Short-Term Inventory-Aware Equipment Management in Service Networks

Yassine Ridouane[1], Natashia Boland[1], Alan Erera[1], and Martin Savelsbergh[1]

[1]*H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Dr NW, Atlanta, GA 30332*

yridouane@gatech.edu

{alan.erera, natashia.boland, martin.savelsbergh}@isye.gatech.edu

Abstract

Logistics companies often operate a heterogeneous fleet of equipment to support their service network operations. This introduces a layer of planning complexity as facilities need to maintain appropriate levels of equipment types to support operations throughout the planning horizon. We formulate an optimization model that minimizes the cost of executing a load plan, assuming knowledge of the trailer inventory distribution in the network at the start of the planning horizon, by possibly substituting the equipment type assigned to loaded movements and by judiciously adding empty equipment repositioning movements. We introduce an integer programming based heuristic, which heavily relies on dynamic variable generation, for its solution. Computational experiments using instances from a major US package express carrier show the efficacy of the solution approach and show the benefit of an optimization-based approach to inventory-aware equipment management: a significant reduction in the cost of empty equipment repositioning movements to avoid equipment shortages (if possible).

Keywords Transportation · Operational planning · Fleet management · Empty repositioning · Package carrier · Column generation

1 Introduction

The major players in the courier and package delivery industry operate very large ground service networks. For instance, in the United States, the UPS small package network has more than 1,800 operating facilities where parcels are processed, and operates a multi-type delivery fleet comprised of 125 thousand vehicles[1]. Similarly, FedEx Ground has more than 600 operating facilities and

[1]UPS Fact Sheet 2020

operates more than 70 thousand vehicles[2]. With the continued growth of e-commerce, the service networks of these carriers are expected to further expand as the demand for parcel delivery continues to increase. The penetration of the internet and the shift of customer shopping trends towards online marketplaces contribute significantly to this increase.

Some of the e-commerce players, such as Amazon.com, have started investing in their own multi-modal package delivery capability, which spans first mile, middle mile, and last mile logistics. This strategic step towards in-house shipping enables these companies to optimize their supply chain from supply sources to fulfillment centers and from fulfillment centers all the way to the doorsteps of their customers. It also reduces their reliance on third party logistics companies and gives more control over their shipping expenditure. This development has forced the existing logistics companies to improve their operational efficiency and to reduce their transportation costs so as to remain competitive and not lose their market share. Achieving operational efficiency includes, among others, the active management of the fleets of equipment types employed in the daily service operations. This is the focus of the research presented in this paper.

Operating a large ground service network involves, among others, ensuring that the right equipment is available at the right time at the right location. A fleet of different types of trailers and containers is used to transport freight between different locations. As demand is naturally imbalanced between regions, some facilities in the network will see more inbound than outbound trailers possibly leading to a build up of trailers that can exceed the facility capacity. Other facilities will see more outbound than inbound trailers possibly leading to equipment stock-outs and delays in executing planned freight movements. Having a heterogeneous fleet of equipment increases the complexity of equipment management as it destroys the self-balancing nature of driver circulations in the network, e.g., a driver can transport a 53-foot trailer from one location to another, but then return with two 28-foot trailers. Hours of service and union regulations may further complicate matters as it can result in (undesirable) bobtail movements, i.e., movements where a driver returns to his domicile in a tractor without pulling any trailer(s). To address equipment surplus or shortage at facilities, carriers reposition equipment – even using one-way rail movements – and lease equipment for short periods of time, all coming at a significant cost.

Effective equipment management requires short-term strategies to react to imbalances in the network as soon as they can be foreseen and long-term strategies that preemptively and proactively place equipment where it will likely be needed based on a demand forecast. At a long-term, tactical level where the planning horizon can span several months, a carrier focuses on equipment fleet size, e.g., whether to expand or shrink the fleet, and redistributing the fleet across the network to prepare for the future, e.g., the peak season, based on a demand forecast. Equipment leasing and procurement decisions are made at this level. These long-term tactical decisions are gener-

[2]FedEx Fact Sheet 2020

ally made infrequently (annually or bi-annually for major carriers). At a short-term, operational level where the planning horizon covers a few days, a carrier focuses on satisfying planned loads (planned movements) that are expected to be executed with high confidence in the upcoming days at least cost, possibly with equipment inventory level targets at facilities at the end of the planning horizon. In this case, accurate information about equipment inventory at facilities and in-transit equipment at the start of the planning period is critical. Short-term management of a fleet of multiple substitutable equipment types is the focus of our research.

The contributions of our research can be summarized as follows:

- We formulate a short-term inventory-aware ground equipment management problem. The input is a load plan and information about equipment inventory at facilities and in-transit equipment, and the output is a minimum cost assignment of equipment types to loaded movements and empty equipment repositioning movements;
- We introduce a time-expanded network formulation for the problem and propose a parsimonious time discretization scheme to control the size of the formulation;
- We develop an efficient and effective heuristic, which involves dynamically generating variables, for the solution of the formulation;
- We conduct an extensive computational study using large-scale instances provided by a major US carrier to assess the benefits of short-term inventory-aware ground equipment management and the efficacy of the proposed heuristic.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss relevant literature. In Section 3, we present a description of the problem and introduce a mixed integer programming formulation. In Section 4, we develop an efficient and effective heuristic for producing high-quality solutions. In Section 5, we give a summary of the results of an extensive computational study to assess the value of inventory-aware equipment management and the performance of the heuristic. Finally, in Section 6, we discuss future research directions.

2 Relevant literature

Equipment management in the trucking industry has been investigated from different perspectives in the literature. Fleet sizing, empty repositioning, and inventory control have been studied in both freight consolidation networks and small package networks and for different types of equipment (e.g., tractors, containers, trailers, etc.). These aspects are inter-connected, but researchers have studied them in isolation as well as in an integrated manner. To the best of our knowledge, there is no prior literature on short-term trailer fleet management with multiple substitutable equipment types. As such management is only possible if accurate equipment inventory information is available, we refer to this problem as *short-term inventory-aware equipment management with multiple substitutable equipment types*. An early classification of empty equipment flow problems was presented by Dejax and Crainic [1987]. Multiple problem-defining characteristics are used, such as the type of flow (empty vs loaded movements), the transportation mode (single mode vs multi-mode), and the fleet homogeneity (single vs multiple substitutable equipment types). A more recent review of fleet planning problems [Baykasoğlu et al., 2019] introduces a multi-modal fleet planning framework with a classification scheme based on problem and modeling characteristics and decision making levels. Carbajal et al. [2013] analyze the relationship between fleet size and empty repositioning. Container planning in multi-modal transportation (especially rail and maritime modes) was studied by Crainic et al. [1993], Imai and Rivera IV [2001], Boile et al. [2008], and Chang et al. [2008]. Trailer repositioning which is critical in so-called ground networks has been investigated by Erera et al. [2009], Du and Hall [1997], and Hall [1999]. Dynamic vehicle allocation problem in Full-Truckload networks was studied by Vasco and Morabito [2016], and Çalışkan and Hall [2003]. Fleet heterogeneity was explored by Jabali et al. [2012] and Gould [1969]. Using equipment substitution to address equipment flow imbalance was studied by Yang et al. [2021] for ground transportation and by Chang et al. [2008] for maritime transportation; compatibility rules restrict the number and type of substitutions. Equipment heterogeneity naturally occurs in other industries as well. In the airline industry, for example, most major carriers (e.g., American Airlines and Delta Airlines) operate different types of aircraft in different markets. A few airline carriers opt for a homogeneous fleet to simplify their operations (e.g., Southwest). Rushmeier and Kontogiorgis [1997] considers a heterogeneous airline fleet assignment problem. In the car rental industry, operating a heterogeneous fleet is crucial to be able to meet different customer preferences and brings many operational challenges. Oliveira et al. [2017] surveys car rental literature and presents a conceptual framework of car rental fleet and revenue management.

Many equipment management problems can be modeled using time-expanded networks. However, the time-expanded networks quickly become prohibitively large and special solution techniques are required to solve them, e.g., column generation. Examples of such an approach include Brouer et al. [2011] who consider a liner shipping cargo allocation problem, and Çalışkan and Hall [2003] and Çalişkan and Hall [2006] who adopted column generation to solve the static and dynamic empty allocation models.

In the United States, the heterogeneity in equipment types employed in the ground networks of less-than-truckload and package express carriers is mainly due to size. The three main equipment types are *short equipment* (also referred to as *pups*) with a typical length of under 28 feet, *long equipment* with length ranging from 40 to 48 feet, and *extra long equipment* with a typical length of 53 feet. Employing different size trailers improves utilization, reduces handling, and increases direct loading opportunities. Moreover, as a tractor can pull a combination of short equipment (typically two pups, but even three pups in some states) or a combination of long and short equipment, this allows loads that are bound to different locations to share a part of their route thereby reducing the number of driver schedules needed to execute loads.

In the work we are presenting in this paper, we are focusing on the ground service networks that operate a heterogeneous fleet of trailers and containers over the road as opposed to other modes of transportation, such as maritime, rail, and air networks which are covered by Imai and Rivera IV [2001], Boile et al. [2008] and Rushmeier and Kontogiorgis [1997]. Similar to the works by Brouer et al. [2011] and Çalişkan and Hall [2006], we also use a time-expanded network and adopt a column-generation based approach to solve the large mixed-integer program. Moreover, we present a novel approach based on the specific structure of our problem to carefully introduce repositioning arcs in the time-expanded network in order to ensure high-quality solutions as well as computational tractability.

3 Problem description

We consider the short-term planning of a fleet comprised of different types of trailers and containers for ground service network of a package express carrier. We are given a *load plan* for the planning period, typically a week. A load plan is the result of a load planning process, that uses a demand forecast (and information on available resource types) to generate timed loads between pairs of locations in the network and a tentative driver schedule to execute the planned loads. The loads are of three types: *loaded*, *empty*, and *bobtail* movements. The empty and bobtail loads present an initial step towards balancing equipment flows in the network. Each load has an associated set of compatible configurations of equipment types that can be assigned to it. Whether one configuration can be substituted by another configuration depends on multiple criteria, such as the size of the equipment, the existence of a pintle for short equipment (required to create a train of trailers), the ability of a facility to handle such equipment types, etc. These criteria can be used to create a substitution matrix that summarizes all the allowable equipment substitutions. During the load planning process a tentative equipment configuration is assigned to each load in the load plan. This tentative assignment is based on recently executed load plans in the hope that few adjustments are needed to account for week-to-week demand changes.

In addition to the load plan, we are given a snapshot of the equipment in the network at the start of the planning period (represented by time 0). This includes the inventory of equipment at every facility at time 0 and the in-transit equipment, i.e., equipment assigned to loads that were dispatched in the past (before time 0) and are expected to reach their destination before the end of the planning (represented by T). The inventory of equipment at the facilities (e.g., in the yard, undergoing maintenance, at a dock being loaded or unloaded) and the equipment assigned to in-transit loads represents the fleet of equipment available to execute the load plan.

Because the primary focus of load planning is ensuring capacity is available to move forecast demand and balancing equipment flow is only secondary consideration, If the load plan is executed as is, i.e., without changing the equipment configurations assigned to the loads or introducing additional empty equipment repositioning movements, equipment stock-outs may occur, which can cause delays in the delivery of demand and may be costly to address at the time they occur. Our primary objective is to minimize the risk of equipment stock-out during the planning horizon (avoiding equipment stock-outs entirely is impossible because of unforeseen events that can happen during the planning period – equipment breakdowns, unexpected changes in demand, etc.) either by changing the equipment configuration assigned to loads or by introducing empty equipment repositioning movements. A secondary objective may be to ensure a minimum target inventory of equipment types at facilities at the end of the planning period.

We will formulate a time-expanded network model for the problem outlined above in which nodes represent facility-time pairs and arcs represent planned timed loads in the load plan or potential empty equipment repositioning movements.

Next, we summarize the notation that we adopt to describe the model and the proposed solution approach. After that, we present a mixed integer programming formulation for the problem.

3.1 Notation

The following parameters are used in the definition of the problem and its mixed integer programming formulation:

- $\mathcal{F}$: The set of facilities in the network.
- $\mathcal{E}$: The set of equipment types. These can differ by size, i.e, short (trailers with a length of less than or equal to 28 feet), long (trailers with a length ranging from 40 to 48 feet), and extra long (trailers with a length of 53 feet). They can also differ by utility, e.g., refrigerated or heated trailer, rail containers, etc.,
- $\mathcal{C}$: The set of equipment configurations. Each configuration is a vector representing a possible combination of units of equipment types in $\mathcal{E}$. Some configurations are only allowed in certain regions. For example, configurations containing three pups are allowed in only 13 states. Let η denote the configuration matrix where a row i represents a configuration c_i in $\mathcal{C}$ and a column j represents an equipment type e_j in $\mathcal{E}$, then an entry η_{ce} represents the number of units of equipment type e in configuration c. An example of η with three equipment types in

$\mathcal{E}$ and four configurations in $\mathcal{C}$ is shown below:

$$\eta = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

In this example, configuration c_2 represents a two-unit train of short equipment of type e_1.

- $\mathcal{L}$: The set of timed loads scheduled to dispatch within the planning period $[0, T]$. A load captures the time and the location where a trailer is to be loaded and the time and the location where it is to be unloaded. A load $l \in \mathcal{L}$ has the following attributes:
  - $o(l) \in \mathcal{F}$: The origin of load l.
  - $t^o(l) \in [0, T]$: The time at which equipment starts being loaded at the origin. Let $\mathcal{T}^o(i)$ denote the set of times at which equipment starts being loaded at facility i.
  - $d(l) \in \mathcal{F}$: The destination of load l.
  - $t^d(l) \in [0, T]$: The time at which equipment ends being unloaded at the destination. Let $\mathcal{T}^d(i)$ denote the set of times at which equipment ends being unloaded at facility i.
  - $q(l) \in \mathcal{C}$: The (initial) equipment configuration assigned to load l.
  - $S_l \subseteq \mathcal{C}$: The set of allowable configurations for load l.
- $\mathcal{N}$: The set of nodes in the time-expanded network. Each node (i, t) in $\mathcal{N}$ represents a facility $i \in \mathcal{F}$ and a time $t \in \mathcal{T}(i)$ with $\mathcal{T}(i)$ representing the set of times for facility i, i.e., $\mathcal{T}(i) = \{t : (i, t) \in \mathcal{N}\}$. The set $\mathcal{T}(i)$ contains times 0 and T and all other $t \in \mathcal{T}(i)$ have $0 < t < T$. Note that $\mathcal{T}(i)$ is not the union of $\mathcal{T}^o(i)$ and $\mathcal{T}^d(i)$. As our goal is to have a parsimonious discretization, $\mathcal{T}(i)$ may exclude times in $\mathcal{T}^o(i)$ and $\mathcal{T}^d(i)$ for facilities with high inbound and outbound activity and $\mathcal{T}(i)$ may include times other than in $\mathcal{T}^o(i)$ and $\mathcal{T}^d(i)$ for facilities with low inbound and outbound activity in order to create more repositioning opportunities.

  For convenience, for time point $t \in \mathcal{T}(i) \setminus \{0, T\}$, we let $t^- = \max\{s \in \mathcal{T}(i) \ : \ s < t\}$ be the preceding time point and $t^+ = \min\{s \in \mathcal{T}(i) \ : \ s > t\}$ be the succeeding time point (and, thus, (i, t^-) and (i, t^+) represent, respectively, the node preceding and the node succeeding (i, t)).

  We also define the sets $\mathcal{L}^-_{(i,t)}$ and $\mathcal{L}^+_{(i,t)}$ as the sets of inbound and outbound loads in $\mathcal{L}$

associated with node (i,t) in $\mathcal{N}$, respectively:

$$\mathcal{L}^-_{(i,t)} = \{l \in \mathcal{L} : \;\; d(l) = i,\, t^- \leq t^o(l) < t\},$$
$$\mathcal{L}^+_{(i,t)} = \{l \in \mathcal{L} : \;\; o(l) = i,\, t^- < t^d(l) \leq t\}.$$

$\mathcal{L}^-_{(i,t)}$ and $\mathcal{L}^+_{(i,t)}$ represent the set of loads that arrive to and depart from facility i, respectively, between the discrete time that precedes t - that we denoted as t^- - and t itself.
To illustrate this notation, we provide a small example with four loads l_1 to l_4 scheduled between two facilities i_1 and i_2 as shown in Figure 1. Both facilities have the same four discrete times represented by red dots $t_0 = 0$ to $t_3 = 16$. As an example, Load l_1 will start loading at origin i_1 at time 1 and is expected to be unloaded at destination i_2 at time 3. We also have that $\mathcal{L}^+_{(i_1,t_1)} = \{l_1\}$, $\mathcal{L}^-_{(i_1,t_1)} = \{l_1\}$, $\mathcal{L}^+_{(i_2,t_2)} = \{l_2, l_3\}$ and $\mathcal{L}^-_{(i_2,t_2)} = \{l_2, l_3\}$.

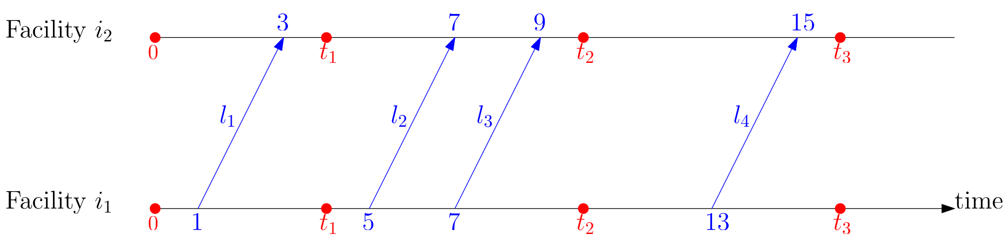

Figure 1: Example of four loads l_1 to l_4, represented by blue arrows, scheduled between two facilities i_1 and i_2. Both facilities have four discrete times $t_0 = 0$, $t_1 = 4$, $t_2 = 10$ and $t_3 = 16$ represented by red dots. The horizontal axis represents time.

- $\mathcal{A}$: The set of arcs linking nodes in $\mathcal{N}$. An arc a linking two nodes (i, t_1) and (j, t_2), represents the possibility of sending empty equipment from facility i at time t_1 and making it available at facility j by time t_2. For a given node $(i,t) \in \mathcal{N}$, we define the sets $\delta^-_{(i,t)}$ and $\delta^+_{(i,t)}$ as the sets of arcs in $\mathcal{A}$ that are inbound and outbound to (i,t) respectively.
  In the previous example with four loads, we define three arcs a_1 linking nodes $(i_1, 0)$ and (i_2, t_1), a_2 linking nodes (i_1, t_1) and (i_2, t_2) and a_3 linking nodes (i_1, t_2) and (i_2, t_3) as shown in Figure 2. As an example, we have that $\delta^+_{(i_1,0)} = \{a_1\}$, $\delta^-_{(i_2,t_1)} = \{a_1\}$, $\delta^+_{(i_1,t_1)} = \{a_2\}$ and $\delta^-_{(i_2,t_2)} = \{a_2\}$.
- I_{ie}: The inventory of equipment type e at facility i at the start of the planning horizon.

3.2 Model

We present a mixed integer programming formulation of the inventory-aware equipment management model. At time 0, each facility i in $\mathcal{F}$ has an initial inventory I_{ie} of equipment type e in $\mathcal{E}$. If the load plan were to be executed without any adjustments, it is possible that the inventory of

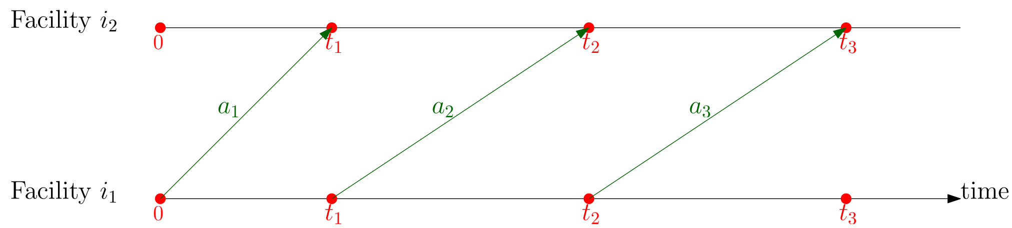

Figure 2: The set of arcs defined for the previous example with four loads. Arc a_1 links nodes $(i_1, 0)$ and (i_2, t_1), a_2 links nodes (i_1, t_1) and (i_2, t_2) and a_3 links nodes (i_1, t_2) and (i_2, t_3).

some equipment type drops below zero during the planning period. Our objective is to prevent this from happening by adjusting the load plan in one of two ways (or both):

1. **Equipment substitution**: assigning different equipment configurations (from the set of eligible equipment configurations) to loads.

2. **Empty repositioning**: adding one or more empty equipment repositioning movements between pairs of facilities to redistribute equipment from locations where there is a surplus to places where there is a shortage of a given equipment type. The judicious timing of any empty equipment repositioning movements is critical.

Equipment substitution and empty repositioning decisions incur costs for carriers. We ignore equipment substitution costs as they are negligible compared to empty equipment repositioning costs. The optimization model seeks to minimize the transportation costs of any added empty equipment repositioning movements. The solution to the optimization model needs to satisfy the following constraints:

1. **Load equipment substitution**: every planned load l can be assigned exactly one equipment configuration in S_l,

2. **Inventory flow balance**: at every facility, the inventory of a given equipment type is monitored during the planning period; properly accounting for arriving and departing loads and any added empty equipment repositioning movements,

3. **Non-negative inventory**: to prevent any equipment stock-out, inventory is not allowed to drop below zero during the planning period.

4. **Target inventory**: planners may or may not require a certain inventory of equipment at a facility at the end of the planning period. Inventory targets can be used to better position the system for anticipated future load demand.

3.2.1 Decision variables

- s_{iet}: inventory of equipment type $e \in \mathcal{E}$ at node $(i,t) \in \mathcal{N}$,
- y_{lc}: whether or not equipment configuration $c \in S_l$ is assigned to load $l \in \mathcal{L}$,
- u_{ae}: number of repositioning movements of equipment type $e \in \mathcal{E}$ added on arc $a \in \mathcal{A}$.

3.2.2 Formulation

$$(\mathcal{IAM}) \quad \min \sum_{a \in \mathcal{A}} \sum_{e \in \mathcal{E}} D_{ae} u_{ae} \tag{1}$$

$$\text{s.t. } s_{ie0} = I_{ie}, \quad i \in \mathcal{F}, e \in \mathcal{E}, \tag{2}$$

$$s_{iet} = s_{iet^-} + \left(\sum_{l \in \mathcal{L}^-_{(i,t)}} \sum_{c \in S_l} \eta_{ce} y_{lc} - \sum_{l \in \mathcal{L}^+_{(i,t)}} \sum_{c \in S_l} \eta_{ce} y_{lc} \right) + \left(\sum_{a \in \delta^-(i,t)} u_{ae} - \sum_{a \in \delta^+(i,t)} u_{ae} \right), \ (i,t) \in \mathcal{N}, \ t > 0, \ e \in \mathcal{E}, \tag{3}$$

$$\sum_{c \in S_l} y_{lc} = 1, \quad l \in \mathcal{L}, \tag{4}$$

$$y_{lc} \in \{0,1\}, \quad l \in \mathcal{L}, \ c \in S_l, \tag{5}$$

$$s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i,t) \in \mathcal{N}, e \in \mathcal{E}, \tag{6}$$

$$u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in \mathcal{A}, \ e \in \mathcal{E}, \tag{7}$$

where D_{ae} represents the cost of executing an empty movement with equipment type e in $\mathcal{E}$, on arc a in $\mathcal{A}$. For simplicity, we use the distance of arc a to represent the cost.

The objective function (1) represents the transportation costs of all the new empty movements generated by the model. Constraints (2) set the initial inventory. Constraints (3) ensure flow balance at each node $(i,t) \in \mathcal{N}$ for each equipment type $e \in \mathcal{E}$. Constraints (4) ensure that every load l is assigned exactly one configuration in the set S_l.

Target inventories at the end of the planning period can be accommodated by adding lower and upper bounds $\underline{s}^T_{ie}$ and $\bar{s}^T_{ie}$ on the variables s_{ieT}, i.e.,

$$\underline{s}^T_{ie} \leq s_{ieT} \leq \bar{s}^T_{ie}. \tag{8}$$

4 Methodology

Instances of the formulation for the short-term inventory-aware ground equipment management problem tend to be very difficult to solve. The main reason is that the size of the instances for the service networks of interest becomes prohibitively large. The number of facilities, the number of equipment configurations, and the number of loads is already very large, but the number of possible empty equipment repositioning movements is astronomical for a fine discretization of time (the number is of the order of $0.5 \times (n \times t)^2 \times e$ with n the number of facilities, t the number of time points (at a facility), and e the number of equipment configurations, e.g., a week-long planning period with an hourly discretization of time would results in the order of $(1500 \times 168)^2 \times 20 \approx 0.64 \times 10^{12}$ possible empty equipment repositioning movements).

In this section, we explore approaches to solve instances of the formulation in a reasonable amount of time by judiciously choosing a discretization of time and generating empty equipment repositioning movement options dynamically.

4.1 Time discretization

The time discretization, i.e., the choice of the sets $\mathcal{T}(i)$ for $i \in \mathcal{F}$ is an essential feature of the inventory-aware equipment management problem and affects two aspects of the model. First, by enforcing that the inventory of equipment at a facility at certain time points is non-negative, we avoid (or minimize) the risk of equipment stock-outs. With a fine discretization, the inventory is monitored at a large number of time points and only short-lived stock-outs occurring between consecutive time points are overlooked. However, a large number of time points implies a large formulation which may cause computational issues. On the other hand, with a coarse discretization we may overlook longer lasting stock-outs that can be detrimental to the carrier's business as it may cause disruptions in operations, e.g., delays and missed service promises. This is illustrated in Figure 3. Second, the set of time points at a facility defines the possible departure times for empty equipment repositioning movements. The larger the number of time points, the more empty equipment repositioning movement options, but the larger the number of time points, the larger the formulation. Thus, the choice of time points is critical when seeking to find high-quality solution in a reasonable amount of time. Finally, it is important to recognize that the times at which you evaluate equipment inventory at a facility and the times at which you consider dispatching empty equipment to another facility do *not* have to be the same.

We focus first on the set of times points at a facility at which we will evaluate the equipment inventory. Our approach is motivated by the fact that a stock-out only occurs at a time when a load departs, i.e., the load requires a certain equipment type, but the inventory of that equipment type at the facility is zero. This implies that ignoring load arrival times and evaluating equipment

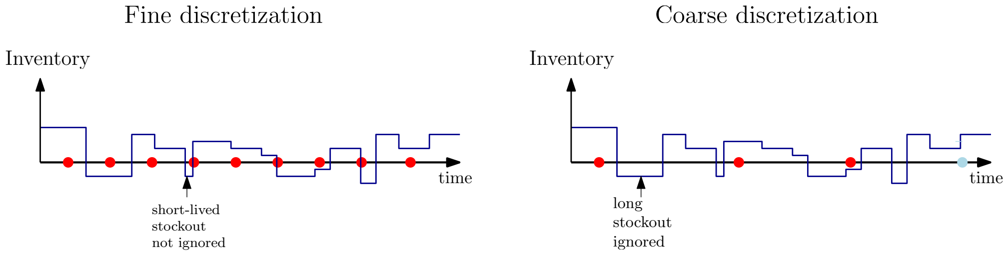

Figure 3: Fine vs coarse discretization and the impact on sensitivity of the model to short and long stock-outs

inventory only at load departure times suffices to identify stock-outs, if any. However, we can do even better. At each facility i, we aggregate inbound and outbound loads in $\mathcal{L}$ into inbound and outbound blocks such that within each inbound block of loads there is no outbound load, and within each outbound block of loads there is no inbound load (as illustrated in Figure 4). Let the set of nodes of the time-expanded network, $\mathcal{N}$, be formed by pairs (i, t) with t the start loading time $t^o(l^\star)$ of the last load $l^\star$ in each outbound block at facility i. (In the worst case, this implies a node for every departing load, i.e., $|\mathcal{N}| = |\mathcal{L}|$.) Figure 4 depicts an example of this aggregation. In

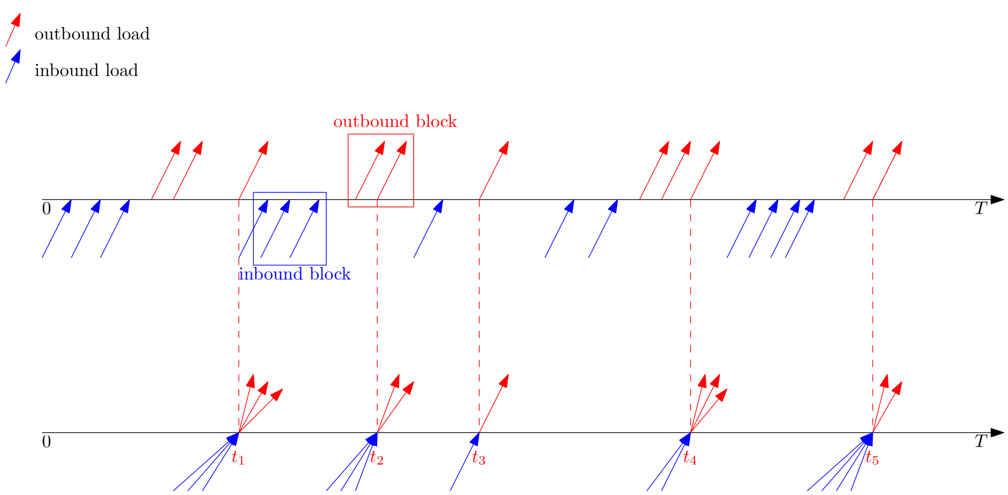

Figure 4: Example of consecutive inbound and outbound blocks at a given facility. Blue and red arrows represent inbound and outbound loads, respectively. The bottom part shows the aggregated form of the loads and final discrete time points t_1 to t_5 that we keep at that facility.

the example, the set of time points at the facility will be $\{0, t_1, t_2, t_3, t_4, t_5, T\}$ and the set of nodes in the time-expanded network for the facility will be $\{(i, 0), (i, t_1), (i, t_2), (i, t_3), (i, t_4), (i, t_5), (i, T)\}$. Next, We formally prove the validity of the aggregation scheme.

Proposition 1. *For a given facility-equipment type pair, there will be no equipment stock-out during the planning period* $[0, T]$ *if and only if the equipment inventory is non-negative at the start of the planning period and at the end of each outbound block.*

Proof. For a given facility i and equipment type e, let $I_{ie} : t \mapsto I_{ie}(t)$ denote the function that monitors the inventory of equipment type e at any time t in $[0, T]$. We want to prove that I_{ie} is a nonnegative function *if and only if* it is nonnegative at the time points in $\mathcal{T}(i)$, i.e.:

$$\forall t \in [0, T] \quad I_{ie}(t) \geq 0 \iff \forall t \in \mathcal{T}(i) \quad I_{ie}(t) \geq 0$$

The direction $\implies$ is trivial as any time-point t in $\mathcal{T}(i)$ is in $[0, T]$. For the direction $\impliedby$ let $\mathcal{T}(i) = \{0 = t_1, t_2, \ldots, t_{|\mathcal{T}(i)|} = T\}$. We consider two cases:

- $t \in [t_j, t_{j+1}]$ with $j \leq |\mathcal{T}(i)| - 1$: By the definition of a block, in the interval $[t_j, t_{j+1}]$ there will first be a set of inbound loads (possibly empty), followed by a set of outbound loads (possibly empty). Thus, there is a unique time point, t^M, at which the maximum inventory during the interval is reached for the first time. Hence, if $t \in [t_j, t_j^M]$ then $I_{ie}(t) \geq I_{ie}(t_j) \geq 0$, and if $t \in [t_j^M, t_{j+1}]$ then $I_{ie}(t) \geq I_{ie}(t_{j+1}) \geq 0$.

- $t \in [t_{|\mathcal{T}(i)|-1}, t_{|\mathcal{T}(i)|}]$: After $t_{|\mathcal{T}(i)|-1}$ there are only arriving loads. Hence, the inventory only increases after $t_{|\mathcal{T}(i)|-1}$. Thus, we have $I_{ie}(t) \geq I_{ie}(t_{|\mathcal{T}(i)|-1}) \geq 0$. ■

Although this aggregation scheme ensures that stock-outs can be avoided, it may have two undesirable features. First, at busy facilities with many daily inbound and outbound loads, the aggregation scheme may generate many time points with little time separation due to many alternating small inbound and outbound blocks. Second, at less busy facilities with few daily inbound and outbound loads or with more inbound than outbound or more outbound than inbound loads, this aggregation scheme may generate few time points. Enforcing no stock-outs at a facility between two consecutive time points that are close in time may be unnecessary and having only a few time points at a facility may prevent necessary empty equipment repositioning movements. To address these issues, at busy facilities we enforce a minimum time separation between time points (τ_m) at which we enforce positive inventory and at less busy facilities we enforce a maximum time separation between time points (τ_M), by adding additional time points if necessary, to ensure sufficient opportunities for empty equipment repositioning.

4.2 Solving the LP relaxation

Even with a judicious choice of time points at facilities, for a large ground service network, the set of possible empty equipment repositioning arcs, $\mathcal{A}$, can be prohibitively large. Including all repositioning arcs in the formulation may result in memory issues or excessive solution times, even for just solving the LP relaxation. Moreover, only a few of the repositioning arcs will likely be chosen in an optimal solution. Therefore, we generate repositioning arc variables dynamically as needed, i.e., we use a column generation approach to solve the LP relaxation. To be able to define the reduced cost of a repositioning arc variable given the solution to a restricted formulation (i.e., a formulation in which many repositioning arc variables have been omitted), we need to look at the dual of the LP relaxation. Let the dual variables associated with Constraints 3, 4, and 5 of the LP relaxation of $\mathcal{IAM}$ be π_{iet}, β_l, and γ_{lc}, respectively. Then the dual problem is

$$(\mathcal{D}-\mathcal{IAM}) \quad \max \sum_{(i,0)\in\mathcal{N}} \sum_{e\in\mathcal{E}} I_{ie}\pi_{ie1} - \sum_{l\in\mathcal{L}} \left(\beta_l + \sum_{c\in S_l} \gamma_{lc}\right) \tag{9}$$

$$\text{s.t.}\ \pi_{iet} - \pi_{iet^+} \leq 0,\ \forall (i,t) \in \mathcal{N},\ 0 < t < T,\ e \in \mathcal{E}, \tag{10}$$

$$\pi_{ieT} \leq 0,\ \forall (i,T) \in \mathcal{N},\ e \in \mathcal{E}, \tag{11}$$

$$\pi_{iet} - \pi_{jet'} \leq D_{ae},\ \forall a = ((i,t) \to (j,t')) \in \mathcal{A},\ e \in \mathcal{E}, \tag{12}$$

$$\eta_{ce}(\pi_{iet} - \pi_{jet'}) - \beta_l - \gamma_{lc} \leq 0, \quad l = ((i,t) \to (j,t')) \in \mathcal{L},\ c \in S_l, \tag{13}$$

$$\gamma_{lc} \geq 0, \quad l \in \mathcal{L},\ c \in S_l, \tag{14}$$

$$\pi_{iet}, \beta_l\ \text{free} \quad (i,t) \in \mathcal{N}, t > 0,\ e \in \mathcal{E}. \tag{15}$$

Observe that Constraints (10) and (11) imply that the dual variables π_{iet} are non-positive and monotonically non-decreasing with respect to t. This observation will be used to speed up the dynamic variable generation strategy.

Next, assume that we have a solution to a restricted LP relaxation that only includes a subset $\mathcal{A}_1 \subseteq \mathcal{A}$ of repositioning arcs, then finding a variable $u_{a'e'}$ with $a' \in \mathcal{A} \setminus \mathcal{A}_1$ and $e' \in \mathcal{E}$ with minimum reduced cost amounts to solving the following pricing problem:

$$\min_{\substack{e\in\mathcal{E},\ a\in\mathcal{A}\setminus\mathcal{A}_1 \\ a=((i,t),(j,t'))}} D_{ae} - \pi_{iet} + \pi_{jet'} \tag{16}$$

If the minimum is non-negative, then the solution to the restricted LP relaxation is also optimal to the (full) LP relaxation. Otherwise, we have identified a variable that should be added to the restricted LP relaxation.

Adding one variable at a time, however, is computationally too expensive as it will require the solution of many (still large) restricted LP relaxations. Therefore, instead, we search for and add a number of negative reduced cost variables in each iteration. This results in the following algorithm for solving the LP relaxation of $\mathcal{IAM}$, where parameter N_{iter} indicates the maximum number of negative reduced cost variables that are generated and added to the restricted LP relaxation in a single iteration:

- **Step 0**: Initialize $\mathcal{A}_1$ with a small subset of repositioning arc variables,
- **Step 1**: Solve the restricted LP relaxation with subset $\mathcal{A}_1$,

- **Step 2**: Generate a set $\mathcal{A}_2 \subseteq \mathcal{A} \setminus \mathcal{A}_1$ of up to N_{iter} negative reduced cost arc repositioning variables. If $\mathcal{A}_2 = \emptyset$, go to **Step 4**,
- **Step 3**: Add the columns in $\mathcal{A}_2$ to $\mathcal{A}_1$. Go to **Step 1**,
- **Step 4**: Stop. An optimal solution to the LP relaxation has been found.

To generate negative reduced cost repositioning arc variables (in **Step 2**), we consider three strategies : BASIC, a simple enumeration strategy, ENHANCED BASIC, a more intelligent enumeration strategy that favors diversity, and EFFICIENT ENHANCED BASIC - a sophisticated enumeration strategy that exploits dual information to guide and restrict the search.

BASIC STRATEGY Our naive enumeration strategy iterates over equipment types and facilities in no particular order. For each combination of equipment type e and facility i, it iterates over the set of facilities that can reach facility i directly, i.e., its inbound arcs, again in no particular order, and for each outbound arc, iterates over the time points in $\mathcal{T}(i)$. If the reduced cost of the associated repositioning arc variable is negative, it is added to the set $\mathcal{A}_2$. The enumeration stops as soon as N_{iter} negative reduced cost variables have been found. The exact same search is performed in each iteration.

ENHANCED BASIC STRATEGY To introduce more diversity in the set of generated negative reduced cost repositioning arc variables, we impose limits on the number of negative reduced cost variables generated for each equipment type e, N_e, for each facility i, N_f, and for each outbound arc, N_a. Furthermore, when sorting is enabled, we iterate over the equipment types and the facilities in a certain order to increase the chances of finding negative reduced cost variables early in the enumeration. We iterate over the equipment types $e \in \mathcal{E}$ in nonincreasing order of

$$\lambda_e = \frac{\textit{\# explored variables with negative reduced cost}}{\textit{\# explored variables}},$$

where λ_e is computed based on information gathered in the previous iteration. Similarly, within each equipment type e, we iterate over the facilities in nonincreasing order of λ_{ei}, defined similar to the quantity λ_e at the facility level. In the first iteration, we set $\lambda_e = 1$ for $e \in \mathcal{E}$ and $\lambda_{ei} = 1$ for $e \in \mathcal{E}, i \in \mathcal{F}$. When sorting is disabled, we use a round robin scheme that works as follows. In each iteration, we start from the last equipment type explored in the previous iteration, and for each equipment type, we start from the last facility explored in the previous iteration.

Moreover, when sorting is enabled, we truncate the search of equipment categories using the λ_e values. Specifically, we stop the enumeration as soon as we reach an equipment category with $\lambda_e = 0$, provided that a minimum number of negative reduced cost variables were found earlier in the iteration. The rationale for this heuristic idea is as follows. If no negative reduced cost variables were found for an equipment type in the previous iteration, i.e., no empty repositioning of equipment appeared advantageous, it is likely that no negative reduced cost variables will be found in the current iteration, and searching for them may be a waste of time. This idea is especially useful in practice, as companies often have large number of equipment types, often more than ten, but primarily use a few, often only three or four. To ensure the linear program is solved to optimality, we do not stop the search early when no negative reduced cost variables have been found up to that point.

In addition to the control parameters N_{iter}, N_e, N_f and N_a, we use the following additional parameters:

$Sort$: A boolean that when set to true activates the sorting of sets when searching for columns with negative reduced cost. Equipment categories and facilities are processed based on the order explained earlier. When set to false, a round robin scheme is used to diversify the processing of equipment types and facilities.

$Best$: A boolean that when set to true selects the N_a most negative reduced cost timed repositioning arcs (i.e., for a pair of facilities), and when set to false selects the first N_a negative reduced cost repositioning arcs.

l, m : These quantities as associated with the round robin scheme. l represents the index of the last equipment type explored in the previous iteration, and m represents a list of indexes of the last facilities (one for each equipment type) explored in the previous iteration.

Algorithm 1 gives the pseudo-code for this strategy.

Efficient Enhanced Basic Strategy The previous strategies may unnecessarily evaluate the reduced cost of many repositioning arc variables. By cleverly exploiting dual information, such evaluations can be avoided, which will improve the efficiency. Furthermore, exploiting dual information may also lead to more effective evaluation orders (e.g., the order in which facilities are examined). For each combination of equipment type e and facility i, the dual variables π_{iet} are nonpositive and monotonically nondecreasing in $t \in \mathcal{T}(i)$, i.e.,

$$\pi_{iet_1} \leq \pi_{iet_2} \leq \cdots \leq \pi_{ieT} \leq 0. \tag{17}$$

This follows from Constraints (10) and (11) in dual formulation $\mathcal{D} - \mathcal{IAM}$.

This property can be exploited to avoid enumerating (some) repositioning arc variables. For a given equipment type e and repositioning arc $a = ((j, t'), (i, t))$, the reduced cost $D_{ae} + \pi_{iet} - \pi_{jet'}$ can be divided into parts π_{iet} and $D_{ae} - \pi_{jet'}$. As π is nonpositive, the first part is always nonpositive and the second part is always nonnegative.

Algorithm 1: ENHANCED-BASIC(N_{iter},N_e,N_f,N_a,$Sort$,$Best$,ℓ,m,λ)

```
F_1, E_1 ← unordered lists of facilities in the network and equipment categories
C ← {}
k_1 ← ℓ + 1 // index of last equipment category searched in previous iteration
if Sort then
    E_1 ← Equipment categories sorted by λ in non-increasing order
    k_1 ← 1
for each equipment type e = k_1, .., |E_1| in E_1 do
    C_e ← {}
    k_2 ← m_e + 1 // index of last facility searched in previous iteration for equipment type e
    if Sort then
        Facilities ← facilities sorted by λ in non-increasing order
        k_2 ← 1
    for each facility i = k_2, .., |F_1| in F_1 do
        T(i) ← set of time-points at facility i in the order of time
        Inbound[i] ← unordered list of facilities j with arc (j, i)
        C_f ← {}
        for each facility j in Inbound[i] do
            C_a ← {}
            for each time-point t in T(i) do
                a = (j, t_j) → (i, t) // available empty repositioning arc
                if D_ae + π_iet − π_jet_j < 0 then
                    C_a ← a
                if |C_a| ≥ N_a then
                    break
            C_f ← C_f ∪ C_a
            if |C_f| ≥ N_f then
                break
        C_e ← C_e ∪ C_f
        if |C_e| ≥ N_e then
            break
    if |C| > 0 & λ_{e+1} = 0 then
        break
    C ← C ∪ C_e
    if |C| ≥ N_iter then
        break
return C
```

By using appropriate orderings, we can stop the enumeration early in three situations. First, suppose that for a given equipment type e the facilities are given in non-decreasing order of

$$\underline{\pi}_{ei} = \min_{t \in \mathcal{T}(i)} \pi_{iet}.$$

Then, we can stop the enumeration as soon as we reach a facility with $\underline{\pi}_{ei} = 0$, as the reduced costs for all repositioning arc variables for all remaining facilities will be nonnegative. Second, suppose that for a given combination of equipment e type and facility i, the inbound arcs (j, i) are given in nondecreasing order of $D_{(ji)e} - \bar{\pi}_{ej}$ with

$$\bar{\pi}_{ej} = \max_{t \in \mathcal{T}(i)} \pi_{jet}.$$

Then, we can stop the enumeration as soon as we reach an inbound arc with $\underline{\pi}_{ei} + D_{(ji)e} - \bar{\pi}_{ej} > 0$. Finally, for a given inbound arc (j, i), because we enumerate time points in increasing order of time, we can stop the enumeration as soon as we reach a repositioning arc with $D_{(ji)e} + \pi_{iet} \geq 0$.

Exploiting dual information as described does require sorting and thus comes at a price, but hopefully the time spent in sorting is offset by far fewer reduced cost evaluations. The effect of the *Sort* parameter is redefined as follows in this variant:

Sort : A boolean that when set to true activates the sorting of sets when searching for columns with negative reduced cost. Equipment categories are processed in nonincreasing order of their contribution to the objective function in the last iteration. For a given equipment category e, facilities are processed in non-decreasing order of $\underline{\pi}_{ie}$. For a given facility i, the inbound arcs (j, i) are processed in non-decreasing order of $D_{(ji)e} - \bar{\pi}_{ej}$. When set to false, a round robin scheme is used to diversify the processing of equipment types and facilities.

Algorithm 2 gives the pseudo-code for this strategy.

Each of the three pricing algorithms discussed above, i.e., BASIC, ENHANCED-BASIC, and EFFICIENT-ENHANCED-BASIC, can be embedded in the iterative algorithm LP-HEUR for approximately solving the LP relaxation of $\mathcal{IAM}$ outlined in Algorithm 3. LP-HEUR uses three additional parameters, K_1, K_2, and N_{LP}. Parameters K_1 and K_2 are used to determine the variant of the simplex algorithm to solve the current restricted linear program. While the number of negative reduced cost variables added in an iteration, say t, is large, the dual simplex method is used, but if after a fixed number of iterations ($t > K_2$) the number of negative reduced cost variables added in an iteration is small ($|\mathcal{C}_t| < K_1$), we switch to using primal simplex method. The primal simplex method is more effective if only a few negative reduced costs have been added. Parameter N_{LP} is a limit on the total number of variables added. When N_{LP} needs to be set to infinity, the linear

Algorithm 2: EFFICIENT-ENHANCED-BASIC(N_{iter},N_e,N_f,N_a,$Sort$,$Best$,ℓ,m)

```
F_1, E_1 ← unordered lists of facilities in the network and equipment categories
C ← {}
if Sort then
    E_1 ← Equipment categories sorted by current objective cost in non-increasing order
    k_1 ← 1
else
    k_1 ← ℓ + 1 // index of last equipment category searched in previous iteration
for each equipment type e = k_1, .., |E_1| in E_1 do
    π̲_e, π̄_e ← minimum and maximum of dual variables π for each facility
    C_e ← {}, r ← 0, r_prev ← 0
    if Sort then
        F_1 ← facilities sorted by π̲_e in non-decreasing order
        k_2 ← 1
    else
        k_2 ← m_e + 1 // index of last facility searched in previous iteration for e
    for each facility i = k_2, .., |F_1| in F_1 do
        if π̲_ei = 0 then
            continue
        Inbound[i] ← unordered list of facilities j with arc (j, i)
        if Sort then
            Inbound[i] ← facilities j with arc (j, i) sorted by D_(ji)e − π̄_ej in non-decreasing order
        C_f ← {}
        for each facility j in Inbound[i] do
            if Sort and D_(ji)e + π̲_ei − π̄_ej > 0 then
                break
            if Sort = False and D_(ji)e + π̲_ei − π̄_ej > 0 then
                continue
            C_a ← {} // list of at most N_a negative reduced cost timed arcs (sorted)
            T(i) ← set of time-points at facility i in the order of time
            for each time-point t in T(i) do
                if D_(ji)e + π_iet ≥ 0 then
                    break
                a ← ((j, t_j), (i, t)) // available empty repositioning arc
                r ← D_(ji)e + π_iet − π_jet_j // reduced cost of arc a
                if r = r_prev then
                    continue // skipping arcs with the same reduced cost as the last one found
                r_prev ← r
                if r < 0 then
                    C_a ← a
                if Best = False and |C_a| ≥ N_a then
                    break
            C_f ← C_f ∪ C_a
            if |C_f| ≥ N_f then
                break
        C_e ← C_e ∪ C_f
        if |C_e| ≥ N_e then
            break
    C ← C ∪ C_e
    if |C| ≥ N_iter then
        break
return C
```

program is solved to optimality. However, when N_{LP} is set to a finite number, and the algorithm is terminated because this limit is reached, only an approximate solution to the linear program is obtained. Solving the linear program approximately can be considered in case solution times become prohibitive.

Algorithm 3: LP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)

```
LP ← LP relaxation of IAM model with an initial set of empty repositioning variables
Terminate ← False
C_t ← ∅
t ← 0
ℓ ← 0
m ← 0 (vector of size |E|)
while Terminate = False do
    Solve LP and retrieve values of dual variables
    C_t ← PRICING-ALGORITHM(N_iter,N_e,N_f,N_a,Sort,Best,ℓ,m)
    if C_t = ∅ then
        break
    C ← C ∪ C_t
    if |C| ≥ N_LP then
        break
    if |C_t| < K_1 & t > K_2 then
        Switch to Primal Simplex when solving LP
    t ← t + 1
    Update LP with new columns in C_t
return C
```

Target inventory constraints. So far, we have ignored any target inventory constraints. Unfortunately, when target inventory constraints are included, a few things change. The monotonicity property of the dual values remains true, as Constraints (10) are unchanged, but the non-positive property of dual values may no longer be satisfied when we enforce maximum target inventory constraints. Let α^l_{ie} and α^u_{ie} denote the dual variables associated with the minimum and maximum target inventory constraints respectively. Constraints (11) become

$$\pi_{ieT} + \alpha^l_{ie} - \alpha^u_{ie} \leq 0 \qquad \forall (i,T) \in \mathcal{N},\, e \in \mathcal{E}. \tag{18}$$

When maximum target inventory constraints are not present, we have

$$\pi_{ieT} \leq -\alpha^l_{ie} \leq 0 \qquad \forall (i,T) \in \mathcal{N},\, e \in \mathcal{E}, \tag{19}$$

which, because α^l_{ie} is non-negative, ensures non-positive dual values. However, in the presence of maximum target inventory constraints, non-positive dual values can no longer be guaranteed.

4.3 Solving the IP

When the substitution variables y_{lc} are fixed, say at values $\bar{y}_{lc}$, then $\mathcal{IAM}$ reduces to a number of minimum cost flow problems, one for each equipment type, with flow variables s_{iet} and u_{ae} as represented in Figure 5.

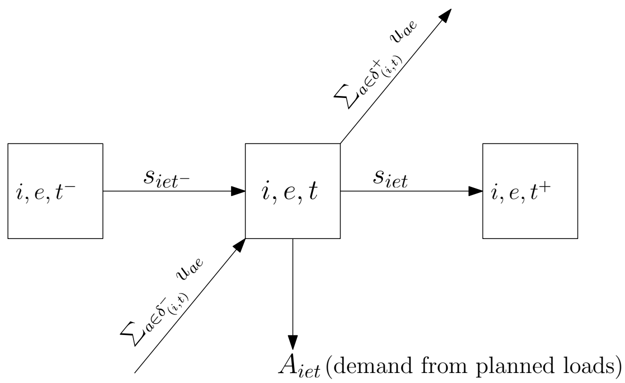

Figure 5: Inventory flow of equipment e at node (i, t).

Here A_{iet} represents the contribution of the planned loads to the inventory of equipment e at node (i, t); it can take on positive or negative values and is calculated as follows:

$$A_{iet} = \sum_{l\in\mathcal{L}^+_{(i,t)}} \sum_{c\in S_l} \eta_{ce}\bar{y}_{lc} - \sum_{l\in\mathcal{L}^-_{(i,t)}} \sum_{c\in S_l} \eta_{ce}\bar{y}_{lc}.$$

More specifically, the resulting problem is

$$\min \sum_{a\in\mathcal{A}} \sum_{e\in\mathcal{E}} D_a u_{ae} \tag{20}$$

$$\text{s.t.} \left(s_{iet_1} + \sum_{a\in\delta^+(i,t_1)} u_{ae} \right) - \left(\sum_{a\in\delta^-(i,t_1)} u_{ae} \right) = I_{ie} - A_{iet_1},\ \forall (i, t_1) \in \mathcal{N},\ e \in \mathcal{E}, \tag{21}$$

$$\left(s_{iet} + \sum_{a\in\delta^+(i,t)} u_{ae} \right) - \left(s_{iet^-} + \sum_{a\in\delta^-(i,t)} u_{ae} \right) = -A_{iet},\ \forall (i, t) \in \mathcal{N},\ t > 1,\ e \in \mathcal{E}, \tag{22}$$

$$s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i, t) \in \mathcal{N}, e \in \mathcal{E}, \tag{23}$$

$$u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in \mathcal{A},\ e \in \mathcal{E}, \tag{24}$$

which, because there is no longer any interaction between equipment types, decomposes into $|\mathcal{E}|$ minimum cost flow problems. This suggests that a branching scheme that focuses on the substitution variables is appropriate for solving $\mathcal{IAM}$.

However, given that even solving the LP relaxation is time consuming for the size of instances that we are interested in, we employ, price-and-branch, a well-known heuristic scheme. In a price-and-branch scheme, the LP relaxation at the root node of the search tree is solved using dynamic pricing of variables, and after that an IP is solved using only the (partial) set of variables generated at the root node. This is a heuristic, because to obtain a proven optimal solution it will be necessary to dynamically generate variables at every node in the search tree (as in branch-and-price algorithms). Algorithm 4 gives the pseudo-code of IP-HEUR.

Algorithm 4: IP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)

$\mathcal{IP} \leftarrow \mathcal{IAM}$ *model with initial set of empty repositioning variables*
$\mathcal{C} \leftarrow \emptyset$
$\mathcal{C} \leftarrow$ LP-HEUR(N_{LP},K_1,K_2,N_{iter},N_e,N_f,N_a,$Sort$,$Best$)
$\mathcal{IP} \leftarrow \mathcal{IAM}$ *model with expanded set of empty repositioning variables, i.e., including variables in* $\mathcal{C}$
Solve $\mathcal{IP}$

5 Computational Study

We have conducted a set of computational experiments to demonstrate the value of the proposed inventory-aware equipment management model ($\mathcal{IAM}$) for a package express carrier operating a large ground service network with a large and heterogeneous fleet of trailers and containers. The experiments assess the computational efficiency of the proposed solution methodology and extract business insight regarding equipment management, by answering the following questions:

- What performance enhancements are achieved by more sophisticated variable pricing schemes, i.e., what performance improvements are observed when employing ENHANCED-BASIC and EFFICIENT-ENHANCED-BASIC rather than the naive pricing scheme BASIC?
- What is the impact of the pricing scheme parameters on the efficiency of solving the LP relaxation of $\mathcal{IAM}$ for a given time discretization?
- What is the impact of having a finer time discretization? What is the trade-off between efficiency (reducing the run-time) and quality (reducing the transportation cost)?
- What is the trade-off between leveraging equipment substitutions and introducing empty repositioning movements to ensure no equipment stock-outs during the planning period?
- What is the trade-off between the equipment fleet size and the empty repositioning costs?

5.1 Instances

We use a set of ten instances in the computational study. The instances are derived from historical weekly load planning data provided by a major U.S. package express carrier. The carrier's ground network has about 2300 facilities, which include company sites, customer locations, and other locations where equipment inventory is monitored, e.g., rail yards. Table 1 summarizes relevant characteristics of the instances.

Instance	# Active Facilities	# Loads	Total Mies	# Time Points	Fleet Size	Empty Legs (%)	Bobtail Legs (%)
1	1,152	181,165	34,145,280	28,274	19,491	33.10	13.70
2	1,149	180,375	33,948,206	28,143	19,554	33.28	13.80
3	1,149	179,619	33,840,054	28,080	19,375	33.09	13.64
4	1,148	179,527	33,858,084	28,029	19,438	32.85	13.49
5	1,147	180,834	34,286,951	28,093	19,763	32.23	13.36
6	1,149	182,867	34,841,019	28,167	20,238	31.77	13.08
7	1,151	185,385	35,699,631	28,364	20,731	31.30	12.80
8	1,149	189,136	36,531,351	28,681	20,939	31.06	12.86
9	1,149	188,987	36,788,322	28,664	20,626	31.00	12.47
10	1,149	191,092	37,636,547	28,803	21,219	30.82	12.36

Table 1: Instance characteristics. A facility is considered active when there is at least one inbound or outbound load at the facility during the week. The fleet size is based on the equipment at an active facility and on the en-route equipment at the start of the planning period. The number of time-points is based on parameters $\tau_m = 30$ minutes and $\tau_M = 1$ day.

The similarities between the instances are a consequence of the fact that they are derived from consecutive weeks of data. Each instance is made up of a weekly load plan that contains all the loads that are scheduled to depart during the week. The timed loads are of two types: (a) loaded movements with an assigned equipment type and a specified volume (as a percentage of equipment capacity), and (b) empty movements with an assigned equipment type but without a specified volume. In addition to the timed loads, a load plan also contains a set of timed bobtail movements that can be leveraged to reposition empty trailers. All these movements have a fixed dispatch and arrival time. These times account for the time required for loading and unloading, so that the dispatch time corresponds to the time equipment is taken from the yard and the arrival time corresponds to the time equipment is delivered to the yard. The instances have about 200 thousand movements, with about 55% of these being loaded, 32% being empty, and 13% being bobtails.

The carrier operates a heterogeneous fleet of 13 equipment types. These differ in terms of characteristics such as size (e.g., 53 foot trailers and 28 foot pups), intermodal compatibility (e.g. containers and trailers on flatcar), ownership (e.g., company, customer, or third party owned), etc. Table 2 gives the composition of the fleet for Instance 1. Only one composite configuration is allowed

Equipment Id	# Units	Percentage (%)
1	77	0.40
2	748	3.84
3	16	0.08
4	3	0.02
5	21	0.11
6	12,541	64.34
7	609	3.12
8	183	0.94
9	73	0.37
10	294	1.51
11	487	2.50
12	542	2.78
13	3,897	19.99

Table 2: Types and number of units of equipment for Instance 1

in the network, namely, the 2-pup train widely used in U.S. ground transportation. Each instance comes with an equipment allowance table that specifies for each load, a set of configurations of equipment types that can be assigned to the load. This table is used to generate the sets S_l for each load l.

Each instance includes a snapshot of the system at the start of the planning period. This snapshot includes the inventory of each facility-equipment type pair, and in-transit equipment that is expected to arrive at a facility during the planning period. As this information was not provided by the package express company, we artificially generated the initial inventories by using the load plan as follows. For each facility-equipment type pair, we calculate, based on inbound and outbound loads, the minimum initial inventory required to ensure that there will be no equipment stock-out during the planning period. We then randomly choose an initial inventory level from a uniform distribution centered around the minimum required inventory. By doing so, each facility in the network has either an surplus or a deficit. A deficit implies that the facility will experience one or more shortages during the planning period unless equipment substitutions and empty repositioning moves are planned.

To account for the possibility of equipment stock-outs, we add an artificial equipment "source" at each node of the time-expanded expanded network and this source can be used to ensure that no stock-out occurs; a high-penalty is incurred when using an artificial source to discourage their use (we prefer the use of equipment substitutions and empty repositioning). The penalty for using an artificial source is set to the cost of movement of 4,649 miles (the longest distance between two locations in the network). All instances are such that if empty repositioning movements can be introduced at any time during the planning period, then stock-outs can be avoided by equipment substitutions and empty repositioning.

The inventory-aware model is coded in C++. Mixed integer programs are solved using the commercial solver Gurobi 9.0 with default settings. All experiments were run in a 20-core machine with Intel(R) Xeon(R) 2.30GHz processors and 256GB of RAM. The optimality tolerance is set to 0.005. No time limit was enforced.

5.2 Inventory-aware equipment management

We start by solving the instances with the Efficient-Enhanced-Basic scheme, where we solve the LP relaxation to optimality ($N_{LP} = \infty$). Table 3 summarizes the results. We report the following statistics:

- **IP_OBJ**: objective value of the IP, i.e., the total miles of empty repositioning introduced,
- **LP_OBJ**: objective value of the LP relaxation,
- **# SUB**: number of loads for which the initial equipment type is replaced,
- **# ITER**: number of iterations, where the first iteration represents the solution of the LP relaxation without any empty repositioning variables,
- **# VAR**: total number of variables added,
- **VG_T**: total time spent searching negative reduced cost variables (in seconds),
- **LP_T**: total time spent solving LP relaxations (in seconds),
- **IP_T**: time spent solving the IP (in seconds),
- **T_T**: total time (in seconds).

We observe that the difference between the objective value of the IP and the objective value of the LP relaxation is very small (less than 0.54% in final gap on average). This shows that our price-and-branch heuristic (Algorithm 4) is effective and little can be gained from a full-blown branch-and-price implementation. The LP and IP objective values represent the total empty repositioning miles added to the original load plan. Comparing these values to the total miles in the original load plan (Table 1), we see that the increase is very small, less than 0.1%. In addition to new empty repositioning movements, the equipment configuration assigned to loads has been changed for about 40,000 loads (about 20% of the total number of loads) in the adjusted load plan.

We observe too that on average about 310,000 variables are generated during the solution of the LP relaxation and that on average this requires about 21 iterations. The total solution time is, on average, a bit less than 4 hours, of which about 4% is spent identifying negative reduced cost

Ins.	IP_OBJ	LP_OBJ	#SUB	#ITER	#VAR	VG_T	LP_T	IP_T	T_T
1	26,753	26,604	39,025	23	330,121	1,118	3,817	13,797	18,732
2	26,876	26,635	38,756	19	309,834	680	3,651	23,124	27,455
3	20,926	20,875	38,270	22	332,775	767	2,472	13,525	16,764
4	21,941	21,836	38,066	22	321,702	625	2,750	14,255	17,630
5	26,071	25,910	38,272	18	262,748	431	1,867	5,881	8,178
6	26,731	26,560	38,696	26	344,667	798	2,492	6,443	9,733
7	24,988	24,916	37,964	19	277,291	414	1,835	6,211	8,460
8	19,878	19,777	40,250	21	335,081	604	2,381	12,626	15,611
9	23,097	22,986	39,453	20	290,226	401	2,364	13,152	15,916
10	22,146	21,983	40,588	20	285,339	455	2,343	13,061	15,859

Table 3: Results using IP-Heur with default parameters $N_{LP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$.

variables, about 18% is spent solving LPs, and about 78% of time is spent solving the IP. A total time of less than 4 hours is acceptable for the intended use of $\mathcal{IAM}$.

Next, we explore the trade-off between equipment substitution and empty repositioning decisions. To do so, we add constraint

$$\sum_{l \in \mathcal{L}} \sum_{\substack{c \in S_l \\ c \neq q(l)}} y_{lc} \leq Cap \tag{25}$$

to $\mathcal{IAM}$, which limits the number of substitutions, and we vary the right hand side. More specifically, we solve the LP allowing no subsitutions and then solve different IPs (with the variables of the final LP) for different limits on the number of substitutions (i.e., different values of Cap).

The results for nine different limits can be found in Table 4. The results clearly demonstrate the benefit of equipment substitutions when ensuring no equipment stock-outs as they decrease the repositioning costs by more than 65% on average.

For Instance 2, we show the trade-off curve in Figure 6. For this case, we need 77,771 repositioning miles to avoid equipment stock-outs when no equipment substitutions are allowed (i.e., $Cap = 0$) as opposed to only 27,668 when no limit is imposed on the number of substitutions (i.e., $Cap = \infty$).

Next, we explore the minimum number of equipment substitutions required to reach the minimum required repositioning miles. This is valuable in practice, because even though equipment substitutions are "free", planners like to adjust the initial load plan as little as possible (i.e., with the fewest equipment substitutions). For that, we take a hierarchical approach where we solve $\mathcal{IAM}$ in the first stage and minimize the number of equipment substitutions in the second stage

Cap	0	50	100	200	500	1000	1,500	2,000	∞
Ins. 1	80,754	67,169	58,996	48,994	34,669	27,755	27,633	27,633	27,633
Ins. 2	77,771	65,208	58,406	49,449	35,683	27,840	27,668	27,668	27,668
Ins. 3	71,835	58,292	51,119	42,218	29,225	21,544	21,376	21,376	21,376
Ins. 4	71,120	59,330	52,550	43,679	30,269	22,623	22,412	22,412	22,412
Ins. 5	78,392	65,688	58,218	48,486	34,315	26,758	26,666	26,666	26,666
Ins. 6	80,764	67,682	59,843	49,707	35,052	27,451	27,255	27,255	27,255
Ins. 7	73,786	60,360	52,942	43,562	30,839	25,212	25,202	25,202	25,202
Ins. 8	72,938	58,729	50,958	41,358	27,323	20,698	20,668	20,667	20,667
Ins. 9	85,958	71,723	62,381	51,030	34,484	24,903	23,984	23,984	23,984
Ins. 10	70,397	59,082	51,641	41,986	28,719	22,910	22,906	22,905	22,905

Table 4: Trade-off between empty repositioning and equipment substitutions (using IP-Heur with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$.

forcing that the minimum repositioning costs found in the first stage do not change. The objective function of the second stage can be formulated as

$$\min \sum_{l \in \mathcal{L}} \sum_{\substack{c \in S_l \\ c \neq q(l)}} y_{lc} \tag{26}$$

and forcing that the minimum repositioning costs found in the first stage do not change is achieved by adding constraint

$$\sum_{a \in \mathcal{A}} \sum_{e \in \mathcal{E}} D_{ae} u_{ae} \leq \Omega^\star \tag{27}$$

where $\Omega^\star$ is the objective value of $\mathcal{IAM}$ model. For the ten instances, we find that this hierarchical approach results in a number of substitutions that is, on average, less than 1% of the total number of loads.

Finally, we explore the trade-off between the fleet size and the required empty repositioning cost. To do so, we vary the initial inventory of equipment in the network. More specifically, for each facility i and equipment type e, we adjust the inventory I_{ie} by multiplying it by a factor $\nu \geq 1$, i.e.,

$$\hat{I}_{ie} = \nu I_{ie},$$

where $\hat{I}_{ie}$ denotes the adjusted initial equipment inventory. The results can be found in Table 5, where **FS** represents the fleet size (with adjusted initial equipment inventory at the facilities. We observe that increasing the fleet size by 10% reduces the empty repositioning costs by about 30%.

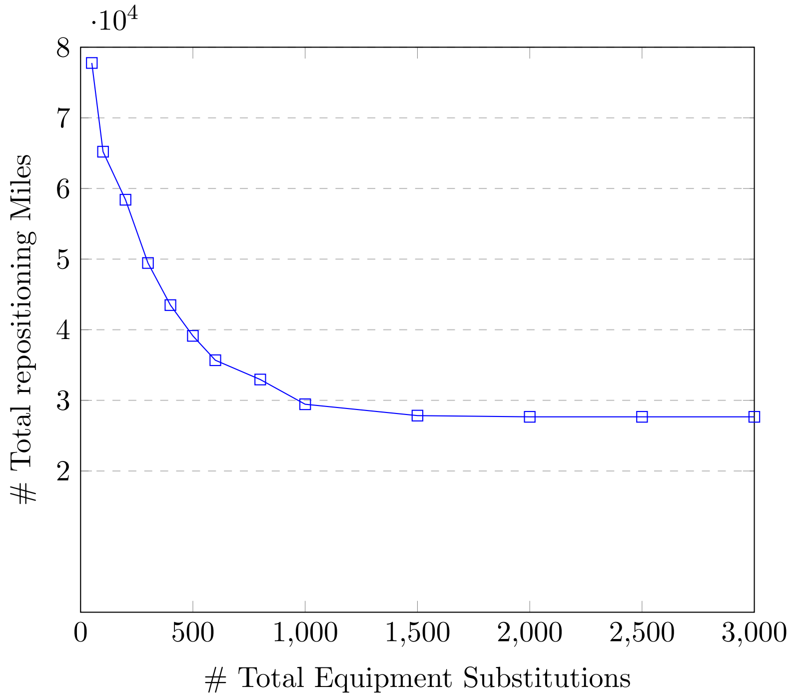

Figure 6: Relationship between the total repositioning cost required (in miles) and the limit on the number of substitutions allowed for Instance 2

As less empty repositioning is required, we see that fewer empty repositioning variables have to be generated (about 15%), which requires fewer iterations (about 27%) and less time (about 29%).

5.3 Impact of Algorithmic Features and Choices

The performance of the price-and-branch heuristic, both in terms of the quality of the solution obtained and the efficiency with which this solution was produced, are impacted by many algorithmic features and choices. In this section, we assess this impact systematically.

5.3.1 Impact of the Discretization Scheme

To assess the impact of the discretization scheme on solution quality and algorithm efficiency, we conduct two experiments. First, we fix the minimum time between two consecutive time points, τ_m, to be 30 minutes and vary the maximum time between two consecutive time points, τ_M. Second, we fix the the maximum time between two consecutive time points, τ_M, to be 24 hours and vary the minimum time between two consecutive time points, τ_m. The goal is to quantify the impact of the number of time points as well as the organization of time points on quality and efficiency.

The results can be found in Tables 6 and 7, where **#TPT** represents the number of time-points, **OBJ** the repositioning cost, and **#P** the number of stock-outs (i.e., total number of equipment

INS.	η	FS	IP-OBJ	LP-OBJ	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	1	19,491	26,753	26,604	328,529	17	397	1,729	5,971	8,096
	1.05	20,300	24,958	24,804	327,981	18	542	1,724	6,240	8,506
	1.10	21,371	18,156	17,975	279,623	12	359	1,440	5,604	7,403
2	1	19,554	26,876	26,635	283,467	10	168	1,407	14,820	16,394
	1.05	20,354	24,439	24,221	250,899	8	145	1,386	5,744	7,274
	1.10	21,458	17,872	17,753	262,690	9	200	1,192	4,502	5,894
3	1	19,375	20,926	20,875	331,735	17	397	1,738	9,691	11,826
	1.05	20,170	18,831	18,756	228,401	8	149	1,221	4,167	5,537
	1.10	21,267	15,611	15,604	239,553	9	182	1,309	2,898	4,389
4	1	19,438	21,941	21,836	321,614	19	409	2,103	5,156	7,668
	1.05	20,245	20,062	19,939	314,385	17	413	2,008	4,913	7,334
	1.10	21,307	16,486	16,399	239,219	10	218	1,536	5,339	7,093
5	1	19,763	26,071	25,910	222,557	9	146	1,305	3,557	5,007
	1.05	20,586	23,565	23,452	207,000	8	138	1,166	5,144	6,448
	1.10	21,690	16,650	16,603	208,589	9	172	1,323	3,528	5,022

Table 5: Impact of fleet size in IP-Heur (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 5$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

shortages observed at the time points). **VG-T**, **LP-T**, **IP-T**, and **TT** represent respectively the time (in seconds) spent in dynamic variable generation, solving the LP, solving the final IP, and the total run-time.

The result in Table 6 show that reducing the maximum time between two consecutive time points from 168 to 12 hours eliminates stock-outs (Instances 1, 2, and 5) and reduces empty repositioning miles (Instances 3 and 4) as the number of repositioning options has increased. Even though the number of empty repositioning variables generated increases by about 62%, this does not always imply an increase in total time, as a larger number of variables typically implies shorter IP solve time. We also observe that the difference in repositioning costs between using $\tau_M = 24$ and $\tau_M = 12$ is small, less than 1%, but that using $\tau_M = 12$ appears to be more efficient (although results differ on different instances). The results clearly suggest that there is no need to reduce the maximum time between consecutive time points even further.

The results in Table 7 show that enforcing a minimum time of one hour between two consecutive time points (i.e., only enforcing that inventory is monitored at least once every hour) greatly reduces the number of iterations (by about 30%) and the number of empty repositioning variables generated (by about 29%). This results in a reduction of total time of about 64%. It also reduces the empty repositioning costs (by about 3%), which is likely due to missing a few short periods of stock-outs (less than one hour). Given that in practice the variability in load and unload times is high (in the order of a few hours), it is reasonable to monitor the inventory using at least once an hour rather

INS.	τ_M	#TPT	OBJ	#P	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	168	26,110	26,482	1	278,450	25	1,311	4,298	21,070	26,679
	24	28,274	26,753	0	330,121	23	1,118	3,817	13,797	18,732
	12	34,503	26,593	0	438,856	31	1,114	4,413	9,029	14,556
2	168	25,993	26,050	1	282,064	20	819	3,901	20,048	24,767
	24	28,143	26,876	0	309,834	19	680	3,651	23,124	27,455
	12	34,357	26,700	0	433,501	23	597	3,165	6,728	10,490
3	168	25,930	22,873	0	286,696	22	1,181	4,024	24,448	29,652
	24	28,080	20,926	0	332,775	22	767	2,472	13,525	16,764
	12	34,307	20,669	0	426,588	26	788	3,696	9,367	13,851
4	168	25,876	24,081	0	251,744	20	791	3,050	14,487	18,328
	24	28,029	21,941	0	321,702	22	625	2,750	14,255	17,630
	12	34,264	21,886	0	419,037	29	1,616	6,465	13,908	21,989
5	168	25,949	25,549	1	220,819	17	677	2,842	13,427	16,946
	24	28,093	26,071	0	262,748	18	431	1,867	5,881	8,178
	12	34,313	25,655	0	408,998	29	1,959	6,275	12,420	20,655

Table 6: Value of maximum time-step τ_M in the discretization and its impact on the performance of IP-Heur (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

than more frequently.

5.3.2 Impact of enhanced variable generation schemes

As solving the LP relaxation represents a significant fraction of the total solution time, we have carefully designed the variable generation scheme. To evaluate the impact of the various ideas and techniques embedded in the variable generation schemes, we compare the efficiency of the three variable generation schemes as well as their impact on the quality of final IP solution (as the different schemes result in different sets of variables, the IP solutions may differ – as may the IP solution times). For ease of notation, we use B, E-B, and E-E-B to represent the Basic, Enhanced-Basic, and Efficient-Enhanced-Basic schemes respectively.

We incorporate one more technique to reduce the computation time of Enhanced-Basic: we terminate dynamic variable generation when the objective value has not changed for three consecutive iterations. In that case, it is likely that we have found the optimal LP objective value, but have not yet been able to prove it. This technique was already used by Gilmore and Gomory [1963] to deal with the tailing-off behavior of column generation schemes. Another option would be to compute a lower bound on the objective value, as suggested in Farley [1990], and terminate when the optimality gap drops below a threshold. However, in our setting Farley's bound is weak and only produces tight lower bounds in the last few iterations. Therefore, we opted for the simple

INS.	τ_m	#TPT	OBJ	#P	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	0	45,844	26,910	0	377,422	42	4,878	5,514	30,791	41,183
	0.5	28,274	26,753	0	330,121	23	1,118	3,817	13,797	18,732
	1	22,578	26,112	0	275,124	19	791	3,364	8,405	12,561
	2	17,544	25,508	0	208,341	15	388	2,225	8,333	10,946
2	0	45,593	26,931	0	390,788	25	1,845	5,466	27,815	35,126
	0.5	28,143	26,876	0	309,834	19	680	3,651	23,124	27,455
	1	22,525	26,478	0	296,670	20	694	3,439	9,555	13,687
	2	17,506	25,525	0	231,463	14	247	2,174	9,546	11,967
3	0	45,396	20,926	0	464,063	32	4,019	6,634	51,687	62,341
	0.5	28,080	20,926	0	332,775	22	767	2,472	13,525	16,764
	1	22,489	20,114	0	309,126	23	872	3,633	9,417	13,922
	2	17,495	19,917	0	224,790	15	363	2,239	7,049	9,651
4	0	45,333	22,004	0	389,565	29	3,304	7,252	24,596	35,151
	0.5	28,029	21,941	0	321,702	22	625	2,750	14,255	17,630
	1	22,436	21,297	0	259,833	20	624	3,316	12,187	16,127
	2	17,460	20,358	0	167,637	13	171	1,755	5,142	7,067
5	0	45,494	26,334	0	309,785	22	1,797	5,814	23,099	30,710
	0.5	28,093	26,071	0	262,748	18	431	1,867	5,881	8,178
	1	22,491	25,620	0	233,298	19	801	3,148	8,781	12,731
	2	17,452	25,056	0	162,416	14	268	2,465	5,778	8,511

Table 7: Value of minimum time-step τ_m in the discretization and its impact on the performance of IP-HEUR (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 6$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

cut-off rule.

We also include a variation of Enhanced-Basic, which we refer to as Enhanced-Basic-Relaxed (E-B-R), in which we start with Enhanced-Basic, but switch to Basic once the number of variables generated in an iteration drops below a threshold (20,000 in our experiments). The rationale behind this idea is that once only a relatively small number of variables is generated, diversity becomes less important and we no longer want to limit the search for negative reduced cost variables.

A summary of the results can be found in Table 8. The results clearly demonstrate the value of

INS.	SCHEME	IP-OBJ	LP-OBJ	#VAR	#ITER	VG-T	LP-T	IP-T	TT
1	B	26,759	26,604	486,192	22	61,331	2,107	6,212	69,650
	E-B	26,759	26,604	332,224	19	46,257	1,972	4,985	53,214
	E-B-R	26,759	26,604	379,114	13	34,829	2,319	8,203	45,351
	E-E-B	26,753	26,604	328,529	17	397	1,729	5,971	8,096
2	B	26,875	26,635	286,526	17	74,203	2,457	12,240	88,900
	E-B	26,876	26,635	370,579	15	38,828	1,815	7,893	48,535
	E-B-R	26,876	26,635	401,141	13	29,816	1,484	4,393	35,693
	E-E-B	26,876	26,635	283,467	10	168	1,407	14,820	16,394
3	B	20,926	20,875	478,398	28	85,886	2,283	9,086	97,255
	E-B	20,962	20,875	296,908	12	65,227	2,953	24,824	93,004
	E-B-R	20,962	20,875	335,564	12	29,322	1,284	5,744	36,350
	E-E-B	20,926	20,875	331,735	17	397	1,738	9,691	11,826
4	B	21,941	21,836	439,791	20	59,004	2,054	8,801	69,859
	E-B	21,971	21,836	264,828	14	49,033	2,447	7,310	58,790
	E-B-R	21,971	21,836	320,145	11	26,062	1,194	4,842	32,098
	E-E-B	21,941	21,836	321,614	19	409	2,103	5,156	7,668
5	B	26,060	25,910	331,972	24	68,607	2,483	6,966	78,057
	E-B	26,071	25,910	215,569	9	24,606	2,039	12,423	39,068
	E-B-R	26,071	25,910	304,651	9	16,226	1,828	5,667	23,721
	E-E-B	26,071	25,910	222,557	9	146	1,305	3,557	5,007

Table 8: Comparison of embedding the different variable generation schemes in IP-Heur (with default parameters $N_{IP} = 1,000,000$, $N_{iter} = 40,000$, $N_e = 5,000$, $N_f = 100$, $N_a = 5$, $Sort = True$, $Best = True$, $K_1 = 5,000$ and $K_2 = 10$).

exploiting dual information as the Efficient-Enhanced-Basic scheme is far more efficient than the Basic and Enhanced-Basic schemes. More specifically, we see that the use of the Efficient-Enhanced-Basic scheme reduces the total time by about 88% compared to Basic and about 82% compared to Enhanced-Basic. The difference is even more pronounced when we compare the time spent in variable generation as the Efficient-Enhanced-Basic scheme reduces this time by about 99.5% compared to Basic and 99.3% compared to Enhanced-Basic. Importantly, the IP objective values reached by the different schemes are similar (the maximum difference is less than

0.1% for all instances).

Ensuring diversification in the initial iterations (Enhanced-Basic-Relaxed) pays off and achieves the smallest number of iterations. As expected, for most instances the Basic scheme generated the largest number of variables.

In Figure 7, we present more detailed information about the solution process for Instance 5. We show for Basic, Enhanced-Basic-Relaxed and Efficient-Enhanced-Basic the objective

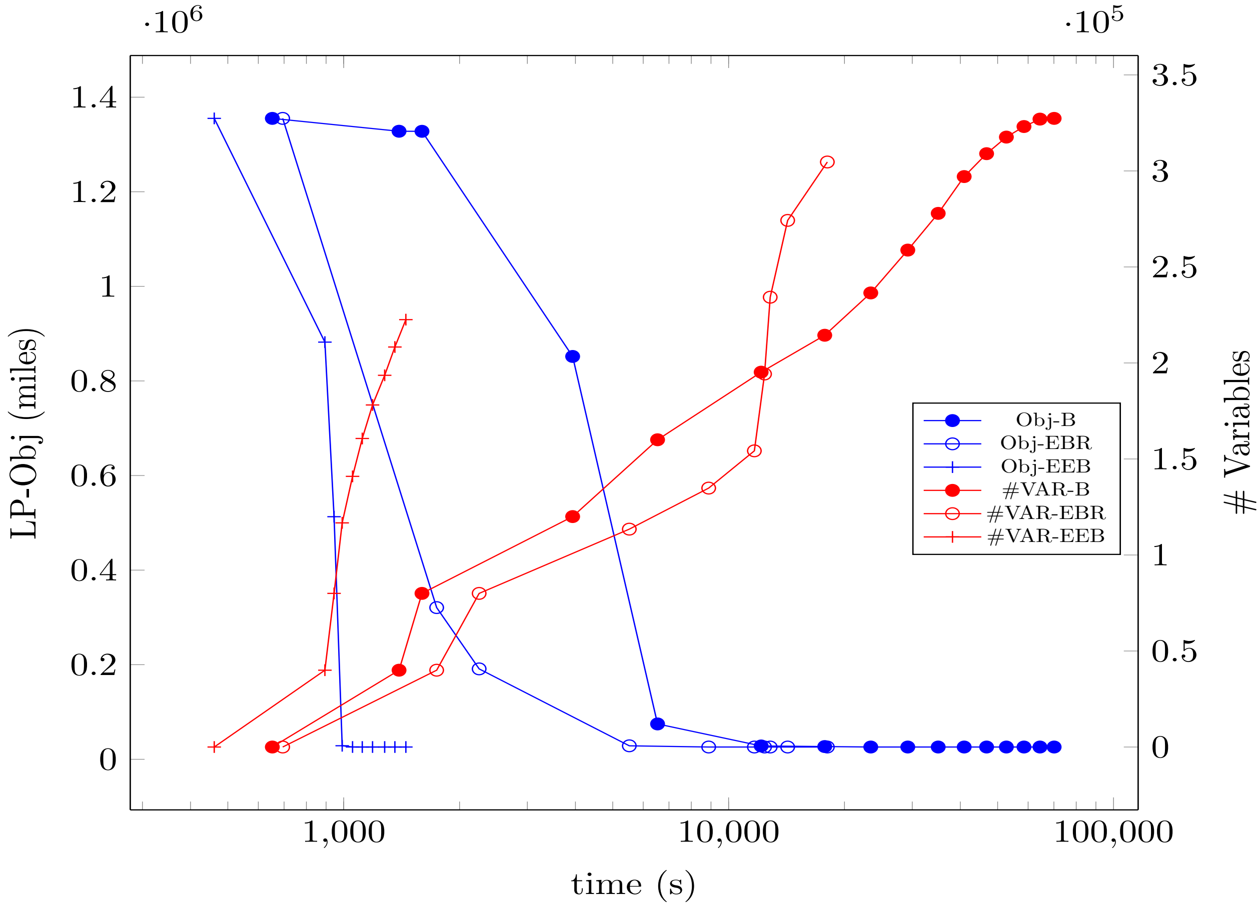

Figure 7: Comparison of the different variable generation schemes in terms of speed of convergence to the optimal objective value of the LP relaxation and the number of variables generated for Instance 5.

value and the number of variables generated at each iteration. The effectiveness of the Efficient-Enhanced-Basic scheme jumps out. The time per iteration is small and convergence to the optimal LP objective value is quick. It also generates fewer variables. (Note that we use a logarithmic scale on the horizontal axis, which obscures the large differences.)

5.3.3 Sensitivity analyses of dynamic variable generation

The Efficient-Enhanced-Basic variable generation scheme has many control parameters (mostly aimed at diversifying the set of variables generated). Here, we conduct a sensitivity analysis to better understand the effect of these parameters, where we focus on computation time and number of variables generated. As a baseline, we use the following configuration Efficient-Enhanced-Basic(40000,5000,100,5,True,True,ℓ,m). To assess the impact of different control parameters we use the following additional statistics:

- **AVG-CO**: average number of variables generated per iteration,
- **AVG-VG**: average generation time per iteration (in seconds),
- **AVG-LP**: average LP solve time per iteration (in seconds),
- **AVG-OBj**: average change in objective function value per iteration (as a percentage),
- **AVG-R**: average ratio of the number of variables generated and the number of variables examined (i.e., including variables with non-negative reduced cost) per iteration (as a percentage),
- **T-T**: total LP solve time (in seconds).

Value of Sorting We solve each instance with sorting enabled and sorting disabled. When sorting is disabled, a round robin scheme is used, as explained in Section 4.2, which also ensures some diversification. The results can be found in Table 9. We observe than when sorting is enabled, we

INS.	Sort	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OBJ	AVG-R	T-T
1	True	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	False	23	354,799	15,426	32.74	93.25	8.66	6.59	3,381
2	True	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	False	25	381,998	15,280	36.94	81.49	8.68	6.22	3,791
3	True	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	False	23	342,896	14,909	31.56	80.33	9.36	6.56	3,136
4	True	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	False	22	333,186	15,145	20.52	81.82	9.89	7.37	2,692
5	True	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	False	19	263,277	13,857	10.86	79.15	11.61	8.07	2,091

Table 9: Impact of sorting on the performance of the Efficient Enhanced Basic scheme.

generate fewer variables (about 6%) and take less time (about 15%).

Value of diversity We assess the value of the diversity created by limiting the number of variables generated for a single facility and a single arc, i.e., N_f and N_a. We compare combinations $(50, 3)$, $(100, 6)$, and $(200, 12)$. The results can be found in Table 10. We observe that when we relax

INS.	(N_f, N_a)	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	(50,3)	32	252,267	7,883	18.38	40.53	6.57	2.62	2,440
	(100,6)	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	(200,12)	26	398,096	15,311	21.93	60.64	8.47	7.19	2,379
2	(50,3)	27	267,990	9,926	30.12	83.86	8.12	3.47	4,333
	(100,6)	19	309,834	16,307	33.84	126.04	10.30	6.92	4,297
	(200,12)	24	365,650	15,235	36.03	103.47	8.69	7.75	4,061
3	(50,3)	30	264,969	8,832	40.45	79.01	7.22	3.05	4,586
	(100,6)	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	(200,12)	29	392,103	13,521	48.45	114.01	7.75	6.02	5,366
4	(50,3)	19	240,483	12,657	34.19	116.31	12.11	4.53	4,086
	(100,6)	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	(200,12)	29	389,674	13,437	35.55	115.33	7.95	7.40	5,226
5	(50,3)	15	192,347	12,823	12.85	71.12	15.19	5.65	1,650
	(100,6)	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	(200,12)	29	345,508	11,914	39.39	119.69	7.44	6.11	5,383

Table 10: Impact of diversity parameters N_f and N_a on the performance of the Efficient Enhanced Basic scheme.

enforcing diversity, i.e, $(N_f, N_a) = (200, 12)$, we generate more variables (about 57%) and increase solution time (about 52%) than when we favor diversity, i.e., $(N_f, N_a) = (50, 3)$.

Value of limits We assess the value of limiting the number of variables generated per iteration N_{iter} (so that new, hopefully more useful, dual information is obtained) and for an equipment type N_e (a high level mechanism to ensure diversity) We compare combinations $(8,000; 1,000)$, $(40,000; 5,000)$, $(80,000; 10,000)$, and $(120,000; 15,000)$. The results can be found in Table 11. We observe that generating too few variables per iteration has a negative effect on solution time (too many iterations), but so does generating too many variables per iteration (solving LP relaxations takes too long).

Value of Initialization Starting with an initial set of empty repositioning variables may result in more useful dual information early on in the solution process. Therefore, we compare starting without empty repositioning variables and starting with a set of initial empty repositioning variables. The approach and the results can be found in Appendix 1.

INS.	(N_{iter}, N_e)	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	(8,000;1,000)	32	118,282	3,696	18.53	103.92	9.53	7.69	5,029
	(40,000;5,000)	24	338,543	14,106	24.88	72.97	8.15	5.22	2,837
	(80,000;10,000)	19	388,553	20,450	25.97	52.42	11.50	5.00	1,710
	(120,000;15,000)	18	473,239	26,291	36.41	75.71	9.75	4.54	2,321
2	(8,000;1,000)	33	144,352	4,374	14.45	53.32	9.35	7.06	3,709
	(40,000;5,000)	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	(80,000;10,000)	19	408,557	21503	24.60	60.75	11.64	5.37	1,989
	(120,000;15,000)	17	470,995	27,706	30.97	82.70	10.41	4.92	2,402
3	(8,000;1,000)	28	115,072	4,110	7.35	25.68	10.87	8.20	1,294
	(40,000;5,000)	21	330,740	15,750	39.44	98.69	10.01	6.11	3,490
	(80,000;10,000)	21	416,621	19,839	25.31	55.14	10.60	4.77	1,893
	(120,000;15,000)	17	455,656	26,803	37.51	82.89	9.08	4.83	2,430
4	(8,000;1,000)	26	105,226	4,047	7.76	52.01	12.14	9.32	2,187
	(40,000;5,000)	25	303,779	12,151	18.78	78.76	8.85	5.73	2,898
	(80,000;10,000)	16	364,298	22,769	32.84	67.70	14.09	5.71	1,905
	(120,000;15,000)	16	441,432	27,590	34.75	95.21	10.19	5.04	2,458
5	(8,000;1,000)	19	92,184	4,852	6.63	55.45	16.75	12.89	1,772
	(40,000;5,000)	18	248,401	13,800	16.25	66.85	10.55	6.86	1,896
	(80,000;10,000)	16	334,042	20,878	17.58	56.83	13.96	5.96	1,423
	(120,000;15,000)	17	465,603	27,388	32.34	109.90	9.75	5.50	2,714

Table 11: Impact of limits N_{iter} and N_e on the performance of the Efficient Enhanced Basic scheme.

6 Conclusion and Final Remarks

In this paper, we have proposed an inventory-aware fleet management methodology that can be used by logistics companies that operate a heterogeneous fleet of trailers and containers. The operational planning problem of managing a fleet of multiple equipment types in a service network is formulated using a mixed-integer program with inventory variables and relies on substituting equipment types and adding empty repositioning movements to ensure the load plan is executable and equipment stock-outs are avoided (if possible) throughout the planning horizon. As company networks and equipment fleet sizes can be huge, the methodology uses a parsimonious discretization of time that is based on the load plan's departure and arrival times. This discretization enables control of the size of a time-expanded network model as inventory is monitored only at specific time points. To solve the model in a timely manner, we have proposed a dynamic variable generation approach (akin to a column generation approach) which generates the repositioning arcs (variables) as needed and ensures the computational tractability. In particular, the Efficient-Enhanced-Basic scheme stands out as the most efficient approach. This proposed approach employs heuristic ideas that make use of the structure of the problem to efficiently and dynamically generate empty repositioning variables. The methodology produces high quality, but not necessarily optimal, solutions in an acceptable amount of time.

We are currently exploring exact methods, using Benders decomposition techniques, to obtain optimal solutions.

7 Acknowledgment

We are grateful for the support of the Transportation Analytics and Operations Research team at *UPS Supply Chain Solutions*. We want to thank them for providing us with very insightful information to help with our questions regarding equipment fleet management in service networks.

References

A. Baykasoğlu, K. Subulan, A. S. Taşan, and N. Dudaklı. A review of fleet planning problems in single and multimodal transportation systems. *Transportmetrica A: Transport Science*, 15(2): 631–697, 2019.

M. Boile, S. Theofanis, A. Baveja, and N. Mittal. Regional repositioning of empty containers: Case for inland depots. *Transportation Research Record*, 2066(1):31–40, 2008.

B. D. Brouer, D. Pisinger, and S. Spoorendonk. Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Information Systems and Operational Research*, 49(2):109–124, 2011.

C. Çalışkan and R. W. Hall. A dynamic empty equipment and crew allocation model for long-haul networks. *Transportation Research Part A: Policy and Practice*, 37(5):405–418, 2003.

C. Çalişkan and R. W. Hall. A static empty equipment allocation model for long-haul networks with constrained crew routes. *IIE Transactions*, 38(11):947–954, 2006.

J. A. Carbajal, A. Erera, and M. Savelsbergh. Balancing fleet size and repositioning costs in ltl trucking. *Annals of Operations Research*, 203(1):235–254, 2013.

H. Chang, H. Jula, A. Chassiakos, and P. Ioannou. A heuristic solution for the empty container substitution problem. *Transportation Research Part E: Logistics and Transportation Review*, 44 (2):203–216, 2008.

T. G. Crainic, M. Gendreau, and P. Dejax. Dynamic and stochastic models for the allocation of empty containers. *Operations research*, 41(1):102–126, 1993.

P. J. Dejax and T. G. Crainic. Survey paper—a review of empty flows and fleet management models in freight transportation. *Transportation science*, 21(4):227–248, 1987.

Y. Du and R. Hall. Fleet sizing and empty equipment redistribution for center-terminal transportation networks. *Management Science*, 43(2):145–157, 1997.

A. L. Erera, J. C. Morales, and M. Savelsbergh. Robust optimization for empty repositioning problems. *Operations Research*, 57(2):468–483, 2009.

A. A. Farley. A note on bounding a class of linear programming problems, including cutting stock problems. *Operations Research*, 38(5):922–923, 1990.

P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem—part ii. *Operations research*, 11(6):863–888, 1963.

J. Gould. The size and composition of a road transport fleet. *Journal of the Operational Research Society*, 20(1):81–92, 1969.

R. W. Hall. Stochastic freight flow patterns: implications for fleet optimization. *Transportation Research Part A: Policy and Practice*, 33(6):449–465, 1999.

A. Imai and F. Rivera IV. Strategic fleet size planning for maritime refrigerated containers. *Maritime Policy & Management*, 28(4):361–374, 2001.

O. Jabali, M. Gendreau, and G. Laporte. A continuous approximation model for the fleet composition problem. *Transportation Research Part B: Methodological*, 46(10):1591–1606, 2012.

B. B. Oliveira, M. A. Carravilla, and J. F. Oliveira. Fleet and revenue management in car rental companies: A literature review and an integrated conceptual framework. *Omega*, 71:11–26, 2017.

R. A. Rushmeier and S. A. Kontogiorgis. Advances in the optimization of airline fleet assignment. *Transportation science*, 31(2):159–169, 1997.

R. A. Vasco and R. Morabito. The dynamic vehicle allocation problem with application in trucking companies in brazil. *Computers & Operations Research*, 76:118–133, 2016.

Y. Yang, Y. Ridouane, N. Boland, A. Erera, and M. Savelsbergh. Substitution-based equipment balancing in service networks with multiple equipment types. *European Journal of Operational Research*, 2021.

Appendix 1. Value of Initialization

We use a greedy heuristic to generate an initial set of repositioning variables that serve as a warm-start for column generation. The heuristic is summarized in Algorithm 5. The idea behind the heuristic is to iterate over all facilities that that would experience equipment stock-out if no adjustment is made in the load plan, then generate inbound repositioning arcs from facilities with sufficient inventory. The results can be found in Table 12. We observe that starting with an initial set of empty repositioning variables has few, if any, benefits; the solution time increases by about 5% (on average). In a real-life environment, where load plans do not change significantly from week to week, initializing with the set of empty repositioning movements performed in the preceding week may be beneficial.

INS.	$\mathcal{A}_1$	#ITER	#VAR	AVG-CO	AVG-VG	AVG-LP	AVG-OB	AVG-R	T-T
1	0	23	330,121	14,353	44.15	119.60	8.50	5.58	4,513
	938	24	334,320	13,930	53.55	113.46	9.17	4.83	4,960
2	0	20	317,790	15,890	18.15	63.50	10.26	6.17	2,271
	994	19	315,351	16,597	33.17	103.91	11.99	6.72	3,731
3	0	22	332,775	15,126	43.62	114.20	9.51	6.25	4,506
	1,713	22	324,951	14,771	49.04	100.93	9.05	5.39	4,193
4	0	22	321,702	14,623	23.99	92.93	9.30	6.58	3,064
	1,808	21	316,971	15,094	26.19	87.73	9.46	6.06	2,803
5	0	18	262,748	14,597	26.33	93.90	10.62	6.67	2,641
	2,164	21	270,507	12881	18.48	52.16	9.20	5.46	1,706

Table 12: Impact of initializing with the set of empty repositioning variables generated by Algorithm 5 with parameters $N_{iter} = 100,000$, $N_e = 10,000$, $N_f = 500$, $N_a = 10$, $Sort = True$, $\epsilon = 0.1$ on the performance of the Efficient Enhanced Basic scheme.

Algorithm 5: INITIALIZATION(N_{iter},N_e,N_f,N_a,$Sort$,ϵ)

$\mathcal{F}_1, \mathcal{E}_1 \leftarrow$ *unordered lists of facilities in the network and equipment categories*
$\mathcal{A}_1 \leftarrow \{\}$
if $Sort$ **then**
  $\mathcal{E}_1 \leftarrow$ *Equipment categories sorted by number of stock-outs in non-increasing order*
for ***each*** *equipment type* ***e*** *in* $\mathcal{E}_1$ **do**
  $\underline{I}_e \leftarrow$ *minimum of inventory level for each facility*
  $\mathcal{C}_e \leftarrow \{\}$
  if $Sort$ **then**
    $\mathcal{F}_1 \leftarrow$ *facilities sorted by* $\underline{I}_e$ *in non-decreasing order*
  for ***each*** *facility* ***i*** *in* $\mathcal{F}_1$ **do**
    if $\underline{I}_{ei} \geq 0 \, and \, Sort$ **then**
      break
    if $\underline{I}_{ei} \geq 0 \, and \, Sort = F$ **then**
      continue
    $Inbound[i] \leftarrow$ *unordered list of facilities* j *with arc* (j, i)
    if $Sort$ **then**
      $Inbound[i] \leftarrow$ *facilities* j *with arc* (j, i) *sorted by* D_{jie} *in non-decreasing order*
    $\mathcal{C}_f \leftarrow \{\}$
    $\mathcal{T}(i) \leftarrow$ *set of time-points at facility* i *in the order of time*
    for ***each*** *facility* ***j*** *in* $Inbound[i]$ **do**
      if $\underline{I}_{ej} < \epsilon * |\underline{I}_{ei}|$ **then**
        continue
      $\mathcal{C}_a \leftarrow \{\}$ // *list of at most* N_a *negative reduced cost timed arcs (sorted)*
      for ***each*** *time-point* ***t*** *in* $\mathcal{T}(i)$ **do**
        if $I_{iet} \geq 0$ **then**
          continue
        $a \leftarrow ((j, t_j), (i, t))$ // *available empty repositioning arc*
        $\mathcal{C}_a \leftarrow a$
        if $|\mathcal{C}_a| \geq N_a$ **then**
          break
      $\mathcal{C}_f \leftarrow \mathcal{C}_f \cup \mathcal{C}_a$
      if $|\mathcal{C}_f| \geq N_f$ **then**
        break
    $\mathcal{C}_e \leftarrow \mathcal{C}_e \cup \mathcal{C}_f$
    if $|\mathcal{C}_e| \geq N_e$ **then**
      break
  $\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup \mathcal{C}_e$
  if $|\mathcal{A}_1| \geq N$ **then**
    break
return $\mathcal{A}_1$