KHALIFA UNIVERSITY

SENIOR RESEARCH PROJECT

An optimization model for designing efficient delivery routes

Author
Ahmed ALY

ID
100045073

Supervisor
Dr. Adriana GABOR

May 10, 2021

Abstract

In this paper we address the territory design problem with the purpose of vehicle routing. Given a road network, the goal is to create a given number of districts to optimize arc routing activities. The territory design problem has applications in various fields such as delivery, political districting, and municipal services. An integer programming model that creates nodes in favor of forming Eulerian subgraphs is studied. The model can be used on tactical planning where demand, dispersion, distance, and parity are considered. Computational studies were performed to test the effect of parity constraints, number and location of depots, and balancing constraints on the objective function value. Road networks comprised of up to 40 nodes, 68 edges, and 995 integer variables were solved successfully.

1 Introduction

In the current digital age, E-commerce has become an increasingly viable and essential factor of trading goods and services. Consumers are now able to acquire what they want from the comfort of their homes. Due to that, a certain infrastructure is needed to deliver such goods from the producer to the consumer. Thus, delivery companies are important to sustain such activities. Moreover, the scale of deliveries is ever increasing with companies delivering products to consumers from one country to another.

Delivery companies strive to decide on efficient routes for their delivery drivers. Various factors play a role in deciding which routes are efficient depending on the desired goals such as minimizing cost or distance traveled.

Delivery companies such as Aaramex, deliver to an entire city like Dubai by dividing it into several smaller territories. Each territory is then assigned a certain number of drivers to fulfill the deliveries to customers. Since the demand in each territory is often stable, drivers get assigned to territories that they usually deliver to. In general, it is beneficial to not change the territories often. This is due to various factors such as the drivers being familiar with the territory they're assigned to and hence are more efficient in their deliveries.

In this paper, we study a Territory Design Problem (**TDP**) that would help delivery companies decide on efficient routes. The goal is to partition a road network into several districts. This problem has various applications in areas such as delivery, political districting, and municipal services.

The paper is organized in the following manner. In section 2, we discuss related literature and highlight relevant work. In section 3, we describe the given problem. In section 4, we provide the Mixed Integer (**MI**) model for the TDP. Section 5 discusses our findings. Section 6 illustrates limitations to existing research and section 7 presents our conclusions and further research.

2 Literature review

The Vehicle Routing Problem *VRP* is a classic combinatorial optimization problem (Laporte, 2009). It aims to find an optimal set of routes for a set of vehicles in order to deliver to a set of customers in a certain territory. The Travelling Salesman Problem *TSP* is a variation of the VRP where it aims to find an optimal route for one vehicle delivering to a set of customers. This has applications in postal services, delivery services.

The Territory Design Problem *(TDP)* has been present since the 1960s. A given region can be represented on a graph $G = (V, E)$ as a network of *nodes* (V) and *edges* (E) joining the nodes with each other. The most common practice is dividing the region into a set of basic units *BU* where each BU is treated as a node of the graph G. The goal of the TDP is to partition a given region into districts such that certain criteria are satisfied. Clustering is one of many optimization approaches used for territory design. Clustering is done by treating each BU as a district and then iteratively merging BUs to form larger and bigger territories

TDP models often aim to find districts that satisfy the following criteria:

- **Connectivity:**
  A territory is considered connected if the BUs comprising it do not intersect with BUs in other territories

- **Compactness:**
  A territory is said to be compact if the BUs are not dispersed.

- **Balance:**
  Territories are balanced if they have an equal activity measure in each territory. Activities can be an equal number of BUs in each territory, work time, distance travelled, etc...

Most work on TDP has been focused on minimizing the *dispersion* function of created districts. The dispersion function is a measure that ensures the compactness of the created districts. One way to model dispersion is to compute the sum of the squared Euclidean distances from the center of the district to a finite set of selected BUs.

(Angélica Salazar-Aguilar et al., 2010) developed a bi-objective model for TDP considering connectivity and balancing constraints along with utilizing the ϵ-constraint model. The ϵ-constraint model allows for certain constraints such as balancing constraints to be violated by a number ϵ. The model was able to solve instances up to 150 basic units and 6 territories.

We previously discussed the importance of dividing a region into smaller districts with certain criteria. One of the methods to do so is using a meta-heuristic algorithm such as Greedy Randomized Adaptive Search Procedure (**GRASP**) and Path Relinking (**PR**)

GRASP is an iterative model that has two phases; construction and local search. The construction phase tries to build a feasible solution while the local search tries to improve it. This process is repeated until a feasible solution is found or if certain conditions are met. Path relinking is a method that explores solutions generated by using elements from other solutions. The main idea of finding a path is realized by the process of creating linear combinations of a certain set of solutions. Thus, this method finds solutions that share similar characteristics with each other.

(Ríos-Mercado and Escalante, 2015) developed a more robust dispersion measure to solve TDP that relies on the diameter of the district instead of its center. They developed a GRASP algorithm with Path Relinking for Commercial Territory Design Problem. They were able to solve instances of up to 500 basic units and 10 territories.

The reactive GRASP is a variation of the GRASP heuristic. It comprises of three main phases; Construction, adjustment, and local search. The construction phase builds an initial solution which does not have to be feasible. The adjustment phase makes sure the constraints are not violated by a certain tolerance parameter. Finally, the local search phase attempts to improve upon the solution such as improving the value of the objective function.

(Ríos-Mercado, 2016) expanded on the work of (Angélica Salazar-Aguilar et al., 2010). They used the reactive GRASP heuristic where they addressed large scale instances that closely resembled real-world cases. The model takes into account the connectivity constraints in the construction phase of the algorithm which guarantees that the connectivity is always kept. The model was able to solve instances for 1000-2000 basic units.

(García-Ayala et al., 2016) attempted to solve the TDP by using arcs instead of the conventional node-based partitioning models. The article attempts to solve the problem of district design with the implementation of arc routing activities. The model is innovative due to the various differences in mathematical structure compared to node-based partitioning models. While most TDP literature include the intuitive measure of compactness, there seems to be few that address *Deadhead* which refers to the distance travelled where no service is performed. The model focuses on TDP with the purpose of vehicle routing. The model has the criteria of connectivity, compactness, deadhead distance, and workload balance. The model proposed by the authors takes Deadhead into account in the strategic modeling phase by introducing parity constraints. Parity was defined as the constraint that penalizes arcs that create a certain number of nodes of odd degree in every subgraph. This constraint will restrict the models to be close to a Eulerian graph, which will in turn reduce deadhead by creating more efficient routes. This model is particularly useful for tactical planning and parameters can be adjusted for the operational phase to account for workload balance on a day to day basis. The model was able to solve networks with up to 401 nodes and 764 edges.

(Ríos-Mercado and Bard, 2019) had a different approach to TDP where instead of minimizing the dispersion function, they maximized it. They solved a Maximum Dispersion Territory Design Problem *(MaxD-TDP)*. The model utilizes a special upper bound constraint that bounds the linearized dispersion objective function. The constraint is based on the assumption that every territory has at least two basic units. They then implement an exact optimization algorithm that searches for the optimal furthest distance d^* over a set $\{d^1, d^2, ..., d^n\}$ with a fixed lower bound L and upper bound U. If the bounds are valid for the given problem then there exists a solution for the objective function $z(L, U) = d^*$. If not, then the algorithm can reduce the search space in the pre-processing phase, practically eliminating many binary variables. Their tests have shown promising results where they were able to solve instances of up to 800 basic units.

3 Problem Description

Given a road map consisting of roads, street-crossings, the distances between any two crossings, and a set of depots, we need to assign a set of roads to each depot. This is done such that a dispersion function, expressed as the total distance assigned to each depot, is minimized.

The project is concerned with dividing a region into a set of districts. These districts will then be used for vehicle routing activities.

4 Mixed Integer Model for TDP

We propose a Mixed Integer Programming *(MIP)* model for TDP. The objective function of the model is to minimize the dispersion function which equivalent to maximizing compactness (García-Ayala et al., 2016). The dispersion function consists of the distances to and from each edge to its assigned depot. Furthermore, the model takes into account workload balance, connectivity, and parity as follows:

- Workload balance: The average workload assigned to a district should be between a lower-bound and upper-bound.

- Connectivity: Each edge is connected to a depot through a path of edges that are also assigned to the same depot.
- Parity: Setting an upper limit on the total number of nodes of an odd degree in each district.

Given an undirected planar graph $G = (V, E)$ we model a road network where each edge $e = (i, j) \in E$ corresponds to a road or street and each node $v \in V$ is a street-crossing or a dead-end. Each edge e has a length ℓ_e and demand d_E. Let $P \subset V$ denote a given subset of k depots where depots are denoted $1, 2, \ldots, k$. Each district associated with depot $p \in P$ is referred to as district p. Then, we describe the problem as finding k-partitions of edges $E = (E_1, \ldots, E_k)$ such that for each $p \in P$ the district $G_p = (V(E_p), E_p)$ meets certain planning criteria. $V(E_p)$ denotes a set of nodes that are incident to at least one edge in E_p. We define $\sigma(e)$ as the set of edges adjacent to an edge $e \in E$. We denote the set of edges adjacent to node $i \in V$ by $\delta(i)$. Furthermore, for any subset $S \in E$, $\sigma(S)$ denotes the cut set of S, that is, the set of edges with one end point in $V(S)$, the set of nodes associated to edges in S, and the other end point in $V \setminus V(S)$.

The model uses the following sets:

- E to represent the set of roads or streets.
- V to represent street-crossings or dead-ends.
- $V^e \in V$ with $i \in V^0$ to be the set of even degree nodes.
- $V^0 \in V$ to be the set of odd degree nodes.
- $P \subset V$ be the subset of k depots.
- $\sigma(S)$ is the *cut set* of S.

The model uses the following known parameters:

- l_e is the length of edge $e \in E$.
- b_{pe} is the minimum distance from depot p to edge e, defined as $min\{f_{pi}, f_{pj}\}$ is the shortest-path distance in G between nodes $i, j \in V$ relative to the length vector ℓ.

- $\widetilde{D} = \sum\limits_{e \in E} \frac{d_e}{|P|}$ is the average demand per district.
- $\tau_1 \in (0, 1)$ is the tolerance for demand balance constraints.
- $\tau_2 \in (0, 1)$ is the tolerance for parity constraints.
- M is a sufficiently large number.

We define the following variables to be used in the model:

- x_{pe}: binary variable equal to 1 if edge $e \in E$ is assigned to depot $p \in P$, 0 otherwise;
- w_{pi}: binary variable equal to 1 if node $i \in V$ is incident to an edge assigned to depot $p \in P$, 0 otherwise;
- z^0_{ip}: binary variable equal to 1 if degree of node $i \in V$ in district $p \in P$ is odd, 0 otherwise;
- z_{ip}: auxiliary variables that relate z^0_{ip} to the degree of node $i \in V$;
- r_i: binary variable equal to 1 if node i "loses" parity, 0 otherwise.

Node $i \in V^e$ is said to "lose parity" if there is at least one district involving i where the degree of node i in that district is odd. A node $i \in V^0$ is said "to lose parity" if there are at least two districts that contain an arbitrary node i where the degree of node i in each of those districts is odd.

The following model is proposed:

$$\text{minimize} \quad g(x) = \sum_{p \in P} \sum_{e \in E} b_{pe} x_{pe} \tag{1}$$

subject to

$$\sum_{p \in P} x_{pe} = 1 \qquad e \in E, \tag{2}$$

$$\sum_{s \in \sigma(S)} x_{ps} - \sum_{s \in S} x_{ps} \geq x_{pe} - |S| \qquad p \in P, e \in E, S \subset E \setminus \sigma(e), \tag{3}$$

$$\sum_{e \in E} d_e x_{pe} \leq \widetilde{D}(1 + \tau_1) \qquad p \in P, \tag{4}$$

$$\sum_{e \in E} d_e x_{pe} \geq \widetilde{D}(1-\tau_1) \qquad p \in P, \tag{5}$$

$$\sum_{e \in \delta(i)} x_{pe} \leq M w_{pi} \qquad p \in P, i \in V, \tag{6}$$

$$w_{pi} \leq \sum_{e \in \delta(i)} x_{pe} \qquad p \in P, i \in V, \tag{7}$$

$$\sum_{e \in \delta(i)} x_{pe} = 2z_{ip} + z^0_{ip} \qquad p \in P, i \in V, \tag{8}$$

$$r_i \leq \sum_{p \in P} z^0_{ip} \qquad i \in V^e, \tag{9}$$

$$|P| r_i \geq \sum_{p \in P} z^0_{ip} \qquad i \in V^e, \tag{10}$$

$$r_i \leq \sum_{p \in P} z^0_{ip} - 1 \qquad i \in V^0, \tag{11}$$

$$|P| r_i \geq \sum_{p \in P} z^0_{ip} - 1 \qquad i \in V^0, \tag{12}$$

$$\frac{1}{|V|} \sum_{i \in V} r_i \leq \tau_2 \quad , \tag{13}$$

$$w_{pi}, x_{pe}, z^0_{ip}, r_i \in \{0, 1\} \qquad p \in P, i, j \in V, e \in E, \tag{14}$$

$$z_{ip} \in \mathbb{N} \cup \{0\} \qquad p \in P, i \in V \tag{15}$$

The objective function (1) accounts for the compactness of districts by measuring the dispersion as a sum of distances to and from each edge $e \in E$ to their respective depot $p \in P$. Constraints (2) force each edge e to be assigned to exactly one depot p.

The purpose of constraints (3) is to ensure district connectivity. This constraint ensures that each territory is a connected subgraph of G. First we will explain the subsets used in the constraint. Let $S \subset E \setminus \sigma(e)$. As explained earlier, $\sigma(e)$ denotes the set of edges that are adjacent to edge $e \in E$. Hence, the set S denotes a subset whose edges are not adjacent to edge e in district p. For example, for edge $(4, 6)$ in Figure 1, $\sigma(e) = \{(3, 6), (2, 4)\}$. It follows that, S is any subset of the set $\{(1, 3), (2, 3), (2, 5), (2, 7), (1, 5)\}$. So, the constraints are applied on all possible subsets of $S \subset E \setminus \sigma(e)$ (i.e. A power set). Note that this gives $2^5|P|$ constraints for only edge $(4, 6)$.

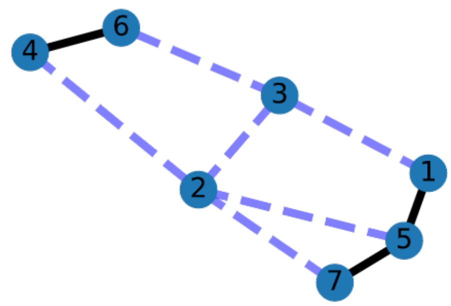


Figure 1: Example graph

The second set to be explained is the cut set $\sigma(S)$ associated with a set S. For any subset $S \subset E$, $\sigma(S)$ is the cut set of S. Let $V(S)$ be the set of nodes associated with the set S. Then $\sigma(S)$, is the set of edges with one node in $V \setminus V(S)$ and one node in $V(S)$. For example, take the subset $S = \{(2,3),(1,3)\}$ in Figure 1. In this case, $V(S) = \{1,2,3\}$ and $V \setminus V(S) = \{4,5,6,7\}$. Using this information we see that $\sigma(S) = \{(1,5),(2,4),(2,5),(2,7)\}$.

Now, we will explain the usage of these sets in relation to constraints (3). We require that the constraint is applied to an edge e that is assigned to a certain district p (i.e. $x_{pe} = 1$). Otherwise, if edge e is not assigned to p (i.e. $x_{pe} = 0$), the constraint becomes redundant. Furthermore, if there is at least one edge $s \in S$ that is not assigned to district p (i.e. $x_{ps} = 0$). Then,

$$\sum_{s \in S} x_{ps} \leq |S| - 1$$

Since

$$\sum_{s \in \sigma(S)} x_{ps} \geq 0$$

we have that

$$\sum_{s \in \sigma(S)} x_{ps} - \sum_{s \in S} x_{ps} \geq 0 - (|S| - 1) = 1 - |S|$$

.

and the constraint becomes redundant too. Therefore, constraints (3) become non-redundant only when all edges in S are assigned to district p. In this case,

$$\sum_{s \in \sigma(S)} x_{ps} \geq 1$$

That is, there must be at least one edge $s \in \sigma(S)$ that is assigned to a district p as well. This rationale is repeated for elements in the set $S \cup \{s\}$, resulting in a territory connected to edge e in district p. There is an exponential number of constraints (3) mainly due to the set S.

Constraints (4) and (5) ensure the balance of the districts within the tolerance τ_1. Where balance refers to the demand of each district. Constraints (6) and (7) identify nodes involved with each district p, where M is the largest node degree in the graph. This means that these two constraints ensure than an edge e is assigned to a depot p if and only if its two incident nodes are assigned to the same depot. Constraints (8) set the degree of node i in district p, and ensure $z^0_{ip} = 1$ when node i has odd degree. A node $i \in V$ in district $p \in P$ is odd if it if is connected to an odd number $2n + 1$ of edges e.

Constraints (9)-(13) are used to limit the imparity gain. Constraints (9) and (10) decide the parity of each node as follows. For each node i of even degree, $r_i = 0$ if and only if $\sum_{p \in P} z^0_{ip} = 0$. This means that in each district involving node i, the degree of the node must be even to keep its parity. This is important since it will lead to partitions that are closer to a Eulerian graph. This in turn will lead to more efficient vehicle routing, thus reducing the deadhead distance. Furthermore, constraints (11) and (12) control the imparity gain relationship for each odd degree node i. Constraints (13) ensures that the percentage of nodes with lost imparity is limited by the tolerance parameter τ_2. Finally, constraints (14) and (15) provide the nature of each decision variable.

5 Computational Experiments

The proposed MIP model was coded in Python 3.7.4 and solved using Gurobi 9.0.3 on an HP Pavilion 15 Intel Core i7-6500U CPU. The program was running on Win 10 (1909) with 12 GB of RAM. For our experiments, we generated 11 road networks. All the road networks follow a grid-like pattern as shown in Figure 2. From henceforth, the graph $G = (V, E)$ acts as the road networks labeled as **RN**. We will reference instances and their corresponding experiment throughout the rest of the document with the

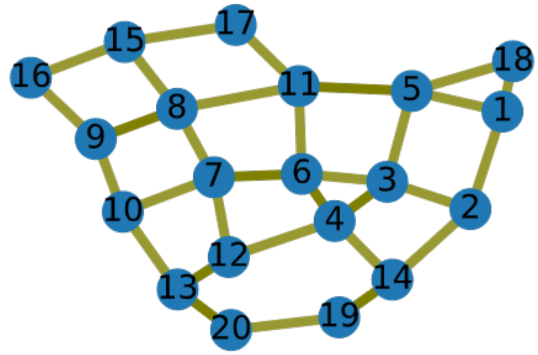


Figure 2: Road Network Pattern

following notation. Instance RN-01-3-05-10 refers to road network RN-01, where the suffix "-3-05-10" means $p = 3$, $\tau_1 = 0.05$ and $\tau_2 = 0.10$. Table-1 displays the networks used in the tests where the last column provides the number of integer variables (**NIV**) for each network.

We ran multiple experiments on the data varying some parameters in order to understand their influence on the data. Road networks of up to 40 nodes, 68 edges, and 995 integer variables were solved successfully. There was no computational time limit set nor was there an optimality gap specified. This is because we did not encounter computational issues or long running times while running the program for the relaxed model..

We ran our tests for different values of the parameters p, $\tau_1 = (0.01, 1)$, and $\tau_2 = (0.01, 1)$. In total, there were 180 trials ran on the program. Of those trails, 179 were solved using the relaxed model. This means that an optimal solution with no disconnections has been found without the need for the connectivity constraints (Constraints 3). This mainly happens due to the selection of depots. It is reasonable to assume that the depots are dispersed throughout the graph. Due to the dispersion minimization function, the program favors connectivity around the depots which ends up reducing the overall dispersion. The single instance that required the use of the original model has been selected in order to demonstrate how that program uses the third constraint. We were able to obtain optimal solutions for 177 out 180 trial runs where three trials yielded infeasible solutions. We summarize the relevant results in the following sub-sections.

Table 1: Data instances

Instance	$\|N\|$	$\|E\|$	$\|P\|$	NIV
RN-01-2	20	31	2	202
RN-01-3	20	31	3	315
RN-02-3	25	42	3	376
RN-02-4	25	42	4	493
RN-03-2	15	21	2	149
RN-03-3	15	21	3	216
RN-04-2	40	68	2	422
RN-04-3	40	68	3	613
RN-04-4	40	68	4	804
RN-04-5	40	68	5	995
RN-05-2	10	12	2	94

5.1 Effect of parity constraints

We ran tests on all 11 road networks where we varied τ_2 between $(0.01, 1.0)$ in order to see the effect of parity constraints on the objective function value. We assumed that by tightening the parity parameter τ_2, the value of the objective function will increase. This was a reasonable assumption since the tighter the constraint controlled by τ_2, the smaller the feasible region would become. We showcase the results in Table 2 using a subset of the instances we ran. The last column shows percentage difference (**PD**) of the objective function value between $\tau_2 = 0.01$ and $\tau_2 = 1.0$. For all instances, the parameter $\tau_1 = 0.05$. As we can see, the objective function value indeed increases as the parity constraint is tightened. The average PD of all cases was 11.74%. The worst case was in instance RN-03-3 at 44.78%. The average PD without instance RN-03-3 is 7.62%. We conjecture that the larger the graph, the lower the PD since instances for the graph RN-04 had the lowest PD at an average of 2.64%. This suggests that for larger graphs with a grid-like structure, the parity constraints do not seem to have much influence over the objective function value.

We will now examine the results in a graphical manner. We consider two trials for the instances RN-02-4-05 in Figure 3. This instance features a graph G with 25 nodes, 42 edges, and four depots. The model of this instance has 493 integer variables. We will use these instances to demonstrate how the parity constraint affects the optimal solution graphically.

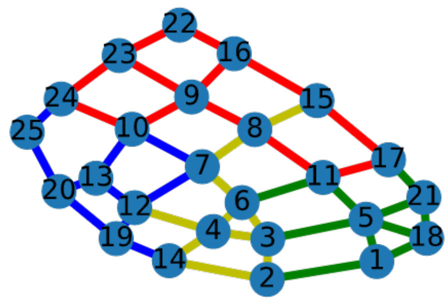
Table 2: Effect of τ_2 on objective function values

Instance	$\|P\|$	$\tau_2 = 1.0$	0.5	0.2	0.1	0.05	0.01	PD(%)
RN-01-2	2	110	114	128	128	128	128	16.36
RN-01-3	3	115	162	172	172	172	172	6.17
RN-02-3	3	155	159	162	162	162	162	4.52
RN-02-4	4	141	145	157	157	157	157	11.35
RN-03-2	2	96	99	110	110	110	110	14.58
RN-03-3	3	67	70	97	97	97	97	44.78
RN-04-2A	2	576	580	585	585	585	585	1.56
RN-04-2B	2	508	510	513	513	513	513	0.98
RN-04-3	3	315	317	332	332	332	332	5.4

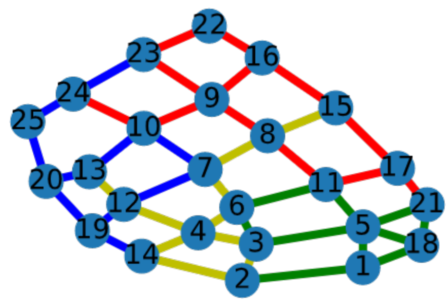
The instance RN-02-4-05-20 in Figure 3(a) used a parity $\tau_2 = 0.5$, which is the same as relaxing the parity constraints. The instance RN-02-4-05-05 in Figure 3(b) used a parity $\tau_2 = 0.05$. Upon solving both instances, we see that the instance in Figure 3(a) has $\sum_{i \in V}^{10} r_i = 3$. This means that 3 nodes lost parity; $V = \{3, 6, 12\}$, that is, they either had an even degree in the original graph but are now odd in Figure 3(a), or they had an odd degree in original graph and are now of odd degree in more than one district. In contrast, the instance in Figure 3(b) has $\sum_{i \in V}^{10} r_i = 0$, meaning none of the nodes lost parity and only gained it.

It is worth noting that one must be careful when choosing the values for parameters τ_1 and τ_2 due to their relation. Bad choices could lead to infeasibility. The parameter τ_2 has a lower bound determined by τ_1 (i.e. $\tau_1 \leq \tau_2$). If τ_2 is set below that, the problem becomes infeasible. The parameter τ_1 should also not be removed (i.e. $\tau_1 = 1$) since that means we can set $\tau_2 = 0$. If that happens, the model could assign all the edges of the graph to one district where there would be no parity loss. Furthermore, the balance tolerance also depends the nature of the graph such as the number of edges, districts, and most importantly, the demand associated with each edge.

We recommend setting $\tau_2 = 0.2$ since, from our research, all other $\tau_2 < 0.2$ yield the same objective function values and all values $\tau_2 > 0.2$ yield results that are feasible but not optimal.



(a) Parity tolerance $\tau_2 = 50\%$



(b) Parity tolerance $\tau_2 = 5\%$

Figure 3: Effect of parity tolerance on instances RN-02-4-05

5.2 Effects of balance constraints

In this section we will examine the effect of changing the balance tolerance. For this purpose, we will illustrate the effects by using instance RN-04-4. Figure 4 presents the results for this road network by using 10 values for the parameter τ_1. The parameter ranges from high ($\tau_1 = 1$) on the left to low ($\tau_1 = 0.01$) on the right in Figure 4. The parameter starts at $\tau_1 = 1$, which is equivalent to completely removing the constraints and moves to a relaxed parameter $\tau_1 = 0.2$ to a strict at $\tau_1 = 0.01$. Note how we did not present a balance tolerance of $\tau_1 = 0$ since that would be infeasible.

As expected, the more we tighten the value for the parameter τ_1, the more the objective function value increases. We found no relation between tightening the constraint to the solution time. Parity loss keeps fluctuating as the value for τ_1 changes but there seems to be no correlation.

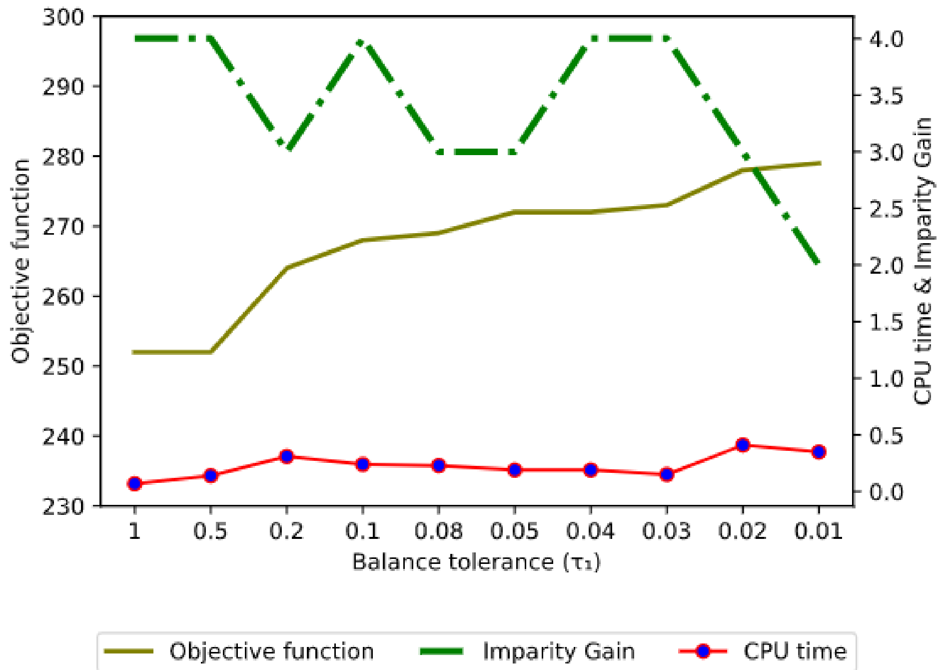


Figure 4: Effects of balance constraints

5.3 Effect of the number and placement of depots

In this section we examine the effects of the number and location of depots on the solution time. In order to do this, we used a total of seven instances.

The instances were all created from RN-04. The network has 40 nodes and 68 edges. Instances RN-04-2A and RN-04-2B all have two differently placed depots and instance RN-04-3 has three depots. Instances RN-04-4A, RN-04-4B, RN-04-4C, RN-04-4D all have four differently placed depots. Finally, instance RN-04-5 has five depots.

Figure 5 features lists the depots for the network RN-04. It plots the CPU time versus the number of integer variables. We can notice that the number of integer variables in the model increases as the number of depots increases. Furthermore, the computation time is increases as the number of integer variables increases. Some instances however, do not conform with

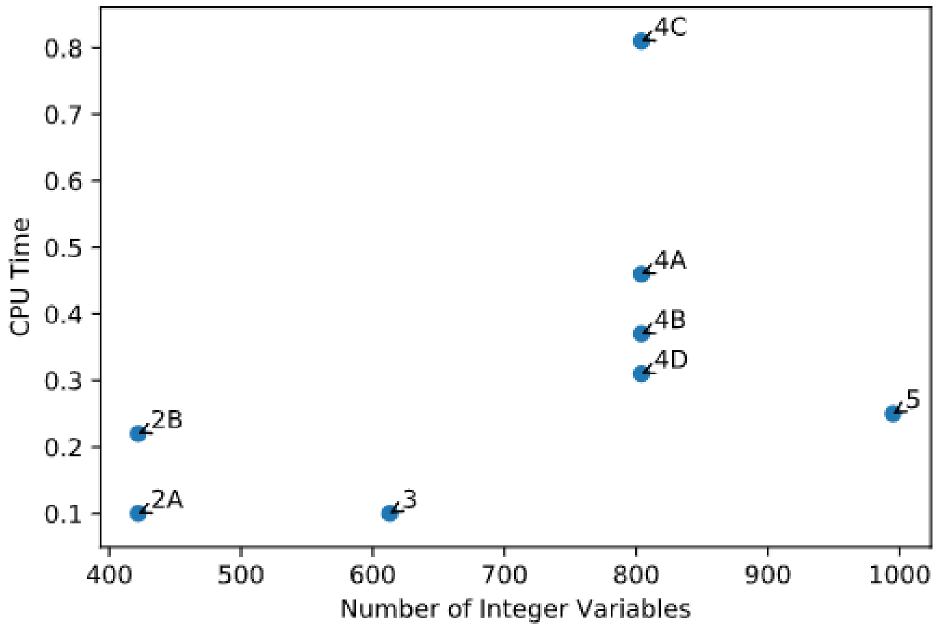
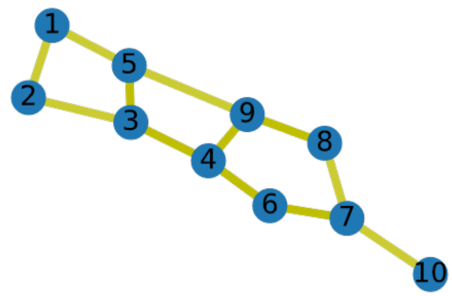


Figure 5: CPU time and number of integer variables for instance RN-04 (40 nodes, 68 Edges)

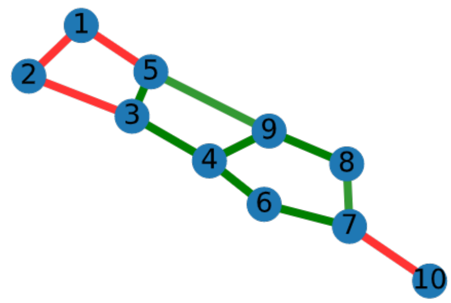
this analysis due to one main reason. We noticed that the more dispersed the depots are, the less the computation time. This implies that the location of the depots plays an important role in computational efforts.

In the case of instance RN-04-4C, we can notice that the solution time is the longest by almost 100%. This is due to placing all the depots extremely close to each other. This further emphasises the importance of the position of depots in relation to computation time.

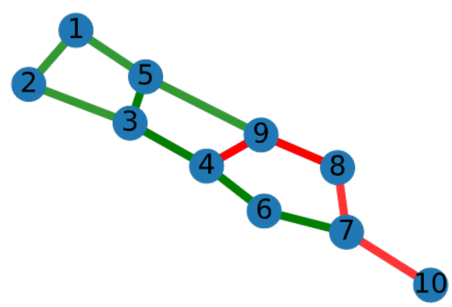
Throughout the computational experiments, we emphasised the selection of depots that are placed far away from each other. This is mainly tied to realism since this is more accurate for real-world situations.



(a) Original Graph of RN-05



(b) RN-05-2 Solved with the relaxed model



(c) RN-05-2 Solved with connectivity constraints 3

Figure 6: Effect of connectivity constraints on district partitioning

5.4 Computations of instances with disconnected districts

For the experiments done in this subsection, we placed the depots in an atypical manner where the depots were placed closed to each other. We did this in order to force the relaxed model without constraints (3) to produce disconnected districts. This was done with the purpose of illustrating how the algorithm would work.

The algorithm works in the following manner:

1. Solve the model without connectivity constraints (3)
2. Identify districts with disconnected edges
3. Identify the edges for the disconnected districts.
4. Solve the relaxed model by adding constraints for the disconnected edges.
5. Identify districts with disconnected edges
6. Repeat until no disconnected districts are found

We will now present the example used to illustrate the connectivity constraints. We selected road network RN-05-2 featuring 10 nodes, 12 edges, and 94 integer variables. We set the parameters $\tau_1 = 0.2$ and $\tau_1 = 0.05$. Figure 6(a) shows the graphical representation of the road network RN-05-2.

We now needed to select two depots that were very close to each other in order to force the creation of disconnected districts. Thus, we selected two depots $p = \{7, 10\}$ that were connected via the same edge $(7, 10)$. The district associated with depot $p = 7$ is represented by the color green and the district associated with depot depot $p = 10$ is represented with the color red.

Figure 6(b) shows the districts created by solving the relaxed model and Figure 6(c) shows the districts created by solving the original model with connectivity constraints 3. Note how the partitioning in Figure 6(b) is disconnected and thus infeasible for the original model. The disconnected components in the solution using the relaxed model were edges

$\{(1,2),(2,3),(1,5)\}$. Thus, we solved the road network once more using by adding constraints for the disconnected edges to the relaxed model until no disconnected districts were found as shown in Figure 6(c).

The optimal solution using the relaxed model in Figure 6(b) was $g(x) = 144$ and the optimal solution using connectivity constraints (3) was also $g(x) = 144$. This shows extremely good results where the objective function value did not increase despite changing the district partitioning for the better.

The computation time using the relaxed model was 0.3 seconds while the computation time using the model with constraints (3) was 1.01 seconds. Note how the solution time using the second model was two times higher than the relaxed model. This suggests that for larger graphs, the computation time will not be good. Indeed, for just edge $(7,10)$ alone, there are $2^9|P|$ connectivity constraints (3).

6 Limitations of existing research

The presented MIP is not able to handle very large graphs. We have experimented with road networks of up to 995 integer variables. But, the number of variables rises exponentially as the graph gets larger. Hence, it would be inefficient to solve large graphs using this MIP.

Another limitation is that while we assume that the higher the parity tolerance τ_2, the lower the deadhead distance, we do not have empirical evidence to support the assumption.

A limitation that has been noticed in all the papers is the lack of a powerful model for both the tactical and operational phase where the models are usually focused on only one with minor adjustments to account for the other.

7 Conclusions and Further Research

In this paper, we studied a territory design model designed with routing problems in mind. The model creates territories that are connected, achieve demand balance, lowers the dispersion, and limits parity loss. The model can be used to plan delivery districts by dividing a given region into several optimized districts for the purpose of arc routing activities. The model is mainly used on tactical level planning. The parameters can be easily adjusted to account for demand or work balance and the parity can be changed to adjust the distance.

For computation experiments, we considered a variety of instances with different parameters. Road networks with up to 40 nodes, 68 edges, and 995 integer variables have been solved. We illustrated how the balance tolerance, parity tolerance, and number and location of depots effect the CPU time, objective function value, and imparity gain.

In most of our instances, the optimal solution with and without the connectivity constraint is the same. This is mainly due to the relative dispersion of the depots throughout the network. The main concern of this model is the distance of the edges from their depots. This promotes the compactness of districts. If a disconnection is found, we can implement the connectivity constraints to maintain the requirements of the original model. The more depots available, provided that dispersion is taken into account, the lower the dispersion function value is.

In this paper, we minimized the dispersion function, taking into account balance and parity constraints. An extension of this work would be to study the deadhead distance and prove the hypothesis, that is, the distance decreases the lower the parity. Furthermore, it would save time if the relaxed model is used and constraints (3) are added only when necessary via an automated process. Another consideration would be to account for directed graphs and see how they change the model. This would be more reflective of real-world cases where some roads only allow traversal in one direction. A possible area of research would be to minimize the deadhead distance subject to a dispersion constraint.

A complicated but possible area of research is finding a model that combines both the tactical and operational phase into a single model. This could be done by creating a model that takes the design and routing decisions simultaneously.

Finally, a promising area of research is developing heuristics that find a good solution in a relatively fast time frame for large networks. This MIP program runs in reasonable time for 200 nodes. A mega-city however has around 10,000 nodes. We could divide the city into smaller regions and use the MIP program to assign streets to depots in those smaller regions. Once those solutions are obtained, they could be used to construct a solution for the whole city. The solution could then be further improved upon by using local search to reassign streets to more appropriate depots. While the proposed MIP might not be efficient for mega-cities, it can be used to create initial solutions that could be further optimized.

References

Angélica Salazar-Aguilar, M., Ríos-Mercado, R. Z., & Luis González-Velarde, J. (2010). A bi-objective programming model for designing compact and balanced territories in commercial districting. *Transportation Research*, *19C*, 885–895. https://doi.org/10.1016/j.trc.2010.09.011

García-Ayala, G., González-Velarde, J. L., Ríos-Mercado, R. Z., & Fernández, E. (2016). A novel model for arc territory design: Promoting eulerian districts. *International Transactions in Operational Research*, *23*, 433–458. https://doi.org/10.1111/itor.12219

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, *43(4)*, 408–416. https://doi.org/10.1287/trsc.1090.0301

Ríos-Mercado, R. Z. (2016). Assessing a metaheuristic for large-scale commercial districting. *Cybernetics and Systems*, *47*, 321–338. https://doi.org/10.1080/01969722.2016.1182361

Ríos-Mercado, R. Z., & Bard, J. F. (2019). An exact algorithm for designing optimal districts in the collection of waste electric and electronic equipment through an improved reformulation. *European Journal of Operational Research*, *276*, 259–271. https://doi.org/10.1016/j.ejor.2018.12.030

Ríos-Mercado, R. Z., & Escalante, H. J. (2015). Grasp with path relinking for commercial districting. *Expert Systems With Applications*, *44*, 102–113. https://doi.org/10.1016/j.eswa.2015.09.019