Short-Term Inventory-Aware Equipment Management in Service Networks

Yassine Ridouane∗1, Natasha Boland1, Alan Erera1, and Martin Savelsbergh1

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

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 by possibly substituting the equipment type assigned to loaded movements and by judiciously adding empty equipment repositioning movements. We analyze the complexity of the optimization model for different settings and introduce a heuristic based 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 can be achieved.

Keywords Equipment type · Inventory management · Empty repositioning · Time-expanded network · Dynamic variable 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

∗Corresponding author.
1UPS Fact Sheet 2020
operates more than 70 thousand vehicles. 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 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 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 generally made infrequently

---

2FedEx Fact Sheet 2020
(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, operational equipment planning 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 present a complexity analysis for specific settings in terms of the number of equipment types;
- 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 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]. 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., 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. An example of such an approach is [Brouer et al., 2011] who consider a liner shipping cargo allocation problem.

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.
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 (or en-route) 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 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 \(\eta\) denote the configuration matrix where rows represent configurations in \(\mathcal{C}\) and columns represent equipment types in \(\mathcal{E}\), then an entry \(\eta_{ce}\) represents the number of units of equipment type \(e\) in configuration \(c\). An example of \(\eta\) with three equipment types in \(\mathcal{E}\) and four configurations in \(\mathcal{C}\) is shown below:

\[
\eta = \begin{pmatrix}
c_1 & e_1 & e_2 & e_3 \\
c_2 & 1 & 0 & 0 \\
c_3 & 2 & 0 & 0 \\
c_4 & 0 & 1 & 0 \\
c_4 & 0 & 0 & 1 
\end{pmatrix}
\]

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 \(T^o(i)\) denote the set of times at which equipment starts being loaded at terminal \(i\).
  - \(d(l) \in \mathcal{F}\): The destination of load \(l\).
t^d(l) ∈ [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 terminal \( i \).

- \( q(l) ∈ \mathcal{C} \): The (initial) equipment configuration assigned to load \( l \).
- \( S_t ⊆ \mathcal{C} \): The set of allowable configurations for load \( l \).

- \( \mathcal{N} \): The set of nodes in the time-expanded network. Each node \((i, t) ∈ \mathcal{N}\) represents a facility \( i ∈ \mathcal{F} \) and a time \( t ∈ \mathcal{T}(i) \) representing the set of times for facility \( i \), i.e., \( \mathcal{T}(i) = \{ t : (i, t) ∈ \mathcal{N} \} \). The set \( \mathcal{T}(i) \) contains times 0 and \( T \) and all other \( t ∈ \mathcal{T}(i) \) have \( 0 < t < T \).

For convenience, for time point \( t ∈ \mathcal{T}(i) \setminus \{0, T\} \), we let \( t^- = \max\{s ∈ \mathcal{T}(i) : s < t\} \) be the preceding time point and \( t^+ = \min\{s ∈ \mathcal{T}(i) : s > t\} \) 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 \((i, t) ∈ \mathcal{N}\), respectively:

\[
\mathcal{L}^-_{(i, t)} = \{ l ∈ \mathcal{L} : d(l) = i, t^- ≤ t^o(l) < t \},
\]

\[
\mathcal{L}^+_{(i, t)} = \{ l ∈ \mathcal{L} : o(l) = i, t^- < t^d(l) ≤ t \}.
\]

- \( \mathcal{A} \): 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 \((i, t) ∈ \mathcal{N}\), 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.

- \( I_{i e} \): 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 \( I_{i e} \) 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 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. This constraint can be generalized to take safety stock considerations into account. Incorporating safety stock can help protect against execution uncertainty (e.g., load and unload times, load cancellation, ad-hoc movements, transit times, etc.). It is also possible to incorporate inventory limits at facilities, e.g., capturing limited yard space.

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

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

\[(I.AM) \quad \min \sum_{a \in A} \sum_{e \in E} D_{ae} u_{ae} \]  
\[s.t. \ s_{ie0} = I_{ie}, \quad i \in F, \ e \in E, \]  
\[s_{iet} = s_{iet}^- + \left( \sum_{l \in L_{i,t}^-} \sum_{c \in S_l} \eta_{c e} y_{lc} - \sum_{l \in L_{i,t}^+} \sum_{c \in S_l} \eta_{c e} y_{lc} \right) + \left( \sum_{a \in \delta_{i,t}^-} u_{ae} - \sum_{a \in \delta_{i,t}^+} u_{ae} \right), \quad (i, t) \in \mathcal{N}, \ t > 0, \ e \in \mathcal{E}, \]  
\[\sum_{c \in S_l} y_{lc} = 1, \quad l \in L, \]  
\[y_{lc} \in \{0, 1\}, \quad l \in L, \ c \in S_l, \]  
\[s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i, t) \in \mathcal{N}, \ e \in \mathcal{E}, \]  
\[u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in A, \ e \in \mathcal{E}, \]  

where \(D_{ae}\) represents the cost of executing an empty movement with equipment type \(e\) in \(\mathcal{E}\), on arc \(a\) in \(A\). 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 \(\bar{s}_{ieT}\) and \(\bar{s}^T_{ie}\) on the variables \(s_{ieT}\), i.e.,

\[s_{ieT}^T \leq s_{ieT} \leq \bar{s}_{ieT}^T. \]  

3.3 A polynomially solvable special case

In service networks with a homogeneous fleet equipment management, i.e., empty repositioning of equipment, can be modeled as a single commodity network flow problem and is therefore solvable in polynomial time. When the fleet is heterogeneous, however, equipment management becomes more difficult. This has been shown formally in [Yang et al.], which presents a complexity analysis of equipment balancing in a flat network with multiple equipment types. The problem becomes NP-hard when the fleet is comprised of three or more equipment types.

Here, we consider configuration matrices with two rows only (i.e., only two configurations are
allowed for the planned loads) and present a condition on the elements of the configuration matrix that allows the problem to be solved in polynomial time.

More specifically, let the configuration matrix be of the form

\[
\eta = \begin{pmatrix}
e_1 & e_2 & \cdots & e_{|E|} \\
e_2 & a_1 & a_2 & \cdots & a_{|E|} \\
e_2 & b_1 & b_2 & \cdots & b_{|E|}
\end{pmatrix}
\]

For the instances that pertain to logistics companies in the U.S., \( a_e \) and \( b_e \) take values between 0 and 3 (\( 0 \leq a_e, b_e \leq 3 \)). As there are two configurations only, we replace variables \( y_{lc} \) in the model with variables \( y_l \) such that:

\[
y_l = \begin{cases} 
1, & \text{if configuration } c_1 \text{ is used on load } l, \\
0, & \text{otherwise.}
\end{cases}
\]

Constraints can be simplified as follows:

\[
s_{iet} = s_{iet-} + \sum_{l \in E_{i,t}^-} (a_ey_l + b_e(1 - y_l)) - \sum_{l \in E_{i,t}^+} (a_ey_l + b_e(1 - y_l)) + \sum_{a \in \delta^-(i,t)} u_{ae} - \sum_{a \in \delta^+(i,t)} u_{ae}
\]

\[
= s_{iet-} + \sum_{l \in E_{i,t}^-} (a_e - b_e)y_l - \sum_{l \in E_{i,t}^+} (a_e - b_e)y_l + b_e(|E_{i,t}^-| - |E_{i,t}^+|) + \sum_{a \in \delta^-(i,t)} u_{ae} - \sum_{a \in \delta^+(i,t)} u_{ae}
\]

When \( a_e - b_e \in \{0, 1, -1\} \) for a given equipment type \( e \), the previous constraint can be written as a flow balance constraint. For example, in case \( a_e - b_e = 1 \), (e.g., \( (a_e, b_e) = (1, 0) \) or \( (a_e, b_e) = (2, 1) \)), we can rewrite the constraint as the following flow balance constraint at node \((i, t)\)

\[
\left( s_{iet} + \sum_{l \in E_{i,t}^+} y_l + \sum_{a \in \delta^+(i,t)} u_{ae} \right) - \left( s_{iet-} + \sum_{l \in E_{i,t}^-} y_l + \sum_{a \in \delta^-(i,t)} u_{ae} \right) = b_e(|E_{i,t}^-| - |E_{i,t}^+|)
\]

This shows that when the configuration matrix has two rows \( a \) and \( b \) such that \( a_e - b_e \in \{0, 1, -1\} \) for each equipment type \( e \) in \( E \), the problem is polynomially solvable as it can be modeled using a minimum cost flow formulation.

The presence of target inventories at the end of the horizon can increase the complexity of the model. In the appendix, we provide a theoretical example with target inventories that can be proven to be NP-hard by using a polynomial reduction from a partition problem.
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 $T(i)$ for $i \in F$ is an essential feature of the inventory-aware equipment management problem and affects two aspects of the model. First, the inventory of equipment at facilities needs to be evaluated at certain time points to avoid (or minimize) the risk of equipment stock outs. The larger the number of time points, the smaller the risk of stock-outs (as a stock-out can only occur between two consecutive time points), but the larger the number of time points, the larger the formulation. 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 evaluating equipment inventory at every load departure suffices to identify stock-outs, if any. However, we can do even better. At each facility $i$, we aggregate inbound and outbound loads in $L$ 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. Let the set of nodes of the time-expanded network, \( \mathcal{N} \), be formed by pairs \((i, t)\) with \(t\) the start loading time to \(l^\star\) 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 1 depicts an example of this aggregation. In the example, the set of time points at the terminal 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 terminal 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 \(\text{Inv}_{ie} : t \mapsto \text{Inv}_{ie}(t)\) denote the function that monitors the inventory of equipment type \(e\) at any time \(t\) in \([0, T]\). We want to prove that \(\text{Inv}_{ie}\) is a nonnegative function if and only if it is nonnegative at the time points in \(\mathcal{T}(i)\), i.e.:

\[
\forall t \in [0, T] \quad \text{Inv}_{ie}(t) \geq 0 \iff \forall t \in \mathcal{T}(i) \quad \text{Inv}_{ie}(t) \geq 0
\]

The direction \(\iff\) is trivial as any time-point \(t\) in \(\mathcal{T}(i)\) is in \([0, T]\). For the direction \(\iff\) 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 \(t \in [t_j, t^M]\) then \(\text{Inv}_{ie}(t) \geq \text{Inv}_{ie}(t_j) \geq 0\), and if \(t \in [t^M, t_{j+1}]\) then \(\text{Inv}_{ie}(t) \geq \text{Inv}_{ie}(t_{j+1}) \geq 0\).
• \( t \in [t_{|T(i)|-1}, t_{|T(i)|}] \): After \( t_{|T(i)|-1} \) there are only arriving loads. Hence, the inventory only increases after \( t_{|T(i)|-1} \). Thus, we have \( Inv_{ie}(t) \geq Inv_{ie}(t_{|T(i)|-1}) \geq 0. \)

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 \( (\tau_m) \) at which we enforce positive inventory and at less busy facilities we enforce a maximum time separation between time points \( (\tau_M) \), 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, \( A \), 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 (i.e., only a few additional empty equipment repositioning movements will be introduced). 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 IAM be \(\pi_{iet}, \beta_l,\) and \(\gamma_{lc}\), respectively. Then the dual problem is

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

s.t. \(\pi_{iet} - \pi_{iet^+} \leq 0, \forall (i,t) \in \mathcal{N}, 0 < t < T, e \in \mathcal{E},\)  
\(\pi_{iet} - \pi_{iet} - \pi_{iet} - D_{ae} \leq 0, \forall (i,t) \in \mathcal{N}, e \in \mathcal{E},\)  
\(\pi_{iet} - \pi_{iet'} \leq D_{ae}, \forall a = ((i,t) \rightarrow (j,t')) \in \mathcal{A}, e \in \mathcal{E},\)  
\(\eta_{ce}(\pi_{iet} - \pi_{jet'}) - \beta_l - \gamma_{lc} \leq 0, \quad l = ((i,t) \rightarrow (j,t')) \in \mathcal{L}, c \in S_l,\)  
\(\gamma_{lc} \geq 0, \quad l \in \mathcal{L}, c \in S_l,\)  
\(\pi_{iet}, \beta_l \text{ free} \quad (i,t) \in \mathcal{N}, t > 0, e \in \mathcal{E}.\)

Observe that Constraints (10) and (11) imply that the dual variables \(\pi_{iet}\) 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 pricing problem:

\[
\min_{(i,t), (j,t')} \min_{e \in \mathcal{E}, a \in \mathcal{A} \setminus \mathcal{A}_1} D_{ae} - \pi_{iet} + \pi_{jet'}
\]

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 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 $T(i)$. If the reduced cost of the associated repositioning arc variable is negative, it is added to the set $A_2$. 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

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

where $\lambda_e$ 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 $\lambda_{ei}$, defined similar to the quantity $\lambda_{e}$ 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 $\lambda_e$ 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 $N_{\text{iter}}$, $N_e$, $N_f$ 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 $\pi_{iet}$ are nonpositive and monotonically nondecreasing in $t \in T(i)$, i.e.,

$$\pi_{iet_1} \leq \pi_{iet_2} \leq \cdots \leq \pi_{iet_T} \leq 0.$$  \hspace{1cm} (17)

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

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 $\pi_{iet}$ and $D_{ae} - \pi_{jet'}$. As $\pi$ is nonpositive, the first part is always nonpositive and the second part is always nonnegative.

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

$$\bar{\pi}_{ei} = \min_{t \in T(i)} \pi_{iet}.$$
Algorithm 1: ENHANCED-BASIC($N_{\text{iter}},N_e,N_f,N_a,\text{Sort} , \text{Best}, \ell,m,\lambda$)

\[ \mathcal{F}_1, \mathcal{E}_1 \leftarrow \text{unordered lists of facilities in the network and equipment categories} \]
\[ \mathcal{C} \leftarrow \emptyset \]
\[ k_1 \leftarrow \ell + 1 \quad \text{// index of last equipment category searched in previous iteration} \]

if Sort then
    \[ \mathcal{E}_1 \leftarrow \text{Equipment categories sorted by } \lambda \text{ in non-increasing order} \]
    \[ k_1 \leftarrow 1 \]

for each equipment type $e = k_1, ..., |\mathcal{E}_1|$ in $\mathcal{E}_1$ do
    \[ \mathcal{C}_e \leftarrow \emptyset \]
    \[ k_2 \leftarrow m_e + 1 \quad \text{// index of last facility searched in previous iteration for equipment type } e \]
    if Sort then
        \[ \text{Facilities} \leftarrow \text{facilities sorted by } \lambda \text{ in non-increasing order} \]
        \[ k_2 \leftarrow 1 \]

for each facility $i = k_2, ..., |\mathcal{F}_1|$ in $\mathcal{F}_1$ do
    \[ T(i) \leftarrow \text{set of time-points at facility } i \text{ in the order of time} \]
    \[ \text{Inbound}[i] \leftarrow \text{unordered list of facilities } j \text{ with arc } (j,i) \]
    \[ \mathcal{C}_f \leftarrow \emptyset \]
    for each facility $j$ in Inbound$[i]$ do
        \[ \mathcal{C}_a \leftarrow \emptyset \]
        for each time-point $t$ in $T(i)$ do
            \[ a = (j,t_j) \rightarrow (i,t) \quad \text{// available empty repositioning arc} \]
            \[ \text{if } D_{ae} + \pi_{iet} - \pi_{jet} < 0 \text{ then} \]
            \[ \mathcal{C}_a \leftarrow a \]
            \[ \text{if } |\mathcal{C}_a| \geq N_a \text{ then} \]
            \[ \text{break} \]
        \[ \mathcal{C}_f \leftarrow \mathcal{C}_f \cup \mathcal{C}_a \]
        \[ \text{if } |\mathcal{C}_f| \geq N_f \text{ then} \]
        \[ \text{break} \]
    \[ \mathcal{C}_e \leftarrow \mathcal{C}_e \cup \mathcal{C}_f \]
    \[ \text{if } |\mathcal{C}_e| \geq N_e \text{ then} \]
    \[ \text{break} \]

if $|\mathcal{C}| > 0 \quad \& \quad \lambda_{e+1} = 0$ then
    \[ \text{break} \]
\[ \mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_e \]
if $|\mathcal{C}| \geq N_{\text{iter}}$ then
    \[ \text{break} \]
\[ \text{return } \mathcal{C} \]
Then, we can stop the enumeration as soon as we reach a facility with $\bar{\pi}_{ei} = 0$, 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

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

Then, we can stop the enumeration as soon as we reach an inbound arc with $\pi_{ei} + D_{(ji)e} - \bar{\pi}_{ej} > 0$. 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 $\bar{\pi}_{ie}$. 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 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 ($|C_t| < K_1$), 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 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 2: EFFICIENT-ENHANCED-BASIC($N_{iter}, N_e, N_f, N_a, Sort, Best, \ell, m$)

$\mathcal{F}_1, \mathcal{E}_1 \leftarrow$ unordered lists of facilities in the network and equipment categories
\n\begin{align*}
\mathcal{C} & \leftarrow \{\} \\
\text{if} & \quad \text{Sort} \quad \text{then} \\
\mathcal{E}_1 & \leftarrow \text{Equipment categories sorted by current objective cost in non-increasing order} \\
k_1 & \leftarrow 1 \\
\text{else} \\
k_1 & \leftarrow \ell + 1 \quad \text{// index of last equipment category searched in previous iteration} \\
\text{for each} & \quad \text{equipment type} \quad e = k_1, \ldots, |\mathcal{E}_1| \quad \text{in} \quad \mathcal{E}_1 \quad \text{do} \\
\bar{\pi}_e, \bar{\pi}_e & \leftarrow \text{minimum and maximum of dual variables} \quad \pi \quad \text{for each facility} \\
\mathcal{C}_e & \leftarrow \{\}, \quad r \leftarrow 0, \quad r_{prev} \leftarrow 0 \\
\text{if} & \quad \text{Sort} \quad \text{then} \\
\mathcal{F}_1 & \leftarrow \text{facilities sorted by} \quad \bar{\pi}_e \quad \text{in non-decreasing order} \\
k_2 & \leftarrow 1 \\
\text{else} \\
k_2 & \leftarrow m_e + 1 \quad \text{// index of last facility searched in previous iteration for} \quad e \\
\text{for each} & \quad \text{facility} \quad i = k_2, \ldots, |\mathcal{F}_1| \quad \text{in} \quad \mathcal{F}_1 \quad \text{do} \\
\text{if} & \quad \bar{\pi}_e = 0 \quad \text{then} \\
\quad & \text{continue} \\
\text{Inbound}[i] & \leftarrow \text{unordered list of facilities} \quad j \quad \text{with arc} \quad (j, i) \\
\text{if} & \quad \text{Sort} \quad \text{then} \\
\text{Inbound}[i] & \leftarrow \text{facilities} \quad j \quad \text{with arc} \quad (j, i) \quad \text{sorted by} \quad D_{ji} - \bar{\pi}_e \quad \text{in non-decreasing order} \\
\mathcal{C}_f & \leftarrow \{\} \\
\text{for each} & \quad \text{facility} \quad j \quad \text{in} \quad \text{Inbound}[i] \quad \text{do} \\
\text{if} & \quad \text{Sort} \quad \text{and} \quad D_{ji} + \bar{\pi}_e - \bar{\pi}_j > 0 \quad \text{then} \\
\quad & \text{break} \\
\text{if} & \quad \text{Sort} = \text{False} \quad \text{and} \quad D_{ji} + \bar{\pi}_e - \bar{\pi}_j > 0 \quad \text{then} \\
\quad & \text{continue} \\
\mathcal{C}_a & \leftarrow \{\} \quad \text{// list of at most} \quad N_a \quad \text{negative reduced cost timed arcs} \quad \text{(sorted)} \\
\mathcal{T}(i) & \leftarrow \text{set of time-points at facility} \quad i \quad \text{in the order of time} \\
\text{for each} & \quad \text{time-point} \quad t \quad \text{in} \quad \mathcal{T}(i) \quad \text{do} \\
\text{if} & \quad D_{ji} + \pi_{iet} \geq 0 \quad \text{then} \\
\quad & \text{break} \\
a & \leftarrow ((j, t_j), (i, t)) \quad \text{// available empty repositioning arc} \\
r & \leftarrow D_{ji} + \pi_{iet} - \pi_{jet} \quad \text{// reduced cost of arc} \quad a \\
\text{if} & \quad r = r_{prev} \quad \text{then} \\
\quad & \text{continue} \quad \text{// skipping arcs with the same reduced cost as the last one found} \\
r_{prev} & \leftarrow r \\
\text{if} & \quad r < 0 \quad \text{then} \\
\quad \mathcal{C}_a & \leftarrow a \\
\text{if} & \quad \text{Best} = \text{False} \quad \text{and} \quad |\mathcal{C}_a| \geq N_a \quad \text{then} \\
\quad & \text{break} \\
\mathcal{C}_f & \leftarrow \mathcal{C}_f \cup \mathcal{C}_a \\
\text{if} & \quad |\mathcal{C}_f| \geq N_f \quad \text{then} \\
\quad & \text{break} \\
\mathcal{C}_e & \leftarrow \mathcal{C}_e \cup \mathcal{C}_f \\
\text{if} & \quad |\mathcal{C}_e| \geq N_e \quad \text{then} \\
\quad & \text{break} \\
\mathcal{C} & \leftarrow \mathcal{C} \cup \mathcal{C}_e \\
\text{if} & \quad |\mathcal{C}| \geq N_{iter} \quad \text{then} \\
\quad & \text{break} \\
\text{return} & \quad \mathcal{C}
Algorithm 3: LP-HEUR($N_{LP}, K_1, K_2, N_{iter}, N_e, N_f, N_a, \text{Sort}, \text{Best}$)

\[\mathcal{LP} \leftarrow \text{LP relaxation of IAM model with an initial set of empty repositioning variables}\]

\[\text{Terminate} \leftarrow \text{False}\]

\[C_t \leftarrow \emptyset\]

\[t \leftarrow 0\]

\[\ell \leftarrow 0\]

\[m \leftarrow 0 \ (\text{vector of size } |E|)\]

\[\text{while } \text{Terminate} = \text{False do}\]

\[\text{Solve } \mathcal{LP} \text{ and retrieve values of dual variables}\]

\[C_t \leftarrow \text{PRICING-ALGORITHM($N_{iter}, N_e, N_f, N_a, \text{Sort, Best, } \ell, m$)}\]

\[\text{if } C_t = \emptyset \text{ then}\]

\[\text{break}\]

\[C \leftarrow C \cup C_t\]

\[\text{if } |C| \geq N_{LP} \text{ then}\]

\[\text{break}\]

\[\text{if } |C_t| < K_1 \& \ t > K_2 \text{ then}\]

\[\text{Switch to Primal Simplex when solving } \mathcal{LP}\]

\[t \leftarrow t + 1\]

\[\text{Update } \mathcal{LP} \text{ with new columns in } C_t\]

\[\text{return } C\]

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 $\alpha_{ie}^l$ and $\alpha_{ie}^u$ denote the dual variables associated with the minimum and maximum target inventory constraints respectively. Constraints (11) become

\[\pi_{ieT} + \alpha_{ie}^l - \alpha_{ie}^u \leq 0 \quad \forall (i, T) \in \mathcal{N}, e \in \mathcal{E}. \quad (18)\]

When maximum target inventory constraints are not present, we have

\[\pi_{ieT} \leq -\alpha_{ie}^l \leq 0 \quad \forall (i, T) \in \mathcal{N}, e \in \mathcal{E}, \quad (19)\]

which, because $\alpha_{ie}^l$ 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 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 2.
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:

$$A_{iet} = \sum_{l \in L^+} \sum_{c \in S_l} \eta_{lce} \bar{y}_{lc} - \sum_{l \in L^-} \sum_{c \in S_l} \eta_{lce} \bar{y}_{lc}.$$ 

More specifically, the resulting problem is:

$$\begin{align*}
\min & \sum_{a \in A} \sum_{e \in E} D_{ae} u_{ae} \\
\text{s.t. } & 
\left( s_{iet_1} + \sum_{a \in \delta^+(i,t)} u_{ae} \right) - \left( \sum_{a \in \delta^-(i,t)} u_{ae} \right) = I_{ie} - A_{iet_1}, \forall (i, t_1) \in N, e \in \mathcal{E}, \\
& 
\left. \left( s_{iet} + \sum_{a \in \delta^+(i,t)} u_{ae} \right) - \left( s_{iet^-} + \sum_{a \in \delta^-(i,t)} u_{ae} \right) = -A_{iet}, \forall (i, t) \in N, t > 1, e \in \mathcal{E}, \right) \\
& 
\left. \begin{array}{c}
s_{iet} \in \mathbb{Z}_{\geq 0}, \quad (i, t) \in N, e \in \mathcal{E}, \\
u_{ae} \in \mathbb{Z}_{\geq 0}, \quad a \in A, \; e \in \mathcal{E},
\end{array} \right)
\end{align*}$$

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 $I AMS$.

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($N_{LP}, K_1, K_2, N_{iter}, N_e, N_f, N_a, Sort, Best$)

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

5 Computational Study

We have conducted a set of computational experiments to demonstrate the value of the proposed inventory-aware equipment management model ($IAM$) 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 $IAM$ 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 terminals, customer locations, and other locations where equipment inventory is monitored, e.g., rail yards. Table 1 summarizes relevant characteristics of the instances.
<table>
<thead>
<tr>
<th>Instance</th>
<th># Active Facilities</th>
<th># Loads</th>
<th>Total Mies</th>
<th># Time Points</th>
<th>Fleet Size</th>
<th>Empty Legs (%)</th>
<th>Bobtail Legs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,152</td>
<td>181,165</td>
<td>34,145,280</td>
<td>28,274</td>
<td>19,491</td>
<td>33.10</td>
<td>13.70</td>
</tr>
<tr>
<td>2</td>
<td>1,149</td>
<td>180,375</td>
<td>33,948,206</td>
<td>28,143</td>
<td>19,554</td>
<td>33.28</td>
<td>13.80</td>
</tr>
<tr>
<td>3</td>
<td>1,149</td>
<td>179,527</td>
<td>33,858,084</td>
<td>28,029</td>
<td>19,438</td>
<td>33.09</td>
<td>13.64</td>
</tr>
<tr>
<td>4</td>
<td>1,147</td>
<td>180,834</td>
<td>34,286,951</td>
<td>28,093</td>
<td>19,763</td>
<td>32.85</td>
<td>13.49</td>
</tr>
<tr>
<td>5</td>
<td>1,147</td>
<td>182,867</td>
<td>34,841,019</td>
<td>28,167</td>
<td>20,238</td>
<td>32.23</td>
<td>13.08</td>
</tr>
<tr>
<td>6</td>
<td>1,149</td>
<td>185,385</td>
<td>36,531,351</td>
<td>28,681</td>
<td>20,939</td>
<td>31.77</td>
<td>12.80</td>
</tr>
<tr>
<td>7</td>
<td>1,149</td>
<td>189,136</td>
<td>36,788,322</td>
<td>28,664</td>
<td>20,626</td>
<td>31.00</td>
<td>12.47</td>
</tr>
<tr>
<td>8</td>
<td>1,149</td>
<td>191,092</td>
<td>37,636,547</td>
<td>28,803</td>
<td>21,219</td>
<td>30.82</td>
<td>12.36</td>
</tr>
</tbody>
</table>

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 (needed to ensure driver cycles). 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 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
<table>
<thead>
<tr>
<th>Equipment Id</th>
<th># Units</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77</td>
<td>0.40</td>
</tr>
<tr>
<td>2</td>
<td>748</td>
<td>3.84</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>0.08</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.02</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>0.11</td>
</tr>
<tr>
<td>6</td>
<td>12,541</td>
<td>64.34</td>
</tr>
<tr>
<td>7</td>
<td>609</td>
<td>3.12</td>
</tr>
<tr>
<td>8</td>
<td>183</td>
<td>0.94</td>
</tr>
<tr>
<td>9</td>
<td>73</td>
<td>0.37</td>
</tr>
<tr>
<td>10</td>
<td>294</td>
<td>1.51</td>
</tr>
<tr>
<td>11</td>
<td>487</td>
<td>2.50</td>
</tr>
<tr>
<td>12</td>
<td>542</td>
<td>2.78</td>
</tr>
<tr>
<td>13</td>
<td>3,897</td>
<td>19.99</td>
</tr>
</tbody>
</table>

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

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).

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

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 $K_2 = 10$.

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 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 IAM.

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

\[
\sum_{l \in L} \sum_{c \in S_l \setminus q(l)} y_{lc} \leq Cap
\]

(25)

to IAM, which limits the number of substitutions, and we vary the right hand side. More specifically, we solve the LP allowing no substitutions 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 3. For this case, we need 77,771 repo-
positioning 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., \( Cap = \infty \)).

![Figure 3: Relationship between the total repositioning cost required (in miles) and the limit on the number of substitutions allowed for Instance 2](image)

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 \( T.A.M \) in the first stage and minimize the number of equipment substitutions in the second stage 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

\[
\min \sum_{l \in \mathcal{L}} \sum_{c \in S_l \setminus \{c \neq q(l)\}} y_{lc}
\]

and forcing that the minimum repositioning costs found in the first stage do not change is achieved
by adding constraint
\[
\sum_{a \in A} \sum_{e \in E} D_{ae} u_{ae} \leq \Omega^* \tag{27}
\]
where \(\Omega^*\) is the objective value of 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 \(\eta \geq 1\), i.e.,
\[
\hat{I}_{ie} = \eta I_{ie},
\]
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%. 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%).

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

Table 5: Impact of fleet size in IP-HEUR (with default parameters \(N_{IP} = 1,000,000, N_{iter} = 40,000, N_e = 5,000, N_f = 100, N_a = 5, Sort = True, Best = True, K_1 = 5,000 and K_2 = 10\).
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, $\tau_m$, to be 30 minutes and vary the maximum time between two consecutive time points, $\tau_M$. Second, we fix the the maximum time between two consecutive time points, $\tau_M$, to be 24 hours and vary the minimum time between two consecutive time points, $\tau_m$. 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 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.

<table>
<thead>
<tr>
<th>INS.</th>
<th>$\tau_M$</th>
<th>$#TPT$</th>
<th>OBJ</th>
<th>$#P$</th>
<th>$#VAR$</th>
<th>$#ITER$</th>
<th>$VG-T$</th>
<th>$LP-T$</th>
<th>$IP-T$</th>
<th>$TT$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>168</td>
<td>26,110</td>
<td>26,482</td>
<td>1</td>
<td>278,450</td>
<td>25</td>
<td>1,311</td>
<td>4,298</td>
<td>21,070</td>
<td>26,679</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>28,274</td>
<td>26,753</td>
<td>0</td>
<td>330,121</td>
<td>23</td>
<td>1,118</td>
<td>3,817</td>
<td>13,797</td>
<td>18,732</td>
</tr>
<tr>
<td>12</td>
<td>34,503</td>
<td>26,593</td>
<td>0</td>
<td>438,856</td>
<td>31</td>
<td>1,114</td>
<td>4,413</td>
<td>9,029</td>
<td>14,556</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>168</td>
<td>25,993</td>
<td>26,050</td>
<td>1</td>
<td>282,064</td>
<td>20</td>
<td>819</td>
<td>3,901</td>
<td>20,048</td>
<td>24,767</td>
</tr>
<tr>
<td>24</td>
<td>28,143</td>
<td>26,876</td>
<td>0</td>
<td>309,834</td>
<td>19</td>
<td>680</td>
<td>3,651</td>
<td>23,124</td>
<td>27,455</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34,357</td>
<td>26,700</td>
<td>0</td>
<td>433,501</td>
<td>23</td>
<td>597</td>
<td>3,165</td>
<td>6,728</td>
<td>10,490</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>168</td>
<td>25,930</td>
<td>22,873</td>
<td>0</td>
<td>238,696</td>
<td>22</td>
<td>1,181</td>
<td>4,024</td>
<td>24,448</td>
<td>29,652</td>
</tr>
<tr>
<td>24</td>
<td>28,080</td>
<td>20,926</td>
<td>0</td>
<td>332,775</td>
<td>22</td>
<td>767</td>
<td>2,472</td>
<td>13,525</td>
<td>16,764</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34,307</td>
<td>20,669</td>
<td>0</td>
<td>426,588</td>
<td>26</td>
<td>788</td>
<td>3,696</td>
<td>9,367</td>
<td>13,851</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>168</td>
<td>25,876</td>
<td>24,081</td>
<td>0</td>
<td>251,744</td>
<td>20</td>
<td>791</td>
<td>3,050</td>
<td>14,487</td>
<td>18,328</td>
</tr>
<tr>
<td>24</td>
<td>28,029</td>
<td>21,941</td>
<td>0</td>
<td>321,702</td>
<td>22</td>
<td>625</td>
<td>2,750</td>
<td>14,255</td>
<td>17,630</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34,264</td>
<td>21,886</td>
<td>0</td>
<td>419,037</td>
<td>29</td>
<td>1,616</td>
<td>6,465</td>
<td>13,908</td>
<td>21,989</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>168</td>
<td>25,949</td>
<td>25,549</td>
<td>1</td>
<td>220,819</td>
<td>17</td>
<td>677</td>
<td>2,842</td>
<td>13,427</td>
<td>16,946</td>
</tr>
<tr>
<td>24</td>
<td>28,093</td>
<td>26,071</td>
<td>0</td>
<td>262,748</td>
<td>18</td>
<td>431</td>
<td>1,867</td>
<td>5,881</td>
<td>8,178</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34,313</td>
<td>26,555</td>
<td>0</td>
<td>408,998</td>
<td>29</td>
<td>1,959</td>
<td>6,275</td>
<td>12,420</td>
<td>20,655</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Value of maximum time-step $\tau_M$ 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 $K_2 = 10$).

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.

Table 7: Value of minimum time-step $\tau_m$ in the discretization and its impact on the performance of 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 $K_2 = 10$).

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

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
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 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 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
<table>
<thead>
<tr>
<th>INS</th>
<th>SCHEME</th>
<th>IP-OBJ</th>
<th>LP-OBJ</th>
<th>#VAR</th>
<th>ITER</th>
<th>VG-T</th>
<th>LP-T</th>
<th>IP-T</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>26,759</td>
<td>26,604</td>
<td>486,192</td>
<td>22</td>
<td>61,331</td>
<td>2,107</td>
<td>6,212</td>
<td>69,650</td>
</tr>
<tr>
<td></td>
<td>E-B</td>
<td>26,759</td>
<td>26,604</td>
<td>332,224</td>
<td>19</td>
<td>46,257</td>
<td>1,972</td>
<td>4,985</td>
<td>53,214</td>
</tr>
<tr>
<td></td>
<td>E-B-R</td>
<td>26,759</td>
<td>26,604</td>
<td>379,114</td>
<td>13</td>
<td>34,829</td>
<td>2,319</td>
<td>8,203</td>
<td>45,351</td>
</tr>
<tr>
<td></td>
<td>E-E-B</td>
<td>26,753</td>
<td>26,604</td>
<td>328,529</td>
<td>17</td>
<td>397</td>
<td>1,729</td>
<td>5,971</td>
<td>8,096</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>26,875</td>
<td>26,635</td>
<td>286,526</td>
<td>17</td>
<td>74,203</td>
<td>2,457</td>
<td>12,240</td>
<td>88,900</td>
</tr>
<tr>
<td></td>
<td>E-B</td>
<td>26,876</td>
<td>26,635</td>
<td>370,579</td>
<td>15</td>
<td>38,828</td>
<td>1,815</td>
<td>7,893</td>
<td>48,535</td>
</tr>
<tr>
<td></td>
<td>E-B-R</td>
<td>26,876</td>
<td>26,635</td>
<td>401,141</td>
<td>13</td>
<td>29,816</td>
<td>1,484</td>
<td>4,393</td>
<td>35,693</td>
</tr>
<tr>
<td></td>
<td>E-E-B</td>
<td>26,876</td>
<td>26,635</td>
<td>283,467</td>
<td>10</td>
<td>168</td>
<td>1,407</td>
<td>14,820</td>
<td>16,394</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>20,926</td>
<td>20,875</td>
<td>478,398</td>
<td>28</td>
<td>85,886</td>
<td>2,283</td>
<td>9,086</td>
<td>97,255</td>
</tr>
<tr>
<td></td>
<td>E-B</td>
<td>20,962</td>
<td>20,875</td>
<td>296,908</td>
<td>12</td>
<td>65,227</td>
<td>2,953</td>
<td>24,824</td>
<td>93,004</td>
</tr>
<tr>
<td></td>
<td>E-B-R</td>
<td>20,962</td>
<td>20,875</td>
<td>355,654</td>
<td>12</td>
<td>29,322</td>
<td>1,284</td>
<td>5,744</td>
<td>36,350</td>
</tr>
<tr>
<td></td>
<td>E-E-B</td>
<td>20,926</td>
<td>20,875</td>
<td>331,735</td>
<td>17</td>
<td>397</td>
<td>1,738</td>
<td>9,691</td>
<td>11,826</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>21,941</td>
<td>21,836</td>
<td>439,791</td>
<td>20</td>
<td>59,004</td>
<td>2,054</td>
<td>8,801</td>
<td>69,859</td>
</tr>
<tr>
<td></td>
<td>E-B</td>
<td>21,971</td>
<td>21,836</td>
<td>264,828</td>
<td>14</td>
<td>49,033</td>
<td>2,447</td>
<td>7,310</td>
<td>58,790</td>
</tr>
<tr>
<td></td>
<td>E-B-R</td>
<td>21,971</td>
<td>21,836</td>
<td>320,145</td>
<td>11</td>
<td>26,062</td>
<td>1,194</td>
<td>4,842</td>
<td>32,098</td>
</tr>
<tr>
<td></td>
<td>E-E-B</td>
<td>21,941</td>
<td>21,836</td>
<td>321,614</td>
<td>19</td>
<td>409</td>
<td>2,103</td>
<td>5,156</td>
<td>7,668</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>26,060</td>
<td>25,910</td>
<td>331,972</td>
<td>24</td>
<td>68,607</td>
<td>2,483</td>
<td>6,966</td>
<td>78,057</td>
</tr>
<tr>
<td></td>
<td>E-B</td>
<td>26,071</td>
<td>25,910</td>
<td>215,569</td>
<td>9</td>
<td>24,606</td>
<td>2,039</td>
<td>12,423</td>
<td>39,068</td>
</tr>
<tr>
<td></td>
<td>E-B-R</td>
<td>26,071</td>
<td>25,910</td>
<td>304,651</td>
<td>9</td>
<td>16,226</td>
<td>1,828</td>
<td>5,667</td>
<td>23,721</td>
</tr>
<tr>
<td></td>
<td>E-E-B</td>
<td>26,071</td>
<td>25,910</td>
<td>222,557</td>
<td>9</td>
<td>146</td>
<td>1,305</td>
<td>3,557</td>
<td>5,007</td>
</tr>
</tbody>
</table>

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 $K_2 = 10$).
generated the largest number of variables.

In Figure 4, 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 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, $\ell, m$). 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 that when sorting is enabled, we generate fewer variables (about 6%) and take less time (about 15%).

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

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

**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 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., $(N_f, N_a) = (50, 3)$.
## Table 10: Impact of diversity parameters $N_f$ and $N_a$ on the performance of the Efficient Enhanced Basic scheme.

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

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 (generated using Algorithm [5]). 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.
## Table 11: Impact of limits \( N_{\text{iter}} \) and \( N_e \) on the performance of the Efficient Enhanced Basic scheme.

<table>
<thead>
<tr>
<th>INS</th>
<th>((N_{\text{iter}}, N_e))</th>
<th>#ITER</th>
<th>#VAR</th>
<th>AVG-CO</th>
<th>AVG-VG</th>
<th>AVG-LP</th>
<th>AVG-OB</th>
<th>AVG-R</th>
<th>T-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(8,000;1,000)</td>
<td>32</td>
<td>118,282</td>
<td>3,696</td>
<td>18.53</td>
<td>103.92</td>
<td>9.53</td>
<td>7.69</td>
<td>5,029</td>
</tr>
<tr>
<td></td>
<td>(40,000;5,000)</td>
<td>24</td>
<td>338,543</td>
<td>14,106</td>
<td>24.88</td>
<td>72.97</td>
<td>8.15</td>
<td>5.22</td>
<td>2,837</td>
</tr>
<tr>
<td></td>
<td>(80,000;10,000)</td>
<td>19</td>
<td>388,553</td>
<td>20,450</td>
<td>25.97</td>
<td>52.42</td>
<td>11.50</td>
<td>5.00</td>
<td>1,710</td>
</tr>
<tr>
<td></td>
<td>(120,000;15,000)</td>
<td>18</td>
<td>473,239</td>
<td>26,291</td>
<td>36.41</td>
<td>75.71</td>
<td>9.75</td>
<td>4.54</td>
<td>2,321</td>
</tr>
<tr>
<td>2</td>
<td>(8,000;1,000)</td>
<td>33</td>
<td>144,352</td>
<td>4,374</td>
<td>14.45</td>
<td>53.32</td>
<td>9.35</td>
<td>7.06</td>
<td>3,709</td>
</tr>
<tr>
<td></td>
<td>(40,000;5,000)</td>
<td>20</td>
<td>317,790</td>
<td>15,890</td>
<td>18.15</td>
<td>63.50</td>
<td>10.26</td>
<td>6.17</td>
<td>2,271</td>
</tr>
<tr>
<td></td>
<td>(80,000;10,000)</td>
<td>19</td>
<td>408,557</td>
<td>21503</td>
<td>24.60</td>
<td>60.75</td>
<td>11.64</td>
<td>5.37</td>
<td>1,989</td>
</tr>
<tr>
<td></td>
<td>(120,000;15,000)</td>
<td>17</td>
<td>470,995</td>
<td>27,706</td>
<td>30.97</td>
<td>82.70</td>
<td>10.41</td>
<td>4.92</td>
<td>2,402</td>
</tr>
<tr>
<td>3</td>
<td>(8,000;1,000)</td>
<td>28</td>
<td>115,072</td>
<td>4,110</td>
<td>7.35</td>
<td>25.68</td>
<td>10.87</td>
<td>8.20</td>
<td>1,294</td>
</tr>
<tr>
<td></td>
<td>(40,000;5,000)</td>
<td>21</td>
<td>330,740</td>
<td>15,750</td>
<td>39.44</td>
<td>98.69</td>
<td>10.01</td>
<td>6.11</td>
<td>3,490</td>
</tr>
<tr>
<td></td>
<td>(80,000;10,000)</td>
<td>21</td>
<td>416,621</td>
<td>19,839</td>
<td>25.31</td>
<td>55.14</td>
<td>10.60</td>
<td>4.77</td>
<td>1,939</td>
</tr>
<tr>
<td></td>
<td>(120,000;15,000)</td>
<td>17</td>
<td>455,656</td>
<td>26,803</td>
<td>37.51</td>
<td>82.89</td>
<td>9.08</td>
<td>4.83</td>
<td>2,430</td>
</tr>
<tr>
<td>4</td>
<td>(8,000;1,000)</td>
<td>26</td>
<td>105,226</td>
<td>4,047</td>
<td>7.76</td>
<td>52.01</td>
<td>12.14</td>
<td>9.32</td>
<td>2,187</td>
</tr>
<tr>
<td></td>
<td>(40,000;5,000)</td>
<td>25</td>
<td>303,779</td>
<td>12,151</td>
<td>18.78</td>
<td>78.76</td>
<td>8.85</td>
<td>5.73</td>
<td>2,898</td>
</tr>
<tr>
<td></td>
<td>(80,000;10,000)</td>
<td>16</td>
<td>364,298</td>
<td>22,769</td>
<td>32.84</td>
<td>67.70</td>
<td>14.09</td>
<td>5.71</td>
<td>1,905</td>
</tr>
<tr>
<td></td>
<td>(120,000;15,000)</td>
<td>16</td>
<td>441,432</td>
<td>27,590</td>
<td>34.75</td>
<td>95.21</td>
<td>10.19</td>
<td>5.04</td>
<td>2,458</td>
</tr>
<tr>
<td>5</td>
<td>(8,000;1,000)</td>
<td>19</td>
<td>92,184</td>
<td>4,852</td>
<td>6.63</td>
<td>55.45</td>
<td>16.75</td>
<td>12.89</td>
<td>1,772</td>
</tr>
<tr>
<td></td>
<td>(40,000;5,000)</td>
<td>18</td>
<td>248,401</td>
<td>13,800</td>
<td>16.25</td>
<td>66.85</td>
<td>10.55</td>
<td>6.86</td>
<td>1,896</td>
</tr>
<tr>
<td></td>
<td>(80,000;10,000)</td>
<td>16</td>
<td>334,042</td>
<td>20,878</td>
<td>17.58</td>
<td>56.83</td>
<td>13.96</td>
<td>5.96</td>
<td>1,423</td>
</tr>
<tr>
<td></td>
<td>(120,000;15,000)</td>
<td>17</td>
<td>465,603</td>
<td>27,388</td>
<td>32.34</td>
<td>109.90</td>
<td>9.75</td>
<td>5.50</td>
<td>2,714</td>
</tr>
</tbody>
</table>
Algorithm 5: Initialization($N_{iter}, N_e, N_f, N_a, \text{Sort}, \epsilon$)

$F_1, E_1 \leftarrow$ unordered lists of facilities in the network and equipment categories

$A_1 \leftarrow \{\}$

if Sort then

$E_1 \leftarrow$ Equipment categories sorted by number of stock-outs in non-increasing order

for each equipment type $e$ in $E_1$ do

$I_e \leftarrow$ minimum of inventory level for each facility

$C_e \leftarrow \{\}$

if Sort then

$F_1 \leftarrow$ facilities sorted by $I_e$ in non-decreasing order

for each facility $i$ in $F_1$ do

if $I_{ei} \geq 0$ and Sort then

break

if $I_{ei} \geq 0$ 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

$C_f \leftarrow \{\}$

$T(i) \leftarrow$ set of time-points at facility $i$ in the order of time

for each facility $j$ in Inbound[i] do

if $I_{ej} < \epsilon * |I_e|$ then

continue

$C_a \leftarrow \{\}$ // list of at most $N_a$ negative reduced cost timed arcs (sorted)

for each time-point $t$ in $T(i)$ do

if $I_{iet} \geq 0$ then

continue

$a \leftarrow ((j,t_j),(i,t))$ // available empty repositioning arc

$C_a \leftarrow a$

if $|C_a| \geq N_a$ then

break

$C_f \leftarrow C_f \cup C_a$

if $|C_f| \geq N_f$ then

break

$C_e \leftarrow C_e \cup C_f$

if $|C_e| \geq N_e$ then

break

$A_1 \leftarrow A_1 \cup C_e$

if $|A_1| \geq N$ then

break

return $A_1$
### 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.

<table>
<thead>
<tr>
<th>INS</th>
<th>$A_1$</th>
<th>#ITER</th>
<th>AVG-CO</th>
<th>AVG-VG</th>
<th>AVG-LP</th>
<th>AVG-OB</th>
<th>AVG-R</th>
<th>T-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>23</td>
<td>14,353</td>
<td>44.15</td>
<td>119.60</td>
<td>8.50</td>
<td>5.58</td>
<td>4,513</td>
</tr>
<tr>
<td></td>
<td>938</td>
<td>24</td>
<td>13,930</td>
<td>53.55</td>
<td>113.46</td>
<td>9.17</td>
<td>4.83</td>
<td>4,960</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>20</td>
<td>15,890</td>
<td>63.50</td>
<td>10.26</td>
<td>6.17</td>
<td>2,271</td>
<td></td>
</tr>
<tr>
<td></td>
<td>994</td>
<td>19</td>
<td>16,597</td>
<td>103.91</td>
<td>11.99</td>
<td>6.72</td>
<td>3,731</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>22</td>
<td>15,126</td>
<td>144.20</td>
<td>9.51</td>
<td>6.25</td>
<td>4,506</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1,713</td>
<td>22</td>
<td>14,771</td>
<td>100.36</td>
<td>9.05</td>
<td>5.39</td>
<td>4,193</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>22</td>
<td>14,623</td>
<td>92.93</td>
<td>9.30</td>
<td>6.58</td>
<td>3,064</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1,808</td>
<td>21</td>
<td>15,094</td>
<td>87.37</td>
<td>9.46</td>
<td>6.06</td>
<td>2,803</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>18</td>
<td>14,597</td>
<td>93.90</td>
<td>10.62</td>
<td>6.67</td>
<td>2,641</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2,164</td>
<td>21</td>
<td>12881</td>
<td>52.16</td>
<td>9.20</td>
<td>5.46</td>
<td>1,706</td>
<td></td>
</tr>
</tbody>
</table>

6 Final Remarks

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. It relies on substituting equipment types and adding empty repositioning movements. As company networks and fleet sizes can be huge the methodology uses a parsimonious discretization of time and employs heuristic ideas to efficiently and dynamically generate empty repositioning variables. The methodology produces high quality, but not necessarily optimal, solutions. We are currently exploring the use of 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


Appendix. The General Case with Target Inventories

We consider the general case where the configuration matrix contains more than two rows and the model needs to satisfy a target inventory at the end of the horizon. This case can be proven to be difficult to solve through the following proposition:

**Proposition 2.** The problem of finding a feasible assignment of equipment configurations to loads to satisfy a non-negative inventory throughout the horizon and a final target inventory is NP-complete.

**Proof.** Transformation from Partition Problem, which is known to be NP-complete.

Partition Problem: Given a set of positive integer variables \( S = \{a_1, a_2, ..., a_n\} \), can we partition it into two subsets \( S_1 \) and \( S_2 \) such that the sum of the numbers in \( S_1 \) is equal to the sum of numbers in \( S_2 \)?

We create one instance of the inventory-aware model with two facilities \( F = \{1, 2\} \) and one equipment category \( A = \{e_1\} \). We use the following configuration matrix with \( n + 1 \) rows:

\[
\eta = \begin{pmatrix}
e_1 \\
c_1 \\
c_2 \\
\cdot \\
c_{n-1} \\
c_n \\
c_{n+1} \\0
\end{pmatrix}
\]

We consider a time horizon \([0, 2]\) and a set of loads \( L = \{l_1, l_2, ..., l_n\} \). Each load \( l_i \) departs from facility 1 at time 1, arrives at facility 2 at time 2, and the set of its eligible equipment configurations is \( S_{l_i} = \{c_i, c_{n+1}\} \). The initial inventories at facilities 1 and 2 are, respectively, \( I_{10} = \sum_{i=1}^{n} a_i \) and \( I_{20} = 0 \). The target inventories are \( s_{12} = s_{22} = \sum_{i=1}^{n} a_i / 2 \).

For each load \( l_i \), we assign a substitution variable \( y_{l_i} \) that is binary such that:

\[
y_{l_i} = \begin{cases} 
1, & \text{if configuration } c_i \text{ is used on load } l_i, \\
0, & \text{otherwise.}
\end{cases}
\]

We create a time expanded network with three nodes \( N = \{(1, 1), (1, 2), (2, 2)\} \) and we assume the set of empty repositioning arcs is empty. The set of inventory flow constraints can be written as follows:
\[ s_{11} = I_{10} - \sum_{i=1}^{n} a_i y_{l_i} \]
\[ s_{12} = s_{11} \]
\[ s_{22} = I_{20} + \sum_{i=1}^{n} a_i y_{l_i} \]
\[ s_{12} = \sum_{i=1}^{n} a_i / 2 \]
\[ s_{22} = \sum_{i=1}^{n} a_i / 2 \]

(28)

This entails that the model is feasible if and only if:

\[ \sum_{i=1}^{n} a_i y_{l_i} = \sum_{i=1}^{n} a_i / 2 \]

This proves that a Yes-instance of PARTITION PROBLEM yields a feasible instance of the inventory-aware model and vice-versa.

\[ \square \]