The Fulfillment Regionalization Problem

Nidhima Grover[1,2], Xiaoyan Si[2], and Alejandro Toriello[1]

[1] *Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA*
[2] *Amazon, Bellevue, USA*

Abstract

In many retail industries, the retailer can choose the inventory location or fulfillment center (FC) that fulfills an order, yielding opportunities for inventory pooling and product selection expansion. However, fulfillment decisions are complex and must consider cost and speed, among various factors. With the unprecedented growth of the retail industry, companies must look for strategies that tackle the scale and complexity of fulfillment decisions. Regionalization is one such strategy, which divides the fulfillment network into a set of inter-connected regions: orders that originate within a region are primarily fulfilled by centers associated with the region. This structure simplifies the design of the fulfillment network, and has recently provided retailers with significant gains in cost and speed. In this study, we propose an optimization model to design the regions while simultaneously assigning fulfillment centers to match each region's demand; to our knowledge, this is a novel problem that has not been studied in the literature. Two of the model's main challenges include a non-linear objective function and contiguity constraints on the regions. We propose a local search heuristic to solve the problem at scale, along with efficient lower bounds to benchmark solution quality. Furthermore, we demonstrate that choosing appropriate parameters for designing regions can have a significant effect on solution quality, and can result in more demand being fulfilled within service guarantee deadlines.

1. Introduction

In today's retail industry, sales to customers occur through a spectrum of channels, including the traditional buying in store, buying online with pickup in store [37], buying online and delivering to the customer, and buying in store but having the order delivered. Compared to traditional walk-in demand, the other formats allow the retailer to choose which fulfillment center (FC) to fulfill orders from. This flexibility allows for inventory pooling, reducing the need for expensive safety stocks in every local store, which in turn opens up storage space for a much broader product selection [39]. Optimizing fulfillment decisions is highly complex, more so if the retailer owns the fulfillment network. Sub-optimal policies can lead to higher fulfillment cost, slower delivery speed, and lower customer satisfaction. With the unprecedented growth of the retail industry, retailers are challenged to re-think their fulfillment policies. Regionalization is one strategy to deal with the complexity of large-scale fulfillment networks: by dividing a national network into regions, regionalization drastically simplifies fulfillment decisions and has recently been shown to offer significant gains to retailers [34].

When inventory of different products is distributed across multiple FCs, it is highly complex to optimize the fulfillment decisions for cost and speed [1]. The complexity comes from demand uncertainty across geography and products, inventory imbalance among the FCs, and capacity constraints within the FCs. In a naive heuristic, one would assign a customer order to the nearest FC until that FC stocks out or reaches its maximum order processing capacity, in which case, the order is routed to the next nearest FC. Such a myopic solution does not preserve inventory or FC capacity for future orders, and may yield higher total cost or slower average fulfillment speed. In the literature, researchers have proposed methods to reduce the impact of myopic real-time decisions, including delaying the fulfillment decision by batching accumulated orders [47], setting thresholds on how much inventory each inventory source can send to a particular fulfillment channel [19], randomizing the fulfillment decision by choosing inventory sources with a given probability [23], or adding opportunity costs to the immediate reward earned from each decision using linear programming (LP) [2] or a primal-dual algorithm [1]. The fulfillment problem is even more complex when the retailer leverages its own transportation network to deliver packages: fulfillment resources such as hub capacities and truck scheduling are deployed before customer orders are realized. In this case, fulfillment decisions must align with resource availability to minimize delayed fulfillment or wasted capacity.

Regionalization is a strategy to manage the complexity of a large fulfillment network. It divides the full network into a set of regions such that each region has sufficient fulfillment capacity to satisfy its demand. Regionalization does not completely cut off cross-region fulfillment and maintains each region's access to the entire product selection. For an order originating in a region, we prioritize its fulfillment from the set of FCs assigned to that region unless the item is out-of-stock in the region, or when it is advantageous to fulfill from another region (e.g., to consolidate with an outstanding shipment). Regions should be large enough that the set of in-region FCs can collectively fulfill the majority (e.g., 70%) of the region's demand, but also compact enough to ensure in-region fulfillment is limited to a small number of FCs and fulfillment is fast. From the perspective of operations, regionalization makes in-region flows more predictable, which simplifies the connectivity of the fulfillment network and improves resource planning. It also enables tighter coordination between inventory management and fulfillment execution, as the two must work together to ensure a high level of in-region fulfillment [21, 34].

The challenge in regionalization is designing regions with the appropriate geographical size and demand, and assigning FC's to regions so that a region's aggregate fulfillment capacity meets its demand. This paper proposes a model and method to solve such a problem. The main contributions are:

1. We formally define the fulfillment regionalization problem, a graph partitioning problem where demand needs to be balanced with multiple supply nodes in each partition. Such a problem setting has not been studied in the literature before to our knowledge. Two of the formulation's main challenges include a non-linear objective function and contiguity constraints for each region.

2. In special cases, we prove that contiguity holds at optimality without the need to enforce via constraints. We show that contiguity may not be guaranteed in general if not enforced, motivating the need to model and constrain contiguity explicitly.

3. We propose a construction and improvement heuristic based on local search to solve the problem at scale. We use contiguity constraints based on shortest-path trees, and show that the price of imposing contiguity is small, usually 1-2%. We also show that the loss in solution quality compared to modeling contiguity exactly is small, again roughly 1-2%.

4. We introduce simple and efficient lower bounds for the problem, and show that the optimality gap is between 11-41% for instances of practical interest, depending on the number of facilities, number of regions, and design parameters for the problem.

5. We present a computational study on practical instances to benchmark solution quality. We evaluate the quality of the region designs and show that choosing the right design parameters can yield lower cost and fast fulfillment speed.

2. Literature Review

There are several streams of literature that motivate or are related to regionalization. [14] discuss the fulfillment policy of an online retailer, where each local distribution center (FDC) has non-overlapping service areas and demand can only spill over to the designated regional distribution center (RDC) if the assigned FDC stocks out. The authors propose a solution to add limited flexibility of cross-FDC fulfillment, and choose not to allow all such possible cross-fulfillment, arguing that increasing flexibility leads to a diminishing benefit but increases risk of complexity. These principles are also found in the literature on manufacturing process flexibility [24]. Such insights translate to controlling the flexibility of national fulfillment and prioritizing flows from regional FCs.

There is also literature that motivates cost and speed improvement in regionalization. Recent work [17] finds that creating service zones for fulfillment facilities through partitioning can improve overall social welfare as measured by travel time plus waiting time at each facility, as opposed to a decentralized equilibrium where each customer individually optimizes their own service time. There is an improvement to fulfillment speed with centralized partitioning, especially when the facilities' capacities do not align with the geographical demand distribution.

As an example of a study that observes cost savings from regionalization, [16] study a distributionally robust newsvendor problem on a metric with multiple facility locations; the objective is to minimize the total overage, underage and fulfillment costs. A hierarchical policy is near-optimal: facilities are assigned to hierarchical clusters, and demand is fulfilled from a nearby cluster while maintaining an inventory balance across clusters; clusters roughly correspond to our regions.

One of the problems closely related to regionalization is political districting, the problem of clustering census blocks into electoral districts, where we need to balance population across districts. Contiguity is one of the problem's main challenges [6]: the sub-graph for each district in the adjacency graph must be connected. The most popular contiguity constraints are due to [40], where district centers send flow to the nodes assigned to the district through the edges [6]; [46] prove the correctness of different flow and cut-based formulations in the literature and compare their strengths. They find that the strongest formulations are cut-based, but they have exponentially many constraints. When compactness is not modeled explicitly, [41] illustrates that exact contiguity constraints allow solutions where districts are not compact. On the other hand, tree-based contiguity constraints (which we use in this study) are faster in practice but more restrictive. For instance, [44] enforce the assignment of adjacent units that lie on the shortest path from a district center to an assigned unit. In school districting, [8] relax these constraints to distance-based contiguity constraints: a unit can be assigned to a district only if at least one of the adjacent units that is closer to the district center is also assigned. In terms of solution approaches, [46] use a randomized heuristic with local search on the classical model in [20], with contiguity modeled using cut-based formulations. Recent work [18] has used enumeration methods to generate feasible districts, using a randomized recursive splitting technique to generate a large number of feasible districts that always fit together to give a feasible plan, with contiguity modeled similarly to [8].

Apart from political districting, there are several related spatial optimization models in the literature. One example is forestry, where planners want to preserve contiguous regions of a forest over a planning period of several decades to maximize the number of species and habitats protected. [10] studies contiguity of the districts thus created and find that rooted models of contiguity are easier to solve compared to "unrooted" models, partly motivating our approach of using a rooted model. Other related districting problems with contiguity requirements involve school redistricting [8, 35], police-zone design [49], and partitioning data across processors in parallel database systems [3].

The above districting problems are graph partitioning problems with only one type of node, and may have constraints on the size of each region. Conversely, facility location problems also involve supply nodes (as does our problem) and ensure that the supply in each partition is at least the demand of the partition, but generally with a single supply node in each partition. Algorithms for this problem and its variants are well studied in the literature [28, 42]. [44] also imposed contiguity in such a problem in the form of shortest-path contiguity constraints, but with a linear objective function. On the other hand, [9] study the robust single-facility location problem, and reformulate the problem as a nonlinear fractional problem for which an optimal solution can be found in sub-quadratic time for some choices of the distance norm. [33] study facility location problems with a general non-linear objective function and devise a heuristic approach based on local-search on the continuous counterpart and rounding to a discrete solution. Researchers have also worked on related problems in power grids, e.g. [22, 43], where a graph must be partitioned into "islands", such that each island satisfies certain operational independence conditions such as supply and demand balance.

Multi-sourcing in facility location problems allows customers to receive inventory from more than one facility, where the fulfillment decision may be probabilistic. [36] address this problem for a single product type and formulate the problem as a nonlinear integer program, while [48] study this problem for multiple product types. More recent work [5] studies this problem when demand is non-stationary under a multi-period two-stage stochastic setting, and [29] allow dynamic sourcing with flexible order-splitting. Our problem is different in the sense that we partition the set of facilities, where each partition is assigned to a set of customers. Unlike multi-sourcing, these sets of facilities assigned to sets of customers are non-overlapping. This also involves the additional decision of creating the sets of customers served by each set of facilities, which makes the problem challenging.

Continuous approximation (CA) methods for the facility location problem [13] consider customer demand density at each location in a continuous space instead of discrete customer locations, and use analytical techniques to determine the optimal facility locations and customer partition. For example, [7] use continuous approximation to partition a region into sub-regions for each facility, such that the demand of the sub-region equals the capacity of the sub-region's assigned facility. They characterize the analytical form of the boundary curves in terms of the utility functions, and illustrate the boundaries for different kinds of utility functions, analyzing their contiguity. For enforcing contiguity in different problems, they add a utility function in the objective as a penalty term that generally provides a contiguous solution, instead of enforcing contiguity strictly.

The remainder of the paper is structured as follows. In Section 3, we define our problem and formulate it as an integer non-linear program, and also as an equivalent mixed-integer linear program. In Section 4, we analyze contiguity, and motivate the need to model contiguity explicitly. In Section 5, we present a heuristic solution approach to solve the problem based on local search, and quantify the loss in solution quality from contiguity constraints. In Section 6, we present simple and efficient lower bounds for the problem. In Section 7, we present a computational study, and conclude and discuss future research directions in Section 8.

3. Problem Formulation

Consider a set of geographical demand areas or *tiles* I, also represented by a set of nodes in an undirected planar graph (I, E); an edge between any two nodes represents geographical adjacency. For adjacent nodes $i, j \in I$, the distance between them is $d_{ij} \geq 0$, which we assume symmetric for simplicity. Distances between non-adjacent nodes are shortest-path distances, and thus satisfy the triangle inequality. A tile $i \in I$ has an average daily demand $q_i > 0$.

The set of fulfillment centers (FC) is K, with $|K| = \ell$. FC $k \in K$ has a shipping capacity C_k^{ship}, the maximum number of units the FC can process in a single day, and a storage capacity C_k^{store}, the maximum number of units the FC can store in one replenishment cycle. The distance from FC k to tile i is d_{ik}, and also satisfies the triangle inequality.

The goal is to partition the tiles into a set of regions J with $|J| = m$, each of which is connected in (I, E), and to assign one or several FCs to each region, such that the FCs assigned to a region have sufficient total shipping capacity to satisfy the region's demand. In addition, we introduce parameters $Q_{\min}^{\text{store}}$ and $Q_{\max}^{\text{ship}}$ to respectively denote the minimum storage and maximum ship capacity of a region; these constrain regions and FC assignments to satisfy practical concerns, such as limiting the number and size of FCs assigned to a region or ensuring most customers have access to timely service from their assigned FCs. The objective is a partition and FC assignment that minimizes the average distance a unit of demand travels from an FC to a tile.

To model contiguity, we use shortest path constraints. To do so, we choose a tile as the root for each region; for root $i \in I$, we let $P_{ii'}$ denote the predecessor tile in a shortest path from tile i' to root i in the adjacency graph (I, E).

To formulate the model, we use x_{ij} to denote the assignment of tile $i \in I$ to a region $j \in J$, y_{kj} to denote the assignment of an FC $k \in K$ to a region $j \in J$, and z_{ij} to indicate whether tile $i \in I$ is the root node of region $j \in J$. We formulate the problem as an integer non-linear program as follows:

$$\min \sum_{j \in J} \sum_{i \in I} \sum_{k \in K} q_i * d_{ki} * x_{ij} * \left(\frac{C_k^{\text{ship}} y_{kj}}{\sum_{k' \in K} C_{k'}^{\text{ship}} y_{k'j}} \right) \tag{1a}$$

$$\text{s.t.} \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \tag{1b}$$

$$\sum_{j \in J} y_{kj} = 1 \quad \forall k \in K \tag{1c}$$

$$\sum_{i \in I} q_i x_{ij} \leq \sum_{k} C_k^{\text{ship}} y_{kj} \quad \forall j \in J \tag{1d}$$

$$\sum_{k \in K} C_k^{\text{store}} y_{kj} \geq Q_{\min}^{\text{store}} \quad \forall j \in J \tag{1e}$$

$$\sum_{k \in K} C_k^{\text{ship}} y_{kj} \leq Q_{\max}^{\text{ship}} \quad \forall j \in J \tag{1f}$$

$$\sum_{j \in J} z_{ij} \leq 1 \quad \forall i \in I \tag{1g}$$

$$\sum_{i \in I} z_{ij} = 1 \quad \forall j \in J \tag{1h}$$

$$x_{i'j} \leq x_{P_{ii'},j} + 1 - z_{ij} \quad \forall i, i' \in I, j \in J \tag{1i}$$

$$x_{ij}, y_{kj}, z_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K \tag{1j}$$

Constraints 1b and 1c respectively ensure that each tile and FC are assigned to exactly one region. Constraint 1d ensures that every region has sufficient fulfillment capacity to serve the demand of the region. We ensure minimum storage capacity with 1e, and limit shipping capacity of a region with 1f. We assume that the demand of a tile in a region is fulfilled from the FCs assigned to a region in proportion to their shipping capacities; operationally, this means the capacity of each FC in the region is utilized similarly. Hence, the objective 1a averages the distance from tile to FC using shipping capacity, a proxy for the average distance that is traveled by a unit of demand. Constraint 1g ensures that every tile is the root node of at most one region, and constraint 1h ensures that every region has exactly one root node. We model contiguity with shortest paths using 1i, ensuring that if a tile is assigned to a region, then all tiles lying on the shortest path from that tile to the root node are also assigned to the same region. This method of modeling contiguity is slightly restricted; that is, shortest-path contiguity implies exact contiguity, but the converse is not true.

Let $D \subseteq I$ denote the set of tiles assigned to a region. The region defined by D is contiguous if the adjacency sub-graph induced by D is connected [46]. In contrast, the region has shortest-path contiguity with root node $i \in D$ if, for every vertex $i' \in D$, all vertices on a shortest path between i' and i in G lie in D [12, 32, 50]. This implies that there exists a path on the induced sub-graph $G[D]$ connecting i with every vertex i' that lies in D, implying that $G[D]$ is connected; hence, D

satisfies exact contiguity. However, exact contiguity does not imply shortest-path contiguity, and [12] provide examples of contiguous solutions that are overlooked in shortest-path contiguity with a given set of root nodes.

3.1. Linearization of Objective Function

The objective function described above captures the average distance traveled by demand; it is a non-linear function, which makes problem (1) particularly challenging. The Charnes-Cooper transformation [11] is a technique to transform a non-linear program with an objective that is a single fraction of two linear functions to an LP using a few additional variables and constraints; however, the technique cannot be extended to a sum of several such ratios. [4] were the first to address this problem by extending [15] to the sum-of-ratios problem. They found that when each ratio is separable in the decision variables, then the solution is unique. In such a case, the objective is quasi-concave, implying that any local optimum is also a global optimum. They present an iterative algorithm for the problem when the coefficients in the constraint matrix are also non-negative, which does not hold true for our problem. Later, [26, 27] developed an efficient branch-and-bound method for the problem of maximizing a sum-of-ratios objective; however, it requires both the numerator and denominator to be positive over the entire polytope. Moreover, to the best of our knowledge, much of the literature studying the problem of maximizing the sum-of-ratios objective has been limited to a continuous domain; [38] provides an extensive review of related algorithms.

Below, we describe a way to reformulate our problem as a mixed integer linear program (MILP); the resulting formulation serves as a benchmark for the quality of our solutions (see Section 7). Let β_{kj} denote the proportion of regional capacity that is offered by facility k, and let

$$\alpha_{ijk} = x_{ij}\beta_{kj}, \qquad \beta_{kj} = \left(\frac{C_k^{\text{ship}} y_{kj}}{\sum_{k' \in K} C_{k'}^{\text{ship}} y_{k'j}}\right) \quad \forall k \in K, j \in J. \tag{2a}$$

We show that the above equations are equivalent to the following linear constraints.

$$\alpha_{ijk} \leq x_{ij} \quad \forall i \in I, j \in J, k \in K \tag{3a}$$

$$\alpha_{ijk} \leq \beta_{kj} \quad \forall i \in I, j \in J, k \in K \tag{3b}$$

$$\alpha_{ijk} \geq x_{ij} + \beta_{kj} - 1 \quad \forall i \in I, j \in J, k \in K \tag{3c}$$

$$\beta_{kj} \leq y_{kj} \quad \forall k \in K, j \in J \tag{3d}$$

$$\sum_{k \in K} \beta_{kj} = 1 \quad \forall j \in J \tag{3e}$$

$$\beta_{kj}/C_k^{\text{ship}} + (1 - y_{kj})/C_k^{\text{ship}} \geq \beta_{k'j}/C_{k'}^{\text{ship}} \quad \forall k, k' \in K, j \in J \tag{3f}$$

$$\alpha_{ijk}, \beta_{kj} \geq 0 \quad \forall i \in I, j \in J, k \in K \tag{3g}$$

The objective then becomes:

$$\min \sum_{j \in J} \sum_{i \in I} \sum_{k \in K} q_i d_{ki} \alpha_{ijk} \tag{3h}$$

Proposition 1. *The model defined by equations 1b-1j and 3a-3h is equivalent to the model defined by equations 1a - 1j.*

Proof. Constraint 3e implies $\beta_{kj} \leq 1 \, \forall k, j$. Hence, the set of constraints 3a-3c and 3g are a McCormick envelope [31] of the bilinear expression 2a. When $x_{ij} = 0$, constraints 3a and 3g imply $\alpha_{ijk} = 0 \, \forall k$ and constraints 3b and 3c are redundant since $z \in [0, 1]$. When $x_{ij} = 1$, constraints 3b and 3c imply $\alpha_{ijk} = \beta_{kj} \, \forall k$ and constraints 3a and 3g are redundant since $z \in [0, 1]$. Hence, the first equation 2a is equivalent to the set of constraints 3a-3c and 3g.

When $y_{kj} = 0$, constraint 3d implies $\beta_{kj} = 0$ and constraint 3f is redundant $\forall k' \in K$. For a region j, let K_j be the set of facilities for which $y_{kj} = 1$. For $k, k' \in K_j$, constraint 3f implies $\beta_{kj}/C_k^{\text{ship}} = \beta_{k'j}/C_k^{\text{ship}} = \beta_j$. Substituting values of β_{kj} in constraint 3e, we get $\sum_{k \in K_j} C_k^{\text{ship}} \beta_j = 1 \Rightarrow \beta_j = 1/\sum_{k \in Kj} C_k^{\text{ship}} \Rightarrow \beta_{kj} = C_k^{\text{ship}}/\sum_{k \in Kj} C_k^{\text{ship}}$ when $y_{kj} = 1$. For $k \in K_j$ and $k' \in K \setminus K_j$, $y_{k'j} = \beta_{k'j} = 0$, constraint 3f is redundant since $\beta_{kj} \in [0, 1]$. Hence, the second equation in 2a is equivalent to the set of constraints 3d-3g. Hence, we can replace the objective function given by equation 1a with equation 3h. □

4. Analysis of Contiguity in Regions

From an operational perspective, we desire contiguous regions for ease of management and for customer service, for example, to avoid frequent abrupt changes in last-mile delivery operations for customers in nearby locations.

A natural question is when we can guarantee contiguity without explicitly modeling it, since we are minimizing distances between tiles and FCs. We find that when regions have exactly one assigned facility, we can guarantee contiguity if tiles have uniform demand. However, when regions have more than one assigned facility, we cannot guarantee contiguity even with uniform demand. We discuss why this happens, which motivates the need to model contiguity explicitly. See also [30] for related results and further discussion.

Proposition 2. *Consider the restriction of our problem in which a single FC serves each region. Letting x_{ik} indicate the assignment of tile i to the region served by k, this corresponds to minimizing $\sum_{i,k} q_i d_{ik} x_{ik}$, subject to constraints analogous to 1b, 1d, and 1j. Suppose demand is uniform, $q_i = q$ for all i, and every facility is identified with a particular tile: for every $k \in K$, there is some tile i with $d_{ik} = 0$. Then there exists an optimal solution that is contiguous, and furthermore satisfies shortest-path contiguity constraints, where each root node is a tile identified with a facility.*

Proof. We first show that each facility will be located inside its own region, i.e., the tile in which the facility is located will be assigned to it. Consider facility k, and suppose the tile i with $d_{ik} = 0$ is not assigned to k, and is instead assigned to k'. Choose some i' assigned to k, and swap the two assignments. The change in cost is $d_{ik} + d_{i',k'} - d_{i,k'} - d_{i',k} = d_{i',k'} - d_{i,k'} - d_{i',k} \leq 0$, where non-positivity follows from the triangle inequality. So we may assume a facility's tile is assigned to it.

We now proceed similarly with other tiles. Suppose the solution is not contiguous; choose a non-contiguous region served by k, and some i assigned to k that is not connected to it within its assigned service region. Consider a shortest path from i to k; choose some i' on this path that is not assigned to k; such an i' must exist since i is not connected to k. Let k' be the facility i' is assigned to, and swap the two assignments. The change in cost is

$$d_{i',k} + d_{ik'} - d_{ik} - d_{i',k'} = d_{i',k} + d_{ik'} - (d_{ii'} + d_{i',k}) - d_{i',k'} = d_{ik'} - d_{ii'} - d_{i',k'} \leq 0,$$

where the first equality follows from the shortest path assumption, and the non-positivity is again a consequence of the triangle inequality. Notice that this swap operation improves or preserves the objective value even if i is connected to k, as long as i' is not assigned to k; therefore, we can also perform it if the solution does not satisfy shortest-path contiguity constraints. □

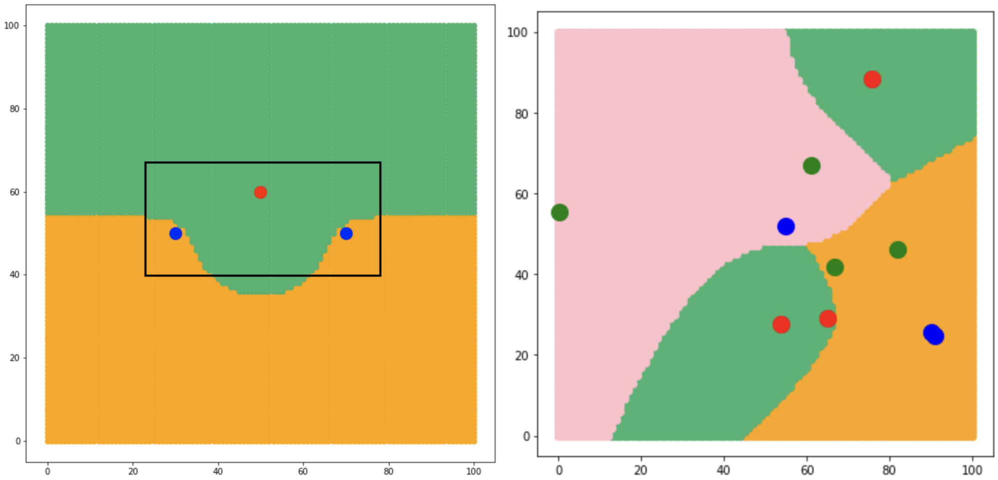
While contiguity holds when each region has exactly one facility, when regions contain multiple facilities and we minimize the objective 1a, guaranteeing contiguity becomes challenging. In all of the examples below, we fix the assignment of facilities to regions, and then assign tiles to regions to minimize the objective 1a.

1. Contiguity depends on the adjacency graph. For example, a solution can be contiguous if the tile set constitutes an infinite space, but no longer contiguous if we restrict the graph in a natural way. Figure 1a shows a simplified setting, where we assign tiles to regions for which the objective function is minimized (assuming Euclidean distances) without imposing additional constraints. If we restrict the adjacency graph to the rectangle shown in the black box, we obtain a non-contiguous solution.

2. Contiguity depends on the manner in which facilities are clustered, which is influenced by constraints 1d, 1e, and 1f. Figure 1b shows an example of non-contiguity in a setting where facilities are clustered into three regions; the green tiles are assigned to the red facilities, forming a non-contiguous region.

3. Contiguity depends on the distances used in the objective 1a. In Figure 2a, tiles i and j are adjacent and have the same demand. Their assignments result in non-contiguous regions, but swapping the assignments actually increases the objective, because the distances between tiles are based on road distances.

The above examples show that it is necessary to explicitly model contiguity in the general setting of the problem. We impose contiguity with shortest-path constraints, similarly to [12, 32, 50]. Shortest path constraints also tend to yield compact regions; see e.g. Figure 3 below. In contrast, exact contiguity constraints often result in non-compact regions [41]. The cost of a solution with exact contiguity constraints will lie between the cost of a solution without contiguity, and one with shortest path contiguity. In the next section, we describe our solution approach and quantify the loss in solution quality compared to exact contiguity.



(a) (b)

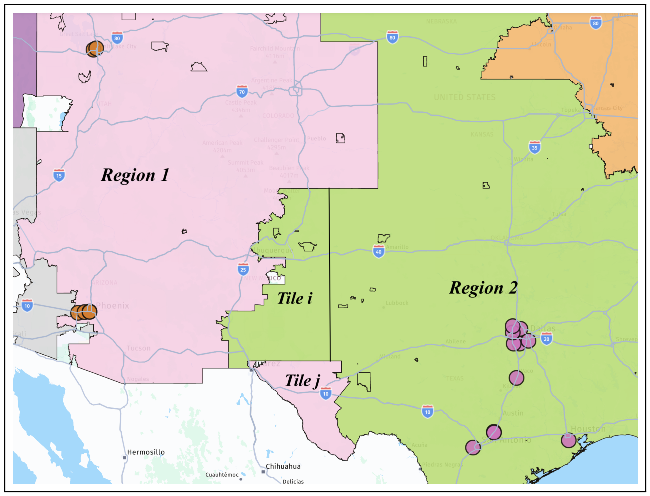
Figure 1: Examples of non-contiguity in a simplified setting with Euclidean distances. In both examples above, all facilities have the same capacity. Each tile is assigned to a cluster of facilities that minimizes the objective in equation 1a with euclidean distances, without additional constraints. Green tiles are assigned to the red facilities, orange tiles are assigned to the blue facilities, and pink tiles are assigned to the green facilities.

5. Solution Approach

Although the problem can be reformulated as an MILP, as we show in Section 3, solving this formulation with optimization software is intractable at practical scale, as we discuss in more detail in Section 7. We therefore propose solving the problem using a heuristic method. Observe that when we fix FC-to-region assignment variables y_{kj}, the objective becomes linear in tile-to-region assignment variables x_{ij}. This motivates our algorithm, as we iteratively fix the former to optimize the latter, and then improve the former using local search. Likewise, we iteratively fix the root node variables z_{ij} in each iteration to be the centroids of the current regions.

The heuristic that we employ is shown in Algorithm 1. It is a construction and improvement heuristic that uses local search in the improvement step. We use an LP to evaluate feasible moves in local search and execute the move with the lowest cost in the LP; this allows moves that may not improve the objective in the original model. We also experimented with an alternate approach in which we only execute improving moves, but it did not result in any significant change in solution quality.

Figure 3 illustrates the solution obtained from the algorithm for an instance with 884 tiles,



(a)

Figure 2: Example of non-contiguity in a practical setting with road distances. In the example above, tiles i and j are assigned to clusters of facilities that minimize the objective in equation 1a with road distances, under capacity constraint 1d and 1j. Green tiles are assigned to the pink facilities, and pink tiles are assigned to the orange facilities; gray lines denote highways.

100 FCs, 10 regions, $Q_{\min}^{\text{store}}$ set to 6% and $Q_{\max}^{\text{ship}}$ to 14% of storage and shipping capacities, respectively. Observe that FCs can be physically outside their assigned regions because the geographic distribution of FCs can be different from that of demand.

We use the contiguity constraints (1i) in step 3 of the construction heuristic and step 6 of the local search in the algorithm. Table 1 shows the difference in cost of the solutions obtained from the construction heuristic without contiguity and with shortest-path contiguity. Since the cost of a solution with exact contiguity constraints lies between the two, this implies that the loss in solution quality from modeling contiguity using shortest-path constraints is small (1-2%) compared to modeling contiguity exactly. This also shows that the price of imposing contiguity itself (the cost difference between solutions with exact contiguity constraints and without contiguity) is small (1-2%). One possible reason is that we assume tiles have uniform demand. By Proposition 2, if regions had a single FC, we would be guaranteed a contiguous optimal solution; it may still be the case that solutions are *almost* contiguous even though we have multiple FCs assigned to a region.

Algorithm 1 Construction and Improvement Heuristic for FRP

Input:
I: Set of tiles
J: Set of regions
K: Set of facilities
$q_i \; \forall i \in I$: Demand of each tile
$C_k^{\text{ship}} \; \forall k \in K$: Shipping capacity of each facility
$C_k^{\text{store}} \; \forall k \in K$: Storage capacity of each facility
$d_{ki} \; \forall k \in K, i \in I$: Distance by road from each facility to each tile
$Q_{\min}^{\text{store}}$: Minimum total storage capacity of facilities in each region
$Q_{\max}^{\text{ship}}$: Maximum total ship capacity of facilities in each region
$G = (I, E)$: Adjacency graph of all tiles
$d_{ii'} \; \forall i, i' \in I$: Distance between every pair of tiles
MaxIter: Maximum number of iterations for local search
MaxDist: Maximum allowed distance for FC moves or swaps in local search

Output:
$x_{ij} \; \forall i \in I, j \in J$: assignment of tiles to regions
$y_{kj} \; \forall k \in K, j \in J$: assignment of facilities to regions

Construction Heuristic:

Use Hess' formulation [20] on tiles with minimum and maximum demand of each region as $Q_{\min}^{\text{store}}$ and $Q_{\max}^{\text{ship}}$ respectively to obtain root node of each region
Compute parameters $d_{ij} \; \forall i \in I, j \in J$: distance from each tile to root node of each region; $d_{kj} \; \forall k \in K, j \in J$: distance from each facility to root node of each region
Compute $P_{ij} \; \forall i \in I, j \in J$: predecessor on shortest path from root node of j to tile i in G
Obtain y_{kj} by minimizing $\sum_{i,j} q_i d_{ij} x_{ij} + \sum_{k,j} C_k^{\text{ship}} d_{kj} y_{kj}$ subject to constraints 1b-1f, 1j.
Fix y_{kj} and minimize 1a s.t. constraints 1b-1f, 1j, 1i.

Local Search:

while iter $\leq$ MaxIter **do**
  $val_0 \leftarrow$ tile assignment LP (TALP): minimize 1a s.t. constraints 1b, 1d, and $x_{ij} \geq 0$ for fixed y_{kj} and z_{ij}.
  FC moves: Evaluate assignment of an FC to its neighboring regions if distance from FC to root node of region $\leq$ MaxDist: if constraints 1e and 1f are satisfied, evaluate objective from TALP, $val_1 \leftarrow$ cost of solution with lowest objective value
  FC swaps: Evaluate swapping region assignments of two FCs in neighboring regions if distance between FCs $\leq$ MaxDist: if constraints 1e and 1f are satisfied, evaluate objective from TALP, $val_2 \leftarrow$ cost of solution with lowest objective value
  If no feasible FC move or swap is found or $val_0 \leq max(val_1, val_2)$: end while
  Execute swap or move with lowest cost: fix y_{kj} and minimize 1a s.t. constraints 1b-1f, 1j, 1i
  Update root nodes to be the centroids of each region
  $iter \leftarrow iter + 1$
end while

Number of FCs	Number of regions	$Q_{\min}^{\text{store}}$ as % of $\sum_k C_k^{\text{store}}$	$Q_{\max}^{\text{ship}}$ as % of $\sum_k C_k$	Solution with SP contiguity constraints (x10^9)	Solution w/o contiguity constraints (x10^9)	Difference (%)
100	10	8	12	17.69	17.49	1.11
100	10	6	14	15.48	15.44	0.24
100	10	0.01	100	15.19	15.14	0.32
100	20	4	6	13.10	12.91	1.49
100	20	3	7	12.63	12.54	0.71
100	20	0.01	100	12.38	12.32	0.46
100	30	2.67	4	12.29	12.11	1.49
100	30	2	4.67	12.07	11.87	1.64
100	30	0.01	100	11.52	11.40	1.00
100	40	2	3	12.37	12.20	1.34
100	40	1.5	3.5	11.76	11.57	1.64
100	40	0.01	100	11.19	10.99	1.82
150	15	5.33	8	14.11	14.04	0.55
150	15	4	9.33	13.77	13.75	0.13
150	15	0.01	100	13.62	13.52	0.69
150	30	2.67	4	12.32	12.26	0.44
150	30	2	4.67	12.31	12.09	1.84
150	30	0.01	100	11.63	11.50	1.19
150	45	1.78	2.67	11.75	11.56	1.67
150	45	1.33	3.11	11.55	11.39	1.40
150	45	0.01	100	11.05	10.92	1.22
150	60	1.33	2	11.61	11.48	1.15
150	60	1	2.33	11.25	11.01	2.26
150	60	0.01	100	11.23	10.68	5.19

Table 1: Comparison of cost of solution from construction heuristic (without local search) without contiguity constraints and with shortest-path contiguity constraints

6. Lower Bounds

Next, we propose lower bounds to benchmark the quality of solutions. When attempting to solve it in Xpress, the MILP in Section 3.1 yields a trivial lower bound. Similarly, solving model (1) using Baron [25] gives a weak lower bound; see Section 7 for more details. With this motivation, we present some simple and relatively efficient lower bound models for the problem.

We begin with a bound in which we allow each FC to define its own region:

$$\min \sum_{i \in I} \sum_{k \in K} q_i d_{ik} x_{ik} \tag{4a}$$

$$\text{s.t.} \sum_{k \in K} x_{ik} = 1, \quad i \in I \tag{4b}$$

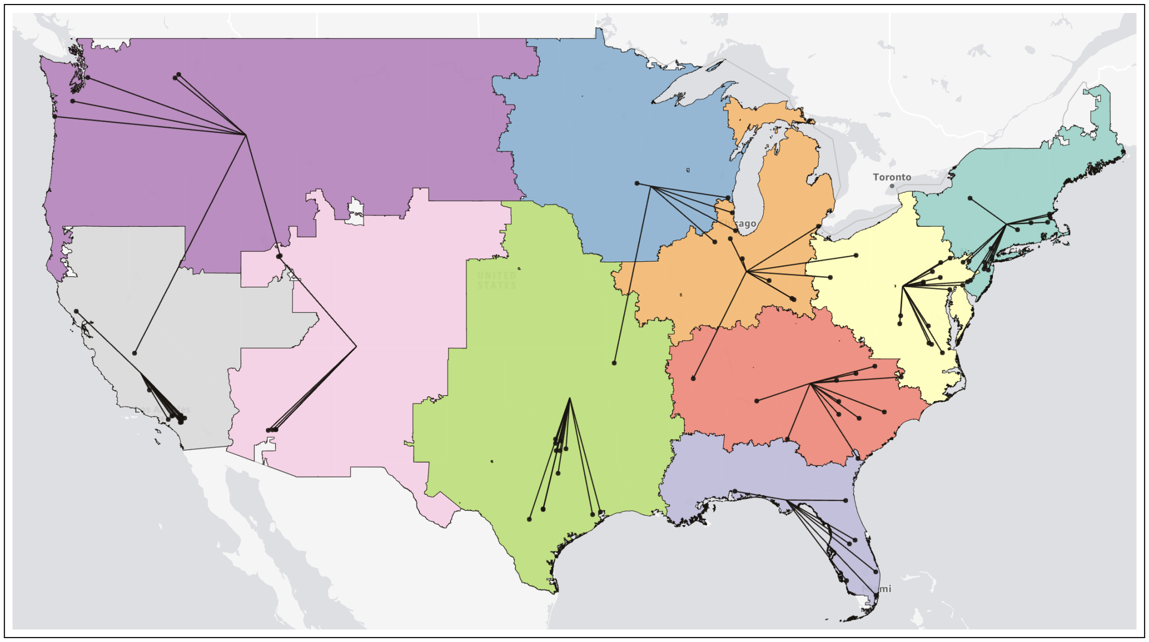


Figure 3: Illustration of the solution obtained from Algorithm 1 for a synthetic US instance with 884 tiles, 100 FCs, 10 regions, and where $Q^{\text{store}}_{\text{min}}$ is 6% and $Q^{\text{ship}}_{\text{max}}$ is 14% of total storage and shipping capacity, respectively. Black dots indicate locations of facilities and arcs represent their assignment to regions.

$$\sum_{i \in I} q_i x_{ik} \leq C^{\text{ship}}_k, \quad k \in K \tag{4c}$$

$$x_{ik} \geq 0, \quad i \in I, k \in K. \tag{4d}$$

In this formulation, x_{ik} represents the fraction of tile i demand assigned to the regions defined by FC k. The constraints ensure every tile's demand is fully served, and that demand assigned to a region defined by FC k does not exceed this FC's shipping capacity.

Proposition 3. The optimum of (4) is a lower bound for the optimum of (1).

Proof. Consider a relaxation of (1) with objective 1a, subject to constraints 1b, 1c, 1d, $x_{ij} \geq 0 \; \forall i, j$, and $y_{kj} \in \{0, 1\} \; \forall k, j$, where we increase J so that $|J| = |K| = \ell$. If $y_{kj} = 0$ for all $k \in K$ and some fixed $j \in J$, define the ratio in 1a to be zero for this j.

Because $|J| = |K|$, we obtain (4) if we identify an FC with a region and vice versa, i.e. when $y_{jk} = 1$ if and only if $j = k$. We establish the proof by showing that such an assignment is always optimal; that is, it is never beneficial to maintain two or more FC's clustered in a single region in this relaxation. Suppose we have a solution (x, y) where two or more FC's are assigned to the same region, $y_{kj} = 1$ for $k \in K' \subseteq K$ and some fixed j, where $|K'| \geq 2$. We construct a solution $(\hat{x}, \hat{y})$ in which the FC's in K' are instead each assigned to a separate region. For simplicity, we identify each FC's region with its index, so that $\hat{y}_{kk} = 1$ for $k \in K'$. For $\hat{x}$, we split the value of the original variable in proportion to C_k^{ship},

$$\hat{x}_{ik} = \frac{x_{ij} C_k^{\text{ship}}}{\sum_{k' \in K'} C_{k'}^{\text{ship}}}, \quad i \in I, k \in K'.$$

All other variables in $(\hat{x}, \hat{y})$ are copied from the original (x, y).

By construction, the new solution satisfies 1b, 1c. For the capacity constraint 1d, we have

$$\sum_{i \in I} q_i \hat{x}_{ik} = \frac{C_k^{\text{ship}}}{\sum_{k' \in K'} C_{k'}^{\text{ship}}} \sum_{i \in I} q_i x_{ij} \leq C_k^{\text{ship}}, \quad k \in K',$$

where the inequality follows because the original x satisfies 1d. Finally,

$$\sum_{i \in I, k \in K'} q_i d_{ik} \hat{x}_{ik} = \sum_{i \in I, k \in K'} q_i d_{ik} x_{ij} \frac{C_k^{\text{ship}}}{\sum_{k' \in K'} C_{k'}^{\text{ship}}} = \sum_{i \in I, k \in K'} q_i d_{ik} x_{ij} \frac{y_{kj} C_k^{\text{ship}}}{\sum_{k' \in K'} y_{kj} C_{k'}^{\text{ship}}},$$

and therefore the new solution's objective matches the original's, which completes the proof. □

By building on the idea behind Proposition 3, we can derive a stronger bound that uses FC pairs, at the expense of a larger and more complex relaxation:

$$\min \sum_{i \in I} \sum_{\substack{k, k' \in K \\ k \leq k'}} \left(\frac{C_k^{\text{ship}} d_{ik} + C_{k'}^{\text{ship}} d_{ik'}}{C_k^{\text{ship}} + C_{k'}^{\text{ship}}} \right) x_{ikk'} \tag{5a}$$

$$\sum_{\substack{k, k' \in K \\ k \leq k'}} x_{ikk'} = 1 \quad \forall i \in I \tag{5b}$$

$$\sum_{k' \in K} y_{kk'} = 1 \quad \forall k \in K \tag{5c}$$

$$\sum_{k \in K} y_{kk} \leq m \tag{5d}$$

$$\sum_{i \in I} q_i x_{ikk'} \leq (C_k^{\text{ship}} + C_{k'}^{\text{ship}}) y_{kk'} \quad \forall k, k' \in K, k < k' \tag{5e}$$

$$\sum_{i \in I} q_i x_{ikk} \leq C_k^{\text{ship}} y_{kk} \quad \forall k \in K \tag{5f}$$

$$x_{ikk'} \leq y_{kk'} \quad \forall i \in I; k, k' \in K, k \leq k' \tag{5g}$$

$$x_{ikk'} \geq 0, y_{kk'} \in \{0,1\} \quad \forall i \in I; k, k' \in K, k \leq k'. \tag{5h}$$

The interpretation for variables and constraints in this formulation is similar to (4). Variable $y_{kk'}$ indicates that FC's $k, k' \in K$ are paired into a region, and y_{kk} indicates that FC k remains unmatched and defines a region by itself. The x variables then define the fractional amount of each tile's demand served by one of the resulting regions.

Proposition 4. The optimum of (5) is a lower bound for (1).

Intuitively, the proof of this bound's validity is similar to the simpler singleton bound (4). In a relaxation with fractional x variables, we can subdivide the FC's assigned to any region into pairs while reducing the cost; if the number of FC's assigned to a region is odd, one of the FC's will be left unmatched and define its own region, as in the previous bound.

Proof. We consider again the relaxation of (1) with objective 1a, subject to 1b, 1c, 1d, $x_{ij} \geq 0 \; \forall i, j$, and $y_{kj} \in \{0,1\} \; \forall k, j$. This time, we increase J so $|J| = \lfloor (\ell - m)/2 \rfloor + m$. If $y_{kj} = 0$ for all $k \in K$ and some fixed $j \in J$, we again define the ratio in 1a to be zero for this j.

Consider any partition of the ℓ FC's into m regions induced by a solution (x, y) of (1). We proceed as in the previous proof, subdividing each region's FC's; however, this time we subdivide into pairs rather than individual FC's. If the number of FC's assigned to a region is odd, one FC will remain unmatched and defines a region as a singleton. This yields $\lfloor (\ell - m)/2 \rfloor + m$ regions after subdividing, of which at most m are defined by singleton FC's, cf. constraint 5d.

The relaxation of (1) just described matches (5) as long as we assume no cluster of three or more FC's will remain in an optimal solution. This part of the proof is identical to the proof of Proposition 3; if the solution includes any such cluster, we can further subdivide it into pairs and at most one singleton. The x variables in the new solution are defined in proportion to the shipping capacities C_k^{ship} of each new region, in an analogous fashion to the previous proof. Suppose the solution includes a region j with a set K' of three or more FC's, where $k, k' \in K'$. If we pair these two FC's, we set $\hat{y}_{kk'} = 1$ and then define

$$\hat{x}_{ikk'} = x_{ij} \frac{C_k^{\text{ship}} + C_{k'}^{\text{ship}}}{\sum_{\varkappa \in K'} C_{\varkappa}^{\text{ship}}},$$

which yields a feasible solution with the same objective value; we omit further details for brevity. □

7. Computational Results

In this section, we first describe the data generation process for creating instances of practical scale that capture real-world features. We then benchmark the quality of solutions generated using the algorithm in Section 5 against the lower bound models described in Section 6, the MILP described in Section 3.1, and the non-linear formulation (1) for these instances. In section 7.3, we evaluate the quality of solutions to find preferable parameter choices and region numbers, based on potential managerial concerns.

7.1. Data Generation

We use 2020 population data from the United States Census Bureau as an approximation of customers' weekly demand during peak periods in units. We randomly sample 150 FC addresses within the US from the repository of retail warehouses contained in CoStar, restricting to square footage greater than 500,000. We generate their latitude and longitude coordinates using the GeoPy package in Python, using the addresses obtained from CoStar. We set a target of total FC shipping capacity that is 10% higher than total demand, and total storage capacity that is six times the total demand. We sample each FC's capacities from a normal distribution with mean set to the average target FC capacity, and standard deviation as 20% of the mean. We generate instances with either 100 or 150 FCs; for instances with 100 FCs, we randomly sub-select them from the 150 FCs, and scale up their capacities such that the total shipping and storage capacity is the same. We use three-digit zip codes (Zip-3) as tiles, and assume uniform demand equal to the average. The assumption of uniform demand is not overly restrictive in practice, as we can always create custom tiles with equal demand. Furthermore, our approach is also applicable when this condition isn't met.

7.2. Comparison of solution quality with benchmarks

We solve the integer programs in steps 1 and 3 of the construction heuristic, and step 6 of the local search in Algorithm 1 with a maximum time limit of 15 minutes and optimality gap of 1%, whichever is achieved first; we set MaxIter = 50 in the local search. We solve linear and integer

Number of FCs (ℓ)	Number of regions (m)	$Q_{\min}^{\text{store}}$ as % of $\sum_k C_k^{\text{store}}$	$Q_{\max}^{\text{ship}}$ as % of $\sum_k C_k^{\text{ship}}$	(4) (x10^9)	(5) (x10^9)	Heuristic Solution (x10^9)	No. of iterations	Run time (hours)	Gap with (4) (%)	Gap with (5) (%)
100	10	8	12	9.69	9.93	16.84	13	2.97	42.44	41.02
100	10	6	14	9.69	9.93	14.97	12	2.64	35.28	33.68
100	10	0.01	100	9.69	9.93	14.74	12	3.74	34.25	32.63
100	20	4	6	9.69	9.83	12.72	9	1.87	23.83	22.74
100	20	3	7	9.69	9.83	12.39	12	2.70	21.77	20.65
100	20	0.01	100	9.69	9.83	11.90	15	4.82	18.53	17.36
100	30	2.67	4	9.69	9.77	11.95	8	1.71	18.88	18.22
100	30	2	4.67	9.69	9.77	11.57	24	5.05	16.27	15.59
100	30	0.01	100	9.69	9.77	11.28	24	7.27	14.07	13.37
100	40	2	3	9.69	9.73	12.19	7	1.00	20.50	20.18
100	40	1.5	3.5	9.69	9.73	11.41	16	2.88	15.03	14.69
100	40	0.01	100	9.69	9.73	10.98	29	6.48	11.75	11.40
150	15	5.33	8	9.43	9.58	13.59	16	12.47	30.60	29.51
150	15	4	9.33	9.43	9.58	13.74	13	7.59	31.35	30.28
150	15	0.01	100	9.43	9.58	13.30	12	10.93	29.10	27.99
150	30	2.67	4	9.43	9.52	11.76	33	16.15	19.77	19.03
150	30	2	4.67	9.43	9.52	11.78	28	18.29	19.95	19.21
150	30	0.01	100	9.43	9.52	11.40	38	33.92	17.23	16.47
150	45	1.78	2.67	9.43	9.48	11.56	15	5.65	18.39	17.98
150	45	1.33	3.11	9.43	9.48	11.24	20	8.60	16.06	15.64
150	45	0.01	100	9.43	9.48	11.04	25	20.63	14.53	14.10
150	60	1.33	2	9.43	9.45	11.47	13	3.87	17.76	17.61
150	60	1	2.33	9.43	9.45	11.14	26	11.33	15.32	15.17
150	60	0.01	100	9.43	9.45	11.19	33	21.19	15.72	15.57

Table 2: Comparison of quality of heuristic solution from Algorithm 1 against lower bounds for varying numbers of facilities, regions, and design parameters.

programs using Xpress 8.14.2 as a solver, implemented in Python 3.9.12. Our machine is a 64-bit x86 Amazon Linux 2 AMI OS with EC2 type m5.24xlarge. The total run time of the heuristic is between one and 33 hours for all instances; as this is a tactical or strategic planning problem, such run times are adequate.

The lower bound models depend on the number of regions, facilities, their capacities and tile demand; they are independent of the design parameters $Q_{\min}^{\text{store}}$ and $Q_{\max}^{\text{ship}}$. Hence, the cost of the singleton lower bound (4) remains constant for instances with the same number of FCs, while the cost of the matching lower bound (5) remains constant for instances with same number of regions and FCs. We set $Q_{\max}^{\text{ship}}$ to be 20% above the average shipping capacity, 40% above, and unbounded. In tandem, we set $Q_{\min}^{\text{store}}$ to 20% below average storage capacity, 40% below, and unbounded, so that instances progressively become less constrained. For the unbounded case, we set $Q_{\min}^{\text{store}} = 0.01\%$ to ensure that every region gets at least one FC. For (5), we solve the problem to an optimality gap of 1% and report the best bound.

Table 2 shows the performance of the solution generated by Algorithm 1 against the lower bounds. Since the cost of the solution decreases as we increase the number of regions, the gaps also decrease. The gap with (4) is between 11.75-42.44%, while the gap with (5) is between 11.40-41.02%, 0.15-1.42% tighter. Additionally, we compared the solution against other benchmarks. The linearized MIP given by equations 1b-1j and 3a-3h results in a trivial lower bound of 0 and an upper bound larger than 45x10^9 for all instances when solved using Xpress 8.14.2 with a maximum time of 48 hours. We also tested Baron on the MINLP formulation (1) for a few instances with a maximum time of 48 hours; however, it also results in weak lower bounds and is unable to find a feasible solution.

To elaborate on the impact that the number of regions has on the cost, take for example the instances with 100 FCs, $Q_{\min}^{\text{store}} = 0.01\%$ and $Q_{\max}^{\text{ship}} = 100\%$; as we increase the number of regions from 10 to 40, the cost decreases by 25%. This intuitively aligns with our results for lower bound models in Section 6. In instances with the same number of facilities, as we increase $Q_{\min}^{\text{store}}$, we naturally observe increasing costs. Moreover, we find that the relative increase in cost from imposing constraints 1e and 1f is only between 2-7% in most of the instances (and between 11-14% for two instances), showing their relatively small effect at these magnitudes, which reflect our motivating application.

7.3. Evaluation of region designs

For the purpose of evaluating region designs, we plot the curve of percentage demand that is covered within a certain distance from fulfillment centers, assuming that the demand of each tile is fulfilled from each FC assigned to a region, in proportion to the shipping capacities of the FCs. We analyze region designs for instances with 100 FCs with 8, 10 and 12 regions, and for instances with 150 FCs with 12, 15 and 18 regions (see Table 3). We choose this range of region numbers because they allow every region's fulfillment center (FC) storage capacities to maintain a minimum required inventory fill rate, which is the percentage of demand that can be satisfied from on-hand inventory, to meet regional demand. Increasing the number of regions beyond this range decreases the fill rate for each region due to a decrease in the region's FC storage capacity. We also vary the capacity bounds for each number of regions, and find that the curves for two different instances for 100 FCs with 10 regions (parameters $Q^{\text{store}}_{\text{min}} = 8\%$ of $\sum_k C^{\text{ship}}_k$; $Q^{\text{ship}}_{\text{max}} = 12\%$ of $\sum_k C^{\text{store}}_k$) and 12 regions (parameters $Q^{\text{store}}_{\text{min}} = 5\%$ of $\sum_k C^{\text{ship}}_k$; $Q^{\text{ship}}_{\text{max}} = 11.67\%$ of $\sum_k C^{\text{store}}_k$) are the farthest apart, and the curves for all other instances with 100 FCs lie between these two curves. The maximum height between these two curves is 7.68% at 348 miles, which means that one design fulfills 7.68% more demand from FCs within 348 miles compared to the other design. This is also roughly the maximum radius for fulfillment of orders within one day in practice [45]. This implies that the impact of choosing an appropriate number of regions and parameters for designing regions can increase the percentage of demand being fulfilled within one day by up to 7.68% (see Figure 4). We also find that if the objective value for one instance is lower than the objective value for another instance, then the curve for the former is above the curve for the latter, which shows that the objective function plays a role in designing regions such that demand is closer to FCs, justifying our choice of the objective function.

Instance	Number of FCs	Number of regions	$Q^{\text{store}}_{\text{min}}$ as % of $\sum_k C^{\text{store}}_k$	$Q^{\text{ship}}_{\text{max}}$ as % of $\sum_k C^{\text{ship}}_k$	Heuristic Solution (x10^9)
0	100	8	11.25	13.75	16.83
1	100	8	10	15	16.84
2	100	8	7.5	17.5	16.94
3	100	10	9	11	16.74
4	100	10	8	12	16.84
5	100	10	6	14	14.97
6	100	12	7.5	9.17	15.09
7	100	12	6.67	10	14.59
8	100	12	5	11.67	14.22
9	150	12	7.5	9.17	15.33
10	150	12	6.67	10	14.88
11	150	12	5	11.67	14.67
12	150	15	6	7.33	14.50
13	150	15	5.33	8	13.59
14	150	15	4	9.33	13.74
15	150	18	5	6.11	13.41
16	150	18	4.44	6.67	13.02
17	150	18	3.33	7.78	12.80

Table 3: Evaluation of specific instances which satisfy practical constraints for inventory pooling.

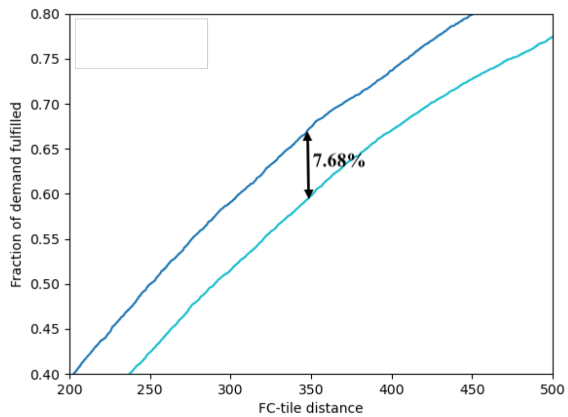


Figure 4: Percentage of demand fulfilled (y-axis) within a certain radius given by x-axis for 100 FCs with 10 regions (bottom curve, instance 4) and 12 regions (top curve, instance 8).

8. Conclusion

We introduced the fulfillment regionalization problem, a graph partitioning problem with demand and supply nodes, which we respectively call tiles and fulfillment centers (FCs). Each set of tiles in the partition (a region) must be contiguous and must have sufficient supply from FCs to fulfill its demand; multiple FCs can be assigned to a region. The objective is to minimize the average distance required to fulfill demand across all regions. When a single FC is assigned to a region and tile demand is uniform, we can achieve contiguity in an optimal solution without requiring it; however, in other cases we must explicitly enforce contiguity, which we do so using shortest-path tree constraints. Empirically, we observe that imposing these constraints results in only a 1-2% loss in the objective function, compared to ignoring contiguity. This shows that the price of contiguity is small, even when further restricted by shortest-path constraints. We proposed a local-search construction and improvement heuristic and benchmarked it against simple and efficient lower bounds. Our experiments demonstrate that choosing appropriate system parameters can lower average fulfillment distance by 16%, resulting in approximately 8% more demand fulfilled within a 350-mile radius, a typical radius for one-day fulfillment.

For future work on this problem, we could improve our lower bounds by considering FC triplets rather than singletons or pairs; however, optimizing such a model could be computationally challenging. It may also be important to study other situations in which contiguity is guaranteed without explicitly requiring it, such as conditions on the geography of FCs assigned to a region. More broadly, it would be useful to study several generalizations of our model, e.g. replacing FC-tile distances with actual middle-mile network distances in the objective, or including FC capacity planning in the region design. Finally, we could consider hierarchical regionalization networks and other situations in which an FC may fulfill demand in multiple regions.

Acknowledgements

The authors are grateful to Benoit Montreuil for providing access to CoStar data, and to Amazon for providing access to appropriate software tools and computing resources. The authors also thank the Modeling and Optimization and Global Transportation teams at Amazon, including Amitabh Sinha for discussion on lower bound model, Louis Faugère for discussion on contiguity, and Jun Xiao for providing adjacency data.

References

[1] Jason Acimovic and Vivek F Farias. "The fulfillment-optimization problem". In: *Operations Research & Management Science in the age of analytics*. INFORMS, 2019, pp. 218–237.

[2] Jason Acimovic and Stephen C Graves. "Making better fulfillment decisions on the fly in an online retail environment". *Manufacturing & Service Operations Management* 17.1 (2015), pp. 34–51.

[3] Fatih Burak Akçay and Maxence Delorme. "Solving the parallel processor scheduling and bin packing problems with contiguity constraints: Mathematical models and computational studies". *European Journal of Operational Research* (2024).

[4] Yoram Almogy and Oded Levin. "A class of fractional programming problems". *Operations Research* 19.1 (1971), pp. 57–67.

[5] Mehdi Amiri-Aref, Walid Klibi, and M Zied Babai. "The multi-sourcing location inventory problem with stochastic demand". *European Journal of Operational Research* 266.1 (2018), pp. 72–87.

[6] Austin Buchanan. "Political districting". In: *Encyclopedia of Optimization*. Springer, 2023, pp. 1–13.

[7] John Gunnar Carlsson, Erik Carlsson, and Raghuveer Devulapalli. "Shadow prices in territory division". *Networks and Spatial Economics* 16 (2016), pp. 893–931.

[8] Felipe Caro, Takeshi Shirabe, Monique Guignard, and Andrés Weintraub. "School redistricting: Embedding GIS tools with integer programming". *Journal of the Operational Research Society* 55.8 (2004), pp. 836–849.

[9] Emilio Carrizosa and Stefan Nickel. "Robust facility location". *Mathematical methods of operations research* 58 (2003), pp. 331–349.

[10] Rodolfo Carvajal, Miguel Constantino, Marcos Goycoolea, Juan Pablo Vielma, and Andrés Weintraub. "Imposing connectivity constraints in forest planning models". *Operations Research* 61.4 (2013), pp. 824–836.

[11] Abraham Charnes and William W Cooper. "Programming with linear fractional functionals". *Naval Research logistics quarterly* 9.3-4 (1962), pp. 181–186.

[12] Thomas J Cova and Richard L Church. "Contiguity constraints for single-region site search problems". *Geographical Analysis* 32.4 (2000), pp. 306–329.

[13] Tingting Cui, Yanfeng Ouyang, and Zuo-Jun Max Shen. "Reliable facility location design under the risk of disruptions". *Operations research* 58.4-part-1 (2010), pp. 998–1011.

[14] Levi DeValve, Yehua Wei, Di Wu, and Rong Yuan. "Understanding the value of fulfillment flexibility in an online retailing environment". *Manufacturing & service operations management* 25.2 (2023), pp. 391–408.

[15] Werner Dinkelbach. "On nonlinear fractional programming". *Management science* 13.7 (1967), pp. 492–498.

[16] Ayoub Foussoul and Vineet Goyal. "Distributionally Robust Newsvendor on a Metric". *arXiv preprint arXiv:2410.12134* (2024).

[17] John Gunnar Carlsson, Xiaoshan Peng, and Ilya O Ryzhov. "Demand Equilibria in Spatial Service Systems". *Manufacturing & Service Operations Management* 26.6 (2024), pp. 2305–2321.

[18] Wes Gurnee and David B Shmoys. "Fairmandering: A column generation heuristic for fairness-optimized political districting". In: *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*. SIAM. 2021, pp. 88–99.

[19] Pavithra Harsha, Shivaram Subramanian, and Joline Uichanco. "Dynamic pricing of omnichannel inventories: Honorable mention—2017 M&SOM practice-based research competition". *Manufacturing & service operations management* 21.1 (2019), pp. 47–65.

[20] Sidney Wayne Hess, JB Weaver, HJ Siegfeldt, JN Whelan, and PA Zitlau. "Nonpartisan political redistricting by computer". *Operations Research* 13.6 (1965), pp. 998–1006.

[21] Informs Resoundingly Human. *2025 Franz Edelman Award finalist: Amazon.* https://resoundinglyhuman.com/episodes/2025-franz-edelman-award-finalist-amazon/. 2025.

[22] Takehiro Ito, Xiao Zhou, and Takao Nishizeki. "Partitioning graphs of supply and demand". *Discrete applied mathematics* 157.12 (2009), pp. 2620–2633.

[23] Stefanus Jasin and Amitabh Sinha. "An LP-based correlated rounding scheme for multi-item ecommerce order fulfillment". *Operations Research* 63.6 (2015), pp. 1336–1351.

[24] William C Jordan and Stephen C Graves. "Principles on the benefits of manufacturing process flexibility". *Management science* 41.4 (1995), pp. 577–594.

[25] Aida Khajavirad and Nikolaos V Sahinidis. "A hybrid LP/NLP paradigm for global optimization relaxations". *Mathematical Programming Computation* 10.3 (2018), pp. 383–421.

[26] Takahito Kuno. "A branch-and-bound algorithm for maximizing the sum of several linear ratios". *Journal of Global optimization* 22.1 (2002), pp. 155–174.

[27] Takahito Kuno. "A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems". *Journal of Global optimization* 33.2 (2005), pp. 215–234.

[28] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha-da-Gama. *Introduction to location science*. Springer, 2019.

[29] Yongzhen Li, Jia Shu, Miao Song, Jiawei Zhang, and Huan Zheng. "Multisourcing supply network design: two-stage chance-constrained model, tractable approximations, and computational results". *INFORMS Journal on Computing* 29.2 (2017), pp. 287–300.

[30] Dániel Marx and Michał Pilipczuk. "Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams". In: *Algorithms-ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*. Springer. 2015, pp. 865–877.

[31] Garth P McCormick. "Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems". *Mathematical programming* 10.1 (1976), pp. 147–175.

[32] Anuj Mehrotra, Ellis L Johnson, and George L Nemhauser. "An optimization based heuristic for political districting". *Management Science* 44.8 (1998), pp. 1100–1114.

[33] Walter Murray and Uday V Shanbhag. "A local relaxation method for nonlinear facility location problems". In: *Multiscale optimization methods and applications*. Springer, 2006, pp. 173–204.

[34] Sean O'Neill. *Sizing down to scale up: How Amazon reworked its fulfillment network to meet customer demand*. https://www.amazon.science/news-and-features/how-amazon-reworked-its-fulfillment-network-to-meet-customer-demand. 2023.

[35] Aysu Ozel, Karen Smilowitz, and Lila Goldstein. "Community-Engaged School District Design: A Stream-Based Approach". *Available at SSRN 4610313* (2023).

[36] Leyla Ozsen, Mark S Daskin, and Collette R Coullard. "Facility location modeling and inventory management with multisourcing". *Transportation Science* 43.4 (2009), pp. 455–472.

[37] Kushal Saha and Subir Bhattacharya. "'Buy online and pick up in-store': Implications for the store inventory". *European Journal of Operational Research* 294.3 (2021), pp. 906–921. DOI: https://doi.org/10.1016/j.ejor.2020.10.006.

[38] Siegfried Schaible and Jianming Shi. "Fractional programming: the sum-of-ratios case". *Optimization Methods and Software* 18.2 (2003), pp. 219–229.

[39] Amanda J. Schmitt, Siyuan Anthony Sun, Lawrence V. Snyder, and Zuo-Jun Max Shen. "Centralization versus decentralization: Risk pooling, risk diversification, and supply chain disruptions". *Omega* 52 (2015), pp. 201–212. DOI: https://doi.org/10.1016/j.omega.2014.06.002.

[40] Takeshi Shirabe. "A model of contiguity for spatial unit allocation". *Geographical Analysis* 37.1 (2005), pp. 2–16.

[41] Takeshi Shirabe. "Districting modeling with exact contiguity constraints". *Environment and Planning B: Planning and Design* 36.6 (2009), pp. 1053–1066.

[42] David B Shmoys, Éva Tardos, and Karen Aardal. "Approximation algorithms for facility location problems". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 1997, pp. 265–274.

[43] Saleh Soltan, Mihalis Yannakakis, and Gil Zussman. "Doubly balanced connected graph partitioning". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2017, pp. 1939–1950.

[44] João C Teixeira and António P Antunes. "A hierarchical location model for public facility planning". *European Journal of Operational Research* 185.1 (2008), pp. 92–104.

[45] UPS. *U.S. Ground Maps*. https://www.ups.com/maps/results?loc=en_US.

[46] Hamidreza Validi, Austin Buchanan, and Eugene Lykhovyd. "Imposing contiguity constraints in political districting models". *Operations Research* 70.2 (2022), pp. 867–892.

[47] Ping Josephine Xu, Russell Allgor, and Stephen C Graves. "Benefits of reevaluating real-time order fulfillment decisions". *Manufacturing & Service Operations Management* 11.2 (2009), pp. 340–355.

[48] Zhishuang Yao, Loo Hay Lee, Wikrom Jaruphongsa, Vicky Tan, and Chen Fei Hui. "Multi-source facility location–allocation and inventory problem". *European Journal of Operational Research* 207.2 (2010), pp. 750–762.

[49] Shixiang Zhu, He Wang, and Yao Xie. "Data-driven optimization for atlanta police-zone design". *INFORMS Journal on Applied Analytics* 52.5 (2022), pp. 412–432.

[50] Andris A Zoltners and Prabhakant Sinha. "Sales territory alignment: A review and model". *Management Science* 29.11 (1983), pp. 1237–1256.

Appendix

Hess' Formulation for Region Centroids

Based on [20], we use the following formulation to generate the centroids of regions that are used in the first step of Algorithm 1. Let $x_{ij} \in \{0,1\}$ indicate the assignment of a tile i to a tile j, if tile j is the centroid of a region; $x_{ii} = 1$ indicates that a tile i is a centroid of a region.

$$\min \sum_{i,j \in I} q_i d_{ij} x_{ij} \tag{6a}$$

$$\text{s.t.} \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \tag{6b}$$

$$\sum_{i \in I} x_{ii} = m \tag{6c}$$

$$x_{ij} \leq x_{jj} \quad \forall i, j \in I \tag{6d}$$

$$\sum_{i \in I} q_i x_{ij} \geq (Q_{\min}^{\text{store}} / \sum_{k \in K} C_k^{\text{store}}) * \sum_{i \in I} q_i \quad \forall j \in I \tag{6e}$$

$$\sum_{i \in I} q_i x_{ij} \leq (Q_{\max}^{\text{ship}} / \sum_{k \in K} C_k^{\text{ship}}) * \sum_{i \in I} q_i \quad \forall j \in I \tag{6f}$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in I \tag{6g}$$

Constraint 6b ensures that every tile is assigned to exactly one region's centroid. Constraint 6c ensures that we have as many centroids as the desired number of regions that we want to create. Constraint 6d ensures that a tile i is assigned to another tile j only if the latter is the centroid of a region. Constraints 6e and 6f restrict the size of a region in terms of its maximum and minimum demand. The objective 6a minimizes the demand-weighted distances between each tile and the centroid of the region to which it is assigned.