ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **23**, Number T, 2020, T5-T17

Vehicles Circuits Optimization by Combining GPS / GSM Information with Metaheuristic Algorithms

Bilal BELDJILALI[1], Belkacem BENADDA[2] and Zakaria SADOUNI[3]

[1]Department of Space Geodesy,
Center of Space Techniques, Algerian Space Agency, Arzew, Algeria
E-mail: bbeldjilali@cts.asal.dz

[2]Department of Telecommunication,
Abou Bakr Belkaid University, Tlemcen, Algeria
E-mail: benadda.belkacem@gmail.com

[3]Laboratoire de Mathématique Pures et Application,
Centre Universitaire de la Mi-Voix, Calais, France
E-mail: zakaria.sadouni@univ-littoral.fr

Abstract. This paper deals with the design of an intelligent decision system based on embedded tracked technology. The proposed system gives users the opportunity to solve the problem of Vehicle Routing Problem with Time Windows (VRPTW) which is a generalization of the construction of Vehicle Routing Problem (VRP) problem. The Window Time signifies that for each depot there is a time interval within which it can be serviced. The proposed system also makes it possible for users to estimate position and track their vehicles fleet remotely through web browsers on any computer or smart phone. Web users are also able to program smart attitudes for their vehicles based on geographical considerations. The build-tracker implements smart reactions to local vehicle driver. The proposed solution is realized in terms of two parts, the tracker installed on the vehicle to follow it, and an intelligence system installed in the server side based on the Metaheuristics including genetic algorithm, which is able to estimate the ideal road serve with the minimum cost.

Keywords: Embedded device, GM862, GPS, GSM, Metaheuristics, Genetic algorithm.

1. Introduction

Because of nowadays-high number of connected objects as well as the need of smart interactions to control them, the need of systems able to provide the locations of several mobile vehicles, in real time implementing Web Intelligence, to handle significant amount of data, predict failures and produce accurate decisions increases. Indeed, many applications can use such platform that helps to prevent against theft, the usage for personal purpose, the measurement for traffic intense, as well as commercial information gathering and healthcare [1]. In this context and since the beginning of the 1970s, and more precisely since the series of oil shocks, the conditions of business growth and economic development have changed the socio-economic landscape. International competition has forced companies to fight for survival, and the major objective of all current businesses is not only productivity, but above all, competitive. It is important to note as a key objective the delivery phase of the product at the right time, the

scheduling is no longer only to produce in a reasonable time with the minimum cost [2].

To contribute in this research area we propose in this paper a smart hybrid solution based embedded system. The device used in this study is the GM862-GPS, it contains a Global Positioning System (GPS) receiver used to collect vehicles locations information and a Global System for Mobile communications (GSM) module to transmit the relevant data. In addition to these two components, the GM862-GPS contain a Python script interpreter and a memory for the user scripts. In the other side, an intelligent platform was designed based Metaheuristics research methods as the Genetic algorithms to minimize the transport cost based on the geographical information. The aim of this work is not to develop a new Metaheuristics algorithm but to choose from literature one can be implemented in our device, the chosen method must respect two conditions, the efficiency in finding the optimized solution in addition to the simplicity and the small size to be implemented in the memory reserved to the Python script. The final solution must be able to estimate effectively the low-cost trajectory with the minimum vehicles in addition to giving the possibility to users to visualize in real time the operation by tracking vehicles using our web application.

This paper is structured as the following: in Section 2 we give a general description about the Routing Problem with Time Windows with mathematical description. The embedded system and the user's platform are illustrated in Section 3. Section 4 describes the solution algorithm and some initial results. The last section presents the final solution as obtained from the genetic algorithm.

2. Vehicle Routing Problem with Time Windows

The visit windows or time window correspond to an interval of time during which a visit to the customer is possible. Two types of existing visit windows exist, one called 'wide window' and another called 'tight window'. The wide viewing window allows a vehicle to arrive outside the customer window but against party a penalty will then be inflicted. On the other hand, the tight visit window does not authorize the arrival of a vehicle outside the visiting hours. The problem of vehicle tours with inspection windows is also called Vehicle Routing Problem with Time Windows (VRPTW) [3], [4]. Fig. 1 gives an example of the VRPTW with 5 clients must be solved using 2 vehicles. The example shows for each client a window time and the request. So any solution must respect these windows taking in consideration also the distance of travel as well as the service times of each customer.

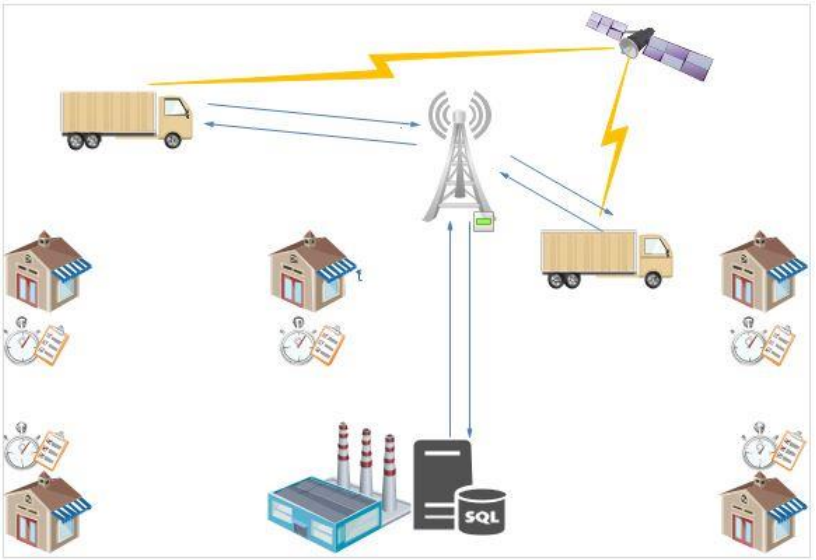

Fig. 1. Example of vehicles rounds problem with time windows

The aim of the solution presented in Fig. 1 is to serve all clients with a limited number of vehicles. The keys parameters that must be taken in consideration are the service time for each client, the vehicle capacity and each client request. The goal of any proposed solution is to

optimize the transport time with respect to the clients' availability which can proportionality reduce the transport cost.

In the following we suppose that:

n: number of clients (or vertices),
m: number of vehicles,
d_i: the customer i request,
q: vehicle capacity,
c_{ij}: the cost of the travel between customer i and j,
b: the upper limit of the summit i visit window,
i_f: the service time of the summit i,
t_{ij}: the transport time between nodes i and j,
u: the moment of visit of the summit i by the vehicle k,
T: a large value (T > 0).

The Vehicle Routing Problem with Time Windows can be mathematically presented as [2], [3], [4]:

$$\min\left(\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \ \ \sum_{k=1}^{m} x_{ijk}\right) \quad (1)$$

(1): Means that the objective of the optimization problem is to minimize the sum of the costs of all the tours.

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ijk} = 1 \quad \nabla 1 \ \leq j \leq n \quad (2)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{m} x_{ijk} = 1 \quad \nabla 1 \ \leq i \leq n \quad (3)$$

(2) and (3): require that each customer be served once and only once.

$$\sum_{i=1}^{n}\sum_{l=1}^{n} x_{ilk} - \sum_{i=1}^{n}\sum_{l=1}^{n} c_{ljk} = 0 \quad (4)$$

(4): ensure flow conservation.

$$\sum_{j=1}^{n} x_{ojk} = 1 \ \nabla 1 \ \leq k \leq m \quad (5)$$

$$\sum_{j=1}^{n} x_{iok} = 1 \ \nabla 1 \ \leq k \leq m \quad (6)$$

(5) and (6): ensure that each round begins and ends at the depot.

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ijk}\, q_i \ \leq Q \ \nabla 1 \ \leq k \leq m \quad (7)$$

(7): are the capacity constraints.

$$u_j^k \ \leq \ b_i \ \nabla 1 \ \leq i \leq m \ \nabla 1 \ \leq k \leq m \quad (8)$$

(8): Check that the visit time of the site is between the lower and upper limits of the site visit window.

$$u_i^k + \ s_i + \ t_{ij} - \ T\left(1 - \ x_{ijk}\right) \leq u_j^k \ \nabla 1 \ \leq i \leq n \ \nabla 2 \ \leq j \leq m \nabla 1 \ \leq k \leq m \quad (10)$$

$$x_{ijk} \in 0{,}1 \ \nabla 0 \ \leq i, j \leq n \ \nabla 1 \ \leq k \leq m \ \ u_i^k \ \geq 0 \quad (11)$$

(9) and (10): Check the consistency of the visit times when two sites follow each other.

To test our solution we propose a scenario that contain 10 customers with different location, service time and request quantity. The demand is supposed arrived at the same time.

Table 1 summarizes clients' window time and demand quantity.

Table 1: Customer information

CUSTOMER NO	DEMAND	READY TIME	DUE DATE
0	0	0	1230
1	10	850	1030
2	30	758	938
3	10	16	400
4	10	640	740
5	10	15	195
6	20	655	752
7	20	108	288
8	20	200	380
9	10	480	660
10	10	249	474

After receiving the request, the first step is to locate all available vehicles, for that, an embedded system with web platform are developed, the following section is a detail description about our proposed solution.

3. Description of the platform

The designed platform uses an embedded tracking component integrated on any vehicle for remote supervision and localization. In our case, the relevant solution uses the GPS to calculate the geographical coordinates of the vehicle [5], [6]. The tracker integrates one of the popular telecommunications services, the GSM [7], which is used as a communication support to transmit several information sensed by the built tracker integrated on the vehicle. The role of the developed central station consists to gather data and upload the vehicle activities into Web database. Users can access the central station through web browsers on any device like computer or Smartphone. The location of the vehicle can be visualized using Google Earth API's [8]. The central station also allows users to program intelligence and smart attitude for supervised vehicles. Indeed, any registered vehicle, on our platform, can react explicitly based on geographical considerations: activates / deactivates itself, sends alerts and notifications. Our localization platform combines two parts as shown in Fig. 2.

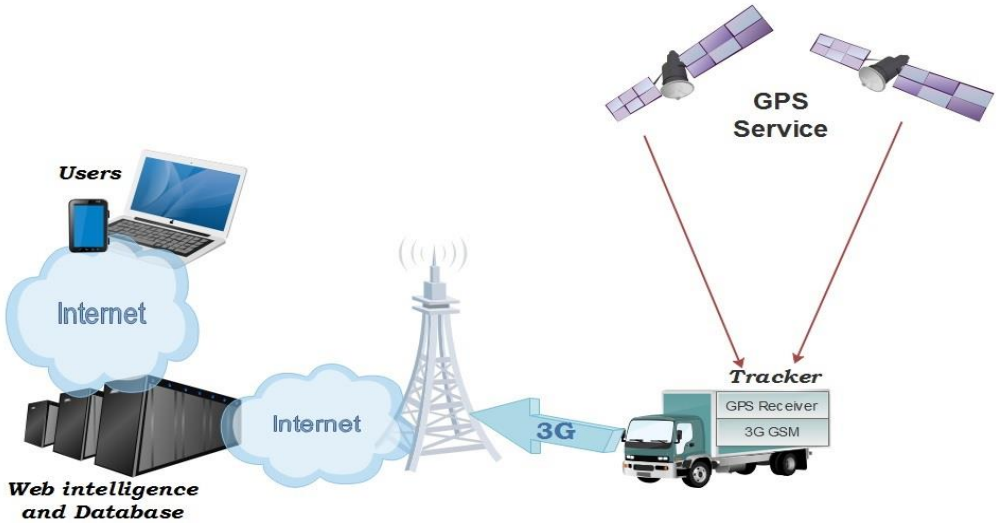

Fig. 2. Proposed platform model

3.1 Tracker intelligence

Our tracker is an embedded sensor, uses the GPS to determine its geographical position, the GSM network to transmit these positions and other information as current time to the database station; the device consists of four modules:

- GPS receiver,
- Processing unit with python interpreter,
- 3G GSM modem,
- Battery based power system.

In our project, all the development is done on the Telit GM862-GPS module shown in Fig. 3, which integrates all the proposed earlier requirements. In addition, the Telit GM862-GPS incorporates 13 general-purpose inputs outputs, GPIO pins. We will use some of these pins to control the behavior of the vehicle as activation / deactivation and alarm. The Telit GM862-GPS can be controlled through USB connection based on AT commands, except that we are interested by automated execution done through Python interpreter embedded in the Telit GM862-GPS module [9].

Fig. 3. The GM862-GPS module

Table 2 summarizes the most important characteristics of the GM862-GPS module

Table 2: The GM862-GPS characteristics

Parameter	**Information**
Dimension	43.9 / 43.9 / 6.9 mm
Weight	20 grams
Supported temperature	-40°c to 85°C
GPS frequency	L1 1575.42
GPS position accuracy	2.5 m
Satellite based augmentation system	EGNOS + WAAS
Band GSM	850/900/1800/1900 MHz
GSM info	GPRS class 10
Python memory	1.9 Mb

The Telit Wireless Solutions explains the Python APIs we used for the GPS interface. Our intelligent script is able to activate the vehicle based on a simple user phone ring; the authorized numbers are stored on the script as constant strings. In addition, we use a internet HTTP request grounded on the 3G GSM network over the GM862-GPS MDM interface to conduct to the Web Intelligent portal position, current time, alarm and the device attitude (On /

Off). Telit Wireless Solutions provide details on the MDM interface commands. The response of the sent HTTP request is able to modify the variables of the script as position and time limitations. Indeed, the script, do additional process to make choices on the vehicle attitude by generating correct logical levels on the GPIO6 and GPIO7 pins. The URL used to request scripts to handle the tracker send's data is repeated automatically every T seconds, which is a time constant fixed by the script programmer [9].

The developed python script, explained by the flowchart illustrate in Fig. 4, is used to drive the GM862-GPS internally and recover the stream string from the GPS interface. The stream string contains the time and the position.

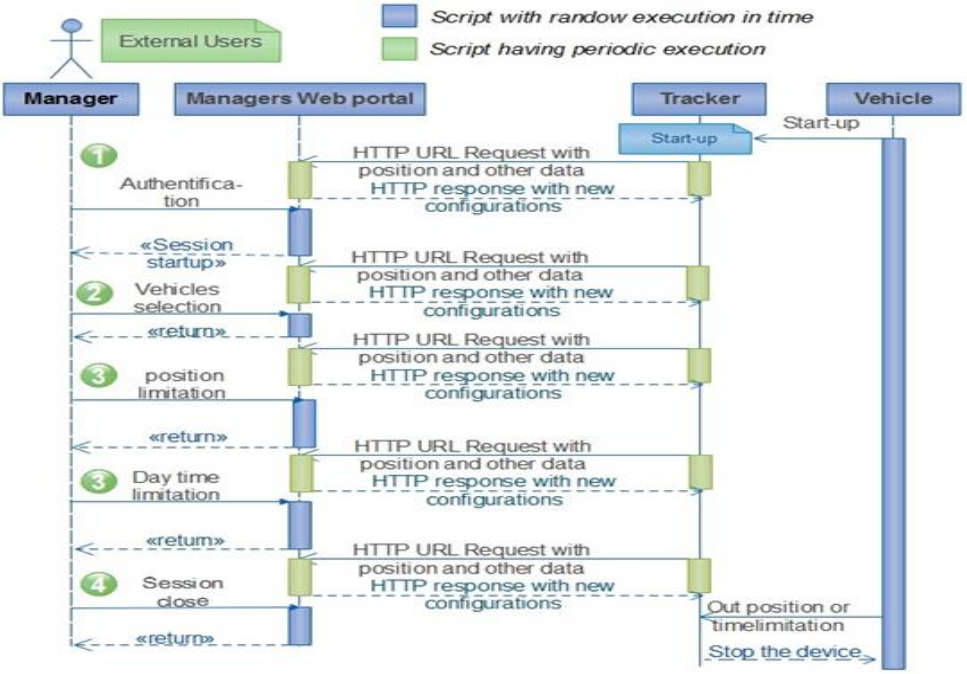

Fig. 4. Sequence diagram modeling the attitude of the tracker within the Web Intelligence.

The prototype that we propose for the GPS/GSM tracker is used to calculate periodically, by stated time interval within the tracker management software, current position and time as well as sensed events like vehicle on/off. Furthermore, we propose to add functionality of making the vehicle unavailable for use under geographical considerations and out of authorized hours. This will assist managers to prevent theft or personal use.

Ones the vehicle is on, it starts the tracker. Repeated script executions, between the tracker and web portal, are done to send relevant information and configuration. Users and managers use the web portal randomly, respecting the represented sequence.

3.2 Web intelligence

Our developed Web intelligence, use dynamic web portals technology and database, as explain in the book by Max Bramer [10]. The database we have developed is used to store traces for the vehicles positions in time, with the ability to display the current position of the selected vehicle [10].

The path or position display make benefit of Google Earth to have simple geographical overview and simplify the managers decisions. Michael P. Peterson [7] provides details on the development based on Maps APIs. The sequence diagram presented in Fig. 4 explains the web logic and intelligence, we propose to implement the Web portal.

The tracker regularly in time sends HTTP requests with position, vehicle attitude and alarms. Web portal scripts allow users to view the travel history and the vehicle information in simple Google Map.

In addition to the device tracking function, our system is able:

- To alert if the vehicle is outside non-authorized geographical region, this decision is to be programmed through the web portal by managers. The tracker makes the correct decision based on saved position.
- To alert if vehicles switch on during non-authorized working hours. The tracker is able to make the decision based on the GPS received time.
- To calculate the optimized road based on the customers coordinate (selected by user) and vehicle coordinate (vehicle in service).

The database structure is presented in Fig. 5.

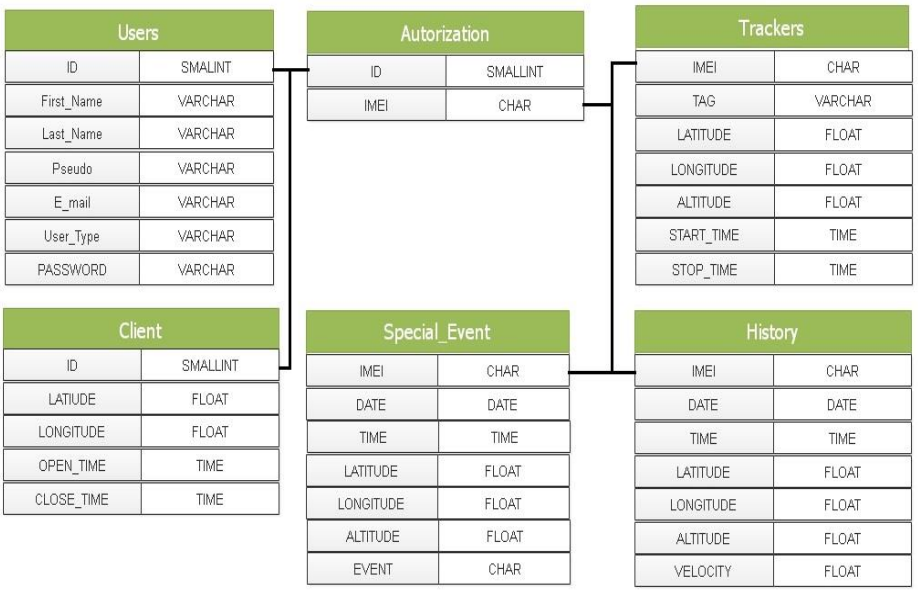

Fig. 5. Database structure

We have developed Web host to manage trackers and sent data. The positions history is displayed on google map as the user request: Fig. 6 shows an execution example.

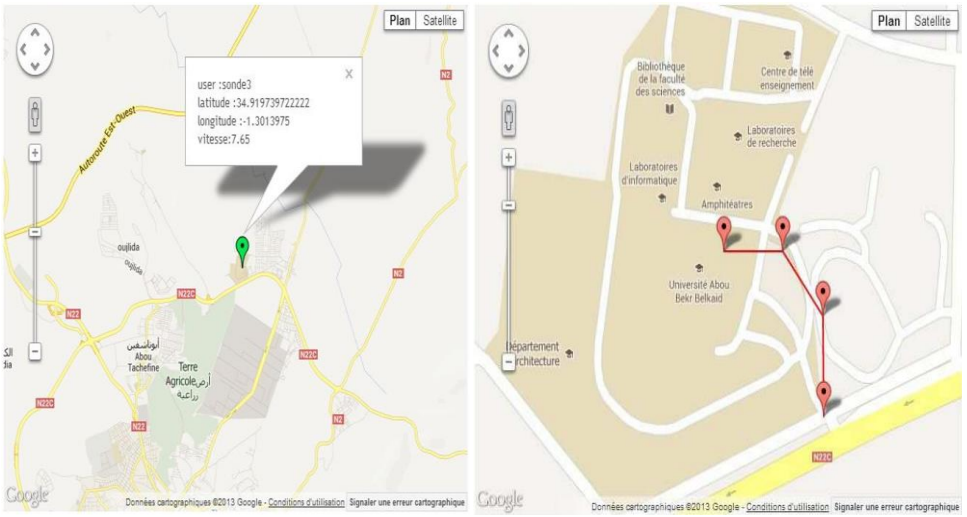

Fig. 6. Web application interface

The host allows managers to initialize variables and control the vehicle behavior: deactivation if out of geographical scope or outside of the allowed exploitation time. For those purposes we have create PhP script and MySQL database responsible for the working intelligence as well as the information management and control. We have defined two

authorized access categories: common users for data consultation and managers for vehicles stored commands and configurations. The whole solution act as simple web portal accessed via web browsers (Fig. 7).

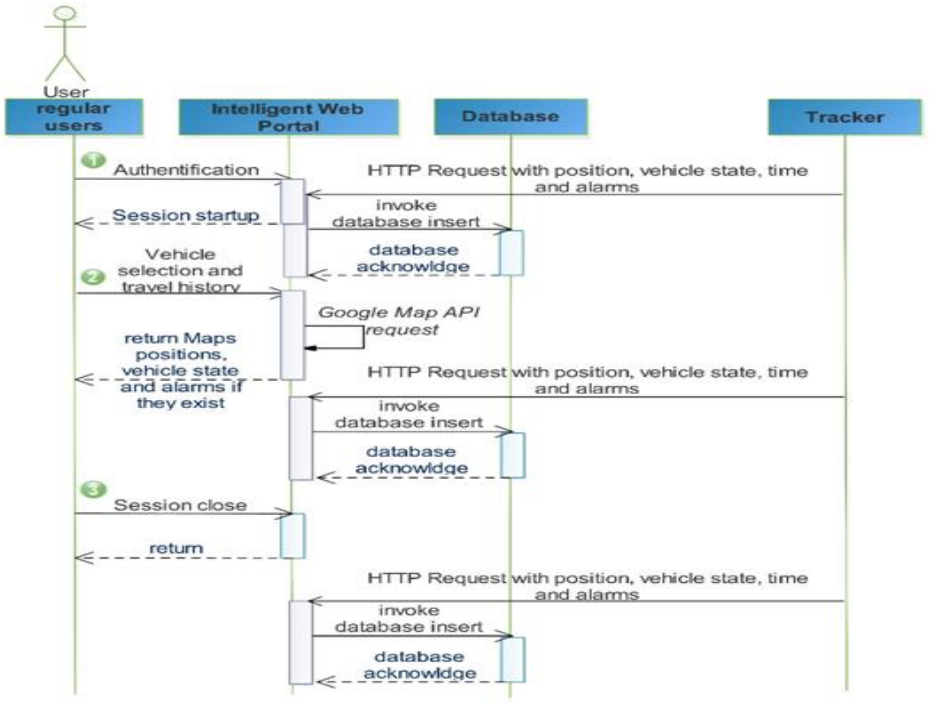

Fig. 7. Sequence diagram modeling the attitude of the users

Fig. 8 represents a systematic presentation, which summarizes the tracking algorithm.

Now, after locating all vehicles the next step is to integrate the position obtained with previous data (customers' locations, service time and request quantity) in the Metaheuristics algorithm to obtain the best routing strategy.

4. Proposed solution

The Metaheuristics method attempts to avoid trapping local minimums by permitting degradation of the solution during a number of iterations. It explores the space of solutions by executing movements at each iteration, from a solution to the best solution in the neighborhood of its note N (s). To avoid cloaking on the solutions already explored, the solutions recently examined will provide a backtracking after these iterations. Many researchers have proposed a heuristic based on the notion of gain, which is undoubtedly the oldest and best-known heuristic for the VRP [11], [12]. The procedure begins with a solution where each customer is served individually in his tour by a single vehicle. Two turns will then be merged to form one tour serving at a lower cost both customers by one of the vehicles. Fig. 9 shows an example (position and window time) of 10 customers will be served.

5. The genetic algorithm

The genetic algorithm is an optimization algorithm based on techniques derived from genetics and natural evolution: crossing, mutation, selection. To get the minimum cost of travel we follow the following steps [16]:

- Step 1: We generate an initial population of size n chromosomes, then randomly select the genes that make up each chromosome: this is the first generation of chromosomes.

- Step 2: Each chromosome is evaluated by the objective function, which allows deducing its fitness value.
- Step 3: The population generation cycle then begins, with each new population replacing the previous one. The number X of generations is determined initially. In each generation, we seize n chromosomes to which we will apply the different genetic operators: selection, crossing, mutation.

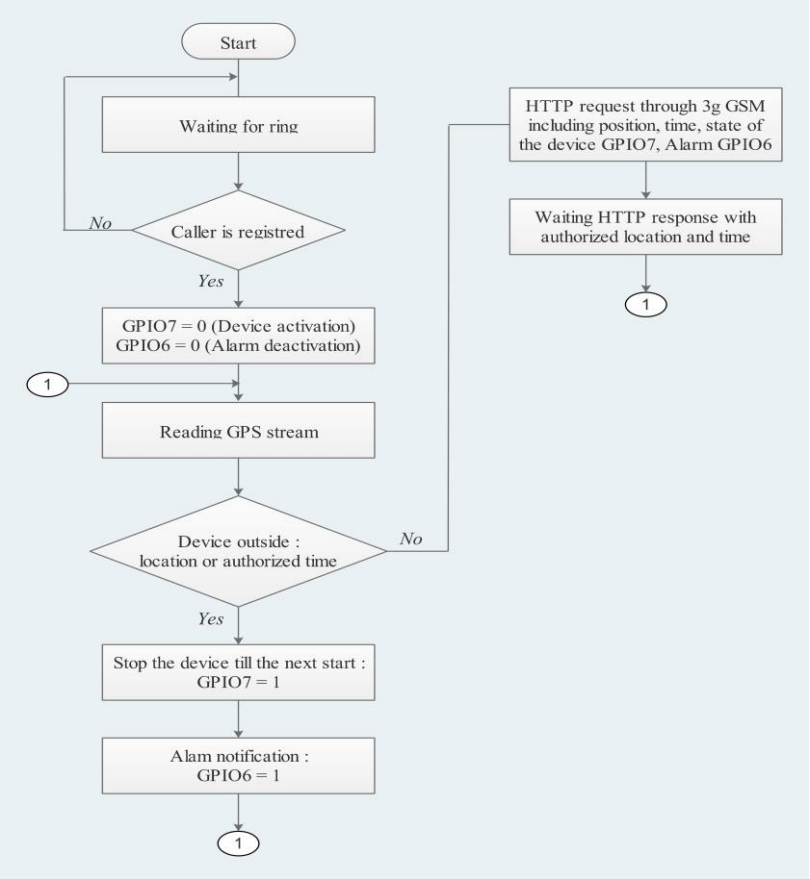

Fig. 8. Systematic presentation of the tracker

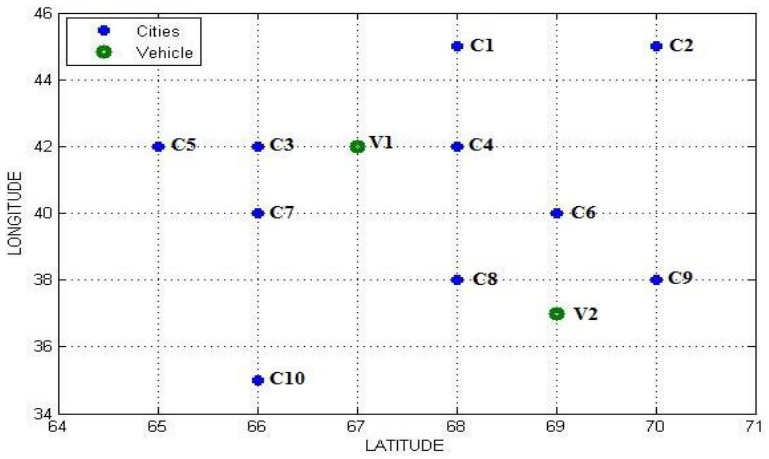

Fig. 9. Customers and vehicles position

We denote by 0 the central depot. The total cost of serving two customers separately by two vehicles is [11], [13], [14]:

$$G = C_{0i} + C_{i0} + C_{0j} + C_{j0} \quad (12)$$

while that of a single vehicle serving successively i and j on the same turn is:

$$G = C_{0i} + C_{ij} + C_{j0} \quad (13)$$

The gain obtained after merging two turns is therefore:

$$G_{oi} = C_{i0} + C_{0j} - C_{ij} \quad (14)$$

Customers i and j are grouped in the same tour if the gain G obtained is maximum by respecting the feasibility of the round obtained after fusion. Clients i and j will then form a new client called the client macro. The tour containing this macro client will be merged with another tour. The time needed is calculated between the cities. It is assumed that the travel time of 90 Km is 60 min.

To determine an initial solution of good quality, nothing is better than to use a recognized heuristic namely that of CW that we have modified in order to take into account the constraints of time windows. The steps of our proposed algorithm are presented in Algorithm, 1 [11], [15].

Algorithm 1: Steps of our proposed algorithm to calculate initial solution

Data: i, j : customers
Capv: Vehicle capacity
DC: Start of the path
FC: End of the path
VL: List of visited summit
Result: G: Gain Table
Begin
m = n = 10
q = 700
DC = null
FC = null
for $i = 1, i \leq 10, i++$ ***do***
for $j = 1, j \leq 10, j++ do$
%Calculate the distance between customers
||x|| = XCOORD(a) – XCOORD(b)
||y|| = YCOORD(a) – YCOORD(b)
$D = \sqrt{x^2 + y^2}$
%Calculate the Gains (G)
$D_a = C_{0i} + C_{i0} + C_{0j} + C_{j0}$
$D_b = C_{0i} + C_{ij} + C_{j0}$
$G_{i,j} = D_a - D_b = C_{i0} + C_{0j} - C_{ij}$
end
end
% Sort the gain in descending order
for $i = 1, i \leq 10, i++$ ***do***
for $j = 1, j \leq 10, j++$ ***do***
if $G_{i,j} \leq G_{i,(j+1)}$ ***then***
$G_{i,j} \rightleftharpoons G_{i,(j+1)}$
end
end
G=0

VL = []
while Capv $\leq$ total capacity ***do***
$Time = a_i^k + s_i + t_{j,i}$
if $Time \leq b_j^k$ ***then***
DC = j
FC = i
VL = ij
G = G + Cost
else
$Time' = a_j^k + s_j + t_{j,i}$
end
if $Time \leq b_i^k$ ***then***
DC = j
FC = i
VL = ji
G = G + Cost
end
Solin1 = Solin1 +1
end
end

The cost calculated for each two cities is presented in Fig. 10.

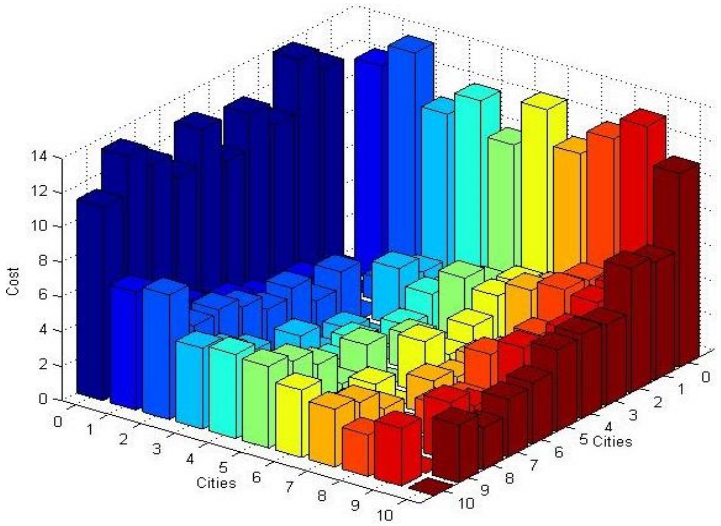

Fig. 10. Cost of travel between each 2 customers

The genetic algorithm output is summarized in Table 3.

6. Conclusion

In this paper we proposed a solution to solve the vehicle tours with Time Window (VRPTW) problem by adapting Metaheuristic methods as genetic algorithm which provide us a good solution in a reasonable time. Our proposed solution is based on a GM862-GPS embedded system and a web platform to manage our database. The application of Metaheuristic research on any type of problem does not guarantee definitive success, but the most important thing is to know how to adapt the algorithm research to the problem posed, and this by adjusting in an appropriate way its different components. For that, many parameters are taken in consideration like customers' locations, service time, and request quantity in addition to the number of

vehicles used in the operation to optimize the travel cost.

Table 3: Genetic algorithm output

First generation		Second generation		Third generation	
Solution	**Cost**	**Solution**	**Cost**	**Solution**	**Cost**
S1	130.03	S1	97.96	S1	156.70
S2	133.58	S2	127.09	S2	161.54
S3	127.81	**S3**	**96.30**	S3	96.29
S4	127.81	S4	126.22	S4	159.09
S5	130.03	S5	127.09	S5	159.67
S6	130.03	S6	126.22	S6	161.54
S7	127.81	S7	96.30	S7	156.70
S8	130.03	S8	126.22	S8	97.55
S9	**127.81**	S9	126.22	**S9**	**96.29**
S10	130.03	S10	126.22	S10	97.55

From the literature, many algorithm are developed to solve these type of problem, the aim of the work is not to develop or improve other one but to find from the existing one which respect some condition as the small size to be implemented in the allocated memory and its capacity to solve problems in quasi real time.

This paper also proved that the presented solution makes it possible for users to determine the location and positions history of a target vehicle in real time. Users can access the web portal and display the tracking information on Google Maps, they are also able to simultaneously see the evolution of the positions of several vehicles, each vehicle identified with different color.

The prototype that we developed consists of two parts. The first one is the sensor installed at the tracked vehicle to determine its position using GPS. Position data is sent through 3G GSM network link. The second part of our system is the Web portal, which provides the user interface and used to recover the row data sent from tracked vehicles.

References

[1] H. LIAO, C. LU and J. SHIN, *Incorporation of GPS and IP camera for people tracking*, GPS Solutions, **16**, 2012, pp. 425–437. https://doi.org/10.1007/s10291-011-0242-8

[2] V. LANDEGHEM., *A bi-criteria heuristic for the vehicle routing problem with time windows*, European Journal of Operations Research, **36**, 1988, pp. 217−226. https://doi.org/10.1016/0377-2217(88)90428-6

[3] E. TAILLARD, P. BADEAU, M. GENDREAU, F. GUERTIN and J.Y. POTVIN, *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Transportation Science, **31**, 1997, pp. 170−186. https://doi.org/10.1287/trsc.31.2.170

[4] M. SOLOMON, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research, **35**, 1987, pp. 254−265. https://doi.org/10.1287/opre.35.2.254

[5] X. GUOCHANG, *GPS: Theory, Algorithms and Applications*, 2nd Edition, Springer, 2003. https://doi.org/10.1007/978-3-540-72715-6

[6] B. BELDJILALI and B. BENADDA, *Optimized Station to Estimate Atmospheric Integrated Water Vapor Levels Using GNSS Signals and Meteorology Parameters*, ETRI Journal, **38**, 2016, pp. 1172−1178. https://doi.org/10.4218/etrij.16.0116.0093

[7] M. TAFERNER, *Wireless internet access over GSM and UMTS*, Springer-Verlag Berlin Heidelberg, 2002. https://doi.org/10.1007/978-3-662-04771-2

[8] P. PETERSON, *Online Maps with APIs and WebServices*, Springer-Verlag Berlin Heidelberg, 2012. https://doi.org/10.1007/978-3-642-27485-5

[9] TELIT WIRELESS SOLUTIONS, *Easy Script in Python for GT863-PY, GT864-PY, GM862QUAD-PY, GM862-GPS, GC864-PY, GC864-PY w/ SIM holder, GE863-PY, GE863GPS, GE863-SIM, GE864-PY, GE-864-QUAD Automotive and GE865-QUAD*, Rev.10, Telit Wireless Solutions. https://www.semiconductorstore.com/pdf/newsite/Telit/GM862-GPS/GM862-

GPS_Software_User_Guide_r4.pdf
[10] M. BRAMER, *Web Programming with PHP and MySQL*, Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-22659-0
[11] Z. YANG, J. P. OSTA, B. VEEN, R. KREVELEN, A. STAM, J. KOK, T. BÄCK and M. EMMERICH, *Dynamic vehicle routing with time windows in theory and practice*, Natural Computing, **16**, 2017, pp. 119–134. https://doi.org/10.1007/s11047-016-9550-9
[12] R. P. ALVAREZ GIL, Z. C. JOHANYAK and T. KOVACS, *Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation*, International Journal of Artificial Intelligence, **16**, 2018, pp. 20−40.
[13] R.-E. PRECUP and R.-C. DAVID, *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*, Butterworth-Heinemann, 2019. https://doi.org/10.1016/C2018-0-00098-5
[14] C. PURCARU, R.-E. PRECUP, D. IERCAN, L.-O FEDOROVICI, R.-C. DAVID and F. DRAGAN, *Optimal robot path planning using gravitational search algorithm*, International Journal of Artificial Intelligence, **10**, 2013, pp. 1−20.
[15] A. SOARES, R. RÂBELO and A. DELBEM, *Optimization based on phylogram analysis*, Expert Systems with Applications, **78**, 2017, pp. 32−50. https://doi.org/10.1016/j.eswa.2017.02.012
[16] M. GEN, R. CHENG and L. LIN, *Network models and optimization: multiobjective genetic algorithm approach*, Springer-Verlag. 2008. https://doi.org/10.1007/978-1-84800-181-7