4th International Conference on Industrial Engineering and Industrial Management
XIV Congreso de Ingeniería de Organización
Donostia- San Sebastián , September 8th -10th 2010

A new metaheuristic procedure for improving the solution of the response time variability problem[*]

Albert Corominas[1], Alberto García-Villoria[1], Rafael Pastor[1]

[1] Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC).
{albert.corominas / alberto.garcia-villoria / rafael.pastor}@upc.edu

Abstract

The response time variability problem (RTVP) is a scheduling problem that arises whenever products, clients or jobs need to be sequenced in such a way that the variability in the time between the instants at which they receive the necessary resources is minimised. The RTVP is an NP-hard problem and heuristic and metaheuristic techniques are needed to solve non-small instances. The best results for the RTVP were obtained with a variable neighbourhood search algorithm and a tabu search algorithm. We propose a simulated annealing-based algorithm to solve the RTVP. A computational experiment shows that, on average, the proposed algorithm improves the best procedures published in the literature.

Keywords: response time variability, scheduling, fait sequences, metaheuristics

1. Introduction

The concept of a *fair sequence* has emerged independently from scheduling problems of diverse environments. The common aim of these scheduling problems, as defined in Kubiak (2004), is to build a fair sequence using n symbols, where symbol i (i = 1,...,n) must occur d_i times in the sequence. The fair sequence is the one which allocates a fair share of positions to each symbol s in any subsequence. This fair or ideal share of positions allocated to symbol i in a subsequence of length k is proportional to the relative importance (d_i) of symbol i with respect to the total copies of competing symbols (equal to $\sum_{i=1..n} d_i$). There is no universal definition of fairness because several reasonable metrics can be defined according to the specific problem considered.

Several fair sequencing problems have emerged, among them the Response Time Variability Problem (RTVP). This problem has been reported for the first time by Waldspurger and Weihl (1994) but formalised several years later by Corominas et al. (2007). In the RTVP, the fair sequence is the one which minimises the sum of the variability in the distances between any two consecutive copies of the same symbol. In other words, the distance between any two consecutive copies of the same symbol should be as regular as possible (i.e., ideally constant).

[*] Supported by the Spanish Ministry of Education and Science under project DPI2007-61905; co-funded by the ERDF.

In practice, the RTVP arises whenever products, clients or jobs need to be sequenced so as to minimise the variability in the time between the instants at which they receive the necessary resources (Corominas et al., 2007). This problem has a broad range of real-world applications. These include, for instance, the sequencing of mixed-model assembly lines under JIT (Kubiak, 1993; Miltenburg, 1989), the resource allocation in computer multi-threaded systems such as operating systems, network servers and media-based applications (Dong et al., 1998; Waldspurger and Weihl, 1994, 1995), the periodic machine maintenance problem when the times between consecutive services of the same machine are equal (Anily et al., 1998; Wei and Liu, 1983), the collection of waste (Herrmann, 2007, 2009), the schedule of commercial videotapes for television (Bollapragada et al., 2004; Brusco, 2008) and the design of sales catalogues (Bollapragada et al., 2004).

Corominas et al. (2007) showed that the RTVP is NP-hard. The problem can be formulated as a mixed integer linear programming as shown by Corominas et al. (2007, 2010). The best MILP model (Corominas *et al.*, 2010) is able to generate optimal solutions in a reasonable time for small instances up to 40 units. For larger instances, several heuristic and metaheuristic algorithms have been proposed for their solution. Waldspurger and Weihl (1994) propose an algorithm that generates a solution randomly. The same authors (Waldspurger and Weihl, 1995) improve their previous results using the Jefferson method of apportionment (Balinski and Young, 1982), a greedy heuristic algorithm which they renamed as the stride scheduling technique. Herrmann (2007) solve the RTVP by applying a heuristic algorithm based on the stride scheduling technique and an aggregation method is proposed in Herrmann (2009). Corominas et al. (2007) proposed five constructive type heuristic algorithms including the Jefferson’s method. Metaheuristics for the RTVP were proposed in Corominas et al. (2008, 2009a, 2009b) and in García-Villoria and Pastor (2009a, 2009b, 2010a, 2010b). These include multi-start and greedy randomized adaptive search procedure (GRASP), variable neighbourhood search (VNS), tabu search (TS), particle swarm optimisation (PSO), electromagnetism-like mechanism (EM), psychoclonal algorithm abd genetic algorithm (GA), respectively.

To the best of our knowledge, any simulated annealing (SA) approach has been proposed to solve the RTVP. In this study we propose a straightforward application of the SA metaheuristic for solving the RTVP. A computational experiment shows that a simple SA-based algorithm is able to improve, on average, the best results published in the literature.

The remainder of the paper is organized as follows. First, Section 2 presents a formal definition of the RTVP. The next section proposes a SA-based algorithm to improve the solution of the RTVP. The results of our computational experiment are shown and discussed in Section 4. Finally, some conclusions are suggestions for future research are provided in Section 5

2. The Response Time Variability Problem

The formulation of the RTVP is as follows. Let n be the number of symbols, d_i the number of copies to be scheduled of symbol i (i = 1,…,n) and D the total number of copies ($D = \sum_{i=1..n} d_i$). Let s be a solution of a RTVP instance that consists of a circular sequence of copies with D positions ($s = s_1 s_2 \mathrm{K}\ s_D$), where s_j is the copy sequenced in position j of sequence s. For all symbol i such that $d_i \geq 2$, let t_k^i be the distance between the positions in which the copies k + 1 and k of symbol 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, $t^i_{d_i}$ is the distance between the first copy of product *i* in a cycle and the last copy of the same symbol in the preceding cycle. Let $\bar{t}_i$ be the average or ideal distance between two consecutive copies of symbol *i* ($\bar{t}_i = D/d_i$). Note that for all symbol *i* in which $d_i = 1$, t^i_1 is equal to $\bar{t}_i$. The objective is to minimise the metric Response Time Variability (RTV) which is defined by the following expression:

$$RTV = \sum_{i=1}^{n} \sum_{k=1}^{d_i} (t^i_k - \bar{t}_i)^2 \tag{1}$$

For an illustration, consider the following example. Let $n = 3$ with symbols *A*, *B* and *C*. Also consider $d_A = 3$, $d_B = 2$ and $d_C = 2$; thus, $D = 7$, $\bar{t}_A = 7/3$, $\bar{t}_B = 7/2$ and $\bar{t}_C = 7/2$. Any sequence such that contains symbol $i \;\; \forall i$ exactly d_i times is a feasible solution. For instance, the sequence (A, B, A, C, B, C, A) is a feasible solution, which has an RTV value equal to $\left(\left(2-7/3\right)^2 + \left(4-7/3\right)^2 + \left(1-7/3\right)^2 \right) + \left(\left(3-7/2\right)^2 + \left(4-7/2\right)^2 \right) + \left(\left(2-7/2\right)^2 + \left(5-7/2\right)^2 \right) = 29/3$.

3. A simulated annealing algorithm for the RTVP

The simulated annealing metaheuristic (SA) was proposed by Kirkpatrick et al. (1983) to solve complex combinatorial optimisation problems, as the RTVP is. Since then, SA has been successfully applied for solving a wide range of combinatorial optimisation problems (Henderson et al., 2003).

SA can be seen as a variant of a local search procedure in which is allowed moving to a worse solution with small probability. The objective of accepting worse solutions is to avoid being trapped into a local optimum. The metaheuristic starts from an initial solution, which is initially the current solution. Then, at each iteration, a new solution from the neighbourhood of the current solution is considered. If the neighbour is not worse than the current solution, then the neighbour becomes the current solution; in the case that is worse, the neighbour can become also the current solution with a probability that depends on: 1) how worse is the neighbour, and 2) the value of a parameter called *temperature*, which is decreased every certain number of iterations. The general pseudo-code of SA (when minimising the objective function) is shown in Figure 1.

Several decisions have to be taken before applying the general scheme of SA to solve the RTVP. Some of these decisions are general and the others are specific for the problem to solve. Specific decisions for the RTVP are the representation of solutions and the neighbourhood of each solution ($N(s)$), the generation of the initial solution and the objective function ($f(s)$) (explained in Sections 3.1, 3.2 and 3.3, respectively). General decisions are the way to decrease the temperature ($A(t)$) and the stopping criterion of the algorithm (explained in Sections 3.4 and 3.5, respectively). Moreover, the parameters of the algorithm need to be fine-tuned before the execution (explained in Section 3.6).

```
Let $f(s)$ be the objective function of the solution $s$ to be minimised
Let $N(s)$ the neighbourhood of solution $s$
Let $A(t)$ the new temperature calculated from the temperature $t$
0.  Set the parameters:
            $t_0$ (initial temperature)
            $itt$ (number of iterations between two consecutive changes of temperature)
1.  $t := t_0$;
2.  $s$ := Generate the initial solution
3.  While stopping criterion is not reached do:
4.          $i := 0$
5.          While $i < itt$ do:
6.                  $s'$ := choose at random a solution from $N(s)$
7.                  $\Delta := f(s') - f(s)$
8.                  If $\Delta \leq 0$ Then $s := s'$
9.                  If $\Delta > 0$ Then $s := s'$ with probability $\exp(-\Delta/t)$
10.                 $i := i + 1$
11.         End while
12.         $t := A(t)$
13. End while
14. Return the best solution found
```

Figure 1. Pseudo-code of SA

3.1. Representation and neighbourhood of solutions

The representation of a solution is the sequence of copies of the symbols, in which each symbol i appears d_i times. The neighbourhood of a solution is generated interchanging each pair of two consecutive copies of the sequence that represents the solution. This neighbourhood has been successfully applied when solving the RTVP with a multi-start and GRASP algorithm (Corominas et al., 2008) and a VNS algorithm (Corominas et al., 2009a).

3.2. Initial solution

The initial solution is generated using the lottery scheduling (Waldspurger and Weihl, 1994) as it is done in previous works published in the literature when an initial solution is required. That is, for each position, a symbol to be sequenced is randomly chosen. The probability of each symbol is equal to the number of copies of this symbol that remain to be sequenced divided by the total number of copies that remain to be sequenced. The random generation of the initial solution for a SA algorithm is usually done in the literature (Dowsland and Adenso-Díaz, 2003).

3.3. Objective function

In the case of the RTVP, the objective function to be minimised is the RTV value of the solution (Equation 1).

3.4. Decreasing the temperature

The temperature of the SA algorithm influences on the probability of acceptance of worse neighbouring solutions. The higher the temperature, the more probable (Step 9 in Figure 1). The most popular way in the literature that obtains good results is the geometric reduction, that is, $A(t) = t.\alpha$, where $\alpha < 1$ (Dowsland and Adenso-Díaz, 2003; Henderson et al., 2003). The α value has to be set; thus, α becomes another parameter of the algorithm.

3.5. Stopping criterion

The algorithm stops when it has run for a preset available time (it is the same criterion that has been usually used in previous proposed metaheuristic algorithms for the RTVP).

3.6. Fine-tuning the algorithm parameters

Fine-tuning the parameters of a metaheuristic algorithm is almost always a difficult task. Although the parameter values may have a very strong effect on the results of the metaheuristic for each problem, they are often selected using one of the following methods, which are not sufficiently thorough (Eiben et al., 1999; Adenso-Díaz and Laguna, 2006): 1) “by hand”, based on a small number of experiments that are not 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) with no apparent explanation.

Adenso-Díaz and Laguna (2006) proposed a new technique called CALIBRA for fine-tuning the parameters of algorithms. CALIBRA is based on using conjointly Taguchi’s fractional factorial experimental designs and a local search procedure. We propose to use CALIBRA for setting the parameter values of the proposed algorithm. A training set of 60 instances, which were generated as explained in Section 4, is used. The following parameter values were obtained: $t_0 = 13$, $itt = 1{,}762$ and $\alpha = 0.9875$.

4. Computational experiment

The two best methods to solve the RTVP are the variable neighbourhood search (VNS) algorithm proposed in Corominas et al. (2009a) and the tabu search algorithm proposed in Corominas et al. (2009b). The TS algorithm is slightly better for solving small and medium RTVP instances whereas the VNS algorithm is clearly the best for solving the largest instances. Therefore, we compare the performance of our proposed SA algorithms with them.

All algorithms are coded in Java and executed on a 3.4 GHz Pentium IV with 1.5 GB of RAM. The same 60 training instances and 740 test instances used in Corominas et al. (2009a, 2009b) are also used in this paper (all instances can be found at https://www.ioc.upc.edu/EOLI/research/). These instances were grouped into four classes (from *CAT1* to *CAT4* with 15 training instances and 185 test instances in each class) according to their size. The instances were generated using the random values of *D* (total number of copies) and *n* (number of symbols) shown in Table 1. For all instances and for each symbol $i = 1,\dots,n$, a random value of d_i is between 1 and $\lfloor D-n+1\ /2.5\rfloor$ such that $\sum_{i=1..n} d_i = D$.

Table 1. Uniform distribution for generating the *D* and *n* values

	CAT1	*CAT2*	*CAT3*	*CAT4*
D	U(25, 50)	U(50, 100)	U(100, 200)	U(200, 500)
n	U(3, 15)	U(3, 30)	U(3, 65)	U(3, 150)

The algorithms were run for 10, 50 and 1,000 seconds for each instance. Table 2 shows the overall average RTV values for the 740 test instances and for each class of instances (*CAT1* to *CAT4*) obtained with the three algorithms, respectively.

Table 2. Average RTV values for the SA, VNS and TS algorithms

		Global	***CAT1***	***CAT2***	***CAT3***	***CAT4***
	SA	108.46	10.26	21.67	45.68	356.24
10 s.	**VNS**	68.60	10.73	23.72	52.87	187.07
	TS	339.59	10.42	25.32	128.29	1,194.31
	SA	50.87	10.26	21.67	44.57	126.98
50 s.	**VNS**	63.96	10.73	23.96	51.80	169.64
	TS	210.47	10.26	22.56	73.26	735.78
	SA	50.75	10.26	21.67	44.55	126.54
1,000 s.	**VNS**	62.24	10.73	23.29	51.40	163.15
	TS	78.62	10.24	21.16	48.12	234.96

After 1,000 computing seconds, the best overall RTV average is obtained with SA, which is 18.46% and 35.45% better than the RTV average obtained with VNS and TS, respectively. Observing the results by class, we can see that TS is still slightly better than SA for solving the small instances: 0.19% and 2.35% better averages for *CAT1* and *CAT2* instances, respectively, although for *CAT1* instances no significant differences (with a confidence level of 95%) are observed. On the other hand, SA outperforms TS and VNS when solving the medium and large instances. Specifically, the SA average for *CAT3* instances is 13.33% and 7.42% better than the VNS and TS averages, respectively, and 22.44 and 46.14% better for *CAT4* instances, respectively.

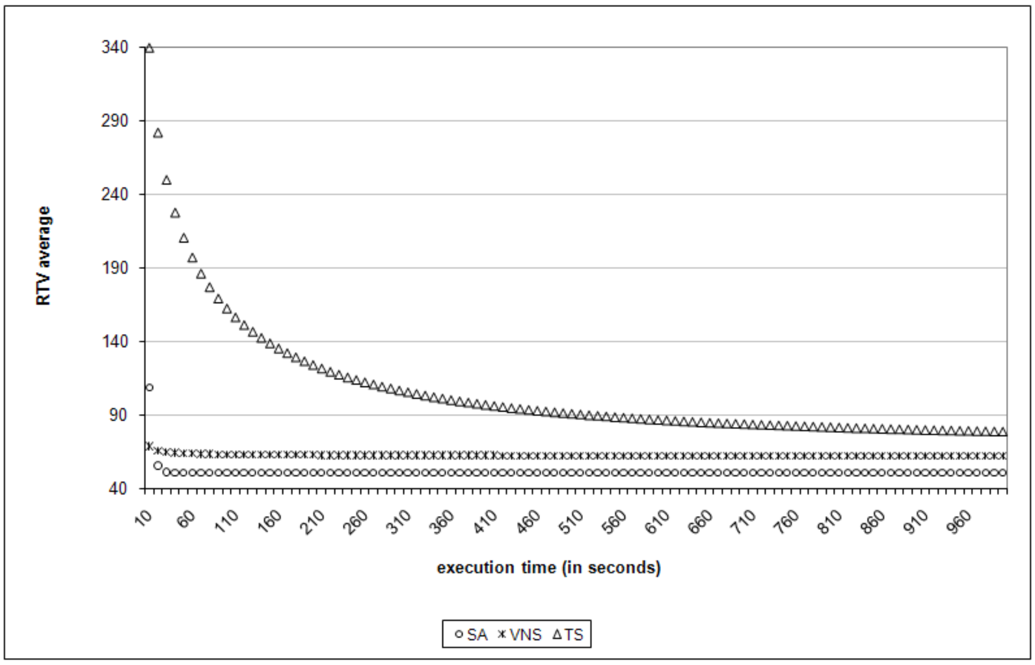

Figure 2. Average RTV values over the computing time

Table 2 also shows that the proposed SA algorithm converges very quickly and the results obtained after 50 computing seconds are very similar to the ones obtained after 1,000 computing seconds. Figure 2 shows the evolution of the average RTV values for the global of all instances during the computing time.

Table 3 shows the number of times that each algorithm reaches the best RTV value obtained by either one after 1,000 computing seconds. The results are shown for the total number of 740 instances and for each class. For the smallest (*CAT1*) instances, TS finds always the best solution, although SA is able to find the best solution for 98.38% of instances. For the *CAT2* instances, TS clearly finds the best solution more times (for the 98.38% of instances) followed by SA, which finds the best solution for 75.67% of *CAT2* instances. For the medium and large instances, the SA algorithm is clearly the one that most times finds the best solution (72.43% and 93.51% of times for the *CAT3* and *CAT4* instances, respectively).

Table 3. Number of times that the best solution is reached

	Global	***CAT1***	***CAT2***	***CAT3***	***CAT4***
SA	629	182	140	134	173
VNS	311	153	87	47	24
TS	457	185	182	82	8

To complete the analysis of the results, we examined the deviation of the results from the best solution obtained by either algorithm. A measure of the deviation (let it be called σ) of the RTV values obtained by each algorithm *alg* = {SA, VNS, TS} was defined for a given instance, *ins*, according to the following expression:

$$\sigma(alg, ins) = \left(\frac{\mathrm{RTV}_{ins}^{(alg)} - \mathrm{RTV}_{ins}^{(best)}}{\mathrm{RTV}_{ins}^{(best)}} \right)^2 \tag{2}$$

where $\mathrm{RTV}_{ins}^{(alg)}$ is the RTV value of the solution obtained with the algorithm *alg* for the instance *ins*, and $\mathrm{RTV}_{ins}^{(best)}$ is the best RTV value of the solutions obtained with the four algorithms for the instance *ins*. Table 4 shows the maximum σ deviation for the total number of instances and for each class. We can see that low deviations are always obtained with the SA algorithm for any instance. That is, when the algorithm does not obtain the best RTV value for a given instance, it obtains a value that is very close to it.

Table 4. Maximum σ values with respect to the best solution found

	Global	***CAT1***	***CAT2***	***CAT3***	***CAT4***
SA	0.10	0.09	0.10	0.05	0.07
VNS	2.69	2.25	1.76	0.96	2.69
TS	36.65	0	0.11	2.00	36.65

5. Future lines of research

In this paper, the response time variability problem (RTVP) is solved. This scheduling problem arises in a variety of real-world environments including mixed-model assembly lines, multi-threaded systems, periodic machine maintenance and waste collection, among others. The aim of the RTVP is to minimise the variability in the distances between any two consecutive copies of the same symbol.

The RTVP is an NP-hard problem and heuristic and metaheuristic methods are needed to solve real-world, non-small instances. Several metaheuristic algorithms have been developed for solving this hard combinatorial optimisation problem. The most efficient algorithms to date for solving the RTVP were a VNS-based algorithm (Corominas et al., 2009a) and a TS-based algorithm (Corominas et al., 2009b).

In this study we propose a straightforward application of SA to solve the RTVP. The results of the computational experiment show that, on average, our algorithm outperforms the existing non-exact procedures in the literature to solve non small instances. Moreover, the proposed algorithm is very robust; that is, all solutions obtained improve the best known or are very close to them.

In our SA algorithm, the neighbourhood of a solution is generated by swapping each pair of consecutive copies of the sequence that represents the solution. Better performances may be obtained by using other neighbourhood structures. Other candidates of neighbourhoods for a future research are, for example, the following ones: 1) by swapping each pair of consecutive or non consecutive copies of the sequence, and 2) by insertion.

The SA may arrive to a different local optimum according to the initial solution that is generated. Another promising line of future research is to hybridize the SA metaheuristic under a multi-start scheme.

References

Adenso-Díaz, B.; 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.; Hassin, R. (1998). The scheduling of maintenance service. Discrete Applied Mathematics, Vol. 82, pp. 27-42.

Balinski, M.L.; Young, H.P. (1982). Fair Representation. Yale University Press, New Haven.

Bollapragada, S.; Bussieck, M.R.; Mallik, S. (2004). Scheduling Commercial Videotapes in Broadcast Television. Operations Research, Vol. 52, pp. 679-689.

Brusco, M.J. (2008). Scheduling advertising slots for television. Journal of the Operational Research Society, Vol. 59, pp. 1363-1372.

Corominas, A.; García-Villoria, A.; Pastor, R. (2009a). Solving the Response Time Variable Problem by means of a Variable Neighbourhood Search Algorithm. 13th IFAC Symposium of Information Control Problems in Manufacturing (INCOM 2009), Moscow, Russia.

Corominas, A.; García-Villoria, A.; Pastor, R. (2009b). Resolución del response time variability problem mediante tabu search., VIII Evento Internacional de Matemática y Computación (COMAT'2009), Universidad de Matanzas, Cuba.

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

Corominas, A.; García-Villoria, A.; Pastor, R. (2008). Solving the Response Time Variability Problem by means of Multi-start and GRASP metaheuristics. Special Issue of Frontiers in Artificial Intelligence and Applications on Artificial Intelligence Research and Development, Vol. 184, pp. 128-137.

Corominas, A.; Kubiak, W.; Pastor, R. (2010). Mathematical programming modeling of the Response Time Variability Problem. European Journal of Operational Research, Vol. 200, pp. 347-357.

Dong, L.; Melhem, R; Mosse, D. (1998). Time slot allocation for real-time messages with negotiable distance constrains requirements. Fourth IEEE Real-Time Technology and Applications Symposium (RTAS'98), Denver, CO. pp. 131-136.

Dowsland, K.A.; Adenso-Díaz, B. (2003). Heuristic design and fundamentals of the Simulated Annealing. Inteligencia Artificial, Vol. 19, pp. 93-102.

Eiben, A.E.; Hinterding, R.; Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. IEEE Transactions on evolutionary computation, Vol. 3, pp. 124-141.

García-Villoria, A.; Pastor, R. (2009a). Introducing dynamic diversity in a discrete Particle Swarm Optimization. Computers & Operations Research, Vol. 36, pp. 951-966.

García-Villoria, A.; Pastor, R. (2009b). Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism. International Journal of Production Research, doi: 10.1080/00207540902862545.

García-Villoria, A.; Pastor, R. (2010a). Solving the Response Time Variability Problem by means of a psychoclonal approach. Journal of Heuristics, Vol. 16, pp. 337-351.

García-Villoria, A.; Pastor, R. (2010b). Solving the Response Time Variability Problem by means of a genetic algorithm. European Journal of Operational Research, Vol. 202, pp. 320-327.

Henderson, D.; Jacobson, S.H.; Johnson, A.W. (2003). The Theory and Practice of Simulated Annealing. Chapter 10 in Handbook of Metaheuristics, Eds. Glover and Kochenberger, Kluwer Academic Publishers, pp. 287-319.

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

Herrmann, J.W. (2009). Using aggregation to reduce response time variability in cyclic fair sequences. Journal of Scheduling, doi 10.1007/s10951-009-0127-7.

Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. (1983). Optimization by simulated annealing. Science, Vol. 220, pp. 671-680.

Kubiak, W. (1993). Minimizing variation of production rates in just-in-time systems: A survey. European Journal of Operational Research, Vol. 66, pp. 259-271.

Kubiak, W. (2004). Fair Sequences. Chapter 19 in Handbook of Scheduling: Algorithms, Models and Performance Analysis, Chapman and Hall.

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

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

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.

Wei, W.D.; Liu, C.L. (1983). On a periodic maintenance problem. Operations Research Letters, Vol. 2, pp. 90-93.