IPIC 2021 | 8th International Physical Internet Conference

Generating clusters for urban logistics in hyperconnected networks

Cyrus Hettle[1], Louis Faugere[2], Simon Kwon[3], Swati Gupta[4], and Benoit Montreuil[5]
1, 3, 4, 5: Georgia Institute of Technology, Atlanta, USA
2: Amazon, Seattle, USA
Corresponding author: chettle@gatech.edu

Abstract: *In the hyperconnected logistics model, a city is represented as a continuous mesh of small regions called unit zones. The clustering problem is to partition the set of unit zones into larger local cells and urban areas, and is critical in defining network operations. We give a mixed integer programming-based method for solving the clustering problem, which combines aspects of graph partitioning and min-cost flow problems. Our model aims to minimize expected operating cost, accounting for s expenses throughout the network, while incentivizing clusters that are resilient, geographically compact, and have balanced demand. To generate meaningful warm-starts for our MIP and achieve computational speedups, we adapt a graph partitioning method called striping. Solutions for the clustering problem can be integrated with methods for other problems in hyperconnected network design, significantly improving their tractability. Our techniques work effectively in tandem with methods for choosing hub candidate locations and routing flow. We show the effectiveness of our methods in redesigning SF Express's hyperconnected network in Shenzhen.*

Conference Topic(s): *Interconnected freight transport, logistics and supply networks, Last mile, City logistics, PI Modelling and Simulation*
Keywords: *Logistics clustering, urban logistics, hyperconnected city logistics, last mile, GIS, Physical Internet, logistics space time network, mixed-integer linear programming*

1 Introduction

This paper proposes methods for designing a clustering of small atoms, called unit zones, into the larger pieces which serve as the backbone for hyperconnected intracity logistics, as proposed in the conceptual framework of (Montreuil et al., 2018). We work in a hyperconnected three-tier urban logistics model as depicted in Figure 1. A city is represented as a continuous mesh of unit zones (UZ), possibly of widely varying sizes, geography, population density, etc. Illustratively, an implementation in Shenzhen has unit zones covering in average about 5,000 inhabitants. These unit zones are served by access hubs (AH), where in the hyperconnected model each zone can be served by multiple nearby access hubs and each access hub can serve multiple unit zones, typically being positioned near their intersection. Unit zones are aggregated into disjoint larger spaces called local cells (LC), which in the USA could correspond to the size of a five-digit zip code. Similarly, each local cell is served by multiple larger processing centers called local hubs (LH), each of which can serve multiple local cells. Local cells are in turn aggregated into large spaces called urban areas (UA), served by gateway hubs (GH). We allow flow between any two hubs that are on the same level (horizontal flow) or one level apart (vertical flow).

In the overall network design problem, there are three high-level questions we seek to answer:

1. Where should hubs, of each type, be opened?
2. How should the set of unit zones be partitioned into local cells? How should the set of local cells be partitioned into urban areas?

3. Given the hubs and clusters, how should flow be directed?

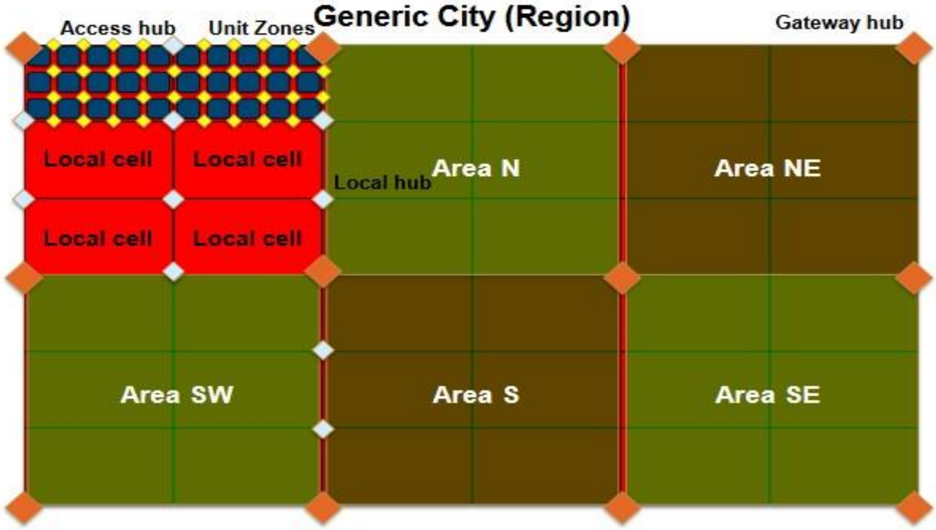

Figure 1: A generic city showing the various levels of region and hub (Montreuil et al., 2018) showing the smallest regions called unit zones, which are clustered into local cells and urban areas.

The clustering problem is to find a good solution to the second question. Such a solution is critical, as in the multi-tier web structure, it is very difficult to design operations without a shape structure. This structuring is analogous to the structuring of space in postal codes, which had been put in place to organize the postal flows yet have been frozen for decades notwithstanding the evolution of demographics and flow patterns, and are now mostly used for political, statistical, and location purposes. We generalize the structure in a wider logistics spectrum, and aim for the clusters to structure hyperconnected operations to evolve over time for optimized impact on logistics efficiency, capability, resiliency, and sustainability.

When performing the clustering, all hub locations are fixed and we approximate flow cost. We employ a robust mixed integer program method, combining aspects of graph partitioning and min-cost flow problems, to obtain a clustering which is expected to have low cost of operation and environmental impact, while enabling fast pickup and delivery and being resilient to network disruptions. We foresee such clusterings being used on two levels: an individual company may use them for its own networks, or a clustering can be used for space structuring for multiple stakeholders in an urban setting, requiring more complex criteria.

In Section 2, we discuss the assumptions and necessary data for our model. Next, in Section 3, we introduce our MIP approach for solving the clustering problem. We discuss its interaction with methods for other aspects of network redesign in Section 4, and give details of implementation in Section 5. Finally, in Section 6, we give an example of this clustering method in work with SF Express, applied to an urban area in northwest Shenzhen, China.

2 Modeling

In this section, we review the assumptions in modeling the city, its demand, and other input for the clustering model. We represent the city as a graph $G = (V, E)$, where the vertex set V is the set of UZ and the edges $e \in E$ represent pairs of adjacent UZ. Most of these adjacency pairs are determined by taking pairs of UZ whose boundaries nearly touch in a GIS map, but some additional edges may be added to connect more isolated parts of the city. We refer the interested reader to Muthukrishnan et al. (2021) on hub candidate selection for additional details of this process.

Unit zones are designed so each can be served by one or a small team of couriers. The size of unit zones can range dramatically to match the density of the urban landscape, from several blocks to a single skyscraper. Designing unit zones is outside the scope of the problems we consider here, but we want an efficient method to produce clusterings that can evolve through time as geography and demand changes require the unit zones to be redesigned. To estimate the operational costs of shipping, we have costs λ_{ij} for each pair of unit zones, and λ_{ih} for each pair of unit zone and hub. These costs may be taken to be simply a straight-line distance, but much more sophisticated measures that make use of city and geographical data are also possible. For instance, this cost may reflect elevation changes, traffic patterns, nearby building type and density, geographical variance of the demand profiles, etc. The other crucial element of the clustering instance is the demand profile. We abstract a demand profile to a single number ϕ_{ij} for each pair i, j of unit zones, representing the demand for commodities from unit zone i to unit zone j. The network structure design problem, which routes the flow more precisely than the clustering model, treats the multi-commodity aspect in more detail.

Note that in the method for determining clustering, described in the mixed integer program in Section 3, we capture the cost of vertical flow between unit zones and local hubs and/or gateway hubs, but do not capture the cost of flow between unit zones and access hubs. Routing this flow, which is carried by the couriers in the hyperconnected framework, is an important aspect of the overall network design problem. Our model can easily be augmented to handle this flow as well, but at significant computational cost given the large number of access hubs in the network. However, the feasibility of any specific UZ-AH flow is not impacted by clustering. Moreover, while the optimal pattern of flow to and from AH for a given UZ might be affected by changing the clustering, any such changes are on a local level, not large and consistent effects throughout the network. Therefore, we approximate the design at the lowest level and consider only higher levels in the web, and assume that UZ-AH assignments are made given the clustering, as in Muthukrishnan et al. (2021).

3 Mixed integer programming model

We propose using a large mixed integer program to solve the clustering problem. This MIP, detailed below, simultaneously produces clusters of unit zones into local cells and local cells into urban areas. Its objective is to minimize operational costs, which we approximate as a weighted sum of flow costs between zones and hubs; expressions for the compactness and balance of the clustering, which affect the efficiency and resiliency of low-level operations not captured in the higher-level flow; and costs for expanding the size of hubs. Horizontal flow is allowed between zones in the same local cell and local cells in the same urban area. Demand is treated on a global level where flow transit to and from a gateway hub are independent processes. Therefore, this model abstracts away some aspects of flow cost, such as the cost of opening different numbers of arcs for flow and the process of flow being aggregated as it moves up the tree towards a gateway hub. Much of this behavior is captured in the network structure design model, whose results can be reincorporated into the clustering model as detailed in Section 5.

3.1 Input and output

As input, the model takes a variety of data about the geography of the region, the unit zone network, and the demand, including a list of n unit zones indexed by i and a list of hubs, indexed by h, with base capacities κ_h and modules with capacities χ_{hm} and prices π_{hm} respectively. For each hub h, the neighborhood $N(h)$ is a set of unit zones within adequate distance to be served by h.

- Unit zone-to-unit zone demand ϕ_{ij}
- Unit zone-to-unit zone distances or costs λ_{ij}
- Unit zone-to-hub distances or costs λ_{ih}

Second, the model has a variety of parameters controlling aspects of the clustering and flow pattern, which can be set and tuned according to the problem instance:

- Maximum number of local cells K
- Maximum number of urban areas K'
- Parameters $hubs_{min}$ ($hubs_{min}'$) denoting the minimum number of LH needed to serve each LC, and GH needed per UA
- Parameters $cluster_{min}$, $cluster_{max}$ denoting the minimum and maximum number of local cells permitted in an urban area
- Parameters ρ^{LH}, ρ^{GH} that limit the proportion of flow between any pair of unit zones i and j that passes through a single local or gateway hub, respectively. Reducing these parameters creates a flow with increased resiliency to localized slowdowns.
- Within-local-cell cost parameter γ^L and within-urban-area cost parameter γ^A, representing the relative cost of horizontal movement compared to vertical movement
- Parameter δ controlling the proportion of the objective devoted to compactness
- Hub capacity thresholds R_l, with associated penalties ρ_l for exceeding them.

As output, the model gives a list of local cells, indexed by $1 \leq k \leq K$; a list of urban areas, indexed by $1 \leq k' \leq K$; and a high-level approximate zone-to-hub and zone-to-zone flows.

3.2 Decision variables and objective

Throughout, we index UZ by i and j, LC by k, and UA by k'. The key decision variables are binary assignment variables x_{ik}^C and $x_{ik}^{A}{}'$. Each x_{ik}^C is 1 if UZ i is in LC k and 0 otherwise. Likewise, $x_{ik'}^A$ is 1 if UZ i is in UA k' and 0 otherwise. Finally, c_{kk}' is 1 if LC k is contained in UA k' and 0 otherwise. Binary e_{ijk}^C are 1 if UZ i and j are both in LC k and 0 otherwise. Likewise $e_{ijk'}^A$ is 1 if UZ i and j are both in UA k' and 0 otherwise.

Flow variables d_{ijh}^{AH}, d_{ijh}^{LH}, d_{ijh}^{GH} are the (nonnegative) quantity of flow from UZ i to UZ j, passing through hub h, where h can be a local or gateway hub. Variables f_{ij}^C and f_{ij}^A are the (nonnegative) flow sent between UZ i and j at the LC level (through unspecified AH, not passing through any LH) and the UA level (through unspecified LH, not passing through any GH.) These capture vertical and horizontal flow, respectively, through the network.

3.3 Objective

The objective OBJ has five components: an estimate of the cost of vertical shipping operations, an estimate of the cost of horizontal shipping operations, a measure of the compactness of the local cells, a measure of the balance of the capacity used at each hub, and (optionally) the cost of opening modules for additional capacity at hubs. The other significant element of the cost of the network is opening and maintaining hubs, which is fixed in the input to the clustering model and which we discuss it in Section 4.

Compactness refers to the shape of LCs: i.e., an LC with a circular shape is more compact than an LC with a more elongated shape. In the hyperconnected framework, operations within

LCs are carried out by riders, who have several fixed routes between UZs in the LCs. In more compact LC, there is a greater choice of efficient routes for riders, creating better resiliency and more flexible operations. Determining these routes is itself a complex problem which this clustering model does not attempt to solve. We quantify compactness (and hence a measure of the corresponding operation cost) by the total pairwise distance between UZs in each LC.

Balance refers to the workload (i.e. capacity demands) on each hub. In operations, it is desirable to keep the workload of the hubs of each type (LH and GH) approximately equal, particularly for resiliency purposes. We quantify this by setting type-dependent thresholds B_l, with associated penalties β_l for exceeding them. If a hub exceeds B_l fraction of its capacity, cost β_l is incurred. Therefore, we express the objective as

$$OBJ = \sum_{i,j,h}(\lambda_{ih} + \lambda_{jh})d_{ijh}^{LH} + \sum_{i,j,h}(\lambda_{ih} + \lambda_{jh})d_{ijh}^{GH} + \sum_{i,j}(\gamma^C f_{ij}^C + \gamma^A f_{ij}^A) \quad (1)$$
$$+ \delta \sum_{i,j,k} \lambda_{ij} e_{ijk}^C + \sum_{l,h} \beta_l b_{lh} + \sum_{h,m} \pi_{hm} z_{hm} ,$$

where the first line of the objective approximates the cost of vertical and horizontal flow, and the second includes a measure of compactness of the local cells, a measure of the balance of the hubs, and the cost of capacity modules added to the hubs.

3.4 Constraints:

We formulate the problem as a mixed integer program, minimizing the objective OBJ while satisfying the following constraints. Local cell and local hub constraints appear in left column, urban area and gateway hub constraints in the right column. Throughout, we index unit zones by i and j, local cells by k, urban areas by k', and local and gateway hubs by h.

Zone-to-cluster assignment constraints

$\sum_k x_{ik}^C = 1 \ \forall i$ (2) $\qquad \sum_{k'} x_{ik'}^A = 1 \ \forall i,$ (3)

$x_{ik}^C, x_{ik}^A \in 0,1$ (4)

Each UZ must be in precisely one LC and precisely one UA.

Tree flow constraints, ensuring contiguity of clusters

We use flow constraints, introduced in Shirabe 2009, to ensure the contiguity of each LC and UA. Each cluster has one of its UZ designated as the root of a tree in the subgraph of G induced by the vertices of the cluster tracked by the decision variables r_{ik}. The choice of root otherwise has no meaning. The remaining constraints force a flow t (abstract and unrelated to the flow of commodities in the network) on the tree. Alternative contiguity constraints may be used, such as the lengthier but stricter separator constraints introduced in Validi et al., 2020.

Multi-level clustering constraints, ensuring proper inclusion

$$cluster_{\min} \le \sum_k c'_{kk} \le cluster_{max} \ \forall k', \quad (5)$$
$$\sum_{k'} c'_{kk} = 1 \ \ \forall k, \quad (6)$$
$$(x_{ik}^C + c'_{kk})/2 \ \ge \ x_{ik'}^A \ \forall i, k, k', \quad (7)$$
$$c_{kk'} \in \{0,1\}. \quad (8)$$

These constraints govern the relationship of LC to UA. The decision variables c'_{kk} track whether LC k is contained in UA k'. Each UA must contain between $cluster_{min}$ and $cluster_{max}$ LCs, and LCs may not overlap multiple UA.

Hub-to-cluster assignment constraints

$a_{hk}^{C} \leq \sum_{i\in N(h)} x_{ik} \quad \forall\, h,k,$ (9)

$\sum_k a_{hk}^{C} \geq hubs_{\min} \forall h,$ (10)

$a_{hk}'^{A} \leq \sum_{i\in N(h)} x_{ik}' \quad \forall\, h,k',$ (11)

$\sum_k a_{hk}'^{A} \geq hubs_{min}' \;\forall\, h,$ (12)

$a_{hk}^{C}, a_{hk}'^{A} \in \{0,1\}.$ (13)

Hubs (both LH and GH) may only serve a cluster (the corresponding decision variable $a_{hk(\prime)}$ is 1) if they are near to it. In particular, the cluster must intersect the hub's neighborhood $N(h)$. Furthermore, each hub must serve at least $hubs_{\min(\prime)}$LC or UA, respectively.

Hub-to-zone assignment constraints

$y_{ikh}^{LH} \leq (x_{ik} + a_{hk}^{C})/2 \quad \forall\, i,k,h,$ (14)

$y_{ih}^{LH} \leq \sum_k y_{ikh}^{LH} \quad \forall i,h,$ (15)

$y_{ik'h}^{GH} \leq (x_{ik'} + a_{hk'}^{A})/2 \;\forall\, i,k',h$ (16)

$y_{ih}^{GH} \leq \sum_{k'} y^{A'_{ikh}} \;\forall\, i,h,$ (17)

$y_{ih}^{AH}, y_{ikh}^{LH}, y_{ih}^{LH}, y_{ik'h}^{GH}, y_{ih}^{GH} \in \{0,1\}.$ (18)

These constraints combine the UZ-to-cluster decision variables x and the hub-to-cluster decision variables a to assign the values of variables y that track whether a UZ i can send flow to a hub h.

Total demand constraints

$\phi_{ij} = f_{ij}^{C} + \sum_h d_{ijh}^{LH} = f_{ij}^{C} + f_{ij}^{A} + \sum_h d_{ijh}^{GH}, \;\forall\, i,j$ (19)

These constraints require that all demand ϕ_{ij} be met. All demand must either be met with horizontal flow within an LC or go to an LH, and likewise must either be met with horizontal flow within a UA or go to a GH.

Total capacity constraints

$\kappa_h + \sum_m z_{hm}\chi_{hm} = c_h \quad \forall\, h,$ (20)

$\sum_{i,j} d_{ijh}^{LH} - b_{lh}\sum_{i,j}\phi_{ij} \leq B_l c_h \;\forall\, l,h,$ (21)

$z_{hm}, b_{lh} \in \{0,1\},\; d_{ijh}^{LH}, d_{ijh}^{GH}, c_h \geq 0.$ (22)

These constraints require that the capacity of all hubs be maintained. The base capacity of a hub h is κ_h, but this can be increased be adding modules (tracked with decision variables Z_{hm}) each with additional capacity χ_{hm}. The thresholds B_l, which determine the balance component of the objective, are checked and b_{lh} is forced to be 1 if the threshold is exceeded.

Horizontal flow constraints

$e_{ijk}^{C} \leq (x_{ik}^{C} + x_{jk}^{C})/2, \;\forall\, i,j,k,$ (23)

$f_{ij}^{C} \leq \sum_k e_{ijk}^{C} \;\forall i,j,$ (24)

$e_{ijk'}^{A} \leq (x_{ik'}^{A} + x_{ik'}^{A})/2, \;\forall\, i,j,k',$ (25)

$f_{ij}^{A} \leq \sum_k e_{ijk'}^{A} \;\forall i,j,$ (26)

$e_{ijk}^{C}, e_{ijk'}^{A}, \in \{0,1\},\; f_{ij}^{C}, f_{ij}^{A} \geq 0.$ (27)

These constraints check the feasibility of horizontal flow. The variables e_{ijk}, which are 1 if both i and j are in cluster k and 0 otherwise, are set using the x_{ik}. Horizontal flow f_{ij} is permitted only if both unit zones are in the same cluster.

Vertical flow constraints

$d_{ijh}^{LH} \leq y_{ih}^{LH} \;\forall\, i,h,$ (28)

$d_{ijh}^{GH} \leq y_{ih}^{GH} \;\forall\, i,h,$ (30)

$d_{ijh}^{LH} \leq \rho^{LH} \cdot \phi_{ij} \ \forall\, i, h,$ (29) $\quad d_{ijh}^{GH} \leq \rho^{GH} \cdot \phi_{ij} \ \forall\, i, h.$ (31)

The first two constraints enforce that flow from a UZ may only be sent to a hub that can serve that UZ (tracked by the decision variable y). We assume that all demand ϕ, and hence all flow d, is scaled to be bounded by 1 (if not, the RHS can be scaled appropriately.) The second two constraints enforce resiliency and robustness in the network. No more than ρ fraction of the flow for a given OD pair can be sent through a particular hub.

4 Integration and interaction with other aspects of network redesign

This clustering model is not a standalone method for network redesign. In this section, we examine how it integrates with methods for solving both hub candidate selection and network structure design problems. Each of these problems addresses one or more difficult computational aspects of the entire redesign problem, and approximates or does not consider the other aspects. The problems complement each other, so that repeatedly iterating between the problems provides a computationally tractable method that addresses all aspects of the high-level network redesign problem. Figure 2 (left) gives an overview of the information each of the three problems outputs and then gives as input to the other problems.

4.1 Hub candidate selection

The hub candidate selection process takes as input a very large set of points in the city (potentially tens of thousands of candidates.) It uses a combination of GIS methods, optimization techniques, and local expertise, incorporating zoning and other factors that may affect hub placement, to reduce the size of this set by roughly an order of magnitude. It both provides initial input to and interacts with the output of the clustering process. Further details, including the shape-based methods for determining candidate feasibility and the programs used to obtain optimal candidate sets, are given in Muthukrishnan et al. (2021). Figure 2 (right) depicts the steps in fine-tuning the candidate list, as well as the methods used.

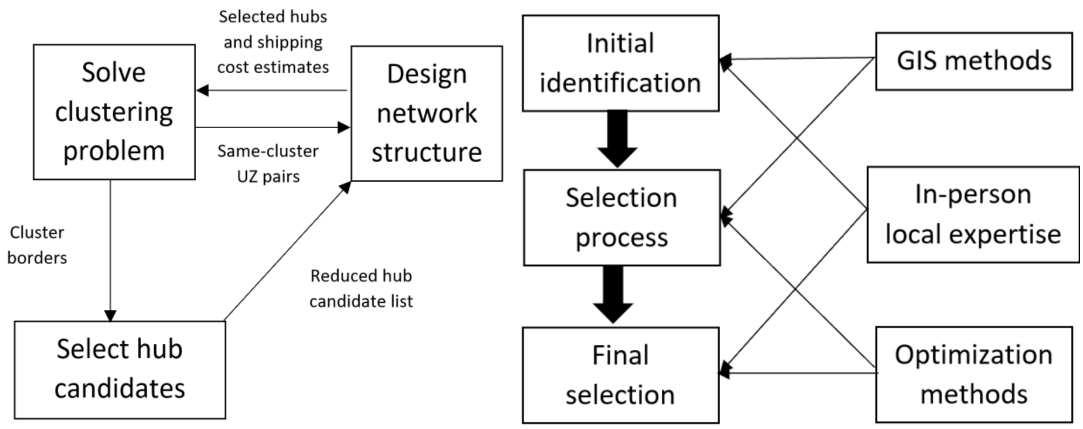

Figure 2: The iteration process and information flow between the methods of the network restructuring process; the steps and techniques used for hub candidate selection

4.2 Network structure design

The network structure design process takes as input a demand profile (modeled as commodities, each with a single origin and destination), a set of hub candidates (with associated costs), and a set of arcs between pairs of hubs, along which flow can be sent (with associated distances and costs.) The objective of structure design is to determine which hubs and which arcs to open and choose a corresponding feasible timed flow pattern, in such a way as to minimize cost. It also includes considerations for specific local requirements, such as varying traffic patterns and legal restrictions on freight transfers. This problem comes with considerable computational challenges in large instances.

While clustering is not a direct input to network structure design, it is implicitly captured in the set of arcs which are chosen out of all possible arcs between pairs of hubs. In the hyperconnected web, only pairs of AH in the same LC may have an arc between them. Therefore, choosing a clustering dramatically reduces the set of available arcs. Furthermore, in combination with the hub candidate selection methods as discussed above, hub candidates far from the boundary of clusters can be discarded. Both of these steps help reduce the size of the structure design instance and aid computational feasibility. In turn, the output of structure design provides a new potential set of hubs, as well as a detailed commodity flow on the graph, when used as input to another iteration of clustering. In particular, the shipping cost terms $d_{ijh}^{LH}\left(\lambda_{ih}+\lambda_{jh}\right)$ and $d_{ijh}^{GH}\left(\lambda_{ih}+\lambda_{jh}\right)$ in the objective of the clustering model are only approximations of the cost incurred in sending a commodity between i and j through hub h. Indeed, most parcels take a multi-hub voyage including AH, LH, and one or more GH, with cost depending on the number and type of vehicles employed on the arcs of that trip, rather than being fixed. Thus, after an iteration of the method for network structuring, which includes in its output the real costs for all arcs used by the plan, in the next iteration of clustering the objective term for all such arcs is replaced by the result from network structuring. This improves the approximation of the objective and allows the clustering to adapt and improve based on its performance.

5 Implementation

In this section, we discuss details of implementation and warm-starting for the MIP approach. This process assumes the following: a graphical model of the city; a demand profile on the city, as in Subsection 2.2; and an existing or starting list of LC and UA assignments for UZ.

5.1 Preprocessing

First, an auxiliary and artificial UZ is created for each GH. These UZ are used only to track intracity demand whose source or destination is a GH. Each such UZ i_g is assigned to its own auxiliary LC k_g, and the constraint $x_{ikg}=0,\ \forall\, i\ \neq\ i_g$ is added to the model for each such i_g, so that the LC consists only of the UZ. These LC exist only for the purpose of tracking flow in the MIP. Next, demand data is broken into intercity and intracity components. All intracity demand (UZ to UZ) becomes a corresponding term ϕ_{ij}. All intercity demand (UZ to GH or GH to UZ) is then assigned to a specific GH based on desired high-level flow patterns in the network. If the city contains multiple GH, this choice may depend on the facilities available at each GH or on the particular source or destination of the flow, so that parcels go to/from a GH that is close to their destination and source. Once this demand is assigned to a GH, it is expressed as a UZ-UZ demand, using the auxiliary UZ constructed for the GH.

5.2 Striping and warm-starting the MIP

To obtain a clustering that can be used to warm-start the MIP, we use the striping path-based algorithm introduced by Hettle et al. (2021). This algorithm takes as input a graph $G=(V,E)$ with vertex weights $w(v)$ for all $v\in V$, a desired number of parts k, a balance parameter ε, and a Hamiltonian path $(v_1,\dots v_n)$ on G. We say that a partition is *consistent* with the Hamiltonian path used as input if each of its parts is a consecutive subpath of $(v_1,\dots v_n)$. Using a dynamic programming framework, the algorithm returns a consistent partition of V into k parts, which correspond to local cells, and each of which is contiguous and is ε-balanced (has total weight within ε fraction of the average). Moreover, the total perimeter of the LCs in this partition is the smallest possible among all possible balanced ε-balanced consistent partitions. We apply this method to the graph G of unit zones, with the weight of each vertex given by its total demand. The choice of path is critical to producing a partition with compact parts, and we construct a Hamiltonian path by applying the uncrossing approximation algorithm for the traveling salesman problem on instances with a Euclidean metric (Van Leeuwen and Schoone, 1981). Even over multiple urban areas, both the path and the striping solution can be computed in under one minute and repeated with different paths and parameters to obtain multiple warm starts. Uncrossing requires at most $O(|V|^3)$ steps, and by using dynamic programming and exploiting the structure of the cut function used to calculate the perimeter objective for compactness, the striping algorithm runs in $O(|V|^2)$ time.

Given the initial warm-start clustering, as well as the auxiliary UZ and LC, we initialize variables x_{ik}, and thus the variables e_{ijk}, a_{hk}, y_{ikh}, and y_{ih}, which depend only on x_{ik}, are also set. Next, we determine the starting values for the variables used in the tree flow constraints for contiguity, for instance by using a breadth-first search tree on the subgraph of G induced by the vertices in each cluster k. Then we determine starting values for the vertical flow variables d_{ijh} and the horizontal flow variables f_{ij}. We assign flow so as to minimize the flow cost terms of the objective, while not yet considering the hub capacity balance term. For each pair (i, j) of unit zones, if the horizontal flow between them is cheaper than the vertical flow, we send all flow horizontally. Otherwise, we send flow vertically by greedily choosing the cheapest paths while respecting the resiliency constraints.

6 Example in southwestern Shenzhen

In this section, we describe an experiment in a large-scale setting, applying our model to a part of the SF Express network in Shenzhen. As input, we take a group of seven LCs in the southwest of Shenzhen, created using the striping method. The demand profile is based on customer behavior and the market share of SF Express, as well as on 1-day delivery times. Over 80% of demand associated with these cells is intercity, going to or from a GH. Therefore, auxiliary UZ are added as described above. For efficiency, some x_{ik} and other decision variables that depend on the x_{ik} were forced to be 0. For instance, for UZ in the easternmost part of the region, corresponding x_{ik} were set to 0 for $k = 1$ (the relatively remote dark blue LC in the southwest). All distances were computed using the length of a geodesic between the two points, with UZs represented by the centroid. Local hub neighborhoods were set to be those UZs whose centroids are within 1000m of the LH. This was sufficient in this instance, where density is still relatively great even in the less populated eastern UZs, but more sophisticated method where the distance changes throughout the city, increasing in less dense areas, are possible. Local hubs had starting capacity 3000 with up to ten optional modules of capacity 250, at cost of 5,000,000 units each.

6.1 Results

The results of applying our clustering model are shown in Figure 3. The model makes significant changes that resulted in more compact and roughly equal-demand clusters. Table 1 shows the breakdown of the various cost components of the objective function for both clusterings. The clustering changes are reflected in a significantly reduced compactness cost, modeling the expected reduction in rider operation costs. The costs of overall flow also decreased slightly, and flow was rerouted to require one fewer module to be added to local cells. Overall, the new clustering reduces the objective value by more than 6%.

Figure 3: The warm-start clustering of local cells (left) and a new proposed clustering (right).

Table 1: The costs of the clusterings in Figure 3.

	Flow cost	Compactness	Balance	Modules	Total cost
Warm-start clustering	4.54×10^8	8.32×10^7	2×10^7	2×10^7	5.77×10^8
New clustering	4.32×10^8	7.62×10^7	2×10^7	1.5×10^7	5.43×10^8

7 Conclusion

The clustering problem in unit zone, local cell, and urban area assignment is critical in designing a logistics hyperconnected web network. We have shown that it can be effectively solved using a mixed integer program which incorporates considerations such as geographic compactness and contiguity, hub demand capacity, and resiliency. Furthermore, the MIP can be effectively warm-started using techniques from graph partitioning, such as striping, and has useful interactions with other problems in hyperconnected logistics network design. Given the large size and manifold considerations of the MIP, computational challenges do remain in very large instances, which further work may improve. Increased use of additional heuristics in concert with the MIP may be able to improve performance by streamlining some aspects of the model or exploiting geographic structure. Further development of the relationship between clustering, network design, and hub candidate selection is also a potentially rich area of study.

References

- Hettle C., S. Zhu, S. Gupta, Y. Xie (2021): Balanced Districting on Grid Graphs with Provable Compactness and Contiguity, https://arxiv.org/abs/2102.05028 (preprint)
- Montreuil B., S. Buckley, L. Faugere, R. Khir, S. Derhami (2018): Urban Parcel Logistics Hub and Network Design: The Impact of Modularity and Hyperconnectivity, Progress in Material Handling Research: 2018, Ed. A. Carrano et al., MHI, Charlotte, NC, U.S.A. https://digitalcommons.georgiasouthern.edu/pmhr_2018/19/
- Muthukrishnan, P., L. Faugere, B. Montreuil (2021): Filtering Potential Sites for Access Hub Deployment in Physical Internet Based Hyperconnected Urban Logistics Network, Proceedings of IPIC 2021 International Physical Internet Conference, 11p.
- Shirabe, T. (2009): Districting modeling with exact contiguity constraints. Environment and Planning B: Planning and Design, 36(6):1053–1066.
- Validi, H., A. Buchanan, E. Lykhovyd (2020): Imposing contiguity constraints in political districting models, http://www.optimization-online.org/DB_HTML/2020/01/7582.html (preprint)
- Van Leeuwen, J., A Schoone (1981). Untangling a traveling salesman tour in the plane. In *Proc. 7th Conf. Graph-theoretic Concepts in Comput. Sci., 1981*.