Procedia Computer Science

Volume 51, 2015, Pages 2227–2236

ICCS 2015 International Conference On Computational Science

A Solution for a Real-time Stochastic Capacitated Vehicle Routing Problem with Time Windows

Pedro J. S. Cardoso[1,2], Gabriela Schütz[1,3], Andriy Mazayev[4], Emanuel Ey[1], and Tiago Corrêa[4]

[1] Instituto Superior de Engenharia, University of the Algarve, Faro, PORTUGAL
[2] LARSys, University of the Algarve, Faro, PORTUGAL
[3] CEOT, University of the Algarve, Faro, PORTUGAL
[4] Dep.to de Engenharia Eletrónica e Informática, University of the Algarve, Faro, PORTUGAL
{pcardoso,gschutz,amazayev,eevieira,tiagodcorrea}@ualg.pt

Abstract

Real-time distribution planning presents major difficulties when applied to large problems. Commonly, this planning is associated to the capacitated vehicle routing problem with time windows (CVRPTW), deeply studied in the literature.

In this paper we propose an optimization system developed to be integrated with an existing Enterprise Resource Planning (ERP) without causing major disruption to the current distribution process of a company. The proposed system includes: a route optimization module, a module implementing the communications within and to the outside of the system, a non-relational database to provide local storage of information relevant to the optimization procedure, and a cartographic subsystem. The proposed architecture is able to deal with dynamic problems included in the specification of the project, namely: arrival of new orders while already optimizing as well as locking and closing of routes by the system administrator. A back-office graphical interface was also implemented and some results are presented.

Keywords: Stochastic Capacitated VRPTW, Real World Application, Enterprise Resource Planning

1 Introduction

Intelligent Fresh Food Fleet Router (*i3FR*) is a project with the objective of building a system to manage and optimize the distribution of fresh products by fresh goods distribution fleets. The project, a joint venture between the University of the Algarve and X4DEV – Business Solutions, is being integrated with an in-production SAGE ERP X3 [SAG15], with the objective of optimizing the distribution of dry, fresh and frozen goods, without causing major disruptions to the existing distribution planning procedures. The project has two main modules in development: (a) a hardware system to acquire data from the distribution vehicles (out of the scope of

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.501

this paper) that will be used to validate the routes and provide information about other parameters, such as the vehicles chambers' temperatures, vehicles' fuel cost, traveled kilometers, and the vehicle's Global Positioning System (GPS) data; and (b) a routing optimization platform which computes routes from the depots to the delivery points using cartographic information and takes into consideration multiple objectives.

The *i3FR* routing system is divided in four main components, namely: (a) *i3FR-Opt* which is where the actual route optimization algorithm is implemented, taking into account a multiple objectives variant of a stochastic capacitated vehicle routing problem with time windows (VRPTW); (b) *i3FR-Hub* which implements all the communications occurring inside and to the exterior of the *i3FR* routing system; (c) *i3FR-DB* which operates on top of a non-relational database to provide local storage of information relevant to the optimization procedure (e.g., data retrieved from the ERP, cartographic information, and computed routes); and (d) *i3FR-Maps* which is a cartography subsystem to retrieve and store routing informations from other systems (e.g., from Google Maps).

The main novelty of this work is the presentation of an architecture capable of integrating an optimization procedure with an existing ERP, taking into consideration the need for a near real-time routing solution with dynamic orders, dynamics of the warehouse picking and vehicle loading, and the interaction with the system administrator.

The document is structured as follows. Section 2 presents a contextualization and the formulation of the problem in study. The third section presents the system's architecture. Section 4 briefly describes the optimization procedure. Examples of the proposed solution are presented in the fifth section. The last section presents some conclusions and future work.

2 Contextualization and Problem Formulation

2.1 Related Works

In its simplest form, the distribution problem we are facing can be formulated as a VRPTW. A solution to the VRPTW problem is a set of routes, each one starting at some depot, that visit/deliver goods to each customer once, within given time intervals, without violating the vehicle capacities. The problem is intrinsically a multi-objective problem. Two general goals are the minimization of the number of routes which corresponds to the number of needed vehicles to process the demand, and the total travel distance or time, corresponding to the distribution procedure costs [Sol87]. However other goals may occur in real problems like the minimization of the difference between the longest and shortest route or the maximization of the minimum load of the vehicles [CSME15]. The optimization of these last two objectives produces balanced routes in terms of worker effort and vehicle loads.

Therefore, several constraints and objectives arise in real problems. For instance, in the distribution of frozen, refrigerated and fresh goods it may be important to minimize the distribution time since maintaining the temperature inside the refrigerated compartments adds significant costs, associated with the consumption of fuel to keep the dedicated freezing engines working. In this sense, a method that minimizes not only the fixed costs for dispatching vehicles, but also the transportation, inventory, energy and penalty costs for violating time-windows is presented in [HHL07]. The same work discusses the time-dependent travel and time-varying temperatures, during the day, which led to the modification of the objective functions as well as the constraints. In [CHC09] a mathematical model is presented which combines production scheduling and vehicle routing with time windows for perishable food products. In [PBFI09] a stochastic problem is considered as the designing motion strategies for a team of mobile agents,

required to fulfill requests for on-site services. The services are generated by a spatio-temporal stochastic process which remains active for a certain deterministic amount of time, and then expires (describing customer impatience), and are fulfilled when one of the mobile agents visits the location of the request. Please refer to [PGGM13] for a more comprehensive review of applications and solution methods for dynamic vehicle routing problems.

Several approaches were made to solve the VRPTW problems. The use of meta-heuristics is a common solution [Mou08, TM08, BG02, LYL11, GTA99, GGLM03]. Other solutions include heuristics like the one for the distribution of fresh vegetables presented in [OS08] in which the perishability represents a critical factor. The problem was formulated as a VRPTW with time-dependent travel-times, where the travel-times between two locations depend on both the distance and the time of the day. The problem was solved using a heuristic approach based on the Tabu Search and performance was verified using modified Solomon's problems. A somewhat similar work was proposed in [TK02], which deals with distribution problem formulated as an open multi-depot vehicle routing problem encountered by a fresh meat distributor. To solve the problem, a stochastic search meta-heuristic algorithm, termed as the list-based threshold accepting algorithm, was proposed. In [AS07] a generalization of the asymmetric capacitated vehicle routing problem with split delivery was considered. The solution determines the distribution plan of two types of products, namely: fresh/dry and frozen food. The problem was solved using a mixed-integer programming model, followed by a two-step heuristic procedure.

There are also a relatively large number of companies providing commercial software which is similar to the one developed in the *i3FR* project [Rou15, Opt15a, Opt15b, New15, Log15].

2.2 Problem Formulation

Mathematically, we have considered an instance of the VRPTW composed of a set of customer locations, C, a set of depot locations, D, and distances $d(i,j) \in \mathbb{R}$ and times $t(i,j) \in \mathbb{N}$ between each pair of locations $i, j \in C \cup D$. Each vehicle has a capacity q_τ for τ in Υ which is the set of storage types (e.g., dry, refrigerated, frozen). Each customer, i, has a time window $[a_i, b_i]$, a service time s_i and a demand $d_{i,\tau}$, for $i \in C$ and $\tau \in \Upsilon$. A solution is a set of m routes, $T_i = (l_{i,1}, l_{i,2}, l_{i,3}, \ldots, l_{i,m_i})$ with $i \in \{1, 2, \ldots, m\}$, such that a route starts and ends at the same depot, $l_{i,1} = l_{i,m_i} \in D$, $l_{i,2}, l_{i,3}, \ldots, l_{i,m_i-1} \in C$ are customers, and a customer is served by a single route (i.e., $\{l_{i,2}, l_{i,3}, \ldots, l_{i,m_i-1}\} \cap \{l_{j,2}, l_{j,3}, \ldots, l_{j,m_j-1}\} = \emptyset$ for $i \neq j$). The feasible routes take into consideration the travel time between customers and the associated service time, such that each vehicle leaves and arrives at the depots and clients in their time windows (in the case of the clients it can arrive before the time window opening, in which case it is forced to wait until the opening moment – the maximum waiting time is an additional parameter). Another restriction states that the capacity of the vehicle, in the different transportation types should not be exceeded, i.e., $\sum_{l \in T_i} d_{l\tau} \leq q_\tau$, $i \in \{1, 2, \ldots, m\}$ and $\tau \in \Upsilon$.

Taking the formulation into consideration, two criteria where minimized: the total number of vehicles required to achieve the service within the restrictions and the total traveled distances (i.e., the sum of the distances made by each vehicle).

3 *i3FR* System

3.1 Overall System

The *i3FR* routing system is composed of several interdependent sub-modules which communicate via HTTP, thus allowing for the system to be scaled from a single-machine environment

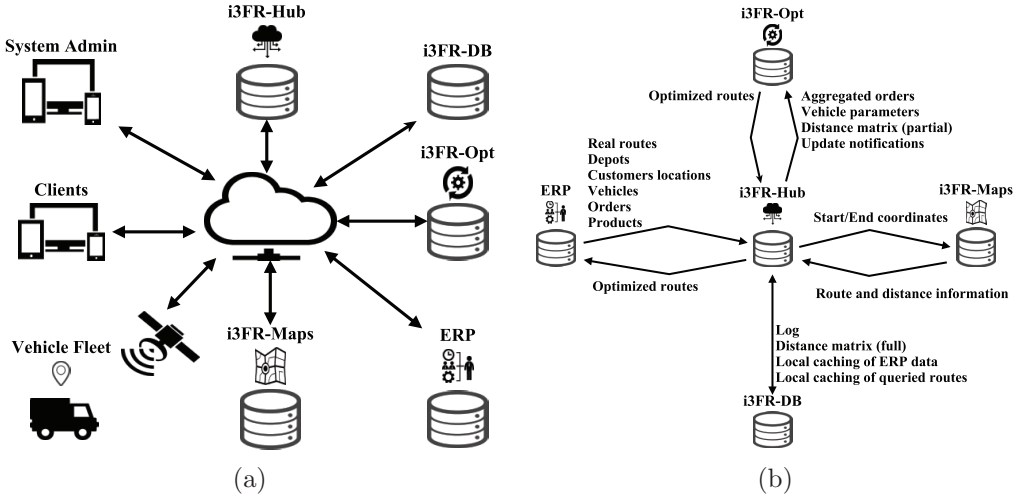

Figure 1: (a) Global system diagram. (b) Inputs and outputs of *i3FR-Hub*.

to a large multi-server production deployment. In more detail, the system is composed of the following modules: (a) *i3FR-Opt* – the actual route optimization algorithm is implemented in the optimization module designated *i3FR-Opt*. The optimization process relies on heuristics [CSME15] that should run in near real-time, taking into consideration the required dynamics of the overall system; (b) *i3FR-Hub* – all communications occurring inside the *i3FR* routing system go through *i3FR-Hub* which acts, as the name suggests, as a hub. For instance, the optimization module, *i3FR-Opt*, sends and receives all the data it requires through a RESTful API (application programming interface) provided by the *i3FR-Hub*. Furthermore, all communication to the external systems also goes through this hub, which acts as a "gateway", a security access provider, and a data validator; (c) *i3FR-DB* – the *i3FR* routing system operates on top of a non-relational database. The database provides local storage of information relevant to the optimization procedure, namely: data retrieved from the ERP (avoiding the overloading of the ERP system with constant requests), cartographic information, computed routes, etc; and (d) *i3FR-Maps* – a cartography subsystem, was also implemented which retrieves routing information from other systems (e.g., Google Maps and Open Street Maps) and stores it on the *i3FR-DB*. Figure 1 depicts the (a) overall system and the (b) type of messages sent between the modules.

3.2 System Dynamics

As already mentioned, the problem in question has a constraint that the solution should be achieved in near real-time since customers tend to send their orders very near to the loading time of the vehicles. In some extreme cases the orders are even made after the beginning of the vehicles' loading, which in those cases will not be unloaded. These last minute orders are then engaged in an existing route, or into a new one if they are not engageable without violating feasibility (e.g., time windows or vehicles capacity). Therefore, from the order formulation by the clients to the delivery of the goods, the process is quite dynamic. These dynamics are supported by human interventions which has a set of actors: clients which send the orders, ERP manager which controls the overall process and remaining workers which, for instance, do

the picking of the goods from the warehouse and load the vehicles.

In a sequence line the process can be described as follows (see Fig. 2 for a sequence diagram of the procedure described next). First a client or a seller sends an order to the ERP. Typically the order is for the next day although it can be for any future request. Then, at a convenient moment, the ERP manager sends a signal to the *i3FR* system in order to start the optimization process. This signal goes to the *i3FR-Hub* which begins by requesting the necessary data from the ERP. In general, the requested data is information relevant to the optimization process which is not yet present in the *i3FR-DB*. On other words, static data (e.g., client delivery locations, vehicle data) was already fetched by *i3FR-Hub* and stored in the *i3FR-DB*, which means that the *i3FR-Hub* requests the order information (e.g., volumes and transportation categories). Nevertheless, after receiving the data, the *i3FR-Hub* checks that all necessary data for the optimization process is present. For instance, if a new customer or delivery location is present in the orders then the new information is retrieved from the ERP (delivery location) and the *i3FR-Maps* is updated. The *i3FR-Maps* stores the routes between all possible delivery locations, that is, it has a $n \times n$ matrix (n being the number of delivery locations) of routes including distances, travel time and corresponding routing directions details. The data in each entry is obtained in a three step procedure. First, MongoDB [Mon15] calculates distances between locations using spherical geometry. This estimation is stored as a first approximation to the real distance between locations. Then, prioritized by the previously estimated distances, the *i3FR-Maps* uses the Open Street Maps Routing Machine (OSRM) [OSR15] to obtain accurate routing data between the delivery locations. Later, the routes present in the solutions returned by the optimizer are retrieved again but this time using Google Maps [Goo15] which takes into consideration other information (e.g., traffic reports). The use of the OSRM overcomes the limit of accesses to Google's API.

On receipt of the orders, the *i3FR-Hub* posts them to the *i3FR-Opt* module which is divided in two submodules: *i3FR-Opt/Server* and *i3FR-Opt/Optimizer*. The division is thread supported and avoids blocking the optimization procedure while communications are processed, i.e., the *i3FR-Opt/Server* takes care of the communication and the *i3FR-Opt/Optimizer* does the computation of the stochastic capacitated VRPTW solutions. During the process, improved solution obtained by the optimizers are sent to the *i3FR-Hub* which stores them in the *i3FR-DB* and routes that information to the ERP.

Two main factors contribute to the dynamism of the process: (a) the arrival of new orders and (b) partial solution locking by the ERP administrator/decision maker. The first factor was a requirement made such that it would be possible to receive late orders while already optimizing. Late orders should be integrated in the already existing solutions without requiring the reinitialization of the optimization procedure. The second factor has to do with the human intervention of the ERP administrator/decision maker. The ERP does not receive a final solution since the optimization process, being an heuristic, does not guaranty an optimal solution (see Sec. 4). Instead, it receives the best solution computed until the moment by the *i3FR-Opt/Optimizer*. In presence of those solutions, the decision maker can block or close certain routes and prepare the picking and loading of the corresponding vehicles. Since the picking and loading is time consuming, this allows a phased load of the vehicles while continuing the optimization procedure and accepting new orders. In this sense three route states were conceived: "open" meaning that all operations can be made to the route; "blocked" which does not allow altering the order in which the clients are visited, motivated by the fact that the vehicles are loaded in the reverse order of the deliveries, but allows for new orders to somehow be accommodated in the vehicle; and "closed" which does not allow any changes to the route.

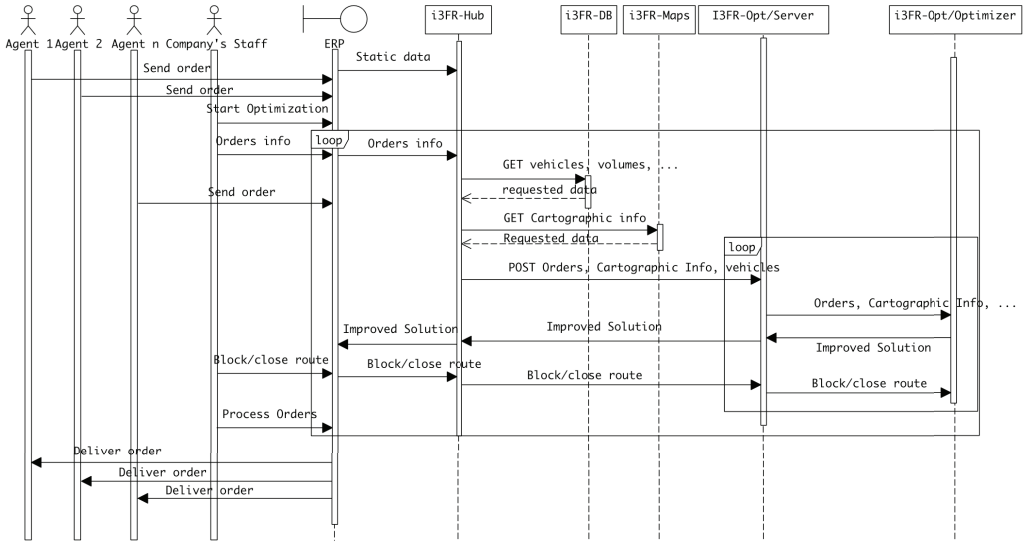

Figure 2: Sequence diagram of the flow from the customers to the optimizer and reverse.

4 Optimization Process

This section briefly describes the optimization procedures which is a hybrid adapted Push Forward Insertion Heuristic (PFIH) with solution seeding and post-optimization. The process, implemented in the *i3FR-Opt*, can be divided into the following stages.

Stage 1. Solution seeding – The optimization procedure starts by computing a seed partial solution, S_{seed}. The partial solution takes into consideration customers' demands and the vehicle capacities to estimate the minimum required number of routes. Once the number of initial routes is computed it is necessary to choose a set of customers and assign them to the new routes. This is done by consecutively choosing the farthest customer from all the routed customers to start a new route. On other words, the first customer to be given a new route is the one farthest away from the depot, c_1. The second route will be started with the customer farther away from the depot and the customer in the first route, c_2. The $n-th$ route will be started with the customer further away from all the previous customers and depot, i.e., $c_n = \arg\max_{i \notin C-C'} \sum_{j \in C'} d(j,i)$, where $C' = \{depot, c_1, c_2, \ldots, c_{n-1}\}$. The motivation to implement this procedure comes from the fact that the selected locations are not, in general, satisfied by the same routes. Possible bad decisions can be later repaired through post-optimization, namely the ejection operator (see Stage 3.). Please refer to [CSME15] for a more detailed explanation of the procedure.

Stage 2. Complete the seeded routes – The second stage consists in the completion of the seeded routes using a PFIH. The PFIH is a greedy constructive heuristic [Sol87, TOS94] proposed by M. Solomon. The method has been implemented and tested by several authors [Rus95, TLO01, Tha99]. In general, the PFIH tour-building procedure sequentially inserts customers into the solution. The procedure can be described by the following steps: (A) Using the seed procedure described in Stage 1 instantiate a set of routes, $S = S_{seed}$, which will contain the final solution; (B) If all customers were placed in a route, then stop the procedure and

return the built solution; Otherwise (C) for all non inserted customers compute their PFIH cost and choose the one with smallest value; (D) Try to insert the customer into an existing route, minimizing the traveled distance and taking into consideration the constraints (customer's time windows and vehicle's capacities); (E) If the insertion of customers is impossible without violating the constraints, start a new route ($depot - customer - depot$) and add it to S; (F) Update the distances, delivery times and vehicle capacities. Return to step (B).

As seen in step (C), the computation of the PFIH cost sets the order in which the customers are inserted in the solution. In our case, the i-customer's cost, $PFIHCost_i$, was defined as $PFIHCost_i = -\alpha d(i,o) + \beta b_i + \lambda(b_i - a_i)^{-1}, i \in C$ where o is a depot, and α, β, and λ are parameters such that $\alpha+\beta+\lambda = 1$. Different α, β and λ allow to give more or less importance to formula parcels, resulting in distinct orderings of the customers. For instance, large values of α will prioritize the insertion of customers near the depot. Larger values of β will make customers with earlier closing window preferable. Finally, larger values of λ will prefer customers with smaller time windows to be inserted first. The steps described are repeated for a set of α, β, and λ parameters (namely, for $(\alpha, \beta, \lambda) \in \{(\alpha, \beta, \lambda) : \alpha, \beta, \lambda \in \{0, 0.1, 0.2, \ldots, 1\} \wedge \alpha + \beta + \lambda = 1\}$) creating a collection of solutions, $PFIHSet$, for the next stage.

Stage 3. Post optimization - The next stage is a cycle which is computed until all routes are closed by the ERP/administrator or all PFIH parameters are tested. The cycle starts by getting (and removing) the most promising solution from $PFIHSet$ and setting an ejection rate value. A tabu list, T, is started which will contain all computed solutions before applying post-optimization, for each ejection rate. Then try at most $MaxTries$ times to improve the solution by applying: (a) a 2-Opt operator (which iterates through all routes, one by one, and tries to rearrange the sequence by which the customers are visited in order to reduce the route distance, maintaining feasibility [CGF$^+$08]); (b) a cross route operator (similar to the One Point Crossover operator of the Genetic Algorithms [MaM13], receives two paths as input, and tries to find a point where the routes can be crossed, thus improving the total distance and without losing feasibility); and (c) a band ejection operator which is a generalization of the radial ejection [SSSWD00](selects a route and, based on the proximity and similarity of the nodes, for each customer located in the route ejects it and a certain number of geographical neighbors which are then reinserted in other routes, without violating the problem's constraints). Please refer to [CSME15] for a more detailed explanation. The first two operators are capable of diminishing the total distance, i.e., doing route optimization. However, they are not capable of reducing the number of routes present in the original solution, which can be achieved using the third operator. The first two stages are quite fast. Therefore it is during the last stage that new orders arriving from the *i3FR-Hub* to the *i3FR-Opt* are treated. On other words, the *i3FR-Opt/Server* thread is responsible for the continuous communications with the *i3FR-Hub*, and whenever new orders arrive they are placed in shared memory. After each cycle the *i3FR-Opt/Optimizer* checks the shared memory for new orders that will be treated as ejected customers, i.e., it tries to insert them in the existing routes or creates a new route if that is not possible. As mentioned, during the process, improved solutions are sent from the *i3FR-Opt* to the *i3FR-Hub* which in turn resends them to the ERP/administrator.

5 Experimental Results

In this section we present a small set of results achieved by the *i3FR* system. For the sake of simplicity, only two scenarios, both with 204 deliveries, for distinct days in the Algarve region where considered (the insertion of new orders after the optimization procedure starts was not

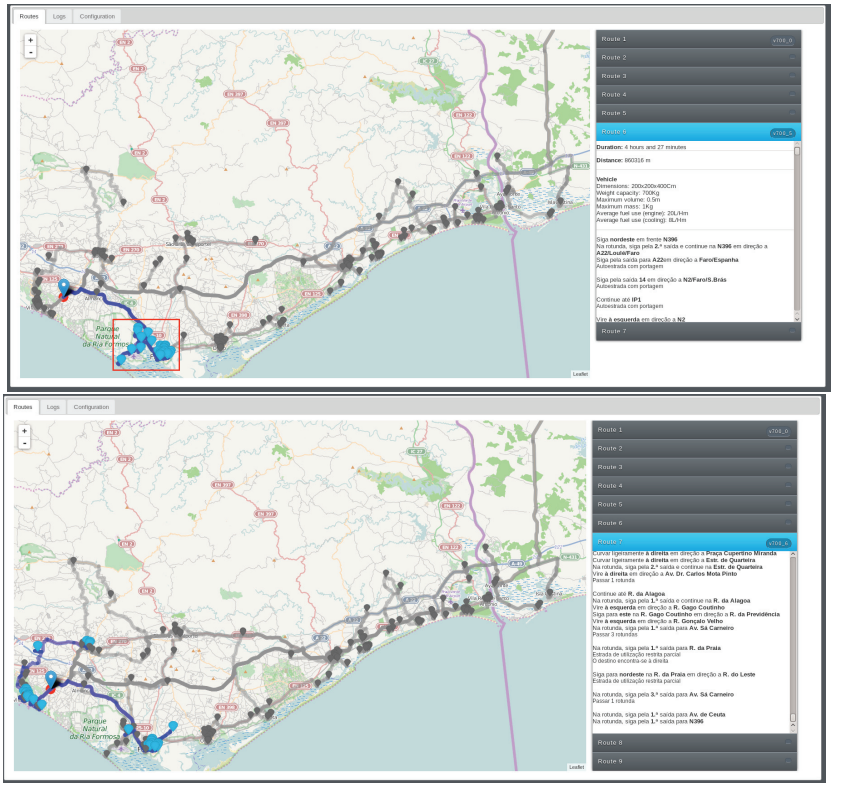

Figure 3: Example of the computed routes represented over the OSM API.

considered). In the first scenario all deliveries had time windows between 9:00 and 17:00. In the second scenario, 52% of the delivery time windows were between 9:00 and 17:00, 28% between 9:00 and 12:00 and the remaining 20% between 14:00 and 17:00. For both cases the routes were obtained after 15 seconds of computation on a commodity computer (Intel i7-4770 processor, with 16Gb of RAM and Kubuntu 14.04).

Figure 3 presents screen shots (first scenario on the top and the second on the bottom) of the back-office interface were the routes where drawn over the Open Street Maps (OSM) API. The depot is marked with the symbol, the active route drawn in blue with delivery locations marked with the symbols, and the the remaining routes in gray with delivery location marked with the symbols. On the right are the route instruction (in Portuguese, the selected language to retrieve the data from Google Maps).

i3FR-Opt returned 7 routes for the less restrictive scenario (the first one). For instance, in this case it is observable that all considered customers from Faro's city, inside the red rectangle, are served by a single route. The second scenario led to a rather different solution with 9 routes where, for instance, more than one route was necessary to serve the same customers in Faro. The additional routes in the second scenario are due to non-intersection of the time windows. Serving clients in the marked area, such as presented in first scenario, would imply considerable waiting times, which is not acceptable in the present business model. In both scenarios the maximum wait time allowed was set to 5 minutes, i.e., a vehicle can arrive up to 5 minutes before the opening of the customers time window.

6 Conclusions and Future Work

This work presented the vehicle routing optimization system developed to be integrated with an existing ERP. The optimization procedure takes into consideration the need for a near real-time routing solution under dynamic orders and interactions with the system administrator. In this sight, this work described the interactions and dependencies between the system's four main components, namely: *i3FR-Opt* (where the computation of the routes is done), the *i3FR-Hub* (implementing a channel to all the communications inside the system and to the exterior), the *i3FR-DB* (provider of local storage to the information relevant to the optimization procedure), and *i3FR-Maps* (a cartography subsystem of routing informations). With this structure it is possible to deal with late orders and different states for the routes, which allows to do a phased picking and loading of the vehicles. As mere examples, some results for the Algarve's region were presented showing different solution depending on the time windows restrictions.

As future work, among other things, the *i3FR* API should be tested with other ERPs and relevant information from the data acquisition module (namely the GPS), referred in the first section, should be used has a memory of the real time taken in each route, validating the values returned by Google Maps and OSRM.

ACKNOWLEDGEMENTS

This work was partly supported by project i3FR: Intelligent Fresh Food Fleet Router – QREN I&DT, n. 34130, POPH, FEDER, the Portuguese Foundation for Science and Technology (FCT), project LARSyS PEstOE/EEI/LA0009/2013. We also thanks to project leader X4DEV – Business Solutions.

References

[AS07] Daniela Ambrosino and Anna Sciomachen. A food distribution network problem: a case study. *IMA Journal of Management Mathematics*, 18(1):33–53, 2007.

[BG02] Olli Bräysy and Michel Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *Top*, 10(2):211–237, 2002.

[CGF+08] Tonči Carić, Ante Galić, Juraj Fosin, Hrvoje Gold, and Andreas Reinholz. A modelling and optimization framework for real-world vehicle routing problems. In Tonci Caric and Hrvoje Gold, editors, *Vehicle Routing Problem*, pages 15–34. InTech, 2008.

[CHC09] Huey-Kuo Chen, Che-Fu Hsueh, and Mei-Shiang Chang. Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, 36(7):2311 – 2319, 2009.

[CSME15] Pedro J. S. Cardoso, Gabriela Schütz, Andriy Mazayev, and Emanuel Ey. Solutions in under 10 seconds for vehicle routing problems with time windows using commodity computers. In *Proc. of the 8th International Conference on Evolutionary Multi-Criterion Optimization, EMO 15*, page (Accepted for publication), 2015.

[GGLM03] Gianpaolo Ghiani, Francesca Guerriero, Gilbert Laporte, and Roberto Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1–11, 2003.

[Goo15] Google Maps. Google Maps. http://maps.google.com, jan 2015.

[GTA99] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. MACS-VRPTW: A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization*. Citeseer, 1999.

[HHL07] Chaug-Ing Hsu, Sheng-Feng Hung, and Hui-Chieh Li. Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering*, 80(2):465–475, 2007.

[Log15] Logvrp.com. Logvrp.com. http://logvrp.com, jan 2015.

[LYL11] Shih-Wei Lin, Vincent F Yu, and Chung-Cheng Lu. A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, 38(12):15244–15252, 2011.

[MaM13] Jorge Magalhães Mendes. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. *WSEAS Transactions on Computers*, 12(4):164–173, 2013.

[Mon15] MongoDB, Inc. MongoDB. http://www.mongodb.com, jan 2015.

[Mou08] Ana Moura. A multi-objective genetic algorithm for the vehicle routing with time windows and loading problem. In *Intelligent Decision Support*, pages 187–201. Springer, 2008.

[New15] Newronia.com. Newronia.com. http://en.newronia.com, jan 2015.

[Opt15a] Optimoroute.com. Optimoroute.com. http://optimoroute.com, jan 2015.

[Opt15b] Optrak.com. Optrak.com. http://optrak.com, jan 2015.

[OS08] Ana Osvald and Lidija Zadnik Stirn. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering*, 85(2):285 – 295, 2008.

[OSR15] OSRM. OSRM – Open Source Routing Machine. http://project-osrm.org, jan 2015.

[PBFI09] Marco Pavone, Nabhendra Bisnik, Emilio Frazzoli, and Volkan Isler. A stochastic and dynamic vehicle routing problem with time windows and customer impatience. *Mobile Networks and Applications*, 14(3):350–364, 2009.

[PGGM13] Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[Rou15] Routyn. Routyn. http://www.routyn.com, jan 2015.

[Rus95] Robert A Russell. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation science*, 29(2):156–166, 1995.

[SAG15] SAGE, ERP X3. SAGE ERP X3, jan 2015.

[Sol87] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[SSSWD00] Gerhard Schrimpf, Johannes Schneider, Hermann Stamm-Wilbrandt, and Gunter Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.

[Tha99] Sam R Thangiah. A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows. *Practical handbook of genetic algorithms*, 3:347–381, 1999.

[TK02] C.D. Tarantilis and C.T. Kiranoudis. Distribution of fresh meat. *Journal of Food Engineering*, 51(1):85 – 91, 2002.

[TLO01] KC Tan, LH Lee, and K Ou. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. *Engineering Applications of Artificial Intelligence*, 14(6):825–837, 2001.

[TM08] Vincent Tam and KT Ma. An effective search framework combining meta-heuristics to solve the vehicle routing problems with time windows. In Tonci Caric and Hrvoje Gold, editors, *Vehicle Routing Problem*, pages 35–56. InTech, 2008.

[TOS94] Sam R Thangiah, Ibrahim H Osman, and Tong Sun. Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. *Computer Science Department, Slippery Rock University, Technical Report SRU CpSc-TR-94-27*, 69, 1994.