WASTE COLLECTION VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

B. Tech

in

Production Engineering

By

A.S. Arjun Raj (114112001)

Albert C George (114112006)

Karthik Sajeev (114112039)

DEPARTMENT OF PRODUCTION ENGINEERING

NATIONAL INSTITUTEOF TECHNOLOGY

TIRUCHIRAPALLI-620015

MAY 2016

WASTE COLLECTION VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

B. Tech

in

Production Engineering

By

A.S. Arjun Raj (114112001)

Albert C George (114112006)

Karthik Sajeev (114112039)

DEPARTMENT OF PRODUCTION ENGINEERING

NATIONAL INSTITUTEOF TECHNOLOGY

TIRUCHIRAPALLI-620015

MAY 2016

BONAFIDE CERTIFICATE

This is to certify that the project titled **WASTE COLLECTION VEHICLE ROUTING PROBLEM WITH TIME WINDOWS** is a bonafide record of the work done by

A.S. Arjun Raj (114112001)

Albert C George (114112006)

Karthik Sajeev (114112039)

in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Production Engineering** of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI,** during the year 2015-2016.

Dr. S. Prasanna Venkatesan
Guide

Dr-Ing M. Duraiselvam
Head of the Department

Project Viva-voce held on ______________________________

Internal Examiner

External Examiner

ABSTRACT

In this study we have tried to solve Waste Collection Vehicle Routing Problem with Time Windows (WCVRPTW) models with the help of different techniques, namely MS Excel Solver simplex optimization and Particle Swarm Optimization (PSO). The models are similar in terms of the time window constraint, homogenous fleet with capacity constraint and the fact that each customer is served only once. The difference between the models is that in the first model, which is based on 'The waste collection vehicle routing problem with time windows in a city logistics context' by Buhrkal, K. et. al., the disposal site and depot are two different nodes, and the vehicle can continue with the route after visiting the disposal site. In the second model, which is based on 'An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows' by Zhu, Q. et. al., both disposal site and depot are considered as same and the route terminates once it reaches the depot.

The first mathematical model is implemented using MS Excel Solver for a small-sized problem consisting of four customers and two vehicles. The objective is to find a route which minimises the travel cost ensuring that all customers are attended to during their available time windows. Simplex optimisation technique is used by solver and the optimal route is generated as output. While this approach is fast and suitable for small-sized problems, it may not be preferred for highly complex problems, which are prevalent in most real-life cases.

The consideration for using PSO is that, WCVRPTW being a NP-Hard problem, makes it difficult to find the optimal solution as the size increases and also the process of defining the problem constraints and definition in Excel Solver for a large scale problem is time consuming and tiring. Thus the choice of Meta heuristic method such as a PSO was done. In the PSO used the solution representation is done using Permutation Encoding and in order to improve the efficiency of the PSO we have implemented it as a Discrete PSO by using List of Moves particle updation method, along with an own heuristic repair method.

Key words: WCVRP-TW, Excel Solver, Discrete PSO, Permutation Encoding, List of Moves particle updation

ACKNOWLEDGEMENTS

We would like to express our gratitude towards our Project Guide **Dr. S Prasanna Venkatasan** for his continuous guidance throughout the course of this endeavour. But for his patience and constant support this project would not have materialized in such a fulfilling way.

We also would like to thank our beloved HOD **Dr-Ing M. Duraiselvam** for his support and flexibility in the choice of the project topic and field.

We also appreciate the help provided by the Research Scholar, **Mr. Rahul**.

Lastly we would also like to thank all our friends and classmates who have helped in bettering this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Waste management refers to the collection of all activities dealing with the management of waste from its inception to its final disposal. Waste collection is a part of waste management, which is concerned with the collection and transport of waste from the point of use to the point of treatment or discharge. It may be carried out by municipal services, public or private corporations or specialized enterprises.

The Vehicle Routing Problem (VRP), which was first introduced by Dantzig and Ramserl in 1959, may be defined as follows: "M vehicles initially located at a depot are to deliver discrete quantities of goods to n customers. Determining the optimal route used by a group of vehicles when serving a group of users represents a VRP problem". Waste collection may thus be regarded as a vehicle routing problem.

A Vehicle Routing Problem may have different objectives based on the specific requirements of the study: minimising total route cost, shortening route distance, finding the most appropriate location for disposal sites or minimising the number of collection vehicles.

WCVRPTW is a NP-Hard problem, meaning that the effort of finding the optimal solution increases exponentially with the size of the problem. Hence heuristics or metaheuristics are used to find near best optimal solutions. In this study we have used the Particle Swarm Optimization metaheuristic, which is a population based computational method that optimizes iteratively.

1.1 SCOPE

This study is based on similar models taken from two papers. Both deal with the WCVRPTW problem, where the objective is to reduce the total travel cost given a fixed network and cost of travel between nodes. Customers in both cases need to be accessed within their specified time windows and vehicles have a fixed capacity after the fulfilment of which waste has to be dumped. However, the two models differ in that while the first model has separate nodes for disposal site and depot, the second has both these facilities combined as a single node.

1.2 METHOD

The first mathematical model is solved using MS Excel Solver. We have created a simple system consisting of 1 depot, 2 disposal sites and 4 customers. 2 vehicles are used for collecting waste. Based on the capacity constraints of the vehicles and the time windows of customers, an optimal route in terms of travel cost is generated using MS Excel Solver.

Particle Swarm Optimisation is used on the second model. A system comprising of 8 nodes and 3 vehicles is taken and the results are obtained using MATLAB.

1.3 THESIS ORGANISATION

The thesis is organised as follows: Chapter 2 deals with the related literature on VRPTW and the use of particle swarm optimisation for vehicle routing problems. Chapter 3 covers the problem description, definition and gives illustrative examples with regard to the model in MS Excel. The same format is followed for the PSO model, which is implemented in MATLAB and is covered in Chapter 4. Chapter 5 discusses the results obtained and finally the conclusion is presented in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

Waste generation has caused extensive public apprehension in the contemporary world. The amount of waste generated and the increasing complexity of some products and components has instigated concern in various levels. Waste collection is an important process in waste management. The Organisation for Economic Co-operation and Development (OECD) in 1997 defines waste collection as follows: Waste collection is the collection and transport of waste to the place of treatment or discharge by municipal services or similar institutions, or by public or private corporations, specialized enterprises or general government. Collection of municipal waste may be selective, that is to say, carried out for a specific type of product, or undifferentiated, in other words, covering all kinds of waste at the same time (OECD, 1997 [10]).

In the waste collection problem, organizations need to take back wastes from the collection point and send them to the treatment facilities. Many collection points need to be collected in order to be sent to the suitable facility. Several constraints like time windows, real time requirements, capacity of the vehicle, multi depot constraints, multi disposals etc. can be recognised as source of intricacies and complexities. For example, if different waste collection sites have varied amounts of wastes to be disposed, there can be only little visibility due to the difficulty in knowing the amount of waste in advance. The vehicle has to go to the disposal sites to empty the accumulated waste in order to the completely collect further waste.

Thus, waste collection can be treated as a Vehicle Routing Problem. The Vehicle Routing Problem (VRP) introduced for the first time by Dantzig and Ramserl in 1959 [2] focused on forward deliveries (from a depot to “n” customers). A waste collection vehicle routing problem typically consists of a fleet of vehicles, disposal facilities, a depot, and a number of waste collection sites or collection points. A vehicle starts and ends at the same depot. If the capacity constraints is exceed, the vehicles go to the disposal sites and empty the containers and continue the waste collection. These activities go on until all the waste have been collected and disposed. The intricacy of the problem depends on the different aspects like number of vehicles used, number of disposal sites, time intervals, capacity revealing the Vehicle Routing

Problem with Time Windows (VRPTW) and the Capacitated Vehicle Routing Problem.

There are three types of waste collection problems: residential, commercial, and industrial (Golden et al., 2002) [4]. Kim et al. (2006) [7] defined the residential waste collection as follows: The residential waste collection generally involves servicing private homes. The number of homes a residential route may service varies widely from 150 to 1,300 homes every day. The frequency of service per week will vary based on the climate, geography, competition and price of service.

Commercial waste collection involves collection of commercial refuse from large containers at commercial locations, according to Golden et al. [4]. The commercial waste collection involves servicing customers such as strip malls, restaurants and small office buildings. Each commercial route of waste management may service 60–400 customers, with two or three disposal trips to dump sites each day. Depending upon the customer base, the same driver may visit the same customer multiple times in one week. The weekly service schedule is fairly static, as most customers do not change the frequency of service often (Kim et al., 2006) [7].

According to Golden et al. [4] roll-on-roll-off problems involve the pickup, transportation, unloading and drop-off of large trailers (or containers) typically located at construction sites. [7] Kim et al. (2006) defined the roll-on-roll-off waste collection as follows : The roll-on-roll-off collection introduces a different routing problem. The differentiator between roll-on roll-off and commercial is the size of the container. A typical commercial container is eight loose yards, while a roll-on-roll-off container may range from 20to 40 loose yards and only one container may be serviced at a time. Note that 1 cubic yard is 0.765m3.

In [13] Juyoung Wy et al., 2013, the researchers consider tractors moving one container at a time between customer locations, a depot, disposal facilities, and container storage yards. The complicated constraints discussed in this study arise from having multiple disposal facilities, multiple container storage yards, seven service types of customer demands, different time windows for customer demands and facilities, various types and sizes of containers, and the lunch break of tractor drivers. The methodology proposed for the problem is a large neighbourhood search based iterative heuristic approach consisting of several algorithms.

Typically, waste collection system (collection and transportation) involves a very high operational cost. Some researchers have been trying to decrease the operational costs involved in the process of waste collection. In [8] Jing-Quan Lia t al. (2008), the researchers consider a truck scheduling problem in the context of solid waste collection, where the primary objective is to minimise the total operational costs. The paper also suggests a methodology to allocate trucks to the existing waste collection sites. Considering the social benefits of solid waste program it is desirable to obtain balanced assignments of collection trips unloading their consignment at the recycling facilities.

[7] Byung-In Kim et al. (2006), address a real life waste collection Vehicle Routing Problem with Time Windows (VRPTW) with consideration of multiple disposal trips and drivers' lunch breaks. Solomon's well-known insertion algorithm is extended for the problem. Solomon's insertion algorithm assumes that each vehicle departs from the depot, serves only one route, and returns to the depot. It must be extended to incorporate disposal operations and serving multiple routes and the lunch break. A capacitated clustering-based waste collection VRPTW algorithm is developed. The constraints considered in the paper are time windows of stops and the depot, vehicle capacity, route capacity, routing time limit per vehicle, disposal trips and driver's lunch break.

[1] Katja Buhrkal et al., 2012, discusses about waste collection vehicle routing problem with time windows in a city logistic context. It is concerned with finding a set of routes for the vehicles, minimizing total travel cost and satisfying vehicle capacity, such that all customers are visited exactly once within their time window. An adaptive large neighborhood search algorithm is proposed for solving the problem. To improve the objective function they have considered two instances or cases. One is the North American case where there is a limit on the number of customers to visit on each route, the total amount collected at customers for each route, which is a route capacity, lunch break (only allowed directly after servicing a node). The other is the Danish case in which only homogenous vehicles are considered, rest break always consists of 45 minutes after maximum 4½ hours driving and if a rest break is taken the lunch break is considered covered. The mathematical model in the aforementioned research paper is adopted to solve the problem in this paper.

Waste Collection Vehicle Routing Problems (WCVRP) problems can be divided into Node Routing Problems and Arc Routing problems. Node Routing Problems have demand on the nodes (or vertices) of a graph. An example of a Node Routing Problem might be a salesman who must travel to each town in a country. In the Arc Routing Problems the demand occurs in each of the arcs (or edges) of the graph. An example of an Arc Routing Problem might be a postman who must visit each house along each street in a town ([6] Han H et al., 2015).

Types / Characteristics	Node Routing Problems	Arc Routing Problems
Objectives	distribution/collection of goods (point to point)	distribution/collection of goods or materials along the arcs (edges) of a road network
Main components	• Vehicles • Depots • Drivers • Road Network	• Vehicles • Depots • Drivers • Road Network
Solutions	A set of routes performed by a fleet of vehicles such that: • each route starts and ends at vehicles' depots • the customers' requirements are satisfied • the operational constraints are fulfilled • the global transportation cost is minimized	A set of routes performed a fleet of vehicles such that: • each route starts and ends at vehicles' depots • the requests for service associated with arcs or edges are satisfied • the operational constraints are fulfilled • the global transportation cost is minimized
Types	• Traveling Salesman Problem (TSP) • Traveling Salesman Problem with Backhauls (TSPB) • Traveling Salesman Problem with Time Windows (TSPTW) • Multiple Traveling Salesman Problem (MTSP) • Capacitated Vehicle Routing Problem (CVRP) • Distance Constrained Vehicle Routing Problem (DCVRP) • Vehicle Routing Problem with Backhauls (VRPB) • Vehicle Routing Problem with Time Windows (VRPTW) • Vehicle Routing Problem with Pickup and Delivery (VRPPD)	All arcs (edges) must be served: • the Chinese Postman Problem (CPP) Only a subset of arcs (edges) must be served: • the Rural Chinese Postman Problem (RCPP) • the Windy Chinese Postman Problem • the Stacker Crane Problem

Fig. 2.1 Node and Arc routing problems

This paper is concerned about Node Routing Problem for commercial waste collection. Microsoft Excel Solver Premium is used to solve the objective function. Solver is part of a suite of commands sometimes called what-if analysis tools. With Solver, you can find an optimal (maximum or minimum) value for a formula in one cell — called the objective cell — subject to constraints, or limits, on the values of other formula cells on a worksheet. Solver works with a group of cells, called decision variables or simply variable cells that participate in computing the formulas in the objective and constraint cells. Solver adjusts the values in the decision variable cells to satisfy the limits on constraint cells and produce the result you want for the objective cell. The primary objective is to minimise the total travelling cost associated with the waste collection management.

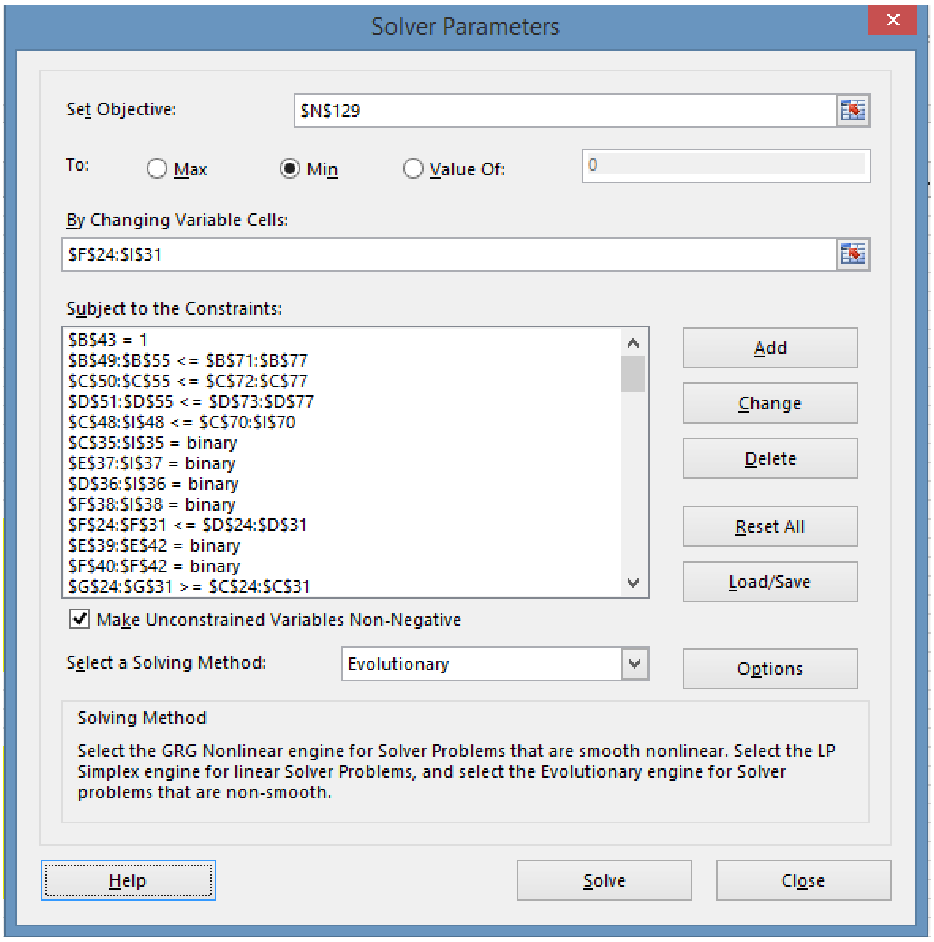

Fig. 2.2 MS Excel Solver

Three algorithms or solving methods in the Solver Parameters dialog box: Generalized Reduced Gradient (GRG) Nonlinear (Use for problems that are smooth nonlinear), LP Simplex (Use for problems that are linear), Evolutionary (Use for problems that are non-smooth). LP Simplex algorithm is used to solve this problem.

But the use of MS Excel Solver in terms of solving the WCVRPTW is that since the WCVRPTW being a NP-Hard problem it makes the solving of it tough and time consuming as the problem size increases. Thus the use of a meta heuristic method PSO was considered.

There has been a good amount of research on the application of PSO for VRPTW and even WCVRPTW. The methodology and particle encoding have been tried out in different ways.

[5] Shen H, et. al, 2009 In this paper they have elaborately explained the variations of PSO used in order to solve VRP problem. Ranging from the use of GL-VRP to hybrid PSO's, the hybrid being with GA operators, Simulated Annealing, Tabu search, Chaos Search. In terms of the particle encoding it was observed that a linear n+2m dimension particle encoding showed that it was competitive with other methods of particle representations.

In the study done by [9] MASROM .S et.al they have used the priority based encoding with n+2m linear dimensional particles. The PSO used is a hybrid with the mutation operator from GA being used along to avoid the local best stagnation. The results of hybrid PSO shows that the there is an improvement in terms of the results found against the normal PSO but at the cost of computational time.

[12] Wen L, et. al, 2008 has made a study of implementing an improved PSO for Multi Depot Vehicle Routing Problem MDVRP. The improvement to the PSO is in the dynamic change of the inertia value by a non-linear time decreasing function.

[3] Geetha S et.al,2010 in her paper she has come up with a hybrid PSO with the use of elitism, crossover and inverse mutation. In this method there has been the use of repair method. The repair method used is 2-opt method. The code was checked against the benchmark data of Augerat, et al set A, Augerat, et al set B, Augerat, et al set P, Christofides and Eilon.

[14] Zhu Q et. al, 2006 in there paper they have used a truncating based discrete PSO with n+k+1 dimensional linear particle in order to solve VRPTW with an illustration with an illustration of its effectiveness with an example.

[11] Venkatesan P S et.al, 2012 the authors have implemented Multi Objective Discrete PSO and the Discreteness is done through list of moves method to optimize the supply chain network. The results are compared with those of the Non-dominated Sorting Genetic Algorithm-II and it has been found that the proposed work has produced a high quality non pareto solutions.

CHAPTER 3

PROBLEM DESCRIPTION

This chapter describes the problem to be solved and the related mathematical model with respect to two papers –

1. 'The waste collection vehicle routing problem with time windows in a city logistics context' by Katja Buhrkal, Allan Larsen and Stefan Ropke [1].
2. 'An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows' by Qing Zhu, Limin Qian, Yingchun Li and Shanjun Zhu[14].

The mathematical model given in first paper has been implemented using the Solver add-in available with MS Excel. The model described in the second paper, which deals with PSO, has been coded on MATLAB.

3.1 MODEL 1

The objective of the Waste Collection Vehicle Routing Problem with Time Windows (WCVRPTW) is to find a set of routes, which would ensure that all customers are visited exactly once, within their time windows and subject to vehicle capacity constraints such that the total travel cost is minimised.

The vehicle routing problem is defined on a graph G = (V, A). V denotes the set of nodes and consists of a depot $V^d = \{0\}$, m disposal sites $V^f = \{1, \ldots, m\}$ and n customers $V^c = \{m+1, \ldots, m+n\}$. Thus, $V = V^d \cup V^f \cup V^c$. A denotes the set of arcs linking various nodes, $A = \{(i,j) | i,j \in V, i \neq j\}$. Travel time and cost associated with the arc (i,j) are denoted by t_{ij} and c_{ij} respectively. The set of vehicles is represented as $K = \{1, \ldots, k\}$. Each node $i \in V$ has an associated service time s_i and time window $[a_i, b_i]$. The amount of waste picked up at each customer $i \in V^c$ is denoted by q_i. All vehicles are assumed to have capacity C.

Three types of variables are used to model the problem: d_{il} denotes the accumulative demand at node $i \in V$ for vehicle $l \in K$, w_{il} denotes the service start time at node $i \in V$ and binary variable $x_{ijl} \in \{0, 1\}$ is one if and only if vehicle $l \in K$ uses arc $(i, j) \in A$.

The objective function for the problem may be mathematically represented as follows:

$$\min \sum_{(i,j) \in A} c_{ij} \sum_{l \in K} x_{ijl} \qquad (1)$$

The objective must be met subject to the following constraints:

$\sum_{j \in V} x_{0jl} = 1$	$\forall\, l \in K$	(2)
$\sum_{i \in V} x_{i7l} = 1$	$\forall\, l \in K$	(3)
$\sum_{i \in V} \quad \sum_{l \in K} x_{ijl} = 1$	$\forall\, j \in V^c$	(4)
$\sum_{i \in V} x_{ijl} = \sum_{i \in V} x_{jil}$	$\forall\, j \in V^c \cup V^f,\, l \in K$	(5)
$a_i \leq w_{il} \leq b_i$	$\forall\, i \in V,\, l \in K$	(6)
$w_{il} + s_i + t_{ij} \leq w_{jl} + (1 - x_{ijl})\, M$	$\forall\, (i,j) \in A,\, l \in K$	(7)
$\sum_{i \in \{0,7\}} d_{il} = 0$	$\forall\, l \in K$	(8)
$d_{il} + q_i \leq d_{jl} + (1 - x_{ijl})\, M$	$\forall\, i \in V \setminus V^f,\, j \in V,\, l \in K$	(9)
$d_{il} \leq C$	$\forall\, i \in V,\, l \in K$	(10)
$d_{il} \geq 0$	$\forall\, i \in V,\, l \in K$	(11)
$x_{ijl} \in \{0,1\}$	$\forall\, (i,j) \in A,\, l \in K$	(12)

The objective function (1) minimises the travel cost. This is mathematically achieved by minimising the value obtained by summing up the cost values for all the relevant arcs.

For the purpose of modelling the problem, the depot is split into a start and an end depot denoted by 0 and 7 respectively. All vehicles must start from the start depot only once. This means that for each vehicle, there must be just one outgoing arc from node 0 which the vehicle can take. Thus, the sum of feasible outgoing arcs from node 0 will be one, and this is depicted by (2). For similar reasons, the sum of feasible incoming arcs for each vehicle into node 7 must be one, and is shown in (3).

Constraint (4) makes sure that all customers are serviced exactly once. Unlike the last two constraints, this constraint does not apply separately for each vehicle. This is because it is sufficient that each customer be served by any one vehicle. Also, this constraint applies only for the customer nodes, meaning that disposal sites may be accessed more than once.

For all nodes other than the depots, and for each vehicle, the number of feasible incoming arcs must be equal to the number of feasible outgoing arcs. This is covered in (5) above.

Constraint (6) ensures that each node is accessed within its specified time window, and constraint (7) checks the feasibility of connecting two nodes based on the values of their service times and travel times between the two. M is a sufficiently large value.

At the beginning and end of the route, i.e. at the start and end depots, the vehicle must be empty. This means that the value of accumulated demand must be zero, and this is covered in (8).

Constraint (9) checks the feasibility of connecting two nodes based on the values of the amount of waste picked up at each customer and the accumulated demand.

Constraint (10) limits vehicle capacity to C, while constraint (11) imposes non-negativity. Constraint (12) emphasises the binary nature of x_{ijl}.

3.2 MODEL 2

The problem considered to solve is similar to that of the problem chosen for the Excel Solver except for the fact that both depot and disposal site are same and the vehicle ends its route once it reaches the depot. The model and solution representation are in reference to the paper, 'An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows' by Qing Zhu, Limin Qian, Yingchun Li and Shanjun Zhu [14].

CHAPTER 4

ILLUSTRATIVE EXAMPLES

4.1 MODEL 1

4.1.1 INPUT DATA

For the purpose of solving the WCVRPTW problem, we have assumed a model comprising of one depot, two disposal sites and four customers. Two vehicles are assumed to be present, whose routes would be obtained as the solution after solving the model. Both vehicles are assumed to be identical in all respects, and both have an equal capacity of 60 kg.

The input values to be fed by the user include travel time and cost of travel between nodes, time windows and service time at each node and the quantity of waste collected at each node.

The cost of travel between any two nodes in the network is represented in a matrix form, where the column number indicates the starting node (i) and the row number indicates the ending node (j), for every possible arc (i,j). Please note that the diagonal elements of this matrix would be zero as the cost of travelling from a node i to the same node i would be zero. Also, the matrix is a symmetric matrix as the cost of travelling from node i to node j is assumed to be the same as the cost of travelling from node j to node i. The cost matrix is shown in Fig. 4.1.

COST								
	0	1	2	3	4	5	6	7
0'	0	4	3	6	7	4	5	0
1	4	0	3	2	5	6	7	4
2	3	3	0	4	2	6	4	3
3	6	2	4	0	3	4	9	6
4	7	5	2	3	0	2	4	7
5	4	6	6	4	2	0	7	4
6	5	7	4	9	4	7	0	5
7	0	4	3	6	7	4	5	0

Fig. 4.1 Cost matrix

Travel time (in minutes) between nodes is represented in a manner similar to the cost matrix, and is shown in Fig. 4.2.

TIME(mins)								
j \ i	0	1	2	3	4	5	6	7
0'	0	20	15	30	35	20	25	0
1	20	0	15	10	25	30	35	20
2	15	15	0	20	10	30	20	15
3	30	10	20	0	15	20	45	30
4	35	25	10	15	0	10	20	35
5	20	30	30	20	10	0	35	20
6	25	35	20	45	20	35	0	25
7	0	20	15	30	35	20	25	0

Fig. 4.2 Travel time matrix

For each node, there exist two values for the time window- earliest starting time (a_i) and latest starting time (b_i). The vehicle must reach and begin service at these nodes at any time which falls between these two extreme values. For the sake of convenience, we have assumed 0 minutes as the start of service time and 900 minutes as the end of service time on a given day. The service time (s_i) at any node gives the amount of time a vehicle would have to spend at a node for loading/unloading waste. The quantity of waste collected (q_i) at each node is necessarily lesser than the vehicle capacity and is relevant only for customer nodes. These values are shown in Fig. 4.3.

	si	ai	bi	qi
0'	0	0	900	0
1	20	0	900	0
2	20	0	900	0
3	5	0	900	30
4	8	0	900	10
5	10	0	900	25
6	7	0	900	35
7	0	0	900	0

Fig. 4.3 Node specific parameters

4.1.2 Decision Variable Cells

As mentioned earlier, the decision variable cells to be manipulated by Solver are those corresponding to the binary variable x_{ijl}, service start time w_{il} and accumulative demand d_{il}. These cells are given a default initial value of zero and are coloured grey in our worksheet, so as to differentiate them from the other cells. This is shown in Fig. 4.4 and Fig. 4.5.

vhs1								
j \ i	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Fig. 4.4 x_{ijl} values for vehicle 1 before solving

	wi1	wi2	di1	di2
0'	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0

Fig. 4.5 w_{il} and d_{il} values before solving

4.1.3 Feeding Constraints

After feeding the relevant data into Solver and selecting the desired method of solving, the decision variable cells are changed by Solver in such a way that they satisfy all the specified constraints and at the same time, minimise the value of the objective cell.

Constraints (2) to (5) are related to the binary value matrix, and thus can be explained using Fig. 4.6.

vhs1															vhs2								
j \ i	0	1	2	3	4	5	6	7	e3	e5	e4		e5	e3	j \ i	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0							0'	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0		0			1		1	0	0	0	1	0	0	0	0
2	0	0	0	0	1	0	0	0		1			0		2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0		0	1		1		3	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0		1	1		0		4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0		0	1		1		5	1	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0		1	1		0		6	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	1					1	7	0	1	0	0	0	0	0	0
e2	1															1							
e5		0	1	0	1	0	1										1	0	1	0	1	0	

Fig. 4.6 Binary value matrices for the two vehicles

Constraint (2) may be checked by obtaining the sum of the elements in column 0 for the two vehicles separately. The presence of '1' corresponding to any node j under column 0 indicates a feasible path from 0 to j. Since there should be only one outgoing path from 0, this sum should equal one. For similar reasons, constraint (3) may be checked by obtaining the sum of the elements in row 7 for the two vehicles separately.

Constraint (4) is checked by taking the combined sum of the elements of corresponding rows of vehicle 1 and vehicle 2. This is done only for rows 3-6, as these as the rows corresponding to customers. Constraint (5) may be validated by comparing the sum of the elements of corresponding rows and columns for each vehicle separately. Constraint (6) limits w_{i1} and w_{i2} to take values between a_i and b_i. This is shown in Fig. 4.7.

ai	bi	wi1	wi2
0	900	0	0
0	900	0	860
0	900	865	0
0	900	0	50
0	900	52	0
0	900	0	20
0	900	25	0
0	900	900	900

Fig. 4.7 w_{il} for the two vehicles

The value of M in constraint (7) is taken to be 9999 for the purpose of calculation. The LHS and RHS of the inequality are modelled as separate matrices, and their corresponding elements are compared. Service time and travel time between nodes are input parameters. The two matrices for vehicle 1 are shown in Fig. 4.8.

vhs1	lhs eq7							
	0	1	2	3	4	5	6	7
0'	0	40	900	35	95	30	57	900
1	20	20	900	15	85	40	67	920
2	15	35	885	25	70	40	52	915
3	30	30	905	5	75	30	77	930
4	35	45	895	20	60	20	52	935
5	20	50	915	25	70	10	67	920
6	25	55	905	50	80	45	32	925
7	0	40	900	35	95	30	57	900

vhs1	rhs eq7	9999						
	0	1	2	3	4	5	6	7
0'	9999	9999	9999	9999	9999	9999	9999	9999
1	9999	9999	9999	9999	9999	9999	9999	9999
2	10864	10864	10864	10864	865	10864	10864	10864
3	9999	9999	9999	9999	9999	9999	9999	9999
4	10051	10051	10051	10051	10051	10051	52	10051
5	9999	9999	9999	9999	9999	9999	9999	9999
6	25	10024	10024	10024	10024	10024	10024	10024
7	10899	10899	900	10899	10899	10899	10899	10899

Fig. 4.8 LHS and RHS matrices for vehicle 1 in constraint (7)

Constraint (8) can be checked by taking the sum of accumulative demand values at the starting and ending depots, for each vehicle as shown in Fig. 4.9. d_{i1} and d_{i2} correspond to vehicles 1 and 2 respectively.

	di1	di2	Sum(0,7) - 1	Sum(0,7) - 2
0	0	0	0	0
1	60	60	C:	**60**
2	60	60		
3	0	30		
4	50	0		
5	0	0		
6	0	0		
7	0	0		

Fig. 4.9 Accumulative demand values

Similar to constraint (7), constraint (9) is also modelled as two matrices- one each for the LHS and RHS. Value of M is taken as 9999 and the quantity of waste collected at each node is an input parameter. The matrices for vehicle 1 are shown in Fig. 4.10.

vhs1	rhs eq9	9999						
	0	1	2	3	4	5	6	7
0'	9999	9999	9999	9999	9999	9999	9999	9999
1	10059	10059	10059	10059	10059	10059	10059	10059
2	10059	10059	10059	10059	60	10059	10059	10059
3	9999	9999	9999	9999	9999	9999	9999	9999
4	10049	10049	10049	10049	10049	10049	50	10049
5	9999	9999	9999	9999	9999	9999	9999	9999
6	0	9999	9999	9999	9999	9999	9999	9999
7	9999	9999	0	9999	9999	9999	9999	9999

lhs eq9	0	1	2	3	4	5	6	7
0'	0	60	60	30	60	25	35	0
1	0	60	60	30	60	25	35	0
2	0	60	60	30	60	25	35	0
3	0	60	60	30	60	25	35	0
4	0	60	60	30	60	25	35	0
5	0	60	60	30	60	25	35	0
6	0	60	60	30	60	25	35	0
7	0	60	60	30	60	25	35	0

Fig. 4.10 LHS and RHS matrices for vehicle 1 in constraint (9)

Constraints (10), (11) and (12) are fed directly into the Solver Parameters dialog box.

4.1.4 Objective Function

The summation matrix is obtained by adding the binary matrices of the two vehicles. The consolidated cost matrix is obtained by multiplying the corresponding elements of this matrix and the cost matrix defined earlier. The sum of all elements of this resultant matrix gives the total travel cost. The cell adjacent to 'Sum:' as seen in Fig. 4.11 is the objective cell, the value of which is minimised by Solver.

	summation xijl							
	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	0	0	1	0	0	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0
5	1	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0

	C*X							
	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	0
2	0	0	0	0	2	0	0	0
3	0	0	0	0	0	4	0	0
4	0	0	0	0	0	0	4	0
5	4	0	0	0	0	0	0	0
6	5	0	0	0	0	0	0	0
7	0	4	3	0	0	0	0	0
			Sum:	28				

Fig. 4.11 Objective cell

4.2 MODEL 2

In this section PSO has been chosen because it is a computational method that optimizes iteratively with the help of a population of candidate solutions in the search space. PSO being behaviour inspired optimization algorithm tries to mimic the food search done by a swarm of birds or a school of fishes. In this, each particle or candidate answers act as a bird in the swarm, having its own memory of best places for the search of food and also follow the group dynamics of following the group's lead. That is influenced by the best of all the individual memories.

4.2.1 INPUT DATA

The number of vehicles considered is k=3 and the number of customers/nodes is n=8. The number of particles (nop) = 200. The number of iterations N = 200. The capacity of the homogenous fleet of vehicles is 8 (hundred kg per vehicle).

In the following table the Time Window for each node is given as [a,b] the range of a & b is between noon and 10pm and noon is taken as 0^{th} hr and similarly 10pm as 10^{th} hr. In terms of demand the unit is 100kgs/node. It is indicated by the Q row in the following table.

Table 4.1 Time window and demand for each node

Node	0	1	2	3	4	5	6	7	8
[a,b]	[0,10]	[1,4]	[4,7]	[1,9]	[2.5,9]	[2,7]	[3,7]	[1,8]	[1,9]
Q(100kgs)	0	2	1.5	1	3	2	2	2.5	3

The distance matrix taken here is a symmetric matrix but it could also be taken as asymmetric. The Distance matrix is as follows:

Table 4.2 Distance matrix

From / To	0	1	2	3	4	5	6	7	8
0	0	40	60	75	90	200	100	160	80
1	40	0	65	40	100	50	75	110	100
2	60	65	0	75	100	100	75	75	75
3	75	40	75	0	100	50	90	90	150

4	90	100	100	100	0	100	75	75	100
5	200	50	100	50	100	0	70	90	75
6	100	75	75	90	75	70	0	70	100
7	160	110	75	90	70	90	70	0	100
8	80	100	75	150	100	75	100	100	0

From the above table both Cost matrix and travel time matrix can formulated by noting the average speed of the vehicle as 50 km/hr and the cost incurred per km of travel is Re.1.

Table 4.3 Travel time matrix (in hrs)

From / To	0	1	2	3	4	5	6	7	8
0	0	0.8	1.2	1.5	1.8	4	2	3.2	1.6
1	0.8	0	1.3	0.8	2	1	1.5	2.2	2
2	1.2	1.3	0	1.5	2	2	1.5	1.5	1.5
3	1.5	0.8	1.5	0	2	1	1.8	1.8	3
4	1.8	2	2	2	0	2	1.5	1.5	2
5	4	1	2	1	2	0	1.4	1.8	1.5
6	2	1.5	1.5	1.8	1.5	1.4	0	1.4	2
7	3.2	2.2	1.5	1.8	1.5	1.8	1.4	0	2
8	1.6	2	1.5	3	2	1.5	2	2	0

4.2.2 PSO Flow Chart

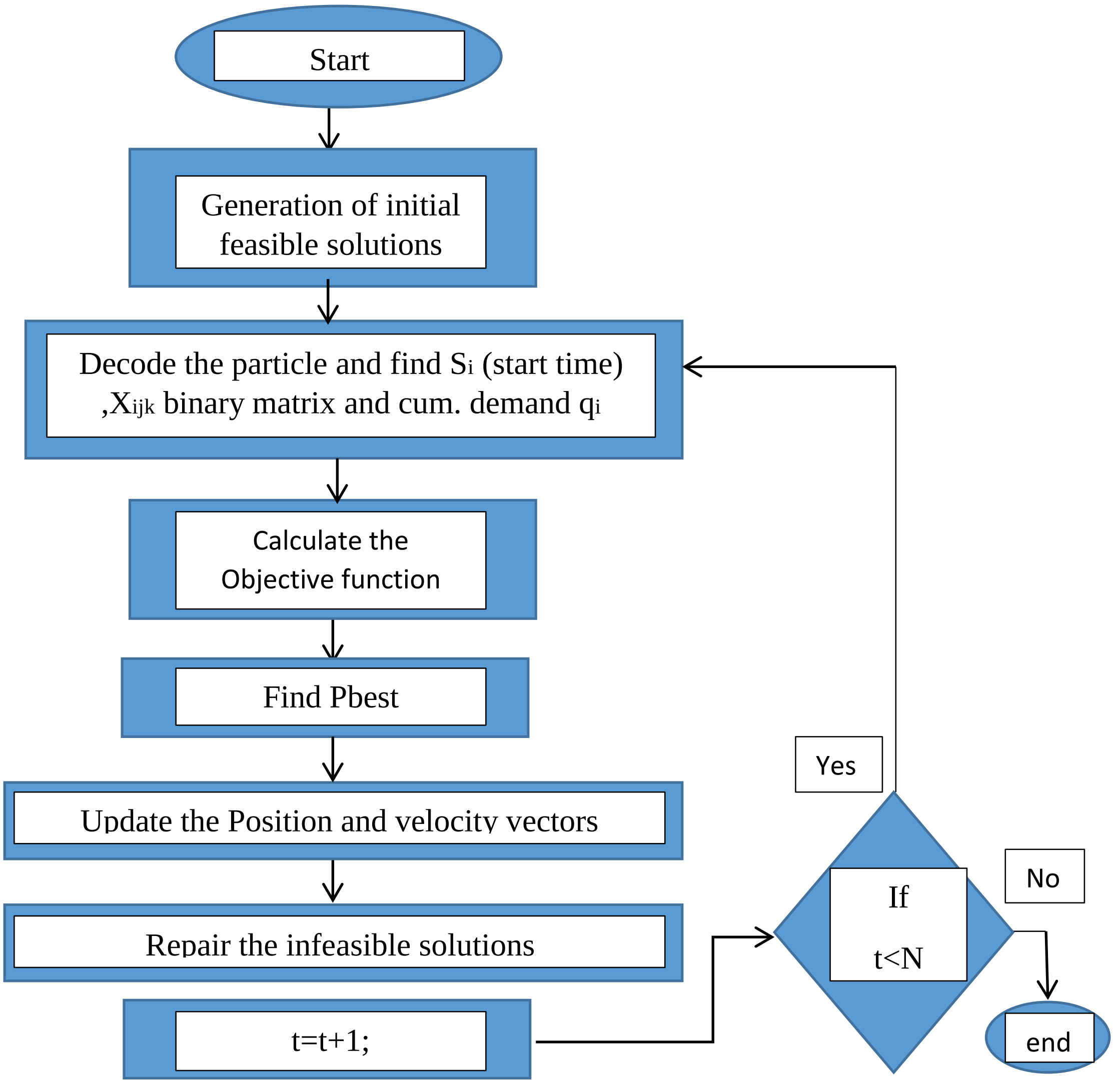

Fig. 4.12 PSO Flow chart

4.2.3 Definition Of Key Terms

4.2.3.1 Solution Representation

Solution Representation or also known as Particle Encoding is the concept of finding a suitable mapping between the problem solutions and the PSO particles ([14] Qing Zhu,et.al, 2006). The Solution representation considered in this paper is a variation of the one existing in the paper [14] by Qing Zhu,et.al, 2006.

Problem Solution:

Table 4.4 Vehicle schedule

Vh1 (route)	**0**	**2**	**8**	**5**	**3**	**0**
Starting time S_i(in x^{th} hr)	0	4	5.5	7	8	9.5
Cum. demand q_i(in kgs)	0	1.5	4.5	6.5	7.5	0
Vh2 (route)	**0**	**4**	**7**	**6**	**0**	
Starting time S_i(in x^{th} hr)	0	2	3.5	4.9	6.9	
Cum. demand q_i(in kgs)	0	3	5.5	7.5	0	
Vh3 (route)	**0**	**1**	**0**			
Starting time S_i(in x^{th} hr)	0	1	1.8			
Cum. demand q_i(in kgs)	0	2	0			

The above table represents one of the feasible solutions and now this solution is converted into a particle vector as follows. The encoding method is Permutation encoding; in this encoding not only the value in the dimension is important even the sequence of the value is important. The first and last terms of the position vector or particle are always zeros and as and when a vehicle reaches full capacity or can't fulfil any other customer time window. Then it goes to the depot cumulative disposal site. To illustrate different vehicles and routes, that is, the sequence of the nodes in the position vector, three different font styles are used namely, normal for Vh1, bold for Vh2 and italicised for Vh3.

The Particle or Position Vector:

0	3	4	2	0	**1**	**5**	**6**	0	*7*	*8*	0

4.2.3.2 Position/particle (pos)

As indicated in the previous sub section and table the position particle is of a linear n+k+1-dimensional vector/array. That represents a candidate solution and hence a group these particles, stored in a matrix of size [nop, n+k+1] are used to search the solution space based on how effective they are with respect to the objective function/fitness value.

4.2.3.3 Fitness Value/Objective function

Fitness Value or Objective Function is nothing but the Mathematical objective function of the Model. In addition to it could also be the sum of both the objective function and penalties given to constrain not able to imply by solution representation. This can be avoided by using the repair method (explained further in chapter).

The objective function in our problem is:

$$\min \sum_{(i,j) \in A} c_{ij} \sum_{l \in K} x_{ijl} \qquad (1)$$

4.2.3.4 Velocity

$$V_{id}(t+1)=w*V_{id}(t)+C_1*rand*[pbest_{id}(t)-pos_{id(t)}(t)]+C_2*rand*[gbest_{id}(t)-pos_{id(t)}(t)] \qquad (12)$$

The above equation gives the guidelines for calculating the velocity or the change in pos vector with each of the iteration for each dimension. It is done by simply adding the velocity vector to the current position vector.

4.2.3.5 Pbest Vector

Pbest is a matrix of size [nop,n+k+1] consisting of all the best solutions of each particle in its history. Improved in each of the iterations as and when a new personal best of each vehicle is found and stored against the corresponding i^{th} row of the Pbest matrix.

4.2.3.6 Gbest Vector

Gbest is a single linear n+k+1 dimensional array/vector. That stores the overall best till the present iteration. The final solution is the gbest at the time when termination condition is met.

4.2.4 Brief Outline Of PSO Algorithm

4.2.4.1 Generating Initial Population of the swarm (feasible solutions)

The pseudo code for generating the initial particle is:

- Initialize a matrix pos (position vectors) of [nop, n+k+1] as zeros.
- Generate nop number of random sequence (abc) of 1 to n numbers.

• Using the sequence as a priority order of the nodes in abc to choose the first feasible node.

• The feasibility is based on the time window constraint and capacity constraint. In time window constraint the particle can come early and or within the time window. If early has to wait till the opening time.

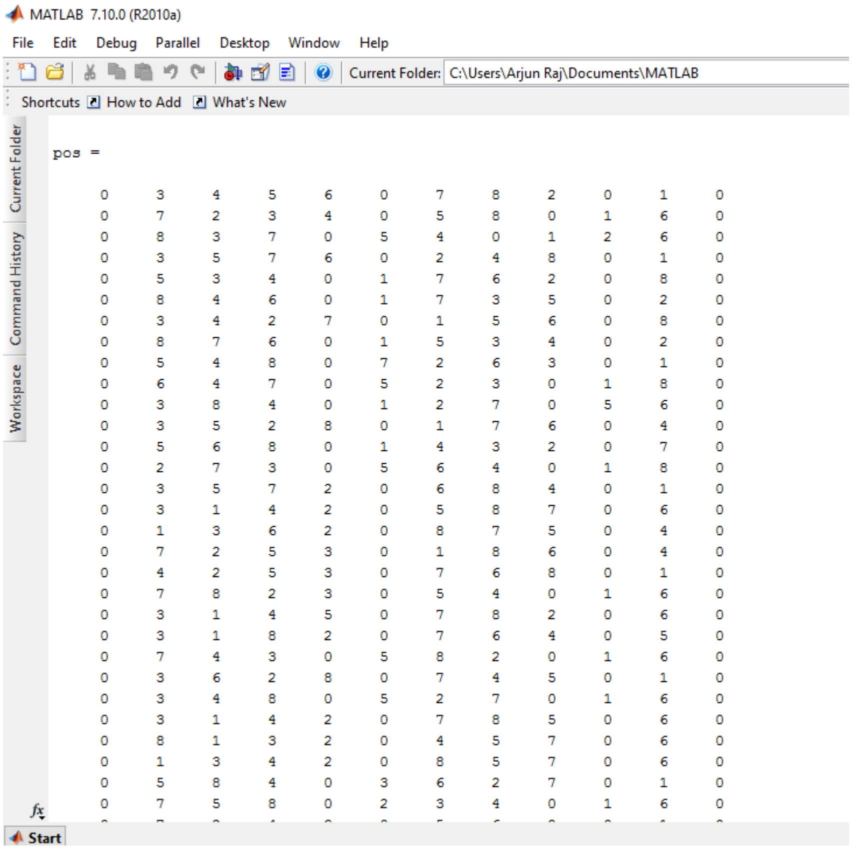

Fig. 4.13 Initial feasible particles

• If there is no feasible solution using any of the node in the abc array then terminate the vehicle.

• It should be noted that there might be cases in which the solution cannot be feasible based on the given sequence/priority. This time those particles are made into zeros and eliminated. Hence the initial solutions generated are in general more than that of the required nop.

• Make the Pbest matrix same as that of the pos and Gbest as the first solution in Pbest.

4.2.4.2 Decoding And Calculating The Objective Function/Fitness Value

The particle is decoded into the essential variables of the mathematical model needed to calculate the objective function and thus the fitness value/objective function is calculated for each and every particle.

In our case the objective function/fitness value is the total cost for the route got by the formula (1).

```
x(:,:,1) =

     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0
     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     1     0     0
     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

```
x(:,:,2) =

     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

```
x(:,:,3) =

     0     1     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

Fig. 4.14 x_{ijk} for a specific particle

```
si =

    4.0000    5.5000    7.5000    8.9000   10.4000         0         0
    3.2000    4.7000    6.2000    7.8000         0         0         0
    1.0000    1.8000         0         0         0         0         0


ans =

     0     5     6     4     3     0     7     2     8     0     1     0
```

Fig. 4.15 A particle mid-way through PSO

4.2.4.3 Finding Pbest For Each Particle

After finding the fitness value of the present particle, this is compared against the historical best of that particle and if it betters it then it replaces it, this done for all the particles.

4.2.4.4 Updating The Position And Velocity Vector

In general, the position vector is updated by adding the velocity vector with the current position vector. This leads to the values in the particle become real numbers but since the nodes are discrete and we need the values to be discrete.

It is essential to use convert those values into integers by truncating. This method does give us good results but the computational time and tendency to stagnate at local max/min is high. Hence in this project the List of Moves method is used in reference to the paper [11] Dr. Prasanna Venkatesan, S. et,al. If keenly observed the velocity vector is divided into three segments and in each segment it tries to take the present particle near to that segments best: Pbest /Gbest. In order to achieve the same goal but still to retain the integer values we use the list of moves method in which we find the number of moves required to reach towards Pbest or Gbest and randomly choose few of the swaps. The number of swaps to choose is based on the C1 and C2 values.

It can be illustrated as follow:

Let the Pos, Pbest & Gbest for the i^{th} particle in j^{th} iteration be as follow:

Velocity due to inertia: $w*V_{id}(t)$

To obtain this effect we generate a random number in the range of 1 to n and then chose those many pairs of array index (dimension index) to be swapped and they are swapped. In this case the number of swaps is 7 and the pairs are given below.

Pos:

0	3	4	2	0	1	5	6	0	7	8	0

List of swaps generated randomly

(3,4)	(2,6)	(9,3)	(5,7)	(8,4)	(6,7)	(10,2)

*

Fig. 4.16 Inertia velocity updation

Velocity due to global search: C_2*rand*[$gbest_{id}(t)-pos_{id(t)}(t)$]

This sect of the velocity tries to randomly figure the distance to reach towards the Pbest. So to simulate it we first find the number of moves to be made to make the present position vector (after the action of the inertia part) to the Pbest vector. That is indicated below and then generate a random binary array with C2 weighing in on the number of ones to be in it. Then all those swaps are done for which the corresponding binary value is 1.

Pos:

0	7	0	6	5	0	3	4	2	1	8	0

List of swaps required to convert the pos into Pbest

(2,9)	(11,3)	(5,4)	(5,7)	(8,9)

*

Binary for pbest simulating the C1*rand

1	0	1	0	0

Pbest

0	2	8	5	3	0	6	7	4	0	1	0

Fig. 4.17 Particle updation due to local search velocity

Velocity due to global search: C_2*rand*[$gbest_{id}(t)-pos_{id(t)}(t)$]

For the Gbest sect of the velocity we do the same as that of the Pbest sect. The pos used is the pos after the action of the both inertia and the Pbest.

Pos:

0	2	0	5	6	0	3	4	7	1	8	0

List of swaps required to convert the pos into Gbest

(10,2)	(7,3)	(5,11)	(7,8)	(10,9)	(10,11)

*

Binary for gbest simulating the C2*rand

0	0	1	1	1	0

Gbest

0	1	3	5	8	0	4	7	6	0	2	0

The updated particle:

0	2	0	5	8	0	4	3	1	7	6	0

*The values selected for swapping are not the nodes but the dimension number of the array

Fig. 4.18 Particle updation due to global search velocity

4.2.4.5 Repairing The Infeasible Solution

The repairing of the infeasible solutions is same as that of generating the initial solution except for when there is an infeasible solution. That particle value is made same as the Pbest of that particle.

4.2.4.6 Finding Gbest For The Present Iteration

After finding the Pbest for each and every particle, its best value is compared with the existing Gbest. If it is better than the existing one then it is taken as the Gbest.

4.2.4.7 Continuing till the ending criteria have reached

The end criteria in this project are chosen to be the number of iterations. When the number of iterations reaches N then the code stops.

CHAPTER 5

RESULTS AND DISCUSSIONS

The mathematical model, problem description and methodology employed have been covered in the previous chapters. This chapter discusses the results obtained in both the cases- using MS Excel Solver and using Particle Swarm Optimisation.

5.1 MS EXCEL SOLVER

The binary variable matrices for the two vehicles give the route to be taken so as to minimise travel cost.

vhs1								
j \ i	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0

vhs2								
j \ i	0	1	2	3	4	5	6	7
0'	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0

Fig. 5.1 Binary variable matrices

The route is extracted from Fig. 5.1 as follows:

1. For vehicle 1,
   - Locations of '1's in the binary table are identified
   - The corresponding arcs (i,j) are noted, where i denotes the column number and j denotes the row number.
   - The combinations obtained are: 0-6, 2-7, 4-2 and 6-4.
   - Individual arcs are joined to obtain a complete route from start depot 0 to end depot 7.
   - Route is: 0-6-4-2-7.
2. For vehicle 2,
   - Follow the same procedure as vehicle 1 to obtain these combinations: 0-5, 1-7, 3-1 and 5-3.
   - Route is: 0-5-3-1-7.

The values generated by Solver for w_{i1} and w_{i2} indicate the time at which vehicle 1 or vehicle 2 must start service at the corresponding node. These values may be obtained from Fig. 5.2.

	si	ai	bi	qi	wi1	wi2	di1	di2
0'	0	0	900	0	0	0	0	0
1	20	0	900	0	0	860	60	60
2	20	0	900	0	865	0	60	60
3	5	0	900	30	0	50	0	30
4	8	0	900	10	52	0	50	0
5	10	0	900	25	0	20	0	0
6	7	0	900	35	25	0	0	0
7	0	0	900	0	900	900	0	0

Fig. 5.2 Service start time

For e.g., from the figure we can say that vehicle 1 must start service at node 6 at time = 25 minutes. A value of 0 indicates that the vehicle does not service the corresponding node. So we can say that vehicle 2 does not visit node 2.

The data obtained from the above two figures may be consolidated into a single table, as shown below:

Table 5.1 Consolidated results using MS Excel Solver

Service start time (in mins)	0	20	25	50	52	860	865	900
Veh 1	0	-	6	-	4	-	2	7
Veh 2	0	5	-	3	-	1	-	7

This table thus shows the route taken by each vehicle and the time at which they start servicing the respective nodes.

5.2 MATLAB- PSO

The final Gbest and Gbestz (objective value) after 200 iterations is shown in Fig. 5.3. The mathematical model was implemented by coding in MATLAB. In order to solve any Vehicle routing problems with time windows, a similar code may be written.

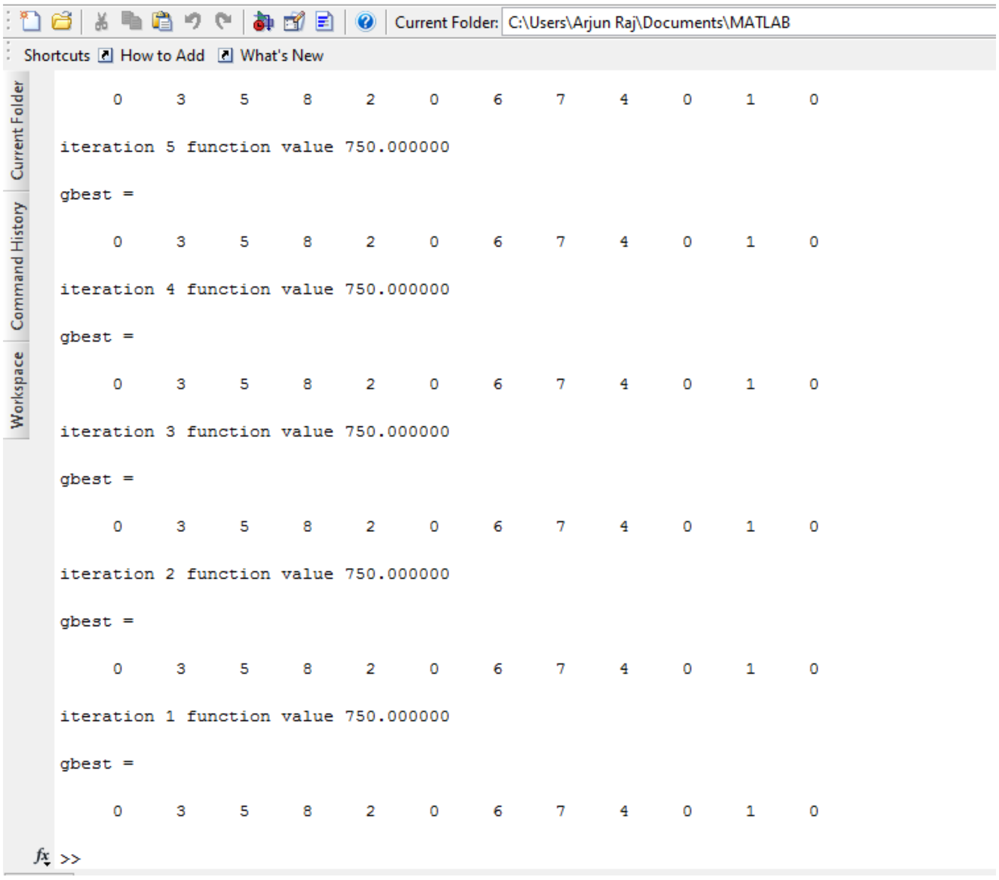

Fig. 5.3 MATLAB Results

CHAPTER 6

SUMMARY AND CONCLUSION

The results obtained after solving the two models- using MS Excel Solver and MATLAB have been shown in the previous chapter. In both cases, the optimal route is generated by minimising the travel cost

However, a direct comparison between the two results is not possible because of the basic difference in the structure of the models. Since the first model has separate nodes for disposal site and depot, it is possible in this case for the vehicle to dump its waste at a disposal site and continue on a different route before finally returning to the depot. This is not possible in the second model as the disposal site and depot are the same node, and the route terminates immediately after waste is dumped.

Another point of interest is that while MS Excel Solver may be regarded as a tool which is easier to use by someone without coding knowledge, its utility begins to drop with increasing problem complexity. In other words, as the size of the problem/ system being dealt with increases, Solver may not be the ideal tool to use.

While the difference may not be obvious in this study- as in this case we have dealt with small-sized problems- when the complexity of the problem increases, the advantages of using a combinatorial optimisation technique like PSO would stand out.

APPENDIX

A sample of the code written in MATLAB pertaining to the particle updation using list of moves is given below.

```
   %inertia swaps
   temp=0;
   inertiav=randi(n*w);
while (inertiav>=1)
swap1=randi([2,n+k]); swap2=randi([2,n+k]);
if(swap1==swap2)
swap2=randi([2,n+k]);
end
temp=pos(i,swap1);
pos(i,swap1)=pos(i,swap2); pos(i,swap2)=temp; inertiav=inertiav-1;
end
tpos=pos(i,:); swap=null(n+k-1,2);swp=1;
%pbest swaps V
 for pio=2:1:n+k;
for tp=pio+1:1:n+k;
if(tpos(1,tp)==pbest(i,pio))
swap(swp,1)=tp; swap(swp,2)=pio; swp=swp+1;
end
end
 end
 bin_p=round(rand(1,swp-1)+(c1/20));
for lop=1:1:swp-1;
swap(lop,:)=swap(lop,:)*bin_p(lop);
end
%pos updated with Vp
temp=0;
for wsp=1:1:swp-1;
if((swap(wsp,1)~=0)&&(swap(wsp,2)~=0))
    temp=pos(i,swap(wsp,1));pos(i,swap(wsp,1))=pos(i,swap(wsp,2));
pos(i,swap(wsp,2))=temp;
end
end
 %gbest swaps
 tpos=pos(i,:); swap=null(n+k-1,2);swp=1;
%gbest swaps V
 for pio=2:1:n+k;
for tp=pio+1:1:n+k;
if(tpos(1,tp)==gbest(1,pio))
swap(swp,1)=tp; swap(swp,2)=pio; swp=swp+1;
end
end
 end
 bin_p=round(rand(1,swp-1)+(c2/20));
for lop=1:1:swp-1;
swap(lop,:)=swap(lop,:)*bin_p(lop);
end
%pos updated with Vp
temp=0;
for wsp=1:1:swp-1;
if((swap(wsp,1)~=0)&&(swap(wsp,2)~=0))
    temp=pos(i,swap(wsp,1));pos(i,swap(wsp,1))=pos(i,swap(wsp,2));
pos(i,swap(wsp,2))=temp;
end
end
```

References

1. **Buhrkal, K., A. Larsen and S. Ropke** (2012) The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia- Social and Behavioral Sciences,* **39**, 241-254.
2. **Dantzig GB, Ramser JH** (1959) The Truck Dispatching Problem Stable. Management Science. 1959; 6(1):80–91**.**
3. **Geetha, S., G. Poonthalir and P.T. Vanathi** (2010) A Hybrid Particle Swarm Optimization with Genetic Operators for Vehicle Routing Problem. *Journal of Advances in Information Technology,* **1.4**, 181-188.
4. **Golden B, Assad A, Wasil E** (2002) Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth P, Vigo D, editors. The vehicle routing problem. SIAM; 2002 p. 245–86.
5. **Hai. S, Liu, T, Zhu, Y, Jin, L** (2009) Particle Swarm Optimization in Solving Vehicle Routing Problem. Second International Conference on Intelligent Computation Technology and Automation, 287-291
6. **Han, H. and E.P. Cueto** (2015) Waste Collection Vehicle Routing Problem: Literature Review. *Promet- Traffic & Transportation,* **27.4**, 345-358.
7. **Kim, B.I., S. Kim and S. Sahoo** (2006) Waste collection vehicle routing problem with time windows. *Computers & Operations Research,* **33**, 3624-3642.
8. **Li, J.Q., D. Borenstein and P.B. Mirchandani** (2008) Truck scheduling for solid waste collection in the City of Porto Alegre, Brazil. *Omega,* **36**, 1133-1149.
9. **Masrom, S., Z.Z.A. Siti, A.M. Nasir and A.S.A. Rahman** (2011) Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Time Windows. *Proceedings of 13th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems,* Wisconsin, USA.
10. **OECD** (1997) OECD Glossary of Environment Statistics, Studies in Methods. 1997
11. **Prasanna Venkatesan, S. and S. Kumanan** (2012) A Multi-Objective Discrete Particle Swarm Optimisation Algorithm for supply chain network design. *International Journal of Logistics Systems and Management,* **11.3**, 375-406.

12. **Wen, L. and F. Meng** (2008) An Improved PSO for the Multi-Depot Vehicle Routing Problem with Time Windows. *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application,* Vol. 1, 852-856.

13. **Wy, J., B.I. Kim and S. Kim** (2013) The rollon-rolloff waste collection vehicle routing problem with time windows. *European Journal of Operational Research,* **224**, 466-476.

14. **Zhu, Q., L. Qian, Y. Li and S. Zhu** (2006) An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows. *2006 IEEE International Conference on Evolutionary Computation,* Vancouver, BC, 1386-1390.