© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 9, No. 2, May-Aug 2018, pp. 41-54. ISSN: 2007-1558.

Comparison between instances to solve the CVRP

Edgar Gonzalo Cossio Franco[1], José Alberto Hernández Aguilar[2],
Alberto Ochoa-Zezzatti[3] & Julio César Ponce Gallegos[4]
[1]*Universidad Da Vinci, Universidad Enrique Díaz de León, Mexico.*
[2]*Universidad Autónoma del Estado de Morelos, Mexico.*
[3]*Universidad Autónoma de Ciudad Juárez, Mexico.*
[4]*Universidad Autónoma de Aguascalientes, Mexico.*
kofrran@gmail.com, jose_hernandez@uaem.mx,
alberto.ochoa@uacj.mx, julk_cpg@hotmail.com

Abstract. The Vehicle Routing Problem or VRP is an approach represented by the problems that faces a vehicle to transport goods on a route (origin-destination) under a defined time and distance. An instance is a set of data prepared specifically in order for analysis and exploration (Column 1 indicates the number of nodes, usually the number 1 is the depot, column 2 is the x coordinate, column 3 is the y-coordinate, column 4 is the demand to be covered by the node) that it was done in MATLAB R2014a software which runs the algorithm VRP with capacities (CVRP), with the structure already mentioned. For their analysis was necessary to use instances obtained from NEO. This paper presents a comparative between instances to solve CVRP and determine which one offers the best solution.

Keywords: CVRP, NEO, Instances, MATLAB.

1 Introduction

The VRP is a complex problem, which aims to find the shortest route taking all points destination [1]. It was proposed by Dantzig and Ramser 57 years ago, the VRP is a very important field of study of scientist and everyone who is in the transportation and logistics.

Several works has been carried trying to solve the problem of vehicle routing as described in [13] where is implemented a genetic algorithm with the purpose to optimize routes of vehicle. In [8] Ponce et al. presented a VRP solution where an Ant Colony Algorithm is implemented a parallelized in CUDA.

By reviewing the literature [8,9] is clear the objective of VRP is to find the best way that provides the best route, time, and less effort of the units. There are various approaches for study.

Received Jan 4, 2018 / Accepted Feb 10, 2018

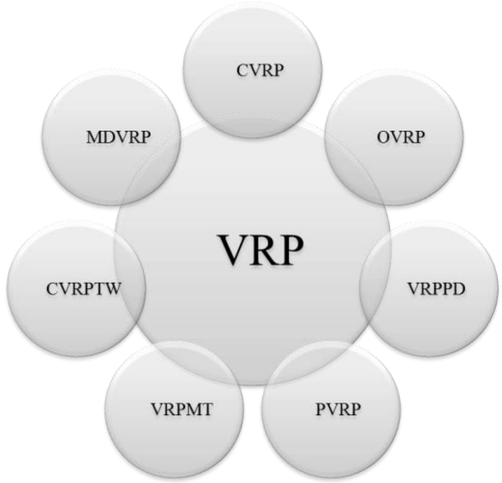

Figure 1. VRP and its different approaches [9]

In the case of the VRP with capacities (CVRP) is counted with a special twist: the ability to load is carried by a driver from a source to a destination. In this context, the CVRP becomes a more specific case of study (in addition to the distance, route and time). The CVRPTW covers the same problem of capacity but the new variant: Delivery of goods on a defined time. This variant involves more effort and commitment in planning, operator units and units.

The MDVRP covers the same capacity problem adding the possibility of multiple deposits, unlike the VRP where there is only one. Originally the VRP provides a single deposit of which the distribution is initiated; in this variant of VRP with multiple deposits, should be carefully managed which customers will be sent from which deposit to improve delivery times and overall route. The OVRP has the same objective as the CVRP but does not return the deposit after serving all customers on the route [10].

The VRPPD includes the possibility that some customers return commodity and this complicates the logistics since each unit must have extra space to receive the goods. The PVRP. In the classical VRP planning is performed per day, in this scheme periodic planning where customers are served with some frequency is performed.

The VRPMT considers that each vehicle can make several trips during the day, covering the route to serve every customer [11]. In the Figure 2 is shown at the left side of figure a deposit and several delivery points and the second image to the right shows (using the VRP) possible routes of solution [2].

The contribution of this study is the comparison and analysis of four instances of NEO [9] with the purpose of evaluating how a solution of VRP using a nearest neighborhood algorithm implemented in MATLAB performs, so it is possible to distinguish as a result the total distance, roads and vehicles in the best instance and its numerical table and visual distribution of the route, as well as the visited points.

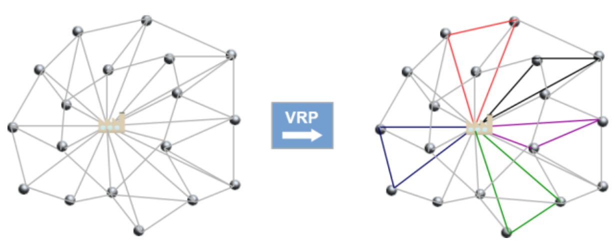

Figure 2. The VRP [2]

2 Formal definition of CVRP

The formal definition for the CVRP is shown next [8]:

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V \setminus \{0\}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad i \in V \setminus \{0\}$$

$$\sum_{i \in V} x_{i0} = K$$

$$\sum_{j \in V} x_{oj} = K$$

$$x_{ij} \in \{0,1\} \qquad \mathrm{i,j} \in V$$

$$\sum_{i \in S} \sum_{j \in S} xij \geq r(S), \qquad \forall S \subset V \setminus \{0\}, \quad S \neq \phi$$

3 Methodology

For this work, the Nearest Neighbor Algorithm (NNA) is used. We use the software package created by Sas Wahid Hamzad [15], composed by three files (NIAlgVRP: Nearest Insertion Algorithm for Solving Vehicle Routing Problem, ConstEvalVRP: Constraint Evaluation of VRP, Problem: CVRP problem consists of n nodes, demands, and x and y coordinates positions), each one has scripts in M (Matlab) language.

The following figure shows the nearest neighbor algorithm.

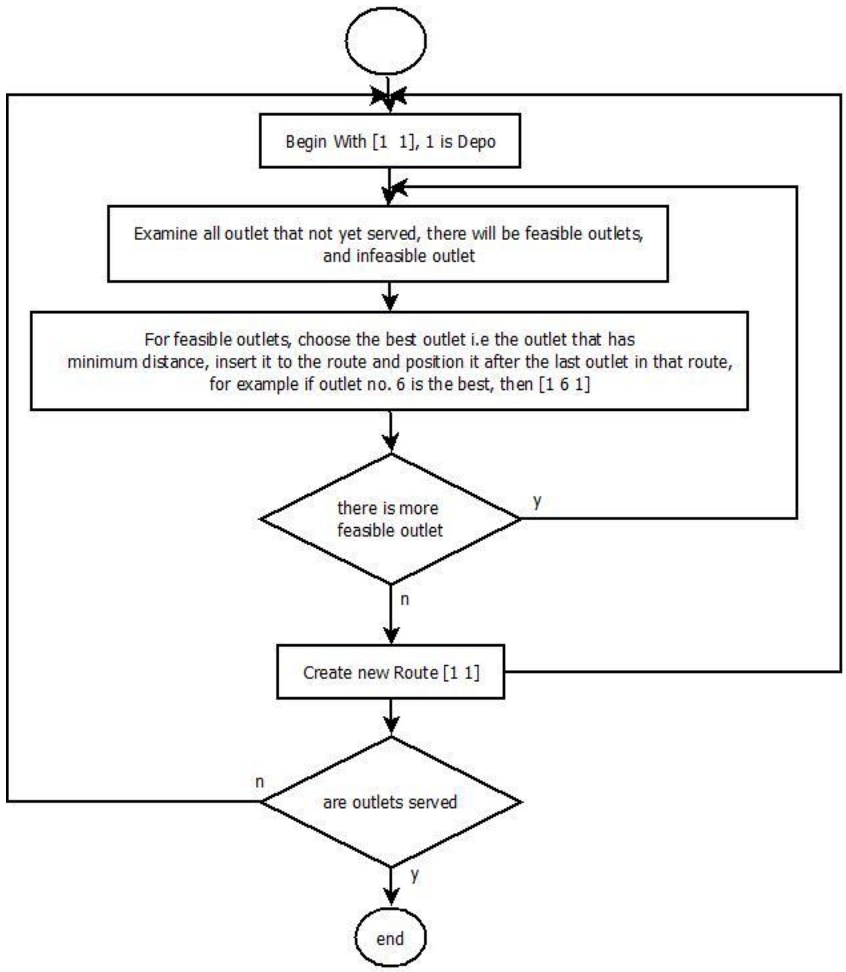

Figure 3. Nearest Neighbor Algorithm Details

We work with instances of Augerat et al. [14]. For this research four instances and optimal solutions were selected. The aim is to compare instances of Augerat et al. by using the NNA algorithm to solve CVRP implemented in MATLAB, and determine whether is it possible to further optimize the optimal solutions proposed by Augerat et al. [14].
The instances that we selected were:

- opt-A-n32-k5
  - NAME : A-n32-k5
  - COMMENT : (Augerat et al, Min no of trucks: 5, Optimal value: 784)
  - TYPE : CVRP

  - DIMENSION : 32
  - EDGE_WEIGHT_TYPE : EUC_2D
  - CAPACITY : 100

- opt-A-n33-k5
  - NAME : A-n33-k5
  - COMMENT : (Augerat et al, Min no of trucks: 5, Optimal value: 661)
  - TYPE : CVRP
  - DIMENSION : 33
  - EDGE_WEIGHT_TYPE : EUC_2D
  - CAPACITY : 100

- opt-A-n33-k6
  - NAME : A-n33-k6
  - COMMENT : (Augerat et al, Min no of trucks: 6, Optimal value: 742)
  - TYPE : CVRP
  - DIMENSION : 33
  - EDGE_WEIGHT_TYPE : EUC_2D
  - CAPACITY : 100

- opt-A-n34-k5
  - NAME : A-n34-k5
  - COMMENT : (Augerat et al, Min no of trucks: 5, Optimal value: 778)
  - TYPE : CVRP
  - DIMENSION : 34
  - EDGE_WEIGHT_TYPE : EUC_2D
  - CAPACITY : 100

In above tables, column 1 represents the number of route or idroute, column two the x coordinate, column three the y coordinate and column four represents the demand. For this research was used as a tool for analysis and modeling MATLAB R2014a version. NEO instances above mentioned were preprocessed and refined, by using Microsoft Excel 2010. The hardware used was a HP Pavilion Gaming computer with an Intel CORE i5 processor and Windows 10 Home edition 64 bits, and NVIDIA GEFORCE GTX 950M video card. Four instances of data from Networking and Emerging Optimization (NEO) are compared. Four elements were identified:

1) The depot
The origin of the vehicles
2) Coordinate x
The point in the plane (horizontal)
3) Coordinate y
The point in the plane (vertical)
4) Demand

Demand of outlets

File containing these instances was created. Function to calculate the total distance of the route is created, as well.

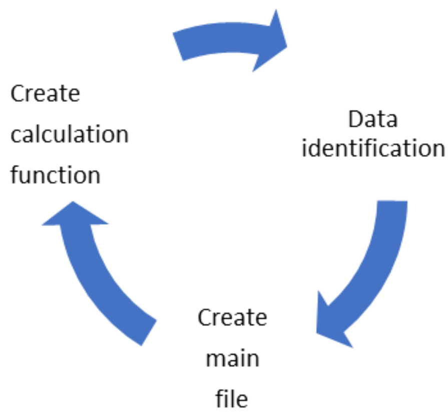

Figure 4. Process to implement the algorithm

4 Methodology

The total distance is calculated, the number of routes for each execution and the number of vehicles.

a) NEO instance whose data are processed and sorted is downloaded.
b) The instance is processed into a spreadsheet in the order requested by the arrangement required in Problem.
d) Matlab is started
e) Open Current Folder section (left side) load the folder with the 3 scripts described in section 2 (NIAlgVRP, ConstEvalVRP, Problem).
f) The script Problem opens.
g) The data is loaded to the script Problem.m
h) Once the data is loaded, right click on the file Problem.m and of the pop-up menu select the Run option.
i) In section Command Window the result of the implementation of the code appears.
j) In the command line in the same section a Command Window call function to graph the data, i.e. the NIAlgVRP function (Problem, n) where n equals the capacity – in this research we consider as capacity the number of vehicles with which it is intended to make the experiment, for this work was subjected to experimentation three scenarios: 30, 300 and 600 vehicles.
k) In section Command Window the result of the implementation of the code appears showing the number of routes, the total distance and the graph with the source and destinations, i.e. routes (one color for each route).

The steps of the experiment are:

1. NEO instances were downloaded.
2. Debug them in a spreadsheet. They set to be processed by the algorithm
3. Load the data file that sends for the main algorithm
4. Execute the algorithm and the data of NE
5. Once shown the results of the algorithm, export the graph with the X and Y points.

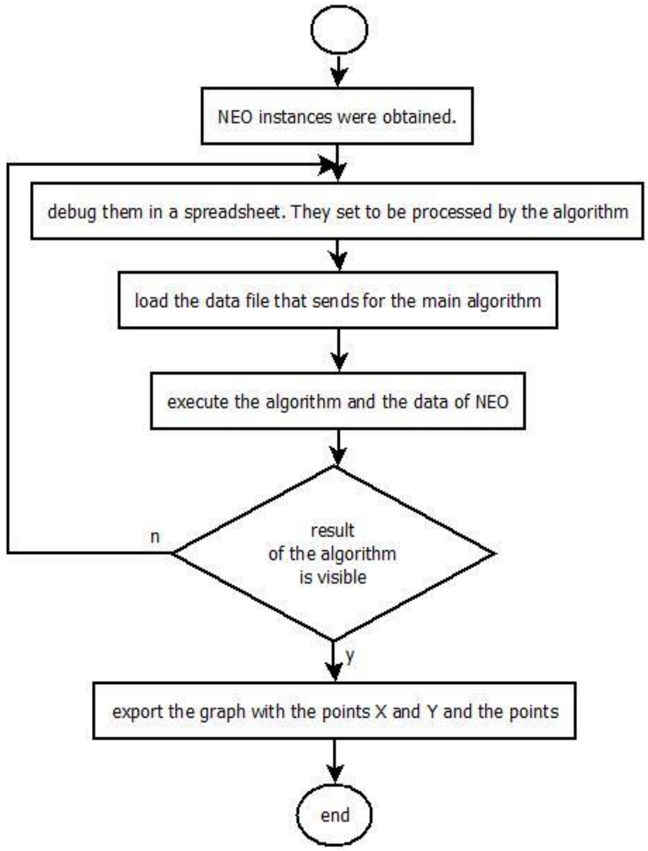

Figure 5. Steps of the experiment

The next image shows the stages:

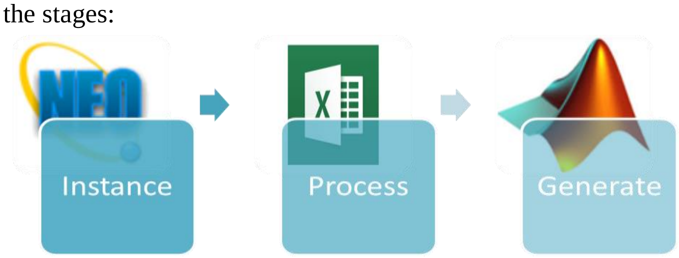

Figure 6. Stages to experiment

Matlab scripts are executed for each instance and solutions are compared and analyzed as described in next section.

5 Results

The use of algorithms to optimize the routes promises an improvement in planning, saving resources and time. In the case of this study the model of implementing instances NEO was shown.

The optimal solutions are shown below [9,14]:

- opt-A-n32-k5
  - Route #1: 21 31 19 17 13 7 26
  - Route #2: 12 1 16 30
  - Route #3: 27 24
  - Route #4: 29 18 8 9 22 15 10 25 5 20

  - Route #5: 14 28 11 4 23 3 2 6
  - cost 784
- opt-A-n33-k5
  - Route #1: 15 17 9 3 16 29
  - Route #2: 12 5 26 7 8 13 32 2
  - Route #3: 20 4 27 25 30 10
  - Route #4: 23 28 18 22
  - Route #5: 24 6 19 14 21 1 31 11
  - cost 661
- opt-A-n33-k6
  - Route #1: 5 2 20 15 9 3 8 4
  - Route #2: 31 24 23 26 22
  - Route #3: 17 11 29 19 7
  - Route #4: 10 12 21
  - Route #5: 28 27 30 16 25 32
  - Route #6: 13 6 18 1 14
  - cost 742
- opt-A-n34-k5
  - Route #1: 18 21 32 28 31 25 13
  - Route #2: 4 26 5 24 30
  - Route #3: 10 17 19 11 23 1 27
  - Route #4: 20 33 16 22 12 3 9 2
  - Route #5: 14 29 8 15 6 7
  - cost 778

Result opt-A-n32-k5

Route_VRP = 1 31 27 17 13 2 8 15 6 1
Route_VRP = 1 25 28 21 30 11 26 7 19 1
Route_VRP = 1 14 22 32 20 18 4 24 9 1
Route_VRP = 1 3 29 5 12 23 10 1
Route_VRP = 1 16 1
NumberOfRoutes = 5

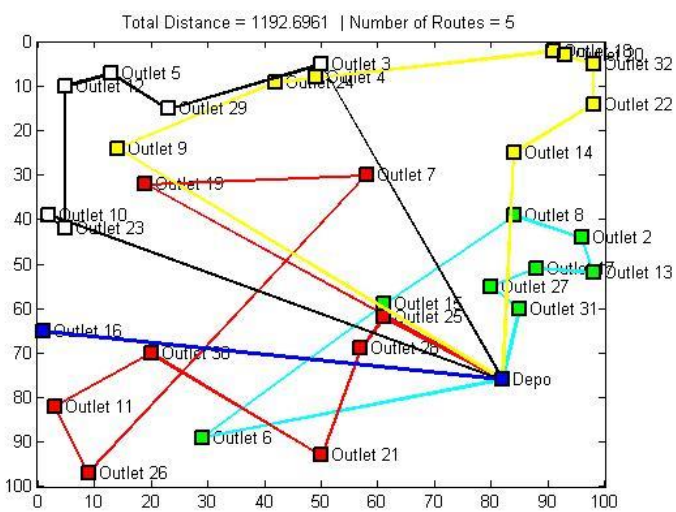

Figure 7. Graphic representation of the routes obtained for instance opt-A-n32-k5

Result opt-A-n33-k5

Route_VRP = 1 23 24 29 19 12 7 25 33 1

Route_VRP = 1 3 16 17 4 30 32 27 1
Route_VRP = 1 21 5 13 11 31 26 6 2 1
Route_VRP = 1 20 15 22 10 18 28 1
Route_VRP = 1 14 9 8 1
NumberOfRoutes = 5

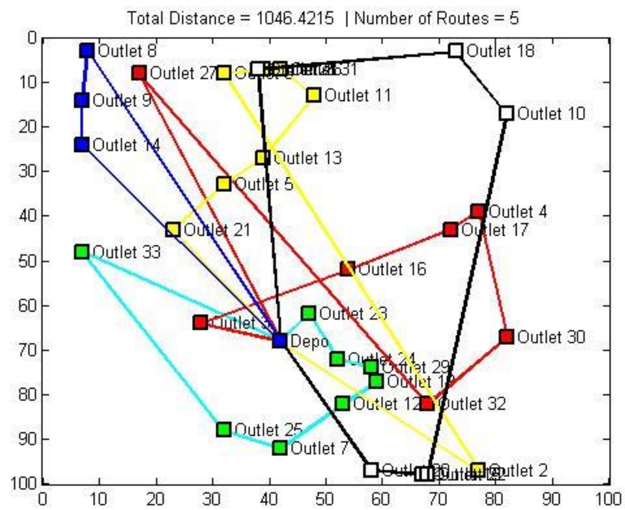

Figure 8. Graphic representation of the routes obtained for instance opt-A-n33-k5

Result opt-A-n33-k6

Route_VRP = 1 29 33 11 15 18 19 7 1
Route_VRP = 1 26 22 2 14 6 32 1
Route_VRP = 1 13 12 17 4 1
Route_VRP = 1 28 31 8 20 21 3 1
Route_VRP = 1 25 24 23 27 5 9 1
Route_VRP = 1 16 10 30 1
NumberOfRoutes = 6

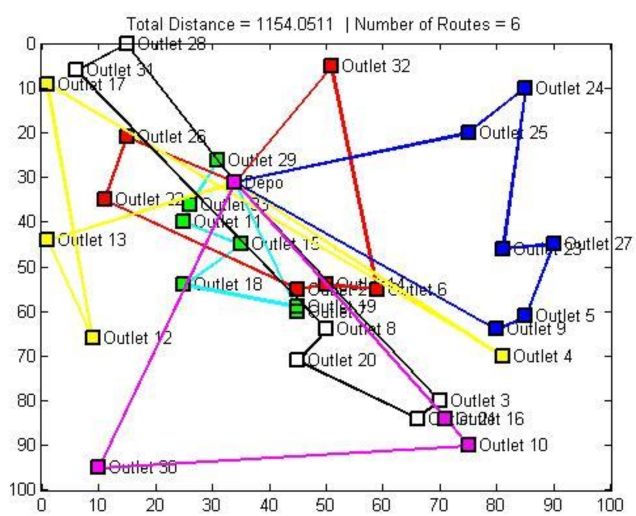

Figure 9. Graphic representation of the routes obtained for instance opt-A-n33-k6

Result opt-A-n34-k5

Route_VRP = 1 21 27 5 19 3 17 34 11 8 28 1
Route_VRP = 1 6 31 25 15 30 16 14 1
Route_VRP = 1 7 9 2 24 32 1
Route_VRP = 1 22 10 23 13 4 33 12 1
Route_VRP = 1 26 29 18 20 1
NumberOfRoutes = 5

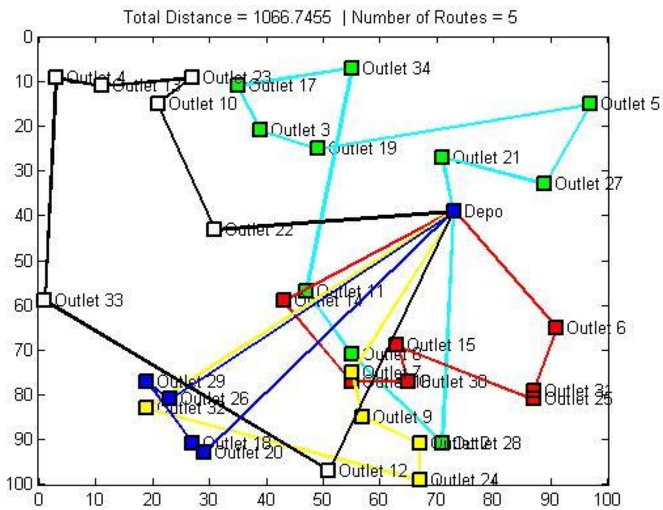

Figure 10. Graphic representation of the routes obtained for instance opt-A-n34-k5

Discussion

Table 1. Comparison with previous results

Instance	Augerat et al.	NNA in Matlab
opt-A-n32-k5	784	784
opt-A-n33-k5	661	661
opt-A-n33-k6	742	742
opt-A-n34-k5	778	778

Table 1 shows NAA is an excellent approach to solve CVRP, in all the cases reach optimal solutions.

Analysis capacities

We selected the opt-A-n34-k5 instance and we work with 3 different capacities, which are shown in Table 2; with 30 and another end 300 to 600. For each of the loads is set to a run NNA algorithm.

Table 2. Analysis of capacity

Capacity	30	300	600
Routes	15	2	1

The first capacity was done with 30 vehicles, for which it was given the function NIAlgVRP (Problem, n) a load of 30 where 30 represents n. The results are shown in the following figure:

>> NIAlgVRP(Problem,30)	
Route_VRP =	1 21 19 3 11 1
Route_VRP =	1 6 15 1
Route_VRP =	1 5 14 1
Route_VRP =	1 8 16 12 13 1
Route_VRP =	1 7 1
Route_VRP =	1 22 10 1
Route_VRP =	1 25 1
Route_VRP =	1 4 1
Route_VRP =	1 17 1
Route_VRP =	1 9 1
Route_VRP =	1 2 1
Route_VRP =	1 23 18 1
Route_VRP =	1 24 1
Route_VRP =	1 26 1
Route_VRP =	1 20 1
NumberOfRoutes =	15
TotalDistance =	1.85E+03

Figure 11. Calculation result 30

In row 1 it is called (in the Command Window section) to function NIAlgVRP (Problem,30). The results of calculations with the points to be visited by each route starting from the depot, in the case of 30 vehicles the result showed 15 routes with a total distance of 1.85E + 03 is displayed.

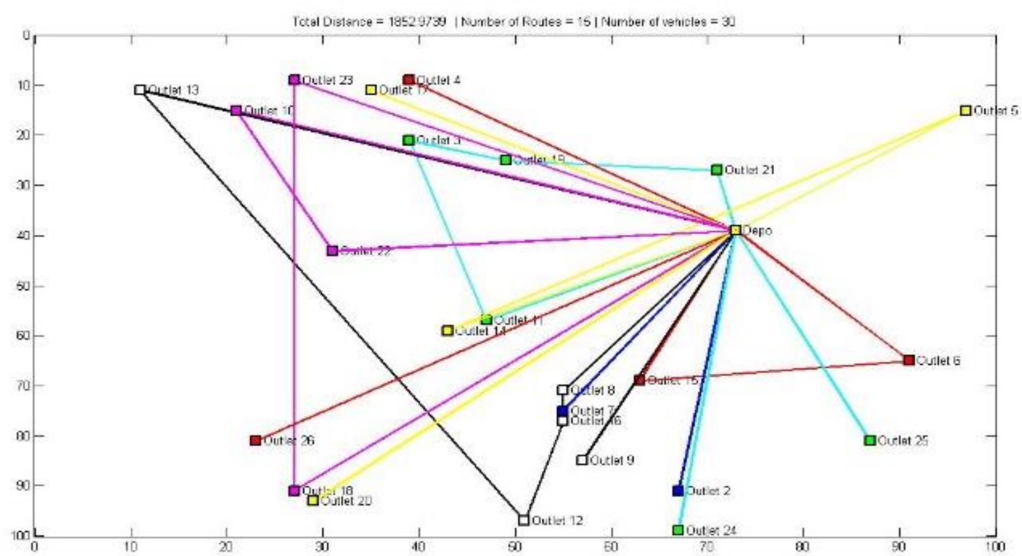

Figure 12. Graph of route 30

In Figure 12 routes each identified by a color display, each starting from the origin to destination, covering every point to completion. The second capacity was done with 300 vehicles, for which it was given the function NIAlgVRP (Problem, n) a load of 300. The results are shown in the following figure:

>>NIAlgVRP(Problem,300)	
Route_VRP =	
Columns 1 through 21	1 21 19 3 4 17 23 10 22 11 14 8 15 7 16 9 2 25 6 5 13
Columns 22 through 23	12 1
Route_VRP =	1 24 20 18 26 1
NumberOfRoutes =	2
TotalDistance =	700.672

Figure 13. Calculation result 300

It calls the NIAlgVRP(Problem,300) function. In the next section the results of calculations with the points to be visited by each route starting from the depot, in the case of 300 vehicles the result showed two routes with a total distance of 700.672.2 appears.

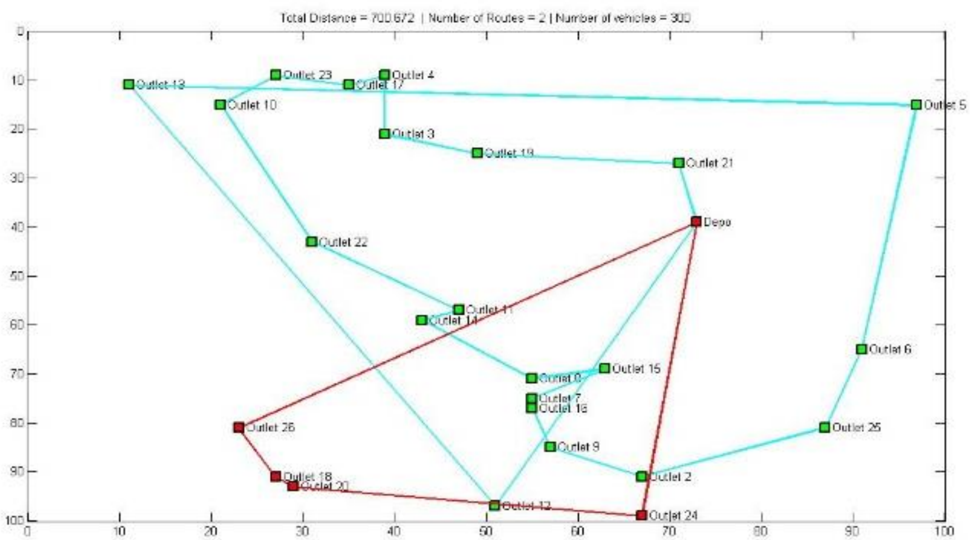

Figure 14. Graph of route 300

In Figure 14, 2 routes identified for each color, starting each one from origin to destination, covering every point to completion shown.

The third experiment was done with 600 vehicles, for which it was given the function NIAlgVRP (Problem, n) a load of 600. The results are shown in the following figure:

>>NIAlgVRP(Problem,600)	
Route_VRP =	
Columns 1 through 20	1 21 19 11 14 8 15 7 16 9 2 25 6 5 4 17 3 22 10 23
Columns 21 through 27	13 26 18 20 12 24 1
NumberOfRoutes =	1
TotalDistance =	551.4081

Figure 15. Calculation result 600

It calls the NIAlgVRP(Problem,600) function. In the next section the results of calculations with the points to be visited by each route starting from the depot, in the case of 600 vehicles the result showed one route with a total distance of 551.4081 appears.

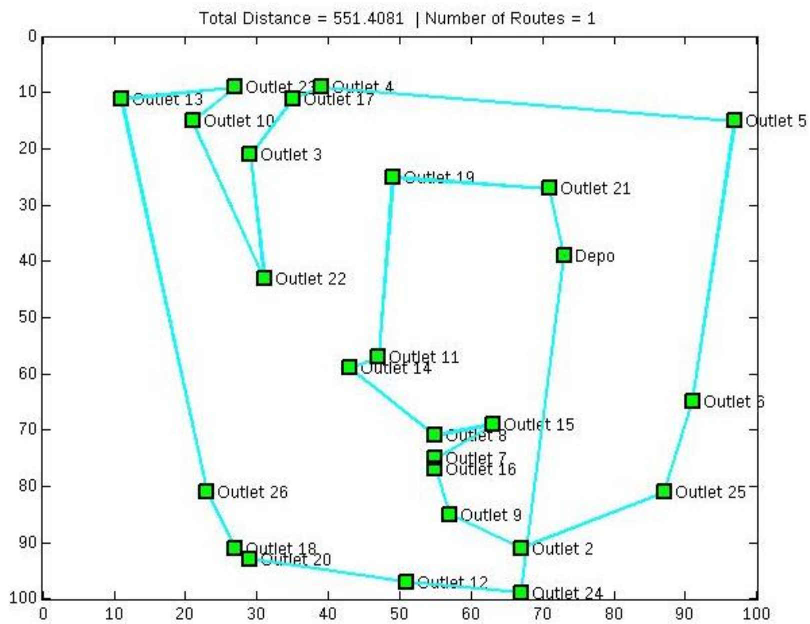

Figure 16. Graph of route 600

Figure 16 shows only 1 route, starting from the origin to destination, covering every point to completion.

6 Conclusions

In the present work four instances were compared and executed; the results showed that this algorithm is reliable because it shows the same results as the optimal solutions that Augerat et al. shows. It was compared an instance with three loads: the first is observed that fewer vehicles (30) the number of routes to be covered and therefore the total distance increases; the second shows that increasing at a rate of 10 (300) to cover routes decreased by 13.33% and therefore the distance is shortened to 37.8% and the last shows that a route is sufficient to cover the demand as much distance as the route itself minimizes the only detail is that 600 vehicles are needed to achieve it and this represents a major investment in resources.

The exercise allowed us to determine that the greater number of people, the lower the number of routes, according to the optimal solutions Augerat et al. as for the case of the load 30, the number of routes was 15; In conclusion, the more load, fewer routes.

Assuming that the algorithm is applied to the public transport; time, resources, fuel, driver's salary, pollution and traffic are optimized. Of the four instances executed, three of them show 5 routes as a result; Opt-A-n32-k5, opt-A-n33-k5 and opt-A-n34-k5, respectively. The opt-A-n33-k5 instance proved to be the most optimal with 661 as cost; this represents an improvement of 15.7% with respect to the opt-A-n32-k5 instance which has 784 as cost.

7 Future work

The possibility of implementing the algorithm Ant Colony Optimization (ACO) to solve CVRP applied to the public transport system. ACO is powerful and has shown that it is possible to solve implementation problems of the real world by analogy with nature as mentioned in [3-7]. We plan to model and simulate the basic behavior of simulated ant colony in the public transport in the metropolitan area of Guadalajara (MAoG). We are planning the application of this study to the creation of smart cities from the computational perspective as mentioned in [12].

References

1. Akbarzadeh-T, M.R., I Kamkar y M Yaghoobi.: Intelligent water drops a new optimization algorithm for solving the Vehicle Routing Problem. Systems Man and Cybernetics (SMC) (2010)
2. Zhang, Q., & Zhen, T.:A Hybrid Metaheuristic Algorithm for theMulti-depot Vehicle Routing Problem with Time Windows.Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference, 7 98-801 (2009).
3. Chen, Z.: An Improved Ant Algorithm for VRP with the Shortest Delivery Time In Distribution Center. Natural Computation, 2007. ICNC 2007.Third International Conference on, 761-765 (2007).
4. Geng-Sheng, W., & Yun-Xin, Y.: An Improved Ant Colony Algorithm for VRP Problem. Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium, 129-133 (2010).
5. Ma, X.: Vehicle Routing Problem with Time Windows Based onImproved Ant Colony Algorithm. Information Technology and Computer Science(ITCS), 2010Second International Conference, 94-97 (2010).
6. Majid Yousefikhoshbakht, E. K.:Solving the vehicle routing problem by a hybrid meta-heuristic algorithm. Journal of Industrial Engineering International, 9 (2012).
7. Qi, C.: An Ant Colony Algorithm with Stochastic Local Search for the VRP. Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference, 464 (2008).
8. Ponce, J. C., Recio, A., Ochoa, A., Ornelas, F., Padilla, A., & Torres, A. (2014). Implementación de un algoritmo de colonia de hormigas en CUDA aplicado al problema de ruteo de vehículos in Aplicaciones Modernas de Optimización, la experiencia entre cuerpos Académicos. UAEM- Artificio Editores, México.
9. NEO. (2015). Recuperado el 18 de Septiembre de 2015, de http://neo.lcc.uma.es/vrp/vrp-flavors/
10. Li, F., Golden, B., & Wasil, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers & Operations Research, 2918-2930.
11. Cattaruzza, D., Absi, N., Feillet, D., & Vidal, T. (2014). A mimetic algorithm for the Multi Trip Vehicle Routing Problem. European Journal of Operational Research, 833-848.
12. Cossio, E., Hernández, A., Ochoa, A., Ponce, J., & Pérez, R. (2016). Implementation of CVRP in a metropolitan area with a high density of vehicle fleet. Congreso internacional industria y organizaciones, CIIO, (pág. 15). Colombia.
13. Sadok, A., Teghem, J., & Chabchoub, H. (2010). A Hybrid Grouping Genetic Algorithm for the Inventory Routing Problem with multi-tours of the Vehicle. International Journal of Combinatorial Optimization Problems and Informatics, 42-61.
14. Augerat, P., Belenguer, J.M., Benavent, E. Corberán, A., Naddef, D., Rinaldi, G. (1995). Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem, Research Report 949-M, Universite Joseph Fourier, Grenoble, France.
15. Matlab (2016). Recuperado el 5 de Noviembre de 2016, de https://www.mathworks.com/matlabcentral/profile/authors/3883690-sas-wahid-hamzah