scheduling formulation. In the first formulation a continuous time representation is considered (similar to the classical one commonly found in the literature). In the second formulation, a discrete time representation is proposed. Numerical results, obtained using an exact algorithm, are presented in order to illustrate and critically discuss each formulation.
INTRODUCTION

Assignment problems arise in different situations where we have to find an optimal way to assign $n$ objects to $m$ other objects in an injective fashion. Assignment problems are a well-studied topic in combinatorial optimization and find numerous applications in production planning, telecommunication, economics, logistics, transportation, and so on. The linear assignment problem (LAP) is one of the oldest and most-studied problems in combinatorial optimization. More frequently for the LAP two or three dimensional decision variables are introduced in order to develop a series of linear constraints. There are many applications of the LAP as for example location and tracing objects in space, scheduling on parallel machines, inventory planning, vehicle and crew scheduling, wiring of typewriters, machine scheduling, berth allocation, quay crane scheduling and so on.

Many algorithms have been developed to solve this problem as well as other aspects of the problem such as asymptotic behavior or special cases (that have been thoroughly investigated). Results published in the literature focused on the solution of classical continuous time formulations and/or finding special cases of the assignment problem. Only few papers presented a conceptually different formulation for the LAP [1]. In this paper, we add to this limited literature and present two new formulations for the assignment problem which we apply it to the case of truck scheduling at a cross dock facility. The two new formulations follow a continuous and a discrete time representation respectively. The major goal of this paper is to evaluate if using a discrete time representation approach can have significant benefits to the CPU time of current exact solution algorithms.

The outline of the paper is as follow. Section 2 presents our case study as well as a review on the scheduling of cross-docking operation. In section 3 the developed mathematical formulations are presented in details. Section 4 gives some numerical results applying the new mathematical formulations and presents a comparison with the classical one. Finally, section 5 draws conclusions indicating future perspectives for this work.

BACKGROUND

In today’s customer driven economy, moving products quickly, efficiently, and cost effectively offers a crucial advantage to companies. To achieve such goals, more and more companies, are finding that cross-docking can play an integral part of their distribution model, partially replacing or complementing existing warehousing facilities. A January 2008 survey of 547 logistics professionals performed by Saddle Creek’s \(^1\) showed that more than 52% of the respondents have already used cross-docking in their distribution operations somehow; and 13% are planning to add cross-docking to their logistics plan in the next 1 to 2 years. Different from warehousing distribution, cross-docking eliminates the process of storing in a warehouse thus reduce the inventory costs.

Cross-docking is a material handling and distribution operation, which moves products quickly and directly from inbound trucks to outbound trucks through the cross-dock facility, where products will be resorted or consolidated, without being stored or only with a short-term storage mostly within 24 hours or sometimes within only one hour. It is actually not a new practice and has been used by few companies for decades, but recently rehashed and merged up for more applications due to its significance in today’s business world than ever. Walmart pioneers cross-docking operations and runs 85% of its products through cross-docking system. By using cross-docking, Walmart is able to reduce its costs of sales by 2-3% and thus offer lower price every day than its competitors [3]. Belk, the largest privately held department store chain in the United States, transfer 90% of its inbound products through cross-dock\(^2\). This strategy results in increasing the throughput, as it needs 21 days to move products from vendor to the store fixture

---

\(^1\) http://outsourced-logistics.com/field_reports/cross_docking_trends_report_0808/
\(^2\) http://www.inboundlogistics.com/cms/article/belk-consolidation-to-the-max/
before cross-docking, but only eight days after cross-docking the products. Cross-docking also stabilizes the item prices, and in turn, makes sale predictable and therefore forecasting more accurate for using cross-docking operation.

Above all, main benefits of cross-docking operations are a) reduces storage of inventory, reduce the handling costs and operating costs; b) reduces or eliminates the warehousing costs; c) accelerate the speed of delivering products to the distributor and consequently to customers; d) increases perspective sale space; e) enables retailers to streamline the supply chain from point of origin to point of sale. Cross-docking is for minimizing warehousing and achieving thimbleful inventories [3]. A cross-docking center that is located well to its stores would make distribution process faster and more efficient, as well as saving transportation costs which, in all, would minimize products handling by reduce the number of touches to a product. The benefits of reducing costs and improving service levels by shifting to cross-docking encourage companies to use this practice more actively.

Literature review

Published research work, so far, is limited to a few papers dealing with truck scheduling for cross-docking systems, while few papers approach the door assignment problem. For the door assignment problem, research is focusing in finding the travel time or travel distance of cargo inside the cross-dockings facility. Tsui and Chang [5, 6] proposed a door assignment method to allocate the inbound trailers and outbound trailers to minimize the total weighted distance traveled by the forklifts in a freight yard. Bozer and Carlo [10] considered a static door assignment problem and presented a MIP model for a LTL cross-docking to minimize travel time of freight from inbound doors to outbound doors. They prohibited more than two adjacent outbound trailers from assigning to the same door to avoid congestion. The proposed approach is only suitable when the numbers of trailers and doors are equal. Since trailers arrival/departure time, operational time for shipment, and the cross-dock capacity can affect the system operation, additional constraints for door assignment in a cross-docking system, such as time window and capacity constraint, in case that the number of trucks is higher than the number of doors available and the capacity of the cross-dock is limited, are considered by Miao et. al [11]. They formulated an integer programming model to optimally assign trucks in order to minimize the operational cost of the cargo shipments and total number of unfulfilled shipments. The two objectives were combined as “total cost” by converting the unfulfilled shipments into penalty cost. Two metaheuristic approaches, Tabu Search and Genetic Algorithms are proposed to solve this problem.

Bartholdi and Gue [7] studied the trailers door assignment problem around the dock to improve worker’s efficiency in terms of travel cost. Three types of typical congestion experienced in cross-docking system were discussed to determine suitable layouts. Wang et. al [12] claimed that minimizing labor costs is not necessarily the only goal in cross-dock operations, due to the need to decrease transportation lead-time in coordination to JIT (just in time) [8], make-to-order or merge-in-transit strategies. They attempted to minimize the time freight spends in a cross-docking facility and they used a dynamic simulation model to get the conclusion that leave-early algorithm can save time for outgoing trucks. Later, Wang and Regan [9] scheduled trailers based on real-time information using simulation. They developed time-oriented scheduling algorithms to measure the average time freight spent at cross-docks. Chen and Lee [15] proposed a branch and bound algorithm and showed its efficiency within up to 60 jobs, where we have two types of jobs: a) download and unpack the inbound products, b) collecting those products with the same destination into an outbound truck. Chen et. al [14] minimized the total scheduling makespan at a two stage cross-dock by building a mixed integer program. Two heuristic methods—Johnson’s rule-based heuristic and dynamic Johnson’s rule-based heuristic—were presented compared with two lower bounds for large number of jobs. However, the model was limited to one outbound trailer with multiple inbound trailers. Finally, Yu and Egbelu [13] discussed the inbound and outbound trucks scheduling problem in cross-dock with temporary storage. Their objective was
find the best truck scheduling sequence for both inbound and outbound trucks to minimize total operation time while a temporary storage buffer is located at the shipping dock. They built a MIP model to solve the cross-docking problem based on assumptions that there is only one receiving dock and one shipping dock, which is not necessarily the case in practice.

Most scheduling problems associated with cross-docking lead to strongly NP-hard formulations using continuous representation of time [10, 14, 15]. In this paper a novel formulation is presented to solve the scheduling problem of assigning inbound trucks at a cross-docking facility. The classical machine scheduling formulation adopted for the cross-docking facility problem is presented for comparison reasons as well as extension of it where the number of decision variable and constraints is decreased. The novel formulation use discrete representation of time resulting a pure integer linear problem. The developed model is solved faster and could provide an exact optimal solution for significant large problems where the classical formulation fail (see section 4). In the following section we present these three models as well as the optimization objective taken under consideration.

MODEL FORMULATION
Receiving commodities into a cross-dock needs to be carefully planned. For a typical cross-docking facility, and before the arrival of inbound trucks, an inbound door has to be allocated to the inbound truck. By means of real-time information technology, the arrival time of inbound trucks may be known as a priori to the cross-docking operators. The inbound trucks carrying various products from different suppliers arrive at the inbound doors of the cross-docking facility. On their arrival, drivers of inbound trucks report to the gatehouse, where staff check the vehicle documentation and direct the driver where to go. All these trucks are either directly assigned to an inbound door or need to wait in line in the parking space in front of the inbound doors that they are assigned. This is due to the limited resource of equipment, workers and doors at the facility, that cannot handle all the inbound trucks upon arrival. Inbound truck door assignment and scheduling (i.e. start and end time that a truck occupies a door) ideally begins before the arrival of inbound trucks. In the present work, the 3 mathematical formulations have as their objective to minimize the total waiting time (i.e. difference between the start time and arrival time of trucks) and the total handling time for all the inbound trucks. There are three reasons we chose to use this objective:

1. Late start time translates into unproductive time, for the driver, spent waiting
2. Performance of inbound door operations is measured by how fast the inbound trucks are served given fixed resources. Since the inbound trucks will leave as soon as they finish unloading all the commodities, the earlier the inbound trucks depart, the less waiting time for its successors, and the faster their services are
3. Late start time means increased number of trucks and congestion at the yard

Chronologically, the first approach for the assignment problem used a continuous time representation [11][14][15] where the start as well as the completion of an activity are variables determined as outputs of the optimization process. Using a continuous time formulation excludes the need to approximate any data and in the same time the obtained solution (if it is reasonable to obtain it) is always the optimal solution to the problem. The developed model in this case corresponds to a mixed integer linear problem (MILP) and as in several industrial problems solving a MILP based on branch-and-bound or branch-and-cut methods can be difficult and, in certain cases, inadequate due to the associated complexity (i.e. number of variables and constraints). Given this complexity, an otherwise sufficient algorithm requires an impractical amount of time simply to find the first integer-feasible solution even when the problem is well-formulated. As an alternative, several researchers developed heuristic algorithms for the assignment problem [11][14] and especially some of them are applicable to the scheduling of the cross-docking operations [2][4]. These heuristic algorithms allow the solution of large-scale
problems. The disadvantage of a heuristic approach is that neither global optimality, nor feasibility, can be guaranteed. In practice, these methods provide a solution for large-scale problems when it is impossible to obtain a feasible solution by an exact procedure.

In the following sub-sections we present three formulations for the assignment of trucks at the inbound doors of a cross docking facility. The first formulation is based on the classical machine scheduling approach with a continuous time representation. The second formulation is based on a continuous time representation but with a reduced number of variables. Finally, the third formulation is based on a discrete time representation. The latter formulation discretizes the time horizon giving rise to a pure integer 0-1 linear model that has a number of attributes that assist in the development of novel solution algorithms (heuristic and combination of exact and heuristic). Before we proceed with the three formulations we present a common, to all three models nomenclature:

Nomenclature

Indexes and Sets:

- \( j, m \) Index of jobs (e.g. index of inbound trucks)
- \( J \) Set of jobs (e.g. set of inbound trucks)
- \( i \) Index of machines (e.g. index of inbound docks)
- \( I \) Set of machines (e.g. set of inbound docks)

Data

- \( AT_j \) Scheduled time of job \( j \) (e.g. arrival time for truck \( j \))
- \( C_{j,i} \) Handling time for job \( j \) at machine \( i \) (e.g. handling time for truck \( j \) at dock \( i \))
- \( S_i \) Time machine \( i \) becomes available for the first time in the planning horizon (e.g. time door \( i \) becomes available for the first time in the planning horizon)
- \( M \) Big positive number

Classical modeling using continuous time formulation and three-index decision variables

The first model (M1) that we present for comparison reasons is the classical modeling approach which applies the classical machine scheduling formulation. The developed model uses two three-index decision variables. The first decision variable \( T_{j,i,k} \) represents the start time of the assigned job to a machine (e.g. a truck to a dock) with a certain order, and the second one \( X_{j,i,k} \) represents the assignment of job to a specific machine (e.g. a truck to a specific dock) with a certain order. In the following nomenclature M1, we present the decision variables of the developed models:

Nomenclature M1

Indexes and Sets:

- \( k, h \) Index of service order of jobs (e.g. service of inbound trucks). (1)
- \( K, H \) Set of service order of jobs (e.g. service of inbound trucks).

Decision Variables:

- \( T_{j,i,k} \) Continuous positive variable expresses the starting time of job \( j \) served as the \( k^{th} \) job at machine \( i \) (e.g. truck \( j \) served as the \( k^{th} \) truck at dock \( i \)).
- \( X_{j,i,k} \) Binary variable, if \( X_{j,i,k} = 1 \) then job \( j \) (e.g. truck \( j \)) is going to be served at machine \( i \) (e.g. dock \( i \)) as the \( k^{th} \) job (\( k^{th} \) truck), otherwise \( X_{j,i,k} = 0 \).

Finally, the objective function of M1 is the minimization of the total service time (starting and handling time) of all jobs (e.g. trucks) to all machines (e.g. docks): \( \text{Min} \sum_{j} \sum_{i} \sum_{k} (T_{j,i,k} + X_{j,i,k}C_{j,i}) \).
In general, we have four groups of constraints that guarantee certain operational conditions. We have constraints that guarantee the sequence of service orders at each dock and constraints expressing that the service start time of a truck should be after its arrival as well as after the service finish time of its predecessor. Constraint set (1) expresses that a job \( j \) (e.g. truck \( j \)) is served in only one machine (e.g. dock \( i \)) with only one order while constraint set (2) ensures that each machine (e.g. dock \( i \)) services one job (e.g. truck \( j \)) at a time. Constraint sets (3) and (4) estimate the start time of service for each job (e.g. truck \( j \)). Constraint set (5) defines the range of the decision variables.

**M1 Formulation Constraints**

\[
\sum_{i} \sum_{k} X_{j,i,k} = 1 \quad \forall j, i, k \quad (1)
\]

\[
\sum_{j} X_{j,i,k} \leq 1 \quad \forall i, k \quad (2)
\]

\[
T_{j,i,k}^s \geq \sum_{m \neq j, h < k} (C_{m,j}X_{m,h}) + S_i - M(1 - X_{j,i,k}) \quad \forall j, i, k > 1 \quad (3)
\]

\[
T_{j,i,k}^s \geq (S_i - AT_j)X_{j,i,k} \quad \forall j, i, k = 1 \quad (4)
\]

\[
T_{j,i,k}^s \geq 0, X_{j,i,k} = \{0,1\} \quad \forall j, i, k \quad (5)
\]

**Bi-index model using continuous time representation (M2)**

The second model (M2) uses a continuous time representation and is an extension of M1. The developed model is based on the idea that in many assignment problems a job is assigned to only one machine. Based on this special characteristic that many assignment problems have, the three-index decision variable introduced in M1 is replaced by two bi-index decision variables and two mono-index decision variables, decreasing the total number of decision variables by a significant amount. In M2, the three-index decision variable representing the start time of service in M1, is replaced by a decision variable with only one index \( (T_j^s) \). In M2, we further introduce two bi-index and two mono-index variables. The first bi-index decision variable \( (X_{j,i}) \) represents the service order of a truck \( j \) which is unique and independent of the door assignment. The second bi-index decision variable \( (W_{j,m}) \) captures that two trucks are served at the same service dock. The first mono-index decision variable \( (f_j) \) expresses if a truck is the first truck to be served at its assigned dock and the second \( (l_j) \) if it the last. This representation decreases the total number of decision variables and the total number of constraints (as compared to M1) resulting (in certain cases)\(^3\) in smaller CPU solution times.

**Nomenclature M2**

**Decision Variables:**

- \( T_j^s \) Continuous positive variable expresses the starting time of truck \( j \).
- \( X_{j,i} \) Binary variable, if \( X_{j,i} = 1 \) then job \( j \) (e.g. truck \( j \)) is assigned at machine \( i \) (e.g. dock \( i \)), otherwise \( X_{j,i} = 0 \).
- \( W_{j,m} \) Binary variable, if \( W_{j,m} = 1 \) then job \( m \) (e.g. truck \( m \neq j \)) is assigned at the same machine (e.g. dock) as job \( j \) (e.g. truck \( j \)), otherwise \( W_{j,m} = 0 \).
The principle of a discrete time representation is to split the time horizon into intervals of equal size and use binary variables to represent the start or finish of an action during a time interval. In this section, we present a novel formulation for the assignment problem -machine scheduling problem- applied to the scheduling of inbound trucks at a cross-docking facility. The main issue concerning this type of formulation, as with every model of discrete time representation [16], is the selection of the time interval size and how this size approximates the continuous time equivalent. The main advantages of this formulation are that the developed model is a pure integer linear model (using only binary variables) while its size could be significant smaller than models using continuous time representation. Due to the discretization of the time horizon,
continuous decision variables are not involved giving rise to the possible development of very sufficient heuristics algorithms (e.g. evolutionary algorithms) that traditionally work well with 0-1 decisions variables. In the following nomenclature the additional parameters and decision variables of M3 are presented:

**Nomenclature M3**

**Indexes and Sets:**

\( t, n \)

Index of time

\( T \)

Set of time periods and the upper bound of the time horizon

**Decision Variables:**

\( SC_{j,i,t} \)

Binary variable, if \( SC_{j,i,t} = 1 \) then job \( j \) (e.g. truck) is assigned at \( t \) period at machine \( i \) (e.g. dock), otherwise \( SC_{j,i,t} = 0 \).

\( CC_{j,i,t} \)

Binary variable, if \( CC_{j,i,t} = 1 \) then job \( j \) (e.g. truck) is served during \( t \) period at machine \( i \) (e.g. dock), otherwise \( CC_{j,i,t} = 0 \).

Two binary decision variables are introduced in model M3; the first one represents the assignment of a job (e.g. truck) to a machine (e.g. dock) at a specific time point and the second one expresses that a job occupies a machine (e.g. the unloading of a truck to a dock) during a specific time period. Given the new time representation there are 7 groups of constraints expressing the operational constraints of a typical assignment problem (e.g. assignment of inbound truck to a cross docking facility) with a discrete time representation:

\[
\sum_i CC_{j,i,t} \leq 1 \quad \forall j, t \quad (17)
\]

\[
\sum_t \sum_i SC_{j,i,t} = 1 \quad \forall j \quad (18)
\]

\[
\sum_j CC_{j,i,t} \leq 1 \quad \forall i, t \quad (19)
\]

\[
CC_{j,i,t} - SC_{j,i,t} \geq 0 \quad \forall j, i, t \quad (20)
\]

\[
\sum_{n=0}^{C_{ij}} CC_{j,i,t+n} - SC_{j,i,t+n} \times (C_{ij} + M) \geq -M \quad \forall j, i, t < T - C_{ij} \quad (21)
\]

\[
- \sum_{n=0}^{C_{ij}} CC_{j,i,t+n} + SC_{j,i,t+n} \times (C_{ij} - M) \geq -M \quad \forall j, i, t < T - C_{ij} \quad (22)
\]

\[
SC_{j,i,t} \geq CC_{j,i,t} - CC_{j,i,t-1} \quad \forall j, i, t > 1 \quad (23)
\]

\[
SC_{j,i,t} = \{0,1\}, CC_{j,i,t} = \{0,1\} \quad \forall j, i, t \quad (24)
\]
Constraint set (17) guarantees that a job \( j \) (e.g. truck \( j \)) is not assigned in more than one machines (e.g. dock), during the time period \( t \) and constraint set (18) that in only one period a job \( j \) (e.g. truck \( j \)) starts its service (e.g. the unloading procedure of its commodity) in only one machine \( i \) (e.g. dock \( i \)). Constraint set (19) expresses that maximum one job \( j \) (e.g. truck \( j \)) could be served in each machine \( i \) (e.g. dock \( i \)) during period \( t \). The fact that there is a service process (e.g. an unloading flow) \( (CC_{j,i,t} = 1) \) of a job \( j \) (e.g. truck \( j \)) during period \( t \) if a job is assigned (e.g. an unloading is established) during this period \( (SC_{j,i,t} = 1) \) is guaranteed by constraint set (20). Constraint sets (21) and (22) express that if a job is assigned in a certain machine (e.g. an unloading is established in a certain dock \( i \)) during a period \( t \) \( (SC_{j,i,t} = 1) \) then the job \( j \) (e.g. truck \( j \)) will be served for \( C_{ij} \) (e.g. handling time of truck \( j \) at dock \( i \)) periods. Constraint set (23) guarantees that if we did not have service process (e.g. an unloading flow) during period \( t-1 \) \( (CC_{j,i,t-1} = 0) \) and we have in period \( t \) \( (CC_{j,i,t} = 1) \) then an assignment (e.g. unloading establishment) is occurred during period \( t \) \( (SC_{j,i,t} = 1) \). Finally the objective function for the developed model, expressing the minimization of the waiting time and handling time, corresponds to the minimization of the starting time of each job (e.g. truck) and handling time which corresponds to minimize the following function:

\[
\text{Min} \sum_{i} \sum_{j} \sum_{t} [(i + 1) * (j + 1) * (t + 1) * SC_{j,i,t} + SC_{j,i,t} HT_{j,i}]
\]

In the latter objective function the decision variable \( SC_{j,i,t} \) (representing the starting point of service) is weighted with a coefficient \( ((i + 1) * (j + 1) * (t + 1)) \). The purpose of the weight is to better balance the waiting time of the trucks and avoid a number of trucks having excessive waiting times. In other words M3 implicitly considers the multi-optimal cases where the same objective function can be achieved with different assignments and thus different distributions of waiting times. M1 and M2 do not have this property.

**NUMERICAL EXAMPLES**

To compare the three proposed formulations 10 test datasets (table 1) were developed with 5 inbound doors, truck inter-arrival times of five minutes. The handling time of each truck was dependent upon on which door it was assigned to. Out of the available doors, one door was randomly selected for each truck as the preferred door (i.e. the door with the minimum handling time). The handling time at the preferred door was generated randomly, with a minimum of thirty minutes and a maximum time of two hours. The handling time of the trucks at the other doors was generated in relation to the door with the minimum handling time, with a lower bound of forty-five minutes and upper bound of three hours.

<table>
<thead>
<tr>
<th>Table 1. Example Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

The experimental results are presented in Figure 1. Note that these results were obtained on Pentium (R)4, CPU 2.40 GHz, RAM 1 GB and CPLEX 9.0 using a C++ implementation. The primary y-axis (i.e. left y-axis) shows the CPU solution time in seconds for models M1 and M2.
The secondary y-axis (i.e. right y-axis) shows the CPU solution time in seconds for model M3. We observe that for smaller problems the CPU solution time is comparable but (as expected) once a small increase in the number of trucks occurs the difference jumps exponentially. We would like to note that for M1 and M2 no solution could be found after 24hrs once the number of trucks exceeded 12. Although the proposed new formulation (M3) works well with small problems (i.e up to 16 trucks per 5 doors) it cannot handle larger problems. Nevertheless it can be used with existing algorithms that combine exact and heuristic solution approaches (18, 19) and partition the solution space.

![FIGURE 1. Computational results](image)

**CONCLUSION**

In this paper two new formulations for the scheduling of inbound trucks to inbound doors at a cross-docking facility were presented and compared to the classical “machine scheduling” based formulation. Numerical results, obtained using an exact algorithm, showed that a discrete time representation based formulation can produce optimal results using current exact solution algorithms in significantly smaller CPU time. Although the proposed discrete formulation works well with small problems it cannot handle larger problems. Nevertheless, it can be used with existing algorithms that combine exact and heuristic solution approaches (17, 18) and partition the solution space. Current research is focusing on evaluating the performance of such heuristics with the proposed discrete time formulation.

**Acknowledgements**

This material has been partially supported by the Intermodal Freight Transportation Institute, University of Memphis (IFTI) and the Kathikas Institute of Research and Technology (KIRT). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the IFTI or KIRT.

**REFERENCES**


