UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Solving the response time variability problem by means of the electromagnetism-like mechanism

Alberto García-Villoria; Rafael Pastor

EOLI: Enginyeria d'Organització i Logística Industrial

IOC-DT-P-2008-03

Març 2008

**Institut d'Organització i Control
de Sistemes Industrials**

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

Alberto GARCÍA-VILLORIA^{*} and Rafael PASTOR
Institute of Industrial and Control Engineering
Technical University of Catalonia (UPC)
{alberto.garcia-villoria / rafael.pastor}@upc.edu

Abstract. The Response Time Variability Problem (RTVP) is an NP-hard combinatorial scheduling problem that has recently appeared 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. 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. The field of Artificial Intelligence has provided us with efficient tools such as metaheuristic techniques for solving complex combinatorial scheduling problems. In a previous study, three metaheuristic algorithms (a multi-start, a GRAP and a PSO algorithm) were proposed to solve the RTVP. These three metaheuristic algorithms have been most efficient, until now, in solving non-small instances of 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 previously mentioned and show that, on average, the EM procedure improves on the obtained results.

Keywords: response time variability, fair sequences, scheduling, metaheuristics, electromagnetism-like mechanism, artificial intelligence

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 optimization 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 regularly sequence models in the automobile industry (Monden, 1983), to broadcast video and sound data frames of applications over asynchronous transfer mode networks as constantly as possible (Dong et al., 1998) and in the periodic machine maintenance problem when the distances between consecutive services of the same machine are equal (Anily et al., 1998).

The RTVP was first time presented by Corominas et al. (2007). To solve it, they proposed a mixed integer lineal programming (MILP) model and five heuristic algorithms. In Corominas et al. (2006), it is proposed another MILP model, which

[†] Sponsored by the Spanish Ministry of Education and Science's project DPI2004-03472; co-funded by the FEDER.

^{*} Corresponding author: Alberto García, IOC – Institute of Industrial and Control Engineering, Av. Diagonal 647 (Edif. ETSEIB), 11th floor, 08028 Barcelona, Spain; e-mail: alberto.garcia-villoria@upc.edu

improved on the one previously mentioned. Finally, García et al. (2006) proposed, three metaheuristic algorithms to solve the problem.

One of the first problems in which seems to have appeared the importance of sequencing *regularly* is on the mixed-model assembly production lines at the Toyota Motor Corporation under the just-in-time (JIT) production system. This problem has a planning stage and a scheduling stage. In the planning stage, the number of units of each type to be sequenced in the planning horizon is determined. Subsequently, or better still, if possible, scheduling is decided simultaneously. In the Toyota system, scheduling is particularly important because it takes into account the production smooth by means of regular sequences.

Monden (1983) explains the following example. Toyota has to make 20,000 Corona units in a month with 20 working days. There are four main types of Coronas: A, B, C and D. This month, the demand for these models is 8,000, 6,000, 4,000 and 2,000 units, respectively. Thus, 1,000 Corona units – 400, 300, 200 and 100 of type A, B, C and D, respectively – must be produced daily. According to Monden, the main objective for the scheduler is to obtain a regular sequence. For example, the sequence A, B, A, C, B, A, D, A, B, C, etc. seems intuitively more regular than the sequence A, A, A, A, B, B, B, C, C, D, etc.

To measure the regularity of a sequence, different metrics can be used. The metric used in the RTVP, according to our experience with practitioners in manufacturing industries, is a reflection of the way in which practitioners refers to a desirable regular sequence.

As the RTVP is a relatively new problem, in this paper we focus our efforts on efficiently solving the RTVP prior to successfully integrating it in planning and scheduling. Artificial Intelligence (AI) techniques are being successfully used to solve a whole range of intractable problems and the metaheuristic framework is one of the most widely used to solve hard combinatorial problems. In this paper, the electromagnetism-like mechanism (EM) algorithm is proposed to solve the RTVP. The EM algorithm is a recent population-based metaheuristic algorithm 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 the EM algorithm iteratively calculates the movement of the points.

The EM algorithm has yielded good results when it has been used to solve several combinatorial optimization problems (Debels and Vanhoucke, 2004 and Yuan et al., 2006). The proposed EM procedure to solve the RTVP is compared with the more efficient procedures for solving non-small instances published until now based on Artificial Intelligence techniques: three metaheuristic procedures presented by García et al. (2006), which are a multi-start, a GRASP (Greedy Randomized Adaptive Search Procedure) and a PSO (Particle Swarm Optimization) algorithm. On average, the EM procedure improves on previous results.

The rest of this paper is organized as follows. Section 2 presents a formal definition of the RTVP and briefly exposes the three metaheuristic procedures presented by García et

al. (2006); Section 3 describes the basic scheme of the EM; Section 4 proposes a procedure based on the EM metaheuristic algorithm for solving the RTVP; Section 5 provides the computational experiments and the comparison with the other metaheuristics; finally, some conclusions and suggestions for a future integration of the planning and the scheduling are given in Section 6.

2. The Response Time Variability Problem (RTVP)

The importance of a regular sequence can be observed in the mixed-model assembly production lines at the Toyota Motor Corporation under the just-in-time (JIT) production system. One of the most important JIT objectives is to avoid any kind of waste or inefficiency. According to Toyota, stock is the main source of waste. To reduce stock, JIT production systems must only produce the necessary product types in the necessary quantities at the necessary time. As Monden (1983) pointed out, one of the main goals that must be attained in order to achieve this is to schedule the units to be produced in such a way that constant consumption rates of the components involved in the production process are maintained. Miltenburg (1989) dealt with this scheduling problem and he assumed that product types require approximately the same number and mix of parts. Thus, he only considers the rate of demand for the product types. Miltenburg proposes four objective functions based on the regularity of scheduling the product types so that the proportion of product type i produced over each time period to the total production is as close to its ideal production as possible. That is, if the number of product types is p ($i = 1, \dots, p$) and the units of product type i to be produced is d_i , then

the total number of units to be produced (D) is equal to $\sum_{i=1}^p d_i$ (time periods $k = 1, \dots, D$)

and the ideal production of product type i at the period time k is $\frac{d_i}{D} k$. This problem is

known as the Product Rate Variation (PRV) problem (Kubiak, 1993). Kubiak and Sethi (1991, 1994) reformulated the PRV problem as an assignment problem and, therefore, it can be solved with an algorithm whose complexity is polynomial in D .

The aim of the Response Time Variability Problem (RTVP) is to minimize the variability in the distances between any two consecutive units of the same type of product. Our experience with practitioners in manufacturing industries showed that they usually refer to a good mixed-model sequence as one in which the distances between the units for the same product type are as regular as possible, rather than in terms of ideal production, as is usual in the literature (Miltenburg, 1989; Kubiak, 1993; Steiner and Yeomans, 1993). Thus, the RTVP is a more realistic problem.

This problem does not only occur in the manufacturing industry, but also in computer multi-threaded systems and network servers (Waldspurger and Wehl, 1995; Dong et al., 1998). For example, the data to be sent by an asynchronous transfer mode network is divided into cells of a fixed size. There are voice and/or video applications whose data cells must be regularly sequenced in the stream.

Other contexts in which the RTVP appears are the periodic machine maintenance problem (Anily et al., 1998) and the scheduling of waste collection (Herrmann, 2007).

Hermann came up with the RTVP while working with a healthcare facility that needed to schedule the collection of waste from waste collection rooms throughout the building. Based on data about how often a waste collector had to visit each room and in view of the fact that different rooms require a different number of visits per shift, the facility manager wanted these visits to occur as regularly as possible so that excessive waste would not collect in any room. For instance, if a room needed four visits per eight-hour shift, it would ideally be visited every two hours.

These real-life problems are usually considered as distance-constrained scheduling problems (Han et al., 1996; Dong et al., 1998). Although the main objective of the distance-constrained problem and the RTVP is to find as regular a sequence as possible, the advantage of the RTVP is that it will always come up with a feasible solution, contrary to the distance-constrained problem.

The RTVP is formulated as follows. Let p be the number of product types, d_i the number of units of the product of type i ($i = 1, \dots, p$) and D the total number of units

($D = \sum_{i=1}^p d_i$). Let s be a solution of an instance in the RTVP that consists of a circular sequence of units ($s = s_1 s_2 \dots s_D$), where s_j is the unit sequenced in position j of sequence s . For all products of type i in which $d_i \geq 2$, let t_k^i be the distance between the positions in which the units $k+1$ and k of the product of type i are found (i.e., the number of positions between them, where the distance between two consecutive positions is considered equal to 1). As the sequence is circular, position 1 comes immediately after position D ; therefore, $t_{d_i}^i$ is the distance between the first unit of the product of type i in a cycle and the last unit of the same type of product in the preceding cycle. Let $\bar{t}_i$ be the average distance between two consecutive units of the product of type i ($\bar{t}_i = D/d_i$). For all products of type i in which $d_i = 1$, t_1^i is equal to $\bar{t}_i$. The

objective is to minimize the $RTV = \sum_{i=1}^p \sum_{k=1}^{d_i} (t_k^i - \bar{t}_i)^2$.

For example, let $p=3$, $d_A=2$, $d_B=2$ and $d_C=4$; thus, $D=8$, $\bar{t}_A=4$, $\bar{t}_B=4$ and $\bar{t}_C=2$. Any sequence is a feasible solution. For example, the sequence (C, A, C, B, C, B, A, C) is a solution, where $RTV = ((5-4)^2 + (3-4)^2) + ((2-4)^2 + (6-4)^2) + ((2-2)^2 + (2-2)^2 + (3-2)^2 + (1-2)^2) = 2+8+2=12$.

The RTVP was first presented by Corominas et al. (2007), in which a simple optimal algorithm for a two-product case, an MILP model to solve the problem optimally and five greedy heuristic algorithms were proposed. The MILP model has a practical limit for obtaining optimal solutions around 25 units to be scheduled. Corominas et al. (2006) improve the MILP model and increase the practical limit to 40 units.

García et al. (2006) solved the RTVP by means of three metaheuristic procedures: a multi-start, a GRASP (Greedy Randomized Adaptive Search Procedure) and a PSO (Particle Swarm Optimization) algorithm. These three algorithms are the most efficient algorithms published to date for solving non-small instances; we compare them with the

proposed EM algorithm in Section 5. Next, the algorithms are briefly explained (for more details of the three algorithm procedures, see García et al., 2006).

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 (Martí, 2003). Random solutions are generated as follows. For each position, a type of product to be sequenced is randomly chosen. The probability of each type of product is equal to the number of units of this type of product 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 optimization ends when no neighbouring solution is better than the current solution.

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 in selecting the next type of product to be added to the solution; the probability of each candidate type of product is proportional to the value of its Webster index, which is based on the parametric method of apportionment with parameter $\delta = 1/2$ (Balinski and Young, 1982). The

Webster index for the type of product $i = 1, \dots, p$ is evaluated as $\frac{d_i}{(x_{it} + \delta)}$, where x_{it} is the number of units of the product of type i in the sequence of length $t = 0, \dots, D$. The local search procedure applied to the initial solutions is the same local search as in the multi-start method.

PSO is a population-based metaheuristic algorithm 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 n -dimensional real point (which represents a feasible solution) and a velocity (the movement of the point in the n -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 algorithm as it looks for an optimal solution. García et al. (2006) propose four PSO variations, although the comparison is only carried out for the best of these (referred to as *PSO-MIF*). Although the PSO algorithm was originally designed for working in an n -dimensional real space, *PSO-MIF* is adapted to work with a sequence of integer numbers that represents the solution. In this adaptation of the PSO algorithm, a point is now the sequence of integer numbers that represents a solution and the velocity is an ordered list of transformations that must be applied to the particle so it changes from its current point to another point; each transformation consists of a pair of positions of the point (sequence) to be swapped. In the case of the velocity to the best point found by the particle, this velocity is a list of transformations needed to obtain the best particle point from the current position; the case is the same for the velocity to the best point found by the swarm. The initial points are generated as in the multi-start method.

3. The EM algorithm

The electromagnetism-like mechanism (EM) algorithm is a new population-based metaheuristic algorithm created by Birbil and Fang (2003). The EM algorithm has been applied to a few problems successfully: for example, the resource-constrained project scheduling problem (Debels and Vanhoucke, 2004), neural network training (Wu et al., 2004), the permutation flowshop scheduling problem (Yuan et al., 2006) or multi-objective optimization problems (Tsou and Kao, 2006).

The EM algorithm basically operates as follows. The EM algorithm 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 minimize the value of the solution). 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 algorithm; for further details, see Birbil and Fang (2003). This algorithm works with a special class of optimization problems with bounded variables in the following form:

$$\begin{aligned} & \min (\max) \quad f(x) \\ & \text{subject to } x \in \mathbb{R}^n \mid l_j \leq x_j \leq u_j, j = 1, \dots, n \end{aligned}$$

where f is the function that evaluates a point (which represents a solution), n is the dimension of the problem (in the case of the RTVP, n 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 algorithm is divided into four phases (which are explained in Subsections 3.1 to 3.4): 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 algorithm is shown in Figure 1.

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

Figure 1. Pseudocode of the EM algorithm.

3.1. Initial population

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

3.2. 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 *lsiter*) is executed.

3.3. 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 algorithm. The charge is first evaluated as follows:

$$q_x = \exp \left(-n \frac{f(x) - f(x^{best})}{\sum_{y \in P} (f(y) - f(x^{best}))} \right) \quad (1)$$

Note that, unlike electrical charges, no signs are associated with the charges. The direction of a particular force between two points is determined once 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:

$$F_x = \sum_{y \in P | y \neq x} \left\{ \begin{array}{ll} (y-x) \frac{q_x q_y}{\|y-x\|^2} & \text{if } f(y) < f(x) \quad (\text{Attraction}) \\ (x-y) \frac{q_x q_y}{\|y-x\|^2} & \text{if } f(y) \geq f(x) \quad (\text{Repulsion}) \end{array} \right\} \quad (2)$$

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

3.4. Movement according to the total force

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

$$x = x + \lambda \frac{F_x}{\|F_x\|} \quad (3)$$

where λ denotes a random number uniformly distributed between 0 and 1 and $\|F_x\|$ is the norm of the force vector. The parameter λ 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).

4. The EM algorithm procedure for the RTVP

The objective function and the equations of the EM algorithm work with points of a region of the n -dimensional real space. Others procedures such as the PSO or other optimization algorithms of real variables are also designed for working in an n -dimensional real space. However, a solution of many combinatorial optimization problems is usually represented as an ordered sequence of integer numbers (as in the RTVP), so these algorithms (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 n -dimensional real point.

To adapt the PSO algorithm to a sequence of integer numbers for the RTVP is done in García et al. (2006), 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 2) would now be the difference between two sequences of integer numbers and this new difference operator would have to be defined.

On the other hand, a sequence of integer numbers can be represented by an n -dimensional real point using random key representation (Bean, 1994), which is used in an EM procedure for solving the permutation flowshop scheduling problem, for example (Yuan et al., 2006). In this paper, random key representation is also used in the EM procedure 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 the EM procedure 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 (1), (2) and (3). Finally, Subsection 4.4 explains the fine-tuning of the parameter values of the EM algorithm: the size of the initial population (m) and the maximum number of iterations of the local search procedure (l_{iter}).

4.1. Random key representation

Random key representation (Bean, 1994) consists of an n -length sequence of different real numbers called *keys*. Let the key sequence be $r = r_1, \dots, r_n$, where r_j is the key of the position j . In the context of the proposed EM procedure, the key sequence has D (number of units to be sequenced with d_i units of the product of type i) keys. As the EM algorithm 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 types of products) that is represented by r is as follows. First, for each position $j = 1, \dots, D$ of r , the key r_j is

associated with a type of product. The association is done in a way that, for each type of product i , there are d_i consecutive keys associated with the type of product i . For each key sequence, the association for the key r_j will be always be with the same type of product, i.e., if, for example, the key r_1 is associated with the product of type A in every key sequence r , r_1 will be associated with this type of product. Next, a new key sequence, r' , is obtained by putting r (and therefore their associated types of products) in descending order according to the values of the keys. Then, for each position $j = 1, \dots, D$, the type of product s_j is the type of product associated with the key r'_j , i.e., the type of product sequenced in the position j is the type of product associated with the key r_j that is in the position j in the key sequence r' .

For example, let a RTVP instance be $p = 3$, $d_A = 2$, $d_B = 2$ 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 ($j = 1, \dots, 8$) is associated with a type of product as follows:

products	A	A	B	B	C	C	C	C
	↑	↑	↑	↑	↑	↑	↑	↑
keys	0.12	0.26	0.67	0.08	0.14	0.45	0.87	0.62

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).

4.2. Initial population

The initial population of points consists of m 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 number uniformly distributed in $[0,1]$ for each key.

4.3. Local search

The local search procedure used in the EM procedure 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 types of products; the first solution found in the neighbourhood that is better than the current solution is selected; the optimization 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 procedure.

4.4. Fine-tuning of the EM parameters

Fine-tuning the parameters of a metaheuristic algorithm is almost always a difficult task. Although the value of the parameters is vital because the results of the metaheuristic for each problem are very sensitive to them, fine-tuning is usually done by intuitively testing several values (Adenso-Díaz and Laguna, 2006).

The problem of fine-tuning the parameters of a metaheuristic algorithm can be approached as an optimization problem, in which the solution consists in finding the parameter values that optimize the running of the metaheuristic for the problem to solve. Since the set of instances of a problem is infinite, we must resign ourselves to a representative training set with which to carry out the optimization.

The Nelder and Mead (N&M) algorithm (Nelder and Mead, 1965) has been chosen to solve the fine-tuning problem because it is a direct algorithm (i.e., it uses only the values of the function). Despite its early publication date, it continues to offer good results and is still referred to in recent papers (Corominas, 2005). The N&M algorithm starts from a v -dimensional point whose coordinates are the v parameters of the objective function and an initial hyper-tetrahedron is formed. For the fine-tuning problem, the parameters of the metaheuristic are used as the coordinates of the points. It is advisable that one of the initial vertices of the hyper-tetrahedron is a known good point; the N&M algorithm ensures that the solution found is never worse than the best of the initial vertices. Then, the points of the hyper-tetrahedron are iteratively moved in the v -dimensional space according to the values of the function of each point until a local optimal point is reached. The function to be used by the N&M algorithm for the fine-tuning problem of a metaheuristic is the sum of the objective function values corresponding to the solutions obtained with the metaheuristic algorithms at each instance of the training set. Usually, the bigger an instance, the bigger the objective function value of their solutions. Therefore, the N&M algorithm will give more relevance to the big instances for fine-tuning the parameters. To prevent this situation for occurring, the objective function values are normalized by dividing them by a lower bound of the instance. The lower bound used for the RTVP is the lower bound (let it be called *LBT*) introduced by Corominas et al. (2007), which is calculated as follows:

$$LBT = \sum_{i=1}^p \left[(D \bmod d_i) \cdot \left(\left\lfloor \frac{D}{d_i} \right\rfloor - \bar{t}_i \right)^2 + (d_i - D \bmod d_i) \cdot \left(\left\lceil \frac{D}{d_i} \right\rceil - \bar{t}_i \right)^2 \right]$$

Two parameters of the EM metaheuristic algorithm need fine-tuning: the size of the population (m) and the maximum number of iterations of the local search procedure (*lsiter*). To set the initial point of the hyper-tetrahedron, a brief experiment was carried out beforehand: the initial value of m was 20 and the initial value of *lsiter* was 6. A set of 60 instances (generated as it is explained in Section 5) was used to fine-tune the EM method and the EM algorithm was run for 50 seconds for each instance in order to value a point. The fine-tuning values of the parameters are finally $m = 24$ and *lsiter* = 4.

5. Computational experiment

The computational experiment for the EM algorithm is carried out for the same instances and conditions used in García et al. (2006). That is, the algorithms ran 740 instances, which were grouped into four classes (185 instances in each class) according to their size. The instances in the first class (called *CAT1*) were generated using a random value of D (number of units) uniformly distributed between 25 and 50, and a random value of p (number of type of products) uniformly distributed between 3 and 15; for the second class (called *CAT2*), D was between 50 and 100 and p between 3 and

30; for the third class (called *CAT3*), D was between 100 and 200 and p between 3 and 65; and for the fourth class (called *CAT4*), D was between 200 and 500 and p between 3 and 150. For all instances and for each type of product $i = 1, \dots, p$, a random value of d_i (number of units of the product i) is between 1 and $\left\lceil \frac{D-p+1}{2.5} \right\rceil$ such that $\sum_{i=1}^p d_i = D$.

The algorithms were coded in Java and the computational experiments were carried out using a 3.4 GHz Pentium IV with 512 Mb of RAM.

For each instance, the four metaheuristics were run for 50 seconds. Table 1 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*).

	EM	Multi-start	GRASP	PSO-MIF
Global	4,289.75	20,050.98	14,422.20	8,130.13
CAT1	19.14	11.33	13.90	68.79
CAT2	54.56	48.1	91.64	445.55
CAT3	247.84	320.63	541.52	3,050.38
CAT4	16,837.46	79,823.89	57,041.74	28,955.82

Table 1. Averages of the RTV obtained values for 50 seconds

For the global of all instances, the EM algorithm is 47.24% better than *PSO-MIF* (which is the best metaheuristic algorithm proposed by García et al., 2006), 70.26% better than the GRASP algorithm and 78.61% better than the multi-start algorithm. Observing the results in Table 1 by class, we can see that a simple algorithm such as the multi-start algorithm obtains the best averages for small instances (*CAT1* and *CAT2*) but a very poor average for large instances (*CAT4*). On the other hand, *PSO-MIF* produces bad results for small and medium instances (*CAT1*, *CAT2* and *CAT3*) and good results for large ones. Finally, the EM algorithm works fine for small instances and for medium and 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 σ) of the RTV values obtained by each metaheuristic mh ($mh = \{EM, \text{multi-start}, GRASP, PSO-MIF\}$) for a given instance, ins , is defined as follows:

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

where $RTV_{ins}^{(mh)}$ is the RTV value of the solution obtained with the metaheuristic mh for the instance ins , and $RTV_{ins}^{(best)}$ is, for the instance ins , the best RTV value of the solutions obtained with the four metaheuristics. Table 2 shows the average σ dispersion for the global and for each class of instances.

	EM	Multi-start	GRASP	PSO-MIF
Global	2.59	16.18	42.28	189.95
CAT1	1.32	0.00	0.30	59.18
CAT2	0.22	0.02	2.35	166.76
CAT3	0.03	0.21	6.06	511.40
CAT4	8.80	64.13	160.40	22.45

Table 2. Average σ dispersion regarding the best solution found for 50 seconds

For the global of all instances, the EM algorithm has the least average σ dispersion: 98.64% better than *PSO-MIF*, 93.87% better than the GRASP algorithm and 83.99% better than the multi-start algorithm. Observing the results in Table 2 by class, we see that the behaviour of the dispersions is almost analogous to the behaviour of the RTV values. For small instances (*CAT1* and *CAT2*), the multi-start algorithm gives the smallest average dispersion and it is followed by the EM algorithm and the GRASP algorithm. However, the dispersion of *PSO-MIF* is considerable because it does not work with the small and medium instances, as shown previously. Moreover, the σ dispersion obtained with *PSO-MIF* for the small and medium instances is always greater than zero, that is, *PSO-MIF* never reaches the best known solution for any of these instances. For the medium and big instances (*CAT3* and *CAT4*), the EM algorithm shows the least dispersion, followed by the multi-star algorithm for the medium instances and by *PSO-MIF* for the big instances and then, much further behind, by the multi-start and GRASP algorithms; and although the multi-start algorithm gives the worst solutions for the *CAT4* instances, it is more stable than the GRASP method. To summarize, the results in Table 2 show that the EM procedure shows on average, a stable behaviour.

In order to see the contribution of the EM algorithm with respect to the other three algorithms, two values are compared for each instance: the best RTV value obtained with the multi-start, GRASP and PSO algorithms (let it be called *MS-PSO-GRASP* value) and the best RTV value obtained with the EM, multi-start, GRASP and PSO algorithms (let it be called *EM-MS-PSO-GRASP* value). Table 3 shows the average of these values for the global and for each class of instances

	MS-PSO-GRASP	EM-MS-PSO-GRASP	EM	Multi-start	GRASP	PSO-MIF
Global	6,293.94	5,070.26	4,289.75	20,050.98	14,422.20	8,130.13
CAT1	11.25	11.21	19.14	11.33	13.90	68.79
CAT2	47.81	45.09	54.56	48.1	91.64	445.55
CAT3	311.70	230.90	247.84	320.63	541.52	3,050.38
CAT4	24,805.03	16,620.65	16,837.46	79,823.89	57,041.74	28,955.82

Table 3. Averages of the RTV obtained values

For the global of all instances, the *EM-MS-PSO-GRASP* results are, on average, 44.76% better than the *MS-PSO-GRASP* results. By observing the results in Table 3, it can be noted that the bigger the instance, the bigger the contribution of the EM algorithm to the group of the metaheuristics *EM-MS-PSO-GRASP*: 0.36% for the *CAT1* instances, 5.69% for the *CAT2* instances, 25.92% for the *CAT3* instances and 45.09% for the *CAT4* instances. The results are the expected according to the values in Table 1 and 2.

It is important to note that 200 seconds are needed to obtain the *EM-MS-PSO-GRASP* value of an instance (50 seconds for each algorithm). Therefore, applying 200 seconds for running only one of the metaheuristic algorithms must be considered and a computational experiment that consists on running each algorithm, for each instance, 200 seconds is carried out. The average of the RTV values obtained is shown for the global and for each class of instances.

	EM-MS- PSO-GRASP	EM	Multi-start	GRASP	PSO-M1F
Global	3,476.96	1,547.73	9,467.74	7,524.63	6,574.48
CAT1	11.21	18.85	10.65	12.68	66.66
CAT2	45.09	54.08	37.35	80.57	448.00
CAT3	230.90	160.01	193.99	484.92	2,986.95
CAT4	13,620.65	5,958.01	37,628.96	29,520.33	22,796.32

Table 4. Averages of the RTV obtained values

According to the results in Table 4, the best strategy is to execute for 200 seconds the EM algorithm. For the global of all instances the EM values are, on average, 55.48% better than the *EM-MS-PSP-GRASP* values, 76.45% better than the *PSO-M1F* values, 79.43% better than the GRASP algorithm values and 83.65% better than the multi-start algorithm values. By observing the results in Table 4, it can be seen that the multi-start algorithm still gives the best results, on average, for small instances (*CAT1* and *CAT2* instances) and the EM algorithm gives the best results for medium and big instances (*CAT3* and *CAT4* instances).

Results in Table 4 also prove that 50 seconds are not enough for the EM metaheuristic algorithm to converge and it has a wide space of improvement. Figure 2 shows how the average of the RTV values for the global of all instances decreases exponentially over the execution time of the EM metaheuristic. We can see that around the first 200 seconds of execution are vital for obtain, on average, a good solution. The rest of time the solutions are being improved but more slowly.

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

Finally, we compare the MILP model proposed by Corominas et al. (2006) with our EM algorithm. Corominas et al. (2006) solved 60 small RTVP instances, with a D value of between 20 and 40 and a p value of between 3 and 15, with the MILP model. We have repeated the experiment by setting the maximum execution time at 2,000 seconds and 55 instances were solved optimally. The results obtained are shown in Table 5.

	MILP (188.19 s)	EM (5 s)	EM (10 s)	EM (100 s)	EM (200 s)	EM (1,000 s.)
	9.86	15.29	15.06	15.02	14.79	14.69

Table 5. Averages of the RTV obtained values and the execution times

For these 60 very small instances, the solutions obtained with the MILP model are better than the solutions obtained with the EM algorithm, as occurs in most of the problems with enough small instances. For small instances and with enough time, it is always advisable to use an exact method. However, if there is not much time available or the instances are not small (as is usual in real life), a heuristic method such as the metaheuristics shown in this paper must be used to obtain good solutions.

6. Conclusions and future research

There are production problems in which the regularity in planning and scheduling must be considered as in the RTVP. Planning and scheduling should be integrated in a single decision level to obtain a global optimal solution. As the RTVP is a relatively new problem, in this paper we have focused our efforts on efficiently solving the RTVP prior to successfully integrating planning and scheduling.

The EM procedure is a population-based metaheuristic algorithm for optimization 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 procedure is presented for solving the RTVP. 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. The RTVP is an NP-hard optimization problem, so Artificial Intelligence techniques must be applied as metaheuristic methods to solve non-small instances. García et al. (2006) have proposed three metaheuristic algorithms for solving the RTVP (a multi-start, a GRASP and a PSO algorithm). This paper proposes to solve the RTVP by means of the EM algorithm. Computational experiments were done and the results show that EM results give a significant improvement when compared with the results of the previous three metaheuristics. Moreover, the EM procedure has a stable behaviour.

As is well known, planning and scheduling must be solved simultaneously to obtain a global optimal solution. In the future, we will study the integration of planning and scheduling in order to be able to choose the best plan between the possible ones. Boysen et al. (2007) stated that: “The quality of sequencing solutions heavily depends on the selected production orders, so that the results of the sequencing problem can likewise serve as a performance measure to evaluate the superior master schedule”. In future research, we will study two performance measures of a plan as an index to choose the most promising plan. The first index for a given plan could be the sum of the RTV values of the solutions obtained for the scheduling problems in the planning horizon. The evaluation of this index may be very expensive in terms of time, so a second index will be studied: the sum of the RTV values of the solutions obtained for the relaxed scheduling problems. The relaxation of a problem could be performed by grouping product types into fictitious product types with a demand equal to the sum of demand of the product types that are grouped together. The total demand (D) of the relaxed problem will be the same as the original problem, but the number of product types (p) will be lower. Therefore, the relaxed problem will probably be solved more quickly.

Another suggestion for future research is the following. With a good heuristic (for example, the EM method proposed in this paper) good solutions to the scheduling problem can be obtained. The solutions obtained could be analyzed to find their good characteristics, which could then be incorporated in the planning decision. For example, let us assume that 13 units of product A must be made in 2 days, and that the total production capacity for these two days, which must be shared with other products, is 103 units (33 units on the first day and 70 units on the second day). If the planning proportionally distributes the units of product A between the two days according to their production capacity, then 4 units of product A will be made on the first day and 9 units of product A on the second day. The RTV value for product A is certain to be higher than 0 because its ideal distance between two consecutive units on the first day will be $33/4 = 8.25$, and on the second day it will be $70/9 = 7.78$, neither of which are integers. Furthermore, for example, it can be observed that it is better to produce 3 units of A on the first day and 10 units of A on the second day because the ideal distances would be $33/3 = 11$ and $70/10 = 7$, respectively, which are integers. Thus, the RTV value for product A could be 0. In future work, we will investigate how to find good characteristics in the solutions.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to the anonymous reviewers and to Professor Albert Corominas (Technical University of Catalonia) for their valuable comments, which have improved the quality of this paper.

This is a preprint of an article whose final and definitive form has been published in the *International Journal of Production Research* © 2009 [copyright Taylor & Francis]; *International Journal of Production Research* is available online at **informaworld**TM

REFERENCES

- Adenso-Díaz, B. and Laguna, M. (2006) 'Fine-tuning of algorithms using fractional experimental designs and local search', *Operations Research*, Vol. 54, pp.99-114.
- Anily, S., Glass, C.A. and Hassin, R. (1998) 'The scheduling of maintenance service', *Discrete Applied Mathematics*, Vol. 82, pp.27-42.
- Balinski, M.L. and Young, H.P. (1982) 'Fair Representation: meeting the ideal of one man, one vote', *Yale University Press*, New Haven CT.
- Bean, J.C. (1994) 'Genetic Algorithms and Random Keys for Sequencing and Optimization', *ORSA Journal on Computing*, Vol. 6, No. 2, pp.154-160.
- Birbil, S.I. and Fang, S.C. (2003) 'An Electromagnetism-like Mechanism for Global Optimization', *Journal of Global Optimization*, Vol. 25, pp.263-282.
- Boysen, N., Flidner, M. and Scholl, A. (2007) 'Production Planning of Mixed-Model Assembly Lines: Overview and Extensions', *Jena Research Papers in Business and Economics*.
- Corominas, A. (2005) 'Empirically Adjusted Greedy Algorithms (EAGH): A new approach to solving combinatorial optimisation problems', *Working paper IOC-DT-P-2005-22*, Universitat Politècnica de Catalunya, Spain.

- Corominas, A., Kubiak, W. and Pastor, R. (2006) 'Solving the Response Time Variability Problem (RTVP) by means of mathematical programming', *Working paper IOC-DT*, Universitat Politècnica de Catalunya, Spain.
- Corominas, A., Kubiak, W. and Moreno, N. (2007) 'Response time variability', *Journal of Scheduling*, Vol. 10, pp.97-110.
- Debels, D. and Vanhoucke, M. (2004) 'An electromagnetism meta-Heuristic for the resource-constrained project scheduling problem', *Working Paper*, Dept. Econ. & BA, Ghent Univ., Hoveniersberg, Belgium.
- Dong, L., Melhem, R. and Mossel, D. (1998) 'Time slot allocation for real-time messages with negotiable distance constraint requirements', *Real-time Technology and Application Symposium*, RTAS, Denver.
- Feo, T.A. and Resende, M.G.C. (1989) 'A probabilistic heuristic for a computationally difficult set covering problem', *Operations Research Letters*, Vol. 8, pp.67-81.
- García, A., Pastor, R. and Corominas, A. (2006) 'Solving the Response Time Variability Problem by means of metaheuristics', *Special Issue of Frontiers in Artificial Intelligence and Applications on Artificial Intelligence Research and Development*, Vol. 146, pp.187-194.
- Han, C.C., K.J. Lin and C.J. Hou (1996) 'Distance-Constrained Scheduling and Its Applications in Real-Time Systems' *IEEE Trans. on Computers*, Vol. 45, pp. 814-826
- Herrmann, J.W. (2007) 'Generating Cyclic Fair Sequences using Aggregation and Stride Scheduling', *Technical Report*, University of Maryland, USA.
- Kennedy, J. and Eberhart, R.C. (1995) 'Particle swarm optimization', *IEEE International Conference on Neural Networks*, Australia.
- Kubiak, W. (1993) 'Minimizing variation of production rates in just-in-time systems: A survey', *European Journal of Operational Research*, Vol. 66, No. 3, pp. 259-271.
- Kubiak, W. and Sethi, S.P. (1991) 'A Note on Level schedules for mixed-model assembly lines in just-in-time production systems', *Management Science*, Vol. 37, pp. 121-122.
- Kubiak, W. and Sethi, S.P. (1994) 'Optimal just-in-time schedules for flexible transfer lines', *The International Journal of Flexible Manufacturing Systems*, Vol. 6, pp. 137-154.
- Martí, R., (2003), 'Multi-start methods', *Handbook of Metaheuristics*, Glover and Kochenberger (eds.), Kluwer Academic Publishers, pp.355-368.
- Miltenburg, J. (1989) 'Level schedules for mixed-model assembly lines in just-in-time production systems', *Management Science*, Vol. 35, No. 2, pp. 192-207.
- Monden, Y. (1983) 'Toyota Production Systems', *Industrial Engineering and Management Press*, Norcross, GA.
- Nelder, J.A. and Mead, R. (1965) 'A simplex method for function minimization', *The Computer Journal*, Vol. 7, pp.308-313.
- Steiner, G. and Yeomans S. (1993): Level Schedules for Mixed-Model, Just-in-Time Processes, *Management Science*, Vol. 39, pp. 728-735.
- Tsou, C.-S. and Kao, C.-H. (2006) 'An Electromagnetism-Like Meta-Heuristic for Multi-Objective Optimization', *Evolutionary Computation*, CEC 2006. IEEE Congress on Computational Intelligence, Canada.
- Waldspurger, C.A. and Weihl, W.E. (1995) 'Stride Scheduling: Deterministic Proportional-Share Resource Management', *Technical Report MIT/LCS/TM-528*, Massachusetts Institute of Technology, MIT Laboratory for Computer Science.

- Wu, P., Yang, K.-J. and Hung, Y.-Y. (2004) 'A study of electromagnetism-like mechanism for training fuzzy neural network' in Proc. 1st Annu. Conf. OR Society at Taiwan, Taipei, pp.471-480.
- Yuan, K., Hennequin, S., Wang, X. and Gao, L. (2006), 'A new heuristic-EM for permutation flowshop scheduling', *12th IFAC Symposium on Information Control Problems in Manufacturing*, INCOM'06, Saint-Etienne, France.