Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism

Alberto García-Villoria, Rafael Pastor

To cite this version:


HAL Id: hal-00545667
https://hal.archives-ouvertes.fr/hal-00545667
Submitted on 11 Dec 2010
Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism

<table>
<thead>
<tr>
<th>Journal:</th>
<th><em>International Journal of Production Research</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID:</td>
<td>TPRS-2008-IJPR-0622.R1</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Original Manuscript</td>
</tr>
<tr>
<td>Date Submitted by the Author:</td>
<td>24-Dec-2008</td>
</tr>
<tr>
<td>Complete List of Authors:</td>
<td>García-Villoria, Alberto; Institute of Industrial and Control Engineering, Department of Management Pastor, Rafael; Institute of Industrial and Control Engineering, Department of Management</td>
</tr>
<tr>
<td>Keywords:</td>
<td>SEQUENCING, META-HEURISTICS, SCHEDULING, JIT</td>
</tr>
<tr>
<td>Keywords (user):</td>
<td>response time variability, electro-like mechanism</td>
</tr>
</tbody>
</table>
Author 1 (corresponding author)
- Name: Alberto García-Villoria
- Affiliations: Institute of Industrial and Control Engineering (IOC), Department of Management, Technical University of Catalonia (UPC)
- Postal address: Av. Diagonal 647 (Edif. ETSEIB), 11th floor, 08028 Barcelona, Spain
- Telephone: +34 93 405 4010
- Fax: +34 93 401 6605
- Email: alberto.garcia-villoria@upc.edu

Author 2
- Name: Rafael Pastor Moreno
- Affiliations: Institute of Industrial and Control Engineering (IOC), Department of Management, Technical University of Catalonia (UPC)
- Postal address: Av. Diagonal 647 (Edif. ETSEIB), 11th floor, 08028 Barcelona, Spain
- Telephone: +34 93 401 1701
- Fax: +34 93 401 6605
- Email: rafael.pastor@upc.edu
Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism

The Response Time Variability Problem (RTVP) is an NP-hard combinatorial scheduling problem that has been recently formalised in the literature. The RTVP has a wide range of real-life applications such as in the automobile industry, when models to be produced on a mixed-model assembly line have to be sequenced under a just-in-time production. The RTVP occurs whenever products, clients or jobs need to be sequenced so as to minimize variability in the time between the instants at which they receive the necessary resources. In two previous studies, three metaheuristic algorithms (a multi-start, a GRASP and a PSO algorithm) were proposed to solve the RTVP. We propose solving the RTVP by means of the electromagnetism-like mechanism (EM) metaheuristic algorithm. The EM algorithm is based on an analogy with the attraction-repulsion mechanism of the electromagnetism theory, where solutions are moved according to their associated charges. In this paper we compare the proposed EM metaheuristic procedure with the three metaheuristic algorithms aforementioned and it is shown that, on average, the EM procedure improves strongly on the obtained results.

Keywords: response time variability; fair sequences; scheduling; just-in-time; metaheuristics; electromagnetism-like mechanism

1. Introduction

The Response Time Variability Problem (RTVP) is a scheduling problem that has recently been defined in the literature (Corominas et al. 2007). The RTVP occurs whenever products, clients or jobs need to be sequenced so as to minimize variability in the time between the instants at which they receive the necessary resources. Although this combinatorial optimisation problem is easy to formulate, it is very difficult to solve (it is NP-hard, Corominas et al. 2007).

The RTVP has a broad range of real-life applications. For example, it can be used to sequence regularly models in the automobile industry (Ding and He 2007), to resource allocation in computer multi-threaded systems and network servers (Waldspurger and Weihl 1995), to broadcast video and sound data frames of applications over asynchronous transfer mode networks (Dong et al. 1998), in the periodic machine maintenance problem when the distances between consecutive services of the same machine are equal (Anily et al. 1998) and in the collection of waste (Herrmann 2007).

One of the first problems in which has appeared the importance of sequencing regularly is at the sequencing on the mixed-model assembly production lines at Toyota Motor Corporation under the just-in-time (JIT) production system. One of the most important JIT objectives is to get rid of all kinds of waste and inefficiency and, according to Toyota, the main waste is due to the stocks. To reduce the stock, JIT production systems require to producing only the necessary models in the necessary quantities at the necessary time (Aigbedo 2004). To achieve this, one main goal, as Monden (1983) says, is scheduling the units to be produced to keep a constant

* An earlier version of this paper has been published on-line at http://hdl.handle.net/2117/2013
consumption rates of the components involved in the production process. Miltenburg (1989) deals with this scheduling problem considering only the demand rates of the models. He proposed four metrics to measure the regularity of a sequence based on the discrepancies, for each model, between the real production rate and the ideal one (i.e., the one that would correspond to a constant rate of production). This problem is known as the Product Rate Variation (PRV) problem (Kubiak 1993). The PRV problem has been reformulated by Kubiak and Sethi (1994) as an assignment problem and, therefore, it can be solved efficiently.

Although the sequencing on the mixed-model assembly production lines is usually considered in the literature as a PRV problem (Miltenburg 1989, Kubiak 1993, Steiner and Yeomans 1993), in our experience with practitioners of manufacturing industries we noticed that they usually refer to a good mixed-model sequence not in terms of ideal production, but in terms of having distances between the units for the same model as regular as possible. Therefore, the metric used in the RTVP reflects the way in which practitioners refer to a desirable regular sequence.

In this paper, the electromagnetism-like mechanism (EM) metaheuristic is proposed to solve the RTVP. EM is a recent population-based metaheuristic that was first proposed by Birbil and Fang (2003). It is based on an analogy with the attraction-repulsion mechanism of electromagnetism theory. Each solution is considered as a point with an electrical charge that is measured by the objective function. This charge determines the magnitude of attraction or repulsion of the other points for applying the electromagnetism equations and EM iteratively calculates the movement of the points.

The EM metaheuristic has yielded good results when it has been used to solve several combinatorial optimisation problems (Debels and Vanhoucke 2006, Debels et al. 2006, Yuan et al. 2006, Maenhout and Vanhoucke 2007, Chang et al. 2009). The EM algorithm proposed to solve the RTVP is compared with efficient procedures for solving non-small instances: the multi-start and the GRASP algorithms proposed in García et al. (2006) and the PSO algorithm called DPSOpoip-c,dyln proposed in García-Villoria and Pastor (2009).

The remainder of this paper is organized as follows. Section 2 presents a formal definition of the RTVP. Section 3 briefly exposes three metaheuristic algorithms presented in García et al. (2006) and García-Villoria and Pastor (2009). Section 4 describes the basic scheme of the EM. Section 5 proposes a procedure an EM algorithm for solving the RTVP. Section 6 provides the computational experiments and the comparison with the other metaheuristic algorithms. Finally, some conclusions and suggestions for future research are given in Section 7.

2. The Response Time Variability Problem (RTVP)

The aim of the Response Time Variability Problem (RTVP) is to minimise variability in the distances between any two consecutive units of the same model to be scheduled.

The RTVP is formulated as follows. Let \( n \) be the number of models, \( d_i \) the number of units to be scheduled of model \( i (i = 1, \ldots, n) \) and \( D \) the total number of units \( D = \sum_{i=1,n} d_i \). Let \( s \) be a solution of a RTVP instance that consists of a circular sequence of units \( (s = s_1s_2\ldots s_D) \), where \( s_j \) is the unit sequenced in position \( j \) of sequence \( s \). For all unit \( i \) in which \( d_i \geq 2 \), let \( \ell_i \) be the distance between the positions in which the units \( k+1 \) and \( k \) of the model \( i \) are found (i.e. the number of positions
between them, where the distance between two consecutive positions is considered equal to 1). Since the sequence is circular, position 1 comes immediately after position D; therefore, \( t'_{d_i} \) is the distance between the first unit of the model \( i \) in a cycle and the last unit of the same model in the preceding cycle. Let \( \bar{t}_i \) be the average distance between two consecutive units of the model \( i \) (\( \bar{t}_i = \frac{D}{d_i} \)). For all symbol \( i \) in which
\[
d_i = 1, \quad t'_i \quad \text{is equal to} \quad \bar{t}_i.
\]
The objective is to minimize the metric Response Time Variability (RTV) which is defined by the following expression:
\[
RTV = \sum_{i=1}^{m} \sum_{k=1}^{d_i} (t'_k - \bar{t}_i)^2
\]  

For example, let \( n = 3, \quad d_1 = 2, \quad d_2 = 2 \) and \( d_3 = 4 \); thus, \( D = 8, \quad \bar{t}_1 = 4, \quad \bar{t}_2 = 4 \) and \( \bar{t}_3 = 2 \). Any sequence is a feasible solution. For example, the sequence (C, A, C, B, C, B, A, C) is a solution, where
\[
RTV = (5 - 4)^2 + (3 - 4)^2 + (2 - 4)^2 + (6 - 4)^2 + (2 - 2)^2 + (2 - 2)^2 + (3 - 2)^2 + (1 - 2)^2 = 12.
\]

3. Three metaheuristic algorithms for the RTVP

Corominas et al. (2007) proposed a mixed integer linear programming (MILP) model to solve the RTVP. Corominas et al. (2006) improved the previous MILP model but the practical limit to obtain optimal solutions is 40 units to be scheduled. Thus, the use of heuristic or metaheuristic methods for solving real-life instances of the RTVP is justified. Corominas et al. (2007) proposed five heuristic algorithms. García et al. (2006) proposed six metaheuristic algorithms: a multi-start, a GRASP (Greedy Randomized Adaptive Search Procedure) and four PSO (Particle Swarm Optimisation) algorithms. Eleven variants of the PSO metaheuristic were also used to solve the RTVP in García-Villoria and Pastor (2009).

The three most effective aforementioned procedures are the multi-start and the GRASP algorithm proposed in García et al. (2006) and the PSO algorithm called \( DPSo_{opoi\_c, dyn} \) proposed in García-Villoria and Pastor (2009). Next, the algorithms are briefly explained (for more details of the three algorithm procedures, see García et al. 2006 and García-Villoria and Pastor 2009).

3.1. The Multi-start algorithm

The multi-start method is based on generating initial random solutions and on improving each of them to find a local optimum, which is usually done by means of a local search procedure (Marti 2003). Random solutions are generated as follows. For each position, a model to be sequenced is randomly chosen. The probability of each model is equal to the number of units of this model that remain to be sequenced divided by the total number of units that remain to be sequenced. The local search procedure used is applied as follows. A local search is performed iteratively in a neighbourhood that is generated by interchanging each pair of two consecutive units of the sequence that represents the current solution; the best solution in the neighbourhood is chosen; the optimisation ends when no neighbouring solution is better than the current solution.
3.2. The GRASP algorithm

GRASP, designed by Feo and Resende (1989), can be considered to be a variant of the multi-start method in which the initial solutions are obtained using directed randomness. The solutions are generated by means of a greedy strategy in which random steps are added and the choice of elements to be included in the solution is adaptive. The random step in the GRASP proposed by García et al. (2006) consists of selecting the next model to be added to the solution; the probability of each candidate model is proportional to the value of its Webster index, which is based on the parametric method of apportionment with parameter $\delta = \frac{1}{2}$ (Balinski and Young 1982). The Webster index for model $i$ ($i = 1, \ldots, n$) is evaluated as $\frac{d_i}{(x_i + \delta)}$, where $x_i$ is the number of units of model $i$ in the sequence of length $t = 0, \ldots, D$ (assume $x_0 = 0$). The local search procedure applied to the initial solutions is the same local search as in the multi-start method.

3.3. The PSO algorithm

PSO is a population-based metaheuristic designed by Kennedy and Eberhart (1995), which is based on an analogy of the social behaviour of flocks of birds when they search for food. The population or swarm is composed of particles (birds), which have an $m$-dimensional real point (which represents a feasible solution) and a velocity (the movement of the point in a $m$-dimensional real space). The velocity of a particle is typically a combination of three kinds of velocities: 1) inertia velocity; 2) velocity to the best point found by the particle; and 3) velocity to the best point found by the swarm. These components of the particles are iteratively modified by the PSO algorithm as it looks for an optimal solution.

In the DPSOpoi-c$_p$dyn algorithm (García-Villoria and Pastor 2009), random modifications to the points of the particles are introduced. The frequency of the modifications changes dynamically according to the homogeneity of the swarm. The aim is to prevent premature convergence and to enable the PSO algorithm to escape a local optimum. Although the PSO metaheuristic was originally designed for working in a $m$-dimensional real space, DPSOpoi-c$_p$dyn is adapted to work with a sequence that represents the solution. In this adaptation, a point is now the sequence of units of the models that represents a solution, and the velocity is an ordered list of transformations that must be applied to the particle so that it changes from its current point to another point; each transformation consists of a pair of positions of the point (sequence) that are to be swapped. The velocity to the best point found by the particle is the list of transformations needed to obtain the best particle point from the current position; the same applies for the velocity to the best point found by the swarm.

4. The EM metaheuristic

The electromagnetism-like mechanism (EM) metaheuristic is a new population-based metaheuristic created by Birbil and Fang (2003). The EM metaheuristic has been applied successfully to the following problems: the project scheduling problem (Debels and Vanhoucke 2006; Debels et al. 2006), neural network training (Wu et al. 2004), the permutation flowshop scheduling problem (Yuan et al. 2006), the nurse scheduling...
problem (Maenhout and Vanhoucke 2007), the single machine scheduling problem (Chang et al. 2009) and multi-objective optimisation problems (Tsou and Kao 2008). On the other hand, in the Birbil’s PhD thesis (Birbil 2002), the EM metaheuristic is compared with other methods and shown to have substantial performance.

The EM metaheuristic operates basically as follows. EM starts with an initial population of solutions that will be attracted to the deep valleys and repulsed from the steep hills (if we wish to minimise the value of the solutions). Each solution can be thought of as a particle charged according to its objective function value. Then, an analogy of the attraction-repulsion mechanism of the electromagnetism theory can be applied. Moreover, some solutions are improved by a local search.

Next, we present the framework of the EM metaheuristic; for further details, see Birbil and Fang (2003). This algorithm works with a special class of optimisation problems with bounded variables in the following form:

\[
\min \left(\max\right) f(x) \\
\text{subject to } x \in \mathbb{R}^m | l_j \leq x_j \leq u_j, j = 1, \ldots, m
\]

where \( f \) is the function that evaluates a point (which represents a solution), \( m \) is the dimension of the problem (in the case of the RTVP, \( m \) would be equal to \( D \), which is the total number of units) and \( x_j \) is the coordinate of the \( j \)th dimension, which is lower bounded by \( l_j \) and upper bounded by \( u_j \).

The EM metaheuristic is divided into four phases (which are explained in Subsections 4.1 to 4.4): 1) the initialization of the population of the points; 2) the application of the local search; 3) the calculation of the total force vector; and 4) the movement according to the total force. The pseudocode of the metaheuristic is shown in Figure 1.

[Insert Figure 1 here]

4.1. Initial population

The metaheuristic starts generating randomly the initial population, which consists of \( p \) points of the feasible domain. Each coordinate of each point is uniformly distributed between their upper and lower bounds.

4.2. Local search

The local search procedure provides the EM algorithm with a good balance between the exploration and exploitation of the feasible region. Birbil and Fang (2003) propose two approaches according to the points to which the local search can be applied: local search applied to all points and local search applied only to the current best point.

Local search applied to all points promotes a more meticulous examination of the region around the points. However local search applied only to the best point usually gives as good results and less time is spent on the local search.

In both cases, a simple local search is recommended rather than a powerful one because it is enough for a good convergence (Birbil and Fang 2003). The local search is not applied until a local optimal point is reached; the local search stops when a number of iterations (let it be called \( \text{lsiter} \)) is executed.
4.3. **Calculation of the total force vector**

The charge of each point \( x \) belonging to the population \( P \) (let it be called \( q_x \)), which determines the intensity of attraction or repulsion of the point, changes at each iteration of the EM metaheuristic. The charge is first evaluated as follows:

\[
q_x = \exp \left( -m \sum_{y \in P} \frac{f(x) - f(x_{best})}{f(y) - f(x_{best})} \right)
\]  

(2)

Note that, unlike electrical charges, no signs are associated with the charges. The direction of a particular force between two points is determined when their objective values have been compared. The total force for each point belonging to the population \( P \) (let it be called \( F_x \)) is evaluated as follows:

\[
F_x = \sum_{y \neq x, y \in P} \begin{cases} 
q_x q_y \frac{(y - x) \cdot (y - x)}{\|y - x\|^2} & \text{if } f(y) < f(x) \quad \text{(Attraction)} \\
q_x q_y \frac{(x - y) \cdot (x - y)}{\|x - y\|^2} & \text{if } f(y) \geq f(x) \quad \text{(Repulsion)}
\end{cases}
\]  

(3)

where \( \|y - x\| \) is the euclidean distance between the two points.

4.4. **Movement according to the total force**

Each point \( x \) belonging to the population \( P \) is moved according to the next equation:

\[
x = x + \lambda \frac{F_x}{\|F_x\|}
\]  

(4)

where \( \lambda \) denotes a random number uniformly distributed between 0 and 1 and \( \|F_x\| \) is the norm of the force vector. The parameter \( \lambda \) is used to ensure that the points have a nonzero probability of moving to the unvisited regions in this direction. Furthermore, the force applied to each point is normalized, so the feasibility is maintained (i.e., each coordinate of each point will be between \( l_j \) and \( u_j \)).

5. The EM algorithm for the RTVP

The objective function and the equations of the EM metaheuristic work with points of a region of the \( m \)-dimensional real space. Others procedures such as PSO algorithms or other optimisation algorithms of real variables are also designed for working in an \( m \)-dimensional real space. However, a solution of many combinatorial optimisation problems is usually represented as an ordered sequence of integer numbers (as in the RTVP), so these metaheuristics (EM, PSO and others) are incompatible with this
representation of the solution as an ordered sequence of integer numbers. There are two ways of applying algorithms of this kind to the RTVP: to adapt the algorithm to work with a sequence of integer numbers or to adapt the representation of the solution as an \( m \)-dimensional real point.

To adapt the PSO algorithm to a sequence of integer numbers for the RTVP is done in García-Villoria and Pastor (2009), as explained in Section 2. As would happen in the EM algorithm, this way involves redefining several mathematical operators used by the algorithm. For example, the difference between two points \((y - x)\) and \((x - y)\) in Equation 3 would now be the difference between two sequences of integer numbers and this new different operator should be defined.

On the other hand, a sequence of integer numbers can be represented by an \( m \)-dimensional real point using random key representation (RK) (Bean 1994). The main advantage of using RK is that each solution corresponds to a real point, so that geometric operations (for example, the evaluation of a point charge (Equation 2)) can be performed on its components. Since geometric operations are the cornerstone of several metaheuristics (such as EM), RK allows a straightforward application of this type of metaheuristics to solve combinatorial optimisation problems. Although there is empirical evidence that Genetic Algorithms that applies RK may obtain slight worst results that Genetic Algorithms adapted to the combinatorial problem (Bean 1994), in other metaheuristics there is no conclusion about which approach is better. In the case of EM algorithms, to the best of our knowledge, most of the papers in which an EM algorithm is proposed to solve a combinatorial optimisation problem RK is used (Debels and Vanhoucke 2006; Debels et al. 2006; Yuan et al. 2006; Chang et al. 2009) and only one paper adapts EM (Maenhout and Banhoucke, 2007). Because RK have been effectively applied to the EM metaheuristic to solve several combinatorial problems, in this paper RK is also used in the EM algorithm for solving the RTVP.

Random key representation for the RTVP is explained in Subsection 4.1. How the initial population is generated is described in Subsection 4.2. The local search used in our EM algorithm is explained in Subsection 4.3. The calculations of the total force vectors and the movements according to the total force are directly implemented according to Equations (2), (3) and (4). Finally, Subsection 4.4 explains the fine-tuning of the parameter values of the EM algorithm: the size of the initial population \((p)\) and the maximum number of iterations of the local search procedure \((\text{lsiter})\).

### 5.1. Random key representation

Random key representation (Bean 1994) consists of an \( m \)-length sequence of different real numbers called keys. Let the key sequence be \( r = r_1, \ldots, r_m \), where \( r_j \) is the key of the position \( j \). In the context of the proposed EM algorithm, the key sequence has \( D \) (number of units to be sequenced with \( d_i \) units of model \( i \)) keys. As the EM metaheuristic works with bounded variables, the values of the keys are bounded between 0 and 1.

Given a key sequence \( r \), the solution \( s \) (sequence of models) that is represented by \( r \) is as follows. First, for each position \( j = 1,\ldots,D \) of \( r \), key \( r_j \) is associated with a model. The association is done in a way that, for each model \( i \), there are \( d_i \) consecutive keys associated with model \( i \). For each key sequence, the association for key \( r_j \) will be always done with the same model, i.e., \( i \), for example, key \( r_1 \) is associated with model A in every key sequence \( r \), \( r_j \) will be associated with this model. Next, a new key sequence, \( r' \), is obtained by putting \( r \) (and therefore their associated models) in descending order.
according to the values of the keys. Then, for each position \( j = 1,\ldots,D \), model \( s_j \) is the model associated with key \( r'_j \), i.e., the model sequenced in position \( j \) is the model associated with key \( r_j \) that is in the position \( j \) in the key sequence \( r' \).

For example, let a RTVP instance be \( n = 3, \ d_a = 2, \ d_b = 2 \) and \( d_c = 4 \). Given the key sequence \( r = (0.12, 0.26, 0.67, 0.08, 0.14, 0.45, 0.87, 0.62) \), each key \( r_j \) \((j = 1,\ldots,8)\) is associated with a model as follows:

<table>
<thead>
<tr>
<th>models</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
<th>C</th>
<th>C</th>
<th>C</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>keys</td>
<td>0.12</td>
<td>0.26</td>
<td>0.67</td>
<td>0.08</td>
<td>0.14</td>
<td>0.45</td>
<td>0.87</td>
<td>0.62</td>
</tr>
</tbody>
</table>

So, the descending ordered key sequence is \( r' = (0.87, 0.67, 0.62, 0.45, 0.26, 0.14, 0.12, 0.08) \) and, therefore, the solution represented is \((C, B, C, C, A, C, A, B)\).

5.2. Initial population

The initial population of points consists of \( p \) solutions generated randomly. As has been introduced previously, each solution is represented by a key sequence where each key value is bounded between 0 and 1. To get a solution, we generate a random value uniformly distributed in [0,1] for each key.

5.3. Local search

The local search procedure used in the EM algorithm is as follows. A local search is performed iteratively in a neighbourhood that is generated by interchanging two units of different consecutive and non-consecutive models; the first solution found in the neighbourhood that is better than the current solution is selected; the optimisation ends when the maximum number of iterations is reached or no neighbouring solution is better than the current solution.

Local search applied to all points and local search applied only to the best point were tested by an initial experiment. To apply the local search only to the best point provided much better solutions for the RTVP, so the local search applied only to the best point is used in the EM algorithm.

5.4. Fine-tuning of the EM parameters

Fine-tuning the parameters of a metaheuristic algorithm is almost always a difficult task. Although parameter values are extremely important because the results of the metaheuristic for each problem are highly sensitive to them, the selection of parameter values is commonly justified in one of the following ways (Eiben et al. 1999, Adenso-Díaz and Laguna 2006): 1) “by hand”, on the basis of a small number of experiments that are not specifically referenced; 2) using the general values recommended for a wide range of problems; 3) using the values reported to be effective in other similar problems; or 4) choosing values without any explanation.

Adenso-Díaz and Laguna (2006) proposed a new technique called CALIBRA for fine-tuning the parameters of heuristic and metaheuristic algorithms. CALIBRA is
based on Taguchi’s fractional factorial experimental designs coupled with a local search procedure. The local search is applied to promising regions of the parameter values and the promising regions are found by means of Taguchi’s experimental designs. One assumption of Taguchi’s experimental designs is the linear interdependence between the parameters, whereas the interdependence is usually non-linear (Adenso-Díaz and Laguna 2006). CALIBRA uses the analysis of the factorial experiment results only as a guideline to narrow the search and to initiate the next round of experiments. Because the search focuses on narrower ranges for each parameter value, the linear assumption becomes less restrictive and the predicted optimal values approach to the true optimal values.

CALIBRA was chosen for fine-tuning the EM algorithm parameters. A set of 60 representative training instances, which were generated as explained in Section 5, was used. The values obtained for the size of the population ($p$) and the maximum number of iterations of the local search ($l_{siter}$) are shown in Table 1.

The GRASP and PSO algorithms (the multi-start algorithm does not have parameters) were also fine-tuned using CALIBRA and the same 60 training instances. GRASP only has one parameter, which is the size of the candidate list ($CL$). $DPSO_{poi-c_p dyn}$ has five parameters: the size of the population ($p$), the coefficient that weights the inertia velocity ($\omega$), the coefficient that weights the velocity to the best particle point ($c_1$), the coefficient that weights the velocity to the best swarm point ($c_2$) and the factor of the degree of the random modifications introduced ($K$). The parameter values obtained are shown in Table 1.

6. Computational experiment

The computational experiment for the EM algorithm is carried out for the same instances and conditions used in García et al. (2006) and in García-Villoria and Pastor (2009). That is, the algorithms ran 740 instances, which were grouped into four classes (from CAT1 to CAT4 with 185 instances in each class) according to their size. The instances were generated using the random values of $D$ (number of units) and $n$ (number of models) shown in Table 2. For all instances and for each model $i = 1, \ldots, n$, a random value of $d_i$ (number of units of model $i$) is between 1 and $\left\lfloor \frac{D - n + 1}{2.5} \right\rfloor$ such that $\sum_{i=1,n} d_i = D$. All algorithms were coded in Java and the computational experiment was carried out using a 3.4 GHz Pentium IV with 512 Mb of RAM.

For each instance, the four algorithms were run for 50 seconds. Table 3 shows the averages of the RTV values to be minimized for the global of 740 instances and for each class of instances (CAT1 to CAT4).

For the global of all instances, the EM algorithm is 18.99% better than $DPSO_{poi-c_p dyn}$, 73.55% better than the GRASP algorithm and 82.48% better than the multi-start algorithm. Observing the results in Table 3 by class, we can see that a simple algorithm

[Insert Table 1 here]

[Insert Table 2 here]

[Insert Table 3 here]
such as the multi-start algorithm obtains the best averages for small and medium instances (CAT1, CAT2 and CAT3) but a very poor average for large instances (CAT4).

On the other hand, DPSOpoi-c,dyn works well for small and large instances (CAT1, CAT2 and CAT4) but obtains bad results for medium instances (CAT3). Finally, the EM algorithm works fine for small and medium instances; and for large instances, which are the most difficult to solve, it obtains the best results.

To complete the analysis of the results, their dispersion is observed. A measure of the dispersion (let it be called $\sigma$) of the RTV values obtained by each algorithm $mh = \{ \text{EM, multi-start, GRASP, DPSOpoi-c,dyn} \}$ for a given instance, $ins$, is defined as follows:

$$\sigma(mh, ins) = \left( \frac{\text{RTV}_{ins}^{(mh)} - \text{RTV}_{ins}^{(best)}}{\text{RTV}_{ins}^{(best)}} \right)^2$$

(5)

where $\text{RTV}_{ins}^{(mh)}$ is the RTV value of the solution obtained with the algorithm $mh$ for the instance $ins$, and $\text{RTV}_{ins}^{(best)}$ is, for the instance $ins$, the best RTV value of the solutions obtained with the four algorithms. Table 4 shows the average $\sigma$ dispersion for the global of 740 instances and for each class of instances.

For the global of all instances, the EM procedure has the second least average $\sigma$ dispersion: 87.24% better than the GRASP algorithm and 79.33% better than the multi-start algorithm, but the DPSOpoi-c,dyn $\sigma$ dispersion is better than the EM procedure $\sigma$ dispersion. Observing the results in Table 4 by class, we see that the EM, multi-start and PSO algorithms have a very stable performance for small instances (CAT1 and CAT2). For medium instances (CAT3), only the EM and multi-start algorithms shows a very stable performance. Finally, no algorithm has a very stable performance for the largest instances (CAT4). This may occur because 50 computing seconds are not be enough time for the algorithms to converge for the CAT4 instances. Table 5 and Table 6 show the averages of the RTV values and the $\sigma$ dispersion, respectively, for all instances and for each class of instance (CAT1 to CAT4) obtained with the four algorithms when they are run for 1,000 seconds.

When a computing time of 1,000 seconds is used—which seems to be long enough for all algorithms to converge (see Figure 2)—the EM algorithm is clearly the best algorithm: it is 76.04%, 77.91% and 78.52% better than the multi-start, GRASP and PSO algorithm, respectively. Moreover, when the EM algorithm has converged, it has a very stable performance for all type of instances (CAT1 to CAT4). That is, when the best solution is not obtained with the EM algorithm, the EM solution is always very close to the best solution.
7. Conclusions and future research

The EM metaheuristic is a population-based metaheuristic for optimisation recently proposed by Birbil and Fang (2003). The method uses an attraction-repulsion mechanism to move the points of the population towards the optimality. In this paper, an EM algorithm is presented for solving the Response Time Variability Problem (RTVP), which has been recently appeared in the literature.

This scheduling problem arises in a variety of real-life environments including mixed-model assembly lines, multi-threaded computer systems, periodic machine maintenance, and waste collection. The aim of the RTVP is to minimize the variability in the distances between any two consecutive units of the same model. Since the RTVP is an NP-hard problem, heuristic and metaheuristic methods are needed to solve real-life instances. García et al. (2006) and Garcia-Villoria and Pastor (2009) have proposed a multi-start, a GRASP and several PSO algorithms for solving the RTVP. A computational experiment was done and the results obtained with the EM algorithm are better than the results of the aforementioned algorithms. Moreover, the EM algorithm has a very stable performance when it has converged.

There are two approaches for applying a metaheuristic that works in a real space for solving combinatorial optimisation problems: to adapt the algorithm for working with a sequence of integer numbers or to adapt the representation of the solution as a real point with a random key representation. One of the best referenced procedures is a PSO algorithm (DPSOpoi-c,dyn) that follows the first approach; on the other hand, the proposed EM algorithm follows the second approach. We propose as a future research to develop a version of DPSOpoi-c,dyn following the second approach, and to develop a version of the EM algorithm following the first approach. The objective is to obtain better results for the RTVP.

Acknowledgements

This paper was supported by the Spanish Ministry of Education and Science under project DPI2007-61905 and co-funded by the European Regional Development Fund (ERDF). The authors wish to express their gratitude to the anonymous reviewers for their valuable comments and suggestions, which have improved the quality of this paper.

References

Balinski, M.L. and Young, H.P., 1982. Fair Representation: meeting the ideal of one man, one vote. *Yale University Press*. New Haven CT.


http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk


Table 1. Parameter values obtained with CALIBRA

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>GRASP</th>
<th>DPSOpoi-c, dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>p =</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL size</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>biter</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p =</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω =</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c1 =</td>
<td>0.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c2 =</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K =</td>
<td>8.70</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Uniform distribution for the $D$ and $n$ values of the test instances

<table>
<thead>
<tr>
<th></th>
<th>CAT1</th>
<th>CAT2</th>
<th>CAT3</th>
<th>CAT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>U(25, 50)</td>
<td>U(50, 100)</td>
<td>U(100, 200)</td>
<td>U(200, 500)</td>
</tr>
<tr>
<td>$n$</td>
<td>U(3, 15)</td>
<td>U(3, 30)</td>
<td>U(3, 65)</td>
<td>U(3, 150)</td>
</tr>
</tbody>
</table>

Table 3. Averages of the RTV values for 50 seconds

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>Multi-start</th>
<th>GRASP</th>
<th>DPSOpoi-c, dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>3,747.05</td>
<td>21,390.40</td>
<td>14,168.83</td>
<td>4,625.54</td>
</tr>
<tr>
<td>CAT1</td>
<td>19.14</td>
<td>12.08</td>
<td>15.47</td>
<td>16.42</td>
</tr>
<tr>
<td>CAT2</td>
<td>54.54</td>
<td>44.36</td>
<td>88.48</td>
<td>51.34</td>
</tr>
<tr>
<td>CAT3</td>
<td>260.79</td>
<td>226.90</td>
<td>510.44</td>
<td>610.34</td>
</tr>
<tr>
<td>CAT4</td>
<td>14,653.72</td>
<td>85,278.25</td>
<td>56,060.92</td>
<td>17,824.04</td>
</tr>
</tbody>
</table>

Table 4. Average σ dispersion regarding the best solution found for 50 seconds

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>Multi-start</th>
<th>GRASP</th>
<th>DPSOpoi-c, dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>10.45</td>
<td>50.56</td>
<td>81.88</td>
<td>4.79</td>
</tr>
<tr>
<td>CAT1</td>
<td>1.62</td>
<td>0.04</td>
<td>0.65</td>
<td>0.76</td>
</tr>
<tr>
<td>CAT2</td>
<td>0.49</td>
<td>0.09</td>
<td>4.38</td>
<td>0.34</td>
</tr>
<tr>
<td>CAT3</td>
<td>0.24</td>
<td>0.13</td>
<td>7.28</td>
<td>5.29</td>
</tr>
<tr>
<td>CAT4</td>
<td>39.45</td>
<td>201.96</td>
<td>315.20</td>
<td>12.78</td>
</tr>
</tbody>
</table>

Table 5. Averages of the RTV values for 1,000 seconds

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>Multi-start</th>
<th>GRASP</th>
<th>DPSOpoi-c, dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>330.29</td>
<td>1,378.58</td>
<td>1,495.12</td>
<td>1,537.34</td>
</tr>
<tr>
<td>CAT1</td>
<td>18.64</td>
<td>10.93</td>
<td>13.59</td>
<td>14.35</td>
</tr>
<tr>
<td>CAT2</td>
<td>52.97</td>
<td>35.48</td>
<td>75.08</td>
<td>46.55</td>
</tr>
<tr>
<td>CAT3</td>
<td>157.20</td>
<td>160.67</td>
<td>428.86</td>
<td>143.96</td>
</tr>
<tr>
<td>CAT4</td>
<td>1,092.36</td>
<td>5,307.25</td>
<td>5,462.95</td>
<td>5,944.51</td>
</tr>
</tbody>
</table>

Table 6. Average σ dispersion regarding the best solution found for 1,000 seconds

<table>
<thead>
<tr>
<th></th>
<th>EM</th>
<th>Multi-start</th>
<th>GRASP</th>
<th>DPSOpoi-c, dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>0.79</td>
<td>4.80</td>
<td>14.47</td>
<td>7.61</td>
</tr>
<tr>
<td>CAT1</td>
<td>1.84</td>
<td>0.04</td>
<td>0.48</td>
<td>0.52</td>
</tr>
<tr>
<td>CAT2</td>
<td>1.03</td>
<td>0.04</td>
<td>5.29</td>
<td>0.53</td>
</tr>
<tr>
<td>CAT3</td>
<td>0.21</td>
<td>0.12</td>
<td>10.98</td>
<td>0.07</td>
</tr>
<tr>
<td>CAT4</td>
<td>0.08</td>
<td>18.98</td>
<td>41.06</td>
<td>29.27</td>
</tr>
</tbody>
</table>
Figure 1. Pseudocode of the EM metaheuristic.

1: P = initial population
2: while the stopping criteria is not reached do
4: $x^{\text{best}} = \text{best point of P}$
3: Local search
4: for each point $x$ do: $F_x = \text{total force vector}(x, P)$
5: for each point $x$ do: Move($x, F_x$)
6: end while

Figure 2. Average of the RTV values obtained during the execution time