Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

JUNKO HOSODA

STEPHEN J. MAHER

YUJI SHINANO

JONAS CHRISTOFFER VILLUMSEN

Location, transshipment and routing: An adaptive transportation network integrating long-haul and local vehicle routing

*The work for this article has been partially conducted within the Research Campus MODAL funded by the German Federal Ministry of Education and Research (fund number 05M14ZAM)

ZIB-Report 21-12 (May 2021)

Herausgegeben vom

Konrad-Zuse-Zentrum für Informationstechnik Berlin

Takustraße 7

D-14195 Berlin-Dahlem

Telefon: 030-84185-0

Telefax: 030-84185-125

e-mail: bibliothek@zib.de

URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064

ZIB-Report (Internet) ISSN 2192-7782

Location, transshipment and routing: An adaptive transportation network integrating long-haul and local vehicle routing

Junko Hosoda[1], Stephen J. Maher*[2], Yuji Shinano[3], and Jonas Christoffer Villumsen[4]

[1]Center for Technology Innovation – Production Engineering, Hitachi Ltd., Kanagawa, Japan

[2]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, United Kingdom

[3]Department of Applied Algorithmic Intelligence Methods (A^{2}IM), Zuse Institute Berlin, Berlin, Germany

[4]European Research and Development Centre, Hitachi Europe Ltd., Copenhagen, Denmark

Abstract

The routing of commodities is a tactical problem in supply chain management that aims to synchronise transportation services connecting a network of warehouses and consolidation locations. This paper considers the routing of commodities in a transportation network that is flexible in response to demand through changes to regional warehouse clustering and the designation of consolidation locations. Traditionally, warehouse clustering and consolidation locations are determined as part of strategic planning that is performed months to years in advance of operations—limiting the flexibility in transportation networks to respond to changes in demand. A mathematical programming-based algorithmic framework is proposed to integrate the strategic decisions of location planning with tactical decisions of vehicle routing and synchronisation. A multi-armed bandit problem is developed to explore warehouse clustering decisions and exploit those that lead to small transportation costs. An extensive computational study will show that the proposed algorithmic framework effectively integrates strategic and tactical planning decisions to reduce the overall transportation costs.

Key words: supply chain management, vehicle routing, location, synchronisation, multi-armed bandit

*Corresponding author: s.j.maher@exeter.ac.uk

1 Introduction

Supply chain networks are complex systems comprised of many interconnected facilities and resources. Facilities that share similar characteristics are typically grouped into tiers, where commodities *flow* down through the tiers—from the manufacturer to the shop front. Efficiency in supply chains is highly dependent on the effective storage, consolidation and transportation of commodities. This paper aims to improve supply chain efficiency with a focus on the consolidation and transportation of commodities between facilities within a single tier of the supply chain network.

Warehouses are large consolidation points that are used to store commodities prior to their transportation to distribution depots. Figure 1 shows the possible movement of commodities through the final three tiers of the supply chain network. Traditionally, the inter-warehouse transfer of commodities relies on point-to-point transportation—providing little opportunity for consolidation. By introducing consolidation at warehouses, the inter-warehouse transportation network can be adapted to reduce operational costs through the clustering of warehouses, selection of consolidation locations and construction of multi-stop vehicle routes.

This paper considers an inter-warehouse transportation problem encountered by our project partner where the warehouse clusters, consolidation locations and transportation schedules are flexible and determined with respect to demand. Our project partner previously constructed

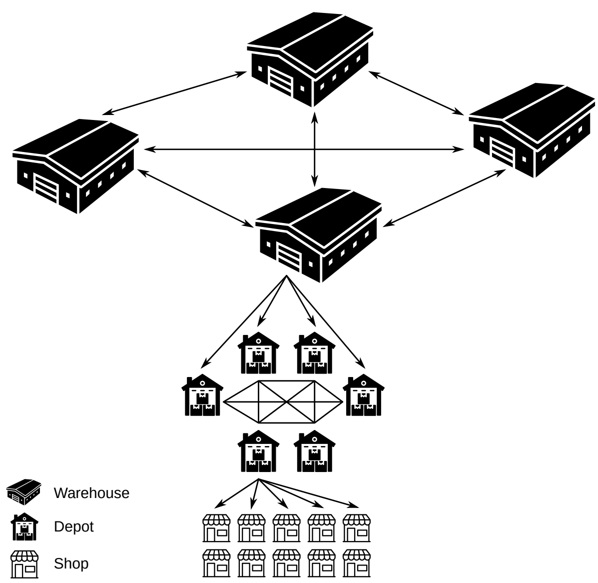


Figure 1: Three tiers in a supply chain network.

numerous large warehouses to consolidate commodities and relied on point-to-point transportation between the warehouses. Since the number of pickup and delivery requests between each pair of warehouses is related to demand, this method of operation has become inefficient. This has prompted a change in operations to design an inter-warehouse transportation network that is adaptable to changes in demand. The warehouses will be redesigned so that they can be simply converted into consolidated locations as required—which are selected after a monthly review of the demand forecasts. Our study develops the tools necessary to identify warehouse clusters and consolidation locations with respect to the forecast demand. The problem considered in this paper integrates location clustering, transshipment and vehicle routing problems, which will be described as the supply chain and service network design problem (SCSNDP).

The contributions of this paper are as follows: We (i) present a new supply chain management application that exploits demand-based location clustering to reduce operational costs in adaptive transportation networks. To address the scale of this application (ii) a mathematical programming-based algorithmic framework is developed to formulate the SCSNDP that effectively integrates location clustering, transshipment and vehicle routing problems. The adaptive nature of the underlying application motivates the development of a reinforcement learning problem. To this end, (iii) a multi-armed bandit (MAB) problem is proposed as a method to explore and exploit warehouse clustering solutions that lead to reduced intra- and inter-cluster transportation costs. Most importantly, (iv) a novel iterative algorithm is developed to synchronise intra- and inter-cluster transportation routes at consolidation locations. The proposed solution algorithm is enhanced by various algorithmic techniques to improve the solution quality of the SCSNDP. Finally, (v) an extensive set of computational experiments will demonstrate the effectiveness of the SCSNDP at reducing transportation costs.

The definition of the SCSNDP and a discussion of related work is presented in Section 2. The mathematical programming problems modelling the major operational decisions of the SCSNDP are presented in Section 3. A description of the solution algorithm and novel techniques used to improve the solution quality of the SCSNDP is provided in Section 4. Section 5 presents the computational experiments that demonstrate the potential of the proposed solution algorithms to reduce transportation costs.

2 The supply chain and service network design problem

The SCSNDP comprises a set of warehouses and a set of pickup and delivery requests between each pair of warehouses. To support the efficient transfer of goods between warehouses, a set of clusters can be defined based on the geographical proximity of the warehouses. Within each of these clusters, any of the warehouses can be selected as a consolidation location for the intra-warehouse transportation of commodities. The combination of the clustered warehouses and the size of the considered geographical region is such that the pickup and delivery of commodities may require transportation on more than one vehicle route. This paper considers the problem of selecting warehouse clusters and consolidation locations and identifying synchronised intra- and inter-cluster routes to satisfy all pickup and delivery requests at the lowest cost.

2.1 Related work

The integration of location, transshipment and vehicle routing in the SCSNDP draws upon concepts from many different supply chain management applications. In particular, location-routing [11,17], vehicle routing with synchronisation [10], service network design (SNDP) [8,26], multi-echelon vehicle routing [18] and vehicle routing with pickup and delivery (PDP) [9, 23] all exhibit characteristics that are observed in the SCSNDP.

The integration of clustering, location and routing is a core feature of the location-routing problem [11, 17]. In response to the difficulty of solving the integrated location and routing problems, Barreto et al. [2] proposed various different clustering methods to improve the solution quality. The impact of the clustering approach is further discussed by Lam and Mittenthal [13] in the context of a multi-depot location-routing problem. Specifically, the location clustering is shown by Lam and Mittenthal [13] to significantly impact the overall vehicle routing cost. The results from Barreto et al. [2] and Lam and Mittenthal [13] motivate the cluster-based adaptive transportation network underlying the SCSNDP.

A major limitation of traditional location-routing problems is that the transportation of commodities between the clusters is typically ignored. The combination of multi-echelon vehicle routing and location-routing problems, as described by Contardo et al. [5], aims to address this limitation. Wang et al. [24] further investigate the multi-echelon location-routing problem by developing alternative customer clustering techniques. While the multi-echelon vehicle routing problem introduces movement of commodities between clusters, this movement is only one directional until the last-mile delivery. Hence, the possibility to transport commodities to and from warehouses located in different clusters is not considered.

An extension of the location-routing problem that introduces adaptive transportation networks is presented by Salama and Srinivas [21]. Specifically, Salama and Srinivas [21] investigate a novel vehicle routing application that integrates truck and drone routing for last-mile delivery. The customer locations are clustered based on demand and whether they can be served by a drone for last-mile delivery. However, the drone capacity restricts the intra-cluster routes to direct links between the cluster centres and the delivery locations.

The cluster-based adaptive transportation network of the SCSNDP draws upon may features from facility location and network design problems. Melkote and Daskin [15] discuss a uncapacitated facility location and network design problem that is a precursor to many location routing problems. In a broader investigation, Contreras et al. [6] propose a framework for classifying problems that combine location and network design. The SCSNDP fits within the framework presented by Contreras et al. [6]; however, the synchronisation of intra- and inter-cluster routes introduces an extension beyond the classes of problems that are discussed.

Focusing on less-than-truckload freight operators, the SNDP is tasked with identifying a transportation schedule for the movement of goods between consolidation locations. Extensive reviews of formulations and solution approaches for the SNDP are provided by Crainic [8] and Wieberneit [26]. Since intra-cluster transportation is largely ignored, previous results for the SNDP that are relevant for the SCSNDP are limited; however, there are some exceptions. Medina et al. [14] present an extension to the SNDP that involves the integration of long-haul and local transportation. Building on the work of Medina et al. [14], Wolfinger et al. [28] consider a multi-modal routing problem that integrates a scheduling component for the first- and last-mile journeys. Heggen et al. [12] describe a more comprehensive model integrating long-haul and local transportation, where a set of regions are prescribed, with long-haul routes transporting commodities between regions. Pickup and delivery routes are then identified to perform the first- and last-mile transportation of commodities within each region. The proposed SCSNDP builds upon the previous work of Medina et al. [14], Wolfinger et al. [28] and Heggen et al. [12] by incorporating flexibility within the warehouse clusters, resulting in a transportation network that is adaptive to changes in commodity demands.

The PDP is a well studied variant of the VRP that shares many characteristics with the SCSNDP. For a general overview of the PDP, the reader is referred to Savelsbergh and Sol [23] and Desaulniers et al. [9]. While the classical PDP is broadly related to the SCSNDP, the variants incorporating cross-docking locations and transshipment are most relevant. Wen et al. [25] and Santos et al. [22] present examples of the PDP that incorporate a single cross-docking location. Thus, transshipment may be required at most once when transporting a commodity between the pickup and delivery locations. An effective large-neighbourhood search heuristic for the PDP with a single cross-docking opportunity was developed by Petersen and Ropke [19]. An extension of the PDP with cross-docking opportunities is presented by Buijis et al. [4], where multiple cross-docking locations exist and synchronisation constraints are introduced to incorporate transshipment between these locations.

The PDP with transshipment is an alternative model that incorporates the transfer of commodities between vehicles, as presented by Mitrović-Minić and Laporte [16] and Rais et al. [20]. Mitrović-Minić and Laporte [16] employ transshipment to restrict the geographical regions that the local distribution vehicles cover. Alternatively, Rais et al. [20] present a variant of the PDP with transshipment where it is possible for the vehicles to end at a depot that is different from the origin depot. A more comprehensive approach to the PDP with transshipment is described by Wolfinger [27]—incorporating time windows, split loads and transshipment. Incorporating transshipment in the PDP through the integration with the SNDP is a major contribution of the SCSNDP.

A common theme of the vehicle routing and location problems presented above is the synchronisation between vehicles. Drexl [10] presents an extensive survey on the use of synchronisation constraints in vehicle routing problems. Most relevant to the work presented in this paper is the operational synchronisation, where vehicle schedules arriving a consolidation locations must be synchronised to enable transshipment. The development of an iterative solution algorithm is motivated by the existence of such synchronisation constraints in the SCSNDP.

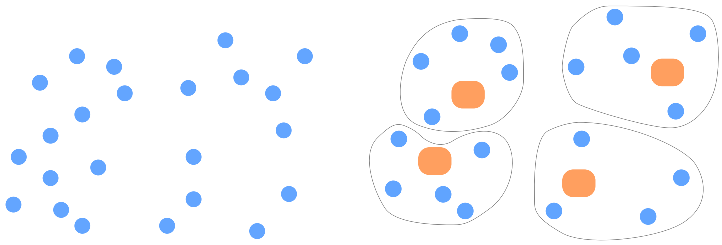
While inter-warehouse transportation is the focus of this paper, the developed methods are applicable in various different settings throughout the supply chain network. In particular, the direct transportation links between consolidation locations—the inter-cluster routes—are synonymous with long-haul routes. Also, the routes originating and terminating at a consolidation location that visit one or more warehouses within the one cluster—the intra-cluster routes—correspond to local routes. Combining intra- and inter-cluster routes has been the focus of recent work on the SNDP [12, 14, 28] and the PDP with cross-docking and transshipment [4, 16, 19, 22, 25, 27]. As such, the modelling and solution methods proposed in this paper aim to contribute to this rich field of research. A major contribution of the SCSNDP is the adaptive transportation network through the selection of consolidation locations. It is the flexibility in the consolidation locations considered in this paper that extends previous approaches integrating intra- and inter-cluster routes.

3 Mathematical modelling

The operational decisions of the SCSNDP will be modelled as three mathematical programming problems—warehouse clustering problem (WCP), SNDP and PDP. The underlying transportation network is modelled as a complete directed graph $\mathcal{G} = (\mathcal{A}, \mathcal{N})$, where $\mathcal{N}$ is the set of warehouse locations and $\mathcal{A}$ represents all direct transportation links between the warehouses. Each warehouse either supplies or requests commodities to or from another warehouse, where the set of commodities is given by $\mathcal{K}$. Commodity $k \in \mathcal{K}$ has an origin location o_k and a destination location d_k, where $o_k, d_k \in \mathcal{N}$, and a total quantity q_k that needs to be transported. Commodities are either transported completely within a cluster or must be transferred from one cluster to another via inter-cluster routes. The commodities where the origin and destination locations are in the same region are described *intra-cluster commodities* (ICC). Similarly commodities that have an origin and destination in two different regions are described as *out of cluster commodities* (OCC). A time window is specified for both the pickup and delivery of each commodity. The pickup time window is denoted by $W_k^p = [e_k^p, l_k^p]$ and the delivery time window is given by $W_k^d = [e_k^d, l_k^d]$. While it is common to associate the time windows with the warehouse locations, it is more practical for the SCSNDP to define the time windows with respect to the commodities. Finally, the arrival and departure of vehicles at consolidation locations is not restricted by commodity time windows, but the specification of business hours. The set of time windows corresponding to the business hours is denoted by $\mathcal{B}$, where business hours b is defined as $[e^b, l^b]$.

3.1 Warehouse clustering problem

The aim of the WCP is to cluster warehouses into *local* groups as measured by some *distance* function. This is achieved by selecting a set of warehouses to be designated as consolidation locations and assigning each warehouse to exactly one consolidation location. An illustrative example of warehouse clustering is given in Figure 2, where the consolidation locations are



(a) Locations (b) Clustering

Figure 2: Warehouse locations with clustering

marked in orange. Most importantly, the consolidation locations form the nodes of the inter-cluster transportation network that will be used when solving the SNDP. The clusters also define the intra-cluster transportation networks that will be used when solving the PDP.

The variables z_j are defined to equal 1 if warehouse $j \in \mathcal{N}$ is selected as a consolidation location. Each warehouse must be assigned to exactly one consolidation location. The assignment of warehouses to consolidation locations is indicated by setting the variable w_{ij} to 1 if warehouse $i \in \mathcal{N}$ is assigned to consolidation location $j \in \mathcal{N}$. The parameter α_{ij} is defined as the distance from warehouse $i \in \mathcal{N}$ to consolidation location $j \in \mathcal{N}$. Finally, the number of clusters is set by the parameter γ. Using the above notation, the WCP is given by

$$\text{minimise} \quad \sum_{(i,j)\in\mathcal{A}} \alpha_{ij} w_{ij}, \tag{1a}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{N}} z_j = \gamma, \tag{1b}$$

$$\sum_{j\in\mathcal{N}} w_{ij} = 1 \quad \forall i \in \mathcal{N}, \tag{1c}$$

$$\sum_{i\in\mathcal{N}} w_{ij} \leq |\mathcal{N}| z_j \quad \forall j \in \mathcal{N}, \tag{1d}$$

$$z_j \in \{0,1\} \; \forall j \in \mathcal{N}, \quad w_{ij} \in \{0,1\} \; \forall (i,j) \in \mathcal{A}. \tag{1e}$$

The objective is defined to minimise the sum of distances between the consolidation locations and all connected warehouse locations. It is important to note, that since the clusters define the consolidation locations that support both the intra- and inter-cluster transportation, this measure can have a big impact on the overall cost of the transportation network. The restriction on the number of clusters is enforced by constraint (1b). Constraints (1c) ensures that each warehouse is assigned to exactly one cluster. Finally, constraints (1d) ensure that warehouses are assigned to cluster j only if warehouse j is selected as a consolidation location.

3.2 Service network design problem

The solution to the WCP identifies a set of disjoint geographical regions and associated consolidation locations. The consolidation locations are denoted by $\bar{\mathcal{N}} \subseteq \mathcal{N}$. The inter-cluster transportation network $\bar{\mathcal{G}} = (\bar{\mathcal{A}}, \bar{\mathcal{N}})$ is formed as a complete directed subgraph of the transportation network, where $\bar{\mathcal{A}} = \mathcal{A}(\bar{\mathcal{N}})$ is the set of directed arcs between the consolidation locations. Building on the illustrative example from Figure 2, the inter-cluster transportation network is presented in Figure 3. Each arc $(i, j) \in \bar{\mathcal{A}}$ connects locations $i \in \bar{\mathcal{N}}$ and $j \in \bar{\mathcal{N}}$ and has an associated travel time of tt_{ij}.

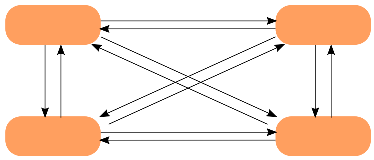


Figure 3: The inter-cluster transportation network underlying the SNDP.

Let $\bar{\mathcal{K}}$ denote the set of OCC. Every commodity $k \in \bar{\mathcal{K}}$ has an origin and destination consolidation location. Let $\Theta : \mathcal{N} \to \bar{\mathcal{N}}$ define a mapping from warehouse locations to the associated consolidation location given by the solution to the WCP. Thus, the origin and destination locations for commodities in the inter-cluster network are given by $\bar{o}_k = \Theta(o_k)$ and $\bar{d}_k = \Theta(d_k)$ respectively. Finally, $\bar{e}_k$ and $\bar{l}_k$ are the earliest delivery start time and latest delivery end time for commodity k at $\bar{o}_k$ and $\bar{d}_k$ respectively. These departure and arrival times are related to the time windows W_k^p and W_k^d, where $\bar{e}_k \geq e_k^p$ and $\bar{l}_k \leq l_k^d$.

The SNDP is modelled using a time expanded network. The planning horizon is defined by the range $[E, L]$, where E is the earliest departure of any commodity and, similarly, L is the latest arrival of any commodity. The set of time points is given by $\mathcal{T} = \{E + m\Delta | m \in \mathbb{Z}_{\geq 0}, E + m\Delta < L\}$, where $\Delta > 0$ is a time discretisation interval. The network is given by a set of nodes $\bar{\mathcal{N}}_{\mathcal{T}}$ for every time point, denoted by (i, t), where $i \in \bar{\mathcal{N}}$ and $t \in \mathcal{T}$. The arcs connecting the nodes in $\bar{\mathcal{N}}_{\mathcal{T}}$ are contained in $\bar{\mathcal{A}}_{\mathcal{T}}$ and are of the form $((i, t), (j, \bar{t}))$, where $(i, j) \in \bar{\mathcal{A}}$ and $t \in \mathcal{T}$ and $\bar{t} \in \mathcal{T}_{ijt}$. The set $\mathcal{T}_{ijt} = \{\lceil t + tt_{ij} + m\Delta \rceil_{\Delta} \,|\, m \in \mathbb{Z}_{\geq 0}, m\Delta \leq B\}$, where $\lceil \cdot \rceil_{\Delta}$ rounds the time up to the nearest discretisation interval, provides flexibility in the travel times between i and j by introducing some buffer time B. Finally, an additional set of arcs $\mathcal{H}_{\mathcal{T}}$ is defined to represent a vehicle waiting at a given node, which are of the form $((i, t), (i, t + \Delta))$, where $t \in \mathcal{T}$ and $t + \Delta \leq L$. The time expanded transportation network is denoted by $\mathcal{G}_{\mathcal{T}} = (\bar{\mathcal{N}}_{\mathcal{T}}, \mathcal{H}_{\mathcal{T}} \cup \bar{\mathcal{A}}_{\mathcal{T}})$.

The solution to the SNDP identifies the capacity of the arcs required to transport all commodities from their origin cluster to their destination cluster. The use of inter-cluster vehicles is indicated by variables $y_{ij}^{t\bar{t}}$, which equals the number of vehicles departing i at t and arriving in j at $\bar{t}$. Only a single type of vehicle is considered in the SNDP, the capacity of which is denoted by U. The flow of commodities through the inter-cluster network is given by the variables $x_{ij}^{kt\bar{t}}$, which equal 1 to indicate that delivery request k is transported along arc (i, j) and has departure time t and arrival time $\bar{t}$. It is only possible for a commodity to move between i and j if an inter-cluster vehicle is also traversing the same arc at the same time. The fixed cost of a vehicle using arc (i, j) is given by f and the cost of transporting one unit of a commodity along (i, j) is given by c_{ij}. Using the above notation, the SNDP is given by

$$\text{minimise} \quad \sum_{((i,t),(j,\bar{t}))\in\bar{\mathcal{A}}_{\mathcal{T}}} f y_{ij}^{t\bar{t}} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\bar{\mathcal{A}}_{\mathcal{T}}} c_{ij} q_k x_{ij}^{kt\bar{t}}, \tag{2a}$$

$$\text{s.t.} \quad \sum_{((i,t),(j,\bar{t}))\in\bar{\mathcal{A}}_{\mathcal{T}}\cup\mathcal{H}_{\mathcal{T}}} x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\bar{\mathcal{A}}_{\mathcal{T}}\cup\mathcal{H}_{\mathcal{T}}} x_{ji}^{k\bar{t}t} = \begin{cases} 1 & (i,t) = (\bar{o}_k, \bar{e}_k) \\ -1 & (i,t) = (\bar{d}_k, \bar{l}_k) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall k \in \mathcal{K}, (i,t) \in \bar{\mathcal{N}}_{\mathcal{T}}, \tag{2b}$$

$$\sum_{k\in\mathcal{K}} q_k x_{ij}^{kt\bar{t}} \leq U y_{ij}^{t\bar{t}} \quad \forall((i,t),(j,\bar{t})) \in \bar{\mathcal{A}}_{\mathcal{T}}, \tag{2c}$$

$$x_{ij}^{kt\bar{t}} \in \{0,1\} \quad \forall((i,t),(j,\bar{t})) \in \bar{\mathcal{A}}_{\mathcal{T}}, k \in \mathcal{K}, \tag{2d}$$

$$y_{ij}^{t\bar{t}} \in \mathbb{Z}_{\geq 0} \quad \forall((i,t),(j,\bar{t})) \in \bar{\mathcal{A}}_{\mathcal{T}}. \tag{2e}$$

The objective of the SNDP is to minimise the fixed cost of using inter-cluster vehicles and costs of transporting all delivery requests. Constraints (2b) balance the flow of commodities through the time expanded transportation network. The capacity restrictions on arc $(i, j) \in \bar{\mathcal{A}}$ are enforced by constraints (2c).

3.3 Pick-up and delivery problem

The intra-cluster transportation networks are induced by the solution to the WCP. For each cluster of warehouses $r \in \mathcal{R}$, intra-cluster transportation routes must i) transport commodities between warehouse locations within the region and ii) transport commodities between the consolidation location and connected warehouse locations. The intra-cluster transportation network in region r is denoted by $\mathcal{G}^r = (\mathcal{A}^r, \mathcal{N}^r)$, where $\mathcal{N}^r \subseteq \mathcal{N}$ are the warehouse locations within region r and $\mathcal{A}^r = \mathcal{A}(\mathcal{N}^r)$ are the transportation links connecting these warehouse locations. An example of the transportation networks that will be used for the PDP is presented in Figure 4.

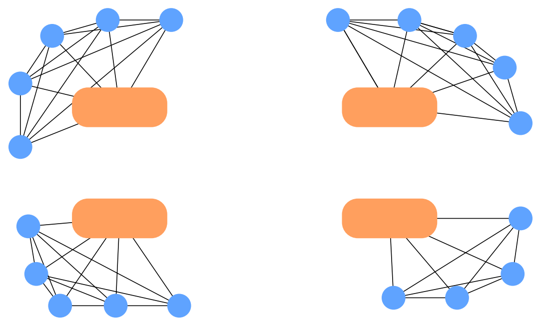


Figure 4: The disjoint intra-cluster networks used for solving the PDP

The set of commodities considered in the PDP for cluster r, denoted by $\mathcal{K}^r$, are the ICCs pertaining to r and OCCs that have either an origin or destination within $\mathcal{N}^r$. For the OCCs, inter-cluster routes will be required to transport them to or from the cluster. Thus, either the origin or destination location of the OCCs will be the consolidation location and not the initial origin or final destination of the commodity. Let $\Phi : \mathcal{R}\times\mathcal{N}\times\mathcal{N} \to \mathcal{N}\times\mathcal{N}$ define a mapping from a given cluster and initial commodity origin and destination locations to the pickup and delivery location for the PDP. Further, let $\hat{o}^r$ and $\hat{d}^r$ denote the origin and destination consolidation locations for the PDP (typically, $\hat{o}^r = \hat{d}^r$). For the PDP in region r, the origin and destination of commodity $k \in \mathcal{K}^r$ is given by:

$$\Phi(r, o_k, d_k) = \begin{cases} (o_k, d_k) & o_k \in \mathcal{N}^r, d_k \in \mathcal{N}^r, \\ (\hat{o}^r, d_k) & o_k \notin \mathcal{N}^r, d_k \in \mathcal{N}^r, \\ (o_k, \hat{d}^r) & o_k \in \mathcal{N}^r, d_k \notin \mathcal{N}^r. \end{cases} \tag{3}$$

For conciseness, the origin and destinations for commodity k in the PDP for cluster r are denoted by $\hat{o}_k^r$ and $\hat{d}_k^r$ respectively, i.e. $\Phi(r, o_k, d_k) = (\hat{o}_k^r, \hat{d}_k^r)$.

Each commodity is assigned a time window for pickup and delivery at its origin and destination, respectively. For commodities where $o_k \in \mathcal{N}^r$, the pickup time window is given by W_k^p. Similarly, for commodities where $d_k \in \mathcal{N}^r$, the delivery time window is given by W_k^d. If $\hat{o}_k^r = \hat{o}^r$ or $\hat{d}_k^r = \hat{d}^r$, then the time windows are connected to the solution of the SNDP. At the consolidation location, the time window for the OCCs are one-sided, since they must synchronise the intra- and inter-cluster vehicles. For commodity k, if $\hat{o}_k^r = \hat{o}^r$, then the pickup time window is given by $[\hat{e}^p_k, L]$, where $\hat{e}^p_k$ corresponds to the inter-cluster vehicle arrival time. If $\hat{d}^r_k = \hat{d}^r$, then the delivery time window is given by $[E, \hat{l}^d_k]$, where $\hat{l}^d_k$ corresponds to the inter-cluster vehicle departure time. In the formulation of the PDP, $[\hat{e}^p_k, \hat{l}^p_k]$ denotes the pickup time window at $\hat{o}^r_k$ and $[\hat{e}^d_k, \hat{l}^d_k]$ denotes the delivery time window at $\hat{d}^r_k$ for commodity k.

The set of all vehicles available in region r is denoted by $\mathcal{V}^r$ and the variables y^v equals 1 if vehicle v is used and 0 otherwise. The variables x^v_{ij} equals 1 if vehicle v uses arc (i, j) in a route, and 0 otherwise. Further, if vehicle v picks up commodity k, then variable h^v_k equals 1. A fixed cost of f, which is the same as in Section 3.2, and a variable cost proportional to total travel time, at a rate of $\kappa > 0$, is applied for using vehicle v. An important feature of the PDP is the tracking of the vehicle travel time and load to ensure the time windows, total travel time and the vehicle capacity are respected. The variables T^v_i and Q^v_i denote the cumulative travel time and load, respectively, of the vehicle on the route up to location i. The cumulative travel time must also incorporate the processing time to load and unload the commodities picked up and delivered at location $i \in \mathcal{N}^r$, which is denoted by g_i. The maximum travel time and capacity of vehicle v is denoted by T^v and Q^v respectively. Finally, to incorporate the warehouse business hours, the variables m^{vb}_i equal 1 if vehicle v visits location i during business hours b, and 0 otherwise.

The formulation of the PDP used in this paper is based upon the mathematical models presented by Desaulniers et al. [9] and Cordeau et al. [7], which is given by

$$\text{minimise} \quad \sum_{v \in \mathcal{V}^r} f y^v + \sum_{v \in \mathcal{V}^r} \sum_{(i,\hat{d}^r) \in \mathcal{A}^r} \kappa T^v_{\hat{d}^r} x^v_{i\hat{d}^r}, \tag{4a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}^r} x^v_{\hat{o}^r_k j} \le h^v_k \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{4b}$$

$$\sum_{v \in \mathcal{V}^r} h^v_k = 1 \quad \forall k \in \mathcal{K}^r, \tag{4c}$$

$$\sum_{j \in \mathcal{N}^r} x^v_{\hat{o}^r_k j} - \sum_{j \in \mathcal{N}^r} x^v_{j \hat{d}^r_k} = 0 \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{4d}$$

$$\sum_{j \in \mathcal{N}^r} x^v_{ij} - \sum_{j \in \mathcal{N}^r} x^v_{ji} = \begin{cases} y^v & i = \hat{o}^r \\ -y^v & i = \hat{d}^r \\ 0 & \text{otherwise} \end{cases}$$

$$\forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r, \tag{4e}$$

$$T^v_j \ge (T^v_i + g_i + tt_{ij}) x^v_{ij} \quad \forall (i, j) \in \mathcal{A}^r, \forall v \in \mathcal{V}^r, \tag{4f}$$

$$Q_j^v \geq Q_i^v + \sum_{\substack{k \in \mathcal{P}^r \\ \hat{o}_k^r = i}} q_k h_k^v - \sum_{\substack{k \in \mathcal{P}^r \\ \hat{d}_k^r = i}} q_k h_k^v \quad \forall (i,j) \in \mathcal{A}^r, \forall v \in \mathcal{V}^r, \tag{4g}$$

$$T_{\hat{o}_k^r}^v + g_{\hat{o}_k^r} + tt_{\hat{o}_k^r \hat{d}_k^r} \leq T_{\hat{d}_k^r}^v \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{4h}$$

$$\hat{e}_k^p \leq T_{\hat{o}_k^r}^v \leq \hat{l}_k^p \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{4i}$$

$$\hat{e}_k^d \leq T_{\hat{d}_k^r}^v \leq \hat{l}_k^d \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{4j}$$

$$\max\{0, \hat{q}_i\} \leq Q_i^v \leq \min\{Q^v, Q^v + \hat{q}_i\} \quad \forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r, \tag{4k}$$

$$T_i^v \leq T^v y^v \quad \forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r, \tag{4l}$$

$$x_{ij}^v \leq \sum_{b \in \mathcal{B}} m_i^{vb} \leq 1, \quad \forall (i,j) \in \mathcal{A}^r, \forall v \in \mathcal{V}^r, \tag{4m}$$

$$e^b m_i^{vb} \leq T_i^v \leq l^b + M(1 - m_i^{vb}) \quad \forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r, \forall b \in \mathcal{B}, \tag{4n}$$

$$x_{ij}^v \in \{0,1\} \quad \forall (i,j) \in \mathcal{A}^r, \forall v \in \mathcal{V}^r, \tag{4o}$$

$$y^v \in \{0,1\} \quad \forall v \in \mathcal{V}^r, \tag{4p}$$

$$m_i^{vb} \in \{0,1\} \quad \forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r, \forall b \in \mathcal{B}, \tag{4q}$$

$$T_i^v \geq 0,\ Q_i^v \geq 0 \quad \forall i \in \mathcal{N}^r, \forall v \in \mathcal{V}^r. \tag{4r}$$

This objective is the linear combination of a fixed cost for each vehicle used and a cost proportional to the travel time. The pickup of the commodity must be performed by a single vehicle, which is enforced by constraints (4b) and (4c). Further, the subsequent delivery of all commodities is enforced by constraints (4d). The flow balance of the vehicles at each node is given by the constraints (4e). Constraints (4f) and (4g) are bookkeeping constraints for the vehicle travel time and capacity respectively. Each commodity must be picked up prior to delivery, which is enforced by constraints (4h). Constraints (4i) and (4j) enforce the pickup and delivery windows for each commodity. Each vehicle has a maximum capacity and travel time, which is enforced by constraints (4k) and (4l). Finally, constraints (4m) states that a vehicle can only visit a warehouse location once and (4n) ensures that this visit occurs during the business hours.

3.4 Direct deliveries by third-party vehicles

The clustering of warehouses and selection of consolidation locations may impact the ability to satisfy all pickup and delivery requests using the synchronised intra- and inter-cluster transportation routes. There are four conditions that lead to an infeasible pickup or delivery request.

If any of these conditions are satisfied, then a third-party vehicle is required to deliver the commodity. Note that since the commodity transportation could span an overnight period, business hours must be considered for each condition. The four direct delivery conditions are:

1. Unsatisfied pickup time window: The latest arrival time for a commodity at the pickup location is exceeded by the minimum travel time from the consolidation location to the pickup location. For a given cluster $r \in \mathcal{R}$, the pickup of commodity $k \in \mathcal{K}^r$ is unsatisfied if $\hat{l}_k^p < E + tt_{\hat{o}^r \hat{o}_k^r}$.

2. Unsatisfied delivery time window with pickup and delivery in the same region: The latest arrival time for a commodity at the delivery location is exceeded by the minimum travel time from the consolidation location to the delivery location via the pickup location. For a given cluster $r \in \mathcal{R}$, the delivery of commodity $k \in \mathcal{K}^r$ is unsatisfied if $\hat{l}_k^d < E + tt_{\hat{o}^r \hat{o}_k^r} + tt_{\hat{o}_k^r \hat{d}_k^r}$.

3. Unsatisfied delivery time window with pickup and delivery in different regions: The latest arrival time for a commodity at the delivery location is exceeded by the sum of the minimum travel times from the pickup location to a consolidation location, between consolidation locations and from a consolidation location to the delivery location. Let $r_p, r_d \in \mathcal{R}$ denote the pickup and delivery regions respectively, $r_p \neq r_d$. The delivery of commodity $k \in \mathcal{K}$ is unsatisfied if $\hat{l}_k^d < \lceil E + tt_{\hat{o}^{r_p} \hat{o}_k^{r_p}} + tt_{\hat{o}_k^{r_p} \hat{d}^{r_p}} + tt_{\hat{d}^{r_p} \hat{o}^{r_d}} \rceil_{\mathcal{B}} + tt_{\hat{o}^{r_d} \hat{d}_k^{r_d}}$, where $\lceil t \rceil_{\mathcal{B}}$ rounds time t up to nearest business hours time window.

4. Exceeding the operation time of an inter-cluster vehicle: Given the latest pickup time of a commodity, its earliest arrival time at the destination consolidation location, with respect to business hours, requires an inter-cluster vehicle to exceed the maximum operation time. The delivery of commodity $k \in \mathcal{K}$ is unsatisfied if $\hat{l}_k^p + tt_{\hat{o}_k^{r_p} \hat{d}^{r_p}} + T < \lceil \hat{l}_k^p + tt_{\hat{o}_k^{r_p} \hat{d}^{r_p}} + tt_{\hat{d}^{r_p} \hat{o}^{r_d}} \rceil_{\mathcal{B}}$.

4 Solution algorithm

A MAB problem [3] has been formulated to integrate warehouse clustering with the intra- and inter-cluster transportation problems. An overview of the solution algorithm, independent of selection strategies for the MAB problem, is presented in Figure 5. The bandit actions are defined by different selections of the total number of clusters γ when solving the WCP. For a given value of γ, the WCP is solved to identify a clustering of the warehouse locations and

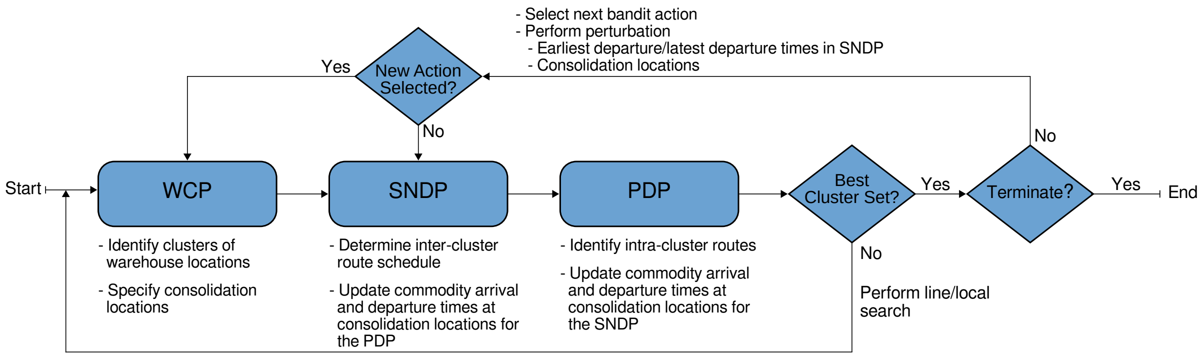


Figure 5: The SCSNDP solution algorithm

specify the consolidation locations. The algorithm then iterates between the SNDP and the PDP to identify and synchronise feasible inter- and intra-cluster transportation routes. Iteration between the SNDP and PDP is essential for finding feasible solutions to the SCSNDP, since the synchronisation of the inter- and intra-cluster transportation is not guaranteed from the solutions of the individual problems. Perturbations to the consolidation locations and vehicle arrival/departure times are employed to diversify the search for improving solutions.

The solution algorithm for the SCSNDP comprises of two major phases. An initialisation phase is employed to find a subset of values for γ that define the actions for the MAB algorithm. The initialisation phase consists of a line search combined with a local search, described in Section 4.1, that solves the SCSNDP for different values of γ. The values of γ with the best initial feasible solution are selected as the action set, denoted by Γ, for the MAB algorithm. The main phase executes the MAB algorithm by selecting γ from Γ and solving the SCSNDP. A detailed overview of the MAB problem and selection strategy will be provided in Section 4.2. During both phases the SNDP and PDP are iteratively solved to identify a feasible synchronisation of the intra- and inter-cluster routes. Section 4.3 details the transformation of the consolidation location time windows into *soft* constraints to allow for flexibility in the SCSNDP solution algorithm. In each iteration, the arrival and departure times of commodities at the consolidation locations are updated to repair the violation of these soft constraints. This time window update procedure is described in Section 4.4. To diversify the search of the feasible region, the perturbation techniques presented in Section 4.5 are employed. Finally, the solution methods for the WCP, SNDP and PDP will be described in Section 4.6.

4.1 Initialisation phase

The first stage of the initialisation phase executes a line search algorithm that samples selections for γ to identify promising regions for the local search. Initially, the feasible choices of cluster limits in the WCP is given by $\{2, 3, \ldots, |\mathcal{N}|/2\}$. Starting with $\gamma = 2$, the line search solves the WCP, and then the SNDP and PDP are iteratively solved until a feasible solution to the SCSNDP is found. The value of γ is then increased by φ, which is set to $|\mathcal{N}|/10$ in our experiments, and the process to find a feasible solution for the SCSNDP is repeated. The line search terminates when a local minimum is found from the evaluated selections of γ, which is triggered by an increase in the objective value for two consecutive selections of γ.

The second stage of the initialisation phase executes an adaptive local search around the best cluster limit identified during the line search, denoted by γ^*. Similar to the line search, the local search finds feasible solutions to the SCSNDP by solving the WCP, SNDP and PDP. The search neighbourhood is given by $\gamma \in [\gamma^* - \lfloor\varphi/2\rfloor, \gamma^* + \lfloor\varphi/2\rfloor]$, where $\lfloor\cdot\rfloor$ corresponds to rounding down. If the best objective value is improved, γ^* and the neighbourhood for γ are updated. Feasible solutions must be found for all γ in the updated neighbourhood. The local search continues until no improvement in the best objective value is found for γ in the defined neighbourhood.

Following the initialisation phase, the three values of γ that resulted in the best objective values for the SCSNDP are selected as the set of actions for the MAB algorithm, denoted by Γ.

4.2 Multi-armed bandit problem

A MAB problem is defined as the sequential selection of actions, from a predefined set of actions, that maximise the total reward for a given payoff function. In a given round t, a player selects an action γ_t from the set of possible actions Γ. The reward from selecting γ in round t is denoted by $r_{\gamma_t t}$, where $r_{\gamma t} \in [0, 1]$. The objective of an MAB problem is to maximise the total payoff from the actions selected across all rounds T, i.e. $\max \sum_{t \in T} r_{\gamma_t t}$.

In the context of the SCSNDP, the predefined set of actions is given by a set of cluster limits found during the initialisation phase. The reward from selecting γ is given by $r_{\gamma_t t} = Z/Z^t$, which is the ratio between the current best objective value and objective value of the best solution for the SCSNDP found during round t. Since the iterative algorithm integrating the SNDP and PDP employs perturbation techniques, solving the SCSNDP for a given γ is a random process.

Thus, the MAB problem for the SCSNDP is a stochastic bandit problem.

Selection strategies for MAB problems aim to balance the exploitation of promising actions and the exploration of alternative actions. In this paper, the selection strategy is based on upper confidence bounds (UCB) [1]. A UCB is computed as the summation of the expected reward and the confidence interval. The expected reward, denoted by $\bar{r}_{\gamma t}$, is the arithmetic mean of the rewards from all rounds up to t where action γ was selected. The confidence interval is given by

$$CI = \sqrt{\alpha \frac{\ln(t+1)}{T_\gamma + 1}}, \tag{5}$$

where α is the width of the confidence interval and T_γ is the number of rounds where action γ was selected. In round t, the next action selected is the one with the largest UCB, given by

$$\gamma_{t+1} := \text{argmax}_{\gamma \in \Gamma}\{\bar{r}_{\gamma t} + CI\}. \tag{6}$$

The termination of a round in the MAB algorithm is based upon the number of iterations since the last improvement in the objective value when using action γ. A single round of the MAB algorithm will involve one or more iterations solving the SNDP and PDP. Let i'_γ denote a *no improvement*-counter and Z_γ as the best objective found for action γ. At the beginning of each round using action γ, $i'_\gamma = 0$ and is incremented each iteration that the objective value Z^t is greater than Z_γ. If an improving solution for action γ is found, then $Z_\gamma = Z^t$ and $i'_\gamma = 0$. The round terminates when i'_γ exceeds a no improvement limit given by R.

A number of perturbation techniques are described in Section 4.5 that impact the execution of SNDP and PDP iterations, and, consequently, the termination of a round in the MAB algorithm. Specifically, the perturbation techniques described in Section 4.5.2 may select consolidation locations that produce inferior solutions and thus triggering the early termination of a round. As a result, an additional condition is imposed in the MAB algorithm whereby a round can only end when the consolidation locations are set to those that achieved an objective value of Z_γ. The handling of this additional condition will be described in Section 4.5.2.

4.3 Connection between the SNDP and PDP

The primary link between the SNDP and the PDP is the commodity departure and arrival times at the consolidation locations. It is through these times that penalties will be imposed within the iterative algorithm.

4.3.1 Adding penalties to the SNDP

The addition of penalty terms in the SNDP requires a modification to constraints (2b). First, the variables $\bar{z}_t^{ek}$ are defined to equal 1 if commodity k departs from origin $\bar{o}_k$ at time t. Similarly, the variables $\bar{z}_t^{lk}$ equal 1 if commodity k arrives at destination $\bar{d}_k$ at time t. These variables are used to identify whether a penalty must be applied for the departure or arrival of commodity k. To impose the penalty term, constraints (2b) are replaced with

$$\sum_{((i,t),(j,\bar{t}))\in\bar{\mathcal{A}}_{\mathcal{T}}\cup\mathcal{H}_{\mathcal{T}}} x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\bar{\mathcal{A}}_{\mathcal{T}}\cup\mathcal{H}_{\mathcal{T}}} x_{ji}^{k\bar{t}t} = \begin{cases} \bar{z}_t^{ek} & i=\bar{o}_k, t\in\{t|t\in\mathcal{T}, t\le\bar{e}_k\} \\ -\bar{z}_t^{lk} & i=\bar{d}_k, t\in\{t|t\in\mathcal{T}, t\ge\bar{l}_k\} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall k\in\mathcal{K}, (i,t)\in\bar{\mathcal{N}}_{\mathcal{T}}, \tag{7a}$$

$$\sum_{t\in\{t|t\in\mathcal{T},t\le\bar{e}_k\}} \bar{z}_t^{ek} = 1 \quad \forall k\in\mathcal{K}, \tag{7b}$$

$$\sum_{t\in\{t|t\in\mathcal{T},t\ge\bar{l}_k\}} \bar{z}_t^{lk} = 1 \quad \forall k\in\mathcal{K}. \tag{7c}$$

$$\bar{z}_t^{ek}\in\{0,1\} \quad \forall t\in\{t|t\in\mathcal{T}, t\le\bar{e}_k\}, k\in\mathcal{K}, \tag{7d}$$

$$\bar{z}_t^{lk}\in\{0,1\} \quad \forall t\in\{t|t\in\mathcal{T}, t\ge\bar{l}_k\}, k\in\mathcal{K}. \tag{7e}$$

Constraints (7a) are the modification of constraints (2b). The addition of the $\bar{z}_t^{ek}$ and the $\bar{z}_t^{lk}$ variables permit the inter-cluster routes to depart before and arrive after the earliest start and latest arrival times respectively. Constraints (7b) and (7c) ensure that exactly one departure and arrival time is selected per commodity, respectively.

In order to penalise early (late) departures (arrivals), objective function (2a) is appended with the terms

$$\sum_{k\in\mathcal{K}} \sum_{t\in\{t|t\in\mathcal{T},t\le\bar{e}_k\}} \phi_t^k \bar{z}_t^{ek} + \sum_{k\in\mathcal{K}} \sum_{t\in\{t|t\in\mathcal{T},t\ge\bar{l}_k\}} \theta_t^k \bar{z}_t^{lk}. \tag{8}$$

The penalty for early departure is defined as $\phi_t^k = Y(\bar{e}_k - t)^2$ and the penalty for late arrival is defined as $\theta_t^k = Y(t-\bar{l}_k)^2$, where Y is a constant penalty factor. However, the penalty coefficients can be defined in any way that drives the iterative algorithm towards a feasible solution.

4.3.2 Adding penalties to the PDP

Similar to the SNDP, penalty terms are included in the PDP to allow flexibility in the departure and arrival of intra-cluster routes at the warehouse and consolidation locations. The variables $\hat{z}_k^{ep}$ and $\hat{z}_k^{lp}$ are defined to equal the amount of time that the start or end of the pickup time window for commodity k is violated, respectively. Similarly, the violation at the start or end of the delivery time window for commodity k is given by the variables $\hat{z}_k^{ed}$ and $\hat{z}_k^{ld}$ respectively. The inclusion of these variables transforms all time windows from hard to soft constraints, which are given by replacing constraints (4i)–(4j) with

$$\hat{e}_k^p - \hat{z}_k^{ep} \leq T^v_{\hat{o}_k^r} \leq \hat{l}_k^p + \hat{z}_k^{lp} \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{9a}$$

$$\hat{e}_k^d - \hat{z}_k^{ed} \leq T^v_{\hat{d}_k^r} \leq \hat{l}_k^d + \hat{z}_k^{ld} \quad \forall k \in \mathcal{K}^r, \forall v \in \mathcal{V}^r, \tag{9b}$$

$$\hat{z}_k^{ep}, \hat{z}_k^{lp}, \hat{z}_k^{ed}, \hat{z}_k^{ld} \geq 0 \quad \forall k \in \mathcal{K}^r. \tag{9c}$$

The violation of the time windows is imposed by appending the terms

$$\sum_{k \in \mathcal{K}} \hat{Y}_k^p \{(\hat{z}_k^{ep})^2 + (\hat{z}_k^{lp})^2\} + \hat{Y}_k^d \{(\hat{z}_k^{ed})^2 + (\hat{z}_k^{ld})^2\} \tag{10}$$

to the objective function (4a). While the squared terms in (10) transform the PDP into a quadratic program, the solution algorithms developed for the PDP effectively handle this modelling feature.

There are two types of time windows of the PDP, those specified at the warehouse locations, i.e. the pickup and delivery windows given by W_k^p and W_k^d, and those at the consolidation locations that are determined from the solution to the SNDP. Since the former time windows are structural constraints of the PDP, these must be satisfied with a higher priority than the latter synchronisation time windows. As such, for commodity k the value of $\hat{Y}_k^p$ is set to $9Y$ if $\hat{o}_k^r$ is the consolidation location and $10Y$ otherwise. Similarly, the value of $\hat{Y}_k^d$ is set to $9Y$ if $\hat{d}_k^r$ for commodity k is the consolidation location and $10Y$ otherwise. This setting for the penalty factor drives feasibility in the intra-cluster transportation while enabling adaptation in the synchronisation with the inter-cluster transportation schedule.

4.4 Updating pickup and delivery time windows

The solution to the SNDP identifies a set of inter-cluster routes transporting commodities between consolidation locations whose departure and arrival times are used to define the delivery and pickup time windows for the OCCs. Specifically,

- if the inter-cluster vehicle transporting commodity k departs $\bar{o}_k$ at t', then for the PDP the delivery time window at $\hat{d}^r_k$ is set to $[E, t']$ (note $\bar{o}_k = \hat{d}^r_k$),
- if the inter-cluster vehicle transporting commodity k arrives at $\bar{d}_k$ at t'', then for the PDP the pickup time window at $\hat{o}^r_k$ is set to $[t'', L]$ (note $\bar{d}_k = \hat{o}^r_k$).

Given a solution to the PDP for region r, the earliest departure and latest arrival times for the OCCs that must be transported from and to region r are updated as follows,

- if the intra-cluster vehicle transporting commodity k, which has been transferred from another region, departs from $\hat{o}^r_k$ at t', then in the SNDP the latest arrival time at $\bar{d}_k$ is set to $\bar{l}_k = \lfloor t' \rfloor_\Delta$ (note $\hat{o}^r_k = \bar{d}_k$),
- if the intra-cluster vehicle transporting commodity k, which will be transferred to another region, arrives at $\hat{d}^r_k$ at t'', then in the SNDP the earliest departure time at $\bar{o}_k$ is set to $\bar{e}_k = \lceil t'' \rceil_\Delta$ (note $\hat{d}^r_k = \bar{o}_k$).

The functions $\lceil \cdot \rceil_\Delta$ and $\lfloor \cdot \rfloor_\Delta$ round the time up and down, respectively, to the nearest discretisation interval based on Δ. Note that the earliest departure and latest arrival times are updated regardless of whether there is a time window violation.

4.5 Perturbation techniques

The primary goal of the iterative algorithm is to identify feasible solutions to the SCSNDP. However, it is also desired to find high quality solutions to reduce transportation costs. As such, perturbation techniques are employed to escape local optimal solutions and search other neighbourhoods for improving feasible solutions.

4.5.1 Shifting earliest departure and latest arrival times

The earliest departure and latest arrival times are shifted when the iterative algorithm appears to have stalled. We define algorithm stalling as follows:

Definition 1. *Let Z_q be the objective value of the SCSNDP in iteration q of the iterative algorithm and $\mathcal{Q} := \{Z_{q-1}, Z_{q-2}, \ldots, Z_{q-Q}\}$ be the set of objective values for the SCSNDP for the preceding Q iterations. In iteration q the algorithm has stalled if*

$$Z_q \in \mathcal{Q} \quad \textit{or} \quad Z_q > \max \mathcal{Q}.$$

Once the algorithm has stalled, the departure and arrival times of the OCCs at the consolidation location are perturbed following the next solve of the PDP. Algorithm 1 is employed to identify which departure and arrival times will be perturbed and the magnitude of this perturbation. The earliest departure and latest arrival times of the SNDP are then updated according to the perturbed and unperturbed times, as explained in Section 4.4. In the implementation of the iterative algorithm Q is set to 10.

Algorithm 1: Perturbing the departure and arrival times

Data: Let $\mathcal{T}_D^r$ and $\mathcal{T}_A^r$ be the set of OCC departure and arrival times at consolidation location r.

Result: The sets $\mathcal{T}_D^r$ and $\mathcal{T}_A^r$ where are subset of the times have been perturbed.

Select $\rho\%$ of the times from each of $\mathcal{T}_D^r$ and $\mathcal{T}_A^r$ uniformly at random.

Denote these selected times as $\bar{\mathcal{T}}_D^r$ and $\bar{\mathcal{T}}_A^r$ respectively.

for $t \in \bar{\mathcal{T}}_D^r \cup \bar{\mathcal{T}}_A^r$ **do**

    Sample t' from a normal distribution with a mean of 0 and standard deviation of Δ.

    Set $t \leftarrow t + t'$

Update $\mathcal{T}_D^r$ and $\mathcal{T}_A^r$ with the perturbed times in $\bar{\mathcal{T}}_D^r$ and $\bar{\mathcal{T}}_A^r$.

4.5.2 Changing the consolidation locations

The designation of warehouses as consolidation locations has a major impact on the intra- and inter-cluster transportation costs. In regards to the inter-cluster transportation schedule, an alternative set of consolidation locations could result in a transportation network $\bar{\mathcal{G}}$ with smaller transportation distances. However, there is a trade-off with respect to the distances in the intra-cluster transportation networks and the number of direct deliveries that are required. Perturbing the consolidation locations during the iterative algorithm enables the exploration of different transportation network configurations.

The mechanisms for perturbing the consolidation locations are presented in Algorithm 2. The behaviour of this perturbation scheme is connected with the selection of actions in the MAB algorithm. It can be seen on line 4, that the next action for the MAB algorithm is selected before any perturbation is performed. As described in Section 4.2, i' is used to represent a *no improvement*-counter (the γ subscript has been dropped for notational convenience). Additionally, this perturbation scheme depends on the algorithm iteration when the best solution was found, denoted by i^*. The perturbation of the consolidation locations is triggered if R iterations of the iterative algorithm are performed without an objective value improvement, i.e.

$i' > R$. However, this perturbation only occurs if γ does not change as a result of the MAB action selection process.

Since it is not guaranteed that a change in the consolidation locations will lead to an improved solution, mechanisms are included to restore the consolidation locations. If a perturbation has been performed, then at least R iterations are executed to find improving solutions. The consolidation locations are restored to those corresponding to the best solution if both $i - i^* > R$ and $i' > R$, where i is the current iteration. If the consolidation locations are restored, then the *no improvement*-counter is decreased by a factor of 2.

Algorithm 2: Changing consolidation locations

Data: The iteration where the best solution was found i^*, no improvement counter i'

Result: Perturbation or restoration of consolidation locations, update the no improvement counter i'

Set *bestCentres* $\leftarrow$ `TRUE` if the current consolidation locations are the same as those in iteration i^*, and `FALSE` otherwise
if $i' > R$ **then**
    if *bestCentres is* `TRUE` **then**
        Select next bandit action
        if *No new action selected, i.e.* $\gamma_t = \gamma_{t-1}$ **then**
            Randomly select a cluster r with probability $|\mathcal{N}^r|/|\mathcal{N}|, \forall r \in \mathcal{R}$
            Randomly select a consolidation location for cluster r from the set $\mathcal{N}^r$
        Set $i' \leftarrow 0$
    else if *R iterations have been performed since i^** **then**
        Restore the consolidation locations to those in iteration i^*
        Set $i' \leftarrow i'/2$

4.6 Solving the mathematical programs

A major feature of the proposed solution algorithm is the use of MIP solvers to make optimal decisions within the heuristic algorithms. Specifically, the WCP and SNDP are both solved directly as MIPs. Alternatively, an insertion heuristic has been used to solve the PDP, which is presented in Algorithm 3 with the key details explained below.

Insertion heuristic for the PDP The delivery route of vehicle v represented by x^v_{ij} are determined using an insertion heuristic and the travel time T^v_i of the vehicle v on the route up to the location i are determined using an exact algorithm. The insertion heuristic uses

Algorithm 3: Insertion heuristic for the PDP

Data: Evaluation value of the first commodity selection g^k, evaluation value of the additional commodity selection g^{kv}, set of unassigned commodities K', set of insertable commodities IK

Result: Delivery route x_{ij}^v and travel times of the vehicle on the route up to the location T_i^v

for $k \in \mathcal{K}$ **do**
  Calculate g^k
  Add k to K'
while K' *is not empty* **do**
  Select commodity k that has the minimum g^k
  Set v $\leftarrow$ empty vehicle
  Add k to IK
  while IK *is not empty* **do**
    Select commodity k and delivery routes that has the minimum g^{kv}
    Remove k from K'
    Determine T_i^v using a MIP solver
    Set $IK \leftarrow$ *empty*
    for $k \in K'$ **do**
      Calculate g^{kv} and delivery routes $x_{io_k}^v$, $x_{o_kj}^v$, $x_{id_k}^v$ and $x_{d_kj}^v$
      if k *is insertable* **then**
        Add k to IK

two evaluation values to select an inserted commodity. The first value is used to select the commodity to add to an empty vehicle. It is calculated for each commodity k and is represented by g^k. The value of g^k is computed as the sum of the earliest delivery time and the length of delivery time window. The value g^{kv} is used to select the commodity k to add to vehicle v, to which one or more commodities are assigned.

Let i_q be the qth location of the current route, q_p and q_d be the inserting position of pickup and delivery locations, respectively. To determine whether the pickup and delivery of commodity v should be inserted into the route of vehicle k, g^{kv} is given by

$$g^{kv} = \min_{k \in \mathcal{K}^r, v \in \mathcal{V}^r} \left\{ g_{q_p q_d}^{kv} \right\}. \tag{11}$$

The value of $g_{q_p q_d}^{kv}$ is the sum of travel time increase, arrival delay, and violation time, which is computed using the following system of equations

$$g_{q_p q_d}^{kv} = \alpha_1 gt_{q_p q_d}^{kv} + \alpha_2 gd_{q_p q_d}^{kv} + \hat{Y}_k^p \{ (\hat{z}_k^{ep})^2 + (\hat{z}_k^{lp})^2 \} + \hat{Y}_k^d \{ (\hat{z}_k^{ed})^2 + (\hat{z}_k^{ld})^2 \}, \tag{12a}$$

$$gt^{kv}_{q_p q_d} = tt_{i_{q_p-1} o_k} + tt_{o_k i_{q_p}} - \mu tt_{i_{q_p-1} i_{q_p}} + tt_{i_{q_d-1} d_k} + tt_{d_k i_{q_d}} - \mu tt_{i_{q_d-1} i_{q_d}}, \tag{12b}$$

$$gd^{kv}_{q_p q_d} = \hat{T}^v_{i_{q_p}} - T^v_{i_{q_p}} + \hat{T}^v_{i_{q_d}} - T^v_{i_{q_d}}. \tag{12c}$$

Let $\hat{T}^v_i$ be the travel time of the vehicle v on the route up to the location i. Since, the aim of the insertion heuristic is to find low cost routes with respect to travel time and time window violations, the weights α_1 and α_2 are set to penalise any increase in the travel time and arrival delay respectively. Finally, μ is the weight for total travel time of the current route.

4.7 Termination conditions

Since there is no guarantee of convergence of the proposed algorithm, a run time limit is required to ensure that solutions are delivered within a reasonable time frame. Alternatively, an early termination could be triggered if the solution is not expected to improve further. This point is identified by counting the number of times the best objective value has been encountered during the search. If the best objective value is encountered a specified number of times, then the algorithm terminates.

5 Computational experiments

The computational experiments evaluate the benefit from the integration of location clustering with transshipment and vehicle routing. First, the improvement in the objective value observed as a result of the integration will be analysed. Second, the run time of the complete algorithm and individual components will be investigated, comparing that to the overall improvement in the best found feasible solutions. Additionally, the behaviour of the MAB algorithm will be analysed with respect to the improvement in the objective value. Finally, the practical features of the results, such as the distribution of costs and the vehicles used, will be discussed.

The solution algorithm proposed in the paper has been implemented in C++. All MIPs are solved using SCIP 7.0.2, with SoPlex 5.0.2 as LP solver. The computational experiments have been performed on a computational cluster comprised of Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz CPUs and 125GB RAM per node.

5.1 Problem instances

The problem instances have been generated based upon the business practices of our industry partner. In current practice, the industry partner guarantees that all pickup and delivery requests are completed within 1-2 business days. As such, the planning horizon considered in the problem instances has a time horizon of 2 days. The key features and parameters used to generate the problem instances are displayed in Table 1.

An important characteristic of the instances is the warehouse locations. Two different methods are used to generate the warehouse locations. The first is to randomly select N points within a square using a uniform distribution. The second defines C subregions within the square, and for each warehouse a location is selected at random within one of these subregions. This second method for generating the warehouse locations is to model the real-world settings

Table 1: Parameters for generating instances

Parameter Type	**Description**
Planning horizon	2 days.
Business hours	6:00 until 20:00 each day.
Warehouse locations	Either randomly within a square or within subregions of a square. The total number of warehouses used in the experiments: $N \in \{25, 50, 100\}$. The number of subregions is given by $C \in \{1, 2, 5, 10\}$, where $C = 1$ is an instance with no subregions.
Commodities	The total number of commodities is given by $K \in \{N, 2N, 4N\}$, where $K \leq 200$. The load q_k of each commodity is selected uniformly at random from the set $\{1, 2, 3\}$. The origin and destination is selected uniformly at random, with $0.25/N$ origin-destination pairs are selected.
Pickup time window	W_k^p: e_k^p selected uniformly at random between 6:00 and 22:00. $l_k^p = e_k^p + \alpha$, where α is selected uniformly at random between 2 and 18 hours, $\forall k \in \mathcal{K}$.
Delivery time window	W_k^d: e_k^d selected uniformly at random between 6:00 and 44:00 (could deliver the next day). $l_k^d = e_k^d + \alpha$, where α is selected uniformly at random between 2 and 18 hours. Also, $l_k^d \geq l_k^p +$ travel time between pick-up and delivery ($tt_{\hat{o}_k^r \hat{d}_k^r}$) + load time ($g_{\hat{o}_k^r}$) + unloading time ($g_{\hat{d}_k^r}$), $\forall k \in \mathcal{K}$. *Note: the time windows are generated to be within the business hours.*
Travel Distance	Haversine distance.
Travel Time	tt_{ij}: Selected uniformly at random in the range $[s\beta - 0.3s\beta, s\beta + 0.3s\beta]$, where β is the travel distance and s is the travel speed of 60km/h, $\forall (i, j) \in \mathcal{A}$.
Vehicle Data	
Loading time	$g_i = 30$ minutes, $\forall i \in \mathcal{N}$.
Unloading time	$g_i = 10$ minutes, $\forall i \in \mathcal{N}$.
Fixed costs	$f = 6000$.
Capacity	$U = Q^v = 100$ units, $\forall v \in \mathcal{V}^r$.
Operation time	$T^v = 10$ hours, $\forall v \in \mathcal{V}^r$.

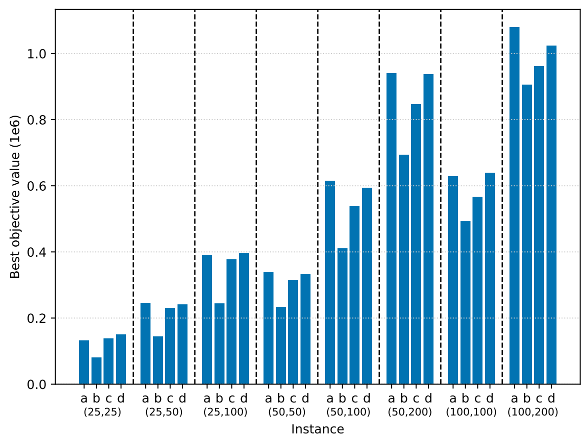
where multiple warehouses could be located within each province, state or prefecture.

The instances are identified by the tuple (N, K, C) where N is the number of warehouses, K is the number of commodities and C is the number of subregions ($C = 1$ represents an unclustered instance). With respect to the formulations of the WCP, SNDP and PDP $N = |\mathcal{N}|$ and $K = |\mathcal{K}|$. For each combination of (N, K, C) from the parameters given in Table 1, 5 different instances are generated. This is to provide a diverse test set to evaluate the proposed solution algorithm. This is particularly important since differences in warehouse locations or time windows can have a significant impact on the intra- and inter-cluster transportation costs. A total of 160 instances have been generated for the computational experiments.

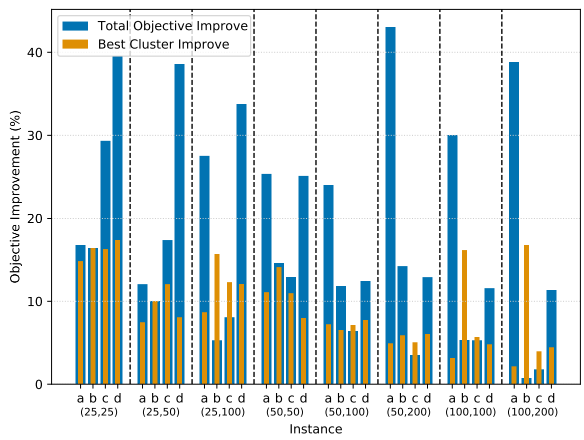
A maximum run time for the algorithm is set to 3600 seconds and the best solution limit is set to 20. In each of the presented figures, the reported values are the arithmetic mean over the 5 random instances for each (N, K, C) combination. To improve the readability of the figures, results are grouped by the tuple (N, K) and the values of C have been replace with the mapping: $a : 1, b : 2, c : 5$ and, $d : 10$. In all cases, times are reported in seconds.

5.2 Evaluating benefits from integration

The selection algorithm of the MAB problem is designed to facilitate the search of clustering solutions to best balance the intra- and inter-cluster transportation costs. Figure 6 presents the objective value of the best found solution for the SCSNDP and the improvement achieved by the MAB algorithm with respect to the first feasible solution. As expected, the increase in the number of warehouse locations and commodities leads to an increase in the overall



(a) Best objective value



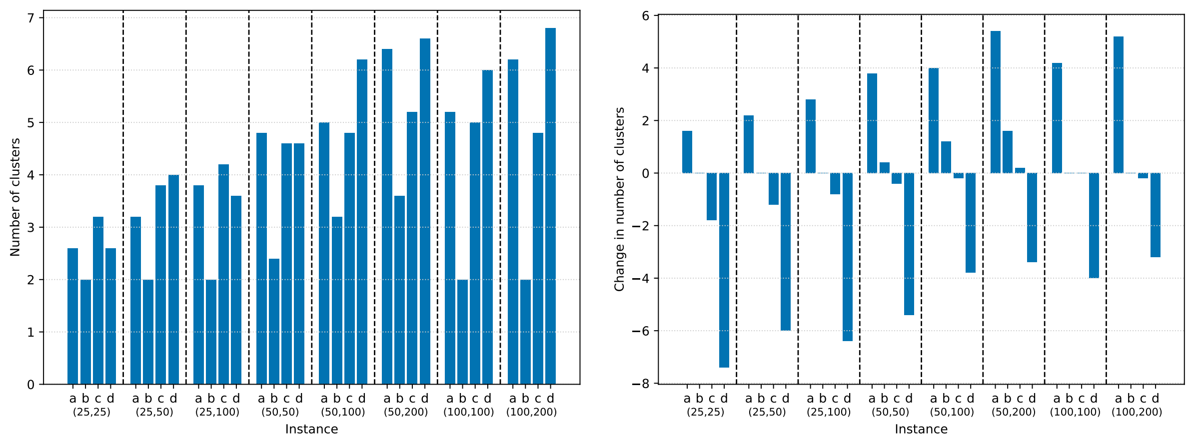
(b) Objective improvement

Figure 6: The best objective value and the improvement observed during the algorithm.

transportation cost. Interestingly, it can be seen in Figure 6(a) that the instances where the locations were clustered into 2 subregions (the columns labelled b) have the lowest costs in each of the instance groups. This is likely due to the fact that the number of clusters in the best solution is the same as the prescribed number of regions. Thus, less effort is required to search for improving numbers of clusters compared to instances with a prescribed number of regions of 1, 5 and 10.

The benefit of integrating clustering, transshipment and routing is highlighted by the results presented in Figure 6(b). The bars represent the percentage improvement in the objective, where blue and orange are with respect to the first feasible solution where $\gamma = C$, i.e. the prescribed number of regions, and $\gamma = \gamma^*$, i.e. the best number of clusters, respectively. The first feasible solutions are the equivalent to using a sequential approach to solve the integrated problem: The WCP is solved, followed by the SNDP and then finally the PDP.

The results presented in Figure 6(b) demonstrate that the integrated problem significantly improves upon the sequential algorithm. When considering the total objective improvement—the blue bars—the improvement in the objective value exceeds 30% for 5 instance sets, with greatest average improvement of 43.02%. For all instances with 25 sites, the objective improvement is greater than 10%, ranging between 10.02% and 39.56%. As the number of sites and commodities increase, the largest improvement is observed for instances where the sites are distributed randomly within a single region (the column labelled a). This observation can be explained by the number of cluster in the best solution and the change from the prescribed number of regions, shown in Figure 7. By comparing Figures 6(b) and 7(b) the greatest improvement in the objective values is directly proportional to the difference between the prescribed regions



(a) Number of clusters

(b) Change from initial

Figure 7: The number of clusters in the best solution.

and the best number of clusters. This result highlights the significant potential from applying a MAB algorithm to search for the best number of clusters, even when a geographical clustering may already exist.

Figure 6(b) also assess the improvement in the objective value if the best number of clusters is known in advance—shown by the orange bars. This result assesses the potential of the algorithmic techniques presented in Section 4 to reduce transportation costs. The improvements in the objective value range from 2.15% (100, 200, 1) to 17.37% (25, 25, 10). As the number of sites and commodities increase, the improvement in the objective value decreases. This is due to the fact that larger instances are more difficult to solve. As such, less iterations are performed, limiting the search for improving solutions. However, across all instances the application of the iterative algorithm leads to a significant reduction in transportation costs.

5.3 Performance of the iterative algorithm

The run time consumed by each of the WCP, PDP and SNDP is presented in Figure 8. An important observation from the Figure 8 is that while many of the instances require the full 3600 seconds of run time, some smaller instances terminate due to the best solution limit. When comparing Figures 6(b) and 8, these instances are able to achieve a large decrease in the transportation costs for the SCSNDP in reasonable run times.

The most striking observation from Figure 8 is that a significant proportion of the run time is consumed by the PDP. On average, the PDP requires 77% of the run time supplied to the MAB algorithm. In comparison, the SNDP requires between 8% and 29% of the run time—with an average of 18.5%. This difference in the run time requirement is a result of the complexity

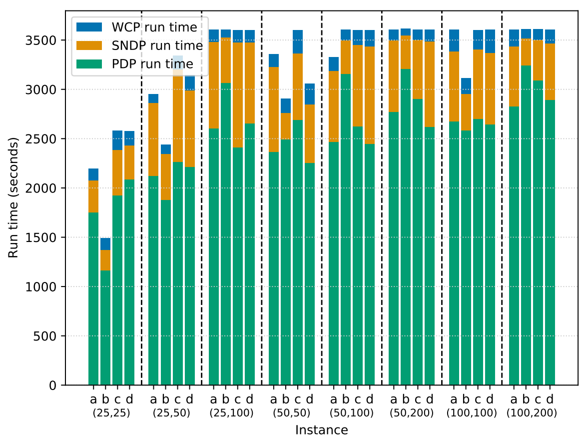
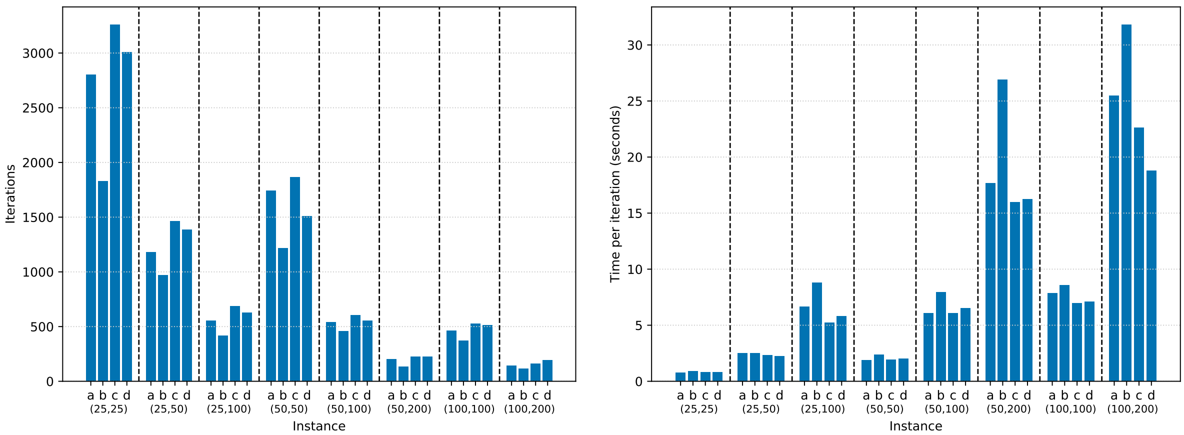


Figure 8: The run time of the solution algorithm.

in solving the corresponding problems. As seen in Figure 7, the number of clusters in the best solution is less than 7 on average across all instances. As a result, the formulations of the SNDP are very small and easy to solve. In regards to the PDP, the increase in the number of warehouse locations and commodities results in a more complex mathematical programming problem, irrespective of the number of clusters. These results point to an area of future work to reduce the run time devoted to solving the PDP.

The number of algorithm iterations presented in Figure 9 highlights that the difficulty to solve the SCSNDP is affected by the number of warehouse locations and commodities. Specifically, the time per iteration, presented in Figure 9(b), suggests that the increase in problem difficulty is more influenced by the number of commodities than the number of warehouse locations. For a fixed number of commodities, the time per iteration is similar across instances with different numbers of warehouse locations. However, for a fixed number of warehouse locations, the time per iteration increases super-linearly with the increase in the number of commodities. The results presented in Figure 9 supports the analysis of Figure 8, since the complexity of the PDP is directly related to the number of commodities.

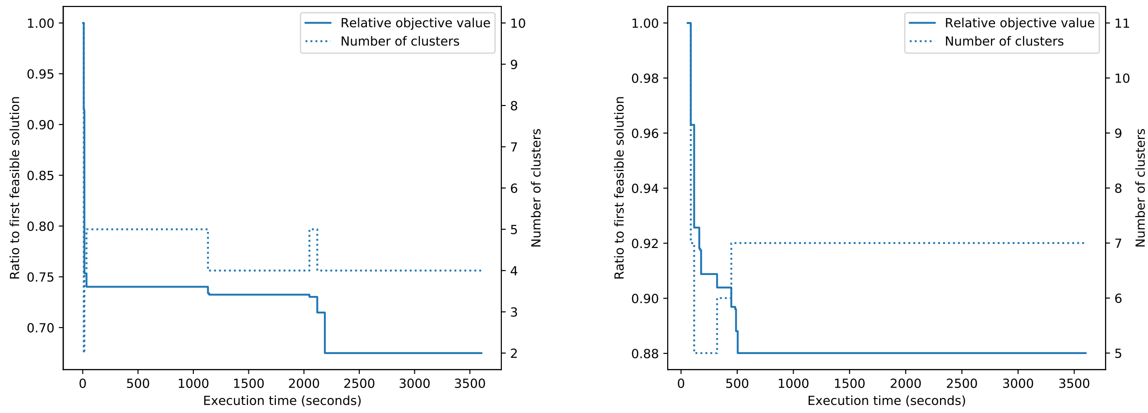


(a) Algorithm iterations (b) Time per iteration

Figure 9: The number of iterations between the SNDP and PDP and the time per iteration.

5.4 Behaviour of MAB algorithm

The ability of the MAB algorithm to search different numbers of clusters and improve the objective value is demonstrated with the examples in Figure 10. The first aspect of the developed solution algorithm that can be observed from these examples is the initialisation phase during the early stages of computation. This phase is characterised by the rapid reduction in the objective value and the many changes to the number of clusters. The benefit of the MAB algorithm can be observed after the initialisation phase, during the main solving process of the developed solution algorithm. Specifically, the MAB algorithm enables the search for solutions to the SCSNDP for a range of values for γ. The impact of this diverse search can be seen by the fact that the initial best value of γ does not lead to the best solution for the SCSNDP. Further, changes in the value of γ can trigger perturbations that helps escape regions of local optima.



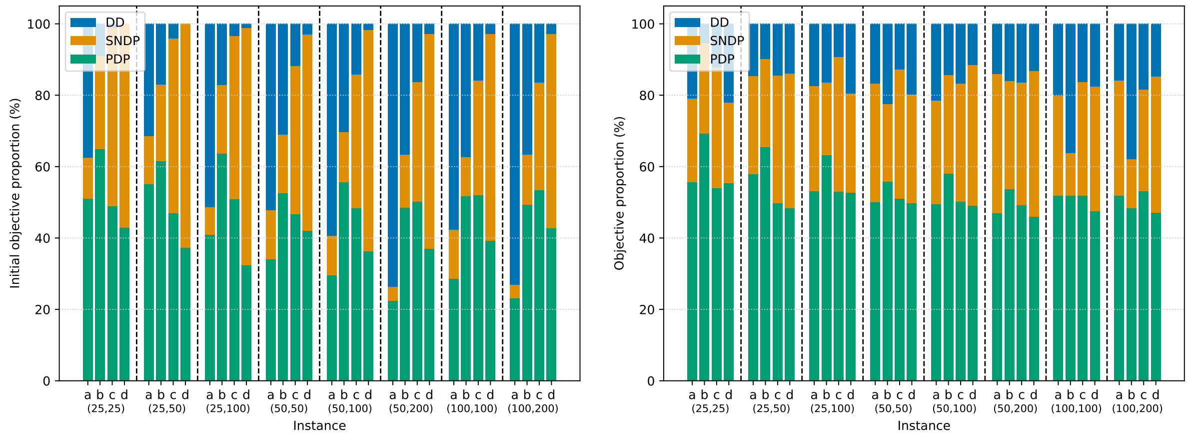
(a) (25, 100, 10) – Instance 1 (b) (100, 100, 1) – Instance 4

Figure 10: Objective value improvement (compared to the first initial solution) and the number of clusters.

5.5 Overview of the best solutions to the SCSNDP

The average objective value of the SCSNDP is presented in Figure 6(a). This objective is the sum of costs from direct deliveries (DD), the inter-cluster transportation (SNDP) and the intra-cluster transportation (PDP). The contribution of each component of the SCSNDP to the overall transportation costs for the initial and best solutions are presented in Figure 11. First, considering the initial feasible solutions, the use of different transportation options is very mixed across the instance set. However, this variation is mostly related to the number of clusters in the initial solution. It can be seen that for smaller values of γ, there is a reliance on direct deliveries. As the value of γ increases, costs shift towards the usage of inter-cluster transportation.

A comparison of Figures 11(a) and 11(b) highlights the behaviour of the proposed algorithm and its impact on the solutions to the SCSNDP. Across all instances, the contributions of the DD, intra- and inter-cluster transportation to the objective are fairly consistent. It appears that the variations in the costs are due to inferior final solutions for the SCSNDP. This is evidenced by instances (100, 100, 2) and (100, 200, 2), where a relatively small number of iterations

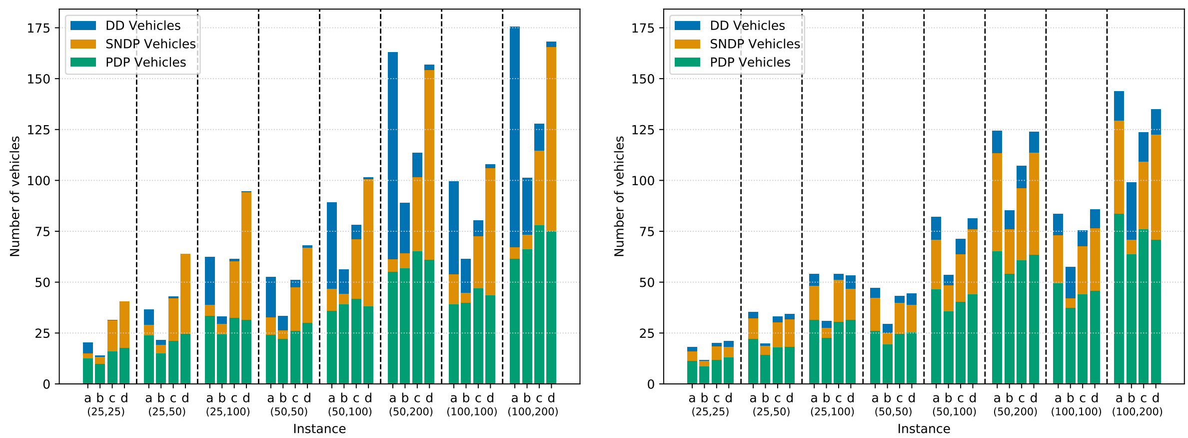


(a) Initial solution (b) Best solution

Figure 11: The distribution of the transportation costs across the different transportation types for the initial feasible and best solution.

are performed (see Figure 9(a))—limiting the efficacy of the developed solution algorithm. Interestingly, a large proportion of the costs are attributed to direct deliveries. Ideally, these costs should be reduced to zero; however, the nature of the problem instances requires their use to satisfy the pickup and delivery requests.

A major goal of the industry partner is to reduce the number of vehicles required to satisfy all pickup and delivery requests. Figure 12 highlights the effectiveness of the developed solution algorithm at achieving this goal. Comparing Figures 12(a) and 12(b), there is a clear decrease in the number of required vehicles across all instances. Similar to the results presented in Figure 11(a), Figure 12(a) shows that there is a reliance on DD vehicles for instances with a



(a) Initial solution (b) Best solution

Figure 12: The distribution of vehicles across the different transportation types.

smaller number of initial subregions—the instances labelled a. Alternatively, for instances with a larger number of subregions—instances c and d—there is a greater dependence on inter-cluster transportation. The distribution of vehicles for the best solutions, shown in Figure 12(b), is much more consistent within a (N, K) instance grouping. It can be observed that the number of warehouse locations and commodities has the biggest impact on the vehicle type usage, and not the number of initial subregions.

6 Conclusions

This paper investigates the integration of location clustering with transshipment and vehicle routing to form an adaptive transportation problem. A collection of mathematical programs have been developed as the basis of an algorithmic framework to find high-quality solutions for the SCSNDP. The integration of location clustering with transshipment and vehicle routing is achieved by formulating an MAB problem. The number of possible actions for the MAB solution algorithm is reduced using a line/local search technique to quickly identify the number of clusters that will lead to a small transportation costs. The synchronisation of intra- and inter-cluster transportation routes is effectively handled through an iterative solution algorithm.

The computational experiments highlight the potential of the MAB problem in reducing the transportation costs for the SCSNDP. The MAB algorithm successfully explores different settings for the number of clusters while exploiting the best performing setting. In many instances, then number of clusters differs from the geographically defined subregions, which leads to a significant reduction in costs. The use of the iterative algorithm and perturbation techniques is capable to reducing the overall transportation costs, typically resulting in a higher utilisation of intra-cluster transportation. Overall, the number of vehicles required to perform all pickup and delivery requests is significantly reduced.

The proposed SCSNDP algorithm exploits the power of mixed integer programming solvers within an iterative algorithmic framework. However, the computational results highlight a number of bottlenecks in the effectiveness of the approach. The most prominent of these is the solution time of the PDP, which greatly limits the number of iterations that can be performed in the given time limits. Through the application of parallel computing and decomposition techniques, we strive to reduce the run times of the PDP and accelerate the solution algorithm of the SCSNDP. It is expected that a decrease in the per iteration time will lead to a large reduction in transportation costs. Decomposition techniques and algorithm enhancements will play a crucial role in delivering high-quality supply chain management solutions.

References

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[2] S. Barreto, C. Ferreira, J. Paixao, and B. S. Santos. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3):968–977, 2007.

[3] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[4] P. Buijs, I. F. Vis, and H. J. Carlo. Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239(3):593–608, 2014.

[5] C. Contardo, V. Hemmelmayr, and T. G. Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39(12):3185–3199, 2012.

[6] I. Contreras and E. Fernández. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680–697, 2012.

[7] J.-F. Cordeau, G. Laporte, J.-Y. Potvin, and M. W. Savelsbergh. Transportation on demand. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 429–466. Elsevier, 2007.

[8] T. G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

[9] G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis. VRP with pickup and delivery. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, chapter 9, pages 225–242. Society for Industrial and Applied Mathematics, 2002.

[10] M. Drexl. Synchronization in vehicle routing—A survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.

[11] M. Drexl and M. Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308, 2015.

[12] H. Heggen, Y. Molenbruch, A. Caris, and K. Braekers. Intermodal container routing: Integrating long-haul routing and local drayage decisions. *Sustainability*, 11(6), 2019.

[13] M. Lam and J. Mittenthal. Capacitated hierarchical clustering heuristic for multi depot location-routing problems. *International Journal of Logistics Research and Applications*, 16(5):433–444, 2013.

[14] J. Medina, M. Hewitt, F. Lehuédé, and O. Péton. Integrating long-haul and local transportation planning: The service network design and routing problem. *EURO Journal on Transportation and Logistics*, 8(2):119–145, 2019.

[15] S. Melkote and M. S. Daskin. An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice*, 35(6):515–538, 2001.

[16] S. Mitrović-Minić and G. Laporte. The pickup and delivery problem with time windows and transshipment. *INFOR: Information Systems and Operational Research*, 44(3):217–227, 2006.

[17] G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

[18] G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45(3):364–380, 2011.

[19] H. L. Petersen and S. Ropke. The pickup and delivery problem with cross-docking opportunity. In J. W. Böse, H. Hu, C. Jahn, X. Shi, R. Stahlbock, and S. Voß, editors, *Computational Logistics*, pages 101–113, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[20] A. Rais, F. Alvelos, and M. Carvalho. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539, 2014.

[21] M. Salama and S. Srinivas. Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, 114:620–642, 2020.

[22] F. A. Santos, G. R. Mateus, and A. S. da Cunha. The pickup and delivery problem with cross-docking. *Computers & Operations Research*, 40(4):1085–1093, 2013.

[23] M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

[24] Y. Wang, K. Assogba, Y. Liu, X. Ma, M. Xu, and Y. Wang. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Systems with Applications*, 104:244–260, 2018.

[25] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau, and G. Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2009.

[26] N. Wieberneit. Service network design for freight transportation: A review. *OR Spectrum*, 30(1):77–112, 2008.

[27] D. Wolfinger. A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. *Computers & Operations Research*, 126:105110, 2021.

[28] D. Wolfinger, F. Tricoire, and K. F. Doerner. A matheuristic for a multimodal long haul routing problem. *EURO Journal on Transportation and Logistics*, 8(4):397–433, 2019.