

An Integrated ACO Approach for the Joint Production and Preventive Maintenance Scheduling Problem in the Flowshop Sequencing Problem

¹Fatima Benbouzid-Sitayeb, Ismaïl Ammi

Laboratoire des Méthodes de Conception de Systèmes (LMCS). BP 68M Oued Smar 16270 Algiers (Algeria)

E-mail. f_sitayeb@ini.dz, ammi_smail@yahoo.fr

²Christophe Varnier, Noureddine Zerhouni

Laboratoire d'Automatique de Besançon (LAB). 25, rue A. Savary 25000 Besançon (France)

E-mail. cvarnier@ens2m.fr, zerhouni@ens2m.fr

Abstract- In this paper, an integrated ACO approach to solve joint production and preventive maintenance scheduling problem in permutation flowshops is considered. A newly developed ant-colony algorithm is proposed and analyzed for solving this problem, based on a common representation of production and maintenance data, to obtain a joint schedule that is, subsequently, improved by a new local search procedure. The goal is to optimize a common objective function which takes into account both maintenance and production criteria. We compare the results obtained with our algorithm to those of an integrated genetic algorithm developed in previous works. The results and experiments carried out indicate that the proposed ant-colony algorithm provide very effective solutions for this problem.

Key words— ACO, Production, Preventive maintenance, Integrated approach, Joint scheduling, Flowshop.

I. INTRODUCTION

Production scheduling and preventive maintenance planning decisions are inter-dependent but most often made independently. Both activities conflict since preventive maintenance activities consume potential production time, but delaying preventive maintenance because of production demands may increase the probability of machine failure, this interdependency seems to be overlooked in the literature.

One finds in the literature two joint scheduling strategies which aim is to solve conflicts between production scheduling and preventive maintenance planning [1]. The sequential strategy consists of two steps: First the scheduling of the production jobs then the insertion of the maintenance tasks, taking the production schedule as a strong constraint. The integrated one consists of simultaneously scheduling both maintenance and production activities based on a common representation of these two activities.

As it has been mentioned, the interaction between production and preventive maintenance, particularly their joint scheduling, is relatively little studied and rather recently in the literature. Little work has been carried out in which preventive maintenance scheduling and flowshop scheduling are jointly considered [2, 3, 4, 5].

Since the last decade, attempts are being made to solve combinatorial optimization problems using Ant Colony Optimization (ACO) algorithms. ACO is a metaheuristic

allowing finding the best solution of hard-to-solve optimization problems [6, 7]. The aim of ACO algorithm is to find a path (a sequence) of minimal cost while respecting the constraints [8].

ACO has been applied successfully to solve scheduling problems. The problem dealt with is flowshop scheduling problem with the objective of minimizing the makespan. Stützel [9] compared ACO algorithm to some basic heuristics for the Flow Shop Problem (FSP) showing that this approach is very promising for the FSP. Rajendran and Ziegler [10] propose two ant-colony optimization algorithms and compared them to heuristic solutions given by Taillard [11]. The comparison shows that the two proposed ant-colony algorithms perform better. Ying and Liao [12] applied the ACO algorithm to permutation FSP and compared its performance with other meta-heuristics such as genetic algorithm, simulated annealing, and neighbourhood search. Their computational results showed that ACO was a more effective metaheuristic for the $n/m/P/C_{max}$ permutation flowshop scheduling problem. Hu and al. [13] proposed an ACO algorithm to solve flowshop rescheduling problem. Mutation operations are employed to enhance the ACO performance avoiding the algorithm convergence to local optimum. Numerical experiments showed that the proposed algorithm has high computation efficiency.

The term "maintenance" is nonexistent in the previously cited literature. As in industry, production scheduling and preventive maintenance planning are typically treated independently in the production systems and also in operations research literature.

Purpose of this work is to present an integrated ACO approach to solve the joint production and preventive maintenance scheduling problem using ACO algorithm. The proposed integrated ant-colony algorithm, called INTACO, is based on a common representation of production and maintenance data. The aim is to optimize a common objective function which takes into account both maintenance and production criteria. The outline of this paper is as follows: section 2 presents the context of the study: production and maintenance data and the objective functions to optimize, section 3 introduces the proposed algorithm, discusses some details of the new local search procedure used and presents the

most important parameter settings. Afterwards, in section 4, convergence properties of the algorithm are investigated so as to obtain a tuned set of parameters. Then we present the obtained simulation results with some benchmark problems which we compare to another metaheuristic developed in previous work. Finally, in section 5, some conclusions are made.

II. MODELS AND PROBLEM STATEMENT

In the following, we will first present the production and the maintenance data. Then the common objective function to optimize

One of the most frequent production scheduling problems is the FSP. This problem can be stated as follows: each of n jobs $1, \dots, n$ has to be processed on m machines $1, \dots, m$ in that order. The processing time of job i on machine j is p_{ij} . The processing times are fixed, nonnegative, and may be 0 if a job is not processed on some machine. Further assumptions are that each job can be processed on only one machine at a time, the operations are not preemptable, the jobs are available for processing at time zero and setup times are sequence independent. Here we consider the Permutation Flow Shop Problem (PFSP) i.e., the same job order is chosen on every machine. The objective then is to find a sequence, i.e., a permutation of the numbers $1, \dots, n$ that minimizes the completion time C_{\max} (makespan) of the last job. Following the Pinedo's notation [14] this problem can be denoted by $F/prmu/C_{\max}$. The PFSP is an NP-Hard problem [15].

The preventive maintenance [16] can be stated as follows: each machine is maintained at fixed time intervals previously established. The preventive maintenance tasks are periodic interventions occurring every T periods and each occurrence depends on the one preceding it on the same machine. A maintenance task consists of elementary operations which processing time p' is evaluated with more or less certainty. Moreover, the periodicity T of these tasks can vary in a tolerance interval noted $[T_{\min}, T_{\max}]$. This interval gives some flexibility to plan maintenance tasks while respecting the production constraint, disturbing the least possible the production schedule, and respecting the maintenance equipment periodicity.

A machine j can be subject to several different maintenance tasks which will be repeated periodically. Let be:

- M_{ij} : the maintenance task i on the machine j .
- T_{ij} : periodicity of the maintenance task M_{ij} .
- $T_{\min ij}$: earliest time separating two consecutives occurrences of the maintenance task M_{ij} ;
- $T_{\max ij}$: latest time separating two consecutives occurrences of the maintenance task M_{ij} ;
- p'_{ij} : processing time of the task M_{ij} . It is supposed to be known and constant.

The earliness and the tardiness of the k^{th} occurrence of the maintenance task M_{ij} is computed as follows:

- t'_{ijk} : execution time of the k^{th} occurrence of the

maintenance task M_{ij} .

- E'_{ijk} : Earliness of the k^{th} occurrence of the maintenance task M_{ij} . $E'_{ijk} = \max(0, t'_{ijk} + p'_{ij} + T_{\min ij} - t'_{ijk} + 1)$
- L'_{ijk} : Tardiness of the k^{th} occurrence of the maintenance task M_{ij} . $L'_{ijk} = \max(0, t'_{ijk+1} - t'_{ijk} - p'_{ij} - T_{\max ij})$

The objective of optimization must be a compromise between the target objective maintenance and production functions.

The production objective function is the completion time of the last job on the last machine (C_{\max}). One will note f_1 this function: $f_1 = C_{\max} = \text{Max}(c_{ij})^1$ (1)

The maintenance objective function respects the maintenance periods which influence the constraints of the production system. One will note f_2 this function:

$$f_2 = \sum_{j=1}^m \sum_{k=1}^{\text{Max}j} E'_{jk} + L'_{jk}{}^2 \quad (2)$$

To optimize the two criteria, we taken into account the following common objective function:

$$f_{obj} = \alpha f_1 + \beta f_2 \quad (3)$$

(α, β) are weights which will measure the respective contributions of production and maintenance in the common objective function. They can depend on the number of tasks, processing time or problem size. They are independent and non complementary.

III. DEVELOPPEMENT OF A NEW INTEGRATED ACO ALGORITHM (INTACO)

We now present the development of the new ant-colony algorithm, called INTACO, which extends the ideas of Ant System (AS) algorithm by Dorigo and al. [7]. INTACO incorporates simultaneously production jobs and maintenance tasks in the construction of a joint ant sequence, uses new mutation operators based on those proposed in [17, 18, 19] adapted to joint production and preventive maintenance sequence allowing the generation of the joint ant sequence neighbourhood, and a new local search technique in the AS algorithm.

In this section, we will first present the proposed representation of a joint schedule. Then we introduce the new integrated ACO algorithm (INTACO)

A. Representation of a joint schedule

We propose to code a joint schedule as a structure with two fields: the first one is a sequence S that represents the production jobs execution order. The second one is a matrix M that represents the maintenance tasks insertion sites. The element $M[i, j]$ represents the insertion of the i^{th} maintenance task of the j^{th} machine in the sequence S .

¹ Completion time of job i on machine j

² Maxj represents the effective occurrence number of the maintenance task M_j .

Example:

Production sequence S	1	9	3	8	5	6	7	4	2
Matrix M	0	1	4	6					
	1	2	5						
	0	4	7	8					

The execution of the tasks on the three machines according of the preceding example is the following:

Machine 1: $M_{01}, P_1, M_{11}, P_9, P_3, P_8, M_{21}, P_5, P_6, M_{31}, P_7, P_4, P_2$

Machine 2: $P_1, M_{02}, P_9, M_{12}, P_3, P_8, P_5, M_{22}, P_6, P_7, P_4, P_2$

Machine 3: $M_{03}, P_1, P_9, P_3, P_8, M_{13}, P_5, P_6, P_7, M_{23}, P_4, M_{33}, P_2$

B. Construction of a joint ant-sequence and its improvement by a local search scheme

We model the problem as a search for the best path in a particular graph $G \langle N, V \rangle$ where N is the set of nodes representing joint solutions, and V represents the difference between the common objective function referred to in formula 3 of two connected solutions (S_i, S_j).

Thus, the integrated strategy consists of simultaneously scheduling both maintenance and production activities. The first cycle of the algorithm commences with joint sequence generated randomly and proceeds by construction, moving from a solution to another. A number of ants, which may be thought of as independent agents, are set in motion, each ant representing a joint sequence.

Each ant i starts from a joint initial solution S_0^i which is randomly generated. At each step, a set of new joint solutions S_k^i is generated by applying mutation to the current one, which represents its neighbourhood $N(S_0^i)$. During her move, each ant will leave trail of pheromone on her path.

Each ant starts the search with a random solution. For us a solution can be complete or partial. A complete solution represents a joint production and preventive maintenance schedule. In this case, a complete initial solution is obtained after the insertion of maintenance tasks, using one of the heuristics developed by Benbouzid and al. [4], on a production schedule generated randomly.

The move from a solution to another in the neighbourhood can be done by mutation in the production sequence, the maintenance matrix, or on both at the same time. We define two types of mutation allowing generating neighbour solutions from the current one. The first one is relative to new mutation reported by Benbouzid and al. [20] and the second one is based on new ideas that are developed in the current work.

(a) Benbouzid and al. [20] developed a couple of new mutation: Random mutation and k-points vertical mutation. Random mutation consists in shifting randomly towards the left or the right one or more maintenance tasks. The principle of the k-points vertical mutation consists on generating randomly k positions in the production sequence S, thus defining k+1 groups. Then the sites of the production jobs which are in each group i are permuted with the group k+2-i. However the maintenance tasks sites inside each

group are preserved.

(b) We propose two new mutations called IIM (Integrated Insertion-Moves) and ISM (Integrated Swap-Moves). It consists in performing the classical insertion-moves and swap-moves [17, 18, 19] on the production sequence S while updating the maintenance matrix M under the changes operate on the sequence S.

Having thus generated a joint ant sequence of production jobs and maintenance tasks, some changes are applied in order to improve the solution. Local search for the FSP usually starts from some initial feasible sequence and then tries to improve it by small changes. In the case of the joint production and maintenance scheduling problem in permutation flowshop, it is to be noted that the 1-point vertical mutation suggested by Benbouzid and al. [20], has emerged as the best improvement for all benchmark problems tested.

C. Updating of trail intensities

Starting from a joint initial sequence, INTACO makes use of trail intensities in order to determine the solution to be appended next step.

At each step, a set of new solutions is generated by applying mutations on the current one, as defined above. When the ant (k) is on one solution (S_k^i), it must choose a solution S_k^j among the set of solutions of its neighbourhood $N(S_k^i)$. This choice is made on the basis of the heuristic information (η_{ij}), and the pheromone information (τ_{ij}), according to the following rule [21]:

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha * \eta_{ij}^\beta}{\sum_{l \in J}^k (\tau_{il}(t))^\alpha * \eta_{il}^\beta} & \text{If } j \in N(S_i^k) \\ 0 & \end{cases} \quad (4)$$

The heuristic information is computed using SPT (Shorting Processing Time) algorithm as follows:

$$\eta_{ij} = \frac{1}{1 + \sum_r p_{jr}} \quad (5)$$

When an ant move to a new solution, chosen in the neighbourhood of the current one, a local update of the pheromone trail is made. When all ants finished a cycle, a global update of the pheromone trail is done on the path of the best joint solution found in the cycle. The best joint sequence so far found is retained, and a new cycle is started. The pheromone update rule is applied as follows:

$$\tau_{ij}^{\text{new}} = (1-\rho) * \tau_{ij}^{\text{old}} + \rho x 1/f_{obj} \quad (6)$$

Once all ants have built their solutions, extra pheromone is added on the path taken by the ant which found the best

³ p_{jr} is the processing time of job j on machine r

⁴ f_{obj} is the objective function referred in formula 3, of the solution given by the best ant.

solution. Pheromone is added only when the common objective function f_{obj} of a solution is improved after mutations.

Since the neighbourhood of the current solution is a set of complete joint solutions generated according to the operators detailed above, the construction of the pheromone and heuristic information matrix will be done dynamically, as describe below.

Pheromone and heuristic information

Let P : dynamic pheromone matrix

H : dynamic heuristic information matrix

Begin

Create a pheromone matrix P and heuristic matrix H

For Each solution Sc

Generate the neighbourhood $N(Sc)$ of the solution Sc

$L \leftarrow N(Sc)$

For each solution $Si \in N(Sc)$

If Si exist in the pheromone matrix

Then use the corresponding pheromone and heuristic information in matrix P and H

Else /* Dynamic creation of pheromone and heuristic information in the corresponding matrix*/

Allocate un new column in P and H

$P[i,j] \leftarrow \tau_0$

$H[i,j] \leftarrow 1/f_{obj}(Si)$

End If

End For

End For

End.

IV. EXPERIMENTAL RESULTS

In this section, we present the results of a series of computational experiments conducted to test the effectiveness of the INTACO, for an integrated resolution of the joint production and preventive maintenance scheduling problem in permutation flowshops. The test problems for evaluating the proposed ant-colony algorithm are generated by following the procedure given by Taillard [11] for permutation flowshop scheduling problems.

To our knowledge, there is no instance for the preventive maintenance scheduling problem. Therefore, we developed a processor to generate benchmarks in preventive maintenance. The used parameters are: the number of machines and maintenance task parameter (T , T_{min} and T_{max}). We generated only one maintenance type per machine for each problem. Moreover, the processing time of a maintenance task is identical for all its occurrences. The objective functions taken into consideration are (§ 2): the production objective function f_1 is the minimization of makespan; the maintenance objective function f_2 is the minimization of the sum of maintenance tasks advances and delays. The contribution of production and maintenance objective functions in the common objective function f_{obj} are equal to 1 ($\alpha = 1$ and $\beta = 1$). The proposed common weighted global objective function will allow tackling the problem in a simplified way.

For evaluating the performance of the INTACO with respect to the common objective function defined in section 2, the results of a research study by Benbouzid and al. [4] are referred to. Benbouzid and al. developed a couple of new heuristic and metaheuristic methods (NEH heuristic, Taboo search and GA) considering the sequential and integrated strategies. They considered the benchmark problems of Taillard [11], and reported the best-heuristic solutions for these benchmark problems with respect to the common objective function. It is to be noted that the integrated GAs has emerged to be the best for all benchmark problems.

A. Parameters tuning

Usually, the first set of tests defines for which values of $\{\alpha, \beta\}$ the algorithm converges to the known optimum of a selected problem. In our case, no competitive results for the joint production and preventive maintenance scheduling problems exist, with state-of-art algorithms. For this reason, we use the Taillard PFSP 20x10 [11]. However, we consider the relative increases in the common objective function yielded by the proposed ant-colony algorithm, relative to the best makespan value reported by the Taillard PFSP 20x10 with upper bound stated at 1659, when comparing the solutions. The best one is the one which gave the least increase.

Table 1 illustrates some of the simulations that have been carried out, using ant colonies, with different sets of $\{\alpha, \beta\}$ parameters. Pheromone was stored on nodes. We don't include the case $\alpha = 0$ and $\beta = 0$ because every ant simply branches randomly from one node to the other, for which the algorithm is expected not to find the optimum.

TABLE I
OPTIMUM FOR DIFFERENT VALUES OF α AND β .
RESULTS FOR EACH PAIR (α, β) ARE GIVEN WITH MAXIMUM NUMBER OF
CYCLES $NC_{max}=1000$, $\rho=0.1$, $m=20$.

	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$
$\alpha = 1$	1756	1764	1793	1788
$\alpha = 2$	1754	1772	1780	1791
$\alpha = 3$	1747	1744	1755	1770
$\alpha = 4$	1740	1755	1749	1762
$\alpha = 5$	1755	1742	1747	1789
$\alpha = 6$	1769	1771	1786	1792
$\alpha = 7$	1792	1769	1777	1755
$\alpha = 8$	1818	1801	1767	1771
$\alpha = 9$	1801	1783	1793	1755
$\alpha = 10$	1797	1793	1796	1774

Table 1 shows that the set of values for which an optimum value for the common objective function is reached is $\alpha = 4$ and $\beta = 1$.

Once established the values of α and β , the influence of the parameters that define the pheromone update can be investigated. Figure 1 shows the performance of the algorithm for different values of the evaporation constant ρ .

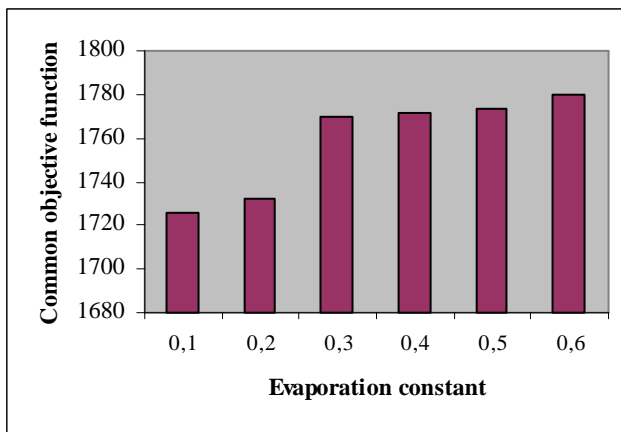


Fig.1. Influence of different setting of the evaporation coefficient to reach the optimum. ($NC_{max}=1000$, $\alpha=4$, $\beta=1$, $m=50$).

From the case of $\rho > 0.1$, the algorithm does not always find the optimum. This is due to the fact that when the evaporation rate is too high, only the edges belonging to the path with the early encountered local minimum receive pheromone update every cycle. The rest of the edges have their pheromone rapidly decreasing to zero with increasing number of cycles, due to the evaporation. At each new cycle, this effect is amplified and the ants will be more and more attracted to the edges that receive pheromone update. The ants will be searching in the small neighbourhood of the local minimum.

We noticed that the simulations that gave us the best results were the ones where the number of ants equals the number of jobs. Therefore, all ants start the resolution with joint schedules where the first job is different from one solution to another. All ants are scattered on the solutions space.

Figure 2 shows the performance of the algorithm for different values of the maximum number of cycles NC_{max} .

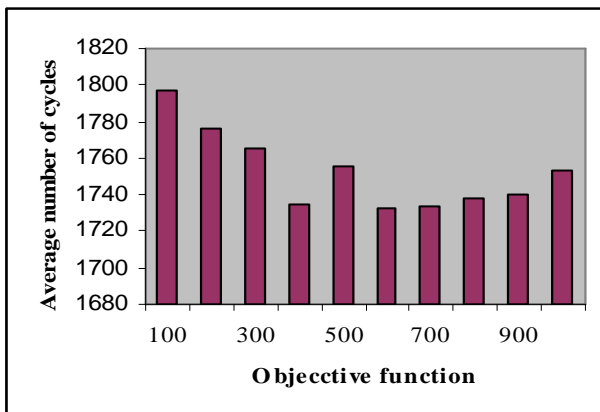


Fig. 2. Influence of different setting of the average number of cycles to reach optimum ($\alpha=4$, $\beta=1$, $\rho=0.1$).

The figure 2 shows that as the number of cycle's increases, the objective function improves until the number of cycles reaches 600 cycles. Above this number, the common objective function increases. Thus, the maximum number of cycles can be state at 600.

B. Performance analysis of the INTACO

After optimization of the parameters in the previous section for the specific Taillard 20x10 flowshop problem, this section shows some simulation results of the INTACO to some more complex benchmark problems, with the values of parameters tuned above.

In ACO, as in many other metaheuristics, the performance critically depends on the appropriate combination of local search with the solution construction mechanism. Therefore, we performed a detailed analysis of the effectiveness of the INTACO without and with the improvement procedure proposed in section III.5 at two levels: on the solution given by each ant (L1) and on the best solution given by the algorithm (L2). The results of evaluation are presented in Table 2.

TABLE II
RESULTS OF THE INTACO WITH DIFFERENT LEVELS OF IMPROVEMENT

Benchmark	INTACO*	L1	L2	Best level
7x7	7436	7409	7409	L1,L2
8x8	9396	9217	9321	L1
15x5	634	598	619	L1
20x5	1436	1261	1424	L1
20x10	1740	1638	1738	L1
20x15	2385	2087	2277	L1
30x15	3092	2782	3014	L1
50x10	3777	3285	3687	L1
75x20	6308	5732	6287	L1
100x10	654	636	652	L1

*: without the improvement procedure

Overall, it is observed from the results that the improvement procedure at L1 performs better than this at L2. The main reasons for such a superior performance are due to the fact that the solution of each ant is improved, at each cycle.

We defined, in section III.3, two types of mutation allowing generating neighbour solutions from the current one. The evaluation of the performance of the proposed mutation is presented in table 3.

TABLE III
RESULTS OF THE EVALUATION OF EACH TYPE OF MUTATION

Benchmark	VM1	VM2	IIM	ISM	Mutation
7x7	7375	7375	7375	7375	-
8x8	9217	9217	9218	9301	VM1,2
15x5	598	598	598	598	-
20x5	1261	1261	1261	1261	-
20x10	1618	1625	1627	1634	VM1
20x15	2069	2055	2077	2084	VM2
30x15	2734	2756	2771	2782	VM1
50x10	3263	3271	3278	3283	VM1
75x20	5698	5712	5718	5720	VM1
100x10	624	625	630	632	VM1

VM1: 1-point vertical mutation; VM2: 2-point vertical mutation;
IIM: Integrated Insertion-moves; ISM: Integrated Swap-moves

Overall, it is observed from the results that the 1-point vertical mutation gives the best results. This higher performance is mainly due to the effectiveness of mutation on the production sequence while respecting the maintenance tasks initial position. This factor assumes greater significance in enhancing the performance of the obtained joint schedule in the case of mutation with 1-point than in the case of two points.

Table 4 shows the comparison between the INTACO to another meta-heuristic: integrated GAs developed in [4]. GAs solutions are obtained after 100 executions of the method. The best result is saved, as well as the associated parameters. The following parameters are the same for all the executions of the genetic algorithms: crossover rate: 0.7; mutation rate: 0.01; renewal strategy: N_best, the replacement is done between the selected population and the crossed one. Population size: between 50 and 100. The crossover and mutation operators used are: k-point horizontal crossover and k-point vertical mutation.

However, since the GAs are known to prematurely converge, the only way to get the best solution was to use a strategy called "sharing" [21]. This strategy allows diversifying the population, avoiding the GAs to get trapped in a local optimum. We use sharing with the following parameters: $\alpha_{\text{Sharing}} = 0.99$ and $\beta_{\text{Sharing}} = 4$.

TABLE IV
COMMON OBJECTIVE FUNCTION GIVEN BY THE INTEGRATED GA AND INTACO

Benchmark	GA	INTACO*	INTACO
7×7	6898	7436	7375
8×8	8720	9396	9201
15×5	605	634	598
20×5	1222	1261	1221
20×10	1686	1740	1614
20×15	2203	2385	2002
30×15	2745	3092	2632
50×10	3698	3777	3183
75×20	5851	6308	5320
100×10	571	654	528

*: without the improvement procedure

Instead the use of "sharing" the INTACO performs better than the integrated GAs for most of the studied benchmarks.

Conclusion

Compared to other metaheuristics such as genetic algorithms, simulated annealing and taboo search, relatively few attempt have been made to solve scheduling problems using ant-colony algorithms.

In this paper, we investigated the problem of the joint production and preventive maintenance scheduling in permutation flowshop using ant-colony algorithm. We proposed a new algorithm, called INTACO, which tries to solve this problem, based on a joint representation of production and maintenance data. The goal is to optimize a common objective function which takes into account both maintenance and production criteria.

For evaluating the performance of the proposed ant-colony algorithm, the results of a recent study by Benbouzid and al. [4] are referred to. The proposed algorithm is enhanced by a local search procedure to yield high quality solutions. The proposed ant-colony algorithm performs better than the integrated genetic algorithms referred to.

Future research will be to investigate a bi-criterion approach to solve this problem.

REFERENCES

- [1] Lee C.Y., Chen Z.L., "Scheduling jobs and maintenance activities on parallel machines". *Naval Research Logistics*, 47:145-165, 2000.
- [2] Aggoune R., "Minimizing the makespan for the flowshop scheduling problem with availability constraints". *European Journal of Operational Research*; 153:534-43, 2004.
- [3] Cassady CR, Kutanoglu E., "Minimizing job tardiness using integrated preventive maintenance planning and production scheduling". *IIE Transactions*; 35(6):503-13, 2003.
- [4] Benbouzid F., Varnier C. & Zerhouni N., "Resolution of joint maintenance/production scheduling by sequential and integrated strategies". *7th International Work Conference on Artificial and Natural Neural Networks IWANN2003, 3-6 June, Spain*; LNCS 2687, 782-789, 2003.
- [5] Ruiz R., García-Díaz J.C, Maroto C., "Considering scheduling and preventive maintenance in the flowshop sequencing problem". *Computers & Operations Research*; 34:3314 - 3330; 2007.
- [6] Dorigo, M., *Optimization, learning and natural algorithms* (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [7] Dorigo, M., Maniezzo, V., Colomi, A. "The ant system: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man and Cybernetics*; Part B 26, 29-41, 1996.
- [8] Dorigo M., DiCaro G., *Ant Colony Optimisation: A new meta-heuristic*. IEEE, 1999.
- [9] Stuetzle, T. "An ant approach for the flowshop problem". In: *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT '98)*, 3. Verlag Mainz, Aachen, Germany, 1560-1564, 1998.
- [10] Rajendran, C., Ziegler, H., "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs". *European Journal of Operational Research*, 155, 426-438, 2004.
- [11] Taillard, E., "Benchmarks for basic scheduling problems". *European Journal of Operational Research*, 64, 278-285, 1993.
- [12] Ying K.C., Liao C.J., "An ant colony colony system for permutation flowshop sequencing". *Computers and operations research*, 31: 791-801, 2004.
- [13] Hu Y., Yan J., Ye F., Yu J., "flowshop rescheduling problem under rush orders". *Journal of Zhejiang University Science*, 6A(10):1040-1046, 2005.
- [14] Pinedo M., *Scheduling: theory, algorithms and systems*. 2nd ed., New Jersey: Prentice-Hall; 2002.
- [15] Rinnooy Kan AHG., *Machine scheduling problems: classification, complexity and computations*. The Hague: Martinus Nijhoff; 1976.
- [16] Biroloni, A., *Reliability engineering, theory and practice*, 4th ed., Berlin: Springer, 2004
- [17] Taillard E., "Some efficient heuristic methods for the flowshop sequencing problem". *European Journal of Operational Research*, 47:65-74, 1990.
- [18] Osman, I., Potts, C., "Simulated annealing for permutation flow shop scheduling". *OMEGA*, 17(6): 551-557, 1989.
- [19] Dannenbring, D., "An Evaluation of Flow Shop Sequencing Heuristics". *Management Science*, 23(11), pp. 1174-1182, 1977.
- [20] Benbouzid F., Varnier C., Zerhouni N., "New Genetic Operator for Solving the Joint Production and Maintenance Scheduling: Application to the Flow Shop Problem". In: *Proceedings of International conference Service System and Service Management ICSSSM'06, October-25-27, 2006, Troyes (France)*, 607-613, 2006.
- [21] Bonabeau E., Dorigo M., Thérault G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.