CHAPTER 11

A NEW GLOCAL (GLOBAL+LOCAL) STRATEGY
FOR FINDING PRACTICALLY VIABLE MAIL
DISTRIBUTION PATH

11.0 INTRODUCTION

The postman / mail delivery agent generally distributes mail by traveling on a vehicle especially when the mail is to be distributed in localities of metropolitan cities and semi-urban areas. For distribution of mail the postman parks his vehicle at a central/ convenient location and reaches delivery points by moving on foot, particularly when there are a couple of delivery points in close proximity. This strategy of distribution of mail is referred to as traverse-park-distribute strategy in this thesis. The amount of distance traveled by the postman using this strategy is generally more than the distance traveled when the postman moves from one delivery point to another without parking his vehicle at a central location. Although the distance traversed is more, this strategy is practically feasible as it emulates the approach adopted by the postman. Hence there is a need to develop computational strategies for finding a practically viable path for distribution of mail for the postman who employs traverse-park-distribute strategy. This requirement adds one more constraint to the RPP:DP problem described in chapter 9. We refer to the problem of finding the practically optimal path for distribution of mail using the traverse-park-distribute strategy as Rural Postman Problem: Delivery Points with Parking (RPP:DPP). We call this as practically optimal, because having decided upon traverse-park-distribute policy for distribution of mail; the path is optimal within the scope of such a strategy and the strategy itself is practically applicable in real life.

We did not find in literature, any reference to works that automatically generate such an optimal path for distributing mail to a set of delivery points. The postal routing software described in [W-16;W-17;W-18] finds optimal route for distribution of postal mail, with

---

A Paper entitled "A GLOCAL (GLOBAL+LOCAL) Strategy for Optimization of Postal Mail Distribution Path" is under preparation
interactively selected parking locations, for countries such as Canada and Finland. In this work we propose a GIS based GLOCAL approach for solving the RPP:DPP problem for postman beats in unstructured localities (ie the problem of automatically finding the best path for distribution of postal mail using traverse-park-distribute strategy) often found in Indian scenario.

The solution to RPP:DPP involves finding clusters of the active delivery points (delivery points that receive mail) which are in close proximity, identifying a cluster center or vehicle parking location close to such a center, and subsequently computing a practically viable global traversal path along with local tours for distribution of mail in the identified clusters. We have devised new intelligent computational strategies employing techniques drawn from cluster analysis, genetic algorithms and graph transformation for finding the solution to RPP: DPP. As the traversal path is generated in two stages namely finding of global and local paths, this approach is described as a GLOCAL (GLOBAL + LOCAL) approach.

The most important step of the GLOCAL approach for automatically finding a practically viable path for distribution of postal mail using the traverse-park-distribute strategy is the grouping of active mail delivery points which are in close spatial proximity. In this work one of the efficient algorithms for spatial clustering namely GDBSCAN [W-20] is modified and adapted for clustering of mail delivery points, which are near to each other. A new computational strategy is devised to find the cluster center which is at a least distance from all the other delivery points in the cluster and further for identification of the parking position using the GIS information of the postman beat.

Once the cluster centers and hence the parking positions for the postman’s vehicle are found, an optimal local tour, that starts and ends at the cluster center/ parking position is generated for each cluster. A new steady state genetic algorithm is devised for the purpose. Further a practically optimal global path to traverse through all the cluster centers and individual mail delivery points that are not grouped into any cluster is generated by employing the computational strategies described in chapter 10. The global tour and the local tours are then combined to generate the overall traversal path for distribution of mail.
A computational strategy is devised for generation of mail distribution schedule that specifies the order and time for delivery of the mail.

This chapter describes computational strategies developed to find the practically feasible traversal path and distribution schedule for delivery of postal mail using traverse-park-distribute strategy. The remaining part of the chapter is divided into eight sections. Section 11.1 describes the overall GLOCAL strategy for generation of practically optimal path for distribution of mail using traverse-park-distribute criterion. Section 11.2 introduces the cluster analysis techniques and presents the adaptation of GDBSCAN algorithm for clustering of mail delivery points. Section 11.3 discusses the genetic algorithm employed for finding the optimal local tour for distribution of postal mail in a cluster. Section 11.4 describes the generation of GLOBAL practically optimal path for distribution of postal mail. Section 11.5 presents the computational procedure for generation of mail distribution schedule. The generation of practically optimal/viable path for distribution and mail distribution schedule for a typical postman beat is illustrated in section 11.6. Section 11.7 discusses the experimental analysis and section 11.8 gives the conclusions.

11.1 THE GLOCAL STRATEGY TO FIND A PRACTICALLY OPTIMAL PATH FOR MAIL DISTRIBUTION

The practically optimal path for distribution of postal mail using the traverse-park-distribute strategy (RPP:DPP) requires finding a GLOBAL traversal path for visiting all the clusters and the individual mail delivery points and also a LOCAL optimal path for distribution of mail in clusters. We refer to the collection of strategies for finding the GLOBAL and LOCAL optimal paths as a GLOCAL strategy. This section provides an overview of the GLOCAL strategy for finding a solution to RPP:DPP (Rural Postman Problem: Delivery Points with Parking). The computational strategies employed for finding the practically optimal path for solving the RPP:DPP problem make use of the GIS representation of the beat of the postman (PGIS-DATA), the spatial data structure (SDSB-PB) and the list of active mail delivery points (the delivery points which receive mail on a given day). The traversal path for distribution of mail is found by first generating an intermediate mail distribution graph (iMDG), which consists of a set of edges that have one
or more active mail delivery points. This graph \((iMDG)\) is further processed to find clusters of mail delivery points by grouping the active delivery points in close proximity and belonging to one connected component of \(iMDG\). A new computational strategy is employed in sequel to find the cluster center for identifying the location to park the vehicle of the postman. Further an optimal local tour is found for distributing mail in each cluster using a newly devised genetic algorithm. Once the local tour is found in each cluster, all the mail delivery points in a cluster are replaced by the cluster center and a new list of delivery points to be visited starting from the Mail Delivery Post Office (MDPO) is formed. This new list of active delivery points consists of cluster centers and individual mail delivery points which could not be grouped into any cluster. These delivery points and cluster centers are now processed to find the mail distribution graph (MDG) as described in chapter 10. Further the process described in chapter 10 finds the global traversal path starting at MDPO, visiting all the delivery points and cluster centers and returning to MDPO. The global traversal path and local tours together give the practically optimal path for distributing postal mail using the traverse-park-distribute criterion. The complete process for the generation of the practically optimal path is outlined as algorithm 11.1.

**Algorithm 11.1 : Generation of Practically Optimal Path for Distribution of Postal Mail using the traverse-park-distribute criterion (GLOCAL approach for optimization of mail distribution path)**

**Input:** The GIS data representation of the postman beat (PGIS-DATA), the set of active delivery points (ADP), Spatial data structure of postman beat (SDS-PB)

**Output:** The practically optimal path for distribution of postal mail (PP), Mail Distribution Schedule (Dist_Sched)

1. Generate an initial mail distribution graph \(iMDG\) consisting of all the edges that have active mail delivery points
2. Find the connected components in \(iMDG\) and label them
3. Find if the mail delivery points in a connected component form one or more clusters (ie they are in close proximity)
4. If there are clusters in the active mail delivery points then group such mail delivery points into one cluster
5. Find the central point of the cluster that is at a least distance from all the other delivery points and check whether vehicle can be parked if not identify a another mail delivery point or parking lot where the postman's vehicle can be parked
6. Find optimal local tours in each cluster starting and ending at the parking position using a steady state genetic algorithm

7. Replace the mail delivery points of the clusters by the corresponding cluster center to generate a new set of active points which are to be visited to distribute the postal mail. An optimal global path that starts and ends at MDPO node and visits all the points in the new set of active points is to be found.

8. Generate a new mail distribution graph MDG that has two classes of edges namely edges that need complete traversal and edges that require partial traversal using algorithm 10.1

9. Generate a connected graph for the mail distribution graph MDG using SDS-PB and algorithm 10.2

10. Generate an Euler graph for the connected graph generated in step 9 using algorithm 10.3 employing the heuristic genetic algorithm discussed in section 10.3.1 (though the EDMONDS algorithm can be used it is not considered in this work)

11. Find the Euler Trail in the Euler graph by using Fleury's algorithm described in chapter 9 to find the practically optimal traversal path for the postman (PP)

12. Find the mail distribution schedule by referring to the euler trail, the local tours in the clusters, the complete and partial traversal of edges (Dist_Sched)

The traversal path found using algorithm 11.1 is generally longer than the one generated using the process described in chapter 10 and also the computation time is more. But this algorithm produces a more practically viable path that takes into account the traversal of postman using two different modes of transport namely vehicle and foot. The computational strategies employed in this process are described in the following sections. Section 11.2 presents the adaptation of GDBSCAN algorithm for clustering of the mail delivery points.

11.2 CLUSTERING OF MAIL DELIVERY POINTS IN CLOSE PROXIMITY

The mail delivery points can be automatically clustered or grouped by employing the techniques of cluster analysis. Clustering or cluster analysis is the process of learning information from data for the purpose of grouping and is the most important unsupervised learning problem. Clustering can be defined as "the process of organizing unlabeled objects into groups made up of similar objects" (Gose et al, 1997). The similarity between objects is measured by a distance measure; two or more objects belong to the same cluster if they are "close" according to the predefined distance measure. This is called distance-based clustering. Another kind of clustering is conceptual clustering; two or more objects
belong to the same cluster if both adhere to a common concept. The goal of clustering or cluster analysis is to determine the intrinsic grouping in a set of unlabeled data. K-Means clustering, K-Medoid clustering, Fuzzy C-Means clustering, ISODATA algorithms are some of the popular clustering algorithms (Gose et al, 1997). Clustering algorithms find applications in many fields such as, Marketing, Biology, Libraries, Insurance, City-planning, Earthquake studies, WWW- document classification etc. City planning and other utility planning activities are facilitated by application of cluster analysis algorithms to spatial data.

Spatial data is generally voluminous and the general cluster analysis algorithms need to be specialized to cluster spatial data. Zhang et al, (1996) present a Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) methodology for clustering of large spatial data sets. The BIRCH methodology produces the best quality cluster of incoming multi-dimensional metric data points using a single scan. The clustering quality is improved by additional scans. Sheikholeslami et al, (1998) present a highly time efficient wavelet transform based algorithm, WaveCluster for clustering of very large spatial data points with arbitrary shape of the spread of points. Wang et al, (1997) present a hierarchical Statistical Information Grid (STING) approach to spatial data mining for very large spatial data sets. They further proved that this method is better than the previous approaches especially for large data sets. Gärtner et al, (2002) propose different method for clustering spatially close and thematically similar data points. Ankerst et al, (1999) present a new algorithm, OPTICS for creating an augmented ordering of a database of points representing its density based clustering structure. Ester et al, (1995) present an R* tree based methodology for spatial clustering in large databases. Ester et al, (1996) present a new clustering algorithm for spatial databases, DBSCAN based on density based notion of clusters to identify arbitrarily shaped clusters. The DBSCAN algorithm takes only one input parameter and guides the user in selecting the parameter number of points in a given area. Yue et al, (2004) present a modification to the DBSCAN algorithm and discuss the use of a greedy strategy to index the clustering space. They also propose a single threshold that can distinguish correctly all clusters in a large spatial dataset. A generalization of
DBSCAN algorithm (GDBSCAN) is presented in [W-20], the article also presents different applications of GDBSCAN to multidimensional data.

The GDBSCAN algorithm is adapted in this work to cluster the mail delivery points in close proximity. The distance between mail delivery points is computed as the minimum on road distance and used for clustering. The distance between any two MDPS is computed based on the road on which the two MDP’s are present. If the two MDP’s are on the same road then the distance between the two MDP’s is the absolute value of the difference in distance of the MDP’s from end point-1. If the two mail delivery points, between which the distance is to be found, are on separate roads, then the minimum distance path amongst the several possible paths is selected and this distance gives the distance between the two mail delivery points. The minimum distance computation between two-mail delivery points on separate roads is depicted in Figure 11.1.

Figure 11.1: Computation of the Distance Between Two MDPS of a Cluster on Different Roads

The distance between the mail delivery points is used for finding the cluster by using the empirically defined density parameters, namely the maximum radius of the neighborhood and minimum number of points in the neighborhood. The complete procedure for clustering of mail delivery points is outlined as algorithm 11.2
Algorithm 11.2: Clustering of Spatially Close Mail Delivery Points

**Input:** List of Active Mail Delivery Points (ADP), Number of Active Mail Delivery Points (NA), GIS-Data for the Postman beat (PGIS-DATA), Density Parameter (area indicated by the radius of neighborhood (rad), minimum number of points in a cluster (minMDP)), Mail Distribution Graph (MDG)

**Output:** Cluster formed according to input parameters and the remaining individual mail delivery points, which are not grouped, into any cluster (CLUSTER)

1. CLNO=1
2. For index=1 to NA do
   a. Find the least cost distance on the road between index MDP and all the other MDP's. Assume that the mail delivery points in different components of MDG are at infinite distance
   b. Find the MDPs, which are less than rad, distance from the index MDP (considering the distance of traversal on the road) and list them in pcluster
   c. If the number of delivery points in pcluster are less than minMDP, then the index MDP is not part of any cluster and is a delivery point which is to be individually visited (CLUSTER(index)=-1)
   d. If the number of delivery points in pcluster are greater than or equal to minMDP then, for all points in pcluster find the active mail delivery points within rad distance (ie find all the points in a neighborhood of radius, rad) and mark all such delivery points to belong to the cluster CLNO (CLUSTER(j)=CLNO, for all j where j MDP belongs to the cluster)
   e. CLNO=CLNO+1
3. End For
4. The variable CLUSTER specifies for every MDP, the cluster to which the MDP belongs, if the value of CLUSTER(i) is -1 then MDP does not belong to any cluster and it is to be individually visited

Once the clusters of mail delivery points are found a center point of the cluster for parking of the postman's vehicle is to be determined. The mail delivery point, which is at a minimum distance from all the other mail delivery points, is designated as the cluster center. The attributes of the designated mail delivery point are further verified to check whether the vehicle can be parked in front of the delivery point, if so the designated mail delivery point is the cluster center/parking location. If parking of the vehicle is not possible in front of the designated mail delivery point then the nearest mail delivery point to the designated center that offers parking facility is selected as the cluster center/parking location. If no delivery point in the cluster offers parking facility then a parking lot in the vicinity is added to the cluster and is made the cluster center/parking location of the postman's vehicle. The complete procedure for finding the parking location/cluster center in each cluster is described in algorithm 11.3. The procedure also generates a cost
adjacency matrix of the graph representing the cluster of mail delivery points. The graph representing the cluster of mail delivery points is a complete graph.

Algorithm 11.3: Location for parking of the postman’s vehicle (Cluster Center) and cost adjacency matrix of the complete graph representing the cluster

**Input:** The active delivery points in a cluster, Number of active delivery points in a cluster (NDP), GIS-DATA of the postman beat

**Output:** The parking position for postman’s vehicle (Cluster_Center), cost adjacency matrix of the complete graph of mail delivery points in the cluster (Cluster_Cost)

1. Find distance between all the pairs of mail delivery points (MDP’s) and construct the cost adjacency matrix Cluster_Cost
2. For i = 1 to NDP do
   a. Find the sum of distances of the i\textsuperscript{th} MDP to all other MDP’s, say SumDist
   b. CumDist(i) = SumDist
3. End For
4. Find the Mail Delivery Point with the minimum Cumulative distance, say P=Min(CumDist)
5. If the P\textsuperscript{th} delivery Point does not have parking facility identify the nearest point with parking facility as the Cluster_Center otherwise the P\textsuperscript{th} delivery point is the Cluster_Center
6. If none of the mail delivery points offer parking facility then parking lot if any in the vicinity is added to the cluster and is made the cluster center, the cost adjacency matrix Cluster_Cost is modified to include the parking lot added.
7. The Cluster_Center indicates the parking position for the vehicle of the postman

The cluster centers and mail delivery points, which are not grouped in any cluster, are the points, which are to be necessarily visited by the postman. These points are treated as new set of mail delivery points and a new mail distribution graph is generated, which is further processed as described in chapter 10 for finding the global path for postman traversal from MDPO to all the MDP’s and cluster centers and back to MDPO, this process is described in section 11.4. Once the postman reaches the cluster center he employs an optimal path for distribution of mail to all the mail delivery points in the cluster. The optimal path for distribution of mail in the cluster is found as described in section 11.3.
11.3 OPTIMAL LOCAL TOUR IN A CLUSTER EMPLOYING GENETIC ALGORITHM

The optimal path for distribution of postal mail in a cluster of mail delivery points starting and ending at a cluster center is a least cost path that visits all the mail delivery points in the cluster. The least cost path that visits all the mail delivery points in a cluster can be found by representing the mail delivery points in the cluster as nodes of a complete graph. A complete graph is a graph, which has an edge connecting every pair of vertices in the graph (Deo, 1973). The optimal path for distribution of mail in a cluster can then be generated, by finding the least cost path that visits all the nodes of a complete graph representing the mail delivery points in a cluster. A path that visits all the nodes of a graph is referred to as a Hamiltonian path (Deo, 1973). The problem of finding the least cost Hamiltonian path in a graph is popularly referred to as the Traveling Salesman Problem.

The traveling salesman problem is a NP complete problem. There are a few exact procedures using branch and bound and dynamic programming techniques for finding the Hamiltonian path for a limited number of vertices (Deo, 1973; Horowitz et al, 1998). More time efficient heuristic procedures are found in literature for solving the traveling salesman problem that finds the optimal Hamiltonian path in a graph. Ravikumar (1999) presents different approaches using pattern recognition techniques for solving the traveling salesman problem. Seuranen (2003) proposes a simple genetic algorithm approach for solving the traveling salesman problem.

In this work a heuristic genetic algorithm based procedure is devised to find the optimal path that visits all the mail delivery points in a cluster. The genetic algorithm employed in this work uses the chromosome structure described in (Seuranen, 2003) and a steady state GA procedure. The GA uses newly defined cross over and mutation operators for evolution of the population.

The structure of the chromosome is depicted in Figure 11.2. Each chromosome has one gene. Each gene has two components, one of the components describes the traversal path in the cluster and the other gives the path distance. The path is specified as a sequence of
mail delivery points to be visited. The path starts and ends at the cluster center. The
distance parameter gives the total length of the distribution path for delivery of mail in the
cluster, starting and ending at the cluster center.

![GENE](image)

\[
\text{Probable Path Cumulative Distance of the path}
\]

Where,
- \( k \) = Number of active delivery points in cluster
- \( V_c \) is the cluster center and \( V_2 \) to \( V_k \) are the
other delivery points in the cluster

**Figure 11.2:** The Chromosome Structure Representing the Solution of Optimal Path for Distribution of Mail in a Cluster

The fitness of the chromosome amongst the population of chromosomes is computed as the
reciprocal of the distance parameter multiplied by a constant value. The fitness of the
chromosome is computed as depicted in equation 11.1.

\[
\text{FitnessOfChromosome} = \frac{\text{FitnessConstant}}{D} \quad \ldots(11.1)
\]

Where,

- **FitnessConstant** is empirically assumed to be 10000
- \( D \) is the cumulative distance of the path for distribution of mail in cluster

The alternate solutions represented by the chromosomes are generated by randomly
choosing the sequence of mail delivery points in the cluster. The total number of different
possible sequences for distribution of mail amongst \( k \) mail delivery points with a pre-
defined cluster center is \((k-1)!\). The GA randomly generates about 20\% the maximum
possible solutions in the form of chromosomes as initial population (Subject to a maximum
limit of 150 chromosomes). This initial population is subjected to evolution using the cross over and mutation operators.

A high probability cross over operator generates two offspring by performing crossover between parent chromosomes, which are selected using the Roulette Wheel Selection. The crossover operator randomly selects a cross over position and copies the second part of the second parent to the second part of the first child and second part of the first parent to the second part of second child. The first child is completed by copying the vertices from the first part of the first parent in such a way that there are no repetitions of the nodes. If all the slots in the path being constructed for the first child are not filled then the remaining slots are filled using the vertices not considered in the offspring being constructed. The first part of the second offspring is also completed in a similar manner. The new cumulative distance is computed for the paths in the two offspring and stored in the respective fields.

Thus the cross over operator produces two offspring with different fitness. The cross over operation is depicted in Figure 11.3 for chromosomes representing the path in a cluster of 5 mail delivery points.

Cross Over Position = 3rd Mail Delivery Point in the sequence

![Cross Over Operator Diagram](attachment:image.png)

**Figure 11.3: The Cross Over Operator for Evolution of Chromosome Population**

The offspring produced by the cross over operator are subjected to a low probability mutation operation to introduce variety to the new solutions/ chromosomes produced. The mutation operator involves selecting two random positions in the chromosome and interchanging the vertices/nodes in the sequence, thus producing a new sequence and
hence a new chromosome. The new cumulative distance of the path is also computed. The mutation operator employed in generation of a new chromosome is illustrated in Figure 11.4

Mutation Positions, 3rd and 5th Vertices in Sequence

\[ ((V_c, V_2, V_3, V_4, V_5, V_c), (D)) \]

Chromosome

Mutation Operation

\[ ((V_c, V_2, V_5, V_4, V_3, V_c), (D^n)) \]

Chromosome

**Figure 11.4:** The Mutation Operator for Evolution of Chromosome Population

The cross over and mutation operators are applied repeatedly to produce new offspring. The offspring replace the least fit individuals in the population if the fitness of the offspring is better than the least fit individuals of the previous generation. This process of evolution continues until either there is no improvement in the fitness of the best individual of the population after a minimum number of iterations or the maximum number of generations has been evolved. The best individual of the population in the GA at termination gives the optimal path for distribution of postal mail in the cluster. The steady state GA procedure used in generation of the optimal path for distribution of mail in the cluster is similar to the steady state GA described in algorithm 10.4 but uses the chromosome representation and operators described in this section. The complete steady state GA procedure for finding the optimal LOCAL tour is described in algorithm 11.4.

**Algorithm 11.4:** Genetic Algorithm for finding the optimal local tour for distribution of mail in a cluster of mail delivery points

**Input:** The mail delivery points in a cluster, Number of mail delivery points in cluster (NDP), Cluster center, GIS-DATA describing the postman beat, the cost adjacency matrix of the mail delivery points in the cluster (Cluster_Cost)

**Output:** The optimal local tour (LTOUR) for distribution of mail in the cluster starting and ending at the cluster center

1. Generate the initial population by randomly creating NPOP chromosomes, where NPOP=0.20*((NDP-1)!!). (An empirical fraction of 0.20, of all the possible solutions ((NDP-1)!!) is the initial population for the steady state GA, further the population size is limited to 150)
2. Find the fitness of each of the chromosome and store in a Fitness array and assign cumulative probabilities for use in roulette wheel selection.

3. Initialize the maximum number of iterations to N, maximum fitness value to -1 and minimum number of iterations to n

4. While the terminating conditions are not met do
   a. Find the maximum fitness amongst the chromosomes of the population
   b. If there is no change in maximum fitness from previous iteration and minimum number of iterations are over then break out of while loop as there is no improvement in the solution
   c. If the iteration is greater than the maximum iterations then break out of while loop
   d. Select two parents (P1 & P2) for reproduction from the population pool using roulette wheel selection. Every individual gets selected based on their fitness value
   e. Generate two offspring (O1 & O2) using the crossover operator
   f. Apply mutation to the offspring
   g. Find the fitness value of the offspring.
   h. Find the two least fit individuals in the current population
   i. If the fitness of the offspring is greater than the least fit individuals replace the least fit individuals with the offspring
   j. Evaluate the fitness of the new population

5. endWhile

6. Find the individual/ chromosome with the highest fitness in the population. The chromosome provides the sequence of distribution of mail to the delivery points in the cluster (LTOUR)

The procedure for finding the LOCAL tour (LTOUR) is invoked, for each of the clusters and cluster centers identified using algorithms 11.2 and 11.3. The postman has to visit all the cluster centers and the mail delivery points, which are not part of any cluster using an practically optimal traversal path on a vehicle. Such a path is referred to as a GLOBAL path. The GLOBAL path starts and ends at the MDPO. The procedure for finding the practically optimal GLOBAL path is described in section 11.4.

11.4 PRACTICALLY VIABLE GLOBAL PATH

The practically optimal and viable GLOBAL path for distribution of postal mail can be found for the postman to traverse on his vehicle to visit all the mail delivery points which are not in any clusters and all the cluster centers/ vehicle parking locations. The process
employed for finding the global path is similar to the one used for finding the optimal path for distribution of postal mail without clustering as described in chapter 10. The cluster centers are treated as individual delivery points that replace all the delivery points in a cluster. This new set of delivery points is processed to find the mail distribution graph (MDG) as described by algorithm 10.1. The MDG is further processed to generate the connected graph, Euler graph and the optimal path as described in chapter 10. The steady state GA described in section 10.3.1 is employed in matching of odd degree vertices while converting a non Eulerian graph to an Eulerian graph.

The GLOBAL path so generated is combined with the LOCAL paths in each of the cluster to find the mail distribution schedule as described in section 11.5.

11.5 MAIL DISTRIBUTION SCHEDULE

The GLOBAL and LOCAL paths are combined to generate the practically optimal/ viable path for distribution of postal mail using the traverse-park-distribute strategy. The practically optimal path so generated and the set of active mail delivery points are processed to find the mail distribution sequence and the time schedule for distribution of postal mail. The mail distribution schedule is generated using the assumption that the mail is delivered to the addressee when the postman traverses on the road for the first time and delivers to all the delivery points in a cluster/ partial edge when he reaches the cluster center/ partial edge for the first time. The distribution schedule in a cluster is determined by the local tour generated using algorithm 11.4. The distribution schedule for the mail delivery points on a partial edge is determined by the distance of the delivery points from the starting point of the partial edge. The distribution schedule for all the other delivery points is decided by the direction of the traversal of the postman on the road. The computational procedure for generation of the mail distribution schedule is outlined in algorithm 11.5.
Algorithm 11.5: Generation of Mail Distribution Schedule for distribution of mail using traverse-park-distribute strategy.

Input: The GLOBAL practically optimal path (PP), The local tours in each of the identified clusters(LTOUR), The list of edges that are to be partially traversed (PEDGE), the list of all the edges that have active mail delivery points(iMDG), The list of active mail delivery points (AMDP)

Output: The Sequence of distribution of mail along with time of delivery(Dist_Sched)

1. Remove the edges that need partial traversal from the list of edges iMDG
2. Rearrange the active mail delivery points (AMDP) in non-decreasing order of the edge ID. The mail delivery points on one road are collected at one location
3. Sort the active mail delivery points lying on each edge/ road in the non decreasing order of the distance of the MDPs from end point 1 of the road
4. Initialize edgeindex to point to first edge in euler trail/ postman path (PP) and vertex to the MDPO node
5. While there are more edges in PP do
   a. Find the first/ next edge in PP, say e
   b. Find the initial vertex of the edge e
   c. Check whether there is a partial edge to be traversed from this vertex, if so generate the distribution schedule for the MDP's in the edge to be partially traversed from PEDGE and store the schedule in DIST_SCHED, Mark the delivery points to which distribution schedule has been generated. If any of the delivery points in the partially traversed edge is a cluster center, add the distribution schedule of the local tour
   d. Find if there are mail to be delivered on the edge/ road (e)
   e. If there are mails to be delivered then
      i. Find the direction of traversal of the edge and assign the distribution schedule for the delivery points on the road according to traversal direction and store it in DIST_SCHED. Also store the distribution time for each. If any of the delivery point is a cluster center, add the distribution schedule of the local tour
      ii. Mark the delivery points to which distribution schedule has been generated
   f. Endif
   g. If the delivery schedule of all the mail delivery points is generated break out of the while loop
6. endwhile
7. DIST_SCHED consists of the distribution schedule for delivery of the mail on the given day along with time of delivery and parking time for parking of vehicle in purely parking lots

The postman employs the practically optimal path and the distribution schedule, generated using the strategies described here, for efficient delivery of the mail to the addressee's using the traverse-park-distribute strategy. A typical illustration of the generation of a practically optimal path and mail distribution schedule is given in section 11.6.
11.6 COMPUTATION OF PRACTICALLY VIABLE PATH AND DISTRIBUTION SCHEDULE: A CASE STUDY

The computational strategies described in the previous sections can be employed in generating a practically optimal and viable path for distribution of postal mail using the traverse-park-distribute criterion. The computational strategies also generate the schedule for distribution of mail along with the time for distribution. The vehicle parking locations are also identified. This section describes the generation of the practically optimal path and mail distribution schedule for the postman beat depicted in Figure 10.5, using the traverse-park-distribute strategy. The active mail delivery points that is the mail delivery points that receive mail on a given day are the same as that employed in section 10.5 and are listed in Figure 11.5 for ready reference.

Figure 11.5: The List of Mail Delivery Points (Ids) which Receive Mail

The active mail delivery points and the PGIS-DATA for the postman beat are processed to generate an intermediate mail distribution graph (iMDG). The iMDG consists of all the edges that have active mail delivery points. The iMDG generated by processing the active delivery points listed in Figure 11.5 is depicted in Figure 11.6.

Figure 11.6: Intermediate Mail Distribution Graph (iMDG) for Delivery Points in Figure 11.5
The intermediate mail distribution graph of Figure 11.6 and the active mail delivery points of Figure 11.5 are processed using algorithm 11.2 to find the clusters in the active delivery points. The maximum radius of the neighborhood is assumed to be 20 meters and the minimum number of delivery points in a cluster is assumed to be 2, these parameters are assigned using practical considerations. The clusters and the delivery points, which are not assigned to any cluster, are listed in Figure 11.7

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Delivery Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>20,21,22,23,24</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>42,43,44,45,46</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>93,94,95</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>117,118,119</td>
</tr>
<tr>
<td>Cluster 5</td>
<td>222,223,224,225,226</td>
</tr>
<tr>
<td>Mail Delivery Points not in any Cluster</td>
<td>96,120,121,176,184,188,249,250,251,252,253,278,282,287,296,302,307</td>
</tr>
</tbody>
</table>

Figure 11.7: The Clusters in the Active Mail Delivery Points

The identified clusters, cluster 1 to cluster 5 are processed using algorithms 11.3 and 11.4 to find the cluster centers/ vehicle parking positions and the optimal local tours for distribution of postal mail. The identified cluster centers and the optimal local tours in each of the clusters are enlisted in Table 11.1
Table 11.1: The Cluster Centers and Optimal Local Tours

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Cluster</th>
<th>Cluster Center/ Parking Location</th>
<th>Remarks</th>
<th>Optimal LOCAL Tour in the cluster determined using GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cluster 1</td>
<td>22</td>
<td>The MDP which is at minimal distance from all the other MDP’s is 21 but as parking is not allowed by 21, the next nearest MDP is chosen as the cluster center</td>
<td>22-21-24-23-20-22 (Total Distance: 60)</td>
</tr>
<tr>
<td>2</td>
<td>Cluster 2</td>
<td>44</td>
<td>44 is the MDP which is at minimal distance from all other MDP’s</td>
<td>44-46-45-42-43-44 (Total Distance: 80)</td>
</tr>
<tr>
<td>3</td>
<td>Cluster 3</td>
<td>94</td>
<td>94 is the MDP which is at minimal distance from all other MDP’s</td>
<td>94-93-95-94 (Total Distance: 40)</td>
</tr>
<tr>
<td>4</td>
<td>Cluster 4</td>
<td>-116</td>
<td>None of the MDP’s in the cluster allow parking, hence a parking lot in the vicinity is added to the cluster and is made the cluster center (the minus sign indicates -116 is a parking lot)</td>
<td>(-116)-118-117-119-(-116) (Total Distance: 60)</td>
</tr>
<tr>
<td>5</td>
<td>Cluster 5</td>
<td>224</td>
<td>224 is the MDP which is at minimal distance from all other MDP’s</td>
<td>224-222-223-226-225-224 (Total Distance: 80)</td>
</tr>
</tbody>
</table>

Once the optimal local tours in each cluster are generated, a practically optimal path for visiting the cluster centers and the delivery points, which are not in any cluster, is to be generated. For generating such an optimal path the cluster centers and the delivery points that are not in any cluster are combined to form a new set of delivery points/cluster centers, which are to be visited. The new sets of delivery points so generated are listed in Figure 11.8.

Figure 11.8: The New Set of Delivery Points/Cluster Centers that are to be Visited

These points are processed along with the PGIS-DATA for the postman beat to generate the Mail Distribution Graph, which has edges that need partial traversal and edges that need complete traversal. The mail distribution graph generated by processing the points in Figure 11.8 is depicted in Figure 11.9.
A careful observation of the mail distribution graph in Figure 11.9 indicates that it is the same as the one in Figure 10.7. This is a coincidence and will not always be the case. This has happened because the clustering of mail delivery points has resulted in clustering of MDP's on partially covered edges only. The MDG is converted to Euler graph in a manner similar to that illustrated in section 10.5.

An Euler trail in the Euler Graph is generated using Fleurys algorithm. This describes the path taken by the postman for distribution of postal mail. The postman path, the cluster centers and the active delivery points, which receive mail, are processed to find the distribution schedule. The traversal path and the distribution schedule generated by the computational strategies are depicted in Figure 11.10.

The parameters of the practically optimal path depicted in Figure 11.10 illustrate that the distance traveled by the postman using this strategy is more than the distance traveled when the postman goes from one delivery point to another without any clustering of mail delivery points. The computational strategies for generation of practically optimal/ viable path for distribution of mail using the traverse-park-distribute strategy are tested using different lists of active delivery points and for different postman beats. The results are presented in section 11.7.
A GLOCAL Strategy

**Optimal Path for Delivery of Mail and Distribution Schedule**

**Postman Path** (Sequence of vertices visited): 1-3-8-12-15-14-10-11-11-14-13-9-5-6-2-1

**Mail Distribution Schedule** (Mail Delivery Point (Delivery Time in seconds))

Note: Parking position is indicated by P, and clusters are shown in curly braces, Negative ID indicates parking lot and the corresponding time is the parking time

\[
\text{Note: Parking position is indicated by P, and clusters are shown in curly braces, Negative ID indicates parking lot and the corresponding time is the parking time.}
\]

\[
\{44(P-20)-46(20)-45(20)-42(15)-43(20)-176(15)-184(15)-188(20)-307(20)-302(20)-296(20)-253(20)-252(15)-251(20)-250(15)-249(20)-224(20)-222(15)-223(20)-226(20)-225(20)) -116(P-5)-118(20)-117(20)-119(20)-120(20)-121(20)-287(20)-288(20)-278(15)-94(P-20)-93(20)-95(20)-96(20)-22(20)-24(15)-23(15)-23(15)-20(20)}
\]

**Distance Traversed by Postman on Global Path:** 7560 mts

**Distance Traversed by Postman on Local Tours:** 320 mts

**Total distance traversed by postman:** 7880 mts

**Transit time on vehicle** (@ average transit rate of 10 mts/sec) : 724 secs

**Transit time on foot** (@ average transit rate of 0.5 mts/sec) : 640 secs

**Time for crossing Junction Points:** 125 secs (Cross over time of each junction point available in GIS data)

**Time for delivery of Mail:** 705 secs (Delivery time of each MDP available in GIS data + Parking time)

**Total Time for traversal and delivery:** 2194 secs: 36.56 mins

Figure 11.10: Optimal Path for Delivery of Mail and Distribution Schedule

**11.7 EXPERIMENTAL ANALYSIS**

The computational strategies for finding the optimal path for distribution of mail using the traverse-park-distribute criterion are implemented on a computer with an INTEL Celeron processor @1.3 GHz and 512 MB memory using MATLAB 7.0 software. The intelligent strategies for generation of practically optimal path for traversal of the postman and the mail distribution schedule are tested on three different postman beats, which are represented by the graphs illustrated in Figures 10.5, 10.11 and 10.12. Each of the postman beat is tested with three different lists of mail delivery points to which mail is to be delivered. These lists of mail delivery points represent active mail delivery points on different days.

The optimal local paths are generated using the steady state GA that finds the Hamiltonian path in the cluster of mail delivery points and the global traversal path is generated using
the steady state GA described in section 10.3.1 and the Fleury's algorithm. The result of applying the computational strategies to the list of active mail delivery points is tabulated in Table 11.2. The distances listed in the table are the minimum distance values obtained in twenty iterations.

Table 11.2: Results of the Generation of Practically Optimal Path for Distribution of Postal Mail using Traverse-Park-Distribute strategy

<table>
<thead>
<tr>
<th>SI No</th>
<th>Graph Representing Beat of Postman</th>
<th>Number of Mail Delivery Points</th>
<th>Number of Clusters Found</th>
<th>Number of New Mail Delivery Points/Cluster Centers</th>
<th>Total Length of Global Path</th>
<th>Total Length of Local Tours</th>
<th>The Combined Length of LOCAL path</th>
<th>Total Time required for Distribution of Mail</th>
<th>Average Computation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G1 (</td>
<td></td>
<td>V</td>
<td>=15),</td>
<td>38</td>
<td>5</td>
<td>22</td>
<td>7560</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>E</td>
<td>=20)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>G2 (</td>
<td></td>
<td>V</td>
<td>=32),</td>
<td>60</td>
<td>9</td>
<td>38</td>
<td>8570</td>
<td>520</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>E</td>
<td>=47)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>G3 (</td>
<td></td>
<td>V</td>
<td>=45),</td>
<td>103</td>
<td>14</td>
<td>60</td>
<td>9550</td>
<td>730</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>E</td>
<td>=62)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: All Distances are in meters

The results in Table 11.2 indicate that the proposed strategies are computationally intensive as the amount of computational time increases with respect to the number of mail delivery points. The computational time depends on the critical operation of clustering of mail delivery points. The time required for clustering increases as the number of mail delivery points in one particular cluster increases. This fact is brought out by the computation time for the generation of practically optimal path in graph G2. The number of mail delivery points in one cluster is very high for each of the instances of the active delivery points for graph G2, and hence the computational time for generation of practically optimal path and mail distribution schedule is more for the test instances of graph G2. The time profile of the computational strategies to generate practically optimal path for distribution of mail using traverse-park-distribute strategy is depicted in Figure 11.11.
The results of the experimentation indicate that the local tour found using the steady state GA described in this chapter gives the optimal value (minimal cost path) in at least 90% of the iterations but the global path length varies and is never greater than 1.25 times the optimal path length as discussed in chapter 10. Hence the genetic algorithms used in this work provide good heuristic solutions to the respective problems.

The total distance traveled using the traversal path generated using this strategy is greater than the distance of the optimal path generated for the postman who travels from one mail delivery point to another without clustering of mail delivery points because of the obvious reasons such as the traversal of the same road multiple times for distribution of mail in the cluster and also during the global traversal. Further the total time required for distribution of mail is also more because of the increased distance of traversal and movement by foot for delivery of mail in a cluster. This aspect is brought out by the values of the practically viable path length and distribution time listed in Table 11.3. The Table 11.3 lists the length of the optimal path generated using the computational strategies outlined in chapter 10 (optimal path for RPP:DP problem) and the length of the practically optimal and viable path generated using computational strategies presented in this chapter (traversal path for the RPP:DPP problem). The total distribution times using the two strategies are also listed.
Table 11.3: The Comparative Results for Traversal Path and Distribution Schedule

<table>
<thead>
<tr>
<th>SI No</th>
<th>Graph Representing Beat of Postman</th>
<th>Number of Mail Delivery Points to which Mail is to be delivered</th>
<th>Length of the Optimal Path for Distribution of Mail (Using strategies discussed in Chapter 10) in meters</th>
<th>Length of the Practically Viable Path for Distribution of Mail using Traverse Park and Distribute strategy in meters</th>
<th>Total Time for distribution of Mail (Using Strategies discussed in Chapter 10.) in seconds</th>
<th>Total Time for Distribution of mail using traverse, park and distribute strategy in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G1</td>
<td>38</td>
<td>7260</td>
<td>7880</td>
<td>1551</td>
<td>2194</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>7890</td>
<td>9090</td>
<td>2059</td>
<td>3120</td>
</tr>
<tr>
<td></td>
<td></td>
<td>103</td>
<td>7840</td>
<td>10280</td>
<td>2849</td>
<td>4432</td>
</tr>
<tr>
<td>2</td>
<td>G2</td>
<td>40</td>
<td>13700</td>
<td>15000</td>
<td>2610</td>
<td>4600</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63</td>
<td>16000</td>
<td>23300</td>
<td>3255</td>
<td>9050</td>
</tr>
<tr>
<td></td>
<td></td>
<td>104</td>
<td>16800</td>
<td>27700</td>
<td>4545</td>
<td>13870</td>
</tr>
<tr>
<td>3</td>
<td>G3</td>
<td>40</td>
<td>21000</td>
<td>21000</td>
<td>2815</td>
<td>4950</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60</td>
<td>18500</td>
<td>23900</td>
<td>3815</td>
<td>6525</td>
</tr>
<tr>
<td></td>
<td></td>
<td>105</td>
<td>22300</td>
<td>34000</td>
<td>4290</td>
<td>16000</td>
</tr>
</tbody>
</table>

The results in Table 11.3 indicate that the length of the practically optimal and viable path generated using the traverse-park-distribute strategy is greater than the corresponding optimal path length generated using the strategies presented in chapter 10. The total time required for distribution of mail is also greater. But the advantage of using this strategy for generation of traversal path for mail distribution is that it generates a path, which is more nearer to the strategy adapted by the postman for distribution of mail in localities of metropolitan cities and semi urban areas.

11.8 CONCLUSION

New computational strategies for generation of practically viable path and mail distribution schedule using the traverse-park-distribute criterion are presented in this chapter. This practically optimal path is a solution to the Rural Postman Problem: Delivery Points with Parking (RPP:DPP). The methodology for finding the practically optimal/ viable path for distribution of postal mail involves automatically identifying clusters of mail delivery points that receive mail (active delivery points) using modified GDBSCAN approach, finding the cluster centers for identifying the location for parking the postman’s vehicle and further finding local and global traversal paths for distribution of postal mail. The optimal local path is obtained for every cluster of mail delivery points using a newly devised steady state genetic algorithm. The practically optimal global path is generated for
traversal of the postman from mail delivery post office to all the cluster centers and the mail delivery points, which are not included in any cluster using the computational strategies described in chapter 10. The new computational strategies using GLOCAL approach presented in this chapter generate a practically viable path for distribution of postal mail employing traverse-park-distribute strategy.

The computational strategies devised and presented in this chapter for distribution of mail can be used in various other applications such as distribution of milk, newspapers, food items etc. The strategies can also be adapted for robot motion planning electronic component manufacture etc.