Optimizing Expeditionary Logistics: Dynamic Discretization for Fleet Management

Haden Boone, Mathieu Dahan
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA
{jboone31@gatech.edu, mathieu.dahan@isye.gatech.edu}

Abstract. We introduce the Expeditionary Logistics Network Design Problem (ELNDP), a new formulation for operational-level planning in expeditionary environments where multi-modal vehicle coordination is critical and penalties for unmet demand dominate transportation costs. ELNDP extends the classical Scheduled Service Network Design Problem by incorporating flexible commodity sourcing and heterogeneous vehicle capabilities, both essential in military logistics. We propose an iterative refinement algorithm based on dynamic discretization discovery (DDD) that iteratively constructs consolidation plans on partially time-expanded networks. Unlike the classical DDD framework, our approach overestimates arc travel times and introduces backward recovery arcs to compute relaxed solutions. We also develop a new procedure for eliminating illegal vehicle cycles arising from explicit vehicle management, and introduce acceleration techniques based on capacity factors and a multi-commodity maximum-flow heuristic. A case study on 53 realistic instances designed with the United States Marine Corps shows that our method increases demand fulfillment by 106.4% and reduces solve times by 28.7% on average compared to benchmark approaches. Finally, our analysis offers managerial insights on how multi-modal coordination, time-discretization granularity, and proactive vehicle repositioning can substantially enhance responsiveness and resource utilization in expeditionary logistics operations.

Key words: Emergency logistics, dynamic discretization discovery, multi-commodity maximum flow, service network design

1. Introduction

Expeditionary logistics operations involve vehicle routing and critical demand fulfillment—typically by military forces—in uncertain, contested, or undeveloped environments (National Research Council et al. 1999, Cheung 2017). These operations may support military objectives, facilitate the movement or evacuation of personnel, or deliver relief supplies to disaster-impacted areas (Bender et al. 2004, United States Marine Corps 2023). Such logistics planning requires the timely delivery of supplies over long distances (Caunhye et al. 2012) and plays a vital role in maintaining global stability and responding to crises efficiently (McKnight 2019, United States Marine Corps 2023).

Rapid deployment and sustainment of expeditionary logistics forces pose significant challenges, particularly due to the remoteness of these environments and the limited availability of local resources. This necessitates the coordinated management of a multi-modal logistics fleet over extended time horizons (Bradford 2006). Operational plans often involve routing passengers and multiple classes of supply to meet time-sensitive demands, where sources may be flexible or dynamically assigned (Joint Chiefs of Staff 2001). This results in a multi-commodity, capacitated vehicle routing problem—one of the most challenging classes of routing problems in the literature.

The scale and complexity of expeditionary logistics operations require careful planning to evaluate multiple courses of action quickly and effectively. However, current analysis methods used by United States Marine Corps (USMC) logisticians remain largely subjective and labor-intensive. While some studies have employed simulation and predictive analytics to forecast logistics requirements (Reith et al. 2016, de Castro et al. 2024), to our knowledge, no prior work has addressed operational-level planning and coordination of logistic assets in expeditionary contexts. Furthermore, although the scheduled service network design problem (SSNDP) has been extensively studied in commercial logistics, existing models often omit key features of expeditionary operations, such as multi-modal fleets and flexible sourcing requirements (Crainic et al. 2016). This work aims to fill this gap by *developing a scalable optimization approach for generating high-fidelity logistics consolidation plans in expeditionary environments.*

1.1. Contributions

This article makes the following contributions:

– We formulate the *Expeditionary Logistics Network Design Problem* (ELNDP), a new scheduled service network design model that captures critical features of military expeditionary operations, including multi-modal fleet management, flexible commodity sourcing, and the dominance of unmet-demand penalties over transportation costs. To our knowledge, this is the first SSNDP model studied for military expeditionary logistics.

– To solve the mixed-integer programming formulation of ELNDP, we develop an iterative refinement algorithm based on the dynamic discretization discovery (DDD) framework originally introduced by Boland et al. (2017). At each iteration, the algorithm computes a relaxed solution on a partially time-expanded network and attempts to convert it into a feasible solution with identical objective value, certifying optimality. If conversion fails, the network is refined to improve the relaxation. We extend the framework with a new procedure for identifying and eliminating illegal vehicle cycles that arise from explicit vehicle coordination, and we design a two-step heuristic based on multi-commodity maximum flows to generate high-quality feasible solutions that accelerate convergence.

– In contrast to previous DDD approaches, which rely on *underestimated* arc travel times, we adopt a conservative strategy that *overestimates* travel times and introduces backward recovery arcs to preserve path feasibility (Theorem 1). This yields two key benefits: (i) the dispatch problem used to convert relaxed solutions can be formulated with significantly fewer binary variables (only one per connector-independent recovery arc), and (ii) out-and-back trips can be implicitly modeled through capacity factors, reducing the occurrence of illegal vehicle cycles and substantially improving solution quality in early iterations.

– Through a computational study on 53 realistic instances across three scenarios designed with the United States Marine Corps, we demonstrate that our approach significantly outperforms benchmark methods. On average, it increases demand fulfillment by 106.4% and reduces solve times by 28.69% compared to a faithful adaptation of the original DDD algorithm from Boland et al. (2017). Our analysis provides managerial insights on how multi-modal coordination, vehicle repositioning, and finer time discretization can substantially enhance responsiveness and resource utilization in expeditionary logistics operations.

1.2. Related Work

In practice, most consolidation plans for expeditionary logistics scenarios are developed manually using heuristic approaches (United States Marine Corps 2023), often resulting in relatively simple operational plans. While reducing the decision-making burden in emergency environments is valuable, the absence of decision support tools can lead to biased or inefficient outcomes (Williams 2010, Joint Chiefs of Staff 2011). Prior work has introduced tools specifically designed to evaluate military expeditionary logistics plans using methodologies from simulation and queuing theory (Rogers et al. 2018, McConnell et al. 2021). Although effective at identifying bottlenecks, these tools require predetermined routes and schedules, limiting their ability to support the development of alternative courses of action. In contrast, our model leverages optimization techniques to jointly consider routing, scheduling, and sourcing decisions, enabling more complex and efficient operational plans.

The literature extensively studies emergency logistics problems arising from natural disasters (Caunhye et al. 2012, Anaya-arenas et al. 2014). Some models address scenarios similar to those in emergency logistics, involving multi-commodity critical demand satisfaction and multi-modal fleet management (Özdamar et al. 2004, Sheu 2007, Lin et al. 2011, Özdamar and Demir 2012), and even incorporate decomposition solution techniques (Fontem et al. 2016, Wang et al. 2021). However, these models primarily rely on heuristic approaches and lack provable optimality guarantees.

Designing logistics consolidation plans in expeditionary environments shares similarities with the service network design problem (SNDP) encountered by commercial trucking carriers (Crainic 2000, Wieberneit 2008, Bakir et al. 2021), where load planning and edge-based vehicle flow decisions are made. Traditional SNDP studies focus on determining dispatch frequencies along each flat network edge (Crainic and Roy 1988, Mangiaracina et al. 2015, Greening et al. 2023). However, these models are not applicable to expeditionary settings, which require deciding specific dispatch times and explicitly managing a constrained vehicle fleet. Although extensions of the SNDP with asset management have been studied (Andersen et al. 2009a), many assume flexible fleet sizes and positioning while requiring only flow balance (Lai and Lo 2004, Pedersen et al. 2009), or aim to generate cyclic route schedules that minimize cost (Andersen et al. 2009b, 2011).

Recent work in service network design demonstrates the effectiveness of time-expanded models—commonly referred to as scheduled service network design problems (SSNDPs)—for traditional minimum-fulfillment-cost vehicle routing problems (Steinzen et al. 2010, Wang et al. 2019a). However, solving the associated integer programs on the resulting time-expanded networks quickly becomes computationally intractable (Boland et al. 2019). To address this, several studies have proposed iterative refinement algorithms based on DDD to improve tractability (Boland et al. 2017, Scherr et al. 2020, Marshall et al. 2021, Hewitt 2022). These approaches, however, do not incorporate key characteristics of expeditionary logistics, such as asset management and flexible sourcing. Our methodology modifies and extends the framework of Boland et al. (2017) to account for these critical features.

Another recently proposed approach for solving SSNDPs involves a consolidation-based formulation in place of a time-expanded network, and has been shown to yield a stronger linear relaxation and reduced symmetry compared to time-space formulations (Hewitt 2023, Hewitt and Lehuédé 2023). However, this formulation requires each commodity to have a fixed origin and destination—precluding flexible sourcing—and assumes that vehicles are available as needed, omitting asset management. Although the authors propose a method for approximating asset management, it does not satisfy the stringent requirements of expeditionary logistics planning. Additionally, the formulation presumes that all commodities can reach their intended destinations by a prescribed deadline—an assumption that does not hold in expeditionary environments.

In contrast to commercial settings, expeditionary environments are characterized by the high cost of unmet demands relative to transportation costs, rendering many existing models either inapplicable or poorly performing. Most SSNDP literature assumes fixed sourcing decisions, whereas expeditionary logistics requires the ability to flexibly source commodities from multiple supply locations. While some SSNDP studies consider heterogeneous vehicles (Wang et al. 2019b), they typically vary only in terms of weight capacity and transportation cost, not in speed, range, or operational domain. Furthermore, whereas most models impose only weight-based vehicle capacity constraints, expeditionary operations require additional commodity-specific constraints—for example, passenger transport must account for both seating and cargo capacity. To our knowledge, we are proposing the first SSNDP model specifically for military expeditionary logistics operations.

2. The Expeditionary Logistics Network Design Problem (ELNDP)

In this section, we formally define the *Expeditionary Logistics Network Design Problem* (ELNDP), which captures the operational constraints and objectives faced by logistics planners in expeditionary environments. We present the modeling assumptions, time-space network structure, and mathematical formulation used to describe the movement of multi-commodity supplies via a heterogeneous fleet of vehicles under time-sensitive demand conditions.

2.1. Expeditionary Logistics Model

We consider an expeditionary logistics planner facing the urgent challenge of delivering diverse "commodities" ranging from personnel to critical equipment across a vast, infrastructure-sparse region under tight deadlines. These operations require coordinating a heterogeneous fleet of connectors (i.e., vehicles), each with unique capabilities and constraints, to fulfill time-sensitive demands originating from multiple, dynamically available supply locations.

Formally, let L be the set of locations in an expeditionary theater. Over a time horizon of T hours, the planner seeks to transport a set K of commodities across the network. Commodities are distributed across multiple locations and may become available at any point during the time horizon, allowing locations to be replenished repeatedly. Similarly, locations may make multiple requests for commodities to be delivered by specified deadlines.

Commodities can only be transported across the network using a dedicated multi-modal fleet of connectors managed by the planner. Let C denote the set of connector types (e.g., aircraft, ships, trucks), each characterized by operational attributes. A connector of type $c \in C$ has an aggregate capacity u^c (e.g., by weight or cargo volume) and commodity-specific capacities $q^{c,k}$ (e.g., passenger seating or fuel storage). Each commodity $k \in K$ requires w^k units of aggregate capacity for transport. Connector types also differ in their speeds and operational constraints. We represent these characteristics using a set A of connector-specific arcs, forming a flat network $G := (L, A)$. Each directed arc $(i, j, c) \in A$ indicates that connector type $c \in C$ can travel from location i to location j (e.g., both locations must have a port for sea connectors) and that the distance is within its operational range. Let $\tau^c_{i,j}$ denote the travel time for connector type c from location i to location j. Figure 1 illustrates the resulting logistics subnetworks for different connector types.

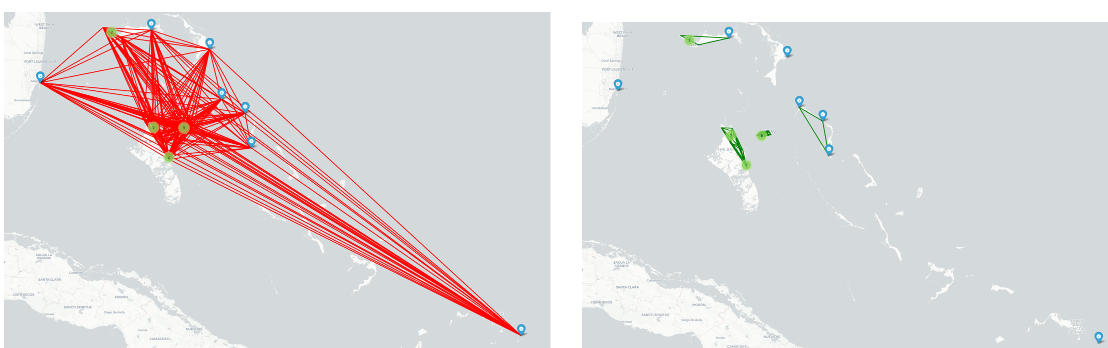


Figure 1 Example of permissible transportation movements for air connectors (left) and land connectors (right).

In expeditionary logistics, a key priority is to fulfill critical demands on time. Because military emergency logistics typically operate under sunk-cost assumptions, the planner's objective is to design a consolidation plan that maximizes on-time demand fulfillment.

2.2. Time-Expanded Networks

Although transportation costs are negligible in ELNDP, the operational urgency in expeditionary environments necessitates logistics plans that consolidate commodities effectively and dispatch connectors with high temporal precision. This can be achieved by discretizing the time horizon (Song and Furman 2013, Erera et al. 2013, Neumann-Saavedra et al. 2016) and modeling the movement of connectors and commodities on a *time-expanded network* (Ford and Fulkerson 1958), in which each physical location is replicated for every discrete time interval.

Let Δ denote the *rounding factor* of the time horizon—that is, the length of each decision time interval. Without loss of generality, we assume that $T = n \cdot \Delta$ (with $n \in \mathbb{Z}_{>0}$) and the set of discrete decision epochs is given by $\mathcal{T} \coloneqq \{\ell \cdot \Delta \mid \ell \in \{0, \ldots, n\}\}$. We then define the time-expanded version of the flat network G as $\widetilde{G} \coloneqq (\widetilde{L}, \widetilde{A} \cup \widetilde{H})$, where the node set $\widetilde{L} \coloneqq L \times \mathcal{T}$ contains all time-space locations, and time-space arcs are partitioned into movement arcs $\widetilde{A}$ and holdover arcs $\widetilde{H}$. For each arc $(i, j, c) \in A$ in the flat network and decision epoch $t \in \mathcal{T}$, we construct a time-space arc $((i,t),(j,t+\tau_{i,j}^{\Delta,c}),c) \in \widetilde{A}$ representing the movement of a connector of type $c \in C$ from the time-space location $(i,t) \in \widetilde{L}$ to the time-space location $(j, t+\tau_{i,j}^{\Delta,c}) \in \widetilde{L}$, where $\tau_{i,j}^{\Delta,c} \coloneqq \Delta \cdot \lceil \tau_{i,j}^{c}/\Delta \rceil$ is the travel time rounded up to the nearest multiple of Δ. In addition, for every location $i \in L$ and epoch $t \in \mathcal{T}$, we create a holdover arc $((i,t),(i,t+\Delta)) \in \widetilde{H}$ to represent the case where a commodity or connector remains stationary between time t and $t + \Delta$. Any time-space arc that leads beyond the time horizon is discarded. The resulting network $\widetilde{G}$, in which every location is duplicated across all epochs, is referred to as the *fully time-expanded network*. An example is shown in Figure 2.

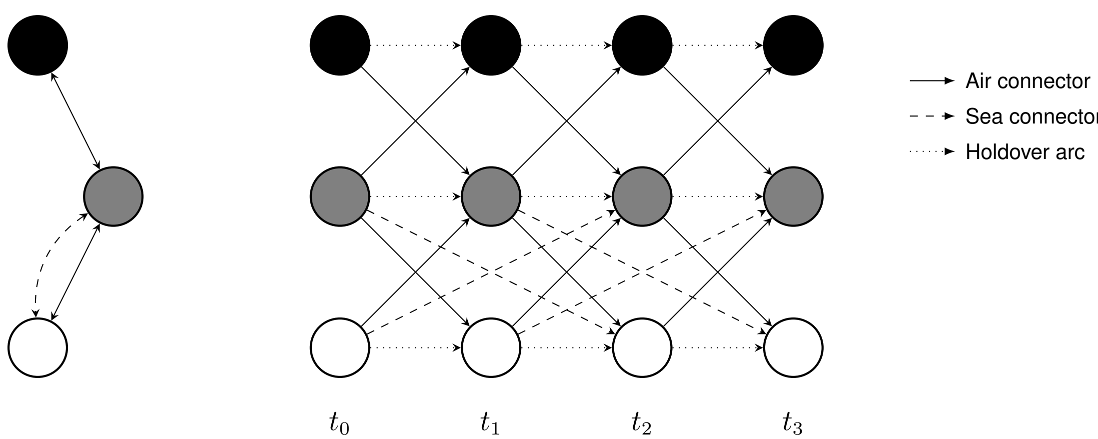


Figure 2 Illustration of a flat network (left) with three locations and arcs for air and sea connectors, and the corresponding fully time-expanded network (right) with four epochs. Air travel requires one epoch; sea travel requires two.

For ease of exposition, we use the notation $\tilde{\imath}$ to represent a time-space location in $\widetilde{L}$ associated with a geographic location $i \in L$, and let $t_{\tilde{\imath}} \in \mathcal{T}$ denote the decision epoch such that $\tilde{\imath} = (i, t_{\tilde{\imath}})$. We augment $\widetilde{G}$ by adding a dummy node $\tilde{\imath}_{end}$ to $\widetilde{L}$, together with time-space arcs $(\tilde{\imath}, \tilde{\imath}_{end}, c)$ in $\widetilde{A}$ for every connector type $c \in C$ and time-space location $\tilde{\imath} \in \widetilde{L}$ with $t_{\tilde{\imath}} = T$. This node represents the fictitious terminal location of connectors at the end of the time horizon. Finally, for each $\tilde{\imath} \in \widetilde{L}$ and $c \in C$, we define $\delta^{+}(\tilde{\imath}, c)$ (resp. $\delta^{-}(\tilde{\imath}, c)$) as the set of arcs in $\widetilde{A} \cup \widetilde{H}$ traversable by connector type c that originate from (resp. terminate at) $\tilde{\imath}$.

For each commodity $k \in K$, let $b^k_{\tilde{\imath}}$ denote the quantity of inventory that becomes available at time-space location $\tilde{\imath} \in \widetilde{L}$, computed by aggregating the inventory replenished at location $i \in L$ over the time interval $(t_{\tilde{\imath}} - \Delta, t_{\tilde{\imath}}]$. Similarly, let $d^k_{\tilde{\imath}}$ denote the demand associated with $\tilde{\imath}$, computed by aggregating the amount of commodity k requested at location $i \in L$ over the interval $[t_{\tilde{\imath}}, t_{\tilde{\imath}} + \Delta)$. We then define $\widetilde{I}^k := \{\tilde{\imath} \in \widetilde{L} \mid b^k_{\tilde{\imath}} > 0\}$ and $\widetilde{D}^k := \{\tilde{\imath} \in \widetilde{L} \mid d^k_{\tilde{\imath}} > 0\}$ as the sets of time-space locations in $\widetilde{G}$ where inventory becomes available and where demand is requested, respectively. For every commodity $k \in K$ and time-space demand location $\tilde{\imath} \in \widetilde{D}^k$, let $v^k_{\tilde{\imath}}$ denote the value of each unit of fulfilled commodity k at $\tilde{\imath}$; this parameter accounts for the criticality of each commodity, as well as the importance of each demand request.

Using the same conservative aggregation process, we define $n^c_{\tilde{\imath}}$ as the number of connectors of type $c \in C$ that become available at $\tilde{\imath}$, and $\widetilde{S}^c := \{\tilde{\imath} \in \widetilde{L} \mid n^c_{\tilde{\imath}} > 0\}$ as the set of time-space locations where connectors of type c become available.

2.3. Mixed-Integer Programming Formulation

We now formulate ELNDP as a mixed-integer program (MIP) on the fully time-expanded network $\widetilde{G}$. To simplify the formulation, we slightly abuse notation by expressing holdover arcs $(\tilde{\imath}, \tilde{\imath}') \in \widetilde{H}$—which are not connector-specific—as $(\tilde{\imath}, \tilde{\imath}', c)$ for every connector type $c \in C$ (equivalently, each holdover arc is duplicated for all connector types). We introduce decision variables $x^c_{\tilde{\imath},\tilde{\jmath}} \in \mathbb{Z}_{\geq 0}$ to represent the number of connectors of type $c \in C$ traversing arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A} \cup \widetilde{H}$. We also define flow variables $f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \geq 0$ for every commodity $k \in K$ on arcs $(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A} \cup \widetilde{H}$. Finally, we define continuous variables $\theta^k_{\tilde{\imath}} \geq 0$ for each demand time-space location $\tilde{\imath} \in \widetilde{D}^k$ of commodity $k \in K$, representing the amount of demand for commodity k that is satisfied at location i by deadline $t_{\tilde{\imath}}$. The MIP formulation of the fully-discretized model—hereafter referred to as FD—is given below:

$$\text{FD:} \quad \max_{x,f,\theta} \quad \sum_{k \in K} \sum_{\tilde{\imath} \in \widetilde{D}^k} v^k_{\tilde{\imath}} \cdot \theta^k_{\tilde{\imath}} \tag{1a}$$

$$\text{s.t.} \quad \sum_{(\tilde{\imath},\tilde{\jmath},c) \in \delta^+(\tilde{\imath},c)} x^c_{\tilde{\imath},\tilde{\jmath}} - \sum_{(\tilde{\jmath},\tilde{\imath},c) \in \delta^-(\tilde{\imath},c)} x^c_{\tilde{\jmath},\tilde{\imath}} = \begin{cases} n^c_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{S}^c, \\ -\sum_{\tilde{\jmath} \in \widetilde{S}^c} n^c_{\tilde{\jmath}} & \text{if } \tilde{\imath} = \tilde{\imath}_{end}, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall \tilde{\imath} \in \widetilde{L},\ \forall c \in C, \tag{1b}$$

$$\sum_{c \in C} \Big(\sum_{(\tilde{\imath},\tilde{\jmath},c) \in \delta^+(\tilde{\imath},c)} f^{c,k}_{\tilde{\imath},\tilde{\jmath}} - \sum_{(\tilde{\jmath},\tilde{\imath},c) \in \delta^-(\tilde{\imath},c)} f^{c,k}_{\tilde{\jmath},\tilde{\imath}} \Big) \begin{cases} \leq b^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{I}^k, \\ = -\theta^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{D}^k, \\ = 0 & \text{otherwise,} \end{cases} \qquad \forall \tilde{\imath} \in \widetilde{L},\ \forall k \in K, \tag{1c}$$

$$\sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \leq u^c \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \qquad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widetilde{A}, \tag{1d}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \leq q^{c,k} \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \qquad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widetilde{A},\ \forall k \in K, \tag{1e}$$

$$0 \leq \theta^k_{\tilde{\imath}} \leq d^k_{\tilde{\imath}}, \qquad \forall \tilde{\imath} \in \widetilde{D}^k,\ \forall k \in K, \tag{1f}$$

$$x^c_{\tilde{\imath},\tilde{\jmath}} \in \mathbb{Z}_{\geq 0}, \qquad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widetilde{A} \cup \widetilde{H}, \tag{1g}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \geq 0, \qquad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widetilde{A} \cup \widetilde{H},\ \forall k \in K. \tag{1h}$$

The objective function (1a) maximizes the total value of on-time demand fulfillment. Constraints (1b) maintain connector flow conservation at each time-space location, requiring all connectors to terminate at the dummy node $\tilde{\imath}_{end}$. Similarly, Constraints (1c) ensure flow balance for each commodity across the fully time-expanded network, accounting for both the inventory that becomes available and the demand fulfilled at each time-space location. Constraints (1d)–(1e) enforce, respectively, aggregate and commodity-specific capacity limits on each connector. Finally, Constraints (1f)–(1h) specify bounds and integrality conditions on the decision variables, ensuring that no more demand is satisfied than what is requested at any time-space location.

The MIP formulation FD is challenging to solve, as it is NP-hard (Crainic and Hewitt 2021) and grows substantially with finer time discretization (Boland et al. 2019). Solutions to FD provide dispatch times for every commodity and connector that traverses an arc, as well as opportunities for consolidation. Consequently, there is an inherent trade-off in choosing the time interval granularity: a smaller rounding factor Δ enables more precise dispatching and more accurate estimation of connector travel times, but leads to larger networks and increased computational burden. Conversely, a larger rounding factor yields a smaller, more tractable MIP but may miss consolidation opportunities and underestimate achievable deliveries. To balance solution quality and tractability, we propose a novel solution method based on the dynamic discretization discovery (DDD) algorithm, originally introduced in Boland et al. (2017), to solve ELNDP efficiently.

3. Dynamic Discretization Discovery in Expeditionary Environments

The DDD algorithm introduced in this section aims to iteratively solve relaxations of the FD formulation by starting from a smaller time-expanded network and progressively discovering new time-space locations to refine the solution. At each iteration, the algorithm attempts to construct a feasible solution with the same objective value as the relaxed solution; if successful, the algorithm terminates with a provably optimal solution to FD. While the original DDD algorithm of Boland et al. (2017) constructs relaxations using an optimistic approach that underestimates arc travel times, we instead propose a conservative formulation that *overestimates* arc travel times. To preserve key properties of the relaxation, we introduce path-maintaining recovery arcs that connect a time-space location to an earlier-timed copy. We present our relaxation model in Section 3.1 and detail the full DDD algorithm, including techniques to eliminate undesirable cyclic behaviors stemming from the explicit modeling of transportation asset management, in Section 3.2. We introduce algorithmic enhancements enabled by our conservative formulation in Section 3.3 and discuss the benefits of our proposed algorithm against a faithful adaptation of the original DDD algorithm in Section 3.4. We provide a detailed flowchart summarizing our algorithm with enhancements in Figure 6.

3.1. Partial Time Expansion with Recovery Arcs

To obtain relaxed solutions to FD by solving MIPs on considerably smaller networks than $\widetilde{G}$ at each iteration of our DDD algorithm, we propose a novel approach to partially expand the flat network $G = (L, A)$. We consider a subset $\widehat{L} \subseteq \widetilde{L}$ of time-space locations and a set $\widehat{A}$ of movement arcs that satisfy the following properties:

PROPERTY 1. *The subset of time-space locations satisfies* $\widehat{L} \supseteq \bigcup_{k \in K} \left(\widetilde{I}^k \cup \widetilde{D}^k \right) \cup \bigcup_{c \in C} \left(\widetilde{S}^c \right) \cup \widetilde{L}_{start} \cup \widetilde{L}_{end}$*, where* $\widetilde{L}_{start} \coloneqq \{(i,0) \mid i \in L\}$ *and* $\widetilde{L}_{end} \coloneqq \{(i,T) \mid i \in L\}$.

PROPERTY 2. *Every flat arc* $(i,j,c) \in A$ *has a timed copy in* $\widehat{A}$ *originating at each timed copy of* i *in* $\widehat{L}$*, except if it exceeds the time horizon.*

PROPERTY 3. *Every movement arc* $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$ *satisfies* $t_{\tilde{\jmath}} = \min\{t_{\tilde{\jmath}'} \mid \tilde{\jmath}' \in \widehat{L}$ *and* $t_{\tilde{\jmath}'} \geq t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j}\}$.

Property 1 states that the time-space location subset $\widehat{L}$ must include every commodity and connector origin, every commodity demand destination, and the first and last timed copy of each location. Property 2 ensures that if a connector of type $c \in C$ can travel from $i \in L$ to $j \in L$ in the flat network G, then it can travel from any timed copy $\tilde{\imath}$ of i in $\widehat{L}$ to a timed copy of j in $\widehat{L}$ via a movement arc in $\widehat{A}$, except if $t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j} > T$. Finally, Property 3 ensures that movement arcs in $\widehat{A}$ only contain the shortest time-space arcs that overestimate the travel time of the associated flat arcs, given the subset of time-space locations $\widehat{L}$. We note that Property 3 differs from Boland et al. (2017), where the authors generate movement arcs that are "too short", i.e., that *underestimate* travel times.

Given the subset of time-space locations $\widehat{L}$, we generalize the definition of holdover arcs $\widehat{H}$ as the set of time-space arcs $(\tilde{\imath}, \tilde{\imath}')$ connecting two consecutive timed copies $\tilde{\imath} \in \widehat{L}$ and $\tilde{\imath}' \in \widehat{L}$ of each location $i \in L$. Finally, to construct a network that provides a valid relaxed solution to FD, we must ensure that any feasible consolidation plan for FD is feasible in the relaxed network. Necessarily for every commodity $k \in K$, any time-space path from an origin $\tilde{\imath} \in \widetilde{I}^k$ to a destination $\tilde{\jmath} \in \widetilde{D}^k$ must have an associated path in the relaxed network to visit the same geographic locations in the same order using the same collection of connectors. However, this is currently not guaranteed due to the overestimation of travel times by the set of movement arcs $\widehat{A}$. As a result, we introduce the concept of *recovery arcs* that allow connectors and commodities to "travel back in time" whenever they traverse an arc that overestimates travel time. Let $\widehat{R}$ be the set of recovery arcs, which satisfies the following property:

PROPERTY 4. *There exists a recovery arc* $(\tilde{\jmath}', \tilde{\jmath}) \in \widehat{R}$ *with* $t_{\tilde{\jmath}} = \max\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L}$ *and* $t_{\tilde{\jmath}^\dagger} < t_{\tilde{\jmath}'}\}$ *if and only if there exists a movement arc* $(\tilde{\imath}, \tilde{\jmath}', c) \in \widehat{A}$ *with* $t_{\tilde{\jmath}'} > t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j}$.

Property 4 states that a recovery arc that connects a time-space location $\tilde{\jmath}' \in \widehat{L}$ to a precedent timed copy $\tilde{\jmath}$ is constructed whenever there exists a movement arc ending at $\tilde{\jmath}'$ that overestimates its travel time. We say that any network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$ that satisfies Properties 1-4 is a *partially time-expanded network* and show an illustration in Figure 3.

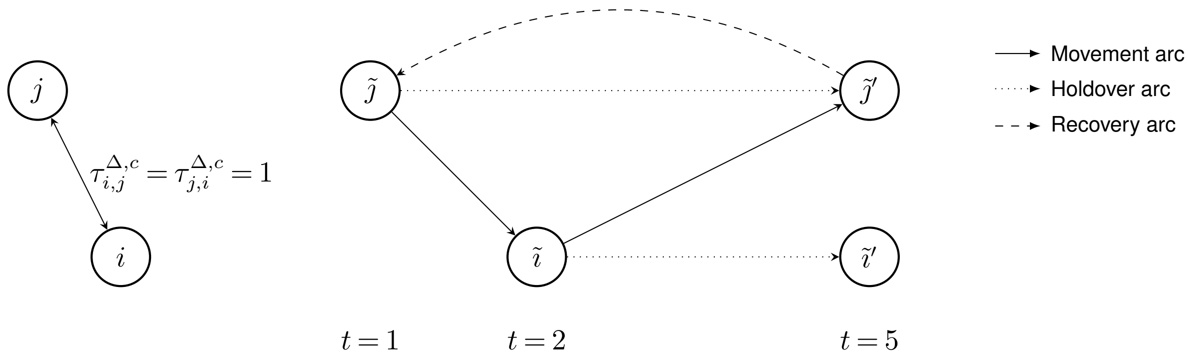


Figure 3 Illustration of a flat network (left) and a partially time-expanded network (right). A recovery arc $(\tilde{\jmath}', \tilde{\jmath})$ is added since the movement arc from $\tilde{\imath}$ to $\tilde{\jmath}'$ overestimates its travel time.

Next, we generalize the MIP formulation FD to generate a consolidation plan on the partially time-expanded network $\widehat{G}$. Analogously to $\widetilde{G}$, we augment $\widehat{G}$ by adding the connectors' fictitious terminal time-space location $\tilde{\imath}_{end}$, reachable from each node in $\widetilde{L}_{end}$. For each $\tilde{\imath} \in \widehat{L}$ and $c \in C$, we also extend the definition of $\delta^+(\tilde{\imath}, c)$ (resp. $\delta^-(\tilde{\imath}, c)$) to represent the set of arcs in $\widehat{A} \cup \widehat{H} \cup \widehat{R}$ traversable by connector type c that originate from (resp. terminate at) $\tilde{\imath}$. We then similarly extend the connector and commodity variables $x^c_{\tilde{\imath},\tilde{\jmath}} \in \mathbb{Z}_{\geq 0}$ and $f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \geq 0$ for every connector type $c \in C$, commodity $k \in K$, and arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R}$. We use the same commodity fulfillment variables $\theta^k_{\tilde{\imath}} \geq 0$ for every commodity $k \in K$ and time-space demand location $\tilde{\imath} \in \widetilde{D}^k$. The resulting MIP formulation, denoted SND($\widehat{G}$), is given as follows:

$$\text{SND}(\widehat{G}): \max_{x,f,\theta} \quad \sum_{k \in K} \sum_{\tilde{\imath} \in \widetilde{D}^k} v^k_{\tilde{\imath}} \cdot \theta^k_{\tilde{\imath}} \tag{2a}$$

$$\text{s.t.} \sum_{(\tilde{\imath},\tilde{\jmath},c) \in \delta^+(\tilde{\imath},c)} x^c_{\tilde{\imath},\tilde{\jmath}} - \sum_{(\tilde{\jmath},\tilde{\imath},c) \in \delta^-(\tilde{\imath},c)} x^c_{\tilde{\jmath},\tilde{\imath}} = \begin{cases} n^c_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{S}^c, \\ -\sum_{\tilde{\jmath} \in \widetilde{S}^c} n^c_{\tilde{\jmath}} & \text{if } \tilde{\imath} = \tilde{\imath}_{end}, \\ 0 & \text{otherwise,} \end{cases} \quad \forall \tilde{\imath} \in \widehat{L}, \forall c \in C, \tag{2b}$$

$$\sum_{c \in C} \Big(\sum_{(\tilde{\imath},\tilde{\jmath},c) \in \delta^+(\tilde{\imath},c)} f^{c,k}_{\tilde{\imath},\tilde{\jmath}} - \sum_{(\tilde{\jmath},\tilde{\imath},c) \in \delta^-(\tilde{\imath},c)} f^{c,k}_{\tilde{\jmath},\tilde{\imath}} \Big) \begin{cases} \leq b^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{I}^k, \\ = -\theta^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{D}^k, \\ = 0 & \text{otherwise,} \end{cases} \quad \forall \tilde{\imath} \in \widehat{L}, \forall k \in K, \tag{2c}$$

$$\sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \leq u^c \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \quad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A}, \tag{2d}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \leq q^{c,k} \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \quad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A}, \forall k \in K, \tag{2e}$$

$$0 \leq \theta^k_{\tilde{\imath}} \leq d^k_{\tilde{\imath}}, \quad \forall \tilde{\imath} \in \widetilde{D}^k, \forall k \in K, \tag{2f}$$

$$x^c_{\tilde{\imath},\tilde{\jmath}} \in \mathbb{Z}_{\geq 0}, \quad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R}, \tag{2g}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \geq 0, \quad \forall (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R}, \forall k \in K. \tag{2h}$$

The SND($\widehat{G}$) model only differs from FD in that it routes connectors and commodities on the partially time-expanded network $\widehat{G}$ as opposed to the fully time-expanded network $\widetilde{G}$. Although SND($\widehat{G}$) is in practice solved on a network with significantly fewer nodes and arcs, we must show that it generates a relaxed solution to FD before designing the DDD algorithm. This includes validating that any partially time-expanded network $\widehat{G}$ maintains feasibility of time-space paths from the fully time-expanded network $\widetilde{G}$. Formally, we define an $\tilde{\imath} - \tilde{\jmath}$ walk in any network as a sequence of directed arcs that connect adjacent nodes, originating at $\tilde{\imath}$ and terminating at $\tilde{\jmath}$. An $\tilde{\imath} - \tilde{\jmath}$ path is an $\tilde{\imath} - \tilde{\jmath}$ walk that traverses distinct nodes, and a cycle is a walk in which only the first and last nodes are equal. We then show the desired relaxation result in the following theorem, proven in Section EC.1:

THEOREM 1. *For any partially time-expanded network $\widehat{G}$, SND($\widehat{G}$) is a relaxation of FD.*

Theorem 1 establishes that solving SND($\widehat{G}$) on a partially time-expanded network yields a relaxed solution of FD. This result is enabled by the recovery arcs, which allow connector and commodity paths in $\widetilde{G}$ to be represented as walks in $\widehat{G}$ that visit the same locations in the same order while preserving commodity consolidations. Consequently, the theorem provides the foundation for the design of our DDD algorithm to solve ELNDP.

3.2. Dynamic Discretization Discovery Algorithm

We now describe our proposed DDD algorithm. It begins by constructing the smallest partially time-expanded network $\widehat{G}$ that satisfies Properties 1–4. At each iteration, the algorithm solves SND($\widehat{G}$), which—by Theorem 1—provides an upper bound on FD. Given a solution to SND($\widehat{G}$), the algorithm attempts to construct a feasible solution to FD with identical objective value. However, the recovery arcs used to define partially time-expanded networks may allow SND($\widehat{G}$) solutions that route connectors along walks that artificially inflate transportation capacity. To address this, the algorithm identifies such undesirable connector walks and eliminates them by augmenting $\widehat{G}$ with newly discovered time-space locations. Once the solution to SND($\widehat{G}$) only contains desirable connector walks, a MIP is used to assign connector dispatch times and convert the SND($\widehat{G}$) solution into a feasible FD solution. If the resulting FD solution attains the optimal value of SND($\widehat{G}$), the algorithm terminates. Otherwise, it identifies additional time-space locations to refine $\widehat{G}$ and continues iterating. We next detail each step of the DDD algorithm.

3.2.1. Initial Network Construction. To satisfy Property 1, we construct the initial partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$ by setting $\widehat{L} = \bigcup_{k\in K} \left(\widetilde{I}^k \cup \widetilde{D}^k\right) \cup \bigcup_{c\in C} \left(\widetilde{S}^c\right) \cup \widetilde{L}_{start} \cup \widetilde{L}_{end}$. That is, we only include commodity origin and demand-destination time-space locations, connector origins, and the first and last timed copies of each geographic location. We then create the set of holdover arcs $\widehat{H}$ by connecting consecutive timed copies of each location. For each flat arc $(i,j,c) \in A$ and each timed copy $\tilde{\imath} \in \widehat{L}$ of i, we add a time-space arc $(\tilde{\imath}, \tilde{\jmath}', c)$ to $\widehat{A}$ satisfying $t_{\tilde{\jmath}'} = \min\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L} \text{ and } t_{\tilde{\jmath}^\dagger} \geq t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j}\}$, provided it does not exceed the time horizon. Furthermore, if $t_{\tilde{\jmath}'} > t_{\tilde{\imath}} + \tau^{\Delta}_{i,j,c}$—that is, if $(\tilde{\imath}, \tilde{\jmath}', c)$ overestimates the travel time from i to j using connector type c—then we add a recovery arc from $\tilde{\jmath}'$ to the preceding timed copy $\tilde{\jmath}$ of j in $\widehat{L}$, if it does not already exist. A detailed description of this procedure is given in Algorithm 1.

Algorithm 1: CREATE INITIAL NETWORK

Input : Directed network $G = (L, A)$, connector and commodity parameters
Output: Partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$

$\widehat{L} \leftarrow \bigcup_{k \in K} \left(\widetilde{I}^k \cup \widetilde{D}^k\right) \cup \bigcup_{c \in C} \left(\widetilde{S}^c\right) \cup \widetilde{L}_{start} \cup \widetilde{L}_{end}$
$\widehat{H} \leftarrow \{(\tilde{\imath}, \tilde{\imath}') \mid \tilde{\imath} \in \widehat{L} \setminus \widetilde{L}_{end}$ and $\tilde{\imath}' \in \arg\min\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L}$ and $t_{\tilde{\imath}^\dagger} > t_{\tilde{\imath}}\}\}$
$\widehat{A}, \widehat{R} \leftarrow \emptyset$
for $(i, j, c) \in A$ **do**
  for $\tilde{\imath} \in \widehat{L}$ **do**
    if $t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c} \le T$ **then**
      $\tilde{\jmath}' \in \arg\min\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L}$ and $t_{\tilde{\jmath}^\dagger} \ge t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c}\}$, $\widehat{A} \leftarrow \widehat{A} \cup \{(\tilde{\imath}, \tilde{\jmath}', c)\}$
      if $t_{\tilde{\jmath}'} > t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c}$ **then**
        $\tilde{\jmath} \in \arg\max\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L}$ and $t_{\tilde{\jmath}^\dagger} < t_{\tilde{\jmath}'}\}$, $\widehat{R} \leftarrow \widehat{R} \cup \{(\tilde{\jmath}', \tilde{\jmath})\}$
      end
    end
  end
end
return $\widehat{G}$

3.2.2. Illegal Connector Cycle Elimination. Given an SND($\widehat{G}$) solution, we aim to convert it into a feasible FD solution with identical objective value. This requires feasibly routing connectors and commodities in $\widetilde{G}$ while maintaining (i) the same sequence of locations visited by each connector, and (ii) identical commodity consolidations during each connector movement. However, an SND($\widehat{G}$) solution may route connectors and commodities along recovery arcs that point backward in time in the partially time-expanded network $\widehat{G}$. While it may still be possible to recover a feasible FD solution from such a solution, the cyclic structure of $\widehat{G}$ can lead to undesirable connector behavior, preventing the objective value of the resulting FD solution from matching that of the SND($\widehat{G}$) solution.

Specifically, SND($\widehat{G}$) permits every connector type $c \in C$ to be routed along *subtours*, i.e., cycles disconnected from the main connector walks that originate in $\widetilde{S}^c$ and end at $\tilde{\imath}_{end}$. It also allows connectors to follow walks that traverse the same movement arc multiple times, thereby creating cycles. Such undesirable cycles may appear at optimality of SND($\widehat{G}$), as they can artificially inflate the transportation capacity of movement arcs, yielding solutions with higher apparent demand fulfillment than any feasible FD solution could achieve.

In contrast, a connector traveling along a walk in $\widehat{G}$ that revisits the same time-space location multiple times *without* traversing the same movement arc more than once may still be converted into a path in $\widetilde{G}$ that preserves both the sequence of visited locations and the associated commodity flows. Figure 4 illustrates the differences between these three types of cycles.

Therefore, before attempting to convert an SND($\widehat{G}$) solution into a feasible FD solution with identical objective value, it must be decomposable into connector walks that satisfy the following two conditions:

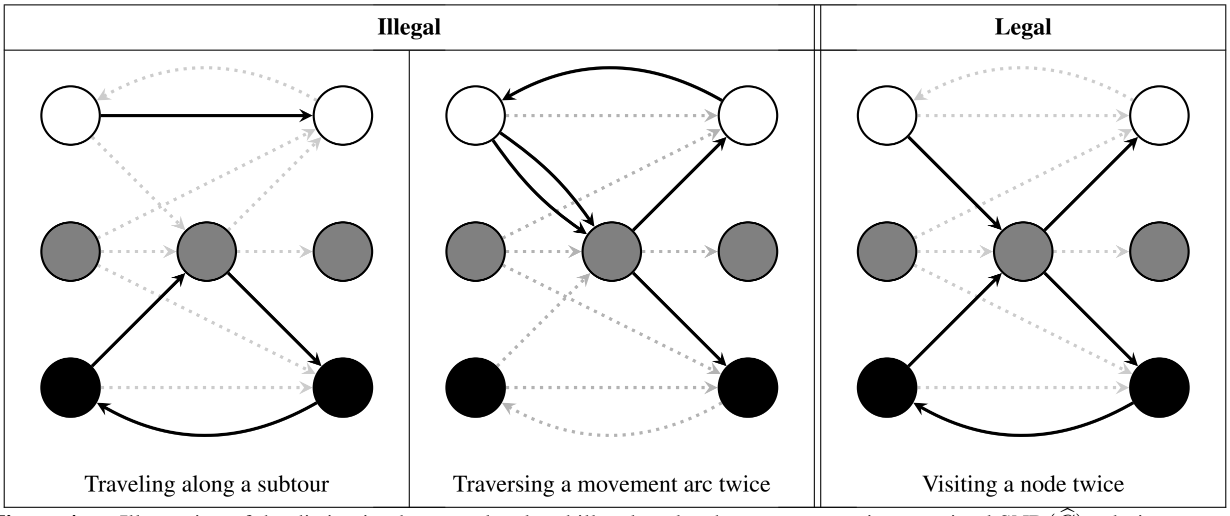


Figure 4 Illustration of the distinction between legal and illegal cycles that may appear in an optimal SND($\widehat{G}$) solution.

CONDITION 1. *Any connector walk in $\widehat{G}$ must originate at a connector source and terminate at $\tilde{\imath}_{end}$.*

CONDITION 2. *Any connector walk in $\widehat{G}$ must not traverse the same movement arc more than once.*

To verify whether these conditions hold for a given SND($\widehat{G}$) solution, we first apply a flow decomposition algorithm (Ahuja et al. 1993) to decompose the flow of each connector type $c \in C$ into a collection of path flows and cycle flows. We then iteratively splice cycles that are reachable from the main connector paths, merging them into walks whenever they do not violate Condition 2. Specifically, if a cycle shares a node with a walk that starts at any connector origin $\tilde{\imath} \in \widetilde{S}^c$ and terminates at $\tilde{\imath}_{end}$, and they do not share any movement arcs, we merge the cycle into the walk to create a new $\tilde{\imath} - \tilde{\imath}_{end}$ connector walk, as illustrated in Figure 5, and update the values of the walk flows and cycle flows. At termination, any remaining cycle flows that could not be spliced into the main connector walks are identified as illegal cycles. A detailed description of this identification process is provided in Algorithm 2.

In Algorithm 2, the while loop (Lines 4 – 8) iteratively splices cycles into walks without violating Condition 2, namely, when they do not share movement arcs. During the splicing process, cycle flows are merged into the spliced walks. Once the while loop terminates, any remaining cycle flows in Ξ^c either violate Condition 2 or cannot be reached from the main connector walks in $\mathcal{W}^c$ and are therefore identified as part of subtours violating Condition 1.

If illegal cycles are identified by Algorithm 2, the next step is to eliminate them by further expanding $\widehat{G}$ and tightening the relaxation problem SND($\widehat{G}$). To this end, note that because the fully time-expanded network $\widetilde{G}$ is acyclic, any cycle in $\widehat{G}$ must contain at least one recovery arc. Removing recovery arcs traversed by illegal cycles thus cuts these undesirable SND($\widehat{G}$) solutions. By Property 4, recovery arcs exist in $\widehat{G}$ when inbound movement arcs overestimate their real travel times. Therefore, we can remove a recovery arc $(\tilde{\jmath}'', \tilde{\jmath})$ by correcting the length of every movement arc $(\tilde{\imath}, \tilde{\jmath}'', c) \in \widehat{A}$ that overestimates its travel

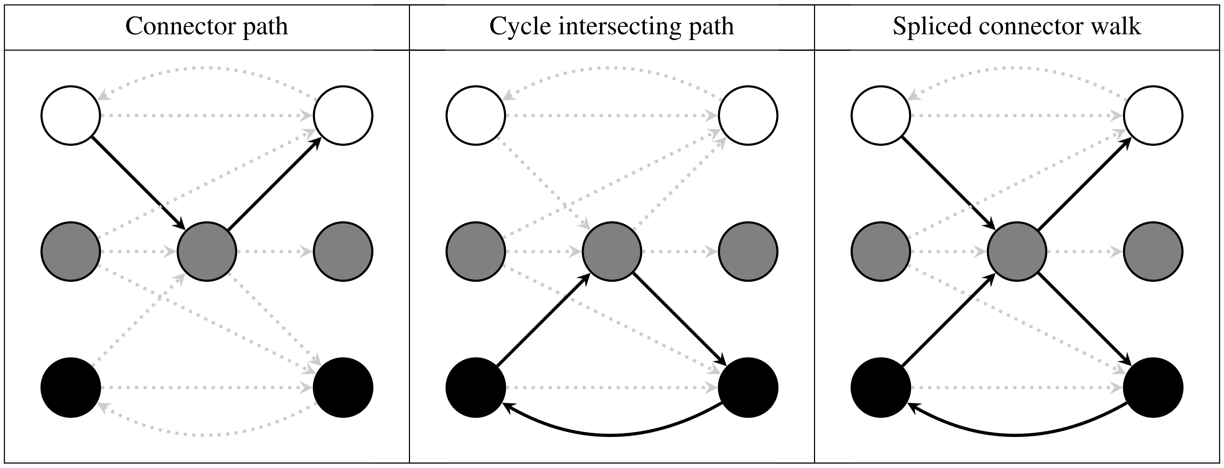


Figure 5 Illustration of the splicing of connector cycles into the main connector walk.

Algorithm 2: IDENTIFY ILLEGAL CYCLES($\widehat{G}, x$)

Input : Partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$, connector flow x
Output: Set of illegal cycles $\mathcal{I}$, walk flow decomposition $(\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c})_{c \in C}$

$\mathcal{I} \leftarrow \emptyset$
for $c \in C$ **do**
  Decompose connector flow x^c into path flows $(\bar{x}^c_W)_{W \in \mathcal{W}^c}$ and cycle flows $(\bar{x}^c_\xi)_{\xi \in \Xi^c}$
  while $\exists (W, \xi) \in \mathcal{W}^c \times \Xi^c$ *such that* W *and* ξ *traverse the same node and* $\xi \cap \widehat{A} \cap W = \emptyset$ **do**
    $W' \leftarrow$ walk obtained by splicing ξ into W, $\alpha \leftarrow \min\{\bar{x}^c_W, \bar{x}^c_\xi\}$
    $\bar{x}^c_{W'} \leftarrow \alpha$, $\bar{x}^c_W \leftarrow \bar{x}^c_W - \alpha$, $\bar{x}^c_\xi \leftarrow \bar{x}^c_\xi - \alpha$
    $\mathcal{W}^c \leftarrow \{\text{walks } W \mid \bar{x}^c_W > 0\}$, $\Xi^c \leftarrow \{\text{cycles } \xi \mid \bar{x}^c_\xi > 0\}$
  end
  $\mathcal{I} \leftarrow \mathcal{I} \cup \Xi^c$
end
return $(\mathcal{I}, (\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c})_{c \in C})$

---

time. This refinement process involves introducing a new time-space location $\tilde{\jmath}'$ satisfying $t_{\tilde{\jmath}'} = t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j} \in (t_{\tilde{\jmath}}, t_{\tilde{\jmath}''})$ (from Property 3) for each such movement arc $(\tilde{\imath}, \tilde{\jmath}'', c)$. The newly discovered time-space locations are then added to $\widehat{L}$, and multiple arcs are adjusted to ensure that the resulting graph $\widehat{G}$ remains a valid partially time-expanded network. We detail the procedure for refining $\widehat{G}$ to remove a recovery arc while preserving Properties 1 – 4 in Algorithm 3.

In Algorithm 3, the first step (Lines 3 – 6) consists of replacing every movement arc ending at $\tilde{\jmath}''$ that has an overestimated travel time with a new movement arc reflecting its exact travel time. This requires adding new timed copies of j in the partially time-expanded network and connecting them with holdover arcs (Line 8). To ensure that Properties 2 – 3 remain satisfied, the algorithm then creates timed copied of each flat arc leaving location j at every newly added timed copy of j, provided they do not exceed the time horizon. Finally, recovery arcs are introduced for any newly constructed movement arc that still overestimates its travel time, thereby ensuring compliance with Property 4.

Algorithm 3: REFINE$(\widehat{G}, (\tilde{\jmath}'', \tilde{\jmath}))$

Input : Partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$, recovery arc $(\tilde{\jmath}'', \tilde{\jmath}) \in \widehat{R}$
Output: New partially time-expanded network $\widehat{G}' = (\widehat{L}', \widehat{A}' \cup \widehat{H}' \cup \widehat{R}')$

$\widehat{L}' \leftarrow \widehat{L}, \quad \widehat{A}' \leftarrow \widehat{A}, \quad \widehat{H}' \leftarrow \widehat{H} \setminus \{(\tilde{\jmath}, \tilde{\jmath}'')\}, \quad \widehat{R}' \leftarrow \widehat{R} \setminus \{(\tilde{\jmath}'', \tilde{\jmath})\}$
$J \leftarrow \{\tilde{\jmath}, \tilde{\jmath}''\}$
for $(\tilde{\imath}, \tilde{\jmath}'', c) \in \widehat{A}$ *such that* $t_{\tilde{\jmath}''} > t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c}$ **do**
  $\tilde{\jmath}' \leftarrow (j, t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c})$
  $\widehat{L}' \leftarrow \widehat{L}' \cup \{\tilde{\jmath}'\}, \quad J \leftarrow J \cup \{\tilde{\jmath}'\}, \quad \widehat{A}' \leftarrow \widehat{A}' \setminus \{(\tilde{\imath}, \tilde{\jmath}'', c)\} \cup \{(\tilde{\imath}, \tilde{\jmath}', c)\}$
end
Sort time-space locations in J in increasing order of their time epochs: $J = \{\tilde{\jmath}'_1, \ldots, \tilde{\jmath}'_{|J|}\}$
$\widehat{H}' \leftarrow \widehat{H}' \cup \{(\tilde{\jmath}'_\ell, \tilde{\jmath}'_{\ell+1}), \ \forall \ell \in \{1, \ldots, |J|-1\}\}$
for $(j, i, c) \in A$ **do**
  for $\tilde{\jmath}' \in J \setminus \{\tilde{\jmath}, \tilde{\jmath}''\}$ **do**
    if $t_{\tilde{\jmath}'} + \tau_{j,i}^{\Delta,c} \leq T$ **then**
      $\tilde{\imath}' \in \arg\min\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L}' \text{ and } t_{\tilde{\imath}^\dagger} \geq t_{\tilde{\jmath}'} + \tau_{j,i}^{\Delta,c}\}, \quad \widehat{A}' \leftarrow \widehat{A}' \cup \{(\tilde{\jmath}', \tilde{\imath}', c)\}$
      if $t_{\tilde{\imath}'} > t_{\tilde{\jmath}'} + \tau_{j,i}^{\Delta,c}$ **then**
        $\tilde{\imath} \in \arg\max\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L}' \text{ and } t_{\tilde{\imath}^\dagger} < t_{\tilde{\imath}'}\}, \quad \widehat{R}' \leftarrow \widehat{R}' \cup \{(\tilde{\imath}', \tilde{\imath})\}$
      end
    end
  end
end
return $\widehat{G}'$

3.2.3. Converting into an FD Solution. When connector flows in an SND$(\widehat{G})$ solution can be decomposed into walks that satisfy Conditions 1 – 2, we can attempt to convert the SND$(\widehat{G})$ solution into a feasible FD solution with identical objective value. From the SND$(\widehat{G})$ solution, we can extract the sequence of locations visited by each connector and commodity, as well as the commodity consolidations during each connector movement. However, some connectors and commodities may traverse recovery arcs that point backward in time. Our objective is to determine whether it is possible to schedule the dispatch of each connector and commodity movement while preserving both the sequence of visited locations and the associated consolidations—*without* traveling backward in time via recovery arcs. If this scheduling proves infeasible, our goal is to refine the partially time-expanded network $\widehat{G}$ by removing recovery arcs to improve the SND$(\widehat{G})$ solution and then attempt the conversion again.

Formally, given an optimal SND$(\widehat{G})$ solution (x, f, θ), let $(\bar{x}^c_W)_{W \in \mathcal{W}^c}$ be the connector walk flows (for every connector type $c \in C$) obtained from Algorithm 2. Using a flow decomposition algorithm, we also decompose the flow f^k of each commodity $k \in K$ into a collection of path flows $(\bar{f}^k_W)_{W \in \mathcal{W}^k}$ originating in $\widetilde{I}^k$ and terminating in $\widetilde{D}^k$. We discard any commodity cycles, as they do not contribute to demand fulfillment. Let $\overline{\mathcal{W}} := \left(\bigcup_{c \in C} \mathcal{W}^c\right) \cup \left(\bigcup_{k \in K} \mathcal{W}^k\right)$ denote the merged set of connector and commodity walks. For each walk $W \in \overline{\mathcal{W}}$, let t^W_{start} (resp. t^W_{end}) denote its start (resp. end) time in $\widehat{G}$, and let a^W_ℓ denote the $\ell-$th arc traversed by W for every $\ell \in \{1, \ldots, |W|\}$.

We now formulate an optimization problem to determine whether feasible dispatch times exist for these connector and commodity walks. Specifically, we define a variable $\gamma_a^W \geq 0$ for every walk $W \in \overline{\mathcal{W}}$ and every arc $a \in W$ to represent the dispatch time when traversing a. However, to preserve commodity consolidations, we relax dispatch times by allowing connectors and commodities to be scheduled to depart from a location *before* they arrive there. For every recovery arc $(\tilde{\imath}', \tilde{\imath}) \in W \cap \widehat{R}$ in a connector or commodity walk, we define a variable $\zeta_{\tilde{\imath}',\tilde{\imath}}^W \geq 0$ (resp. $\sigma_{\tilde{\imath}',\tilde{\imath}}^W \in \{0,1\}$) to represent the amount of time that (resp. whether or not) the connector or commodity departs from location i before it arrives. For readability, we use $\gamma_{a_\ell}^W$ to refer to the dispatch variable associated with the $\ell-$th arc a_ℓ^W in a walk $W \in \overline{\mathcal{W}}$. We also introduce a dummy variable $\gamma_{a_{|W|+1}}^W$ to denote the arrival time at the end of walk W. The connector-commodity dispatch (CCD) problem can then be formulated as the following MIP, denoted CCD($\overline{\mathcal{W}}$), which aims to minimize the number of times backward time travel is needed to ensure consolidation feasibility:

$$\text{CCD}(\overline{\mathcal{W}}): \min_{\gamma,\zeta,\sigma} \quad \sum_{W \in \overline{\mathcal{W}}} \sum_{(\tilde{\imath}',\tilde{\imath}) \in W \cap \widehat{R}} \sigma_{\tilde{\imath}',\tilde{\imath}}^W \tag{3a}$$

$$\text{s.t.} \quad \gamma_{a_{\ell+1}}^W - \gamma_{a_\ell}^W \geq \begin{cases} \tau_{i,j}^{\Delta,c} & \text{if } a_\ell^W = (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A}, \\ -\zeta_{\tilde{\imath}',\tilde{\imath}}^W & \text{if } a_\ell^W = (\tilde{\imath}',\tilde{\imath}) \in \widehat{R}, \\ 0 & \text{if } a_\ell^W \in \widehat{H}, \end{cases} \quad \forall W \in \overline{\mathcal{W}},\ \forall \ell \in \{1,\dots,|W|\}, \tag{3b}$$

$$\zeta_{\tilde{\imath}',\tilde{\imath}}^W \leq (t_{\tilde{\imath}'} - t_{\tilde{\imath}})\sigma_{\tilde{\imath}',\tilde{\imath}}^W, \qquad \forall W \in \overline{\mathcal{W}},\ \forall (\tilde{\imath}',\tilde{\imath}) \in W \cap \widehat{R}., \tag{3c}$$

$$\gamma_{a_1}^W \geq t_{start}^W, \qquad \forall W \in \overline{\mathcal{W}}, \tag{3d}$$

$$\gamma_{a_{|W|+1}}^W \leq t_{end}^W, \qquad \forall W \in \overline{\mathcal{W}}, \tag{3e}$$

$$\gamma_a^{W_1} = \gamma_a^{W_2}, \qquad \forall a \in \widehat{A},\ \forall (W_1,W_2) \in \overline{\mathcal{W}}^2 \mid a \in W_1 \cap W_2, \tag{3f}$$

$$\gamma_a^W \geq 0, \qquad \forall W \in \overline{\mathcal{W}},\ \forall a \in W, \tag{3g}$$

$$\zeta_{\tilde{\imath}',\tilde{\imath}}^W \geq 0, \qquad \forall W \in \overline{\mathcal{W}},\ \forall (\tilde{\imath}',\tilde{\imath}) \in W \cap \widehat{R}., \tag{3h}$$

$$\sigma_{\tilde{\imath}',\tilde{\imath}}^W \in \{0,1\}, \qquad \forall W \in \overline{\mathcal{W}},\ \forall (\tilde{\imath}',\tilde{\imath}) \in W \cap \widehat{R}. \tag{3i}$$

The objective function (3a) of CCD($\overline{\mathcal{W}}$) minimizes the number of times that connectors and commodities must depart from a location before arriving at that location. Constraints (3b) link the dispatch times along each walk. They account for travel times and allow connectors and commodities to depart from a location associated with a recovery arc $(\tilde{\imath}', \tilde{\imath})$ before arriving there, while incurring a penalty $\sigma_{\tilde{\imath}',\tilde{\imath}}^W$ enforced by Constraints (3c). Constraints (3d)–(3e) ensure that connectors and commodities begin their travels only after becoming available and reach their destinations before their respective deadlines. Constraints (3f) enforce all commodity consolidations by requiring any two commodities or connectors that traverse the same movement arc in the optimal SND($\widehat{G}$) solution be dispatched at the same time. Finally, Constraints (3g)–(3i) define the decision variables.

The CCD($\overline{\mathcal{W}}$) problem characterizes the level of infeasibility of the current SND($\widehat{G}$) solution. If its optimal value is zero, the dispatch times along each walk are feasible and can be directly used to convert the SND($\widehat{G}$) solution into an FD solution with the same objective value. This is done by converting each connector-type walk flow $(\bar{x}^c_W)_{W\in\mathcal{W}^c}$ for every $c \in C$ and each commodity path flow $(\bar{f}^k_W)_{W\in\mathcal{W}^k}$ for every $k \in K$ into path flows in $\widetilde{G}$ using the dispatch times, as described in Algorithm 4.

Algorithm 4: CONSTRUCT FLOW($\mathcal{W}, (\bar{g}_W)_{W\in\mathcal{W}}, (\gamma^W)_{W\in\mathcal{W}}$)

Input : Set of walks $\mathcal{W}$ in $\widehat{G}$, walk flows $(\bar{g}_W)_{W\in\mathcal{W}}$, dispatch time vectors $(\gamma^W)_{W\in\mathcal{W}}$
Output: Flow vector $(g^c_{\tilde{\imath},\tilde{\jmath}})_{(\tilde{\imath},\tilde{\jmath},c)\in\widetilde{A}}$ in $\widetilde{G}$

for $W \in \mathcal{W}$ **do**
  $\tilde{\ell}_{start} \leftarrow$ time-space origin of W in $\widehat{G}$, $\tilde{\ell}_{end} \leftarrow$ time-space destination of W in $\widehat{G}$
  $P_W \leftarrow \tilde{\ell}_{start} - \tilde{\ell}_{end}$ path in $\widetilde{G}$ such that $P_W \cap \widetilde{A} = \bigcup_{a=(\tilde{\imath},\tilde{\jmath},c)\in W} \{(\tilde{\imath}',\tilde{\jmath}',c)\in\widetilde{A} \mid t_{\tilde{\imath}'} = \Delta\cdot\lfloor\gamma^W_a/\Delta\rfloor\}$
end
for $(\tilde{\imath},\tilde{\jmath},c) \in \widetilde{A}\cup\widetilde{H}$ **do**
  $g^c_{\tilde{\imath},\tilde{\jmath}} \leftarrow \sum_{W\in\mathcal{W}} \bar{g}_W \cdot \mathbb{1}_{\{(\tilde{\imath},\tilde{\jmath},c)\in P_W)\}}$
end
return $(g^c_{\tilde{\imath},\tilde{\jmath}})_{(\tilde{\imath},\tilde{\jmath},c)\in\widetilde{A}\cup\widetilde{H}}$

Conversely, if the optimal value of CCD($\overline{\mathcal{W}}$) is strictly positive, then the recovery arcs $(\tilde{\imath}',\tilde{\imath}) \in \widehat{R}$ for which $\sigma^W_{\tilde{\imath}',\tilde{\imath}} = 1$ at optimality (for some walk $W \in \overline{\mathcal{W}}$) indicate where the partially time-expanded network $\widehat{G}$ must be refined to improve the SND($\widehat{G}$) solution. In such cases, we execute Algorithm 3 to remove these recovery arcs.

3.2.4. Overall DDD Algorithm. With these subroutines in place, we can now present our DDD algorithm for solving ELDNP—henceforth denoted SOLVE-ELNDP. The algorithm first constructs an initial partially time-expanded network $\widehat{G}$ using Algorithm 1. Then, at each iteration, it solves SND($\widehat{G}$) using a MIP solver. Algorithm 2 is used to detect any illegal cycles in the SND($\widehat{G}$) solution. If such cycles are found, SOLVE-ELNDP refines $\widehat{G}$ by discovering new time-space locations (Algorithm 3) to remove the associated recovery arcs and eliminate the illegal cycles, before resolving SND($\widehat{G}$). When the SND($\widehat{G}$) solution no longer contains illegal cycles, SOLVE-ELNDP decomposes it into a set $\overline{\mathcal{W}}$ of connector and commodity walks, and solves the dispatch problem CCD($\overline{\mathcal{W}}$). If the optimal value of CCD($\overline{\mathcal{W}}$) is strictly positive, SOLVE-ELNDP again refines $\widehat{G}$ by removing the recovery arcs used for backward time travel in the CCD($\overline{\mathcal{W}}$) solution, and then resolves SND($\widehat{G}$). When the optimal value of CCD($\overline{\mathcal{W}}$) is zero, SOLVE-ELNDP terminates with an optimal FD solution constructed from the SND($\widehat{G}$) and CCD($\overline{\mathcal{W}}$) solutions (Algorithm 4). The algorithm is guaranteed to terminate, as the fully time-expanded network $\widetilde{G}$ is finite. The complete procedure is detailed in Algorithm 5.

Algorithm 5: SOLVE-ELNDP

Input : Expeditionary network $G = (L, A)$, connector and commodity parameters
Output: Optimal solution (x^*, f^*, θ^*) of FD

$\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R}) \leftarrow$ CREATE INITIAL NETWORK (Algorithm 1)
do
  Solve SND($\widehat{G}$): $(x, f, \theta) \leftarrow$ optimal solution
  $(\mathcal{I}, (\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c})_{c \in C}) \leftarrow$IDENTIFY ILLEGAL CYCLES($\widehat{G}, x$) (Algorithm 2)
  if $\mathcal{I} \neq \emptyset$ **then**
    for $\xi \in I$ **do**
      Select $(\tilde{\imath}', \tilde{\imath}) \in \xi \cap \widehat{R}$
      $\widehat{G} \leftarrow$ REFINE($\widehat{G}, (\tilde{\imath}', \tilde{\imath})$) (Algorithm 3)
    end
  else
    Decompose every commodity flow f^k (for $k \in K$) into path flows $(\bar{f}^k_W)_{W \in \mathcal{W}^k}$
    $\overline{\mathcal{W}} \leftarrow \left(\bigcup_{c \in C} \mathcal{W}^c\right) \cup \left(\bigcup_{k \in K} \mathcal{W}^k\right)$
    Solve CCD($\overline{\mathcal{W}}$): $(\gamma, \zeta, \sigma) \leftarrow$ optimal solution, $z \leftarrow$ optimal value
    if $z > 0$ **then**
      for $(\tilde{\imath}', \tilde{\imath}) \in \widehat{R}$ *such that* $\sum_{\{W \in \overline{\mathcal{W}} \,|\, (\tilde{\imath}', \tilde{\imath}) \in W\}} \sigma^W_{\tilde{\imath}', \tilde{\imath}} > 0$ **do**
        $\widehat{G} \leftarrow$ REFINE($\widehat{G}, (\tilde{\imath}', \tilde{\imath})$) (Algorithm 3)
      end
    else
      for $c \in C$ **do**
        $(x^{c^*}_{\tilde{\imath}, \tilde{\jmath}})_{(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A} \cup \widetilde{H}} \leftarrow$ CONSTRUCT FLOW($\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c}, (\gamma^W)_{W \in \mathcal{W}^c}$) (Algorithm 4)
      end
      for $k \in K$ **do**
        $(f^{c,k^*}_{\tilde{\imath}, \tilde{\jmath}})_{(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A} \cup \widetilde{H}} \leftarrow$ CONSTRUCT FLOW($\mathcal{W}^k, (\bar{f}^k_W)_{W \in \mathcal{W}^k}, (\gamma^W)_{W \in \mathcal{W}^k}$) (Algorithm 4)
      end
      $\theta^* \leftarrow \theta$
    end
  end
while $\mathcal{I} \neq \emptyset$ *or* $z > 0$
return (x^*, f^*, θ^*)

3.3. Algorithmic Enhancements for Expeditionary Logistics

While SOLVE-ELNDP is guaranteed to eventually terminate with an optimal solution, we can leverage structural properties of expeditionary logistics networks to accelerate its convergence.

3.3.1. Capacity Factor. Expeditionary logistics operations often take place in austere environments far from established infrastructure. As a result, supply bases are typically remote relative to demand locations, and connectors frequently perform out-and-back excursions early in the time horizon to move supplies from remote bases to more central staging points for subsequent distribution. Because supply bases rarely generate demand themselves, the return journeys of these out-and-back trips often carry no commodities. This pattern occurs often enough at optimality of ELDNP to justify incorporating it into our network formulation.

When constructing a partially time-expanded network, some arcs in $\widehat{A}$ inevitably overestimate travel times. We can exploit this "excess" time by embedding the concept of out-and-back cycles into the capacity of these arcs. Specifically, for each arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$, we define a *capacity factor*, given by $\lfloor (t_{\tilde{\jmath}} - t_{\tilde{\imath}} - \tau^{\Delta,c}_{i,j})/(\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}) \rfloor$, which represents the number of additional out-and-back trips a connector of type c could complete between i and j within the allotted time $t_{\tilde{\jmath}} - t_{\tilde{\imath}}$, beyond the initial trip from i to j. We then modify Constraints (2d) – (2e) in SND($\widehat{G}$) as follows:

$$\sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \le \left(1 + \left\lfloor \frac{t_{\tilde{\jmath}} - t_{\tilde{\imath}} - \tau^{\Delta,c}_{i,j}}{\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}} \right\rfloor \right) \cdot u^c \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \quad \forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \tag{2d*}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \le \left(1 + \left\lfloor \frac{t_{\tilde{\jmath}} - t_{\tilde{\imath}} - \tau^{\Delta,c}_{i,j}}{\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}} \right\rfloor \right) \cdot q^{c,k} \cdot x^c_{\tilde{\imath},\tilde{\jmath}}, \quad \forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \ \forall k \in K. \tag{2e*}$$

Given an optimal SND($\widehat{G}$) solution (x, f, θ)—computed with the modified Constraints (2d*) – (2e*) and containing no illegal cycles—the dispatch problem CCD($\overline{\mathcal{W}}$) must be adjusted to reflect the realized travel time along each arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$, defined as

$$\overline{\tau}^{\Delta,c}_{i,j} \coloneqq \tau^{\Delta,c}_{i,j} + (\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}) \cdot \left(\max \left\{ \left\lceil \frac{\sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath},\tilde{\jmath}}}{u^c \cdot x^c_{\tilde{\imath},\tilde{\jmath}}} \right\rceil, \max_{k \in K} \left\{ \left\lceil \frac{f^{c,k}_{\tilde{\imath},\tilde{\jmath}}}{q^{c,k} \cdot x^c_{\tilde{\imath},\tilde{\jmath}}} \right\rceil \right\} \right\} - 1 \right).$$

This expression captures the number of out-and-back trips needed to transport the commodities scheduled to traverse $(\tilde{\imath}, \tilde{\jmath}, c)$, given the connector capacity constraints.

We then modify Constraints (3b) in CCD($\overline{\mathcal{W}}$) as follows:

$$\gamma^W_{a_{\ell+1}} - \gamma^W_{a_\ell} \ge \begin{cases} \overline{\tau}^{\Delta,c}_{i,j} & \text{if } a^W_\ell = (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \\ -\zeta^W_{a_\ell} & \text{if } a^W_\ell \in \widehat{R}, \\ 0 & \text{if } a^W_\ell \in \widehat{H}, \end{cases} \qquad \forall W \in \overline{\mathcal{W}}, \ \forall \ell \in \{1, \dots, |W|\}. \tag{3b*}$$

3.3.2. Maximum-Flow Heuristic. To accelerate the convergence of SOLVE-ELNDP, we design a two-step heuristic for computing a feasible FD solution at each iteration, thereby providing a lower bound on the optimal value of FD. The heuristic first constructs a feasible connector flow in $\widetilde{G}$ from an SND($\widehat{G}$) solution, and then computes a commodity flow to maximize demand fulfillment given the connector movements.

Given an SND($\widehat{G}$) solution (x, f, θ)—possibly containing illegal cycles—we decompose the connector flow into walk flows $(\bar{x}^c_W)_{W \in \mathcal{W}^c}$ for each $c \in C$ using Algorithm 2. Next, we solve a small linear program (LP) to determine feasible connector dispatch times *without* enforcing commodity consolidations. Specifically, using notation similar to that introduced in Section 3.2.3, we define dispatch-time variables $\gamma^W_a \ge 0$ for every arc $a \in W$ and formulate the following connector dispatch (CD) LP, denoted CD($\mathcal{W}^c$), for each connector type $c \in C$:

$$\text{CD}(\mathcal{W}^c): \ \min_{\gamma} \quad \sum_{W \in \mathcal{W}^c} \sum_{a = (\tilde{\imath}, \tilde{\jmath}, c) \in W \cap \widehat{A}} (\gamma^W_a - t_{\tilde{\imath}}) \tag{4a}$$

$$\text{s.t.} \quad \gamma^W_{a_{\ell+1}} - \gamma^W_{a_\ell} \ge \begin{cases} \overline{\tau}^{\Delta,c}_{i,j} & \text{if } a^W_\ell = (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \\ 0 & \text{if } a^W_\ell \in \widehat{H} \cup \widehat{R}, \end{cases} \quad \forall W \in \mathcal{W}^c, \ \forall \ell \in \{1, \dots, |W| - 1\}, \tag{4b}$$

$$\gamma^W_a \ge t_{\tilde{\imath}}, \qquad \forall W \in \mathcal{W}^c, \ \forall a = (\tilde{\imath}, \tilde{\jmath}, c) \in W \cap \widehat{A}. \tag{4c}$$

This LP aims to construct feasible connector paths in $\widetilde{G}$ that preserve the sequence of locations visited in the SND($\widehat{G}$) solution, without using any recovery arcs. Constraints (4b) – (4c) ensure dispatch feasibility by requiring dispatch times to be no earlier than the scheduled times in the SND($\widehat{G}$) solution and no earlier than the connectors' realized arrival times. The objective function (4a) minimizes the total delay between dispatch times and scheduled times in the SND($\widehat{G}$) solution.

Note that at optimality of CD($\mathcal{W}^c$), some dispatch times may exceed the time horizon, in which case the corresponding arcs are discarded when constructing connector paths in $\widetilde{G}$. Using the walk flows computed from Algorithm 2 and the dispatch times obtained from CD($\mathcal{W}^c$) (for every $c \in C$), we construct a feasible connector flow $\widehat{x}$ in $\widetilde{G}$ using Algorithm 4.

Next, we determine the best-performing commodity flow that maximizes demand fulfillment given the connector flow $\widehat{x}$. We formulate this as a multi-commodity maximum flow problem on the network induced by $\widehat{x}$. Specifically, we define the graph $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H})$ as follows:

$$\widehat{L} = \bigcup_{k \in K} \left(\widetilde{I}^k \cup \widetilde{D}^k\right) \cup \bigcup_{c \in C} \{\tilde{\imath}, \tilde{\jmath} \in \widetilde{L} \mid \widehat{x}^c_{\tilde{\imath},\tilde{\jmath}} > 0\}, \qquad \widehat{A} = \{(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A} \mid \widehat{x}^c_{\tilde{\imath},\tilde{\jmath}} > 0\},$$

and we let $\widehat{H}$ contain the holdover arcs connecting consecutive timed copies of nodes in $\widehat{L}$. For each commodity $k \in K$, we introduce commodity flow variables $f^{c,k}_{\tilde{\imath},\tilde{\jmath}}$ for each arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H}$ and fulfillment variables $\theta^k_{\tilde{\imath}}$ for each demand node $\tilde{\imath} \in \widetilde{D}^k$. We then formulate the following multi-commodity maximum-flow problem, denoted MCMF($\widehat{x}$):

$$\text{MCMF}(\widehat{x}): \max_{f,\theta} \sum_{k \in K} \sum_{\tilde{\imath} \in \widetilde{D}^k} v^k_{\tilde{\imath}} \cdot \theta^k_{\tilde{\imath}} \tag{5a}$$

$$\text{s.t.} \quad \sum_{c \in C} \Big(\sum_{(\tilde{\imath},\tilde{\jmath},c) \in \delta^+(\tilde{\imath},c)} f^{c,k}_{\tilde{\imath},\tilde{\jmath}} - \sum_{(\tilde{\jmath},\tilde{\imath},c) \in \delta^-(\tilde{\imath},c)} f^{c,k}_{\tilde{\jmath},\tilde{\imath}} \Big) \begin{cases} \le b^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{I}^k, \\ = -\theta^k_{\tilde{\imath}} & \text{if } \tilde{\imath} \in \widetilde{D}^k, \\ = 0 & \text{otherwise}, \end{cases} \quad \forall \tilde{\imath} \in \widehat{L}, \ \forall k \in K, \tag{5b}$$

$$\sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \le \left(1 + \left\lfloor \frac{t_{\tilde{\jmath}} - t_{\tilde{\imath}} - \tau^{\Delta,c}_{i,j}}{\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}} \right\rfloor \right) \cdot u^c \cdot \widehat{x}^c_{\tilde{\imath},\tilde{\jmath}}, \qquad \forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \tag{5c}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \le \left(1 + \left\lfloor \frac{t_{\tilde{\jmath}} - t_{\tilde{\imath}} - \tau^{\Delta,c}_{i,j}}{\tau^{\Delta,c}_{i,j} + \tau^{\Delta,c}_{j,i}} \right\rfloor \right) \cdot q^{c,k} \cdot \widehat{x}^c_{\tilde{\imath},\tilde{\jmath}}, \qquad \forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}, \ \forall k \in K, \tag{5d}$$

$$0 \le \theta^k_{\tilde{\imath}} \le d^k_{\tilde{\imath}}, \qquad \forall \tilde{\imath} \in \widetilde{D}^k, \ \forall k \in K, \tag{5e}$$

$$f^{c,k}_{\tilde{\imath},\tilde{\jmath}} \ge 0, \qquad \forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H}, \ \forall k \in K. \tag{5f}$$

The commodity flow $\widehat{f}$ and demand fulfillment $\widehat{\theta}$ obtained at optimality of MCMF($\widehat{x}$), combined with the feasible connector flow $\widehat{x}$, yield a feasible solution to FD. Although its objective value $z_{\text{MCMF}(\widehat{x})}$ is not guaranteed to match that of the SND($\widehat{G}$) solution $z_{\text{SND}(\widehat{G})}$, it can be used to compute an optimality gap that serves as a relaxed stopping criterion within SOLVE-ELNDP. Specifically, since the optimal value of SND($\widehat{G}$) upper bounds the optimal value of FD, an optimality gap associated with the feasible FD solution $(\widehat{x}, \widehat{f}, \widehat{\theta})$ is given by $(z_{\text{SND}(\widehat{G})} - z_{\text{MCMF}(\widehat{x})}) / z_{\text{SND}(\widehat{G})}$.

3.3.3. Enhanced DDD Algorithm. Figure 6 presents a flowchart of SOLVE-ELNDP (Algorithm 5) with all algorithmic enhancements incorporated.

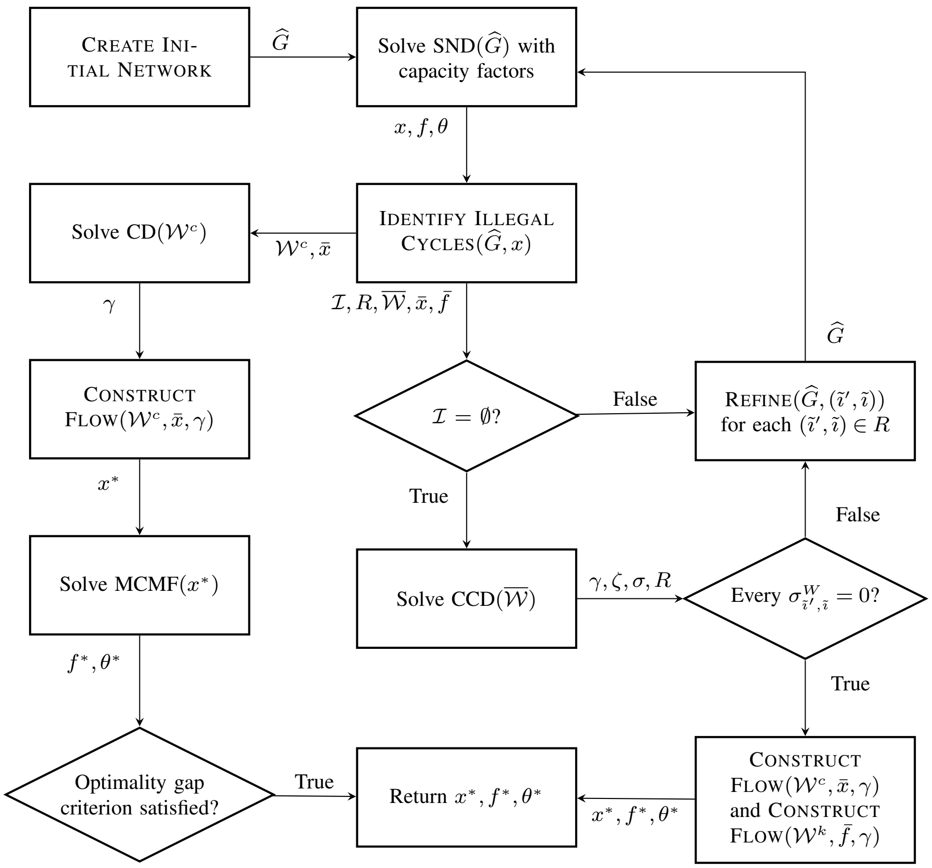


Figure 6 Detailed flowchart of the proposed DDD algorithm for solving ELNDP, including acceleration techniques.

3.4. Comparison to SOLVE-CTSNDP (Boland et al. 2017)

We conclude this section by contrasting our model with the original DDD algorithm SOLVE-CTSNDP introduced by Boland et al. (2017). Their work studies the Continuous-Time Service Network Design Problem (CTSNDP), where a service provider seeks to design a minimum-cost consolidation plan for commodities (characterized by fixed origin-destination pairs) while simultaneously deciding how many connectors to contract for each arc in the service network. In their setting, connectors are not managed across the network; instead, it is assumed, as is common in the service network design literature, that sufficient supply of connectors is available for hire on each arc. To solve this problem, the authors propose SOLVE-CTSNDP, which constructs a partially time-expanded network with optimistic travel times. In this network, arcs are "too short" in that they *underestimate* their travel times. The algorithm solves a service network design problem on this smaller optimistic network to obtain a relaxed solution and then attempts to convert it into a feasible CTSNDP solution with the same objective value. If conversion fails, the algorithm records a lower bound and iterates.

Our work differs from this original study in both problem setting and solution methodology. First, our ELNDP model explicitly coordinates a fleet of heterogeneous connectors across network arcs and allows flexible sourcing of commodities—both critical modeling features in expeditionary logistics environments. Second, from a methodological standpoint, we designed our DDD algorithm using a conservative formulation that *overestimates* arc travel times and introduces recovery arcs. This brings two key benefits. It allows the dispatch problem CCD($\overline{\mathcal{W}}$) in SOLVE-ELNDP to be formulated with significantly fewer binary variables—one for each connector-independent recovery arc—whereas the dispatch problem in SOLVE-CTSNDP requires binary variables for every connector-dependent movement arc, resulting in faster solve times. It also implicitly models out-and-back trips through capacity factors, which substantially improve solution quality in the early iterations of the algorithm. Finally, because our model explicitly manages connectors, SOLVE-ELNDP must address connector cycles—some of which must be preserved while others must be eliminated—a challenge that does not arise in SOLVE-CTSNDP.

To demonstrate the value of the proposed conservative formulation with recovery arcs, we adapt SOLVE-CTSNDP to solve FD as a point of comparison. We design SOLVE-CTSNDP*, which iteratively solves relaxations of FD on optimistic partially time-expanded networks with underestimated travel times, mirroring the approach of Boland et al. (2017). If a relaxed solution contains illegal connector cycles, the optimistic partially time-expanded network is refined by *lengthening* arcs with underestimated travel times. Once the relaxed solution contains no illegal cycles, we formulate a dispatch problem as a MIP to convert the relaxed solution into a feasible FD solution with the same objective value. If this conversion fails, the dispatch MIP identifies which arcs must be lengthened to refine the optimistic partially time-expanded network and cut off the current relaxed solution. Otherwise, the algorithm terminates with an optimal FD solution. At each iteration, we also use the heuristic developed in Section 3.3.2 to compute a feasible FD solution and its associated optimality gap.

A formal definition of an optimistic partially time-expanded network and a detailed description of SOLVE-CTSNDP* are provided in Section EC.2. In the next section, we show that SOLVE-ELNDP consistently outperforms SOLVE-CTSNDP*, thereby justifying the use of a conservative formulation.

4. Case Study: Expeditionary Logistics Operations

In this section, we apply our methodology to a collection of instances representing expeditionary logistics scenarios that have been validated by subject matter experts at the United States Marine Corps to closely reflect real-world conditions. The purpose of this case study is both to demonstrate the efficacy of our proposed algorithm and to examine the impact of finer decision-time discretization on expeditionary logistics objectives. As benchmarks, we compare the performance of SOLVE-ELNDP against two alternatives:

(i) directly solving the fully time-expanded formulation FD, and (ii) using SOLVE-CTSNDP*, i.e., the adapted version of SOLVE-CTSNDP described in Section EC.2, under various rounding factors. We provide an overview of the scenarios and instances in Section 4.1, present comparative performance results in Section 4.2, and derive managerial insights in Section 4.3.

4.1. Expeditionary Scenarios

We consider 53 instances across three expeditionary scenarios developed by subject matter experts and validated by the United States Marine Corps. The first is set in the Pacific Ocean and is referred to as the "Okinawa" scenario. The second is set off the coast of southern California and is referred to as the "SoCal" scenario. The third is set in the Gulf of Mexico and is referred to as the "Bahamas" scenario.

The Okinawa scenario has the fewest locations and arcs but spans the longest time horizon, with demand requests distributed over time. The SoCal scenario includes more locations and arcs, but the time horizon is much shorter and demand requests are more urgent. The Bahamas scenario is the most challenging one, with the largest network in terms of nodes and arcs, and demand requests that are both the most urgent and the largest in quantity, placing significant strain on vehicle capacity resources.

Each scenario is specified by (i) a list of supply bases and their commodity inventories, (ii) a list of potential Expeditionary Advanced Bases (EABs) that can serve as both transshipment and demand locations, (iii) a list of potentially available connectors with their initial locations, speeds and capacities, and (iv) a list of potential demand requests from EABs. To construct notional test instances, we sample from these inputs: first selecting a subset of EABs to be open during the time horizon, then sampling a subset of potential supply requests from those locations, and finally sampling a subset of available connectors to be initially stationed at those locations. In these scenarios, stakeholders aim to maximize total demand fulfillment, so we assign identical criticality parameters across all commodities and demand requests. We summarize the key characteristics of the resulting 53 instances in Table 1, illustrate the theaters and network node locations in Figure 7, and list connector and commodity attributes in Section EC.3.1.

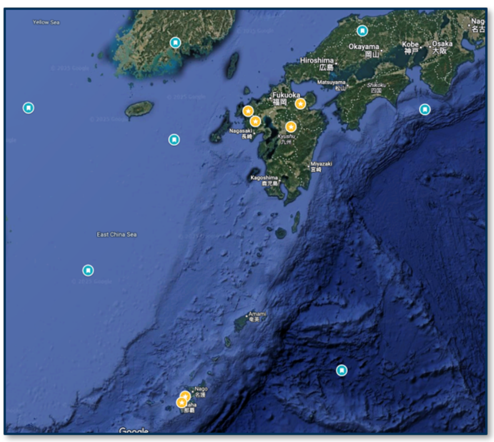
4.2. Numerical Experiments

We solved each instance using SOLVE-ELNDP with all acceleration techniques, SOLVE-CTSNDP*, and the FD method (i.e., directly solving FD) under rounding factors of 6 hours, 3 hours, 1 hour, and 1 minute. All approaches were implemented in Python v3.10, with optimization problems solved using Gurobi v10.0.3 on a computer cluster, each job limited to one core and 8 GB of memory. Each run was terminated either upon reaching a 1% optimality gap or after one hour. Because stakeholders emphasized that solution speed is of paramount importance, we adopted this relatively strict time limit. Complete numerical results are reported in Section EC.3.2 and analyzed below.

Table 1 Summary of instance characteristics

Region	Okinawa	SoCal	Bahamas
# Instances	24	13	16
Time horizon	31 days	5 days	3 days
# Locations	[3, 6]	7	[12, 25]
# Connectors	[12, 32]	[27, 58]	[13, 95]
# Connector types	[8, 9]	[7, 9]	[7, 11]
# Connector type-dependent flat arcs	[24, 150]	[54, 72]	[336, 2080]
# Commodity types	[23, 53]	[27, 57]	3
# Demand requests	[29, 60]	[55, 70]	[33, 60]
# Total demand units	[27k, 191k]	[37k, 125k]	[5.7M, 17M]

Okinawa



SoCal

Bahamas

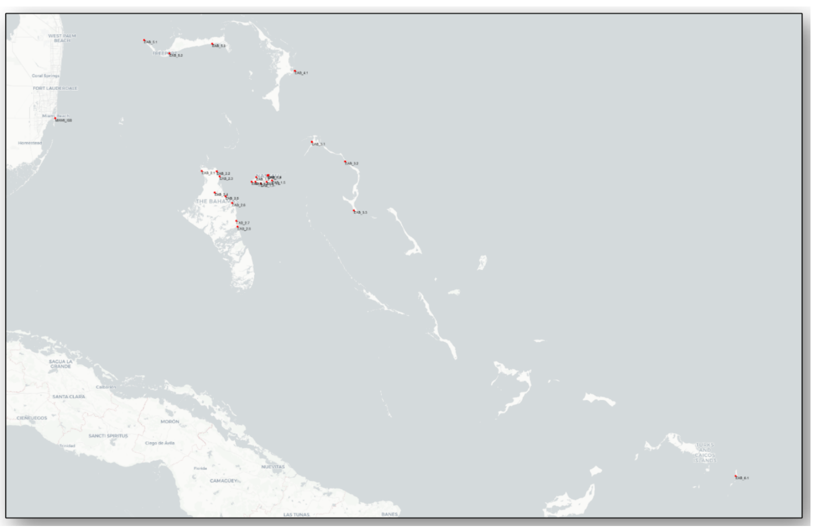


Figure 7 Images of the theaters and EAB locations for each expeditionary scenario.

Figure 8 presents for each approach the average demand fulfilled, CPU time, and optimality gap at termination across instances, stratified by scenario and rounding factor. Optimality gaps are computed using the best bounds obtained across all three approaches. Overall, SOLVE-ELNDP outperforms the benchmark methods in terms of solution quality. Considering the best-performing rounding factor for each method, SOLVE-ELNDP achieves a 104.47% higher fulfillment rate than FD in Okinawa, 2.67% higher in SoCal, and 65.96% higher in the Bahamas. Moreover, across all instances and rounding factors, SOLVE-ELNDP delivers an average improvement in demand fulfillment of 106.40% relative to SOLVE-CTSNDP*, accompanied by a reduction in the average optimality gap from 32.97% to 16.4%. We note that the Bahamas scenario is the most challenging setting, as less than 10% of the requested demand can be satisfied at optimality, placing considerable strain on all solution methods. Nevertheless, SOLVE-ELNDP with a 1-minute rounding factor increases average demand fulfillment by 65.83% compared to FD with a 6-hour rounding factor. This result highlights the value of finer time discretization in expeditionary logistics planning and demonstrates the effectiveness of DDD algorithms in improving solution quality under highly constrained conditions.

Although FD can obtain near-optimal solutions in the SoCal and Bahamas scenarios for the coarsest rounding factor faster than the DDD algorithms, its performance deteriorates sharply for finer resolutions, running out of memory on all 53 instances when the rounding factor is reduced to one minute. Overall, SOLVE-ELNDP proves more favorable in terms of runtime efficiency as well, achieving average CPU times 28.69% faster than those of SOLVE-CTSNDP*.

These results underscore the advantages of the conservative formulation underlying SOLVE-ELNDP, which enables scalable, high-quality expeditionary logistics plans where baseline methods struggle. They also highlight the benefits of using finer rounding factors, which improve asset coordination and lead to higher demand fulfillment. Notably, the best fulfillment rates are achieved at different levels of temporal resolution across scenarios—1 hour for Okinawa, 3 hours for SoCal, and 1 minute for the Bahamas—illustrating the importance of balancing time discretization with solution quality.

A key advantage of DDD algorithms is that they solve MIPs of reduced size. Figure 9 reports the average number of variables in the final MIP solved by each approach. For FD, this corresponds to the full formulation, while for the DDD algorithms it reflects the last evaluated SND($\widehat{G}$) model. Both DDD variants yield substantially smaller models than FD, with SOLVE-ELNDP using 94.36% fewer variables than FD and 74.76% fewer than SOLVE-CTSNDP*.

Because of explicit asset management, a time-consuming step for both DDD algorithms is the elimination of illegal connector cycles. To conclude our benchmark comparison, Figure 9 also shows the average number of iterations each DDD algorithm spends removing these cycles. We find that SOLVE-ELNDP requires fewer cycle-elimination iterations than SOLVE-CTSNDP* in the Okinawa and SoCal scenarios, but significantly more in the Bahamas—despite still terminating earlier and producing higher-quality solutions.

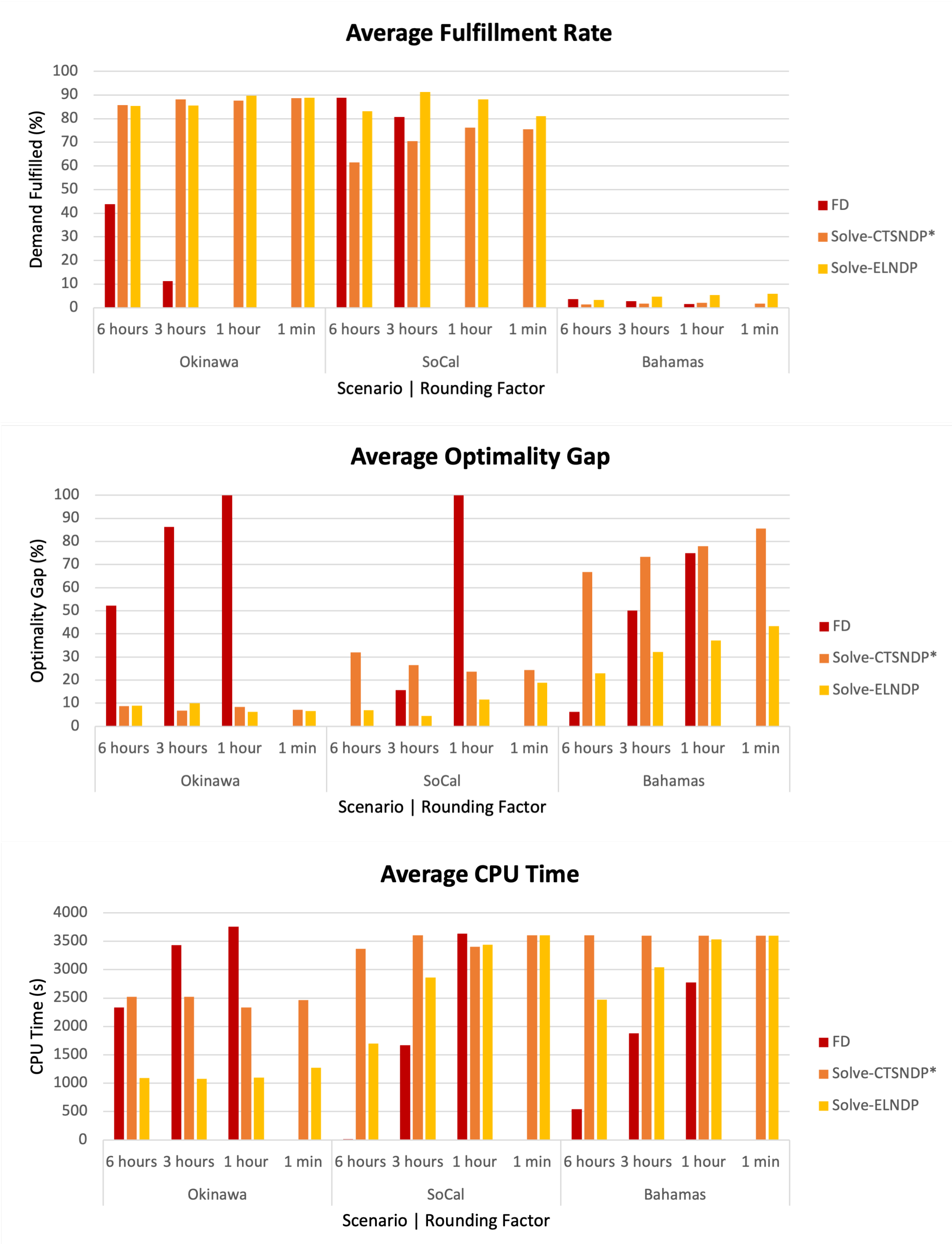


Figure 8 Average demand fulfillment, optimality gap, and CPU time at termination by scenario, approach, and rounding factor. Optimality gaps are computed using the best bound across approaches.

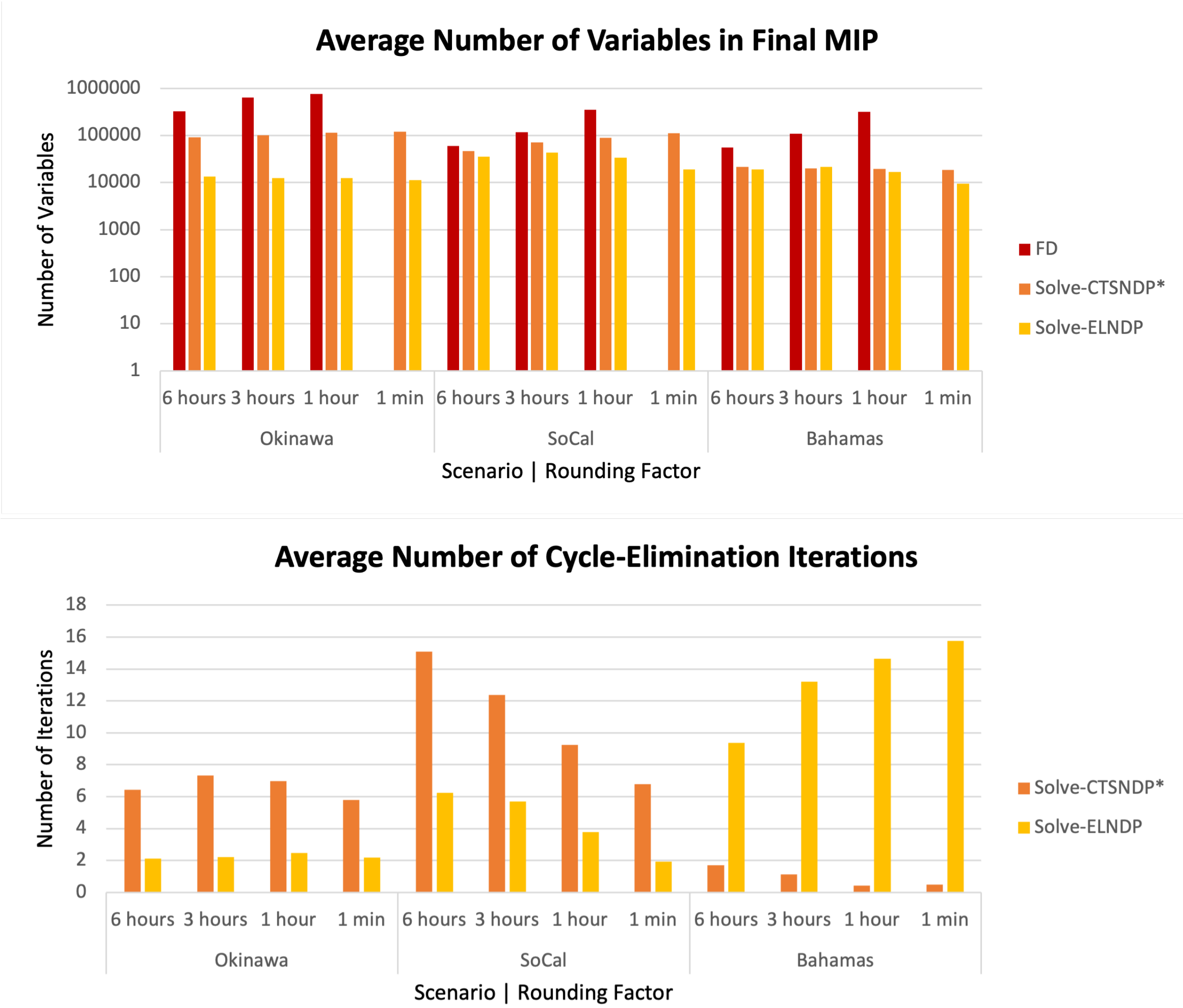


Figure 9 Average number of variables (log scale) and cycle-elimination iterations at termination by scenario, approach, and rounding factor.

4.3. Managerial Insights

We analyze the solutions to each instance of the Bahamas scenario returned by SOLVE-ELNDP under the 1-minute rounding factor to gain insights into which connectors are most valuable, where they should be positioned, and how these operational decisions affect demand fulfillment. Figure 10 illustrates the relationship between connector speed and the average distance traveled, while Figure 11 shows the distribution of connector domains (air, sea, and ground) employed for commodities' first and last movement legs. Figure 12 presents the distribution of the average number of movement arcs per commodity, indicating the typical path complexity in optimal plans. Similar analyses for the other scenarios are provided in Section EC.3.3.

We first observe in Figure 10 that faster connectors tend to be utilized more frequently. Among the three connector domains, air connectors such as KC-130s (cargo aircraft) and MV-22s (tiltrotor aircraft) are con-

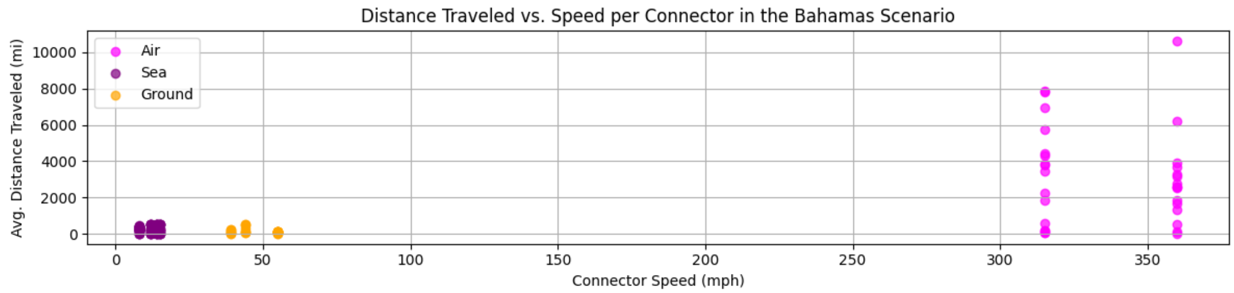


Figure 10 Average distance traveled by each connector type as a function of its speed. Each data point corresponds to a connector type—colored by domain—in an instance of the Bahamas scenario.

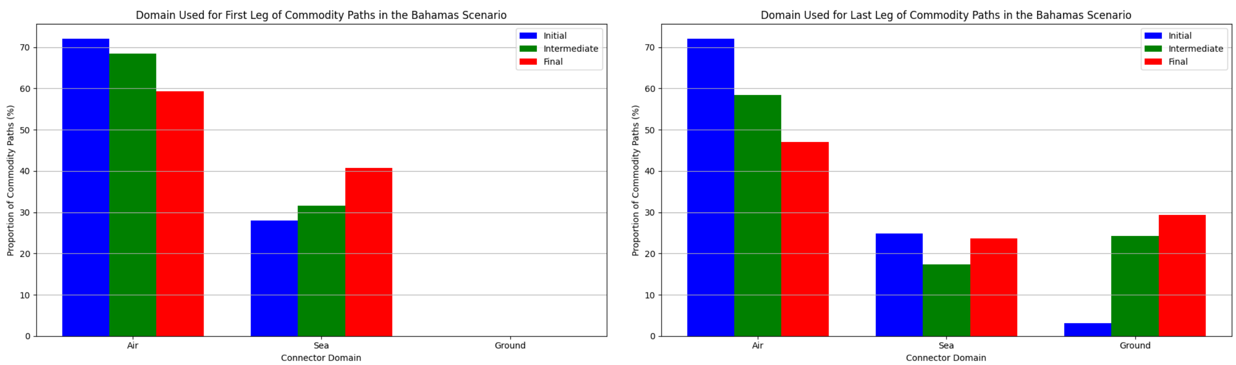


Figure 11 Distribution of connector domains used for transporting commodities on their first (left) and last (right) legs, stratified by whether the corresponding demand occurs in the initial, intermediate, or final third of the time horizon in the Bahamas scenario.

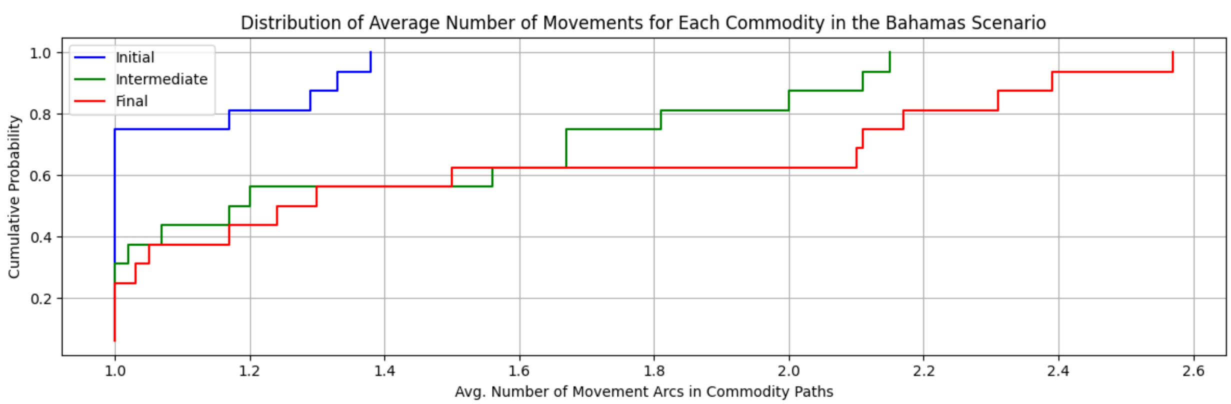


Figure 12 Empirical cumulative distribution function of the average number of movements per commodity across instances of the Bahamas scenario, stratified by whether the demand occurs in the initial, intermediate, or final third of the time horizon.

sistently the fastest and most heavily employed across all instances. This predominance is partly driven by the high concentration of demand requests occurring early in the time horizon. Such initial requests must be fulfilled rapidly, primarily via direct transportation using air connectors due to their superior speed and flexibility in accessing both supply and demand locations. This observation is corroborated by Figures 11 and 12, which indicate that nearly 75% of initial demands are satisfied through direct shipments, predominantly by air. In contrast, commodities serving later demand requests follow significantly more complex itineraries that combine multiple connector types across domains.

Finally, Figure 13 reports the average frequency with which each connector type's first movement is a "deadhead," meaning a leg performed without carrying commodities. The results show that some ground connectors initially stationed at EABs are misaligned with the optimal transshipment hubs identified by SOLVE-ELNDP, prompting empty repositioning before transporting commodities. This finding provides practical guidance to planners on how to improve the initial staging of connectors to reduce nonproductive movements at the onset of an operation.

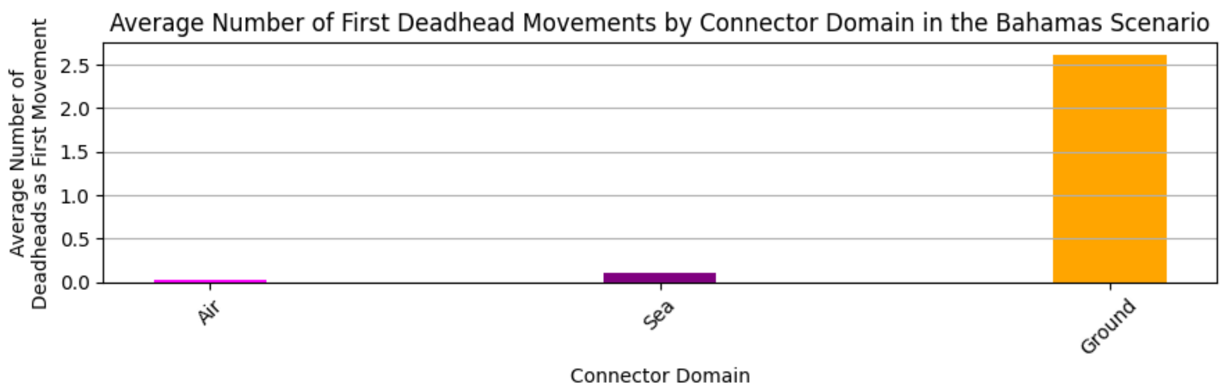


Figure 13 Average frequency of connectors' first legs being deadheads across instances of the Bahamas scenario.

5. Conclusions and Future Work

This work introduces the Expeditionary Logistics Network Design Problem (ELNDP), a novel model that captures the complex and high-stakes challenges of expeditionary logistics. By incorporating critical operational features—such as negligible transportation costs, flexible commodity sourcing, and multi-modal fleet coordination—ELNDP extends the classical Scheduled Service Network Design Problem (SSNDP) with explicit connector management, enabling a more realistic representation of military logistics environments. To solve ELNDP, we propose SOLVE-ELNDP, a dynamic discretization discovery (DDD) algorithm adapted from Boland et al. (2017) and tailored to the expeditionary setting. Unlike prior approaches, SOLVE-ELNDP employs a conservative relaxation strategy that overestimates arc travel times and introduces recovery arcs, ensuring computational tractability while enabling high-fidelity modeling. We further enhance the algorithm by implicitly incorporating out-and-back trips via capacity-factor adjustment and by developing a two-step multi-commodity maximum-flow based heuristic, which accelerate convergence and improve solution quality. Through a comprehensive case study based on scenarios validated by United States Marine Corps subject matter experts, we show that SOLVE-ELNDP consistently outperforms benchmark models—increasing demand fulfillment while reducing solve times relative to both fully time-expanded formulations and an adapted optimistic DDD baseline (SOLVE-CTSNDP*). These results underscore the potential of optimization-based decision support tools to enhance operational planning in expeditionary logistics. They also provide actionable managerial insights into how multi-modal coordination, time-discretization granularity, and proactive connector repositioning can improve the responsiveness and overall performance of expeditionary logistics operations.

Our computational results also reveal current challenges. The primary driver of higher solve times is the need to eliminate illegal connector cycles in the partially time-expanded networks, especially in large instances with dispersed high-volume demands. These findings highlight promising avenues for future research, including developing path-based formulations, hybrid heuristic approaches, or learning-augmented refinement strategies to accelerate convergence. While several enhancements to the original DDD algorithm have been proposed in the literature (Hewitt 2019, Marshall et al. 2021, Van Dyk and Koenemann 2024), further methodological advances are needed to incorporate the connector management requirements central to expeditionary environments.

Ultimately, this work represents a foundational step toward responsive, data-driven decision support tools for expeditionary logistics—tools capable of enhancing operational readiness and resilience in dynamic and resource-constrained environments. A promising direction is to extend ELNDP to handle uncertainty through stochastic and robust formulations that account for variable demand arrivals, connector availability, and disruptions, enabling solutions that balance expected performance with guaranteed service levels. In parallel, embedding SOLVE-ELNDP in a rolling-horizon framework that re-optimizes as new information becomes available could support adaptive, near-real-time expeditionary planning.

6. Code and Data Disclosure

The code to support the numerical experiments in this paper can be found in the uploaded zip file.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2039655 and by the Office of Naval Research under Grant No. N00014-24-1-2047. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications* (Englewood Cliffs, NJ: Prentice Hall).

Anaya-arenas AM, Renaud J, Ruiz A (2014) Relief distribution networks: a systematic review. *Annals of Operations Research* 223(1):53–79, ISSN 02545330, URL http://dx.doi.org/10.1007/s10479-014-1581-y, num Pages: 53-79 Place: New York, Netherlands Publisher: Springer Nature B.V.

Andersen J, Christiansen M, Crainic TG, Grønhaug R (2011) Branch and Price for Service Network Design with Asset Management Constraints. *Transportation Science* 45(1):33–49, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.1100.0333, publisher: INFORMS.

Andersen J, Crainic TG, Christiansen M (2009a) Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies* 17(2):197–207, ISSN 0968-090X, URL http://dx.doi.org/10.1016/j.trc.2008.10.005.

Andersen J, Crainic TG, Christiansen M (2009b) Service network design with management and coordination of multiple fleets. *European Journal of Operational Research* 193(2):377–389, ISSN 0377-2217, URL http://dx.doi.org/10.1016/j.ejor.2007.10.057.

Bakir I, Erera A, Savelsbergh M (2021) Motor carrier service network design. Crainic T, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics, chapter 14*, 427–467 (Springer).

Bender A, Cottle J, Craddock T, Dowd J, Feese R, Foster B, Gainey J, Jimenez I, Johnson B, Johnson T, Lemmon J, Levendofske M, Liskey D, Oliphant A, Olvera D, Partington W, Peace S, Tanks P (2004) *Seabasing and joint expeditionary logistics*. Ph.D. thesis, Naval Postgraduate School, Monterey, California, URL https://calhoun.nps.edu/server/api/core/bitstreams/63d86357-1eec-4bed-a883-2be4a226f5c7/content.

Boland N, Hewitt M, Marshall L, Savelsbergh M (2017) The Continuous-Time Service Network Design Problem. *Operations Research* 65(5):1303–1321, ISSN 0030-364X, URL http://dx.doi.org/10.1287/opre.2017.1624, publisher: INFORMS.

Boland N, Hewitt M, Marshall L, Savelsbergh M (2019) The price of discretizing time: a study in service network design. *EURO Journal on Transportation and Logistics* 8(2):195–216, ISSN 2192-4376, URL http://dx.doi.org/10.1007/s13676-018-0119-x.

Bradford JC (2006) The Missing Link: Expeditionary Logistics. URL https://www.usni.org/magazines/naval-history-magazine/2006/february/missing-link-expeditionary-logistics.

Caunhye AM, Nie X, Pokharel S (2012) Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences* 46(1):4–13, ISSN 0038-0121, URL http://dx.doi.org/10.1016/j.seps.2011.04.004.

Cheung J (2017) Expeditionary logistics overview. *Navy Supply Corps Newsletter* 80(1):12–14, ISSN 0360716X, URL https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=0360716X&v=2.1&it=r&id=GALE%7CA503262327&sid=googleScholar&linkaccess=abs, publisher: U.S. Department of the Navy, Supply Systems Command.

Crainic TG (2000) Service network design in freight transportation. *European Journal of Operational Research* 122(2):272–288, ISSN 0377-2217, URL http://dx.doi.org/10.1016/S0377-2217(99)00233-7.

Crainic TG, Hewitt M (2021) *Service Network Design*, 347–382 (Cham: Springer International Publishing).

Crainic TG, Hewitt M, Toulouse M, Vu DM (2016) Service Network Design with Resource Constraints. *Transportation Science* 50(4):1380–1393, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.2014.0525, publisher: INFORMS.

Crainic TG, Roy J (1988) OR tools for tactical freight transportation planning. *European Journal of Operational Research* 33(3):290–297, ISSN 0377-2217, URL http://dx.doi.org/10.1016/0377-2217(88)90172-5.

de Castro BA, Pochmann PGC, Neves EB (2024) Artificial Intelligence Applications in Military Logistics Operations. Rocha A, Fajardo-Toro CH, Rodríguez JMR, eds., *Developments and Advances in Defense and Security*, 89–100 (Singapore: Springer Nature), ISBN 978-981-9988-94-5, URL http://dx.doi.org/10.1007/978-981-99-8894-5_8.

Erera A, Hewitt M, Savelsbergh M, Zhang Y (2013) Improved Load Plan Design Through Integer Programming Based Local Search. *Transportation Science* 47(3):412–427, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.1120.0441, publisher: INFORMS.

Fontem B, Melouk SH, Keskin BB, Bajwa N (2016) A decomposition-based heuristic for stochastic emergency routing problems. *Expert Systems with Applications* 59:47–59, ISSN 0957-4174, URL http://dx.doi.org/10.1016/j.eswa.2016.04.002.

Ford LR, Fulkerson DR (1958) Constructing Maximal Dynamic Flows from Static Flows. *Operations Research* 6(3):419–433, ISSN 0030-364X, URL http://dx.doi.org/10.1287/opre.6.3.419, publisher: INFORMS.

Greening LM, Dahan M, Erera AL (2023) Lead-Time-Constrained Middle-Mile Consolidation Network Design with Fixed Origins and Destinations. *Transportation Research Part B: Methodological* 174:102782, ISSN 0191-2615, URL http://dx.doi.org/10.1016/j.trb.2023.102782.

Hewitt M (2019) Enhanced Dynamic Discretization Discovery for the Continuous Time Load Plan Design Problem. *Transportation Science* 53(6):1731–1750, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.2019.0890, publisher: INFORMS.

Hewitt M (2022) The Flexible Scheduled Service Network Design Problem. *Transportation Science* 56(4):1000–1021, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.2021.1114, publisher: INFORMS.

Hewitt M (2023) Consolidation-Based Modeling for the Scheduled Service Network Design Problem. Buscher U, Neufeld JS, Lasch R, Schönberger J, eds., *Logistics Management*, 3–24 (Cham: Springer Nature Switzerland), ISBN 978-3-031-38145-4, URL http://dx.doi.org/10.1007/978-3-031-38145-4_1.

Hewitt M, Lehuédé F (2023) New formulations for the Scheduled Service Network Design Problem. *Transportation Research Part B: Methodological* 172:117–133, ISSN 0191-2615, URL http://dx.doi.org/10.1016/j.trb.2023.04.002.

Joint Chiefs of Staff (2001) Joint Doctrine for Global Distribution. Joint Publication 4-09, URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6d71a946e49b0caea24e5ebdd834a97a366f32ff.

Joint Chiefs of Staff (2011) Planner's Handbook for Operational Design. Joint Publication J-7, URL https://www.jcs.mil/Portals/36/Documents/Doctrine/pams_hands/opdesign_hbk.pdf#:~:text=URL%3A%20https%3A%2F%2Fwww.jcs.mil%2FPortals%2F36%2FDocuments%2FDoctrine%2Fpams_hands%2Fopdesign_hbk.pdf%0AVisible%3A%200%25%20.

Lai MF, Lo HK (2004) Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A: Policy and Practice* 38(4):305–328, ISSN 0965-8564, URL http://dx.doi.org/10.1016/j.tra.2003.08.003.

Lin YH, Batta R, Rogerson PA, Blatt A, Flanigan M (2011) A logistics model for emergency supply of critical items in the aftermath of a disaster. *Socio-Economic Planning Sciences* 45(4):132–145, ISSN 0038-0121, URL http://dx.doi.org/10.1016/j.seps.2011.04.003.

Mangiaracina R, Song G, Perego A (2015) Distribution network design: a literature review and a research agenda. *International Journal of Physical Distribution & Logistics Management* 45(5):506–531, ISSN 0960-0035, URL http://dx.doi.org/10.1108/IJPDLM-02-2014-0035, publisher: Emerald Group Publishing Limited.

Marshall L, Boland N, Savelsbergh M, Hewitt M (2021) Interval-Based Dynamic Discretization Discovery for Solving the Continuous-Time Service Network Design Problem. *Transportation Science* 55(1):29–51, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.2020.0994, publisher: INFORMS.

McConnell BM, Hodgson TJ, Kay MG, King RE, Liu Y, Parlier GH, Thoney-Barletta K, Wilson JR (2021) Assessing uncertainty and risk in an expeditionary military logistics network. *The Journal of Defense Modeling and Simulation* 18(2):135–156, ISSN 1548-5129, URL http://dx.doi.org/10.1177/1548512919860595, publisher: SAGE Publications.

McKnight T (2019) Expeditionary Forces Are America's Military Crown Jewel. URL https://www.usni.org/magazines/proceedings/2019/january/expeditionary-forces-are-americas-military-crown-jewel.

National Research Council, Commission on Physical Sciences, Mathematics, and Applications, Naval Studies Board, Committee on Naval Expeditionary Logistics (1999) *Naval Expeditionary Logistics: Enabling Operational Maneuver from the Sea* (National Academies Press), ISBN 978-0-309-06429-3.

Neumann-Saavedra BA, Crainic TG, Gendron B, Mattfeld DC, Römer M (2016) Service Network Design of Bike Sharing Systems with Resource Constraints. Paias A, Ruthmair M, Voß S, eds., *Computational Logistics*, 352–366 (Cham: Springer International Publishing), ISBN 978-3-319-44896-1, URL http://dx.doi.org/10.1007/978-3-319-44896-1_23.

Pedersen MB, Crainic TG, Madsen OBG (2009) Models and Tabu Search Metaheuristics for Service Network Design with Asset-Balance Requirements. *Transportation Science* 43(2):158–177, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.1080.0234, publisher: INFORMS.

Reith J, Van Drew J, Hines T (2016) Operational Logistics Planner for a Leaner, More Capable Expeditionary Army. URL https://www.army.mil/article/176706/operational_logistics_planner_for_a_leaner_more_capable_expeditionary_army.

Rogers MB, McConnell BM, Hodgson TJ, Kay MG, King RE, Parlier G, Thoney-Barletta K (2018) A Military Logistics Network Planning System. *Military Operations Research* 23(4):5–24, ISSN 1082-5983, URL https://www.jstor.org/stable/26553094, publisher: Military Operations Research Society.

Scherr YO, Hewitt M, Neumann Saavedra BA, Mattfeld DC (2020) Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological* 141:164–195, ISSN 0191-2615, URL http://dx.doi.org/10.1016/j.trb.2020.09.009.

Sheu JB (2007) An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E: Logistics and Transportation Review* 43(6):687–709, ISSN 1366-5545, URL http://dx.doi.org/10.1016/j.tre.2006.04.004.

Song JH, Furman KC (2013) A maritime inventory routing problem: Practical approach. *Computers & Operations Research* 40(3):657–665, ISSN 0305-0548, URL http://dx.doi.org/10.1016/j.cor.2010.10.031.

Steinzen I, Gintner V, Suhl L, Kliewer N (2010) A Time-Space Network Approach for the Integrated Vehicle- and Crew-Scheduling Problem with Multiple Depots. *Transportation Science* 44(3):367–382, ISSN 0041-1655, URL http://dx.doi.org/10.1287/trsc.1090.0304, publisher: INFORMS.

United States Marine Corps (2023) Tentative Manual for Expeditionary Advanced Base Operations, 2nd ed. URL https://www.marines.mil/Portals/1/Docs/230509-Tentative-Manual-For-Expeditionary-Advanced-Base-Operations-2nd-Edition.pdf?ver=05KvG8wWlhI7uE0amD5uYg%3D%3D.

Van Dyk M, Koenemann J (2024) Sparse dynamic discretization discovery via arc-dependent time discretizations. *Computers & Operations Research* 169:106715, ISSN 0305-0548, URL http://dx.doi.org/10.1016/j.cor.2024.106715.

Wang X, Crainic TG, Wallace SW (2019a) Stochastic Network Design for Planning Scheduled Transportation Services: The Value of Deterministic Solutions. *INFORMS Journal on Computing* 31(1):153–170, ISSN 1091-9856, URL http://dx.doi.org/10.1287/ijoc.2018.0819, publisher: INFORMS.

Wang Y, Peng S, Xu M (2021) Emergency logistics network design based on space–time resource configuration. *Knowledge-Based Systems* 223:107041, ISSN 0950-7051, URL http://dx.doi.org/10.1016/j.knosys.2021.107041.

Wang Z, Qi M, Cheng C, Zhang C (2019b) A hybrid algorithm for large-scale service network design considering a heterogeneous fleet | Request PDF. *ResearchGate* URL http://dx.doi.org/10.1016/j.ejor.2019.01.022.

Wieberneit N (2008) Service network design for freight transportation: a review. *OR Spectrum* 30(1):77–112, ISSN 1436-6304, URL http://dx.doi.org/10.1007/s00291-007-0079-2.

Williams B (2010) Heuristics and Biases in Military Decision Making. *Military Review* .

Özdamar L, Demir O (2012) A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review* 48(3):591–602, ISSN 1366-5545, URL http://dx.doi.org/10.1016/j.tre.2011.11.003.

Özdamar L, Ekinci E, Küçükyazici B (2004) Emergency Logistics Planning in Natural Disasters. *Annals of Operations Research* 129(1):217–245, ISSN 1572-9338, URL http://dx.doi.org/10.1023/B:ANOR.0000030690.27939.39.

Online Appendix

EC.1. Proof of Theorem 1

Let $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H} \cup \widehat{R})$ be a partially time-expanded network and let (x, f, θ) be a feasible solution of FD. We aim to construct a feasible solution $(\bar{x}, \bar{f}, \theta)$ of SND$(\widehat{G})$ with identical objective value.

To this end, we first map each edge of $\widetilde{G}$ to a path in $\widehat{G}$. Specifically, for every arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widetilde{A}$, we consider the $\tilde{\imath}' - \tilde{\jmath}'$ path in $\widehat{G}$ satisfying $t_{\tilde{\imath}'} = \max\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L} \text{ and } t_{\tilde{\imath}^\dagger} \leq t_{\tilde{\imath}}\}$ and $t_{\tilde{\jmath}'} = \max\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L} \text{ and } t_{\tilde{\jmath}^\dagger} \leq t_{\tilde{\jmath}}\}$, and constructed as follows: The path originates at location $\tilde{\imath}' \in \widehat{L}$ and first traverses the movement arc $(\tilde{\imath}', \tilde{\jmath}^1, c) \in \widehat{A}$ for some timed copy $\tilde{\jmath}^1 \in \widehat{L}$ of j—which exists via Property 2. If $t_{\tilde{\jmath}^1} < t_{\tilde{\jmath}'}$, then the path traverses holdover arcs until reaching $\tilde{\jmath}'$. If instead $t_{\tilde{\jmath}^1} > t_{\tilde{\jmath}'}$, then the path traverses a recovery arc to reach $\tilde{\jmath}'$. We note that such a recovery arc exists due to Property 4 since $t_{\tilde{\jmath}^1} - t_{\tilde{\imath}'} > t_{\tilde{\jmath}} - t_{\tilde{\imath}} = \tau^{\Delta,c}_{i,j}$ by definition of $\tilde{\imath}'$ and $\tilde{\jmath}'$. Furthermore, Property 3 implies that one recovery arc is needed to reach $\tilde{\jmath}'$ from $\tilde{\jmath}^1$. Similarly, we map every arc $(\tilde{\imath}^1, \tilde{\imath}^2) \in \widetilde{H}$ to $(\tilde{\imath}^{1\prime}, \tilde{\imath}^{2\prime}) \in \widehat{H}$ satisfying $t_{\tilde{\imath}^{1\prime}} = \max\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L} \text{ and } t_{\tilde{\imath}^\dagger} \leq t_{\tilde{\imath}^1}\}$ and $t_{\tilde{\imath}^{2\prime}} = \max\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L} \text{ and } t_{\tilde{\imath}^\dagger} \leq t_{\tilde{\imath}^2}\}$ whenever $\tilde{\imath}^{1\prime} \neq \tilde{\imath}^{2\prime}$.

For every arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R}$, we denote by $\varphi(\tilde{\imath}, \tilde{\jmath}, c)$ the subset of arcs in $\widetilde{A} \cup \widetilde{H}$ that map to $(\tilde{\imath}, \tilde{\jmath}, c)$ using the process defined above. Then, we construct a solution $(\bar{x}, \bar{f}, \theta)$ of SND$(\widehat{G})$ as follows:

$$\forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R},\ \bar{x}^c_{\tilde{\imath},\tilde{\jmath}} = \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} x^c_{\tilde{\imath}',\tilde{\jmath}'}$$
$$\forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A} \cup \widehat{H} \cup \widehat{R},\ \forall k \in K,\ \bar{f}^{c,k}_{\tilde{\imath},\tilde{\jmath}} = \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} f^{c,k}_{\tilde{\imath}',\tilde{\jmath}'},$$

and θ has identical values to those in the feasible solution of FD. The solution to SND$(\widehat{G})$ is obtained by transferring and aggregating connector and commodity variable values to the (possibly multiple) mapped arcs in $\widehat{G}$.

Let P be an $\tilde{\imath} - \tilde{\imath}_{end}$ path in $\widetilde{G}$ taken by a connector of type $c \in C$. From Property 1, since $\tilde{\imath} \in \widetilde{S}^c \subset \widehat{L}$ and $\tilde{\imath}_{end} \in \widehat{L}$, then P is mapped to an $\tilde{\imath} - \tilde{\imath}_{end}$ walk in $\widehat{G}$. Similarly, since commodity origins and destinations are in $\widehat{G}$, then any commodity $\tilde{\imath} - \tilde{\jmath}$ path in $\widetilde{G}$ is mapped to an $\tilde{\imath} - \tilde{\jmath}$ walk in $\widehat{G}$. Therefore, connector and commodity flow conservation constraints (2b)-(2c) are naturally satisfied.

Furthermore, by identically aggregating connector and commodity variables on their mapped arcs, the capacity constraints (2d)-(2e) are satisfied:

$$\forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A},\quad \sum_{k \in K} w^k \cdot \bar{f}^{c,k}_{\tilde{\imath},\tilde{\jmath}} = \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} \sum_{k \in K} w^k \cdot f^{c,k}_{\tilde{\imath}',\tilde{\jmath}'} \leq u^c \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} x^{c,k}_{\tilde{\imath}',\tilde{\jmath}'} = u^c \cdot \bar{x}^{c,k}_{\tilde{\imath},\tilde{\jmath}}$$
$$\forall (\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A},\ \forall k \in K,\quad \bar{f}^{c,k}_{\tilde{\imath},\tilde{\jmath}} = \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} f^{c,k}_{\tilde{\imath}',\tilde{\jmath}'} \leq q^{c,k} \cdot \sum_{(\tilde{\imath}',\tilde{\jmath}',c) \in \varphi(\tilde{\imath},\tilde{\jmath},c)} x^{c,k}_{\tilde{\imath}',\tilde{\jmath}'} = q^{c,k} \cdot \bar{x}^{c,k}_{\tilde{\imath},\tilde{\jmath}}.$$

In conclusion, $(\bar{x}, \bar{f}, \theta)$ is a feasible solution to SND$(\widehat{G})$ with an objective value equal to that of (x, f, θ) in FD, and SND$(\widehat{G})$ is a relaxation of FD.

EC.2. Optimistic Dynamic Discretization Discovery for Solving ELNDP

In this section, we describe our adaptation of the original DDD algorithm—SOLVE-CTSNDP from Boland et al. (2017)—to solve ELNDP and serve as a benchmark for comparing with our proposed DDD algorithm, SOLVE-ELNDP. Because ELNDP includes modeling features such as connector management and flexible commodity sourcing—absent from Boland et al. (2017)—we adapt SOLVE-CTSNDP to optimally solve FD. The original DDD algorithm differs from our approach in that it underestimates travel times when constructing partially time-expanded networks. In Section EC.2.1, we describe properties that must be satisfied by partially time-expanded networks with such optimistic arcs. We then present the subroutines in Sections EC.2.2–EC.2.4 to create an initial optimistic partially time-expanded network, eliminate illegal connector cycles by lengthening movement arcs, and attempt to convert relaxed solutions into FD solutions. The overall algorithm, SOLVE-CTSNDP*, is described in Section EC.2.5.

EC.2.1. Optimistic Partially Time-Expanded Network

Given the flat network $G = (L, A)$ and the rounding factor Δ, we say that $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H})$ is an *optimistic partially time-expanded network* if the set of time-space locations $\widehat{L}$ and the set of movement arcs $\widehat{A}$ satisfy Properties 1–2, as well as the following property:

PROPERTY 5. *Every movement arc* $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$ *satisfies* $t_{\tilde{\jmath}} = \max\{t_{\tilde{\jmath}'} \mid \tilde{\jmath}' \in \widehat{L} \text{ and } t_{\tilde{\jmath}'} \leq t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c}\}$.

Additionally, the set of holdover arcs $\widehat{H}$ must connect consecutive time-space locations in $\widehat{L}$. Property 5 naturally extends the early arrival and longest feasible arc properties of Boland et al. (2017).

Consolidation plans can then be generated by solving SND($\widehat{G}$), which routes connectors and commodities on the now optimistic partially time-expanded network $\widehat{G}$. Theorem 1 can be extended to show that for any optimistic partially time-expanded network $\widehat{G}$, SND($\widehat{G}$) is a relaxation of FD.

EC.2.2. Initial Optimistic Partially Time-Expanded Network Construction

To construct the smallest initial optimistic partially time-expanded network that satisfies Properties 1, 2, and 5, we implement a similar procedure as in Section 3.2.1, except that movement arcs may underestimate travel times and no recovery arcs are constructed. The details are provided in Algorithm 6.

EC.2.3. Illegal Connector Cycle Elimination

An SND($\widehat{G}$) solution (x, f, θ) on an optimistic partially time-expanded network may also contain the illegal cycles described in Section 3.2.2. Such cycles form as a result of the underestimation of travel times and prevent the direct application of SOLVE-CTSNDP. Similarly, these cycles may inflate transportation capacities and lead to demand fulfillments that are not achievable by any FD solution. Therefore, we augment SOLVE-CTSNDP with IDENTIFY ILLEGAL CYCLES($\widehat{G}, x$) (Algorithm 2) to decompose the flow of connectors into walk flows $(\bar{x}_W^c)_{W \in \mathcal{W}^c}$ for every $c \in C$, and detect cycles that violate Conditions 1 and 2.

Algorithm 6: CREATE INITIAL OPTIMISTIC NETWORK*

Input : Directed network $G = (L, A)$, connector and commodity parameters
Output: Optimistic partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H})$

$\widehat{L} \leftarrow \bigcup_{k \in K} \left(\widetilde{I}^k \cup \widetilde{D}^k\right) \cup \bigcup_{c \in C} \left(\widetilde{S}^c\right) \cup \widetilde{L}_{start} \cup \widetilde{L}_{end}$
$\widehat{H} \leftarrow \{(\tilde{\imath}, \tilde{\imath}') \mid \tilde{\imath} \in \widehat{L} \setminus \widetilde{L}_{end}$ and $\tilde{\imath}' \in \arg\min\{t_{\tilde{\imath}^\dagger} \mid \tilde{\imath}^\dagger \in \widehat{L}$ and $t_{\tilde{\imath}^\dagger} > t_{\tilde{\imath}}\}\}$
$\widehat{A} \leftarrow \emptyset$
for $(i, j, c) \in A$ **do**
  for $\tilde{\imath} \in \widehat{L}$ **do**
    $\tilde{\jmath}' \in \arg\max\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L}$ and $t_{\tilde{\jmath}^\dagger} \le t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c}\}$, $\widehat{A} \leftarrow \widehat{A} \cup \{(\tilde{\imath}, \tilde{\jmath}', c)\}$
  end
end
return $\widehat{G}$

To eliminate an illegal cycle, we select a movement arc in the cycle that underestimates its travel time. We correct its length and add the corresponding time-space location in $\widehat{L}$, and then refine $\widehat{G}$ to maintain Properties 1, 2, and 5. This adjusted refinement procedure is detailed in Algorithm 7.

Algorithm 7: REFINE*$(\widehat{G}, (\tilde{\imath}, \tilde{\jmath}, c))$

Input : Optimistic partially time-expanded network $\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H})$, movement arc $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$
Output: New optimistic partially time-expanded network $\widehat{G}' = (\widehat{L}', \widehat{A}' \cup \widehat{H}')$

$\tilde{\jmath}' \leftarrow (j, t_{\tilde{\imath}} + \tau_{i,j}^{\Delta,c})$, $\tilde{\jmath}'' \in \arg\min\{t_{\tilde{\jmath}^\dagger} \mid \tilde{\jmath}^\dagger \in \widehat{L}$ and $t_{\tilde{\jmath}^\dagger} > t_{\tilde{\jmath}}\}$
$\widehat{L}' \leftarrow \widehat{L} \cup \{\tilde{\jmath}'\}$, $\widehat{A}' \leftarrow \widehat{A} \setminus \{(\tilde{\imath}, \tilde{\jmath}, c)\} \cup \{(\tilde{\imath}, \tilde{\jmath}', c)\}$, $\widehat{H}' \leftarrow \widehat{H} \setminus \{(\tilde{\jmath}, \tilde{\jmath}'')\} \cup \{(\tilde{\jmath}, \tilde{\jmath}'), (\tilde{\jmath}', \tilde{\jmath}'')\}$
for $(\tilde{\ell}, \tilde{\jmath}, c') \in \widehat{A}'$ **do**
  if $t_{\tilde{\ell}} + \tau_{\ell,j}^{\Delta,c'} \ge t_{\tilde{\jmath}'}$ **then**
    $\widehat{A}' \leftarrow \widehat{A}' \setminus \{(\tilde{\ell}, \tilde{\jmath}, c')\} \cup \{(\tilde{\ell}, \tilde{\jmath}', c')\}$
  end
end
for $(j, \ell, c') \in A$ **do**
  $\tilde{\ell}' \in \arg\max\{t_{\tilde{\ell}^\dagger} \mid \tilde{\ell}^\dagger \in \widehat{L}'$ and $t_{\tilde{\ell}^\dagger} \le t_{\tilde{\jmath}'} + \tau_{j,\ell}^{\Delta,c'}\}$, $\widehat{A}' \leftarrow \widehat{A}' \cup \{(\tilde{\jmath}', \tilde{\ell}', c')\}$
end
return $\widehat{G}'$

EC.2.4. Converting into an FD Solution

Given an SND$(\widehat{G})$ solution that does not contain any illegal cycles, we next formulate a dispatch problem that aims to construct a feasible FD solution with the same objective value. Let $\overline{\mathcal{W}}$ denote the set of connector and commodity walks obtained via Algorithm 2 and by decomposing the flow of commodities. In Boland et al. (2017), the dispatch problem aims to minimize the number of commodity movements that violate travel times. We adjust their formulation to also account for connector movement as follows: Using notation similar to that introduced in Section 3.2.3, we define for every connector and commodity walk $W \in \overline{\mathcal{W}}$ and every arc $a \in W$ the variable $\gamma_a^W \geq 0$ to represent the dispatch time when traversing arc a. For every movement arc $(\tilde{\imath}, \tilde{\jmath}, c) \in W \cap \widehat{A}$, we also define $\zeta_{\tilde{\imath},\tilde{\jmath}}^{W,c} \geq 0$ to represent the selected travel time on a and $\sigma_{\tilde{\imath},\tilde{\jmath}}^{W,c} \in \{0,1\}$ to reflect whether or not the selected travel time is shorter than its actual travel time $\tau_{i,j}^{\Delta,c}$. The dispatch problem, denoted CCD*($\overline{\mathcal{W}}$), can be formulated as the following MIP:

$$\text{CCD*}(\overline{\mathcal{W}}): \min_{\gamma,\zeta,\sigma} \sum_{W \in \overline{\mathcal{W}}} \sum_{(\tilde{\imath},\tilde{\jmath},c) \in W \cap \widehat{A}} \sigma_{\tilde{\imath},\tilde{\jmath}}^{W,c} \tag{EC.1a}$$

$$\text{s.t.} \quad \gamma_{a_{\ell+1}}^W - \gamma_{a_\ell}^W \geq \begin{cases} \zeta_{\tilde{\imath},\tilde{\jmath}}^{W,c} & \text{if } a_\ell^W = (\tilde{\imath},\tilde{\jmath},c) \in \widehat{A}, \\ 0 & \text{if } a_\ell^W \in \widehat{H}, \end{cases} \quad \forall W \in \overline{\mathcal{W}},\ \forall \ell \in \{1,\ldots,|W|\}, \tag{EC.1b}$$

$$\zeta_{\tilde{\imath},\tilde{\jmath}}^{W,c} \geq \tau_{i,j}^{\Delta,c} - (\tau_{i,j}^{\Delta,c} - t_{\tilde{\jmath}} + t_{\tilde{\imath}})\sigma_{\tilde{\imath}',\tilde{\imath}}^{W,c}, \quad \forall W \in \overline{\mathcal{W}},\ \forall (\tilde{\imath},\tilde{\jmath},c) \in \overline{\mathcal{W}} \cap \widehat{A}, \tag{EC.1c}$$

$$\gamma_{a_1}^W \geq t_{start}^W, \quad \forall W \in \overline{\mathcal{W}}, \tag{EC.1d}$$

$$\gamma_{a_{|W|+1}}^W \leq t_{end}^W, \quad \forall W \in \overline{\mathcal{W}}, \tag{EC.1e}$$

$$\gamma_a^{W_1} = \gamma_a^{W_2}, \quad \forall a \in \widehat{A},\ \forall (W_1,W_2) \in \overline{\mathcal{W}}^2 \mid a \in W_1 \cap W_2, \tag{EC.1f}$$

$$\sigma_{\tilde{\imath}',\tilde{\imath}}^W \in \{0,1\}, \quad \forall W \in \overline{\mathcal{W}},\ \forall (\tilde{\imath},\tilde{\jmath},c) \in W \cap \widehat{A}. \tag{EC.1g}$$

Similarly, CCD*($\overline{\mathcal{W}}$) quantifies the level of infeasibility of the SND($\widehat{G}$) solution. If its optimal value is zero, the dispatch times along each walk can be used to convert the SND($\widehat{G}$) solution into a feasible FD solution with the same objective value, guaranteeing its optimality for FD. On the other hand, if the optimal value of CCD*($\overline{\mathcal{W}}$) is strictly positive, the movement arcs $(\tilde{\imath},\tilde{\jmath},c)$ for which $\sigma_{\tilde{\imath},\tilde{\jmath}}^{W,c} = 1$ at optimality for some walk $W \in \overline{\mathcal{W}}$ must be lengthened using REFINE*($\widehat{G}, (\tilde{\imath},\tilde{\jmath},c)$) to improve the SND($\widehat{G}$) solution.

EC.2.5. Overall SOLVE-CTSNDP* Algorithm

The formal description of our adaption of SOLVE-CTSNDP for solving ELDNP is provided in Algorithm 8.

Note that at each iteration of SOLVE-CTSNDP*, the two-step heuristic developed in Section 3.3.2 can be applied to construct a feasible FD solution from an SND($\widehat{G}$) solution and compute its associated optimality gap.

Algorithm 8: SOLVE-CTSNDP*

Input : Expeditionary network $G = (L, A)$, connector and commodity parameters
Output: Optimal solution (x^*, f^*, θ^*) of FD

$\widehat{G} = (\widehat{L}, \widehat{A} \cup \widehat{H}) \leftarrow$ CREATE INITIAL OPTIMISTIC NETWORK* (Algorithm 6)
do
  Solve SND($\widehat{G}$): $(x, f, \theta) \leftarrow$ optimal solution
  $(\mathcal{I}, (\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c})_{c \in C}) \leftarrow$IDENTIFY ILLEGAL CYCLES$(\widehat{G}, x)$ (Algorithm 2)
  if $\mathcal{I} \neq \emptyset$ **then**
    for $\xi \in I$ **do**
      Select $(\tilde{\imath}, \tilde{\jmath}, c) \in \xi$ such that $t_{\tilde{\jmath}} < t_{\tilde{\imath}} + \tau^{\Delta,c}_{i,j}$
      $\widehat{G} \leftarrow$ REFINE*$(\widehat{G}, (\tilde{\imath}, \tilde{\jmath}, c))$ (Algorithm 7)
    end
  else
    Decompose every commodity flow f^k (for $k \in K$) into path flows $(\bar{f}^k_W)_{W \in \mathcal{W}^k}$
    $\overline{\mathcal{W}} \leftarrow \left(\bigcup_{c \in C} \mathcal{W}^c\right) \cup \left(\bigcup_{k \in K} \mathcal{W}^k\right)$
    Solve CCD*$(\overline{\mathcal{W}})$: $(\gamma, \zeta, \sigma) \leftarrow$ optimal solution, $z \leftarrow$ optimal value
    if $z > 0$ **then**
      for $(\tilde{\imath}, \tilde{\jmath}, c) \in \widehat{A}$ *such that* $\sum_{\{W \in \overline{\mathcal{W}} \,|\, (\tilde{\imath}, \tilde{\jmath}, c) \in W\}} \sigma^{W,c}_{\tilde{\imath},\tilde{\jmath}} > 0$ **do**
        $\widehat{G} \leftarrow$ REFINE*$(\widehat{G}, (\tilde{\imath}, \tilde{\jmath}, c))$ (Algorithm 7)
      end
    else
      for $c \in C$ **do**
        $(x^{c^*}_{\tilde{\imath},\tilde{\jmath}})_{(\tilde{\imath},\tilde{\jmath},c) \in \tilde{A} \cup \tilde{H}} \leftarrow$ CONSTRUCT FLOW$(\mathcal{W}^c, (\bar{x}^c_W)_{W \in \mathcal{W}^c}, (\gamma^W)_{W \in \mathcal{W}^c})$ (Algorithm 4)
      end
      for $k \in K$ **do**
        $(f^{c,k^*}_{\tilde{\imath},\tilde{\jmath}})_{(\tilde{\imath},\tilde{\jmath},c) \in \tilde{A} \cup \tilde{H}} \leftarrow$ CONSTRUCT FLOW$(\mathcal{W}^k, (\bar{f}^k_W)_{W \in \mathcal{W}^k}, (\gamma^W)_{W \in \mathcal{W}^k})$ (Algorithm 4)
      end
      $\theta^* \leftarrow \theta$
    end
  end
while $\mathcal{I} \neq \emptyset$ *or* $z > 0$
return (x^*, f^*, θ^*)

EC.3. Additional Case Study Details

EC.3.1. Case Study Characteristics

Tables EC.1 and EC.2 present the characteristics of the connector and commodity types used in our case study outlined in Section 4.

Table EC.1 Connector characteristics

Connector type	Domain	Speed (mi/h)	Range (mi)	Capacity			
				Passengers	Fuel (lb)	Total volume (cu ft)	Total weight (lb)
KC-130	Air	360	4,000	92	10k	3,600	42k
MV-22	Air	315	380	24	2k	1,025	6k
ASC	Sea	8	500	0	3k	23,000	200k
LAW	Sea	16	1,000	75	60k	35,000	1.7k
LCU-1700	Sea	13	1,200	0	6k	42,350	280k
SLV	Sea	14	11,750	50	60k	64,000	1.3M
OSV	Sea	12	3,500	36	60k	10,000	200k
T-EPF	Sea	15	4,700	312	10k	297,600	1.2M
MTVR	Ground	55	300	15	1.8k	400	14k
MTVR-Trailer	Ground	55	300	15	1.8k	650	26k
Railway	Ground	10	122	0	0	500,000	10M
Truck-5	Ground	39	93	0	0	3,500	35k
Truck-6	Ground	44	104	0	7.2k	2,500	35k

Table EC.2 Commodity characteristics. Fuel is stored in dedicated tanks and does not utilize the aggregated connector volume capacity reported in Table EC.1.

Class of supply	Commodity description	# Commodity types	Volume (cu ft)	Weight (lb)
I	Food, water	3	[1.2, 64]	[22, 2274]
II	Equipment	10	[0.035, 60]	[2,16]
III	Fuel	3	0	1
V	Ammunition, fuses, detonators, missiles	39	[0.1, 57.0]	[20, 2900]
VII	Launchers, vehicles	8	[0.24, 11648]	[9, 28000]
PAX	Passengers	1	14	210

EC.3.2. Computational Results

The detailed results of the computational study conducted in Section 4.2 are presented in Table EC.3.

Table EC.3 Comparison of solution approaches on the 53 instances from Section 4 with a 1-hour time limit. The columns respectively present the scenarios, rounding factors, solution methods, average fulfillment rates, average optimality gaps, average CPU times, percentages of instances with nontrivial solutions (fulfilling a strictly positive amount of demand), average number of variables in final MIP, and average number of cycle-elimination iterations.

Scenario	Round. factor Δ	Solution Method	Fulfill. rate (%)	Opt. gap (%)	CPU time (s)	Nontrivial sol. (%)	# Variables in final MIP	# Cycle elim. iterations
Okinawa	6 hours	FD	43.85	52.29	2,337	45.8	325.4k	–
		SOLVE-CTSNDP*	85.79	8.75	2,524	100.0	91.5k	6.42
		SOLVE-ELNDP	85.39	8.87	1,088	100.0	13.3k	2.13
	3 hours	FD	11.32	86.39	3,432	12.5	649.0k	–
		SOLVE-CTSNDP*	88.12	6.82	2,521	100.0	102.0k	7.33
		SOLVE-ELNDP	85.56	9.92	1,080	100.0	12.6k	2.21
	1 hour	FD	0.00	100.00	3,757	0.0	765.2k	–
		SOLVE-CTSNDP*	87.70	8.37	2,337	100.0	115.6k	6.96
		SOLVE-ELNDP	89.66	6.23	1,101	100.0	12.4k	2.46
	1 min	FD	–	–	–	–	–	–
		SOLVE-CTSNDP*	88.61	7.07	2,462	100.0	121.5k	5.79
		SOLVE-ELNDP	88.78	6.70	1,275	100.0	11.4k	2.17
SoCal	6 hours	FD	88.93	0.14	16	100.0	60.8k	–
		SOLVE-CTSNDP*	61.50	32.07	3,367	100.0	46.6k	15.08
		SOLVE-ELNDP	83.18	6.98	1,699	100.0	35.5k	6.23
	3 hours	FD	80.75	15.58	1,673	84.6	119.1k	–
		SOLVE-CTSNDP*	70.59	26.47	3,603	100.0	71.8k	12.38
		SOLVE-ELNDP	91.30	4.53	2,863	100.0	43.7k	5.69
	1 hour	FD	0.00	100.00	3,631	0.0	349.6k	–
		SOLVE-CTSNDP*	76.20	23.69	3,405	100.0	89.5k	9.23
		SOLVE-ELNDP	88.22	11.65	3,441	100.0	34.0k	3.77
	1 min	FD	–	–	–	–	–	–
		SOLVE-CTSNDP*	75.55	24.43	3,607	100.0	113.0k	6.77
		SOLVE-ELNDP	81.10	18.88	3,607	100.0	19.0k	1.92
Bahamas	6 hours	FD	3.60	6.32	543	93.8	55.7k	–
		SOLVE-CTSNDP*	1.32	66.80	3,602	100.0	21.7k	1.69
		SOLVE-ELNDP	3.35	22.86	2,473	100.0	19.1k	9.38
	3 hours	FD	2.84	50.01	1,879	50.0	109.3k	–
		SOLVE-CTSNDP*	1.81	73.32	3,596	100.0	19.9k	1.13
		SOLVE-ELNDP	4.62	32.15	3,042	100.0	21.6k	13.19
	1 hour	FD	1.54	75.01	2,775	25.0	321.2k	–
		SOLVE-CTSNDP*	2.03	77.89	3,596	100.0	19.4k	0.44
		SOLVE-ELNDP	5.32	37.18	3,531	100.0	16.8k	14.63
	1 min	FD	–	–	–	–	–	–
		SOLVE-CTSNDP*	1.73	85.61	3,597	100.0	18.9k	0.50
		SOLVE-ELNDP	5.97	43.44	3,596	100.0	9.4k	15.75

EC.3.3. Additional Scenario Analysis

We analyze the solutions to each instance returned by SOLVE-ELNDP using the best-performing-on-average rounding factor for each remaining scenario (Socal and Okinawa). Similarly to Section 4.3, Figure EC.1 presents the relationship between connector speed and the average distance traveled, Figure EC.2 illustrates the distribution of connector domains used for commodities' first and last movement legs, Figure

EC.3 reports the distribution of the average number of movement arcs per commodity, and Figure EC.4 shows the average frequency with which each connector type's first movement is a deadhead.

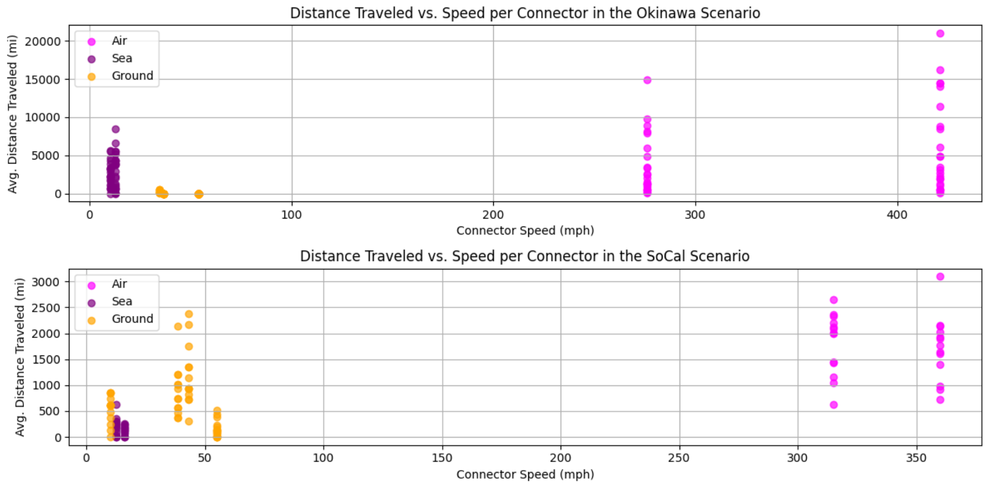


Figure EC.1 Average distance traveled by each connector type as a function of its speed. Each data point corresponds to a connector type—colored by domain—in an instance of the Okinawa and SoCal scenarios.

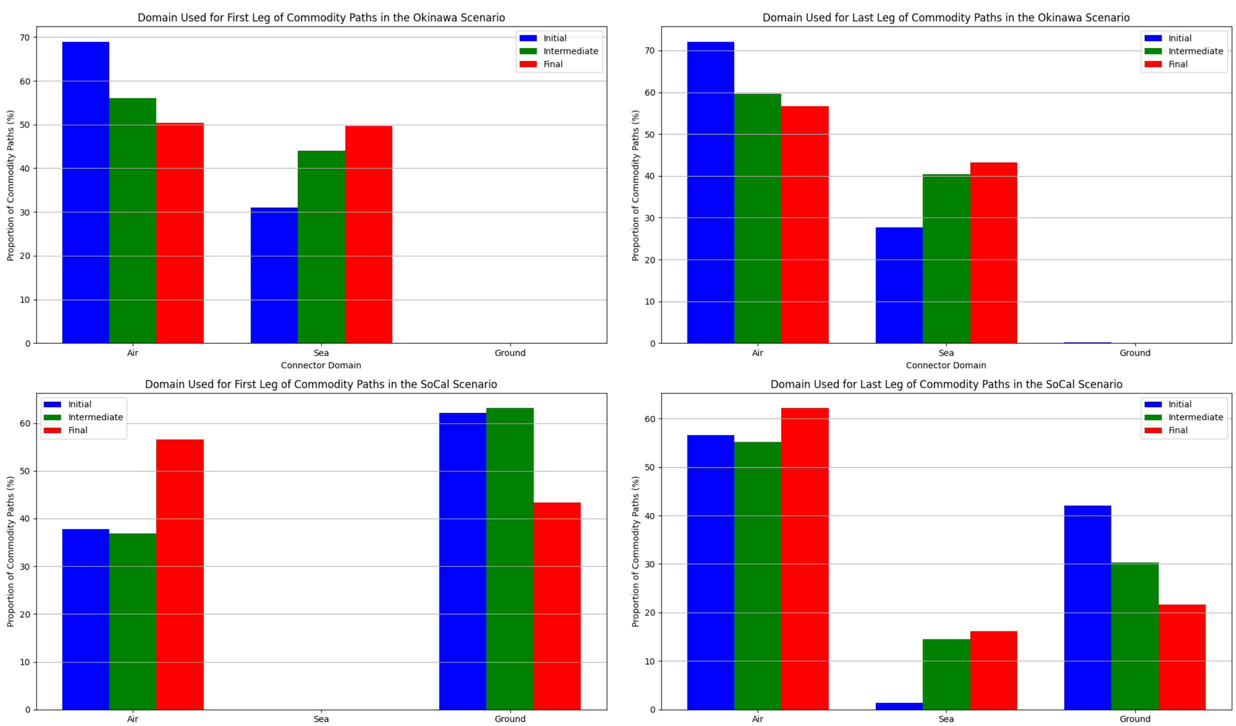


Figure EC.2 Distribution of connector domains used for transporting commodities on their first (left) and last (right) legs, stratified by whether the corresponding demand occurs in the initial, intermediate, or final third of the time horizon in the Okinawa and SoCal scenarios.

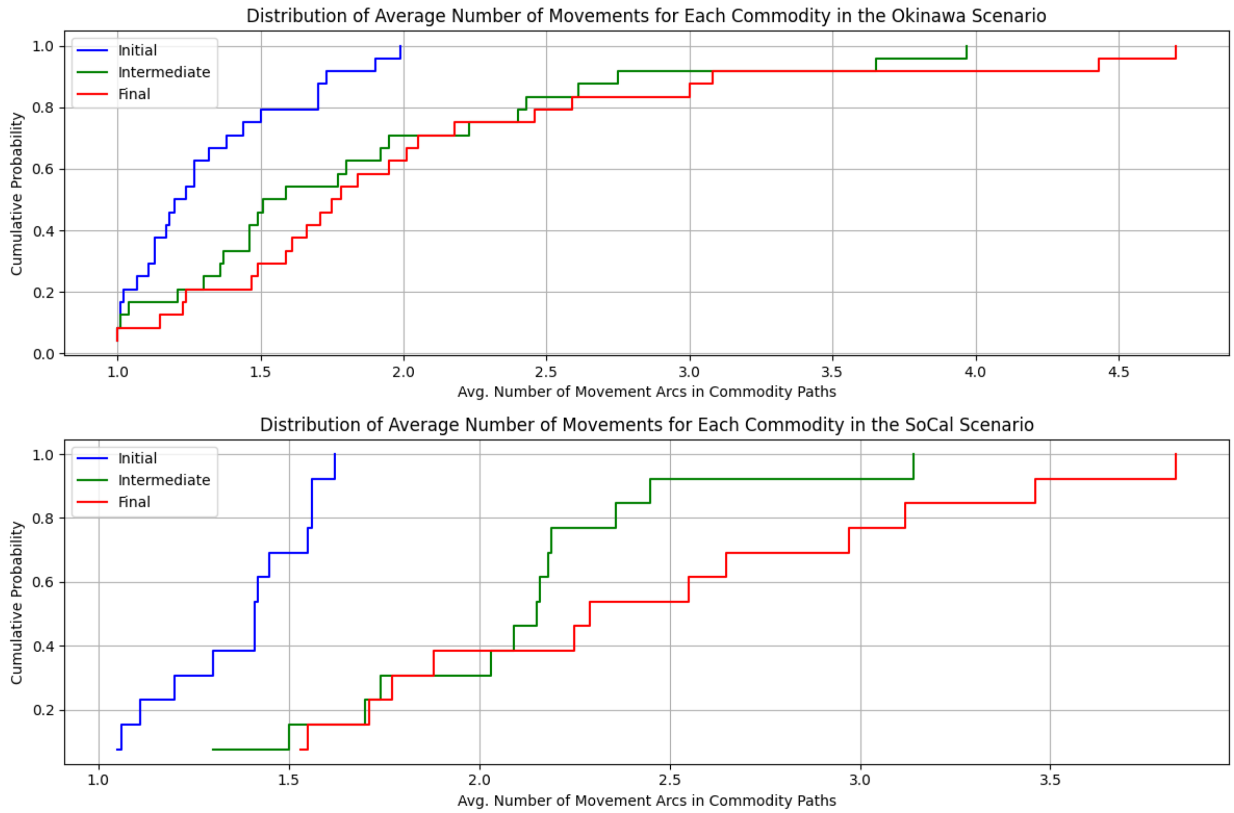


Figure EC.3 Empirical cumulative distribution function of the average number of movements per commodity across instances of the Okinawa and SoCal scenarios, stratified by whether the demand occurs in the initial, intermediate, or final third of the time horizon.

We observe slight differences in the characteristics of the consolidation plans constructed by SOLVE-ELNDP across scenarios. First, the Okinawa scenario relies more heavily on sea connectors, while the SoCal scenario more frequently employs ground connectors, reflecting the distinct topographical features of each theater. Second, we find that commodity itineraries in these two scenarios are more complex than in the Bahamas scenario, occasionally requiring more than four connectors to reach their destinations. Finally, the SoCal scenario exhibits the highest frequency of misplaced connectors in the air domain. Examination of the solutions reveals that air connectors often begin at EABs and subsequently deadhead to supply bases to satisfy urgent demand requests. This suggests that, in the SoCal setting, proactively stationing additional air connectors at supply bases could improve responsiveness to time-sensitive demands.

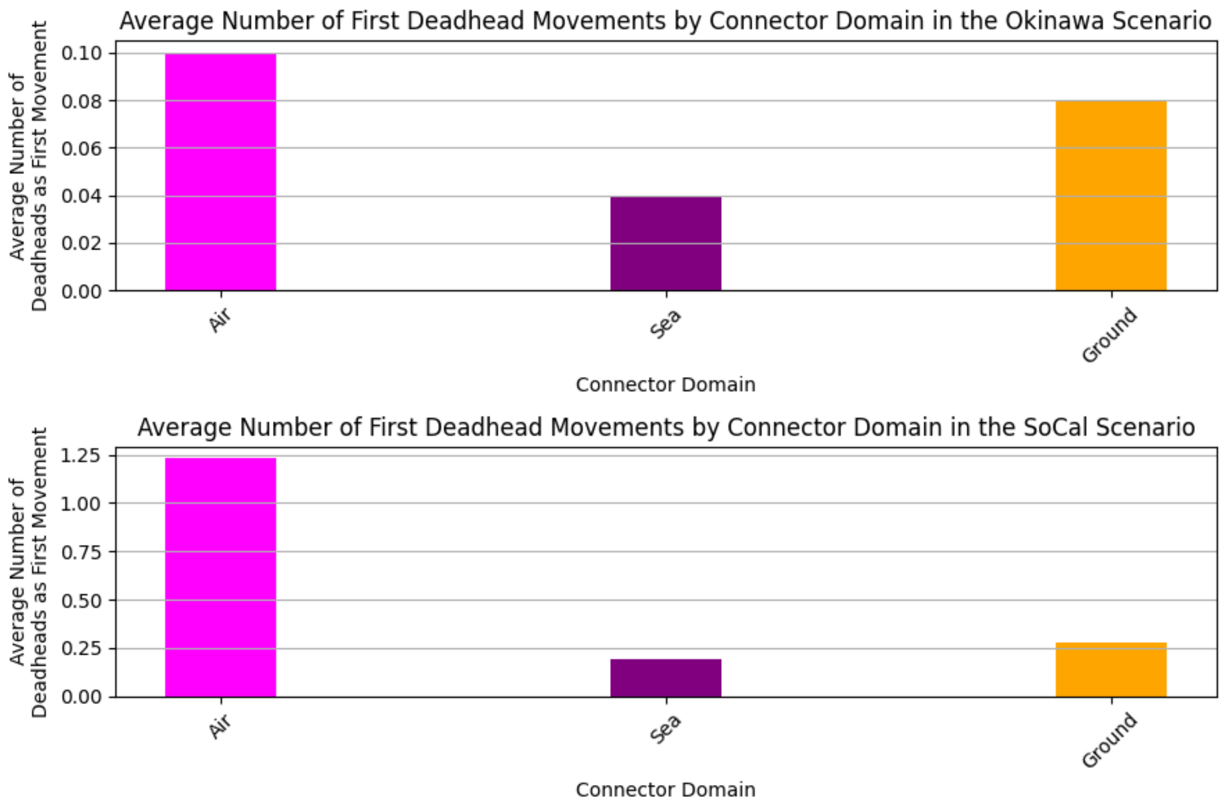


Figure EC.4 Average frequency of connectors' first legs being deadheads across instances of the Okinawa and SoCal scenarios.