

# A Machine Scheduling Approach to Improving Fleet Utilization for Carriers

Soonhui Lee<sup>1</sup>, Jonathan Turner<sup>2</sup>, Mark S. Daskin<sup>3</sup>, Tito Homem-de-Mello<sup>4</sup>, Karen Smilowitz<sup>5</sup>

November 12, 2008

<sup>1</sup>soonhui@u.northwestern.edu, <sup>2</sup>jonathan@northwestern.edu,  
<sup>3</sup>m-daskin@northwestern.edu, <sup>4</sup>tito@northwestern.edu, <sup>5</sup>ksmilowitz@northwestern.edu  
Department of Industrial Engineering and Management Sciences, Northwestern University,  
Evanston 60208

## Abstract

Carriers are under increasing pressure to offset rising fuel charges with cost cutting or revenue generating schemes. One opportunity for cost reduction lies in asset management. This paper shows how a machine scheduling approach can be used to assign truck loads to delivery times and trucks. We present several models to find optimal assignments when delivery times are flexible. The models have two objectives: minimizing needed assets and minimizing the costs of schedule deviation. We further investigate the implications of this multi-objective framework by demonstrating how improvements in fleet usage translate into savings which carriers can use as incentives to promote flexible delivery times for customers.

## 1 Introduction

The price of oil more than tripled recently, as reported in the World Oil Market Chronology (2008). Resulting increases in fuel charges have penetrated many sectors of the U.S. economy, largely because of the cost to ship goods. Carriers, which often have razor-thin profit margins, often pass these expenses directly to their customers through fuel surcharges.

A carrier that is able to cut costs during a period of increasing fuel prices has a distinct advantage over its competitors and can thereby improve its market position. Minimizing driver payroll and maximizing fleet utilization are two cost cutting options. Because hiring and retaining drivers is becoming increasingly difficult, carriers are often hesitant to reduce driver wages. Therefore, they must focus on fleet utilization to reduce costs.

In this paper, we introduce a problem motivated by a carrier facing similar cost cutting pressures. In the short run, the company wants to use fewer vehicles to deliver the same number of loads. In the long run, more efficient fleet usage can lead to increased revenue opportunities as the company can serve more loads with a better understanding of load profitability. For this carrier, delivery requests from customers are typically high during morning hours. As Figure 1 illustrates, the carrier uses many trucks during these hours. During other hours part of the fleet is idle. The carrier would like to create better schedules to balance work throughout the day and reduce the fleet size.

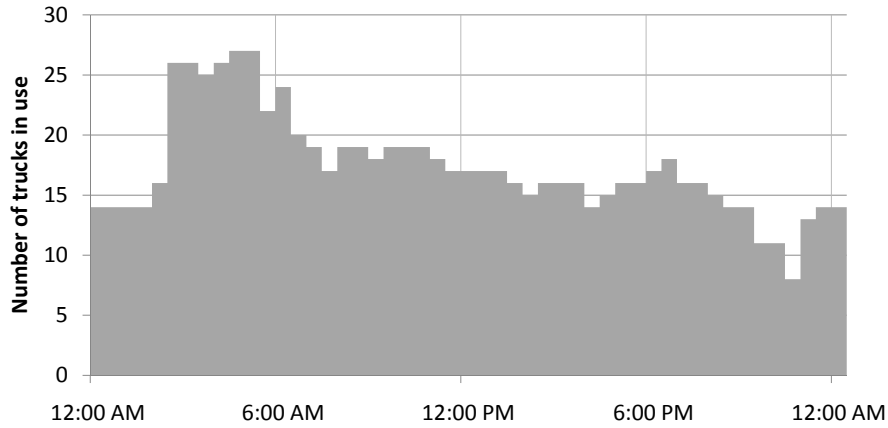


Figure 1: Trucks in use throughout a typical 24 hour period

We only consider loads that follow the pattern shown in Figure 2. The carrier transports loads between a shipper (its customer), and a consignee. Because trucks are typically washed at the terminal before they begin a delivery (although this is not always necessary), all loads begin and end at the terminal. The consignee has a requested delivery time for each load. From this time we can compute a departure and return time for each load. Because of this simple delivery structure the problem can be modeled as a scheduling problem with no routing component. Trucks carry only one load at time (i.e., they have a capacity of one). Given a set of loads with delivery request times and task durations we wish to minimize the number of trucks needed to deliver all loads on time.

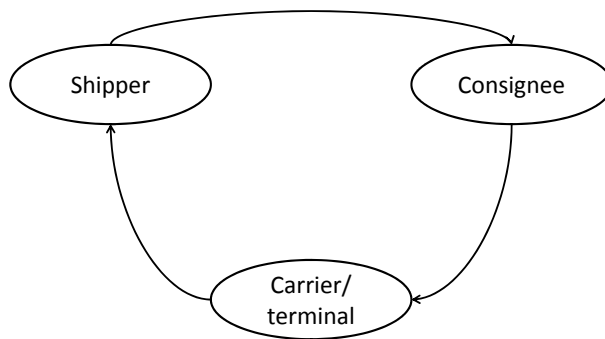


Figure 2: Standard delivery pattern

Some reduction in fleet size can be obtained through common scheduling techniques. Figure 3 and Figure 4 illustrate how simple improvements to a schedule can achieve this. These figures show nine time periods over which five loads are to be delivered. If these loads are placed on a schedule

arbitrarily, as in Figure 3, four trucks are needed. Assigning load D to truck 1, load C to truck 2, and load E to truck 3, however, eliminates the need for truck 4.

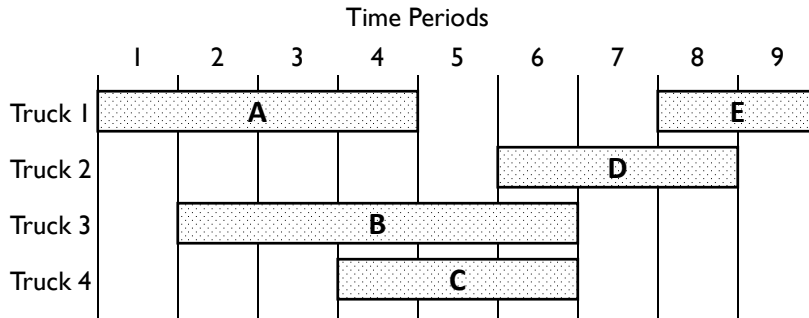


Figure 3: Loads Placed Arbitrarily

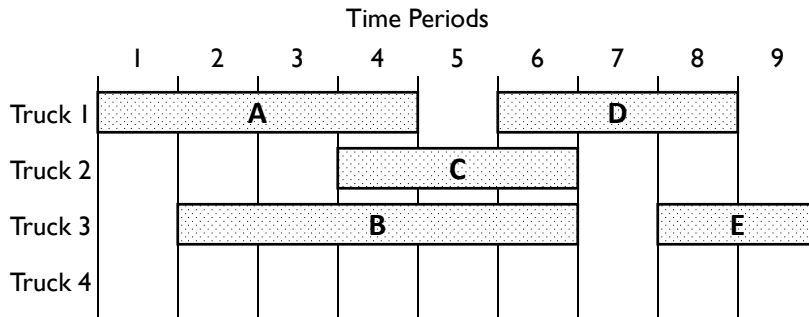


Figure 4: Loads Placed Optimally

While a carrier can improve fleet utilization through optimal assignment of trucks to loads, it is possible to achieve further improvement by shifting load start times. By offering price discounts, as it is often done in the airline industry, the carrier can induce the customer to accept an alternate delivery time. The carrier can use the savings obtained from better asset utilization to fund the price discounts. As Figure 5 illustrates for the earlier example, shifting load B one time period earlier and shifting load C one time period later reduces the number of needed trucks to two.

Because of the conditions of the problem — no routing consideration and all trucks have capacity one — we have essentially a machine scheduling model. Assigning loads to start times and resources

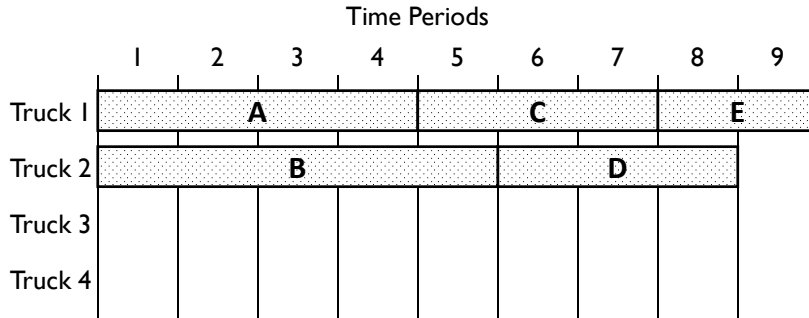


Figure 5: Loads Placed Optimally After Shifting Start Times

is common in the machine scheduling literature, though in that context loads are jobs and trucks are machines. Hence, throughout the remaining sections we adopt the terminology used in the machine scheduling literature.

While some machine scheduling papers address aspects of this problem, much of the literature focuses on job-related performance metrics such as maximum or average tardiness, and maximum or average completion time. By way of contrast, machine-related metrics focus on the number of machines required to perform a set of tasks or the maximum number of machines in use at any time or the average machine utilization. In this paper we address a machine scheduling problem that includes both job-related and machine-related metrics.

In the machine scheduling literature we typically either see the objective of minimizing the resources needed to complete all jobs by their requested times or the objective of minimizing the total lateness and/or tardiness of all jobs. In this paper, we present an optimization model with two objectives: the first is to minimize the number of machines needed to complete all jobs and the second is to minimize the incentives needed to influence customers to accept alternate completion times. The model can be solved quickly for reasonably large problems, and can be used to create real-time schedules for a trucking company. Combining these two objectives adds a new problem to the machine scheduling literature as well as to the trucking literature. In the delivery context, the delivery times are flexible but we recognize that there are costs associated with delivering a shipment early or late.

In summary, this paper contributes the following to the machine scheduling and transportation literature:

- We show that certain trucking problems can be solved as machine scheduling problems.
- We give a new multi-objective framework for solving machine scheduling problems — that of minimizing the number of machines needed as well as the costs associated with tardiness or earliness.

- We illustrate the tradeoff between these two objectives and show how the savings from reductions in machine or fleet usage can translate into incentives to encourage customers to accept alternative job completion or delivery times.

In Section 2, we review related work. In Section 3, we describe the mathematical models for scheduling and vehicle assignment problems. In Section 4, we discuss model properties. In Section 5, we present computational results. In Section 6, we give concluding remarks and suggest future research.

## 2 Literature review

The machine scheduling literature spans many different scenarios and problem types. Because the problem discussed in this paper assumes that machines are identical and that jobs are processed in parallel, we refer the reader to Cheng and Sin (1990) and Mokotoff (2001), which provide comprehensive surveys on parallel machine scheduling, and Kovalyov et al. (2007) and Kolen et al. (2007) for surveys on fixed interval scheduling. Discussions of general issues on scheduling theory can be found in Pinedo (1995) and Parker (1995).

We highlight the literature on the following problems: job scheduling with a fixed start time and processing time but with time windows (ranging from total flexibility to no flexibility) and with objectives of (1) minimizing the number of machines that accommodate all jobs and (2) minimizing total lateness or earliness of jobs.

Many authors have studied the problem of determining the minimum number of machines that can process a given set of jobs which must be scheduled within various time windows. Gertsbakh and Stern (1978) considered a job-scheduling problem when the desired starting times are either fixed or variable. When they are fixed it is called the fixed job schedule problem (FSP). When the starting times are variable it is called the variable job schedule problem (VSP). They provided an approximation algorithm for the VSP.

The problems addressed in Fischetti et al. (1987), Kroon et al. (1997), Garcia and Lozano (2005), and Ceder (2005) are variants of the problem in Gertsbakh and Stern (1978). Fischetti et al. (1987) and Fischetti et al. (1992) considered the bus driver scheduling problem (BDSP) where the goal is to find the minimum number of drivers to cover the bus schedule with restrictions on working hours. They proposed a polynomial algorithm for the BDSP. Kroon et al. (1997) considered non-identical machine scheduling when jobs belong to job classes and each machine can handle only jobs from a subset of the job classes. This problem often arises in gate assignment at airport terminals. If the number of job classes is fixed, then the problem is polynomially solvable; otherwise, the problem is NP-hard. The FSP addressed in Gertsbakh and Stern (1978) is a special case of this problem. Rojanasoonthon and Bard (2005) studied non-identical machine scheduling including priority among job classes. Garcia and Lozano (2005) considered the production and delivery scheduling problem with time windows as a variant of VSP. Ceder (2005) applied possible shifts in departure times using a technique based on a step function called a deficit function, and a trip insertion heuristic to reduce the fleet size.

When the starting times have total flexibility, meaning that a job can start at any point in time during which the machines are available, the problem of assigning jobs to machines with minimal number of machines is the bin packing (BP) problem, which is known to be NP-hard (see Garey

and Johnson (1979)). The BP problem has also been applied to multiple machine scheduling (see Coffman et al. (1978) and de Carvalho (1999)). In the context of vehicle routing, Bodin et al. (2000) studied the problem of minimizing the total travel time and the number of vehicles needed to provide the desired service. They showed that when loads begin and end at a depot (as in the problem in this paper) the problem can be formulated as a BP problem.

The studies mentioned above treat time windows in the constraints. There are also studies that handle them in the objective function through earliness and tardiness penalties. This type of problem is often referred as to the Earliness-Tardiness (ET) problem in scheduling. Various ET problems with assumptions on the due date, assignment procedure, and penalty function are reviewed in Baker and Scudder (1990).

In this paper one of the objectives in the multi-objective model shown in Section 3.2 is to minimize the total penalties when each job has a given desired start/completion time. Zhu and Heady (2000) presented a mixed integer programming formulation for minimizing job earliness and tardiness in a nonidentical multi-machine scheduling where setup times are sequence dependent, and due dates and penalties differ for each job. It differs from the problem presented in this paper because we do not assume that the number of available machines is as given; in fact, minimizing machines is the second objective in the multi-objective model of Section 3.2.

To the best of our knowledge, there is no literature dealing with the problem with the objective of saving resources as well as minimizing the total penalties for deviating from the desired starting times or completion times.

### 3 Problem Description and Formulation

Consider the following job scheduling problem. Let  $\mathcal{I}$  denote a set of jobs to be processed by identical machines. As in the fixed job scheduling problems (FSP), each job  $i$  has a fixed starting time  $s_i$  and a processing time  $p_i$ . A machine can process, at most, one job at a time.

In Section 3.1, we show an integer programming formulation for job scheduling when the start times for a given job are completely flexible. The goal is to assign jobs to machines and set start times using as few machines as possible. In Section 3.2 we introduce different levels of starting-time flexibility for different jobs with associated penalties. Thus, the objective in this problem is to minimize both the number of machines and the total penalties.

#### 3.1 Total flexibility in starting times, no penalty

In this section, we present an initial model which creates a schedule using as few machines as possible given total flexibility in start times and no penalties for tardiness or earliness. The inputs are the set of jobs and their processing times. The model gives a lower bound on the number of machines needed when flexibility is limited.

##### INPUTS

$\mathcal{I}$  = set of jobs

$\mathcal{J}$  = set of machines

$\tau$  = total length of time during which each machine is available

$p_i$  = processing time of job  $i$

#### DECISION VARIABLES

$X_{ij} = 1$  if machine  $j$  is assigned to job  $i$ , 0 otherwise.

$V_j = 1$  if machine  $j$  is used, 0 otherwise.

$$\min \sum_{j \in J} V_j \tag{1a}$$

subject to

$$\sum_{j \in J} X_{ij} \geq 1 \quad \forall i \in \mathcal{I} \tag{1b}$$

$$\sum_{i \in \mathcal{I}} p_i X_{ij} \leq \tau V_j \quad \forall j \in \mathcal{J} \tag{1c}$$

$$X_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I}; \forall j \in \mathcal{J} \tag{1d}$$

$$V_j \in \{0, 1\} \quad \forall j \in \mathcal{J} \tag{1e}$$

We assume that each job can start at any point in time. The objective function (1a) minimizes the number of machines. Constraint (1b) requires that each job be assigned to at least one machine. Given the objective of minimizing the number of machine, this will hold at equality. Constraint (1c) ensures that each machine is used for no more than  $\tau$  units.

We have jobs of different processing times which can be processed in any order on identical machines with the goal of minimizing the number of machines needed to finish the jobs during length of time  $\tau$ . This is the bin packing problem, which is well known to be NP-hard. In the bin packing problem, objects of different sizes are packed into bins of a fixed capacity in a way that minimizes the number of bins. Approximation algorithms as well as exact solution methods which are useful for small instances have been studied (see Coffman et al. (1997) and Vanderbeck (1999)).

### 3.2 Limited flexibility in starting times, with penalties

In this section we present a model which introduces a penalty for scheduling a load at a time other than the requested completion time (or starting time). The total flexibility problem in the previous section models scheduling decisions in continuous time. The limited flexibility problem presented below models decisions in discrete time. Penalties are therefore charged for schedule deviations in fixed increments of time. Lateness and earliness are often defined in terms of a threshold, such as 30 minutes late, or one hour early. In addition, it is advantageous from a modeling perspective, as we shall see below. There is some flexibility in choosing a proper discretization; however, the following assumption must be satisfied:

**Assumption 1** *The discretization must be such that all processing times and starting times are multiples of the width of the time window.*

We introduce flexibility with respect to the starting time of job  $i \in \mathcal{I}$  in the form of a set of possible starting times denoted by  $\mathcal{T}_i$ . Let each starting time have an associated penalty,  $c_{ik}$ . For example, job  $a$  has a requested fixed starting time  $s_a$  with 5 possible times; i.e.,  $|\mathcal{T}_a| = 5$ ,  $\mathcal{T}_a = \{s_a + \delta_k, k = 0, \dots, 4\}$  where  $\delta_0 = 0$ , and  $\delta_k$  for  $k = 1, \dots, 4$  can be positive or negative. No penalty is imposed when the job starts at  $s_a$  (i.e.,  $c_{a0} = 0$   $c_{ik} \geq 0$  for  $k = 1, \dots, 4$ ). In this example, the customer accepts 5 different time slots for delivery. Generally, different customers have different operational flexibilities. Moreover, customers can have different flexibilities over different time periods which is captured by modeling flexibility by job rather than by customer—the composition and size of  $\mathcal{T}_i$  varies by job. For example, a customer may require the first job of the day to be completed at a specific time (with no flexibility) but may be quite flexible regarding the completion times of subsequent jobs during the day.

The following models provide a schedule in two steps. The phase 1 model determines the number of required machines and the starting times of each job. Given those starting times and the number of machines, the phase 2 model assigns jobs to machines.

Our goal in phase 1 is to minimize both the number of machines and the total penalty of assigning jobs to different times. The model computes the number of machines needed in each time period without explicitly assigning jobs to machines, unlike the total flexibility model described in Section 3.1.

### Phase 1 IP formulation

#### INPUTS

$\mathcal{T}$  = set of time periods

$\mathcal{T}_i$  = set of possible times for starting job  $i$

$a_{ikt} = 1$  if job  $i$  which starts at time  $k \in \mathcal{T}_i$  is active at time  $t$

$c_{ik}$  = cost of starting job  $i$  at time  $k \in \mathcal{T}_i$

#### DECISION VARIABLES

$Y_{ik} = 1$  if job  $i$  starts at time  $k \in \mathcal{T}_i$ , 0 otherwise

$V$  = number of machines needed

$$\min V \tag{2a}$$

$$\min \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{T}_i} c_{ik} Y_{ik} \tag{2b}$$

subject to

$$\sum_{k \in \mathcal{T}_i} Y_{ik} \geq 1 \quad \forall i \in \mathcal{I} \tag{2c}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{T}_i} a_{ikt} Y_{ik} - V \leq 0 \quad \forall t \in \mathcal{T} \tag{2d}$$

$$Y_{ik} \in \{0, 1\} \quad \forall i \in \mathcal{I}; \forall k \in \mathcal{T}_i \tag{2e}$$

The objective function (2a) minimizes the number of machines needed. The objective function (2b) minimizes the total cost of assigning jobs to different times. Constraints (2c) state that each job must be assigned to a starting time in the eligible set of times for that job. Constraints (2d) counts, for each time period in the set  $\mathcal{T}$ , the number of assigned jobs. The value of  $V$  must exceed this summation for every time period. Constraints (2e) ensure the decision variables  $Y_{ik}$  are binary.

The model with two objectives, (2a) and (2b), can be solved by the constraint method or the weighting method as explained in Cohon (1978). In the constraint method, we use objective (2b) alone with the following constraint,

$$V \leq v^{\max} \quad (2f)$$

where  $v^{\max}$  is an input, the maximum number of machines allowed. This sets an upper bound on the number of utilized machines. In the weighting method, we combine (2a) and (2b) into one objective of minimizing a weighted sum of the number of machines needed and the cost of assigning jobs to different times as follows,

$$\min \alpha V + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{T}_i} c_{ik} Y_{ik} \quad (2g)$$

where  $\alpha$  is a weight on minimizing the number of machines. We use the constraint method for the computational results in Section 5.

It is worth noting that the definition of  $a_{ikt}$  allows the duration of a job to depend on the selected starting time. This is important in trucking operations as deliveries scheduled for peak traffic hours may take longer than deliveries made at other times of the day. In our computational results discussed in Section 5, we assumed that the duration of each job was independent of the job's starting time.

As shown in Section 4, the phase 1 model with partial and total flexibility is NP-complete. For large instances, it requires the development of specialized algorithms. For smaller instances, a standard algorithm such as branch and bound may suffice.

Given the starting times and the number of machines obtained from the phase 1, the phase 2 model assigns jobs to machines.

## Phase 2 IP formulation

### INPUTS

$\hat{a}_{it} = 1$  if job  $i$  is active at time  $t$ ; 0 otherwise (note that  $\hat{a}_{it} = \sum_{k \in \mathcal{T}_i} a_{ikt} Y_{ik}$ )

$\hat{V}$ , as determined in phase 1

### DECISION VARIABLES

$V_j$  and  $X_{ij}$  as in Model (1)

$$\min \sum_{j \in \mathcal{J}} V_j - \hat{V} \quad (3a)$$

subject to

$$\sum_{i \in \mathcal{I}} \hat{a}_{it} X_{ij} \leq V_j \quad \forall t \in \mathcal{T}, \quad \forall j \in \mathcal{J} \quad (3b)$$

$$\sum_{j \in \mathcal{J}} X_{ij} = 1 \quad \forall i \in \mathcal{I} \quad (3c)$$

$$X_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I}; \forall j \in \mathcal{J} \quad (3d)$$

$$V_j \in \{0, 1\} \quad \forall j \in \mathcal{J} \quad (3e)$$

The objective function (3a) minimizes the deviation of the total number of machines assigned from the number of machines determined in phase 1. Constraints (3b) ensure, for each time period in the finite set  $\mathcal{T}$  and for each machine, that a job cannot be assigned to a machine unless the machine is used. Constraints (3c) state that each job must be assigned to one machine. Constraints (3d) and (3e) state that the decision variables are binary.

Rather than solving the phase 2 model as a mathematical program, the greedy algorithm below can be used to assign scheduled jobs to machines. The advantage of doing so is that the greedy algorithm exploits the structure of the problem to find an exact solution. As a result, the algorithm has low complexity as we shall see in Section 4.

## Greedy algorithm

Notation

- $\tilde{\mathcal{I}}$ : set of unassigned jobs
- $\hat{s}_i$ : starting time of job  $i$  determined in phase 1
- $m$ : machine number
- $t_m$ : next available starting time for machine  $m$

**step 0** Set  $m = 1$ ,  $t_m = 0$ ,  $\tilde{\mathcal{I}} = \mathcal{I}$ . Sort  $\tilde{\mathcal{I}}$  in nondecreasing order of starting time.

**step 1** If  $\tilde{\mathcal{I}} = \emptyset$ , stop. Otherwise, let  $i_o$  be the first job in  $\tilde{\mathcal{I}}$ . Go through the machines available times  $t_1, \dots, t_m$  and assign the job  $i_o$  to the first machine such that  $\hat{s}_{i_o} \geq t_m$ . Let  $\tilde{m}$  denote the machine. Set  $t_{\tilde{m}} = \hat{s}_{i_o} + p_{i_o}$ .

**step 2** If there are no machines such that  $\hat{s}_{i_o} \geq t_m$ , set  $m = m + 1$ , and set  $t_m = \hat{s}_{i_o} + p_{i_o}$ .

**step 3** Update  $\tilde{\mathcal{I}}$ , i.e.,  $\tilde{\mathcal{I}} = \tilde{\mathcal{I}} \setminus i_o$ . Then go to step 1.

## 4 Model properties

In this section we discuss properties of the phase 1 and 2 models introduced in Section 3.

**Theorem 1** *The complexity of the phase 1 model in terms of flexibility is the following:*

- (1) *With total flexibility, the phase 1 problem is NP-complete.*
- (2) *With partial flexibility, the phase 1 problem is NP-complete.*
- (3) *With no flexibility, the phase 1 problem is polynomially solvable.*

**Proof**

(1): We show that the partition problem can be polynomially reduced to the phase 1 problem. The partition problem decides whether a given set of integers can be partitioned into two halves that have the same sum. Let  $J = \{p_1, \dots, p_n\}$  where  $p_i, i = 1, \dots, n$  are integers. Let  $\tilde{J} = \{2p_1, \dots, 2p_n\}$ . It is easy to see that solving the partition problem on  $\tilde{J}$  is the same as solving the partition problem on  $J$ . We consider the partition problem on  $\tilde{J}$ . Consider the IP formulation of the weighting method of the phase 1. Let  $\mathcal{T}_i = \mathcal{T}$ ;  $c_{ik} = 0, \forall i, k$ ;  $\alpha = 1$ ; and  $\tau = \sum_{i=1}^n p_i$ . If  $\hat{V} = 2$  from the phase 1 model, we can answer ‘yes’ to the partition problem. Since the partition problem is NP-complete, the phase 1 model with total flexibility is also NP-complete. (2): It is immediate since the model with total flexibility is a particular case of partial flexibility ( $c_{ik} = 0 \forall i, k$ ). (3): When there is no flexibility, solving the phase 1 model is the same as solving a coloring problem in the interval graph. This idea has been also discussed in Fischetti et al. (1987). Let  $G(V, E)$  be the graph with  $V$  the set of vertices and  $E$  the set of edges. Let  $V$  be the jobs. Link  $e_{ij}$  will exist between vertices  $i$  and  $j$  if processing of job  $i$  overlaps with job  $j$ . A coloring of  $G$  generates a machine assignment with the minimal number of machines which is the same as the number of colors needed. The coloring problem in the interval graph can be solved in polynomial time since interval graphs are perfect graphs (for definitions see Golubic (1980)).  $\square$

We now analyze the ability of the greedy algorithm to obtain the optimal and complexity of the greedy algorithm in phase 2. The theorems below demonstrate the efficacy of the method.

**Theorem 2** *The greedy algorithm presented in Section 3.2 yields the minimum number of machines as in phase 1.*

**Proof**

Recall that phase 1 yields the minimum number of needed machines (denoted  $\hat{V}$ ) as well as the optimal starting times of each job. Let  $\tilde{v}$  be the number of machines given by the greedy algorithm. We claim that  $\tilde{v} = \hat{V}$ . Clearly  $\tilde{v} \geq \hat{V}$ , and suppose that  $\tilde{v} > \hat{V}$ . Consider the first job assigned to machine  $\tilde{v}$ ; call it  $j$ . By the construction of the greedy algorithm, we know that at the start of job  $j$ ’s processing, each of the machines  $1, \dots, \tilde{v} - 1$  must be processing other jobs — otherwise, if this were not true for some machine  $m \in \{1, \dots, \tilde{v} - 1\}$ , then job  $j$  would have been assigned to machine  $m$ . In other words, the optimal schedule yielded by phase 1 requires that at least  $\tilde{v} > \hat{V}$  jobs be processed simultaneously, which leads to a contradiction since  $\hat{V}$  is the minimum number of machines needed to process all jobs. Thus,  $\tilde{v} = \hat{V}$ .  $\square$

**Theorem 3** *Recall that  $|\mathcal{I}|$  is the number of jobs to be scheduled and  $\hat{V}$  is the number of machines returned by phase 1. The greedy algorithm is a polynomial-time algorithm with complexity  $O(\min\{|\mathcal{I}| \log |\mathcal{I}|, |\mathcal{I}| \hat{V}\})$ .*

**Proof**

The algorithm sorts  $|\mathcal{I}|$  jobs at step 0. Then, beginning with the earliest starting time in  $\tilde{\mathcal{I}}$ , the algorithm compares each starting time with the available time  $t_m$  of the current machines  $m = 1, \dots, \hat{V}$ , and stops when a machine is found such that  $\hat{s}_i \geq t_m$ . The algorithm repeats

until there are no unassigned jobs. Therefore, the algorithm requires  $|\mathcal{I}|\hat{V}$  computations in worst case. Given that worst case complexity of the sorting algorithm is  $O(|\mathcal{I}|\log|\mathcal{I}|)$ , the computational complexity of the greedy algorithm is  $O(\max\{|\mathcal{I}|\log|\mathcal{I}|, |\mathcal{I}|\hat{V}\})$ .  $\square$

## 5 Case study

In this section we demonstrate how the phase 1 limited flexibility model can reduce fleet requirements. This analysis is based on proprietary data from a Chicago-based transportation carrier that have been altered slightly. The data represents a typical set of loads delivered over one week. There are 151 loads with processing times between 2 and 39 hours, with an average of 8.8 hours.

Time is discretized into 30 minute slots so that the phase 1 model has 336 time indexes over the course of the week. In keeping with Assumption 1, we chose a discretization so that all requested load start and completion times coincide with a time index. A more refined discretization of time would allow more accurate rounding of processing times but would increase the number of variables needed. Since the resulting problem was not very large we solve it using standard branch and bound.

Figure 6 shows a solution to the phase 1 limited flexibility model using only one value, the requested start time, in the sets of possible start times,  $\mathcal{T}_i$ , for each load. This corresponds to a ‘no flexibility’ solution. As the figure shows, there are significant peaks in the vehicle usage throughout the week, particularly during the early morning hours of Tuesday and Wednesday. To deliver all loads during the peak volume time period, the minimum number of needed vehicles is 27.

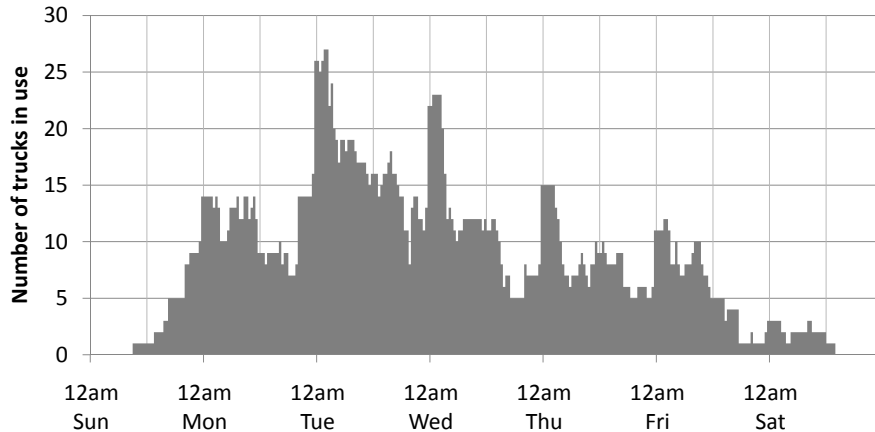


Figure 6: No Flexibility Solution - 27 Trucks, 0 Penalty

Equation (4) illustrates how we computed the cost  $c_{ik}$  of starting each load  $i$  at time  $k$  when load  $i$  is requested for delivery at time  $r_i$ .

$$c_{ik} = \alpha_i * \max(k - r_i, 0)^{\gamma_i} + \beta_i * \max(r_i - k, 0)^{\eta_i} \quad (4)$$

The first term corresponds to lateness costs and the second term corresponds to earliness costs. Each load corresponds to a customer and we have based  $\alpha_i$  and  $\beta_i$  on the size of the customer initiating the load request. By doing so, the model prioritizes shifting the start time of some customers' loads over others. The parameters  $\gamma_i$  and  $\eta_i$  make the penalties nonlinear, so we can more severely penalize increasingly late or early loads.

Figure 7 shows the results of solving phase 1 of the limited flexibility model using the constraint method with different values of  $v^{\max}$ . The curve illustrates the tradeoff between the shifting penalties and reducing the fleet size. Figure 8 shows the utilization profile at several points on the tradeoff curve. Each utilization profile contains a picture of the tradeoff curve that identifies the corresponding point on the tradeoff curve. Solving the model with 151 loads took less than one second each time.

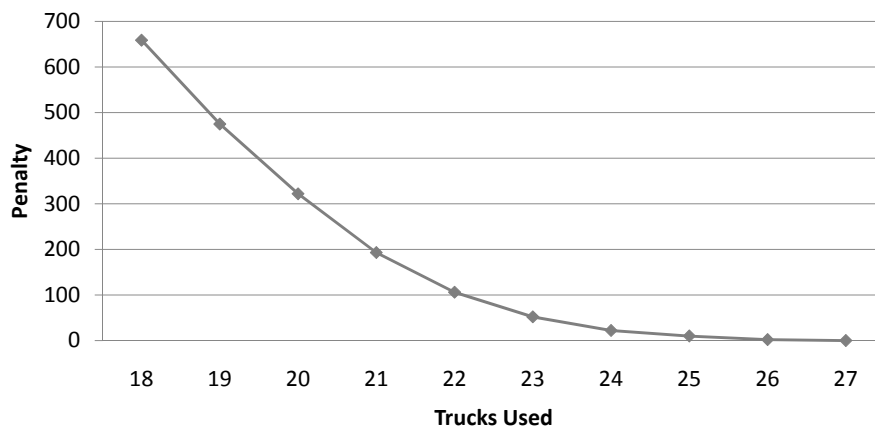


Figure 7: Cost of reducing fleet size

It is worth noting that the limited flexibility model increases the utilization of many, if not all, of the vehicles in use. If a truck is delayed in traffic and returns much later than the expected delivery time, this will cause serious scheduling disruptions. Because the processing times in the phase 1 model are deterministic, conservative quantiles of the delivery time distributions for each customer can be used rather than expected values.

## 6 Conclusions and Future Study

The models presented here are useful for solving problems in two contexts: machine scheduling and trucking. This paper presents a trucking problem which, under given operating conditions, becomes a machine scheduling problem. The machine scheduling literature is extensive, covering many varieties of problems. In this paper, we consider identical parallel machines, a window of acceptable delivery times, costs associated with earliness and tardiness within the time window, and two objectives: minimizing the number of needed machines and minimizing the costs of earliness and tardiness.

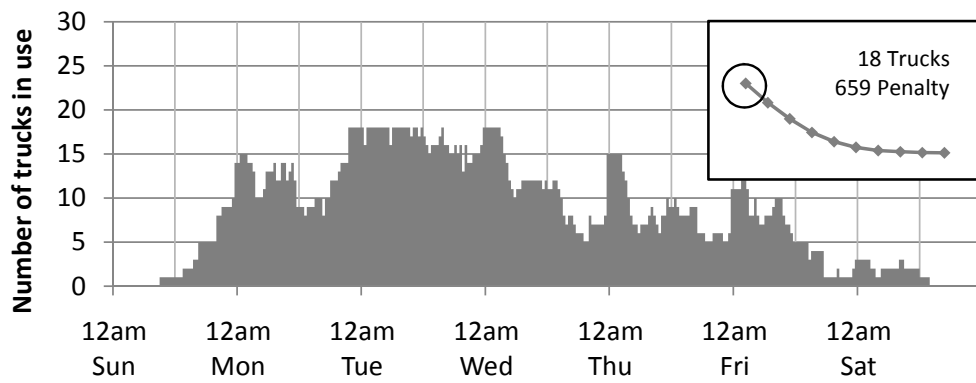
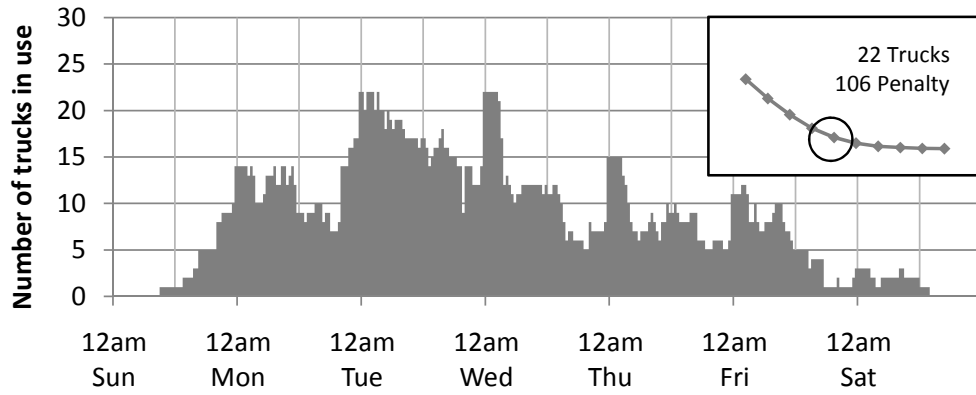
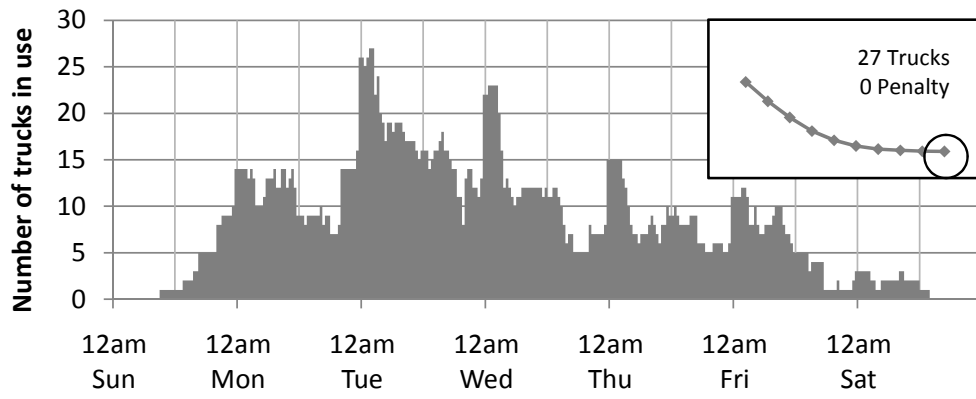


Figure 8: 3 Points on the Tradeoff Curve

The completion times of automated tasks have much lower variability in the machine scheduling context than the delivery times do in the transportation context. The use of delivery time windows acknowledges that a load delivered five minutes early or late is often considered on time, but delivering one hour late, even if the customer is alerted in advance, may have a cost. The discrete time model presented here relates the costs of deliberately delivering loads early or late to a reduction in the carrier’s fleet size.

In the machine scheduling context, penalties for early and late completion times of jobs are incurred for violating customers’ preferences. A carrier that deliberately delivers loads early or late can achieve cost reductions, however, through improvements in fleet utilization. The cost reductions can be passed on to customers to persuade them to accept early and late delivery.

The carrier on which the case study is based experiences a peak number of shipments during certain hours. Some of the loads requested during this peak are requested because of the customers’ inventory replenishment policies. Many other customers, however, request delivery times arbitrarily. The shifting costs in our limited flexibility model reflect the flexibility or rigidity of customers, as well as how important they are to the carrier’s business. The carrier can then reduce its fleet size by making less costly adjustments to its schedule, and pass on savings to customers who are willing to accept early or late delivery.

We have illustrated the relationship between earliness and tardiness costs and machine or truck resource reductions in Figure 7. Using a constraint method and a sample set of data, a carrier can choose an operating point along a tradeoff like this. In all our experiments, which included as many as 151 loads, phase 1 of the limited flexibility model ran in less than one second. After loads have been assigned a start time, the greedy method, which we proved yields the same number of machines or trucks as the number determined by the phase 1 model and which we showed runs in polynomial time, can be used to assign jobs to machines or loads to trucks.

The limited flexibility model presented has many benefits. Although it is NP-complete, branch-and-bound procedures can solve it quickly at least for the size of problems encountered by the carrier that motivated this research. The model also applies to machine scheduling problems as well as transportation problems, it captures the relationship between earliness/tardiness and resource usage, and it enables a carrier to influence the behavior of its customers. It can be improved, however. The model can only be applied to limited transportation problems because routing is not included; loads must go out and back without serving additional demand. Although it was based on a real industry example, this assumption is unlikely to hold for many other carriers. Adding routing to our model will add to its complexity, but heuristic solution approaches to routing problems with delivery time windows at each demand point may be able to solve a very difficult real world problem.

The models also assumes that all loads are known in advance. In reality, at a given time before the delivery date, there are three types of loads: loads that have been previously assigned a delivery time, loads which are newly requested but which have not been assigned a delivery time, and loads that can be anticipated. The model we present can be used on the first two load types. In our current research, we are developing a model that includes anticipated loads as well.

Finally, we note that job processing times in the transportation context are stochastic due to the uncertainty of traffic and loading and unloading times. Therefore it will be important to extend our models to incorporate this form of uncertainty as well as the uncertainty associated with future job requests. These improvements can help trucking companies and others who might benefit from

the machine scheduling problem structure realize greater savings through better resource utilization caused by improved customer behavior.

## References

- K. R. Baker and G. D. Scudder. Sequencing with earliness and tardiness penalties - a review. *Operations Research*, 38(1):22–36, 1990.
- L. Bodin, A. Mingozzi, R. Baldacci, and M. Ball. The rollon-rolloff vehicle routing problem. *Transportation Science*, 34(3):271–288, 2000.
- A. Ceder. Estimation of fleet size for variable bus schedules. *Transit: Bus, Rural Public and Intercity, and Paratransit*, (1903):3–10, 2005.
- T. C. E. Cheng and C. C. S. Sin. A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3):271–292, 1990.
- E. G. Coffman, M. R. Garey, and D. S. Johnson. Application of bin-packing to multiprocessor scheduling. *Siam Journal on Computing*, 7(1):1–17, 1978.
- E. G. Coffman, M. R. Garey, and D. S. Johnson. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, MA, USA, 1997.
- J. L. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York, 1978.
- J. M. V. de Carvalho. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, 86:629–659, 1999.
- M. Fischetti, S. Martello, and P. Toth. The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6):849–858, 1987.
- M. Fischetti, S. Martello, and P. Toth. Approximation algorithms for fixed job schedule problems. *Operations Research*, 40:S96–S108, 1992.
- J. M. Garcia and S. Lozano. Production and delivery scheduling problem with time windows. *Computers and Industrial Engineering*, 48(4):733–742, 2005.
- M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. WH Freeman and Company, New York, 1979.
- I. Gertsbakh and H. I. Stern. Minimal resources for fixed and variable job schedules. *Operations Research*, 26(1):68–85, 1978.
- M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
- A. W. J. Kolen, J. K. Lenstra, C. H. Papadimitriou, and F. C. R. Spijksma. Interval scheduling: A survey. *Naval Research Logistics*, 54(5):530–543, 2007.

- M. Y. Kovalyov, C. T. Ng, and T. C. E. Cheng. Fixed interval scheduling: Models, applications, computational complexity and algorithms. *European Journal of Operational Research*, 178(2): 331–342, 2007.
- L. G. Kroon, M. Salomon, and L. N. VanWassenhove. Exact and approximation algorithms for the tactical fixed interval scheduling problem. *Operations Research*, 45(4):624–638, 1997.
- E. Mokotoff. Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research*, 18(2):193–242, 2001.
- R. G. Parker. *Deterministic Scheduling Theory*. Chapman & Hall, London, 1995.
- M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
- S. Rojanasoonthon and J. Bard. A grasp for parallel machine scheduling with time windows. *Inform Journal on Computing*, 17(1):32–51, 2005.
- World Oil Market Chronology. 2003 to 2008 World Oil Market Chronology. 2008. URL [http://en.wikipedia.org/wiki/2003\\_to\\_2008\\_world\\_oil\\_market\\_chronology](http://en.wikipedia.org/wiki/2003_to_2008_world_oil_market_chronology).
- F. Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86(3):565–594, 1999.
- Z. W. Zhu and R. B. Heady. Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers and Industrial Engineering*, 38(2):297–305, 2000.