



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Resolution of Scheduling Problem of the Production Systems by Sequential and Parallel Tabu Search

^{1,2}K. Belkadi, ²M. Gourgand and ¹M. Benyettou

¹Département d'Informatique, Université Mohamed Boudiaf, USTO,
BP 1505 Oran M'Naouer, 31000 Oran, Algérie

²Université Blaise Pascal, Clermont Ferrand, Campus of Cézeaux, 63177 Aubière Cedex, France

Abstract: The objective of this study is the resolution of a combinative optimization problem. This problem is scheduling in the production systems of the type Hybrid Flow Shop (HFS). This resolution is made with using Sequential and Parallel Tabu Search (TS). In this study we propose an implementation of a parallel algorithm of Tabu Search based on Multiples research with Synchronous Interaction (TS_MSI). The basic idea of this method is that several processes (threads) carry out each one a TS and they communicate between them at defined points of synchronization. Numerical experiments are carried out to make a comparison in term of quality of solution, between the sequential TS and parallel TS. The results obtained show well that this strategy of parallelization (TS_MSI) improves the results compared to the sequential TS.

Key words: Parallel Tabu search, production system, hybrid flow shop, optimisation, scheduling

INTRODUCTION

The scheduling problems in the production system are generally problems which make party of the class of the problems known as Np-Complete. The strategies of resolution (meta-heuristics) are applied more and more to this type of problem. These meta-heuristics and among them Tabu Search (TS), represents an effective strategy for the search of the approximate solutions of the combinative optimization problems. But the disadvantage of this method more evoked is the computing time and the quantity of memory used and moreover they are closely related on the size of the problem and obtaining a certain quality of solution. With the increase of the parallel machines, powerful workstations and great communication networks, the parallelization of meta-heuristic makes it possible to accelerate the search of the approximate solutions (reduction of the execution time) and also to improve quality of the solution obtained. So these algorithms become interesting to parallel.

For that, in the case of the parallelization of the TS, we adapted and implemented Parallel Tabu Search based on Multiples research with Synchronous Interaction (TS_MSI). In this strategy, each process executes independent sequential TS and communicates with the other processes at defined points of synchronization. Thus they exchange information. Exchanged information

is in the majority of the cases the good solutions found since the last step of synchronization. The different TS choose a new solution to start again their research after synchronization. Four possible implementation plans are instancied according to the unicity of the initial solution and the unicity of the strategy of research. Synchronization depends on a frequency fixed a priori and the model of programming which supports this execution is of type Master/Slaves (Crainic *et al.*, 1995; Porto and Ribeiro, 1996).

MATERIALS AND METHODS

Hybrid flow shop problem: In the literature, the notion of hybrid flow shop has emerged in the 70s. An organization in Hybrid Flow Shop (HFS) is consisted of a set of stages, where each stage is composed of one or more parallel machines. The various jobs visit the stages in the same order. On each stage, a job is treated by only one machine. Between each stage, the jobs can wait or not in limited or unlimited stocks (Billaut *et al.*, 2000; Vignier *et al.*, 1999) (Fig. 1).

The treated problem is the scheduling of N jobs in entry of the system and the assignment of these jobs on the machines. We research a scheduling and an assignment which optimize a criterion of performance. The criterion to be optimized is C_{max} (the time of total completion of the jobs (works)).

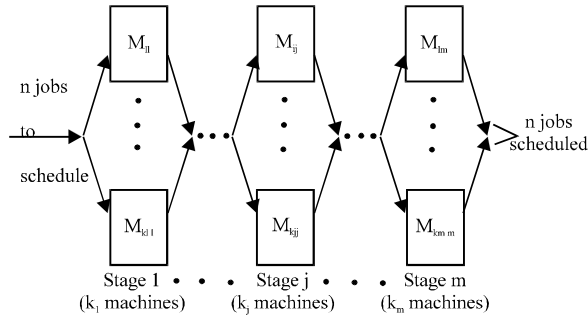


Fig. 1: Hybrid flow-shop (HFS)

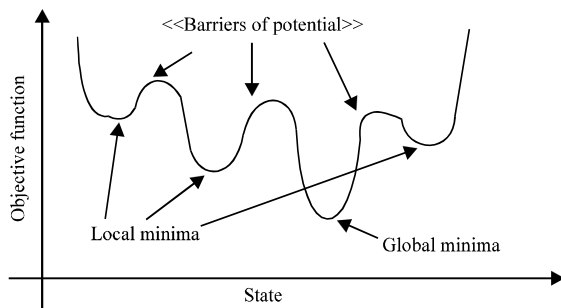


Fig. 2: Objective function

Sequential tabu search: The method of Tabu Search (TS) is high level heuristics or metaheuristics which was introduced by Glover (1998). He details the various algorithms containing tabu lists. Since, many contribution were made to this method, in particular where Widmer (1991) applied it to the optimization of the flexible manufacturing systems.

The TS is based on local research but with a probability to continue research beyond the local optima. This is done by accepting a solution of quality worse than the current solution. What makes it possible to be extracted from the local minima. At each stage, we choose the best solution among the whole of the solutions which belong in the neighbouring of the current solution, even if this solution is less interesting than the preceding one (Alg. 1). The fact of using local research with the possibility of selecting a neighbour (X') which decreases the value of the objective function, can involve cycles (reconsideration of the preceding solutions) in the exploration of research (Fig. 2). To avoid this problem, we avoid in the TS revisiting the solutions already tested. We say: these solutions are Tabu for a certain time. The prohibited movements are put in a list called Tabu List and are prohibited with the current iteration of the algorithm. These movements remain Tabu for one duration equal to the size of the Tabu List (Talbi, 1999).

Algorithm 1: Algorithm of the sequential RT

```

Function Sequential Tabu (X_init)
- Initialize the Tabu list T =  $\phi$ ;
- Generate an initial solution: X_init;
- X_current  $\leftarrow$  X_init;
- X_best  $\leftarrow$  X_init;
- Iter  $\leftarrow$  0;
Begin
While((Iter < max_iter) ou (S(X_current)/T =  $\phi$ )) Do
// S(X_current): set of acceptable
// movements of X_current
Iter  $\leftarrow$  Iter + 1;
Best_Move  $\leftarrow$  0;
For (any movement candidate) Do
Choose sk  $\in$  S(X_current)/T such as:
sk(X_current)  $\leftarrow$  optimum (s(X_current) :
s  $\in$  S(X_current)/T);
// sk is an acceptable movement
// candidate (not belonging to the tabu list)
X_next  $\leftarrow$  sk(X_current);
// X_current is reached by applying the
// acceptable movement sk to
// the current solution
move_value  $\leftarrow$  H(X_next) - H(X_current);
If (move_value < best_value)
Then Best_move  $\leftarrow$  move_value ;
X'  $\leftarrow$  X_next;
Endif
EndDo
Update of the short-term memory function
(Update of tabu list);
If (H(X') < H(X_best))
Then X_best  $\leftarrow$  X';
X_current  $\leftarrow$  X';
Endif
EndWhile
Return X_best;
End
    
```

The sequential TS is based essentially on the fixing of its parameters which are:

Size of the Tabu List (the period during which a movement remains tabu): It is a parameter which is fixed at the starting of the execution and which remains unchanged.

Size of the candidates movements list: The strategy used is the Strategy of Better Improvement (SBI), for that a candidate list was used containing the best evaluated movements. Because the system of neighbouring used for the HFS is very large, it is very difficult to completely evaluate all the neighbours of a solution at each iteration. For that subset or sample of these neighbours generated by randomise was considered. The first solutions of this subset are put in the list. This list contains the following informations: the solution (scheduling + assignment), the cost of the solution and attributes of movements used to succeed to this solution.

Criterion of aspiration: It cancels the tabu state of a movement if this movement is well adapted. The level of

aspiration is reached when the cost of the solution generated by this movement is lower than the value of aspiration of this movement.

Criterion of intensification-diversification: To reinforce (to intensify) research in the good areas, the choice of the best improvement was selected where at each iteration a list of good movements is created. To diversify research, after a number of iterations without improvements, a jump is carried out (Kangourou jump) (Fleury, 1993).

The parallel Tabu search: Crainic (1997 and 2002) presented a taxonomy which summarizes the strategies of parallelization of the TS. This taxonomy has the advantage to being more general. It makes it possible to be applied to any problem of combinative optimization. According to this taxonomy, a strategy of parallelization is defined by three large axes: the control of execution which can be centralized (1-control) or be distributed (P-controls), the type of control which can be synchronous or asynchronous and strategy of research which is a possible configuration of the operating parameters.

We distinguish also the existence of two approaches to parallelize research according to the number of trajectories implied in parallel research. The parallelization of the TS can be with single trajectory (Single Walk) (Blesa *et al.*, 2001; Porto and Ribeiro, 1996) or with multiple trajectories (Walks Multiple) (Van and Martin, 2001; Taillard, 1998; Crainic *et al.*, 1995; Porto and Ribeiro, 1996; Glover *et al.*, 1998; Cavalcante *et al.*, 2001; Talbi, 1999).

In the implementation of the parallel TS for the scheduling problem of the HFS (Aribi and Belkadi, 2003), we adapted and implemented the parallel algorithm of Tabu search based on Multiples research with Synchronous Interactions (TS_MSI) of the taxonomy of Crainic (1997). In this strategy, each process executes independent sequential TS and communicates with the other processes at defined points of synchronization. The various TS choose a new solution to start again their research after synchronization. We have instanced four possible implementation plans according to if the initial solution is single or multiple and if the strategy of research is single or multiple. Synchronization depends on a frequency to fix a priori and the model of programming which supports this execution is centralized on Master/Slaves.

Description of the TS_MSI algorithm: The algorithm of the TS_MSI follows a model of programming in Master/Slaves. The Master initialises the different TS

which are launched by the slaves. At each point of synchronization, the slaves turn over their final state to the thread Master which chooses a solution among the turned over solutions and starts again research of the slaves until reaching a maximum number of synchronizations (Alg. 2). At each launch, the slaves preserve the same strategies of research and empty all the Tabu lists (Alg. 3.).

Algorithm 2: Algorithm of master-TS_MSI

```

Begin
- Generate Sol_Init ;
- Sol_Init_1 ← Sol_Init ; //initial solution slave 1
For any slave_j : j = 2, 3, ..., P Do
  If MP // Multiple Point
    Then Sol_Init_j ← Neighbor (Sol_Init)
    Else Sol_Init_j ← Sol_Init
  EndIf
EndDo
If MS // Multiple Strategy
  Then Generate P strategies of research
  Else Generate only one strategy of research
  EndIf
- Assign the strategies to the slaves;
- Calculate Best_Global_Sol ;
- Calculate size of the interval of synchronization:
  Nbr_iter_Synch ;
- Calculate the number of synchronization:
  Nbr_Synch ;
- Nbr ← 0 ;
- Launch the P Slaves;
Repeat
  WaitFor (End_Of_Cycle j) j = 1, 2, ..., P;
  // to await the end of all the slaves
  Suspend (Pj) j = 1, 2, ..., P ;
  //to suspend the execution of all the slaves
  Sol_current ← choose_Sol (Sol_current j)
  for j = 1, 2, 3, ..., P ; //strategy of choice
  If H(Best_Global_Sol) > H(Sol_current)
  Then Best_Global_Sol ← Sol_current
  EndIf
  Sol_current_1 ← Sol_current;
  For any slave_j : j = 2, 3, ..., P Do
  If MP // Multiple Point
  Then
    Sol_current_j ← neighbor (Sol_current)
  Else Sol_current_j ← Sol_current
  EndIf
  EndDo
  begin_again (Pj) j = 1, 2, 3, ..., P
  Nbr ← Nbr + 1 ;
  Until (Nbr > Nbr_Synch)
End

```

Algorithm 3: Algorithm of the slave-TS_MSI

```

Begin
  End_Of_Cycle ← False;
  Iter ← 0 ;
  Repeat
  Call Sequential_Tabu (Sol_current_j);
  Iter ← Iter + 1 ;
  Until (Iter > Nbr_Iter_Synch);
  End_Of_Cycle ← True;
End

```

Parameters of the TS_MSI: The parameters which influence the algorithm of the TS_MSI are:

Strategy of research used: The TS launched by Threads can share or not the initial solution and can also share or not the strategy of research; from where four classes of execution which can be instanced: SPSS (Single Point Single Strategy), MPSS (Multiple Point Single Strategy), SPMS (Single Point Multiple Strategy) and MPMS (Multiple Point Multiple Strategy).

Frequency of synchronization: It determines the number of points of synchronization to carry out all along research. It also determines the number of iteration to carry out in a TS between two points of synchronization.

Number of Multiple Tabu search (a number of slaves or Threads): This number determines the number of research which are launched in parallel.

Choice strategy of initial solution after synchronization: At each point of synchronization, the Master chooses a solution among the solutions turned over by the slaves. This choice is not inevitably the best solution because this last can create a deteriorating determinism. Three choice strategies of solution are studied: Best, Mean and Prob. In Best, we choose the best solution among the local current solutions. In Mean, we choose the current solution which has the cost value near the average of all the cost values of the current solutions. In Prob, the choice of the solution is a probabilist and each solution has a certain probability to being chosen and it depends on its cost value compared to the sum of cost values of the other solutions.

RESULTS AND DISCUSSION

The Tabu Search (TS) was tested on Hybrid Flow Shop (HFS) with two stages and with three machines in the first stage and two machines in the second stage (Fig. 3). The type of HFS is a HFS with a stock at the entry of the first stage and between the stages. The number of jobs is $N = 5, 10, 15$ and 20 . The treated problem is the scheduling of N jobs in entry of the system and the assignment of these jobs on the machines. We research a scheduling and an assignment which optimize a criterion of performance. The criterion to be minimized is the C_{max} (the time of total completion of the jobs (work)). By using the notation of Vignier *et al.* (1999), we can represent it by $FH2 (P3, P2)||C_{max}$.

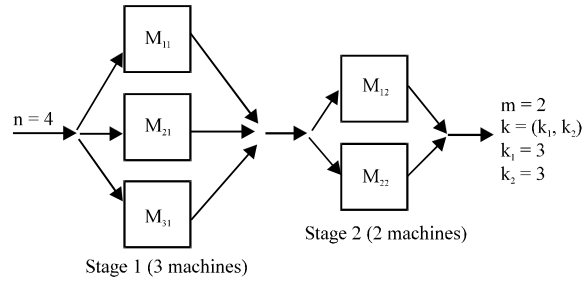


Fig. 3: $FH2 (P3, P2)||C_{max}$

The sequential TS applied to HFS (Sahraoui *et al.*, 2002; Belkadi *et al.*, 2003) gave good results with the size of the Tabu List fixed at 7, the size of the list of the candidates at 500 and the number of iteration at 1000. The initial solution is generated by randomize.

The obtained results are shown in Table 1 (values of cost Average C_{max} and execution time) (average of 10 tests of the C_{max} value). These values go to be used as reference for a comparison between the sequential TS and the parallel TS.

Comparison between the sequential TS (TS_Sequ) and the parallel TS (TS_MSI): The algorithm of TS_MSI was implemented on an architecture Bi-processor (Intel Literatur Center, 2000) and it was tested on Hybrid Flow Shop of the same type as that studied for sequential TS (Aribi and Belkadi, 2003). The strategy of selected parallelization is with centralized synchronization (Master/Slaves). In the case of the parallel TS (TS_MSI), a study of choice of parameters was made and that made it possible to fix their values. The values of these parameters are as follows:

- The frequency of synchronization is fixed at 20%,
- The number of slaves at 5,
- The strategy of choice is Mean
- And the schema of Tabu parallelism is MPMS.

These values will be used as reference for the comparison carried out in term of quality of solution between the sequential TS and the parallel TS.

Figure 4 represents this comparison of quality of solution and it comprises two sets of data: the series concerning TSSequ and that concerning the TS_MSI

Table 1: Average C_{max} and average CPUs for sequential Tabu Search		
No. of jobs	Average (C_{max})	Average CPUs (sec)
5	145.0	1
10	300.5	6
15	417.9	24
20	525.5	78

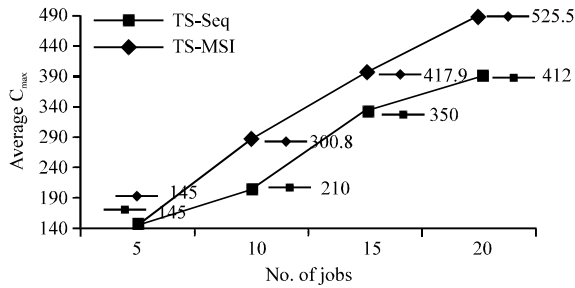


Fig. 4: Comparison between the TS_Sequ and the TS_MSI for N = 5, 10, 15 and 20

on the problem FH2 (P3, P2)||C_{max} for the number of jobs (N = 5, 10, 15 and 20).

According to the graphs of the Fig. 4 we notice well the improvement which the strategy of parallelization of Tabu search based on Multiples research with Synchronous Interaction (TS_MSI) compared to the sequential TS. This improvement remained constant with the increase in the complexity of the studied HFS (increase in the number of jobs N = 5, 10, 15 and 20).

We notice that several Tabu search carried out in parallel and synchronize with themselves to guide their research (TS_MSI) is a powerful strategy of parallelism, in particular when various Tabu search follow different strategies of research (MS: Multi Strategy) and start their research in dispersed points (MP: Multi point) of the space of research.

CONCLUSIONS

The objective of this study was the resolution of a combinative problem of optimization by using a resolution method which is Taboo Search (TS). This method is iterative metaheuristic described as local research in the broad sense. The goal is to parallel this method to improve the results (C_{max}) by using the TS_MSI. According to the results obtained, we notice that the fact of using and applying parallelism to the Tabu resolution method (TS_MSI) enabled us to improve the results of the method by giving better results that the sequential TS and that for an equivalent computing time. The required result is a good scheduling at the entry of the Hybrid Flow Shop (HFS) which optimizes (minimizes) the C_{max} (makespan). The results obtained by TS_MSI are an improvement of the results obtained previously by the TS sequential.

The results obtained can be obtained in smaller CPUs times if the implementation is done on a multiprocessor architecture instead of a biprocessor. The next work concerns an implementation of this

method of parallelization on an architecture of the type Grid computing and with an asynchronous mode of communication instead of a synchronous mode as it is done in this study. Among work under development, we are making a comparative study with other methods of optimization (GA, SA, PSO...) and to compare the results.

The originality of the study is the adaptation and the implementation of the TS_MSI to solve the scheduling problem in the production systems of the type hybrid Flow shop. The results obtained by TS_MSI compared to TS sequential are promising.

REFERENCES

Aribi, A. and K. Belkadi, 2003. Parallel computing and meta-heuristics. The 2nd International Workshop on Advanced Computation for Engineering Application ACEA'2003, Electronic Research Institute (ERI), Cairo, Egypt.

Belkadi, K., M. Sahraoui and M. Gourgand, 2003. Méthode de Recherche Tabou Appliquée aux Systèmes de Type Flow Shop Hybride. Conférence Internationale sur la Productique (CIP'2003), Alger, Algérie.

Billaut, J.C., A. Vignier, C. Proust and M.C. Portmann, 2000. A Survey of Hybrid Flow Shop Problems. Ordonnancement Déterministe pour l'Informatique et la Production (ODIP), Ecole d'été, Aussois, France.

Blesa, M.J., L. Hernández and F. Xhafa, 2001. Parallel skeletons for Tabu search method. 8th International Conference on Parallel and Distributed Systems, Kyongju City, Korea.

Cavalcante, C.C.B., V.C. Cavalcante, C.C. Ribeiro and C.C. Souza, 2001. Parallel Cooperative Approches for the Labor Constrained Scheduling Problem. in Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, pp: 201-225.

Crainic, T.G., 1997. Towards a Taxonomy of Parallel Tabu Search Heuristics. Techrep, Centre de Recherche sur le Transport, Université de Montréal, Canada.

Crainic, T.G., 2002. Parallel Strategies for Meta-heuristics. Publications CRT 02-26, Université de Montréal, Canada.

Fleury, G., 1993. Méthodes Stochastiques and Déterministes pour les Problèmes NP-Difficiles. Thèse de Doctorat, Université Blaise Pascal, Clermont Ferrand, France.

Glover, F., G.A. Kochenberger and B. Alidaee, 1998. Adaptive Memory Tabu Search for Binary Quadratic Programs. Manage. Sci., 44: 336-345.

- Intel Literature Center, 2000. Intel Server Board For Pentium III Processors. 2000. Intl. Corporation, Number 1-800-548-7725.
- Porto, S.C. and C.C. Ribeiro, 1996. A Case Study of Parallel Synchronouse Implementation for Tabu Search Based on Neighborhood Decomposition. *Investigacion Operativa*, 5: 233-259.
- Sahraoui, M., K. Belkadi and M. Gourgand, 2002. Etude Comparative de Trois Méthodes de Résolution pour les Flow Shop Hybrides. Actes des premières Journées d'Informatique pour l'Entreprise (JIE'02), Université Saad Dahleb Blida, pp: 242- 251.
- Taillard, E., 1998. Parallel Tabu Search Technics for the Job Shop Scheduling Problem. *ORSA J. Computing*, 6: 108-116.
- Talbi, E.G., 1999. A Parallel Adaptative Tabu Search Approach. LIFL, URA-369 CNRS, Université de Lille.
- Van, D. and L. Martin, 2001. Strategies for the Parallel Implementation of Metaheuristics. Laboratoire PRISM-CNRS et Department of Computer Science, Catholic University of Rio De Janeiro.
- Vignier, A., J.C. Billaut and C. Proust, 1999. Les Problèmes d'Ordonnement de Type Flow-Shop Hybride : état de l'art. *RAIRO, Recherche Opérationnelle*, 33: 117-183.
- Widmer, M., 1991. Modèles mathématiques pour un gestion efficace des ateliers flexibles. Presses Polytechniques et Universitaires Romandes.