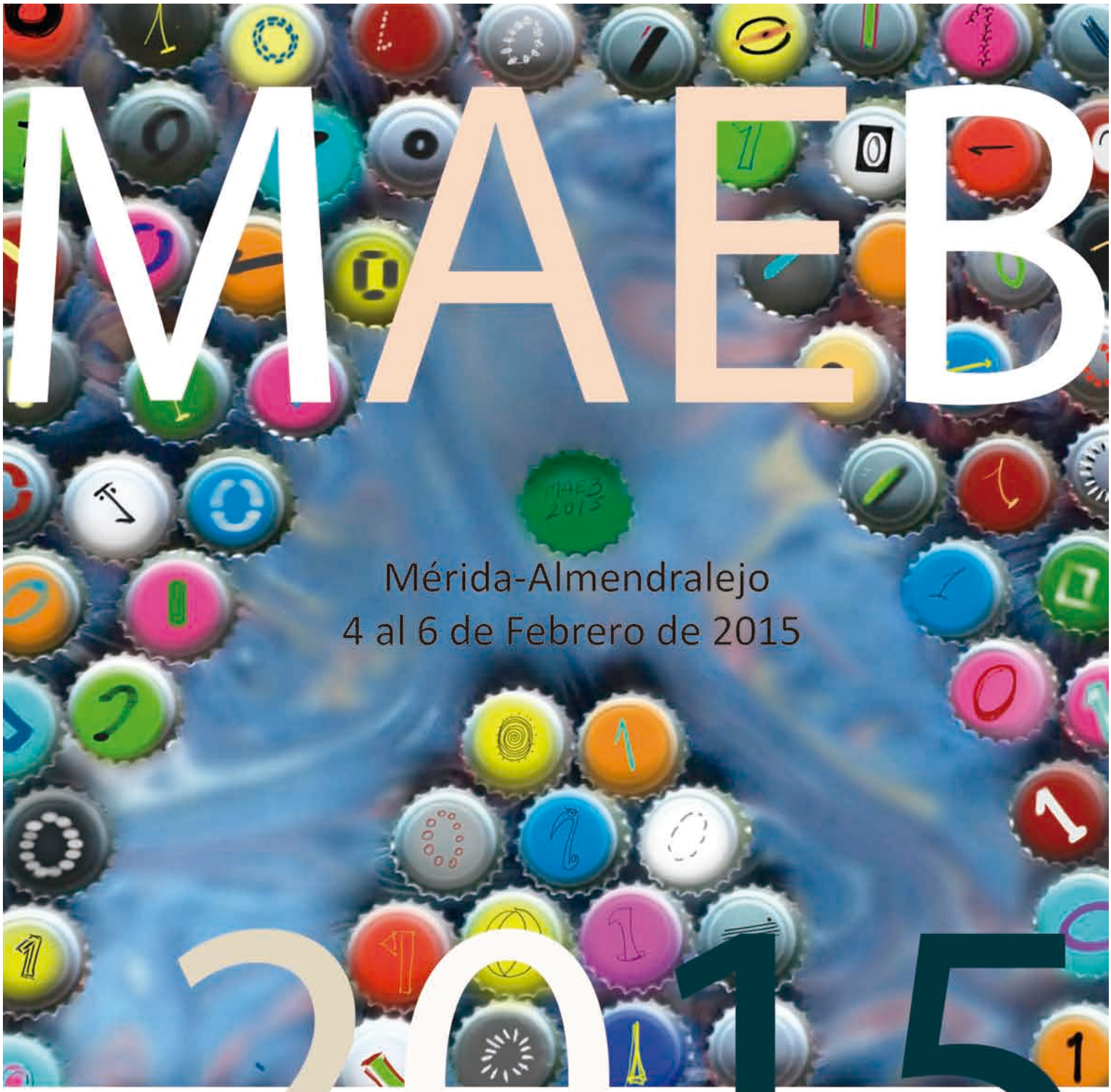


X Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados
MAEB 2015

Actas del X Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados

MAEB 2015

Editadas por:

Francisco Chávez de la O
Rafael M. Luque Baena
Francisco Luna
Francisco Fernández de Vega

Mérida - Almendralejo
4, 5 y 6 de febrero de 2015

Edita: Francisco Chávez de la O, Rafael M. Luque-Baena, Francisco Luna, Francisco Fernández de Vega
Centro Universitario de Mérida
Universidad de Extremadura

Derechos reservados

© Los autores

Diseño de cubierta: Cayetano Cruz García

ISBN: 978-84-697-2150-6

Comité Organizador

Cayetano Cruz García
Universidad de Extremadura

Josefa Díaz Álvarez
Universidad de Extremadura

Francisco Chávez de la O
Universidad de Extremadura

Francisco Fernández de Vega
Universidad de Extremadura

Francisco Luna
Universidad de Extremadura

Rafael M. Luque-Baena
Universidad de Extremadura

Comité Director

Enrique Alba
Universidad de Málaga

Abraham Duarte
Universidad Rey Juan Carlos

Francisco Fernández de Vega
Universidad de Extremadura

José Antonio Gámez
Universidad de Castilla-La Mancha

Francisco Herrera
Universidad de Granada

J. Ignacio Hidalgo
Universidad Complutense de Madrid

César Hervás
Universidad de Cordoba

Rafael Martí
Universidad de Valencia

Juan Julián Merelo
Universidad de Granada

José A. Moreno
Universidad de La Laguna

Luciano Sánchez
Universidad de Oviedo

Comité de Programa

Jesús Aguilar, Universidad Pablo de Olavide
Enrique Alba, Universidad de Málaga
Carlos Andrés Romano, U. Politécnica de Valencia
Ada Álvarez, Universidad Autónoma de Nuevo León
Ramón Álvarez-Valdés, Universidad de Valencia
Lourdes Araujo, Universidad Nacional a Distancia
Jaume Bacardit, University of Nottingham
Julio R. Banga, CSIC
Joaquín Bautista, Universidad Politécnica de Cataluña
José Manuel Benítez, Universidad de Granada
Christian Blum, Universidad Politécnica de Cataluña
Rafael Caballero, Universidad de Málaga
Vicente Campos, Universitat de València
Jorge Casillas, Universidad de Granada
Pedro A. Castillo Valdivieso, Universidad de Granada
Francisco Chicano, Universidad de Málaga
Francisco Chávez, Universidad de Extremadura
Carlos A. Coello Coello, CINVESTAV - IPN
Ángel Corberán, Universidad de Valencia
Oscar Cordón, Universidad de Granada
Carlos Cotta, Universidad de Málaga
Antonio Córdoba, Universidad de Sevilla
Bernabé Dorronsoro, Universidad de Luxemburgo
Abraham Duarte, Universidad Rey Juan Carlos
Richard Duro, Universidad da Coruña
Adenso Díaz, Universidad de Oviedo
José Egea, Universidad Politécnica de Cartagena
Francisco Javier Elorza, U. Politécnica de Madrid
Antonio J. Fernández, Universidad de Málaga
Francisco Fernández, Universidad de Extremadura
Miguel Frade, Instituto Politécnico de Leiria
Mario Garcia, Instituto Politécnico de Tijuana
Maribel García Arenas, Universidad de Granada
Eduardo García Pardo, Universidad Rey Juan Carlos
Carlos García, Universidad de Córdoba
Nicolás García, Universidad de Córdoba
Salvador García, Universidad de Jaén
Raúl Giraldez, Universidad Pablo de Olavide
José Luís González-Velarde, Inst. Tec. de Monterrey
Antonio González, Universidad de Granada
Pedro González, Universidad de Jaén
José Antonio Gutiérrez, Universidad de Córdoba
José Antonio Gámez, U. de Castilla-La Mancha
Juan A. Gómez Pulido, Universidad de Extremadura
Francisco Herrera, Universidad de Granada
Cesar Hervás, Universidad de Córdoba
José Ignacio Hidalgo, U. Complutense de Madrid
María José del Jesús, Universidad de Jaén
Angel A. Juan, Universitat Oberta de Catalunya
Manuel Laguna, Universidad de Colorado
Dario Landa Silva, University of Nottingham
Helena Ramalhinho Lourenco, U. Pompeu Fabra
José Antonio Lozano, Universidad del País Vasco
Manuel Lozano, Universidad de Granada
Francisco Luna, Universidad de Málaga
Gabriel J. Luque, Universidad de Málaga
Rafael M. Luque-Baena, Universidad de Extremadura
Luís Magdalena, European Centre for Soft Computing
Rafael Martí, Universitat de València
Francisco Martínez, Universidad de Córdoba
Belén Melián, Universidad de La Laguna
Alexander Mendiburu, Univ. del País Vasco
Julián Molina, Universidad de Málaga
J. Marcos Moreno, Universidad de La Laguna
José A. Moreno, Universidad de La Laguna
Antonio J. Nebro, Universidad de Málaga
Julio Ortega, Universidad de Granada
Domingo Ortiz, Universidad de Córdoba
Luis de la Ossa, Universidad de Castilla-La Mancha
José Otero, Universidad de Oviedo
Joaquín Pacheco, Universidad de Burgos
Juan J. Pantrigo, Universidad Rey Juan Carlos
Francisco Parreño, Universidad de Castilla-La Mancha
David Pelta, Universidad de Granada
Antonio Peregrín, Universidad de Huelva
José Miguel Puerta, U. de Castilla-La Mancha
Cynthia Pérez, Tecnológico de Sinaloa
M. Elena Pérez, Universidad de Valladolid
Juan R. Rabuñal, Universidad da Coruña
Ignacio Requena, Universidad de Granada
José Riquelme, Universidad de Sevilla
Jose Luís Risco-Martín, U. Complutense de Madrid
Víctor Rivas, Universidad de Jaén
Rubén Ruiz, Universidad Politécnica de Valencia
Roger Ríos, Universidad Autónoma de Nuevo León
Sancho Salcedo, Universidad de Alcalá
Roberto Santana, Universidad Politécnica de Madrid
Antonio Sanz Montemayor, U. Rey Juan Carlos
Thomas Stützle, Université Libre de Bruxelles
Yago Sáez, Universidad Carlos III de Madrid
Ana María Sánchez, Universidad de Granada
Luciano Sánchez, Universidad de Oviedo
Leonardo Trujillo, Instituto Tecnológico de Tijuana
Ángel Udías, Universidad Rey Juan Carlos
Miguel A. Vega Rodríguez, U. de Extremadura
Sebastián Ventura, Universidad de Córdoba
José Luís Verdegay, Universidad de Granada
Gabriel Villa, Universidad de Sevilla
Pedro Villar, Universidad de Granada
Juan Villegas, Universidad Autónoma Metropolitana
Gabriel Winter, U. de las Palmas de Gran Canaria
Amelia Zafra, Universidad de Córdoba

Presentación

El presente volumen contiene los trabajos presentados en el X Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'15), celebrado en Mérida y Almendralejo, los días 4, 5 y 6 de febrero de 2015.

En esta edición, se recogen 97 trabajos, que han sido cuidadosamente revisados por un amplio comité de programa internacional, y que muestran las líneas más actuales de investigación en el dominio.

Esta décima edición de MAEB ha permitido volver a Mérida, punto de inicio de esta serie de congresos que se han venido celebrando bianualmente desde el año 2002, y que desde entonces ha recorrido la geografía Española.

MAEB es hoy una referencia como lugar de encuentro e intercambio de conocimientos y experiencias entre investigadores del área de las metaheurísticas, los algoritmos evolutivos y bioinspirados, y ha llegado a consolidarse como uno de los más exitosos congresos nacionales del área TIC.

Los miembros del comité de organización quieren expresar el más sincero agradecimiento a todos los autores, ponentes, organizadores de sesiones especiales, miembros del comité de programa, comité director y asistentes en general, así como a todas las instituciones colaboradoras que han permitido que esta nueva edición sea una realidad.

Damos la bienvenida a todos, deseando que esta edición de MAEB nos enriquezca científicamente y nos brinde la oportunidad de conocer un poco más de cerca Extremadura y algunas de sus ciudades más significativas.

El comité organizador

A Refined GRASP Algorithm for the Extended Car Sequencing Problem

Elvira Laković[1], Manuel Chica[1], Sergio Damas[1], Joaquín Bautista[2], and Oscar Cordón[1,3]
[1]European Centre for Soft Computing, 33600 Mieres, Spain
[2]ETSEIB, Universitat Politecnica de Catalunya, 08028 Barcelona, Spain
[3]DECSAI and CITIC-UGR, University of Granada, 18071 Granada, Spain

Abstract— **In this study, we investigate the performance of GRASP (Greedy Randomized Adaptive Search Procedure) mataheuristic that has been used for solving extended optimization version of CSP (Car Sequencing Problem) [15] with respect to different parameters and proposed modifications for the algorithm. While maximum appearance frequency of option in segment is usually a hard constraint, in this study we treat all capacity constraints as soft constraints by giving them same priority when evaluating candidates for the position in the sequence, in order to achieve more balanced sequences as initial solutions for the local improvement phase. The main disadvantage of the described greedy algorithm using static value of α GRASP parameter is that it uses all the good options at the beginning of the sequence, leaving the worse options for the last part of the sequence. Based on this, changing the starting position of the sequence and order of moves for the local search has been investigated. Results show that the starting position of the sequence for the local search matters and taking into account changing the starting position of the sequence for exploring moves for the local improvement can lead to better solutions.**

Key words— **Car sequencing problem, GRASP, Metaheuristics**

I. Introduction

Since its first introduction [1] car sequencing problem (CSP) has drawn a lot of attention and has been widely studied. CSPs are constraint satisfaction problems [2]. The goal of CSP is to find specific schedule for manufacturing cars along assembly line while satisfying capacity constraints. The problem has been shown to be NP-hard [3], [4].

Addressing CSP as a satisfiability problem includes using exact approaches like integer and constraint programming [5], [6], [7]. Integer programming approach where the objective is the minimization of number of violations of the constraints has been proposed in [7]. Constraint programming has been shown to be effective for instances with as many as 200 cars to sequence and utilization of options up to 90 % [5]. Constraint programming is mainly based on filtering algorithms [12]. In order to solve some instances these algorithms are efficient in reducing search space. However, different approaches based on heuristic like local search [8], [9], [10], genetic algorithms [12] and ant colony optimization [10] have been proposed in order to avoid exhaustively and try to find approximately optimal solutions.

In most of the cases local search uses initial sequences constructed by random permutation of the set of cars to manufacture. The construction of initial sequences in a greedy way has been proposed in [10]. They compared six different greedy heuristics for the car sequencing problem and have shown that heuristic significantly improves the search process. Many local search approaches have been proposed [8], [9], [10] and their performances have been demonstrated on benchmark instances of the CSPLib [13], some of which we are also using in our experimental study.

From an industrial point of view some sequences may be more attractive than the others. The need for integration of additional options in each car calls for more balanced sequences by utilization of options in segments of specific size. Even though workloads are undesirable in terms of efficiency, in real industrial systems daily production needs to be carried out. From that perspective, sequences with balanced workloads over the entire time frame are more desirable than those with workloads at the beginning or the end of the time frame. Violations of capacity constraints do not have to be of the same severity and therefore they can be penalized with different weights. Converting CSP from a feasibility problem into optimization problem, where the goal is to minimize maximum load capacity constraint violations, has been proposed in [14]. In the same study they consider a beam search approach for proposed conversion. Hybrid approach based on ACO, beam search and constraint programming has been proposed in [16]. Latest study shows that the Lagrangian-ACO hybrid is best performing method for up to 300 cars for the optimization version of car sequencing [17]. For purposes of regulating both workloads and usage of components over time, additional group of soft constraints based on the minimal appearance frequency of option in segment is introduced in extension of proposed optimization version of CSP (xCSP) [15]. This way more balanced sequences are obtained which leads to more regular production flow.

In this study, we investigate the performance of GRASP (Greedy Randomized Adaptive Search Procedure) mataheuristic introduced by Feo and Resende to solve set covering problems [19], that has been used for solving extended optimization version

of CSP [15] with respect to different parameters and proposed modifications for the algorithm. First focus of the study is on the effectiveness of greedy phase of GRASP guided by heuristic that involves evaluating candidates in an equal weighted manner. This means treating all capacity constraints as soft constraints and putting equal priority in minimizing their violation when electing candidates for the position in the sequence. Second focus of the study is on the effectiveness of changing starting position of the sequence for the local search algorithm that uses solutions constructed during greedy phase of GRASP as initial.

This paper is structured as follows. Formalization of problem is presented in section 2. Section 3 describes GRASP metaheuristic and explored modifications for both greedy and local improvement phases. Experiments carried out along with discussion of the results are in section 4. Conclusions and plans for further research are given in section 5.

II. The Car Sequencing Problem

A. Formalization of Basic Model

Let (C, O, D, d_i, o_{ji}, p_j, q_j) represent an instance of car sequencing problem where C is the number of different car classes, O is the number of different options required for installation, D is the total number of cars to sequence in the considered time frame, d_i is the number of cars of class i=1...C to sequence ($\sum d_i$ = D), o_{ji} is the number of units of option $j = 1...O$ required for installation by car of class $i = 1...C$ and p_j/q_j form a capacity constraint which means that for any sequence of q_j consecutive cars, at most p_j of them may require an option $j = 1...O$.

For $\forall t = 1...D$ and $\forall j = 1...O$ there is a segment [$l_j(t)$, t] of length π where $|\pi| \leq q_j$, $l_j(t) = max[1, t+1-q_j]$ is the starting position of that segment and x_{it} equals to 1 if car of class i will occupy position t in the sequence, otherwise equals 0. The solution process of the basic model for the CSP consists in finding right values of x_{it} that satisfy the following constraints:

$$\sum_{i=1}^{C} x_{it} = 1, \forall t = 1...D \tag{1}$$

$$\sum_{t=1}^{D} x_{it} = d_i, \forall i = 1...C \tag{2}$$

$$\sum_{i=1}^{C} \sum_{k=l_j(t)}^{t} o_{ji} x_{ik} \leq min\{t, p_j\} \quad \forall = 1...C, \forall j = 1...O, \forall t = 1...D, \quad l_j(t) = max[1, t+1-q_j] \tag{3}$$

$$x_{it} \in \{0, 1\}, \forall i = 1...C, \forall t = 1...D \tag{4}$$

Constraint set (1) makes sure that in every position t of the sequence only one unit is placed for production; set (2) ensures that all cars of each required class are placed in the sequence; set (3) form a capacity constraint which ensures that for any sequence of q_j consecutive cars, at most p_j of them may have an option j=1...O; set (4) provide integrity of x_{it}.

This model allows incorporation of the additional constraints in the assignment of options in the first positions of the sequence by considering intervals smaller than q_j, opposed to the traditional model that usually limits the range of t values between q_j and D for the constraints set (3) [15].

B. Extension of Basic Model (xCSP)

From a practical point of view some feasible sequences are more attractive than others if certain additional features are incorporated. Because of the need for integration of different features in each car, different constraints may arise. In order to regulate both workloads and usage of components over time a new set of soft constraints is introduced into the basic CSP model [15]. Taking into consideration minimum load constraints in addition to maximum load constraints, more balanced sequences are obtained which leads to a more regular production flow. This new group of capacity constraints is presented with $r_j \setminus s_j$, which means that for any sequence of s_j consecutive cars, at least r_j of them may require an option $j = 1...O$ and

$$\sum_{i=1}^{C} \sum_{k=h_j(t)}^{t} o_{ji} x_{ik} \geq min\{t, r_j\} \quad \forall j = 1...O, \forall t = 1...D, h_j(t) = max[1, t+1-s_j] \tag{5}$$

For $\forall t = 1...D$ and $\forall j = 1...O$ there is a segment [$h_j(t)$, t] of length π where $|\pi| \leq s_j$ and $h_j(t) = max[1, t+1-s_j]$ is the starting position of that segment. As noted for the basic model, extended model as well allows incorporation of the additional constraints in the assignment of options in the first positions of the sequence by considering intervals smaller than s_j [15].

Constraints can have different priorities. This means that violating different constrains does not have to be penalized with the same weights. Based on this, four measures of distance between the goals and the sequence $U = u(1)..u(D)$ where $u(t) \in \{1..C\}$ represents the class of car that occupies position $t = 1..D$ in the time frame D, can be identified [15]:

Upper bound overassignment is the positive difference between the number of times option $j = 1...O$ actually appears and the maximum number of times

it is allowed to appear in segment $[l_j(t), t] : t = 1...D$.

$$p_{jt}(U)^+ = [\sum_{k=l_j(t)}^{t} o_{j,u(k)} - min\{t, p_j\}]^+$$
$$\forall j = 1...O, \forall t = 1...D, l_j(t) = max\{1, t+1-q_j\}$$
$$p_{jt}(U)^+ \in [0, q_j - p_j) \quad (6)$$

where $[x]^+ = max\{0, x\}$ and $o_{j,u(t)}$ is the number of units of option $j = 1..C$ required by a unit of the class of car that occupies position $t = 1..D$ in the sequence $U = u(1)..u(D)$.

Upper bound underassignment is the positive difference between maximum number of times option $j = 1...O$ is allowed to appear and the number of times it actually appears in segment $[l_j(t), t] : t = 1...D$.

$$p_{jt}(U)^- = [min\{t, p_j\} - \sum_{k=l_j(t)}^{t} o_{j,u(k)}]^+$$
$$\forall j = 1...O, \forall t = 1...D, l_j(t) = max\{1, t+1-q_j\}$$
$$p_{jt}(U)^- \in [0, p_j) \quad (7)$$

Lower bound overassignment is the positive difference between the number of times option $j = 1...O$ actually appears and the minimum number of times it is allowed to appear in segment $[h_j(t), t] : t = 1...D$.

$$r_{jt}(U)^+ = [\sum_{k=h_j(t)}^{t} o_{j,u(k)} - min\{t, r_j\}]^+$$
$$\forall j = 1...O, \forall t = 1...D, h_j(t) = max\{1, t+1-s_j\}$$
$$r_{jt}(U)^+ \in [0, s_j - r_j) \quad (8)$$

Lower bound underassignment is the positive difference between minimum number of times option $j = 1...O$ is allowed to appear and the number of times it actually appears in segment $[h_j(t), t] : t = 1...D$.

$$r_{jt}(U)^- = [min\{t, r_j\} - \sum_{h=l_j(t)}^{t} o_{j,u(k)}]^+$$
$$\forall j = 1...O, \forall t = 1...D, h_j(t) = max\{1, t+1-s_j\}$$
$$r_{jt}(U)^- \in [0, r_j) \quad (9)$$

In order to penalize violation of different constraints corresponding weights denoted with P^+_{jt}, P^-_{jt}, R^+_{jt}, R^-_{jt} are assigned to each of the distance measures p^+_{jt}, p^-_{jt}, r^+_{jt}, r^-_{jt} respectively for $j = 1..O$ and $t = 1..D$ of the corresponding segment. The cost of the sequence is defined as a weighted sum of upper and lower overassignments and underassignments (10).

$$min \sum_{t=1}^{D} \sum_{j=1}^{O} (P^+_{jt} p^+_{jt} + P^-_{jt} p^-_{jt} + R^+_{jt} r^+_{jt} + R^-_{jt} r^-_{jt}) \quad (10)$$

Solution process can be formulated as a search for the minimal cost sequence formed of cars to be produced while satisfying the following constraint sets:

$$\sum_{i=1}^{C} x_{it} = 1, \forall t = 1...D \quad (11)$$

$$\sum_{t=1}^{D} x_{it} = d_i, \forall i = 1...C \quad (12)$$

$$p^-_{jt} - p^+_{jt} + \sum_{i=1}^{C} \sum_{k=l_j(t)}^{t} o_{ji} x_{ik} = min\{t, p_j\}$$
$$\forall j = 1..O; \forall t = 1..D; l_j(t) = max\{1, t+1-q_j\} \quad (13)$$

$$r^-_{jt} - r^+_{jt} + \sum_{i=1}^{C} \sum_{k=h_j(t)}^{t} o_{ji} x_{ik} = min\{t, r_j\}$$
$$\forall j = 1..O; \forall t = 1..D; h_j(t) = max\{1, t+1-s_j\} \quad (14)$$

$$x_{it} \in \{0, 1\}, \forall i = 1...C, \forall t = 1...D \quad (15)$$

$$p^+_{jt}, p^-_{jt}, r^+_{jt}, r^-_{jt} \geq 0, \forall j = 1..O; \forall t = 1..D \quad (16)$$

While (11), (12), and (15) are same as (1), (2), and (4) in the basic model, additional constraints sets like (13) and (14) measure differences between the actual and maximum allowed assignments for all options and segments and differences between the actual and minimum allowed assignments for all options and segments respectively. Overassignment and underassignment variables must be nonnegative (16).

In case that more than one sequence that satisfies all constraints is obtained, all feasible sequences are equally valid unless specified otherwise. However, although violation of constraints is undesirable, in real production systems manufacture must be carried out. Penalizing violation of different constrains provides possibility for finding sequences in accordance with the priorities in the manufacture.

III. GRASP Approach for xCSP

In order to obtain acceptably good solutions with low computational effort a GRASP (Greedy Randomized Adaptive Search Procedure) method was proposed in [15]. Providing diversified initial set of solutions for starting local search by a randomized greedy construction procedure and that way avoiding local optimality is the main reason why GRASP has been used for solving xCSP. GRASP consists of two phases: the greedy construction procedure and the improvement phase. In the following sections we describe these two phases together with our contribution to each one of them.

A. Construction Procedure

During the construction procedure a solution sequence is constructed by iteratively choosing the next car type to sequence according to a heuristic function. The maximum load capacity constraint is usually a hard constraint. While in [15], selection of candidates which have smallest number of violations of maximum load constraint is performed first and then their evaluation according to three other capacity constraints is done, in this study we treat all capacity constraints as soft constraints giving them the same priority when evaluating candidates for the position in the sequence in order to achieve more balanced sequences as initial solutions for the local improvement phase. At each step of the construction procedure, each class of car i with remaining demand is evaluated by following heuristic function:

$$\begin{aligned} g_i = \sum_{j=1}^{O} (&0{,}25 * p_{jt}(S_i)^+ + 0{,}25 * p_{jt}(S_i)^- \\ &+0{,}25 * r_{jt}(S_i)^+ + 0{,}25 * r_{jt}(S_i)^-) \\ S &= s(1)...s(t-1) \\ S_i &= S \cup i \end{aligned} \tag{17}$$

The list of best candidates called restricted candidates list (RCL) is formed out of candidates that are better than a given threshold value (α). A value-based restriction criteria is used for forming RCL since it is the most commonly used strategy [18]. In the experimental study, different α values are considered in order to explore the effectiveness of heuristic information in constructing solutions. One candidate is then chosen randomly from RCL for the position t in the sequence. The random choice of a candidate from RCL represents a probabilistic component of the GRASP metaheuristic and enables different solution at each iteration. The process of construction is repeated until the entire sequence has been built.

B. Improvement Phase

In order to improve the solutions a local search phase is carried out since it is not guaranteed that the solutions produced during the greedy phase are locally optimum.

Two classical types of local improvement [8] are considered in [15]. They are applied once in a specific order, first 2-swap then 1-insertion on each sequence build during greedy phase of GRASP.

For each position $k = 1...D$ in the sequence the key idea of the local improvement is to find the greatest interval around k that does not contain any other unit of car $s(k)$ in the sequence S obtained during the greedy phase of the algorithm. When the corresponding interval around k is found, exchange of unit of car $s(k)$ with all the units of cars included in the interval is explored. If during the exploration a sequence with better cost is obtained then k is reset on the starting position ($k = 1$) and the exchange is carried out. The stopping criteria is when all k values have been explored and none offers a sequence with better cost or when the number of evaluations of cost has reached 100 000. Insertion local improvement works analog to the exchange one. Insertion of a unit of car $s(k)$ is tested in all the positions included in the greatest interval around k that does not contain any other unit of car $s(k)$.

In this study, we consider modifications for the local improvement proposed in [15]. First we explore whether changing starting position of the sequence for the local improvement can lead to a more efficient local search when initial solutions are constructed using a static α value. The main disadvantage of the described greedy algorithm is that it uses all the good options at the beginning of the sequence, leaving the worse options for the last part of the sequence. Based on this, a modification has been explored to start local improvement from the end of the sequence and first explore neighborhoods that can offer better end parts of the solution. While starting the search from the end of the sequence different order of moves within intervals has been additionally explored. In particular, we analyze three different approaches for the improvement phase:

1 LS1 Local search starts from K = 1 (beginning of the sequence) and swap and insertion of car types within intervals obtained around each position in the sequence starts from the beginning of the interval (first bounds) [15].

2 LS2 Local search starts from K = D (or solution size, end of the sequence) and swap and insertion of car types within intervals obtained around each position in the sequence starts from the beginning of the intervals (first bounds).

3 LS3 Local search starts from K = D (or solution size, end of the sequence) and swap and insertion of car types within intervals obtained around each position in the sequence starts from the end of the intervals (second bounds).

IV. Experimental study

A. Experimental setup

In order to be able to test the validity of algorithms on more realistic problems, CSPLib benchmark library has been constructed [13]. 70 problem instances are grouped in 7 subsets of 10 instances based on utilization percentage. All these instances are feasible ones, containing 200 cars to sequence and 5 options. Out of this group of 70 instances we considered 4 instances denoted as $p60/05$, $p75/05$, $p80/03$ and $p90/08$ of 60 %, 75 %, 80 % and 90 % percentage of utilization respectively. Additionaly we considered three more instances (denoted as $p200$, $p300$ and $p400$) from group of 30 hard problems by Caroline Gagne [13]. These instances are also feasible, have 200, 300 and 400 cars respectively to sequence, 5 options and high utilization rate (over 90 %).

Because of the practical importance of the problem and decreasing effectiveness of proposed approaches with harder instances, for the experimental study we have chosen different instances according to the total number of cars to sequence (200, 300 and 400) and complexity (60 %, 75 %, 80 % and 90 %). According to [15], instances considered for the experimental study were adjusted for the xCSP considering the new group of capacity constraints $r \setminus s$ in following way: $r_j = p_j - 1$ and $s_j = q_j$ for each option $j = i..O$. The strategy used for determining α value is a static one. This means that values were fixed before starting the search. Three values of α were considered: *0.25, 0.5, 1* for measuring effectiveness of heuristic information in greedy phase of GRASP. Then, best results obtained during the construction procedure have been chosen as a set of starting solutions for the local improvement. The effectiveness of the proposed modifications in local improvement phase was measured in terms of solution quality, computational effort and robustness. For each approach we present average CPU time (in seconds) on 1000 iterations of GRASP algorithm (construction phase + local search), number of times it found the best solution (hits), as well as the cost of local optimal solution and values of four distance measures (p^+, p^-, r^+, r^-). Setting different priorities for violations of capacity constraints is important in practical terms. Based on this, three test cases with different combination of values for weights (P^+, P^-, R^+, R^-) associated with four identified measures were considered. All experiments were run on an Intel Core 2 Quad.

B. Results

The appropriate choice of the value of the RCL parameter (α) is relevant for solution quality. Results obtained during the greedy phase of GRASP for different values of parameter α are presented in Table I. As α value changes from $\alpha = 1$ (purely random) to $\alpha = 0{,}25$ (almost greedy) better solutions are obtained which means that the usage of heuristic information gives better results.

In Table II are presented results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0.99, 0.01, 0, 0). The aim is to find a solution sequence in which the appearance of option j is as close as possible to the value p_j but without exceeding it. This means that the biggest priority is given to the satisfaction of maximum load constraint p_j/q_j associated with p^+ measure. In terms of solution quality LS3 finds solutions of the best quality for all instances. Improvement is especially visible for instances $p90/08$, $p200/01$, $p300/04$, $p400/07$. For instance $p60/05$ approach LS3 finds a solution with upper bound $p^+ = 0$ unlike the two other approaches which is a goal set by combination of weights. For three out of seven instances ($p6/05$, $p75/05$, $p80/03$) LS3 finds solutions that satisfy the goal. LS2 approach showed the shortest average computation time (4/7 instances). However, LS3 is the one which most often found the best solutions, at the cost of computation times that are longer than those of LS2 (4/7 instances), but shorter with respect to those of LS1. The high number of hits (in the cases of $p60/05$, $p75/05$ and $p80/05$) showed by LS3 also illustrates the effectiveness of the approach.

TABLE I: Results for greedy phase for variation of parameter α

Instance	α	Cost	p^+	p^-	r^+	r^-
P60/05	1	351	116	678	523	87
	0.5	323	62	615	530	85
	0.25	**299**	**15**	**568**	**529**	**84**
P75/05	1	354	144	510	697	65
	0.5	316	98	459	672	35
	0.25	**268,5**	**11**	**372**	**664**	**27**
P80/03	1	365	161	552	677	70
	0.5	312	85	478	645	40
	0.25	**267**	**20**	**417**	**616**	**15**
P90/08	1	377	225	438	815	30
	0.5	324,5	135	349	799	15
	0.25	**266,5**	**29**	**243**	**789**	**5**
P200/01	1	413,5	286	372	954	42
	0.5	347,5	189	264	930	7
	0.25	**287,5**	**76**	**167**	**907**	**0**
P300/04	1	605,5	397	562	1398	65
	0.5	519	277	447	1340	12
	0.25	**445,5**	**141**	**325**	**1315**	**1**
P400/07	1	789,5	501	803	1775	79
	0.5	717	400	698	1735	35
	0.25	**590,5**	**178**	**496**	**1684**	**4**

In Table III are presented the results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0, 0, 0.01, 0.99). The aim is to find a solution sequence in which the appearance of option j is as close as possible to the value r_j but not less than it. This means that the biggest priority is given to the satisfaction of minimum load constraint r_j/s_j associated with r^- measure. For six out of seven instances r^- value is equal to zero with all three approaches. This means that solutions that satisfy goal set with combination of weights are found regardless of the starting point and the order of moves performed for local search. Improvement in terms of solution quality with approach LS3 is visible for the instance p60/05. LS2 approach showed the shortest average computation time for all seven instances. However, LS3 is the one which most often found the best solutions, at the cost of computation times that are longer than those of LS2, but shorter with respect to those of LS1 except in the case of the instances $p300/04$ and $p400/07$ when computation time with LS3 is considerably longer.

In Table IV are presented results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0.35, 0.15, 0.15, 0.35). Same first order priority is given to the satisfaction of maximum (p^+) and minimum (r^-) load constraints. On the other hand, same second order priority is given to the

TABLE II: Results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0.99, 0.01, 0, 0)

		(0.99, 0.01, 0, 0) $\alpha = 0{,}25$						
Instance	Approach	Time	Hits	Cost	p^+	p^-	r^+	r^-
P60/05	LS1	1,991	1	6,56	1	557	521	79
	LS2	1,417	1	6,53	1	554	521	76
	LS3	**1,286**	**798**	**5,54**	**0**	**554**	526	82
P75/05	LS1	2,533	1	3,63	0	363	670	35
	LS2	**1,55**	8	3,62	0	362	669	33
	LS3	1,932	**623**	**3,62**	**0**	**362**	664	28
P80/03	LS1	2,607	1	3,9	0	390	643	35
	LS2	**2,084**	5	3,9	0	390	648	40
	LS3	2,482	**554**	**3,9**	**0**	**390**	637	29
P90/08	LS1	2,712	1	14,19	12	231	786	7
	LS2	**2,476**	1	11,16	9	225	790	8
	LS3	2,77	1	**3.18**	**1**	**219**	792	12
P200/01	LS1	2,78	1	46,89	46	135	910	1
	LS2	**2,49**	1	42,83	42	125	915	0
	LS3	2,59	1	**27,9**	**27**	**117**	909	1
P300/04	LS1	5,593	1	112,78	111	289	1322	2
	LS2	4,363	1	78,73	77	250	1329	4
	LS3	**4,34**	1	**56,7**	**55**	**225**	1331	3
P400/07	LS1	7,774	1	156,17	153	470	1692	11
	LS2	6,434	1	131,07	128	435	1694	3
	LS3	**6,19**	**2**	**80,06**	**77**	**383**	1709	17

TABLE III: Results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0, 0, 0.01, 0.99)

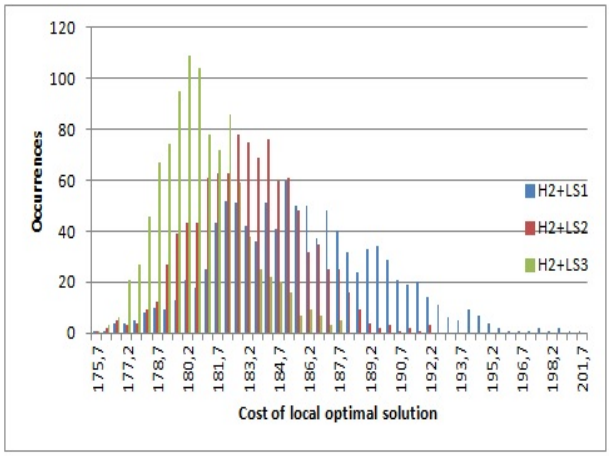
		(0, 0, 0.01, 0.99) $\alpha = 0{,}25$						
Instance	Approach	Time	Hits	Cost	p^+	p^-	r^+	r^-
P60/05	LS1	3,459	1	41,36	73	635	473	37
	LS2	**2,238**	1	42,37	63	624	475	38
	LS3	2,282	**127**	**33,3**	81	649	**459**	**29**
P75/05	LS1	3,593	2	6,2	63	441	620	0
	LS2	**2,103**	89	6,15	60	443	615	0
	LS3	2,433	**578**	**6,15**	50	433	**615**	**0**
P80/03	LS1	3,143	2	5,87	54	465	587	0
	LS2	**2,052**	18	5,86	73	485	586	0
	LS3	2,618	**332**	**5,86**	68	480	**586**	**0**
P90/08	LS1	2,832	209	7,63	90	325	763	0
	LS2	**0,87**	343	7,63	74	309	763	0
	LS3	1,029	**621**	**7,63**	68	303	**763**	**0**
P200/01	LS1	1,277	541	9	114	212	900	0
	LS2	**0,212**	659	9	112	210	900	0
	LS3	1,001	**953**	9	101	199	**900**	**0**
P300/04	LS1	2,732	994	13,14	175	359	1314	0
	LS2	**1,486**	972	13,14	167	351	1314	0
	LS3	4	**995**	**13,14**	165	349	**1314**	**0**
P400/07	LS1	5,649	887	16,77	259	580	1677	0
	LS2	**4,033**	**944**	**16,77**	267	588	**1677**	**0**
	LS3	9,09	937	16,77	257	578	1677	0

satisfaction of constraints associated with measures p^- and r^+. The aim is to find solution sequences in which in first line appearance of option j do not exceed value p_j and is not less than value r_j but in second line is close as possible to them. For all seven instances approach LS3 finds solution of best quality with respect to the two other approaches. For instances $p200/01$, $p300/04$, $p400/07$ both p^+ and p^- values are considerably lower than with other two approaches. For six out of seven instances r^- value is zero. Although LS3 gives considerably lower r^- value for instance $p60/05$, it didn't manage to find a sequence with $r^- = 0$. LS2 approach showed the shortest average computation time for five out of seven instances. However, LS3 is the one which most often found the best solutions, at the cost of computation times that are longer than those of LS2 (5/7) and sometimes longer than those of LS1 (3/7). LS3 gives shortest average time and solution of the best quality for the instances $p300/04$ and $p400/07$. The higher number of hits for three out of seven instances showed by LS3 also illustrates the effectiveness of approach. In the case of instances $p60/05$, $p200/01$, $p300/04$ and $p400/07$ number of hits (1) is the same for all three approaches but solution obtained with LS3 has considerably better quality. In the case of $p75/05$ instance with LS3 approach sequence with both upper (p^+) and lower (r^-) bound equal to zero is obtained.

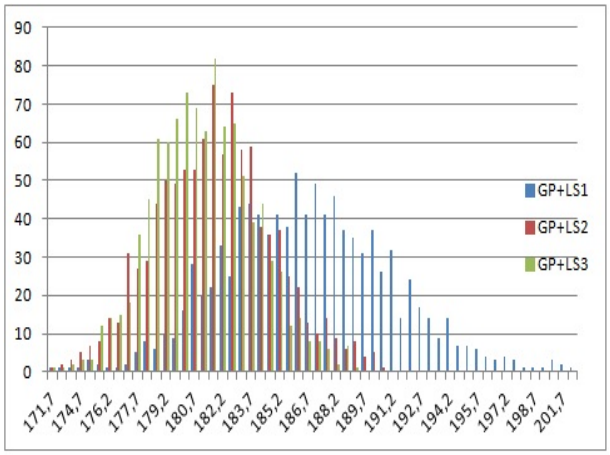
For three instances ($p60/05$, $p200/01$ and $p400/07$) with different number of cars to sequence and complexity results obtained by applying local search with combination of weights (P^+, P^-, R^+, R^-) = (0.35, 0.15, 0.15, 0.35) to each of the 1000 solutions constructed with heuristic information with $\alpha = 0,25$ are presented in Figure 1. For all three instances, in terms of solution diversity LS1 produced the most diverse solutions with respect to two other approaches but with smallest number of occurrences. While LS3 approach produced less diverse solutions with respect to two other approaches but with much bigger number of occurrences. Distance between the value of average solution and the value of the best solution found during local improvement phase is smallest for LS3 approach for all three instances.

V. Conclusions

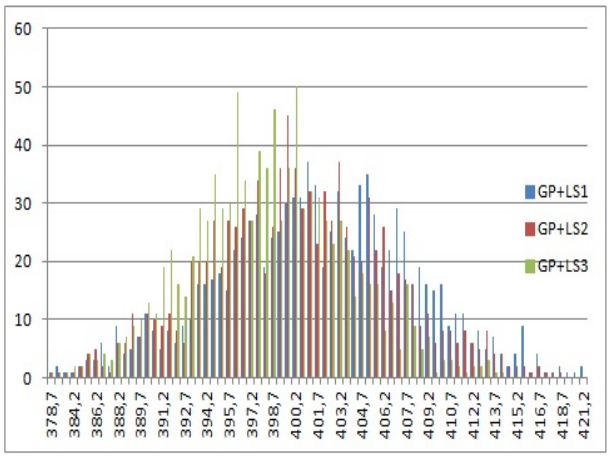
Although sometimes at cost of slightly higher computational time, LS3 approach finds best solutions in terms of solution quality with highest number of occurrences for all seven instances with respect to two other approaches. This study shows that the starting position of the sequence for the local search is important when initial solutions are constructed with greedy algorithm using a static α value. Taking into account changing the starting position of the sequence for exploring moves for the local improvement can lead to better and more efficient solutions. However, a more sophisticated metaheuristic needs to be constructed in order to find better solutions for harder instances with bigger number of cars (over 200) and higher complexity (90 % and over). Focus of future research will also be on modeling multiobjective models and appropriate methods for solving them, since minimization of work overload



(a) p60/05



(b) p200/01



(c) p400/07

Fig. 1: Distribution of local search solution values for each instance for combination of weights (P^+, P^-, R^+, R^-) = (0.35, 0.15, 0.15, 0.35)

TABLE IV: Results for the different approaches for the combination of weights (P^+, P^-, R^+, R^-) = (0.35, 0.15, 0.15, 0.35)

		(0.35, 0.15, 0.15, 0.35) $\alpha = 0{,}25$						
Instance	Approach	Time	Hits	Cost	p^+	p^-	r^+	r^-
P60/05	LS1	3,115	1	175,7	13	566	484	39
	LS2	**2,158**	1	173,7	6	560	486	42
	LS3	2,177	1	**165,2**	**2**	**560**	**469**	**29**
P75/05	LS1	2,829	1	153,2	6	374	631	1
	LS2	**2,321**	1	150,7	1	**363**	637	1
	LS3	2,556	**93**	**149,7**	**0**	369	**629**	**0**
P80/03	LS1	2,634	1	152,2	2	**396**	607	3
	LS2	**2,371**	1	150,7	1	399	601	1
	LS3	2,688	**2**	**150,2**	**1**	401	**598**	**0**
P90/08	LS1	2,696	1	155,7	12	232	778	0
	LS2	**2,372**	2	153,7	8	**223**	783	0
	LS3	2,735	**13**	**151,7**	**4**	227	**775**	**0**
P200/01	LS1	2,789	1	171,7	44	134	908	0
	LS2	**2,449**	1	170,7	76	167	907	0
	LS3	2,591	1	**164,7**	**30**	**121**	**907**	**0**
P300/04	LS1	5,334	1	281,2	113	293	1318	0
	LS2	4,319	1	263,2	141	325	**1315**	1
	LS3	**4,219**	1	**256,2**	**63**	**242**	1319	**0**
P400/07	LS1	7,331	1	378,7	158	478	1678	0
	LS2	6,37	1	365,7	128	440	1690	4
	LS3	**6,129**	1	**347,7**	**96**	**408**	**1686**	**0**

and leveling part usage in model sequencing are often desirable but conflicting objectives.

Acknowledgments

This work is supported by Ministerio de Economía y Competitividad under BES-2012-056565 (TIN2011-29824-C02-01), SOCOVIFI2 (TIN2012-38525-C02-01 and TIN2012-38525-C02-02) and PROTHIUS-III (DPI2010-16759), including EDRF funding.

References

[1] Bruce D. Parrello, Waldo C. Kabat and Larry Wos, *Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem*, Journal of Automated reasoning 2, no. 1 (1986): 1-42.

[2] Edward Tsang, *Foundations of constraint satisfaction*, Vol. 289. London: Academic press, 1993.

[3] Ian P. Gent, *Two results on car-sequencing problems*, Research Reports of the APES Group, APES-02-1998, Available from http://www. dcs. stand. ac. uk/ apes/apesreports. html (1998).

[4] Tamás Kis, *On the complexity of the car sequencing problem*, Operations Research Letters 32, no. 4 (2004): 331-335.

[5] Mehmet Dincbas, Helmut Simonis and Pascal Van Hentenryck, *Solving the Car-Sequencing Problem in Constraint Logic Programming*, In ECAI, vol. 88, pp. 290-295. 1988.

[6] Pascal Van Hentenryck, Helmut Simonis, and Mehmet Dincbas, *Constraint satisfaction using constraint logic programming*, Artificial intelligence 58, no. 1 (1992): 113-159.

[7] Marc Gravel, Caroline Gagne and Wilson L. Price, *Review and comparison of three methods for the solution of the car sequencing problem*, Journal of the Operational Research Society 56, no. 11 (2005): 1287-1295.

[8] Markus Puchta and Jens Gottlieb, *Solving car sequencing problems by local optimization*, Journal of the Operational Research Society 56, no. 11 (2005): 1287-1295.In Applica-

tions of Evolutionary Computing, pp. 132-142. Springer Berlin Heidelberg, 2002.
[9] Laurent Perron and Paul Shaw, *Combining forces to solve the car sequencing problem*, In Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 225-239. Springer Berlin Heidelberg, 2004.
[10] Jens Gottlieb, Markus Puchta, and Christine Solnon, *A study of greedy, local search, and ant colony optimization approaches for car sequencing problems*, In Applications of evolutionary computing, pp. 246-257. Springer Berlin Heidelberg, 2003.
[11] Terry Warwick, and Edward PK Tsang, *Tackling car sequencing problems using a generic genetic algorithm*, Evolutionary Computation 3, no. 3 (1995): 267-298.
[12] Jean-Charles Régin and Jean-François Puget, *A filtering algorithm for global sequencing constraints*, In Principles and Practice of Constraint Programming-CP97, pp. 32-46. Springer Berlin Heidelberg, 1997.
[13] Ian P. Gent and Toby Walsh, *CSPLib: a benchmark library for constraints*, In Principles and Practice of Constraint Programming–CP99, pp. 480-481. Springer Berlin Heidelberg, 1999.
[14] Joaquín Bautista, Jordi Pereira and Belarmino Adenso-Díaz, *A Beam Search approach for the optimization version of the Car Sequencing Problem*, Annals of Operations Research 159, no. 1 (2008): 233-244.
[15] Joaquín Bautista, Jordi Pereira and Belarmino Adenso-Díaz, *A GRASP approach for the extended car sequencing problem*, Journal of Scheduling 11, no. 1 (2008): 3-16.
[16] Dhananjay Raghavan Thiruvady, Bernd Meyer and Andreas Ernst, *Car sequencing with constraint-based ACO*, In Proceedings of the 13th annual conference on Genetic and evolutionary computation, pp. 163-170. ACM, 2011.
[17] Dhananjay Thiruvady, Andreas Ernst and Mark Wallace, *A Lagrangian-ACO matheuristic for car sequencing*, EURO Journal on Computational Optimization (2014): 1-18.
[18] Mauricio GC Resende and Celso C. Ribeiro, *Greedy randomized adaptive search procedures: Advances, hybridizations, and applications*, In Handbook of Metaheuristics, pp. 283-319. Springer US, 2010.
[19] Thomas A. Feo and Mauricio GC Resende, *Greedy randomized adaptive search procedures*, Journal of global optimization 6, no. 2 (1995): 109-133.