Solving the Response Time Variable Problem by means of a Variable Neighbourhood Search Algorithm

Corominas, Albert*. García-Villoria, Alberto. Pastor, Rafael*****

**Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Spain; e-mail: albert.corominas@upc.edu*
***Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Spain; e-mail: alberto.garcia-villoria@upc.edu*
****Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Spain; e-mail: rafael.pastor@upc.edu*

Abstract: The Response Time Variability Problem (RTVP) is a NP-hard combinatorial scheduling problem which has recently reported and formalised in the literature. This problem has a wide range of real-world applications in mixed-model assembly lines, multi-threaded computer systems, network environments and others. The RTVP arises whenever products, clients or jobs need to be sequenced in such a way that the variability in the time between the points at which they receive the necessary resources is minimized. The best results in the literature for the RTVP were obtained with a psychoclonal algorithm. We propose a Variable Neighbourhood Search (VNS) algorithm for solving the RTVP. The computational experiment shows that, on average, the results obtained with the proposed algorithm improve strongly on the best obtained results to date.

1. INTRODUCTION

The Response Time Variability Problem (RTVP) is a combinatorial scheduling problem that has been first time reported in Waldspurger and Weihl (1994) and was first time formalised in 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 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 resource allocation in computer multi-threaded systems and network servers (Waldspurger and Weihl, 1994, 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. To achieve this, one main goal, as Monden (1983) says, is scheduling the units to be produced to keep constant consumption rates of the components involved in the production process. Miltenburg (1989) deals with this scheduling problem and assumes that models require approximately the same number and mix of parts. Thus, only the demand rates for the models are considered. In our experience with practitioners of manufacturing industries, we noticed that they usually refer to a good mixed-model sequence 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

Corominas et al. (2007) proposed a mixed integer linear programming (MILP) model to solve the RTVP. Corominas et al. (2009) proposed an improved MILP model and increased the practical limit for obtaining optimal solutions from 25 to 40 units to be scheduled. Thus, the use of heuristic or metaheuristic methods for solving real-life instances of the RTVP is justified. Waldspurger and Weihl (1995) used the Jefferson method of apportionment (Balinski and Young, 1982), a greedy heuristic algorithm which they renamed as the stride scheduling technique. Herrmann (2007) solved the RTVP by applying a heuristic algorithm based on the stride scheduling technique. Corominas et al. (2007) proposed four other greedy heuristic algorithms. García et al. (2006) proposed six metaheuristic algorithms: a multi-start, a greedy randomized adaptive search procedure (GRASP) and four variants of a discrete particle swarm optimization (PSO) algorithm. Other ten discrete PSO algorithms were proposed in García-Villoria and Pastor (2007). A cross-entropy method approach was used in García-Villoria et al. (2007). The Electromagnetism-like Mechanism (EM) was proposed to solve the RTVP in García-Villoria and Pastor (2008a). Finally, the best results recorded to date have been obtained with a Psychoclonal algorithm (García-Villoria and Pastor, 2008b).

To improve the results obtained in prior studies, we propose to use a Variable Neighbourhood Search (VNS)-based algorithm for solving the RTVP. VNS is a metaheuristic used to solve combinatorial optimization problems (Mladenović and Hansen, 1997), as it is the RTVP. This metaheuristic is based on changing systematically the neighbourhood during a local search. The proposed VNS algorithm is compared with the most efficient procedure for solving non-small instances published in the literature, which is a psychoclonal algorithm proposed in García-Villoria and Pastor (2008b). On average, the proposed VNS algorithm improves more than 61% on the best previous results reported in the literature.

The remainder of the paper is organized as follows: Section 2 presents a formal definition of the RTVP and describes briefly the psychoclonal algorithm used for solving the problem. Section 3 proposes a VNS algorithm for solving the RTVP. Section 4 presents the computational experiment and the comparison between our algorithm and the psychoclonal algorithm. Finally, the conclusions are given in Section 5.

2. THE RESPONSE TIME VARIABILITY PROBLEM

The RTVP is designed to minimize variability in the distances between any two consecutive units of the same model and is formulated as follows. Let n be the number of models, d_i the number of units of model i to be scheduled ($i = 1,\ldots,n$), and D the total number of units ($D = \sum_{i=1..n} d_i$). Let s be a solution of an instance in the RTVP. It consists in a circular sequence of units ($s = s_1 s_2 ... s_D$), where s_j is the unit sequenced in position j of sequence s. For each model i in which $d_i \geq 2$, let t_k^i be the distance between the positions in which units $k+1$ and k of model i are found. We consider the distance between two consecutive positions to be equal to 1. Since the sequence is circular, position 1 comes immediately after position D; therefore, $t_{d_i}^i$ is the distance between the first unit of model i in a cycle and the last unit of the same model in the preceding cycle. Let $\overline{t_i}$ be the average distance between two consecutive units of model i ($\overline{t_i} = D/d_i$). Note that for each model i in which $d_i = 1$, t_1^i is equal to $\overline{t_i}$. The aim is to minimize the metric response time variability (RTV) which is defined by the following expression:

$$RTV = \sum_{i=1}^{n} \sum_{k=1}^{d_i} \left(t_k^i - \overline{t_i} \right)^2. \tag{1}$$

For example, let $n = 3$, $d_A = 2$, $d_B = 2$ and $d_C = 4$; thus, $D = 8$, $\overline{t_A} = 4$, $\overline{t_B} = 4$ and $\overline{t_C} = 2$. Any sequence that contains model i $(\forall i)$ exactly d_i times is a feasible solution. For example, the sequence (C, A, C, B, C, B, A, C) is a feasible solution, where:

$$RTV = \left((5-4)^2 + (3-4)^2\right) + \left((2-4)^2 + (6-4)^2\right) + \left((2-2)^2 + (2-2)^2 + (3-2)^2 + (1-2)^2\right) = 12.$$

As has been introduced in Section 1, the psychoclonal algorithm proposed in García-Villoria and Pastor (2008b) is the best procedure to date for solving the RTVP. Psychoclonal is an evolutionary metaheuristic first time proposed in Tiwari et al. (2005). According to the authors, this metaheuristic inherits its characteristics from the need hierarchy theory of Maslow (1954) and the clonal selection principle (Gaspar and Collard, 2000). The basic scheme of the psychoclonal metaheuristic is the following: 1) An initial population of solutions is generated and a function to evaluate the fitness of a solution is given; 2) The best solutions are selected and cloned in a number proportional to their fitness; 3) The generated clones are hypermutated (hypermutation is an operator that modifies the solution with a rate inversely proportional to the fitness of the solution); 4) A new population is formed by the best clones and by new solutions generated at random; 5) Steps 2-4 are repeated until a stop condition is reached. This metaheuristic was adapted to solve the RTVP (for a more detailed explanation, see García-Villoria and Pastor, 2008b).

3. A VNS ALGORITHM FOR SOLVING THE RTVP

Variable Neighbourhood Search (VNS) is a metaheuristic proposed recently in Mladenović and Hansen (1997) for combinatorial optimization. The basic idea of VNS is applying a systematic change of neighbourhood within a local search method (Mladenović and Hansen, 1997). According to the strategies used in changing neighbourhoods and in selecting the neighbour to be the current solution, several extensions have been proposed, but most of them keep the simplicity of the basic idea (Mladenović et al., 2003). VNS is based on the following three simple facts (Hansen and Mladenović, 2003): 1) a local minimum with respect to one neighbourhood structure is not necessarily so with another, 2) a global minimum is a local minimum with respect to all possible neighbourhood structures, and 3) It have been observed empirically that for many problems local minima with respect to one or several neighbourhood structures are relatively close to each other.

In the basic VNS proposed in (Mladenović and Hansen, 1997) there is a local search step, which can be costly for large instances of some problems (Hansen and Mladenović, 2003). In Hansen and Mladenović (1998) is proposed the Reduced VNS (RVNS), in which the local search step is removed. In this paper we propose a RVNS-based algorithm for solving the RTVP because it is shown in García et al. (2006) that the local search proposed in their paper for large RTVP instances is very costly. The general scheme of RVNS is shown in Fig. 1.

For the proposed RVNS algorithm, we have selected the following three neighbourhood structures: 1) interchanging each pair of two consecutive units of the sequence that represents the current solution (N_1), 2) interchanging each

1. Select the set of neighbourhood structures N_k (k=1..k_{max}), where k_{max} is the cardinality of the set
2. Let S an initial solution
3. While stopping condition is not reached do:
4. Set $k = 1$
5. While $k \le k_{max}$ do:
6. Select a solution S' at random from $N_k(S)$
7. If the acceptance criteria is satisfied, then set $S = S$' and set $k = 1$; otherwise set $k = k + 1$
8. End While
9. End While
10. Return S

Fig. 1. General scheme of RVNS

pair of consecutive or no-consecutive units of the sequence (N_2), and 3) inserting each unit in each position of the sequence (N_3). Note that all local optima with respect N_2 are always local optima with respect N_1 because the neighbourhood of a solution S with respect to N_1 is a subset of the neighbourhood of S with respect to N_2. Therefore, if there is not a neighbour of S with respect to N_2 that is better than S, there is not either a neighbour of S with respect to N_1 better than S. Thus, it seems that the first neighbourhood is unnecessary according to the aforementioned first and second facts in which are based VNS. To justify the addition of this neighbourhood, Section 4 will show the benefits of adding N_1 to our RNVS algorithm. The initial RTVP solution is generated as in the psychoclonal algorithm (García-Villoria and Pastor, 2008b). That is, 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 stopping condition of the algorithm is a preset time run. The original acceptance criteria used in Hansen and Mladenović (1998) is that the neighbour solution S' was better than the current solution S. But the chosen acceptance criteria for our algorithm is that the neighbour solution S' was better than or equal to the current solution S, as it is done in Tasgetiren et al. (2007). Its aim is to facilitate escaping from local optima.

4. COMPUTATIONAL EXPERIMENT

The psychoclonal algorithm proposed in García-Villoria and Pastor (2008b) is the most efficient algorithm in the literature for solving non-small RTVP instances. Therefore, we compared the performance of our proposed RVNS algorithm with that psychoclonal algorithm. In what follows of this section, we refer to our RVNS algorithm as $RVNS_{(1,2,3)}$ and the psychoclonal algorithm as Psycho. In order to justify the use of the neighbourhood N_1, we run also a RVNS algorithm without this neighbourhood structure (i.e., only N_2 and N_3 are used); we refer to this algorithm as $RVNS_{(2,3)}$.

The computational experiment was carried out for the same instances and conditions that were used in García-Villoria and Pastor (2008b). That is, the algorithms were run for 740 instances which were grouped into four classes (185 instances in each class) according to size. The instances in the first class (*CAT1*) were generated using a random value of D (number of units) distributed uniformly between 25 and 50, and a random value of n (number of models) distributed uniformly between 3 and 15; for the second class (*CAT2*), D was between 50 and 100 and n between 3 and 30; for the third class (*CAT3*), D was between 100 and 200 and n between 3 and 65; and for the fourth class (*CAT4*), D was between 200 and 500 and n between 3 and 150. For all instances and for each unit $i = 1,\ldots,n$, a random value of d_i (number of units of model i) was between 1 and $\lfloor (D-n+1)/2.5 \rfloor$ so that $\sum_{i=1..n} d_i = D$. The two algorithms were coded in Java and the computational experiment was carried out using a 3.4 GHz Pentium IV with 1.5 GB of RAM.

All algorithms were run for 50 seconds for each instance. Table 1 shows the averages of the RTV values to be minimized for the total of 740 instances and for each class of instances (*CAT1* to *CAT4*) obtained with the algorithms.

Table 1. Average RTV values for 50 seconds

	$RVNS_{(1,2,3)}$	$RVNS_{(2,3)}$	Psycho
Total	63.96	86.78	235.68
CAT1	10.73	10.63	14.92
CAT2	23.69	23.23	44.25
CAT3	51.80	53.39	137.07
CAT4	169.64	259.86	746.50

Table 1 shows that our proposed algorithm $RVNS_{(1,2,3)}$ is, on average,72.86% better than the results obtained using the best method proposed in the literature. Moreover, for each type of class of instances, the $RVNS_{(1,2,3)}$ algorithm always obtains better results than Psycho: 28.08%, 46.46%, 62.21% and 77.28% for *CAT1*, *CAT2*, *CAT3* and *CAT4* instances, respectively. We can see that the larger is the RTVP instance (and, therefore, harder to be solved), better is $RVNS_{(1,2,3)}$ compared with Psycho. Comparing $RVNS_{(1,2,3)}$ with $RVNS_{(2,3)}$, it is observed in Table 1 that very similar results are obtained for the small and medium instances (*CAT1*, *CAT2* and *CAT3* instances); on the other hand, an improvement of 34.72% is obtained for the largest instances (*CAT4* instances) when the neighbourhood N_1 is used.

Table 2 shows the number of times that each algorithm reaches the best RTV value obtained by either one. The results are shown for the total number of 740 instances and for each class.

Table 2. Number of times that the best solution is reached for 50 seconds

	$RVNS_{(1,2,3)}$	$RVNS_{(2,3)}$	Psycho
Total	587	443	57
CAT1	162	168	51
CAT2	140	144	6
CAT3	124	94	0
CAT4	161	37	0

As expected from the results in Table 1, Table 2 shows that $RVNS_{(1,2,3)}$ reaches the best solution more times. For the total number of instances, the best solution is obtained in 79.32%, 59.86% and 7.7% of cases by $RVNS_{(1,2,3)}$, $RVNS_{(2,3)}$ and Psycho, respectively.

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 algorithm alg = { $RVNS_{(1,2,3)}$, $RVNS_{(2,3)}$, Psycho } for a given instance, ins, is defined as follows:

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

where $RTV_{ins}^{(alg)}$ is the RTV value of the solution obtained with the algorithm alg for the instance ins, and $RTV_{ins}^{(best)}$ is, for the instance ins, the best RTV value of the solutions obtained with the three algorithms. Table 3 shows the average σ dispersion for the total of 740 instances and for each class of instances. The low dispersion of the two RVNS algorithms for all classes of instances means that both algorithms have a very stable behaviour. That is, when an RVNS algorithm does not obtain the best RTV value for a given instance, it obtains a value that is close to it. Psycho-RTVP has a quite stable behaviour, but its dispersion is much bigger than the dispersion of the RVNS algorithms because the Psycho performance is worse.

Table 3. Average σ dispersion regarding the best solution found for 50 seconds

	$RVNS_{(1,2,3)}$	$RVNS_{(2,3)}$	Psycho
Total	0.018	0.162	8.059
CAT1	0.030	0.020	1.003
CAT2	0.024	0.009	1.748
CAT3	0.015	0.029	5.442
CAT4	0.004	0.592	24.043

The difference of the results obtained with the three algorithms may be due to that 50 seconds is not time enough for the convergence of the algorithms for all instances, especially the largest ones. Fig. 2 shows that 1,000 computing seconds seems long enough for all algorithms to converge.

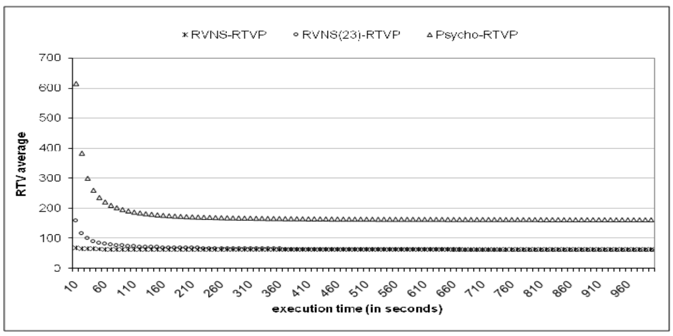

Fig. 2. Average RTV values over the computing time

Tables 4 and 5 shows the average RTV values and the average σ dispersion, respectively, for the total of 740 instances and for each class of instances obtained for 1,000 seconds.

Table 4. Average RTV values for 1,000 seconds

	$RVNS_{(1,2,3)}$	$RVNS_{(2,3)}$	Psycho
Total	62.24	62.06	161.60
CAT1	10.73	10.63	14.90
CAT2	23.29	23.19	39.90
CAT3	51.40	51.46	122.38
CAT4	163.15	162.95	469.23

Table 5. Average σ dispersion regarding the best solution found for 1,000 seconds

	RVNS	$RVNS_{(2,3)}$	Psycho
Total	0.026	0.019	4.100
CAT1	0.030	0.020	0.994
CAT2	0.024	0.008	1.256
CAT3	0.024	0.024	3.984
CAT4	0.026	0.026	10.166

Using 1,000 seconds of computing time, Psycho improves its average RTV value a 31.43% regarding the values obtained with 50 computing seconds. Nevertheless, $RVNS_{(1,2,3)}$ is still 61.49% better on average for all instances than Psycho, and 27.99%, 41.63%, 58.00% and 65.23% better for *CAT1*, *CAT2*, *CAT3* and *CAT4* instances, respectively. Moreover, we can see in Table 5 that $RVNS_{(1,2,3)}$ still obtains the best solutions or solutions very close to the best. Note that 50 seconds is almost enough time for $RVNS_{(1,2,3)}$ to converge, since it improves, on average, only 2.69% with 1,000 computing seconds.

Comparing $RVNS_{(1,2,3)}$ versus $RVNS_{(2,3)}$ for the total of all instances and for each class of instances, we can see in Table 4 and 5 that there are not significant differences between the quality of the solutions obtained with both algorithms. We expected that $RVNS_{(1,2,3)}$ and $RVNS_{(2,3)}$ give similar results when both algorithms have time to converge. The reason is that the only difference between the two algorithms is that the neighbourhood structure N_1 is not included in $RVNS_{(2,3)}$ but, as it has been explained in Section 3, all local optima with respect the neighbourhood structure N_2 (used in both algorithms) are always local optima with respect N_1, that is, N_1 is *dominated* by N_2.

Thus, the great advantage of using N_1 in $RVNS_{(1,2,3)}$ is that it helps to the algorithm to converge very fast without detrimental of its performance. This is very useful for large instances or when little computational time is available. For example, $RVNS_{(1,2,3)}$ obtains an average RTVP value for the largest instances (*CAT4*) with 10 seconds equal to 187.07, whereas the average value obtained with $RVNS_{(2,3)}$ for *CAT4* instances with 10 seconds is 550.50. The reason is because, at the beginning of the search, it is easier to find a neighbour better than the current solution using the neighbourhood structure N_1 instead of N_2. To demonstrate that, we run two times the VNS algorithm for 5 seconds for all 185 *CAT4* instances. The first time only N_1 was used (RVNS$_{(1)}$); the second time only N_2 was used (RVNS$_{(2)}$). During the 5 seconds, RVNS$_{(1)}$ generated, on average, for the total of *CAT4* instances 134,112.25 solutions, where 2.67% (3,578.74), 16.32% (21,881.05) and 81.02% (108,652.46) were better, equal and worse than the current solution, respectively. On the other hand, RVNS$_{(2)}$ generated, on average, for the total of CAT4 instances 145,364.11 solutions, where 0.42% (604.86), 6.43% (9,343.08) and 93.16% (135,416.17) were better, equal and worse than the current solution, respectively.

Finally, we compare the MILP model proposed by Corominas et al. (2009) with our RVNS$_{(1,2,3)}$ algorithm and with the psychoclonal algorithm. Corominas et al. (2009) 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 6.

Table 6. Averages of the RTV values and the execution time(in seconds)

	MILP	RVNS$_{(1,2,3)}$		Psychoclonal	
RTV	9.86	10.06	10.06	14.49	12.49
Time	188.19	0.1	10	0.1	10

Table 6 shows that RVNS$_{(1,2,3)}$ is able to converge very quickly to near optimal solutions for small instances. With only 0.1 seconds of computing time, the quality of the solutions obtained with RVNS$_{(1,2,3)}$ is very close to that obtained with MILP (only 1.99% worse). On the other hand, the psychoclonal algorithm needs 10 seconds to obtain solutions that are, on average, 21.06% worse than those from MILP.

5. CONCLUSIONS

The Response Time Variability Problem is a scheduling problem that has been acquiring a greater importance on the mixed-model assembly production lines since Toyota popularized the just-in-time production system (Monden, 1983; Miltenburg, 1989). 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. Other real-life applications of the RTVP shown in the literature are present in computer multi-threaded systems and network servers (Waldspurger and Weihl, 1994, 1995; Dong et al., 1998), in periodic machine maintenances (Anily et al., 1998) and in the collection of waste (Herrmann, 2007).

The computational experiment shows the following two points:

1. A straightforward implementation of an algorithm based on the simple metaheuristic RVNS improves strongly all the methods published in the literature, including also the algorithms based on more complex metaheuristics as Particle Swarm Optimization (García et al., 2006; García-Villoria and Pastor, 2007), Cross-Entropy method (García-Villoria et al., 2007), Electromagnetism-like Mechanism (García-Villoria and Pastor, 2008a) and Psychoclonal approach (García-Villoria and Pastor, 2008b).
2. The addition of the dominated neighbourhood structure N_1 in our RVNS algorithm makes it to converge faster to solutions of good quality. This observation may be extended to other problems and VNS algorithms, in which the addition of dominated neighbourhood structures can help them to be more efficient.

The VNS metaheuristic is very easy to be hybridized with any another metaheuristic. Since the good results obtained in the literature (Hansen and Mladenović, 2003), the hybridization of VNS with other metaheuristics proposed in the literature to solve the RTVP as PSO (García-Villoria and Pastor, 2007), EM (García-Villoria and Pastor, 2008a) or Psychoclonal (García-Villoria and Pastor, 2008b) seems a promising future line of research.

ACKNOWLEDGEMENTS

The authors are grateful to the anonymous reviews for his valuable comments which have helped to enhance this paper. 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).

REFERENCES

S. Anily, C.A. Glass and R. Hassin (1998). The scheduling of maintenance service. *Discrete Applied Mathematics,* **82**, 27-42.

M.L. Balinski, and H.P. Young (1982). Fair Representation. Yale University, USA.

A. Corominas, W. Kubiak and N. Moreno (2007). Response time variability. *Journal of Scheduling,* **10**, 97-110.

A. Corominas, W. Kubiak and R. Pastor (2009). Mathematical Programming Modeling of the Response Time Variability Problem. *European Journal of Operational Research*. doi: 10.1016/j.ejor.2009.01.014

L. Dong, R. Melhem and D. Mosse (1998). Time slot allocation for real-time messages with negotiable distance constrains requirements. *Fourth IEEE Real-Time Technology and Applications Symposium,* 131-136. Denver.

A. García, R. Pastor and A. Corominas (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,* **146**, 187-194.

A. García-Villoria and R. Pastor (2007). Introducing dynamic diversity into a discrete particle swarm optimization. *Computers & Operations Research,* In Press, Corrected Proof, available online, doi:10.1016/j.cor.2007.12.001.

A. García-Villoria, R. Pastor and A. Corominas (2007). Solving the Response Time Variability Problem by means of the Cross-Entropy Method. *Special Issue on Production Line Systems: Concepts, Methods and Applications of the International Journal of Manufacturing Technology and Management,* In Press, Corrected Proof, to be published.

A. García-Villoria and R. Pastor (2008a). Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism. *Working paper IOC-DT-P-2008-03,* Technical University of Catalonia, Spain.

A. García-Villoria and R. Pastor (2008b). Solving the Response Time Variability Problem by means of a psychoclonal approach. *Journal of Heuristics,* In Press, Corrected Proof, available online, doi:10.1007/s10732-008-9082-2.

A. Gaspar and P. Collard (2000). Two models of immunization for time dependent optimization. In: *Proceeding of the IEEE International Conference on Systems Manufacturing and Cybernetics,* 113-118.

P. Hansen and N. Mladenović (1998). *Meta-heuristics, Advances and Trends in Local Search Paradigms for Optimization*, 433-458. Kluwer Academic Publishers.

P. Hansen and N. Mladenović (2003). *Handbook of metaheuristics*, Chapter 6. Kluwer Academic Publishers.

J.W. Herrmann (2007). Generating Cyclic Fair Sequences using Aggregation and Stride Scheduling. *Technical Report,* University of Maryland. USA.

A.H. Maslow (1954). Motivation and personality. Harper & Bros, USA.

N. Mladenović and P. Hansen (1997). Variable neighbourhood search. *Computers & Operations Research,* **24**, 1097-1100.

N. Mladenović, J. Petrović, V. Kovačević-Vujčić and M. Čangalović (2003). Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search. *European Journal of Operational Research,* **151**, 389-399.

J. Miltenburg (1989). Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science,* **35**, 192-207.

Y. Monden (1983). *Toyota Productions Systems*. Industrial Engineering and Management Press. Norcross.

M.F. Tasgetiren, Y-C. Liang, M. Sevkli and G. Gencyilmaz (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research,* **177**, 1930-1947.

M.K. Tiwari, A. Prakash, A. Kumar and A.R. Mileham (2005). Determination of an optimal sequence using the psychoclonal algorithm. *ImechE, Part B: Journal of Engineering Manufacture,* **219**, 137-149.

C.A. Waldspurger and W.E. Weihl (1995). Stride Scheduling: Deterministic Proportional-Share Resource Management. *Technical Report MIT/LCS/TM-528,* Massachusetts Institute of Technology. USA.

C.A. Waldspurger and W.E. Weihl (1994). Lottery Scheduling: Flexible Proportional-Share Resource Management. *First USENIX Symposium on Operating System Design and Implementation*.