An interdisciplinary method for a generic vehicle routing problem decision support system

Bernat Gacias, Julien Cegarra, Pierre Lopez

HAL Id: hal-00370060
https://hal.science/hal-00370060

An interdisciplinary method for a generic vehicle routing problem decision support system

Bernat GACIAS [a,b], Julien CEGARRA [c], Pierre LOPEZ [a,b]

[a] *CNRS; LAAS; 7, avenue du Colonel Roche, F-31077 Toulouse, France*

[b] *Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France*

[c] *CLLE; Université de Toulouse; 5, allées Machado, F-31058 Toulouse, France*

Abstract

In this paper, we propose an interdisciplinary method for the logistics of transportation. It focuses on the design of a generic architecture for the vehicle routing problem (VRP). We highlight that human factors and dynamics aspects are generally ignored in the classical approaches to solve the vehicle routing problem. In our study, a link is done between methods of operations research (specific methods to solve vehicle routing problems and constraint programming techniques) and a work domain analysis technique coming from cognitive ergonomics. The proposed architecture allows to consider and to process the constraints identified by the work domain analysis during the problem resolution. It is also well adapted to the consideration of the Human as a main actor in the decision-making process.

Key words: Decision support system, work domain analysis, vehicle routing problem, constraint programming.

1 Introduction

Nowadays the optimization of the vehicle routing problem (VRP) has become one of the main issues for the companies involved in goods and services production. Indeed, aggressive competition forces them to guarantee a minimum level of quality of service for the customers. In transportation logistics for example, it implies an important effort to satisfy the customer demands at the right time.

This interest from the companies has been transferred to the researchers. For instance, the VRP has been widely studied by the Operations Research scientific community for about 15 years. Therefore, there exists a large number of methods to solve very efficiently the various existing variants of the problem [1,10,18].

However we consider that the traditional way to solve the VRP has two important limitations when considering the human factors and the dynamics of the contextual constraints. In order to overcome these limitations, we propose a generic architecture for a decision support system for the vehicle routing problems. An interdisciplinary approach with two components is proposed: (1) an ecological interface

---

[*] This paper was not presented at any other journal. Corresponding author B. Gacias. Tel. +33-561-336-419. Fax +33-561-336-936.

Email addresses: `bernat.gacias@laas.fr` (Bernat GACIAS), `julien.cegarra@univ-jfc.fr` (Julien CEGARRA), `pierre.lopez@laas.fr` (Pierre LOPEZ).

based on the abstraction hierarchy resulting from a work domain analysis [16,20]; (2) solving mechanisms based on Operational Research techniques, in particular Constraint Programming [7,17].

Both components are described in next sections. Then, an architecture for the decision support system is proposed. A scenario illustrates how the system works for the design and the modification of the routes. Finally, testing prospects on real problems are referred.

2 Vehicle routing problems

The *vehicle routing problem* (VRP) consists in determining the routes of a fleet of vehicles for the transportation of goods or passengers according to some customer demands (delivery, pick-up...) (Figure 1).
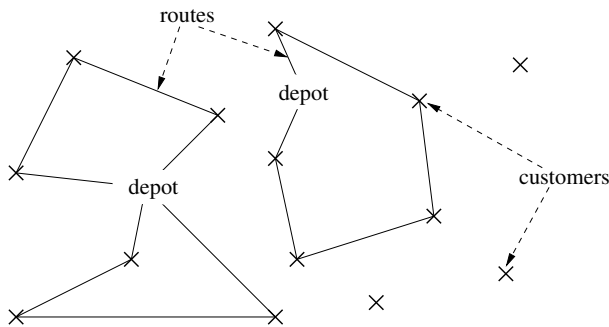

Fig. 1. Vehicle routing

The changes of vehicles fleets in real environment spark off a large number of constraints to take into account for determining the routes. Operational research methods exist to efficiently solve the various existing variants of the problem [1,10,18], for example metaheuristics such as tabu search [4,6] or genetic algorithms [3,15].

However we consider that the traditional way to solve the VRP has two important limitations. The first one is that human factors are not much considered in the modeling phase of the problem. Now, humans play a major role by carrying out specific operations and making decisions whenever perturbations occur. For example, when supervising the routes, the user of the resolution methods has enough knowledge and know-how to anticipate emergencies, vehicle breakdowns, traffic jams, driver substitutions, and so on (see [5] for a discussion on human contributions).

It is important to allow the human be part of the system. We consider that the robustness of the proposed solutions may increase if the human is allowed to act on constraints. It is usually noted that experienced individuals build schedules that are robust enough to disturbances [5]. For example, the experienced individuals may assign the operations to the least flexible resources with the intention of preserving the most flexible resources for later unexpected disturbances. Another example is the use of more than one hundred different "human" heuristics to tentatively anticipate problems in order to build robust schedules.

The second limitation is that these models are not ready to deal with the rapid changing situations. In the logistics domain the constraints may even change, in some extreme cases, before the end of the modeling phase. The model still has to remain valid recovering these changes.

In the litterature, we find some works with similar objectives [12,13]. In [12], the author proposes a job scheduling system, based on the work domain analysis, where human factors are highly considered but operational research methods are not exploited. In [13], the authors present a human-machine system based on constraint propagation to solve the job scheduling problem. The authors themselves admit the lack of human consideration in the problem resolution.

3 Constraint programming

3.1 Generalities

To integrate, interactively and incrementally, the constraints into the solving algorithms, we use Constraint programming (CP) techniques [7,17]. The CP paradigm presents some interesting advantages for

the problematics at hand.

First, it allows us to consider separately the formal definition of the problem (CSP), analysis mechanisms (propagation techniques) and the solving itself (Figure 2). This characteristic is probably the main originality of CP; this is particularly interesting for devising flexible systems and for considering the cooperation inside the Human-Machine system.
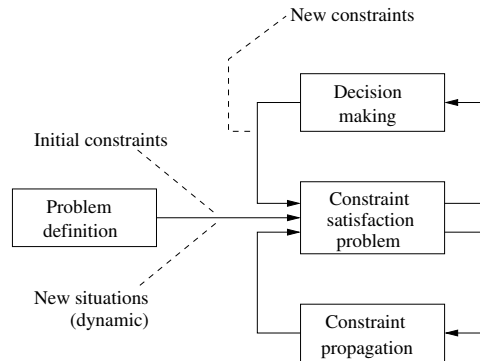

Fig. 2. Constraint programming

Besides, with the CP approach we have the advantage to associate a particular processing for each type of constraints following the assertion "1 constraint = 1 algorithm". This principle allows us to decompose the global system of constraints in as many sub-problems as different types of existing constraints, and in this way to specialize the constraint propagation process (filtering the inconsistent values of a variable domain).

Another advantage is that side constraints or preferences can be considered. These constraints are rather particular since they do not define hard restrictions. Provided that these side constraints might over-constrain (that is, make unsatisfiable) the problem under consideration, we can then solve a so-called "maximum Constraint Satisfaction Problem" (Max-CSP in short). In particular the CSP theory offers a formal framework to deal with *weighted CSPs* (wCSPs), where the objective is to minimize the total weight of unsatisfied side constraints. In the same context, CP allows us to consider techniques based on *model inversion* in order to determine the constraints to relax in case of unsatisfiable problems [2]. After this analysis, the user could modify the constraints following the propositions of the model inversion techniques or he might choose to relax other more suitable constraints, using his knowledge of the real problem, in order to obtain a satisfiable problem.

Finally, we can reach a reactive behavior particularly adapted to the system dynamics by incrementally adding new constraints and more particularly studying the formalisms of dynamic CSPs [14] and mixed and conditional CSPs [9].

3.2 Specific algorithms for the VRP

An advantage of the proposed approach is the possibility to mix CP mechanisms and other specific solving techniques. In [8], the authors propose a methodology based on shortest path determination to associate the solving techniques to the real problem. These associations are made from the similarities between the real problem, once we have the model, and the classical problems from the literature for which the efficient methods and algorithms are well known.

This idea might be easily incorporated in the proposed system. After applying CP techniques, we may consider to use other solving techniques which are very efficient to solve problems close to the real problem. The decisions made by these techniques and algorithms may be then returned to the CP solving mechanism.

4 Work domain analysis

We propose a work domain analysis (WDA) for the VRP. This analysis method facilitates the identification of the problem constraints. It is a first step before the development of a solving system. We believe

that the model derived from this kind of analysis can be well adapted to take into account the user and more generally the human factors perspective (see for instance [11,12]). The underlying idea is that if the model contains all constraints having an influence on the way the solution is built, it also considers constraints relating to the Human. On the other hand, if the great majority of constraints are considered in the model, normally it will be adapted to deal with the unexpected and to resist the long-term changes of the situations.

We propose an early decomposition of the domain, analysis and identification of the structural limits of the problem. We use the abstraction hierarchy proposed in [16,20] for this analysis. The abstraction hierarchy ensures an exhaustive decomposition of the work domain necessary to take into account all the restrictions (Figure 3).
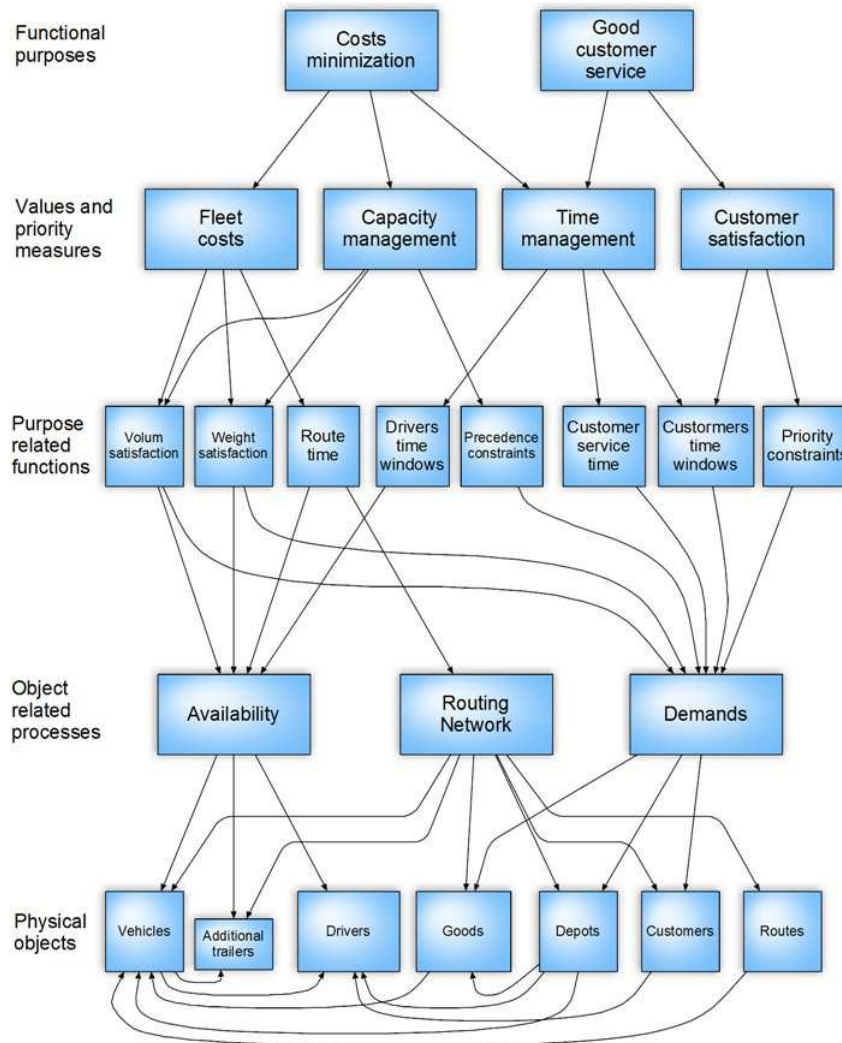
Fig. 3. Work domain analysis of the VRP using an abstraction hierarchy

The abstraction hierarchy levels represent the levels of user reasoning to solve the problem. We decide to keep the 5 levels proposed in [20] which are all necessary for the considered problem (see Section 6).

At first level, we find the primary objectives of the work system. For the VRP, these objectives are costs minimization and to keep a satisfying quality of service. At second level, we find the criteria to be used to decide whether the work system is achieving its purposes. These criteria are fleet costs, capacity management, time constraints management (customer time service and time windows) for costs minimization and demands satisfaction, and also time constraints to reach a sufficient quality of service.

As we can see, sometimes the criteria may be contradictory. For example, a delivery delay might lead to a decrease of transportation costs, by using fewer drivers for example. We propose to offer several

evaluation criteria in order to pick the best adapted solution (for example, increase customer satisfaction or decrease transportation costs).

The functions required to achieve the purposes are shown at third level. These functions point the constraints to consider in order to achieve the objectives of the system. Finally, on the two remaining levels, we find the physical objects of the vehicle routing problem (vehicles, drivers, merchandises, depots, customers, demands, and routes) and the process defining the capacities and limits of physical objects. For example, the number of drivers may be limited, and this constraint has to be considered for CP (to determine a solution satisfying the constraints) and for the human (who could, for instance, looking for additional drivers in order to relax the constraint).

The decomposition of the problem remains very generic considering studies and problem descriptions found in the litterature. A validation for different real problems is necessary. Whatever, we consider that most of the vehicle routing problems are covered for the proposed work domain analysis and only some particular problems may required other components not represented on it.

5 Proposed architecture

We consider the WDA as a first step for the design of a VRP decision support system (DSS). A solving system based on optimization techniques is also part of the DSS. The user could interact with the DSS through an *Ecological Interface* (EI) [21,22]. The human could participate in the modeling and also in the problem solving (Figure 4). The Ecological Interface is the interface capable to represent the abstraction hierarchy as an external mental model for the resolution of the problem. The physical and the functional information are displayed for the interface in order to make constraints and complex relationships in the work environment perceptually obvious to the user. This allows more of users' cognitive resources to be devoted to higher cognitive processes such as problem solving and decision making. By reducing mental workload and supporting knowledge-based reasoning, EI aims to improve user performance and overall system reliability for both anticipated and unanticipated events in a complex system.
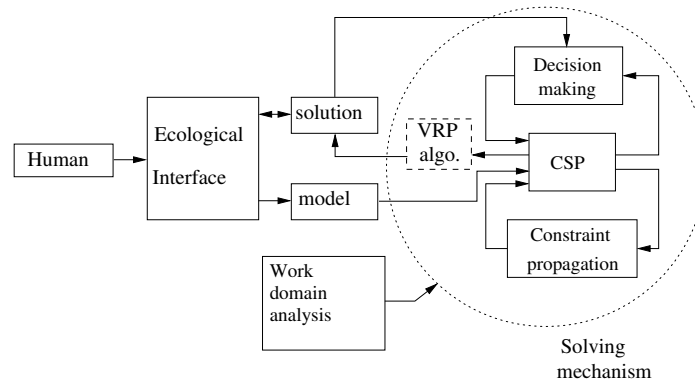

Fig. 4. Decision support system

As we can see, in the proposed architecture for the decision support system the tasks are shared between the solving mechanism and the human.

The main task for the user is to define the restrictions of the problem; mainly, he has to select which constraints are activated during the execution on line. In the case of unsatisfiable problems, this selection might be guided by the propositions resulting from the model inversion techniques. He could also express his preferences for the resolution strategy (this feature will also be taken into account for the selection of the resolution algorithms) and he could modify all problem data. Finally, the user picks the solution to be executed.

On the other hand, the tasks assigned to the solving mechanism (solving methods, propagation techniques, initial heuristics) are: to select the algorithms to be useful depending on the context; to propose and to evaluate a set of feasible solutions. In order to help the user to choose the solution, we propose a visual branching scheme of the solutions space using the criteria to judge the performance of the solutions. That

way, the user has the possibility to choose a solution taking into account several objectives or preferences that were not announced in the model but can help to increase the efficiency of the solution.

The other feature that supports the user to deal with the unexpected is that the system must offer the possibility to locally modify and re-evaluate a solution once such solution (or a set of solutions) is proposed.

6 Illustrative example

We illustrate how the proposed system is going to work with a simple scenario of a vehicle routing problem.

We consider a 4-customer problem. Each customer $C_{i,i=1..4}$ has a demand of merchandise $d_{i,i=1..4}$ and a service time window to respect $TWC_{i,i=1..4}$. A fleet of 3 vehicles $V_{j,j=1..3}$ is at our disposal with a fixed capacity ($Q = 7$ units of product) to serve the customers. The routes start and finish at the unique available depot C_0. 3 drivers $D_{k,k=1..3}$, with their time windows $TWD_{k,k=1..3}$, are available. The drivers time windows allow us to consider workers shifts and time constraints due to work regulations.

In Table 1, we show the problem data, and customers and drivers time windows. Besides, we consider an allocation vehicle-customer (merchandise) constraint: customer C_4 has to be served by vehicle V_2. This constraint may correspond for example to the fact that the demanded products are different, and V_2 is the only vehicle adapted to transport the merchandise of C_4. In order to simplify the example the distances of the route network (depot and customers) are not considered.

As we have explained, the proposed system allows the user to choose the best solving strategy for the real problem. In the example, the user imposes to serve all customers.

(a) Customers data				(b) Drivers data	
i	d_i	TWC_i		k	TWD_k
C_1	2	$[4,8]$		D_1	$[0,8]$
C_2	3	$[5,10]$		D_2	$[0,8]$
C_3	2	$[0,2] \cup [10,12]$		D_3	$[0,8]$
C_4	5	$[8,12]$			

Table 1
Example data

Once the model is obtained and the user has activated the different constraints, the abstraction hierarchy allows us to act on all features of the scenario (stressed in grey in Figure 5), the solving mechanism starts the constraint propagation. We begin for the time windows constraints, drivers D_1, D_2, and D_3 can serve customers C_1, C_2, C_3 but nobody can serve customer C_4. The model inversion techniques identified that the problem is unsatisfiable because C_4 cannot be served. The user analyses the situation and takes the best solution to solve the conflict. In that case the user has several options, for example to ask to C_4 whether it is possible to serve him inside $[0,8]$ time interval, or discard to serve C_4, or to hire a new driver. Among all of the options we suppose that the user decides to split the time window for driver D_1 ($TWD_1 = [0,4] \cup [8,12]$). Then D_1 serves C_4.

From the allocation constraint V_2-C_4, we deduce that D_1 is necessarily allocated to V_2. Now, we also have that C_1 cannot be served by D_1 because of the new time window for the driver. So C_1 has to be assigned to D_2 or D_3. We are able to discard the solutions where C_1 and C_4 are at the same route, because they have to be served by different drivers.

Finally, we take into account the vehicle capacity ($Q = 7$). The solutions with C_2 and C_4 at the same route are eliminated. Two sets of solutions are satisfiable: either drivers D_2 and D_3 serve customers $\{C_1, C_2, C_3\}$ and driver D_1 serves customer $\{C_4\}$, or drivers D_2 and D_3 serve customers $\{C_1, C_2\}$ and driver D_1 customers $\{C_3, C_4\}$.
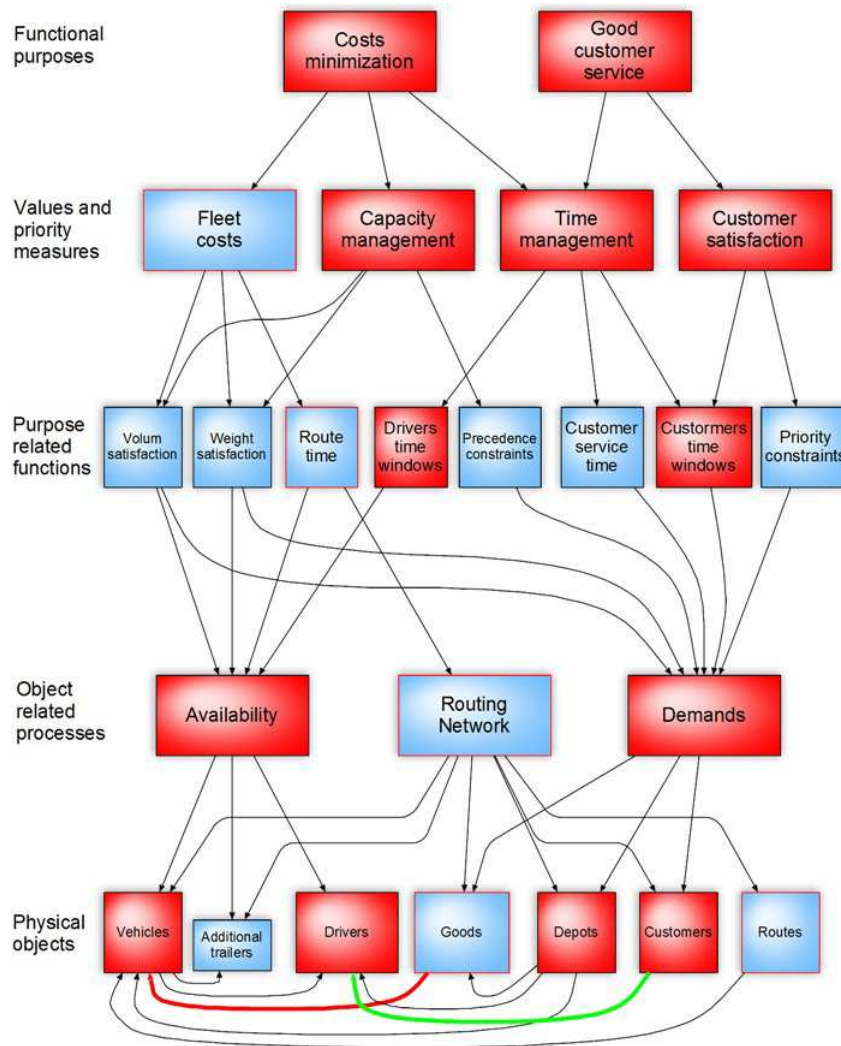
Fig. 5. Modeling the problem under study

Now, the user pick the best adapted solution for the problem. In the real vehicle routing problems there exist several criteria to make the choice of the solution. For that reason, we propose a visual branching scheme using the main criteria and the user preferences (number of kilometers, number of vehicles, number of delays...) as a choice of branching. Because, for example, the user could prefer to increase the total time of route but using only two vehicles (with D_1 and D_2). Finally, customers $\{C_1, C_2, C_3\}$ are assigned to D_2 and $\{C_4\}$ to D_1.

Now, we consider an unexpected event such as driver D_2 is late. The user analyses the situation and observes that D_2 could have some problems to serve customer C_3 because he has only the first two hours of his shift to serve him. The decision of the user is to allocate customer C_3 to D_1. That decision is integrated in the solving mechanism. The propagation mechanism starts again, but now only one kind of solution is possible; D_2 serves customers $\{C_1, C_2\}$ and D_1 the customers $\{C_3, C_4\}$.

The solving mechanism propagates the times of route, and we find that C_3 must be served before C_4 by D_1, and D_2 has to serve $\{C_1, C_2\}$ but we don't know the direction of the route. The solutions are evaluated taking into account the preferences (if there exist) of the user for the visual branching scheme and he chooses the most suitable set of routes for the real problem.

7 Expected results and conclusions

In this paper, we have proposed a decision support system architecture for the vehicle routing problem. The properties of the system are well adapted to the problem topology. The main contribution of this interdisciplinary approach is to base the initial model of the problem on a work domain analysis

which allows the identification of the structural boundaries of the problem to supervise the constraint programming. Besides, the solving mechanisms are compatible with potential user interventions.

We have also carried out a work domain analysis (WDA) for the vehicle routing problem based on an abstraction hierarchy. It has to be the starting point for the ecological interface design suited to the problem. The human-machine interface based on the proposed WDA, the constraint programming approach and the model proposed in [19] is being developed. The validation of the interface is planned as future work.

In this context, contacts with a company involved in waste collecting have been established. We are going to propose an interface able to take into account all the problem variables just as three different problem cases: home rubbish collection, industrial waste collection and recyclable product collection. Common properties are shared by the three problem variants but each one needs different solving methods. Once again the proposed decision support system is well adapted for the problem.

Finally, we plan to compare the system with two works with similar objectives [12,13]. This theoretical and methodological comparison may point the difficulties that could be found by the researchers when interdisciplinary approaches are setting up.

References

[1] C. Barnhart and G. Laporte, editors. *Handbooks in operational research and management science: Transportation*, volume 14. North-Holland, 2007.

[2] I. Bazet, G. de Terssac, and J. Erschler. La gestion des contraintes : conflit et combinaison de règles. *Actes du colloque "Recherche et Ergonomie"*, pages 100–107, Toulouse, 1998.

[3] J. Berger and M. Barkaoui. A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 54:1254–1262, 2003.

[4] O. Bräysy and M. Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 10(2):211–237, 2002.

[5] J. Cegarra. A cognitive typology of scheduling situations: A contribution to laboratory and field studies. *Theoretical Issues in Ergonomics Science*, 9(3):201–222, 2008.

[6] J-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 58(8):928–936, 2001.

[7] R. Dechter. *Constraint Processing*. Morgan Kaufmann, San Francisco, USA, 2003.

[8] M. Desrochers, C. V. Jones, J. K. Lenstra, M. W. P. Savelsbergh, and L. Stougie. Towards a model and algorithm management system for vehicle routing and scheduling problems. *Decision Support Systems*, 25(2):109–133, 1999.

[9] E. Gelle and B. Faltings. Solving mixed and conditional constraint satisfaction problems. *Constraints*, 8(2):107–141, 2003.

[10] B. Golden, S. Raghavan, and E. Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43. Springer, 2008.

[11] P. G. Higgins. *Job Shop Scheduling: Hybrid Intelligent Human-Computer Paradigm*. PhD thesis, University of Melbourne, Australia, 1999.

[12] P. G. Higgins. *Human performance in planning and scheduling: fieldwork studies, methodologies and research issues*, chapter "Architecture and Interface Aspects of Scheduling Decision Support", pages 245–279. Taylor and Francis, London, 2001.

[13] O. Lassila and S. Smith. Toward the development of mixed-initiative scheduling systems. In *ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop Proceedings, San Francisco (CA)*, pages 145–154, 1994.

[14] S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In *Proceedings of AAAI-90, Boston , MA*, pages 25–32, 1990.

[15] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[16] J. Rasmussen, A. M. Pejtersen, and L. P. Goodstein. *Cognitive Systems Engineering*. Wiley, New York, 1994.

[17] F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.

[18] P. Toth and D. Vigo, editors. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[19] W. van Wezel and J. Cegarra. Developing user-oriented scheduling algorithms. In *14th International Annual EurOMA Conference, Ankara, Turkey*, 2007.

[20] K. J. Vicente. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1999.

[21] K. J. Vicente. Ecological interface design: Supporting operator adaptation, continuous learning, distributed, collaborative work. *Proceedings of the Human Centered Processes Conference*, pages 93–97, 1999.

[22] K. J. Vicente. Ecological interface design: Progress and challenges. *Human Factors*, 4:62–78, 2002.