Complexity of Capacitated Vehicles Routing Problem Using Cellular Genetic Algorithms

Nora Salah Niazy and Amr Badr

Arab Academy For Science, Technology And Maritime Transport (AASTMT), Egypt

Summary
(Cellular Genetic Algorithms) CGAs are a subclass of Genetic Algorithms (GAS) it enhanced the population diversity and exploration in which the tentative solutions thanks to the existence of small overlapped neighborhoods .It well suited for complex problems as one of structured algorithms .The study was conducted on the behavior of these algorithms has been Performed in terms of quality of solutions exist, at the time of implementation, And a number of function evaluations effort. We have chosen the benchmark Augerat et al. Set A, Augerat et al. Set B and Augerat et al. Set P to test (cellular genetic algorithm) And compared with some other GAS. We have deceived that CGAs are capable of Always find optimal to the problem in a few times and reasonable.

Key words:
Vehicle routing problem, capacitated vehicles routing problem ,bin packing problem, travelling salesman problem.

1. Introduction

The Vehicle Routing Problem (VRP) is one of the most challenging combinatorial optimization task. More than 40 years ago, this problem is to design a set of the best ways for a fleet of vehicles to serve a particular group of customers. This interest in VRP is motivated by both its practical relevance and its considerable difficulty. The objective of the VRP is to deliver a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. In figures below we can see a picture of the input to a typical problem of VRP is one of the possible outcomes.

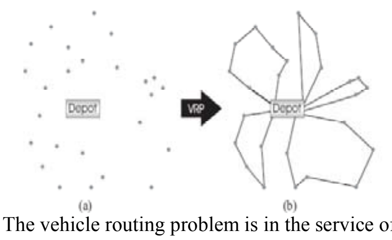

Fig. 1 The vehicle routing problem is in the service of a group of geographically and distributed to customers points from a depot using the roads a minimum cost [2, 9, 7, 10, 11, 12, 17, 18,20,21 ,22and23]

1.1 Features of the VRP

- Depots (number, location).
- Vehicles (capacity, costs, time to leave, driver, rest period, type and number of vehicles, max time).
- Customers (demands, hard or soft time windows, pickup and delivery, accessibility restriction, split demand, priority)
- Route Information (maximum route time or Distance) Cost on route [7, 8]

1.2 Problem definition

The VRP Is a known integer programming problem, VRP is NP-hard meaning that the computational effort required to solve this problem increases exponentially with the problem size. Due to its theoretical and practical interest (it has numerous real world applications, given that distribution is a major part of logistics and a substantial cost for many companies), the VRP has received a great amount of attention since its proposal in the 1950's. [5, 8, 9, 11, 15, 19, 23] Such problems it is often desirable to obtain approximate solutions, so you can find them fast enough and accurate enough for this purpose. Typically, this task is accomplished using different methods of heuristic, That rely on some insight into the nature of the problem The VRP arises naturally [Dantzing & Ramser 1959Central problem in the areas of transport and the distribution and logistics services . In some sectors of the market, and means of transport on the high proportion of value added goods [3, 15, 20] Therefore, the use of computerized methods of transport and often leads to significant savings ranging between 5% to 20% in the total costs as reported in [Toth & Vigo 2001][4,23] Usually, in the real world VRPs, and appeared a lot of restrictions on the side.

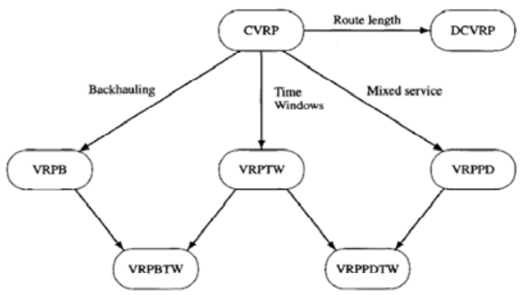

Fig. 2 The basic problems of the VRP class and their interconnection

Manuscript received February 5, 2012
Manuscript revised February 20, 2012

1.3 Some of the most important constraints

- Each vehicle has Capacitating limited (Capacitated VRP – CVRP)
- All customers must be provided within a certain time frame and (VRP with time windows – VRPTW)
- vendor uses several depots to provide customers (multi-depot VRP – MDVRP)
- Customers may return of some goods to the depot (VRP with pickup and Delivering- VRPPD)
- The customers Can serve from different vehicles (Split Delivery VRP – SDVRP)
- Some values such as the number of customers, and the demands of them, service time or travel time is random (stochastic VRP - SVRP)
- The deliveries may be done in some days (Periodic VRP - PVRP) [1, 8, 9, 15, 17, 20, 23]

The paper is arranged in the following manner .Section 2 gives a mathematical Formulation for the VRP .Our CGA is described in Sect .3, Section 4 presents the Results of our algorithms, and compares them with those of some heuristics in the Literature .The conclusions and some future work lines are presented in Sect .5.

- Related Work

A number of formulations have been proposed for this NP-hard Problem .for a comprehensive account we refer to the survey papers of Christofides et al. (1979),Bodin et al. (1983),Christofides (1985) Laporte and Nobert (1987) (exact algorithms),Golden and Assad (1988) Laporte (1992),Fisher (1995),Laporte and Osman (1995) (bibliography) Laporte (1997) (annotated bibliography)
Toth and Vigo (1998) (exact algorithms) and Golden et al. (1998) (heuristics),Toth and Vigo (2002) (first six chapters),Cordeau et al. (2007)

2. Vehicle Routing Problem's Formulation

The VRP is a combinatorial problem whose ground set is the edges of a graph G(V,E).[Among the different variants of VRP we work here with the Capacitated VRP (CVRP), in which every vehicle has a uniform capacity of a single commodity.

2.1 The CVRP is defined on an

Undirected graph G = (V,E) where V = {v0, v1, . . , vn } is a vertex set and E = {(vi, vj)/vi, vj ∈ V, i < j} is an edge set. Vertex v0 stands for the depot, and it is from where m identical vehicles of capacity must be able to serve all the cities or customers, representative of a group of n vertices {v1, . . . , vn}. We determine on E (a non-negative cost) distance matrix C = (Cij) between customers vi and vj. Let V1, . . . ,Vm be a partition of V, a route Ri is a permutation of the customers in Vi specifying the order of visiting them, starting and finishing at the depot v0.
The cost of a given route Ri ={vi0, vi1, . , vik+1}, where VIJ ∈V and vi0 = vik+1 = 0 (0 denotes the depot.[1,2,7,8,9,11,12,13,14,15,18,19,21], is given by:

$$Cost(R_i) = \sum_{j=0}^{k} c_{j,j+1}$$

And the cost of the problem solution (S) is:

$$F_{CVRP}(S) = \sum_{i=1}^{m} Cost(R_i)$$

2.2 The problem CVRP consists of determining a set of m vehicle routes:

1. Minimum total cost.
2. Starting and ending at the depot v0 .
3. Each customer is visited exactly once by exactly one vehicle; subject to the restrictions
4. The total demand of any route does not exceed.

$$Q(\sum_{v_j \in R_i} q_j \le Q) \qquad \text{(Equation .1)}$$

5. The total distance of any route is not larger than a preset bound

$$D(Cost(R_i) \le D) \qquad \text{(Equation .2)}$$

6. All vehicles have the same and carrying capacity of one type of commodity.[2,7,12,13,21,22,23]

The number of vehicles as a value or variable input resolution. In this study, the length of routes Is minimized independently of the number of used vehicles in a TSP its clear from the description of us that VRP is closely related to two combinatorial difficult problems from other side can take an instance of the Multiple Travelling Salesman Problem (MTSP) just by but Q = ∞. An MTSP instance can be transformed instance by adjoining to the graph k − 1 additional copies of node 0 (depot) and its incident edges there are no edges among the k depot nodes. On the other side, the question of whether there is a solution possible to a certain moment of the VRP is Instance of the Bin Packing Problem (BPP. Therefore, it is very difficult to VRP To be solved in practice because of the interaction of these two underlying difficulty Models)TSP and BPP .(Actually More cases can be solved and from the VRP Two in terms of size smaller than those TSP[1,6,7,11,13,14,15,19,23].

3. Cellular Genetic Algorithms

CGA algorithms that uses to solve VRP in this paper and Genetic Algorithms GAS which make a good comparative between them using different instance and discuss them simply using CGA with specific recombination and mutation operators which improve the solution.

Algorithm 1.1 Pseudo-code of a canonical CGA

```
.proc Evolve)cga // (Parameters of the algorithm in 'cga'
.GenerateInitialPopulation)cga.pop(;
.Evaluation)cga.pop(;
.while !StopCondition ()do
.for individual ← 1 to cga.popSize do
.neighbors ←
CalculateNeighborhood)cga,position)individual((;
.parents ← Selection)neighbors(;
.offspring ← Recombination)cga.Pc,parents(;
.offspring ← Mutation)cga.Pm,offspring(;
.Evaluation)offspring(;
.Replacement)position)individual(,auxiliary pop,offspring(;
.end for
.cga.pop ← auxiliary pop;
.end while
.end proc Evolve
```

3,1 Proposed Solution

Provide pseudo-code for a canonical CGA. As can be seen it, Begins with the generation and preliminary assessment of the population, Then and The previously mentioned genetic operators) selection, recombination, mutation, and replacement (Applied recursively to each individual until the end of and met the requirement. A key issue is characterized by cellular model is that these Genetic operators are applied within the neighborhood of individuals, so Does not allow for individuals who belong to different neighborhoods on the interaction.
According to CGA, As Is stored individuals composing the population of the next generation in to help people, and when you are finished, replace the first step in the atomic existing population. So that in this model, all individuals in the population is updated at one time, and equivalent, and the establishment of individuals Consists only of individuals in current population (This is not the former Created during the same iteration). [2]
The fitness value assigned to individuals is computed as follows

$$f_{eval}(S) = F_{CVRP}(S) + \lambda.overcap(s) \quad \text{(Equation.3)}$$

The Objective is to minimizing our $f_{eval}(S)$ in (Equation.3). Punishable by the fitness value only just in case that the capacity of any vehicle and/or at any time is exceeded, of any route. [2]

Functions 'overcap(S)' return the overhead in capacity of the solution with respect to the maximum allowed value for each route. These values returned by `overcap(s) is weighted by multiplying it by factors λ. In this work λ =1000 . [9]

3.2 Problem Representation

In a GA, each individual acting one candidate solution. Candidate solution must be on an instance of CVRP determined the number of vehicles required, the allocation of the demands of these vehicles, as well as arrange for deliveries all the way. Representation consists of flipping True numbers. Each will contain all the flipping clients and the way Lines (the demarcation of different ways) Therefore, we will use a permutation of numbers [0. . . N - 1] with length n = k +c to a solution for CVRP with c customers and up to a maximum of k + 1 possible ways.

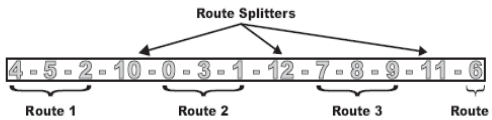

Fig. 3 Individual representing a solution for 10 customers and 4 vehicles

Is the representation of clients with numbers [0. . . C - 1], while the route splitters belong to the road Domain [c. . . N - 1]. That, given the nature of the chromosome (Note flipping Numbers must be an integer) lines the way will be different numbers, although It should be possible to use the same number through the appointment of lines When you use other possible chromosome configuration.
Each path consists of customers between two route splitters of the individual. Such as in the fig. 3 we plot representing the individual possible solution to the instance of a virtual CVRP with clients who use 10 A maximum of 4 vehicles. Values [0,. . . 0.9), while representing clients (10,..., 12] the route splitters. Route 1 begins in the depot, and customer visits 4-5-2 (In that order), and return to the depot. It goes through route 2 of the depot for 0-3-1 customers and revenue. The vehicle from Route 3 starts at the depot and Customer visits 7-8-9. Finally, on Route 4, and visiting only customer 6 from depot. And allows the empty roads in this representation by simply placing Adjacent route splitters are the way to any client without their.[2,9,11]

3.3 Recombination

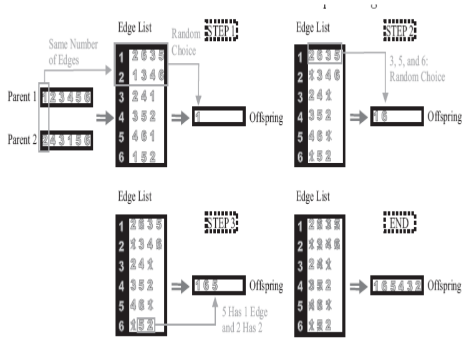

Fig. 4 Edge recombination operator(ERX)

It uses GAS and unites in the engine to combine parts in the second (or Patterns in order to transfer more (Hoping) a good building blocks which and their offspring. Recombination operator, which we use is the recombination edge Operator (ERX)], where it was reported to a large extent in most Suitable for comparison with other public operators permutations such as the crossover (OX or partially matched crossover (PMX).ERX build a offspring While preserving the edges of the two parents. Therefore, use the list of edge. This list includes, for each city, the edges of parents to begin or end in that (see fig 4)

After you create a list of the edge of the parents and two ERX algorithm one solution builds children before proceeding as follows. The first gene of offspring be selected between the first and one of both parents. In particular, there are fewer of the edges of the selected gene. In case of a tie, the first will be selected genes from the original first. And other genes are selected by taking among their neighbors and one with a shorter list edge. Relations Broken by choosing the first city I found that to meet this standard shorter list. Once the gene is selected, it is removed from the list [2,9,21]

3.4 Mutation

Algorithm 1.2 The mutation algorithm

```
.proc Mutation)pm, ind(
//'pm' is the mutation probability, and 'ind' is the individual to mutate
.for i ← 1 to ind.length ()do
.if rand0to1 ()<pm then
.r =rand0to1 ();
.if r < 0.33 then
.ind.Inversion)i, randomInt)ind.length(((;
.else if r > 0.66 then
.ind.Insertion)i, randomInt)ind.length(((;
.else
.ind.Swap)i, randomInt)ind.length(((;
.end if
.end if
.end for
.end proc Mutation
```

Mutation operator that we use in our systems it rep-resents an important base in the context of the deve- lopment because it is responsible for the introduction of a high degree Diversity in each generation And Mutation consists of the application of the insertion or a reflection of swaps Genes on an equal basis with the possibility. That three mutation operators (see Fig. 5)

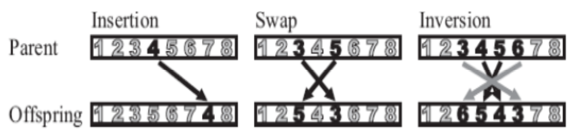

Fig. 5 The three different used mutation

Combining three are in the new combined operator. Operator insertion identifying the gene (either the client or route splitter, and the inclusion of that elsewhere have been randomly selected from the same individual. Swapping) lies in the random selection of two genes in a solution and swaps. Finally, a reflection of Unlike visit The order of genes between two points have been randomly selected from permutation .Observe that the changes caused may occur within or between the route, Way in all the three operators. And Stated in formal terms, given a potential solution $S = \{s1, \ldots, sp-1, sp, sp+1, \ldots, sq-1, sq, sq+1, \ldots, sn\}$, where p and q are randomly Selected indicators, n is the total number of customers in addition to Obtain the number of route splitters through)n = k +c(, Then get the new solution after the application of S all the mechanisms proposed different is shown below:

Insertion : $S = \{s1, \ldots, sp-1, sp+1, \ldots, sq-1, sq, sp, sq+1, \ldots, sn\}$,

Swap : $S = \{s1, \ldots, sp-1, sq, sp+1, \ldots, sq-1, sp, sq+1, \ldots, sn\}$,

Inversion : $S = \{s1, \ldots, sp-1, sq, sq-1, \ldots, sp+1, sp, sq+1, \ldots, sn\}$.[2,,9,11,12]

4. Experimentation

In this section we describe the experiments applying these ideas to a set of problems taken from a [the web site]. The chosen algorithms have been implemented in Java on a 2.5 GHz PC (windows XP service back II operating system). They have been tested with instances C1, C2, C3, C4, C5, C6, C7, C8, and C9 as illustrated in table [1].

Table 1 problems definitions

	problem name	Nodes number	vehicles number	Type
C1	A-n32-k5	32	5	A
C2	A-n36-k5	36	5	A
C3	A-n39-k6	39	6	A
C4	B-n31-k5	31	5	B
C5	B-n35-k5	35	5	B
C6	B-n41-k6	41	6	B
C7	P-n16-k8.vrp	16	8	P
C8	P-n19-k2.vrp	19	2	P
C9	P-n22-k8.vrp	22	8	P

As illustrated in table [1], the first three instances are CVRP of type Augerat et al. Set A with customers number equal to 32, 36, and 39 respectively and vehicles number 5, 5, and 6 respectively. The second three instances are CVRP of type Augeat et al. Set B with customers number equal to 31, 35, and 41 respectively and their vehicles number are the same as the first three instances. The last three instances are CVRP of type Augerat et al. Set P with customers number equal to 16, 19, and 22 respectively and their vehicles number are 8, 2, and 8 respectively.

We have used two different algorithms, the first one is panmictic genetic algorithm and the second one is cellular genetic algorithm. The difference among them is just the structured population that used in cellular GA.

The parameterization of the studied algorithms is described in table [2]. For each individual, the recombination probability is set to 0.65 in all of our algorithms. The mutation probability for an individual is 0.85, and its alleles are mutated with probability $1/L$, being L the length of the chromosome. The neighborhood used in the cellular case is compact 13 (C_{13}). Population size is 100 individuals in any case, and we perform 100 independent runs to get reliable statistical results.

4.1 Cellular versus Panmictic GAs

The cellular GA is compared versus the panmictic GA in terms of evaluations to find a solution, obtained fitness values, time, and hit rate (percentage of successful runs) in table 3, 4, 5, and 6. In table 3, 4, and 5, the X column refers to the mean value while the other two columns refer to the maximum and minimum values.

Parameterization used in cellular case

Table 2 of Parameterization	
Population size	100 individuals(10*10)
Parent selection tournament	Current individual+ binary
Recombination	ERX,pc=0.65
Mutation of individuals Inversion(same Prob),Pm=0,85	Insertion ,Swap or
Bit mutation probability Length)	Pbm=1.2/L(L=Individual
Replacement	Rep-if-not-worse
Neighborhood	Compact 13(C13)

Table 3 fitness values

	Algorithm	X	Max	Min
C1	**Cellular**	798.62	1235	774
	Genetic	838.29	976	775
C2	**Cellular**	840.44	1564	792
	Genetic	857.91	967	796
C3	**Cellular**	864.24	1793	815
	Genetic	877.53	989	823
C4	**Cellular**	687.97	1071	670
	Genetic	685.75	720	670
C5	**Cellular**	981.8	1512	948
	Genetic	978.24	1068	951
C6	**Cellular**	929.07	4699	820
	Genetic	900.36	1120	823
C7	**Cellular**	444.57	455	444
	Genetic	449.72	481	444
C8	**Cellular**	210.81	289	206
	Genetic	226.26	258	206
C9	**Cellular**	598.14	626	590
	Genetic	636.33	762	590

As it can be seen in table 3, cellular GA is more efficient that panmictic GA, since it beats the panmictic GA in all the chosen instances in the average of the obtained fitness. Although cellular GA overcame panmicit GA in fitness values, panmicit GA overcame cellular GA in both average evaluations number and average time as illustrated in table 4 and 5. The obtained results is quiet rational since the overlapped small neighborhoods of cellular GA helps exploring the search space because the induced slow diffusion (which affect the results in table 4 and 5) of solutions through the population provided a kind of exploration (diversification), while exploitation (intensification) takes place inside each neighborhood by genetic operations. In case of panmicit GA, the diffusion of solutions is faster than cellular GA which leads to increasing exploitation and that implies to stucking in local optimum and finishing in shorter time with lower number of evolutions as illustrated in table 4 and 5.

Table 4 evaluations

	Algorithm	X	Max	Min
C1	**Cellular**	7.79E+05	1.38E+06	7.52E+04
	Genetic	3.58E+04	7.12E+04	1.41E+04
C2	**Cellular**	3.31E+06	5.00E+06	2.30E+04
	Genetic	41,468.14	68806	25048
C3	**Cellular**	2.69E+06	3.73E+06	3.72E+04
	Genetic	4.77E+04	8.61E+04	2.57E+04
C4	**Cellular**	8.92E+05	2.79E+06	1.10E+04
	Genetic	3.22E+04	5.60E+04	1.94E+04
C5	**Cellular**	1.12E+06	1.57E+06	8.22E+04
	Genetic	4.20E+04	7.33E+04	1.76E+04
C6	**Cellular**	6.14E+06	1.00E+07	1.65E+04
	Genetic	5.73E+04	1.18E+05	3.52E+04
C7	**Cellular**	1.85E+05	1.78E+06	1.94E+04
	Genetic	1.64E+04	2.41E+04	1.16E+04
C8	**Cellular**	1.18E+05	7.08E+05	1.16E+04
	Genetic	1.77E+04	3.29E+04	4.56E+03
C9	**Cellular**	6.30E+05	1.15E+06	2.13E+05
	Genetic	2.15E+04	3.96E+04	7.53E+03

Table 5 time

	Algorithm	X	Max	Min
C1	**Cellular**	43,404.83	74390	4250
	Genetic	2,102.36	4578	828

C2	**Cellular**	201,698.99	331282	1500
	Genetic	2,367.27	3875	1437
C3	**Cellular**	91,942.98	2393156	531
	Genetic	2,924.56	5156	1578
C4	**Cellular**	10,516.22	32062	141
	Genetic	1,642.31	2844	984
C5	**Cellular**	14,703.28	20437	1109
	Genetic	2,380.90	4094	1015
C6	**Cellular**	90,659.67	147281	235
	Genetic	3,584.89	7453	2203
C7	**Cellular**	8,806.43	86891	891
	Genetic	449.20	656	313
C8	**Cellular**	3,074.72	17485	297
	Genetic	666.85	1219	172
C9	**Cellular**	21,428.44	38015	7047
	Genetic	1,038.89	1875	360

4.2 Comparison our results against best known results

In this section we compare our obtained results against the best known results in the literature, as any rigorous work should do. As it shown in table 6 both cellular and panmicit GA got better results than the best know for all the chosen instances except for *C7*. In the last three instances (*C7, C8,* and C9), while both cellular and panmicit got the same fitness values, but the hit rate (number of times the algorithm got fitness value equal or better than the best known value) for cellular is quite higher than panmicit. In the other instances (*C1, C2, C3, C4, C5,* and *C6*), cellular GA has overcome panmicit GA.

Table 6 best results

	Algorithm	Best fitness	Hit rate	Best known
C1	**Cellular**	774	85	784
	Genetic	775	10	
C2	**Cellular**	792	71	799
	Genetic	796	1	
C3	**Cellular**	815	87	831
	Genetic	823	6	
C4	**Cellular**	670	57	672
	Genetic	670	6	
C5	**Cellular**	948	82	955
	Genetic	951	13	
C6	**Cellular**	820	85	829
	Genetic	823	6	
C7	**Cellular**	444	0	435
	Genetic	444	0	
C8	**Cellular**	206	91	212
	Genetic	206	5	
C9	**Cellular**	590	98	603
	Genetic	590	27	

5. Conclusion and future work

In this paper we present the problem of capacitated vehicle routing problem (CVRP) by using cellular Genetic Algorithm. CVRP is an integer programming problem which falls into the category of NP-hard problems with the goal of minimizing the total distances. We also compare the behavior of cellular GA against panmicit GA in solving CVRP. We obtained that the cellular GA is more exploitative than panmicit GA since it got a quite higher hit rate (robustness). On the other hand, panmicit GA finds the solution to most CVRP instances with less evaluations numbers than cellular GA but with lower hit rate, since it easily gets stuck in local optima.
As a future work we intend to use the cellular GA in solving more complex problems. We also intend to use the parallel cellular GA in solving CVRP in order to reduce the time needed to find an optimum solution.

REFERENCES

[1] http//:neo.lcc.uma.es/radi-aeb/WebVRP/.
[2] Cellular Genetic Algorithms by Enrique Alba, Bernabe Dorronsoro ISBN Number :0387776095, 9780387776095, 978-0387776095.
[3] G .B .Dantzig and R.H .Ramser" .The Truck Dispatching Problem ."Management Science 6, 80–91 .1959.
[4] P .Toth, D .Vigo" :The Vehicle Routing Problem ." Monographs on Discrete Mathematics and Applications . SIAM, Philadelphia .2001.
[5] J.K .Lenstra and A.H.G .Rinnooy Kan .Complexity of vehicle routing and scheduling problems .Networks, 11:221–227, 1981.
[6] T.K .Ralphs, L .Kopman, W.R .Pulleyblank, and L.E . Trotter Jr .On the capacitated vehicle routing problem . Mathematical Programming Series B,94:343–359, 2003.
[7] Vehicle Routing Problems)VRPs(Technische Universiteit Eindhoven .Short Course; November 28-29-2000.
[8] The Vehicle Routing Problem/edited by Paolo Toth, Daniele Vigo p .cm .—SIAM monographs on discrete mathematics and application includes bibliographical references and index ISBN 89871-579-2.
[9] Engineering evolutionary intelligent systems /edited by Ajith Abraham .Crina Grosan .Witold Pedrycz)Eds (ISBN 978-540-75395-7.
[10] Parallel and serial algorithms for Vehicle Routing Problems Christopher Groer, Doctor of Philosophy, 2008 Dissertation directed by :Professor Bruce Golden Robert H .Smith School of Business.
[11] GVR :a New Genetic Representation for the Vehicle Routing Problem Francisco B .Pereira1,2, Jorge Tavares2, Penousal Machado1,2, Ernesto Costa2.
[12] Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms Enrique Alba1 and Bernab´e Dorronsoro2.
[13] What You Should Know about the Vehicle Routing Problem Gilbert Laporte.
[14] Capacitated Vehicle Routing and Some Related Problems Ted Ralphs, Joe Hartman ,Matt Galati Industrial and

Systems Engineering Lehigh University Rutgers University, November 27, 2001.
[15] Vehicle routing and scheduling by Martin savelsbergh the logistics institute Georgia institute of technology.
[16] A new bi-level formulation for the Vehicle Routing Problem and a solution method using a genetic algorithm Yannis Marinakis · Athanasios Migdalas · Panos M .Pardalos.
[17] Introduction to VRP 1 .Basic concepts lecture notes Daniele Vigo University of Bologna Dept .of Electronics, Computer Science and Systems) deis (and II Faculty of Engineering 1. rev 1.0, September 11, 2007.
[18] The Vehicle Routing Problem latest advances and new challenges Bruceg golden ,s .raghaven, Edward wasil Editors ISBN 978-0-387-77777-1.
[19] Handbook on modeling for discre optimization. Edited by Gautam Appa Operational Research Department London School of Economics leonidas pitsoulis department of Mathematical and Physical Sciences- Aristotle University of Thessaloniki H.Paul Williams Operational Research Department London School of Economics Sprin
[20] http://en.wikipedia.org/wiki/Vehicle_routing_problem.
[21] http://neo.lcc.uma.es/cEA-web/VRP.htm.
[22] http://www.idsia.ch/~monaldo/vrp.html.
[23] http://www.computational-logistics.org/computation/index.php/Capacitated_VRP-CVRP.