This repo stores instances of a variant of the Districting Problem motivated by a real-world application of an e-commerce smart logistics company in Vietnam.

Each instance is a connected graph G(V, A) with a set of nodes G distributed in a planar map and a set of edges A. The distance between two nodes is calculated using Euclidean distance. For each polygon, the number of customers and the number of orders are estimated based on historical data. Drivers are available to serve customers within these polygons on a daily basis. Each driver is assigned to a set of polygons, forming an operating cluster. These clusters must be connected, compact, and nonoverlapping. The objective is to determine the optimal assignment of polygons to drivers, which aims to balance the number of customers, orders, and familiarity levels without violating constraints.

1. Input:

1.1. Directory description

The input folder contains 4 types of instances for different experiments.

- S: small instances generated by the method in Ríos - Mercardo (2021) [1]. In their works, the distribution of customer and order quantities are based on historical data from their industrial partner.
- L1: Some instances to assess the impact of unusual customer and order volumes. These instances are generated by the method in [1] with abnormal polygons that have large numbers of customers and orders (about 10% of polygons have a large number of customers and orders)
- L2: Some instances to evaluate the simultaneous impact of abnormal workload and driver familiarity by generating instances with abnormal workload and heterogeneous driver familiarity.
- R: real-life instances collected from our partner.

1.2. Instance description

Each instance is named according to the structure: [S/L1/L2/R]-DUn-Pm-Tτ-i, where n is the number of nodes (polygons) of the graph, m is the number of clusters to be generated (equivalent to the number of drivers), $\tau \in \{0.05, 0.1\}$ and i is the instance index. Each instance file represents the following inputs:

- Line 1: number of nodes n = |G|
- Lines 2--(n+1) (the next n lines): information of each node

[node_index x_coordinate y_coordinate number_of_customers number_of_orders 0 0]

*Line n+2: number of edges L = |A|

*Lines (n+3)--(n+3+l): adjacency list

[from_edge to_edge]

*Line n+4+i:

[number_drivers number_clusters tolerance1 tolerance2 0.05 0.05]

*Lines n+5+i--n+5+i+L: familiarity matrix, each line (n+5+i+L) shows the familiarity degree of driver i with each node

3. Output

The output folder contains 4 types of instances for different experiments (S, L1, L2, R).

An output file is named by input_instance_name-i, where i is the order of running time. In each file, the result values are shown under each description line. For example:

description line		instance_name	algorithm_name	iter_idx	best_objective	running_time		
values		DU10-P2-0	local_search	0	1396.190948	0.260076		
description line		nb_riders	nb_polygons					
values		2	10					
node list assigned to rider		rider 0 = {5, 3, 4, 7, 6}						
description line		best_rider_2_centroid (centroid of rider cluster)	best_rider_2_obj (obj value of rider cluster)	best_rider_2_radius (total distance of rider cluster)	best_rider_2_act_1 (number of customers in rider cluster)	best_rider_2_act_2 (number of orders in rider cluster)	best_rider_2_act_3 (irrelevant)	best_rider_2_area (irrelevant)
values		5	677.37	677.37	13.15	34.47	0	0
node list assigned to rider		rider 1 = {8, 9, 2, 1, 0}						
description line		best_rider_2_centroid (centroid of rider cluster)	best_rider_2_obj (obj value of rider cluster)	best_rider_2_radius (total distance of rider cluster)	best_rider_2_act_1 (number of customers in rider cluster)	best_rider_2_act_2 (number of orders in rider cluster)	best_rider_2_act_3 (irrelevant)	best_rider_2_area (irrelevant)
values		8	718.82	718.82	13.82	36.72	0	0
description line		obj	time	iter	vio			
values		3507.96	0	0	1678.08			

==

Note: Some files store the objective function value (obj), time (time), iteration number (iter) and violation value (vio) when the improvement is found. They are used to visualize the search process of the proposed algorithm.

REF

[1] R. Z. Ríos-Mercado, A. M. Álvarez Socarrás, A. Castrillón, M. C. LópezLocés, A location-allocation-improvement heuristic for districting with multiple-activity balancing constraints and p-median-based dispersion minimization, Computers & Operations Research 126 (2021) 105106