Control and Cybernetics

vol. **29** (2000) No. 3

An adaptive hybrid metaheuristic for permutation flowshop scheduling

by

João Vitor Moccellin and Maristela O. dos Santos

University of São Paulo
São Carlos – SP, Brasil
E-mail: jvmoccel@prod.eesc.sc.usp.br

Abstract: In this paper we present a hybrid Tabu Search — Simulated Annealing heuristic for the minimal makespan flowshop sequencing problem. In order to evaluate the effectiveness of the hybridization we compare the hybrid metaheuristic with pure Tabu Search and Simulated Annealing heuristics. The results from computational experience are discussed.

Keywords: flowshop sequencing, hybrid metaheuristics, tabu search, simulated annealing.

1. Introduction

This paper deals with the Permutation Flowshop Scheduling problem. Since it is NP-hard, many heuristic methods have been proposed for sequencing jobs in a flowshop with the objective of minimizing makespan. A class of such heuristics finds a good solution by improving an initial sequence for the jobs through neighborhood search techniques as Tabu Search (TS) and Simulated Annealing (SA). Recently, both TS and SA methods have been formulated for solving this scheduling problem, see, for instance, Widmer and Hertz (1989), Osman and Potts (1989), Reeves (1993), Moccellin (1995), Ishibuchi, Misaki and Tanaka (1995), Zegordi, Itoh and Enkawa (1995), Nowicki and Smutnicki (1996), Park and Kim (1998), and Moccellin and Nagano (1998).

A promising approach for solving the problem is the formulation of hybrid search heuristics by combining TS and SA techniques so that the consequent procedure is more effective than either *pure* TS or SA methods. Regarding the flowshop sequencing problem there is the paper of Díaz (1996) in which an initial solution improved by Simulated Annealing has been used as the starting job sequence for a Tabu Search procedure, having as objective function the a hybrid heuristic named Genetic Simulated Annealing for minimal makespan flowshop scheduling.

In this paper we present a hybrid Tabu Search — Simulated Annealing heuristic with the objective of minimizing makespan in Permutation Flowshops. In order to evaluate the effectiveness of the hybridization we compare the hybrid metaheuristic with pure Tabu Search and Simulated Annealing heuristics, denoted by FShop.TS5 (Moccellin and Nagano, 1998) and SAfshopH (Moccellin, 1994). The results from computational experience are discussed.

2. Pure tabu search and simulated annealing heuristics

2.1. The FShop.TS5 heuristic

FShop.TS5 is a tabu search heuristic in which the initial solution is the same as the one of the FSHOPH heuristic (Moccellin, 1995). The FShop.TS5 tabu search procedure is defined as follows:

Neighborhood structure

Removing a job from its position and inserting it at another position obtains a sequence neighbor. This is known as the shift neighborhood and its size for an arbitrary n-job sequence is $(n-1)^2$.

Neighborhood examination

In the FShop.TS5 heuristic partial neighborhood examination is proposed. The basic idea is to reduce the neighborhood examination in order to allow more iteration. At each step of the tabu search procedure the neighbor sequences are randomly generated and the number of neighbors to be evaluated is given by:

$$f(n) = (n-1)^2 \text{ for } n < 30, \text{ and} \tag{1}$$

$$f(n) = 1741 - 900\exp(-0.04(n-30)) \text{ for } n \geq 30, \tag{2}$$

where n = number of jobs.

These expressions were obtained by former experience on a research carried out by Moccellin (1995), and they are illustrated in Fig. 1. Moccellin has noted that for problems having more than 60 jobs the performance of the FSHOPH heuristic reached low values. This is due to the unreasonable computation time for each tabu search iteration, and, as a consequence, the total number of iterations must be reduced, unless CPU time is of no concern. On the other hand, for problems having up to 40 jobs the FSHOPH heuristic has yielded very good solutions in reasonable computation times. In the FSHOPH heuristic a sequence neighbor is obtained by just exchanging two jobs in any pair of positions. This type of neighborhood is known as the interchange one and its size for each sequence is $n(n-1)/2$, where n is the number of jobs. Let us note that the size

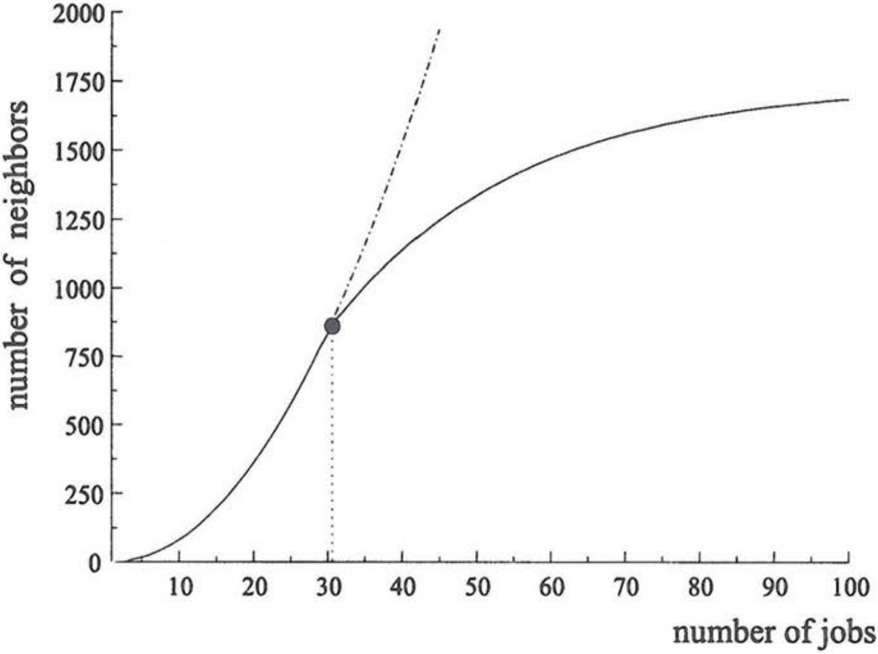


Figure 1. Number of neighbors evaluated at each iteration of the FShop.TS5 heuristic

Taking the above experimental information into account, for $n > 40$ an exponentially smoothed function $f(n)$ was adjusted, so that even for large problems the number of neighbors to be evaluated at each tabu search iteration is less than 1770. By using a similar approach, the function $f(n)$ was obtained, regarding the shift neighborhood, which is used in this paper.

Stopping condition

The stopping condition for the tabu search procedure is given by a parameter named Nbmax (maximum number of successive iterations without improving the current best solution).

The basic idea of the partial neighborhood examination is to share the computational effort between neighborhood search and the total number of iterations. Having this in mind the selection of the Nbmax stopping condition according to Table 1 is proposed, based on the stopping condition used in the FSHOPH heuristic. As can be easily noted, in the tabu search procedure the larger the parameter Nbmax, the longer the computation times. The stopping condition proposed for the FSHOPH heuristic has arisen from an experiment

Number of jobs	Number of machines	Nbmax
$n \leq 30$	$m \leq 20$	n
$31 \leq n \leq 49$	$m < 15$	n
$50 \leq n \leq 65$	$m < 10$	n
$66 \leq n \leq 100$	$m < 7$	65
$31 \leq n < 50$	$15 \leq m \leq 20$	30
$50 \leq n < 66$	$10 \leq m \leq 20$	30
$66 \leq n \leq 100$	$7 \leq m < 15$	30
$66 \leq n \leq 100$	$15 \leq m \leq 20$	10

Table 1. Selection of Nbmax for FShop.TS5

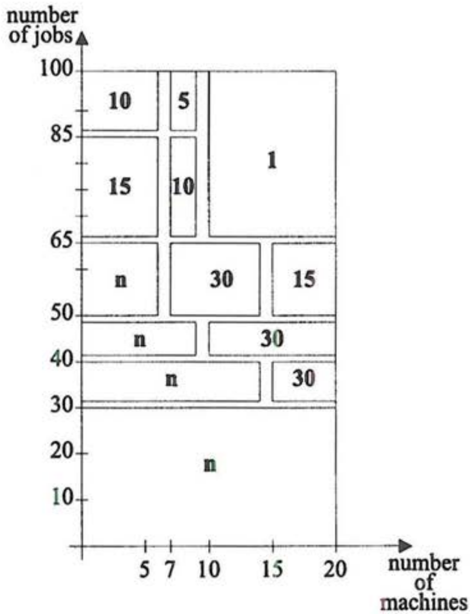


Figure 2. Nbmax parameter for FSHOPH

2.2. The SAfshopH heuristic

SAfshopH is a simulated annealing heuristic having the same initial solution as the FShop.TS5. The simulated annealing procedure we have used for the SAfshopH heuristic, in the research reported in this paper, is similar to the procedure SA(S,R) proposed by Osman and Potts (1989), namely: At each iteration the algorithm randomly selects a neighbor solution in the shift neighborhood of

The "cooling" schedule is given by $T_{k+1} = T_k/(1 + bT_k)$ where $b = (T_1 - T_K)/((K-1)T_1T_K)$, T_1 is the initial temperature, $T_K = 1$ is the final temperature, and K is the total number of iterations.

During a neighborhood search the algorithm always accepts a neighbor solution that improves the objective function value. On the other hand, a generated neighbor whose objective value is worse than that of the current solution is accepted with a probability P given by: $P = \exp(-d/T)$, where d is the increase in the objective function value, and T is the temperature parameter. T is high in the initial iterations of the search so that many "uphill moves" can be accepted, and then decreases until it reaches a low final value. In the final iterations the algorithm becomes like a descent method.

3. The hybrid TS-SA heuristic

The hybrid approach we introduce, which is denoted by HybTSSA, has two basic steps:
Step 1 — The initial sequence is the same as the one for both of the pure heuristics FShop.TS5 and SAfshopH.
Step 2 — The second step improves the initial sequence through the following procedure: to get a neighbor of a sequence S we remove a job from its position and insert it at another position, that is, we also use the shift neighborhood. At each iteration we examine a partial neighborhood $PN(S)$ of the current sequence S, which is randomly generated.

The number of neighbor-sequences of $PN(S)$ is given by:

$$NbPN(S) = p(n-1)^2 \text{ for } n < 30, \text{ and} \tag{3}$$

$$NbPN(S) = p[1741 - 900\exp(-0.04(n-30))] \text{ for } n \geq 30, \tag{4}$$

where p is an experimental parameter and n is the number of jobs to be processed.

It is worth noting that the above expressions are, respectively, the expressions (1) and (2) concerning the FShop.TS5 heuristic multiplied by the experimental parameter p.

An admissible move related to sequence S is the sequence S' that minimizes the makespan over the set $PN(S)$. If the makespan value of sequence S' is less than or equal to the makespan of S, the move is accepted. Otherwise it is accepted with probability $\exp(-\Delta/t)$, where Δ is the increase in the objective function value, and t is the current temperature which decreases as the number of iterations increases. When a move is accepted, it remains a tabu move during 7 accepted-move iterations.

The hybrid procedure is stopped after a fixed total number of evaluated n-job sequences $TNbS(m, n)$ according to Table 2. The values for the stopping condition in Table 2 were obtained from previous simulation by using the total number of sequences that are evaluated by using the tabu search procedure FShop.TS5. The stopping condition $TNbS(m,n)$ is also used in the SAfshopH heuristic.

Number of (machines, jobs)	$TNbS(m,n)$	Number of (machines, jobs)	$TNbS(m,n)$	Number of (machines, jobs)	$TNbS(m,n)$
(4, 30)	43732	(7, 30)	59666	(10, 30)	73503
(4, 40)	77742	(7, 40)	162358	(10, 40)	125728
(4, 50)	105623	(7, 50)	165856	(10, 50)	154223
(4, 60)	133502	(7, 60)	178017	(10, 60)	166257
(4, 70)	137452	(7, 70)	135330	(10, 70)	173862
(4, 80)	166050	(7, 80)	107244	(10, 80)	189378
(4, 90)	155455	(7, 90)	128733	(10, 90)	189406
(4, 100)	194680	(7, 100)	108136	(10, 100)	190884

Table 2. Stopping condition

4. Computational experience

The FShop.TS5, SAfshopH and HybTSSA heuristics have been evaluated on a total of 480 problems with the number of machines $m \in \{4, 7, 10\}$ and the number of jobs $n \in \{30, 40, 50, 60, 70, 80, 90, 100\}$. Each of the $m \times n$ combinations was replicated 20 times. The operation processing times were randomly generated integers uniformly distributed over the interval $[1, 100]$. In the computational tests the heuristics were coded in Turbo-Pascal and have been run on a microcomputer Pentium II 200 Mhz. The coefficient p from expressions (3) and (4), and the initial temperature $T1$ for the simulated annealing part of the hybrid procedure are the sensitive experimental parameters of the HybTSSA heuristic. In order to find the best values for these parameters we have tested all problems for $p \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $T1 \in \{20, 30, 45, 60, 70\}$. The final temperature is taken equal to 1.

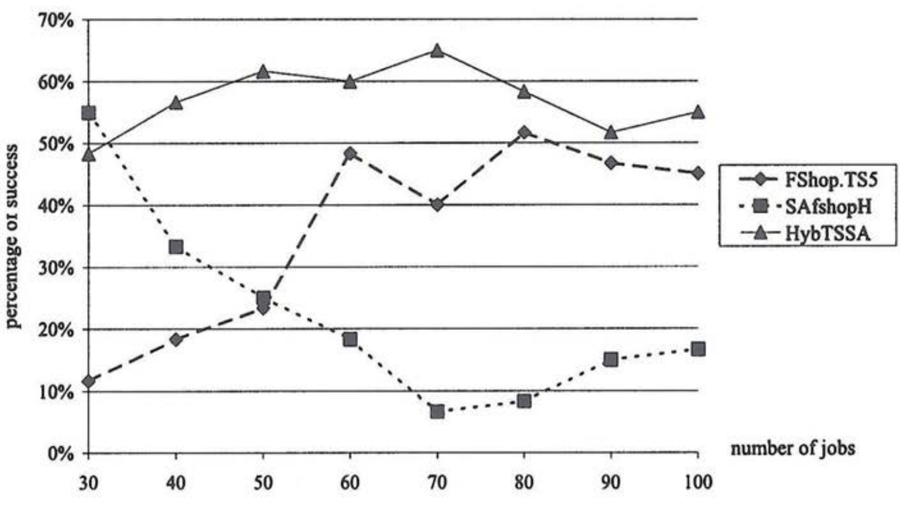
The best values for the pair $(p, T1)$ we have found so far are $(0.6, 30)$ and $(0.8, 30)$.

The principal results from the computational experience are the following:

Tables 3 and 4 summarize the comparison of the heuristics showing the percentage of success of each heuristic method. This percentage is defined as the total number of times the heuristic obtains the best makespan divided by the number of solved problems. Obviously, when two or more heuristics obtain the best makespan for the same problem all of them reach success, and consequently their percentages of success are simultaneously improved. The results from

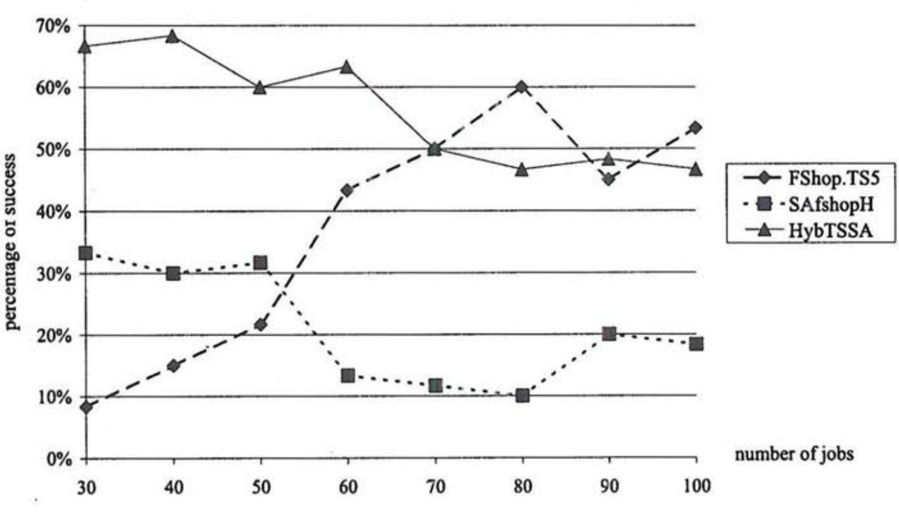
Number of jobs	FShop.TS5 (%)	SAfshopH (%)	HybTSSA (%)
30	11.7	***55.0***	48.3
40	18.3	33.3	***56.7***
50	23.3	25.0	***61.7***
60	48.3	18.3	***60.0***
70	40.0	6.7	***65.0***
80	51.7	8.3	***58.3***
90	46.7	15.0	***51.7***
100	45.0	16.7	***55.0***

Table 3. Percentage of success ($p = 0.6$, $T1 = 30$)



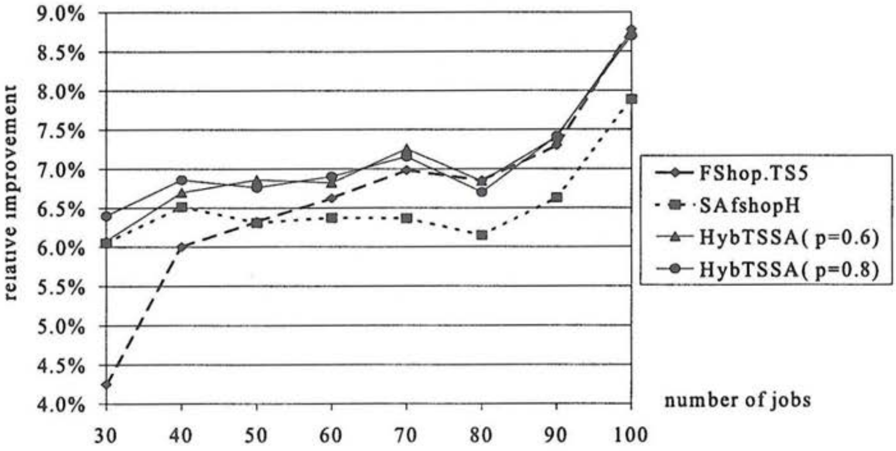
Number of jobs	FShop.TS5 (%)	SAfshopH (%)	HybTSSA (%)
30	8.3	33.3	***66.7***
40	15.0	30.0	***68.3***
50	21.7	31.7	***60.0***
60	43.3	13.3	***63.3***
70	***50.0***	11.7	***50.0***
80	***60.0***	10.0	46.7
90	45.0	20.0	***48.3***
100	***53.3***	18.3	46.7

Table 4. Percentage of success ($p = 0.8$, $T1 = 30$)



Number of jobs	FShop.TS5 (%)	SAfshopH (%)	HybTSSA ($p = 0.6$) (%)	HybTSSA ($p = 0.8$) (%)
30	4.25	6.05	6.08	***6.40***
40	6.00	6.52	6.70	***6.85***
50	6.33	6.31	***6.86***	6.76
60	6.63	6.38	6.82	***6.90***
70	6.98	6.37	***7.26***	7.16
80	***6.84***	6.15	***6.84***	6.70
90	7.30	6.63	7.40	***7.41***
100	***8.78***	7.88	8.75	8.70

Table 5. Relative improvement ($p = 0.6$ and 0.8, $T1 = 30$)



The percentages of success presented in Tables 3 and 4 are average values concerning the set $\{4, 7, 10\}$ of machines. For problem instances with a fixed number of jobs, different results can be obtained according to the number of machines. However, in the computational experience we have carried out such results do not reach significant differences from the average over the set of machines.

Table 5 shows the average relative improvement in percent, according to the number of jobs (average values over the set $\{4, 7, 10\}$ of machines). These results are also illustrated in Fig. 5. We define the relative improvement (RI) by:

$RI = (M1 - M)/M1$ where

$M1$ = makespan of the initial solution, and

M = makespan of the best sequence that has been found by the heuristic.

It is worth remembering that all heuristics start from the same initial solution. In the computational simulation the average CPU times have been ranged from 2 to 176 seconds.

5. Final remarks

The results from Table 3 show that the hybrid TS-SA heuristic with $(p, T1) = (0.6, 30)$ outperforms both FShop.TS5 and SAfshopH heuristics for all problems having the number of jobs $n \geq 40$. However, for the 30-job problems the SAfshopH has been the best. Table 4 shows that the hybrid TS-SA heuristic with $(p, T1) = (0.8, 30)$ outperforms both FShop.TS5 and SAfshopH heuristics for problems having the number of jobs up to 60. For larger problems it shares the best performances with the FShop.TS5 heuristic.

The above results would suggest the adoption of an "adaptive" hybrid TS-SA heuristic whose basic feature would be to assume the best setting for the parameter p according to the number of jobs to be scheduled. The transition n-value could be 60. Therefore, the parameter p is taken equal to 0.8 for $n \leq 60$ and $p = 0.6$ for larger problems. The results from Table 5, concerning the relative improvements on the initial solution, are in accordance with the suggested adaptive hybrid heuristic.

These are partial results since we are testing the hybrid procedure for other sets concerning the experimental parameters p and $T1$. Anyway, our expectation about hybrid metaheuristics for minimal makespan flowshop sequencing seems to be correct, that is, a hybrid procedure can be designed in order to draw upon the advantageous features of pure metaheuristics.

Acknowledgements. The authors thank the referees for their valuable comments on this paper. The research reported in this paper was partially supported by a grant from the "Conselho Nacional de Desenvolvimento Científico

References

DÍAZ, B.A. (1996) An SA/TS Mixture Algorithm for the Scheduling Tardiness Problem. *European Journal of Operational Research*, **88**, 516-524.

ISHIBUCHI, H., MISAKI, S. and TANAKA, H. (1995) Modified Simulated Annealing Algorithms for the Flow Shop Sequencing Problem. *European Journal of Operational Research*, **81**, 388-398.

MOCCELLIN, J.V. (1994) Comparison of neighbourhood search heuristics for the flow shop sequencing problem. *Fourth International Workshop on Project Management and Scheduling*, July 1994, Leuven-Belgium, Proceedings, 228-231.

MOCCELLIN, J.V. (1995) A New Heuristic Method for the Permutation Flow Shop Scheduling Problem. *Journal of the Operational Research Society*, **46**, 883-886.

MOCCELLIN, J.V. and NAGANO, M.S. (1998) Evaluating the Performance of Tabu Search Procedures for Flow Shop Sequencing. *Journal of the Operational Research Society*, **49**, 1296-1302.

MURATA, T., ISHIBUCHI, H. and TANAKA, H. (1996) Genetic Algorithms for Flowshop Scheduling Problems. *Computers & Industrial Engineering*, **30**, 1061-1071.

NOWICKI, E. and SMUTNICKI, C. (1996) A Fast Tabu Search Algorithm for the Permutation Flow-shop Problem. *European Journal of Operational Research*, **91**, 160-175.

OSMAN, I.H. and POTTS, C.N. (1989) Simulated Annealing for Permutation Flow-Shop Scheduling. *Omega*, **17**, 551-557.

PARK, M.-W. and KIM, Y.-D. (1998) A Systematic Procedure for Setting Parameters in Simulated Annealing Algorithms. *Computers & Operations Research*, **25**, 207-217.

REEVES, C.R. (1993) Improving the Efficiency of Tabu Search for Machine Sequencing Problems. *Journal of the Operational Research Society*, **44**, 375-382.

WIDMER, M. and HERTZ, A. (1989) A New Heuristic Method for the Flow Shop Sequencing Problem. *European Journal of Operational Research*, **41**, 186-193.

ZEGORDI, S.H., ITOH, K. and ENKAWA, T. (1995) Minimizing Makespan for Flow Shop Scheduling by Combining Simulated Annealing with Sequencing Knowledge. *European Journal of Operational Research*, **85**, 515-531.