DOI : 10.1109/IECON.2012.6389511

New Hopfield Neural Network for joint Job Shop Scheduling of production and maintenance

N. Fnaiech [(1)], H. Hammami [(1)], A. Yahyaoui [(1)], C. Varnier [(2)], F. Fnaiech[(1)], *Senior Member, IEEE,* N. Zerhouni[(2)]

[(1)] University of Tunis, Ecole Supérieure des Sciences et Techniques de Tunis,
5, Avenue Taha Hussein, 1008 Tunis - Tunisia
[(2)] University of Franche-Comté, ENSMM, Laboratory of Automatics of Besançon,
25, rue Alain Savary, 25000 BESANCON - France

E-mail: fnaiech@ieee.org

***Abstract*- Job Shop Scheduling is one of the most difficult problems in industry and it is the main interest of the major researchers in the manufacturing research area. This problem becomes crucial when the production planning and maintenance have to be jointly solved. Several heuristics and intelligent methods have been so far proposed in the literature and applied. This work deals with a Hopfield Neural Network (HNN) method used for solving the JSP taking into account the maintenance tasks. While this method had been already proposed in the literature to solve the JSP alone, our main improvement of this method is to take into account the maintenance periods by extending the Hopfield net to handle the joint problem. Experimental study shows that the proposed HNN algorithm gives efficient results for the resolution of the joint job shop scheduling problem.**

***Keywords*—Availability, Optimization Methods, Manufacturing Planning, Manufacturing Scheduling, Maintenance, Computer Integrated Manufacturing, Resource Management, Manufacturing Automation Software, Production Management, Hopfield Networks.**

I. INTRODUCTION

Job shop scheduling problems (JSSP) are considered among the most complicated problems in industry (NP-hard). The JSSP aims to allocate a number of machines over time to perform a set of jobs with respect to resource and sequence constraints in order to optimize certain criterion; in our case the main purpose is minimizing the makespan. Also, as the production and the maintenance are two functions acting on the same resources, the joint scheduling problem of their operations appears a tedious task. In the literature, several researchers consider that the machines are always available all the time. However, in many realistic situations machines can be unavailable during a certain periods due to the maintenance activities.

Many techniques of resolution of JSSP have been applied to find good solutions. Among them are heuristic methods [3]; genetic algorithms [5], [6]; Taboo search [7] intelligent methods [8], [9]; neural networks [10], [11] etc. Recently, neural networks have been used to solve JSSP. Foo et *al.* [10] first used HNN with integer linear programming as an extension to minimize the makespan [12], [13]. Thereafter, several works using neural network have been applied to solve JSSP. Willems et *al.* [1], [2], [3] first proposed a constraint satisfaction neural network with HNN structure to model JSSP and respect resource and sequence constraints. Yahyaoui et *al.* [4] proposed a modification of the heuristic proposed by Willems et *al.* [1], [2], [3]. Their contribution consists in a suitable choice of the HNN initialization starting time of the JSSP to reduce the number of searching cycle and thus speed up HNN. All these works consider only JSSP without integrating maintenance activities which is more realistic in manufacturing companies.

Many researchers are interested in solving joint maintenance and production scheduling problem using many exact methods and heuristics such in [14], [15], [16], [17], [18], genetic algorithms for JSSP [19]. A survey of scheduling problems with availability constraints was proposed in [20]. It shows that very few works were done on the job-shop case. Neural Networks are widely used for solving Job shop scheduling problems of production only, for that reason, we found interesting to apply them to solve joint production and maintenance scheduling problems.

In this paper, a new HNN taking into account unavailability of the machines due to the preventive maintenance is proposed. By using the same architecture of the Hopfield Neural Network used by [1], [2], [3] and adding the procedure of initialization proposed by [4], we try to build another HNN for joint production scheduling problem and maintenance.

The remainder of this paper is organized as follows. Section II reviews the HNN presented by Willems et *al.* to solve JSSP with respect to resource and sequence constraints. Section III gives the heuristic of initialization proposed by Yahyaoui et *al.* [4] combined with the HNN. In section IV, we present the proposed new HNN which takes into account maintenance activities. Experimental example is studied in section V and the results are discussed; finally, a conclusion and future directions are given at the end of this paper.

II. REVIEW OF THE HOPFIELD NEURAL NETWORK PROPOSED BY WILLEMS ET AL. [1], [2], [3]

A. Formulation of the JSSP

Job Shop Scheduling Problem is one of the most interesting issues in manufacturing companies. It is defined as *n* jobs to be scheduled on *m* machines in a described order. Each job may have different number of operations and has its own processing order on machines. The processing time of each operation is fixed in advance. Operations cannot be interrupted once started. We focus our study on this kind of scheduling which is deterministic and static. The JSSP is considered as resolved if it respects two types of constraints: sequence constraints and resource constraints. These constraints are formulated by Willems et *al.* [1], [2], [3] using integer linear programming and model their resolution by HNN. They use these notations to describe the different constraints of the problem:

n	Number of jobs
i, p	Indexes of jobs
m	Number of machines
k,l,h,f,s	Index of a machine
j	Index of operation
O_{ijk}	Operation *j* of job *i* to be processed on machine *k*
t_{ijk}	Processing time of the operation O_{ijk}
S_{ijk}	Starting time of the operation O_{ijk}
H	A constant
Y_{ipk}	An indicator variable denotes which job proceeds on machine *k*
W_f	A positive weight of the feedback updating connections (e.g. +0.1, -0.1)

1) Sequence constraints

Sequence constraints or precedence constraints means that two operations of the same job cannot be processed simultaneously which can be formulated by:

$$S_{ijl} - S_{i(j-1)k} - t_{i(j-1)k} \geq 0 \quad (1)$$

For a JSSP with *n* jobs and *m* machines, it requires *n(m-1)* equations sequence constraints. The HNN architecture of this sequence constraint proposed by Willems et *al.* [1], [2], [3] is given by Fig. 1(a) with its simplified symbol (SC-unit) given in Fig. 1(b).

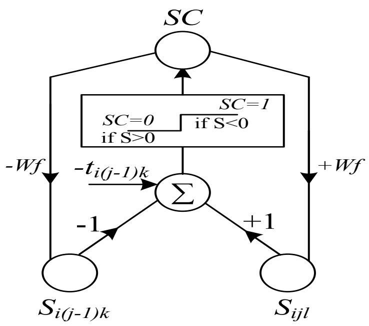

Fig.1.(a) HNN representation of sequence constraint

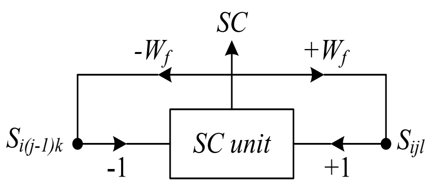

Fig.1.(b) Simplified symbol of sequence constraint

2) Resource constraints

Resource constraint means that no more than one operation can be performed on a machine at the same time. According to Willems et *al.* [1], [2], [3] for two operations O_{ijk} and O_{plk} that share the same resource *k*, it can be formulated by these two inequalities to respect resource constraints:

If operation O_{plk} is processed before O_{ijk}

$$S_{plk} - S_{ijk} + H(1 - Y_{ipk}) - t_{ijk} \geq 0 \quad (2)$$

If operation O_{ijk} is processed before O_{plk}

$$S_{ijk} - S_{plk} + HY_{ipk} - t_{plk} \geq 0 \quad (3)$$

Where Y_{ipk} is a decision variable that indicates which job is processed before the other on machine *k.*

$$Y_{ipk} = \begin{cases} 1 & \text{if } S_{ijk} \leq S_{plk} \\ 0 & \text{if } S_{ijk} \geq S_{plk} \end{cases}$$

And *H* represents an arbitrary positive number bigger than the sum of all processing times t_{ijk}.

$$H > \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ijk} \quad (4)$$

For a JSSP with *n* jobs and *m* machines, it requires *mn(n-1)* equations resource constraints. The HNN architecture of these resource constraints proposed in [1], [2], [3] is given by Fig. 2(a) with its simplified symbol (RC-unit) given in Fig. 2(b).

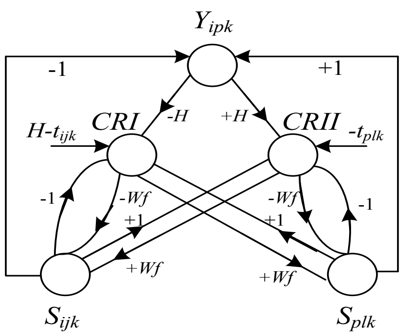

Fig.2.(a) HNN representation of resource constraints

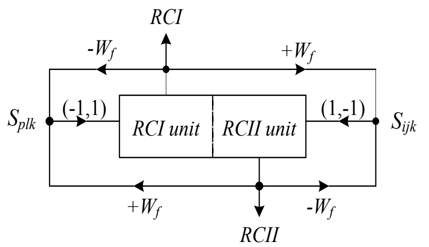

Fig.2.(b) Simplified symbol of resource constraints

It is interesting to mention that these proposed architectures aim to adjust, in an automatic way, the various values of starting times S_{ijk} to satisfy the prescribed constraints. It is due to the presence of return connections
($-W_f$) and ($+W_f$). For exemple if SC = 1 (the sequence constraint is violated), then S_{ijl} and $S_{i(j-1)k}$ will be modified step by step ($-W_f$ or $+W_f$) until have SC = 0, so the sequence constraint will be satisfied. Also for the resource constraints.

The simulation of several JSSP depends on the choice of the initialization of the starting times S_{ijk}. Willems and Brandts [1] used the arbitrary choice of the starting times. But, Yahyaoui et *al.* in [4] have proposed a new method of initialization to improve the searching speed of an optimal or near optimal solution using HNN. The next section will resume this initialization procedure.

III. HEURISTIC OF INITIALIZATION PROPOSED BY YAHYAOUI ET AL. [4]

To improve the results obtained by Willems et *al.* [1], [2], [3], Yahyaoui et al. [4] proposed a new heuristic of initialization of starting time based on a preprocessing procedure of the problem to be solved in order to speed the HNN and find an optimal or near optimal solution. This heuristic initialization consists in initializing different starting times by assuming that all the resources are continuously available for only each job and independently of the others [4]. It means that during this initialization respect at the beginning the sequence constraints between the operations without including resource constraints between the jobs, so the number of equations in every iteration can be reduced. It remains to respect resource constraints. Simulation results of this heuristic applied to the HNN proposed in [1], [2], [3] shows that this method guaranteed the acceleration of the neural network towards an optimal or near optimal solution minimizing the makespan C_{max} of a JSSP.

The proposed work is based on adding an additional Hopfield Neural Network which models the maintenance activities. Thus the resulting HNN is similar to the network proposed by Willems et *al.* [1], [2], [3] and uses the improved facilities of initialization developed by Yahyaoui et *al.* [4].

The combination of their works [1] and [4] was effective in terms of resolution, speed of convergence, quality of the solution obtained as well as the reduction of the computation time; however, the contribution of the unavailability of machines due to maintenance tasks will be more reasonable and realistic which is detailed in the next section.

IV. NEW HOPFIELD NEURAL NETWORK MODEL FOR JOINT SCHEDULING PRODUCTION AND MAINTENANCE

A. Description of the joint production and maintenance scheduling problem

Many studies in the literature which are dedicated to scheduling problems consider that the resources are always available. Nevertheless, this hypothesis is not faithful to the reality of manufacturing shops due to the maintenance tasks on machines. Indeed, scheduling maintenance activities, in given intervals, can affect the scheduling of production, since it is necessary to guarantee the availability of production tools and eliminate risks associated with machine breakdowns. So, these two activities appear as significant as antagonistic functions in production workshops.

In this paper, we treat a combined maintenance and production scheduling in job shop scheduling problem in order to minimize the makespan C_{max}. Consider that the durations of unavailability of machines are known and fixed in advance; we thus speak about a preventive maintenance. The number of maintenance tasks is different from one machine to another, and they do not necessarily have the same size.

Based on the same architecture of HNN construction, another Hopfield Neural Network modeling the preventive maintenance is then added.

Figure 3 gives the representation in Gantt format the proposed problem.

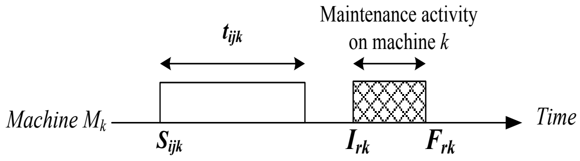

Fig. 3. Representation problem

where:

I_{rk} : The initial date of the r^{th} maintenance task on machine *k*.

F_{rk}: The final date of the r^{th} maintenance task on machine *k*.

Considering that the maintenance tasks are already fixed in advance, the HNN which will be defined has to distinguish three possible cases depending on the position of S_{ijk} with regard to the maintenance task:

- *First case:* If an operation begins after the end of a maintenance task, so there is no change of the value S_{ijk}.
- *Second case:* If an operation is executed and ends before the beginning of a new maintenance task, there is no change of the value S_{ijk}.
- *Third case:* If there is an overlapping between an operation of production and an interval of unavailability, we have to shift the operation by steps the value of S_{ijk} until the elimination of the overlapping between the production task and the maintenance task.

Mathematically, we can model these three cases by the following inequations:

If $$F_{rk} \geq S_{ijk} \quad (5)$$

and $$S_{ijk} + t_{ijk} \geq I_{rk} \quad (6)$$

Then $$S_{ijk} = S_{ijk} + W_f \quad (7)$$

Otherwise S_{ijk} remains constant.

From this mathematical model, one can propose HNN architecture which takes into account the availability of machines; the problem is related to Unavailability Constraints (UC-unit). Only one period [I_{rk} F_{rk}] of unavailability on a single machine *k* and by meeting an only one starting time S_{ijk} is modeled by Fig. 4(a). The simplified symbol of this proposed architecture is given in Fig. 4(b).

In this model, the only variable which may change is S_{ijk}. This change depends on the state of *U*, *V* and L_{ijkr} which are three boolean variables of decision.

For (***U = 1***) and ***(V = 1)***, it means that an operation of production begins before the initial date of the maintenance task or inside the maintenance task (5) and its processing time falls in conflict with the period of the unavailability of the machine (6). Thus it is necessary to shift the starting time S_{ijk} till the end of the interval of unavailability.

Consequently, the new starting time S_{ijk} of the operation becomes equal to the final date F_{rk} of the maintenance task on machine *k*.

This case will be formulated by this condition:

If $\boldsymbol{U=1}$ it means $F_{rk} \geq S_{ijk}$ (8)
And $\Rightarrow \boldsymbol{L_{ijkr}=1}$
$\boldsymbol{V=1}$ it means $S_{ijk} + t_{ijk} - I_{rk} \geq 0$ (9)

Thus $$\boldsymbol{S_{ijk} = S_{ijk} + W_f \times L_{ijkr}} \quad (10)$$

Therefore the starting time S_{ijk} is shifted by W_f.
If both ($U = 0$) and ($V = 0$) or one of them is equal to 0, the variable of decision L_{ijkr} will be equal to 0 and thus S_{ijk} will not be shifted.

So: $$\boldsymbol{S_{ijk} = S_{ijk} + W_f \times 0 = S_{ijk}}$$

B. *Global architecture of the HNN for joint production and preventive maintenance scheduling*

A joint job shop scheduling is considered as feasible if it respects all the constraints of the production workshop. Indeed, it has to respect jointly three types of constraints which are sequence constraints, resource constraints and unavailability constraints of the machines. Therefore, the resolution of joint production and maintenance JSSP consists of finding the various suitable starting times S_{ijk} while satisfying the various constraints given by the equations quoted above (8 - 10).

Based on the three models given by Fig. 1(a)-(b), Fig. 2(a)-(b) and Fig. 4(a)-(b), the global architecture of the proposed HNN given by Fig. 5 (a), (b) and (c) which contains three types of constraints is set by the following. The starting times S_{ijk} will depend on three levels which are sequence constraints, resource constraints and unavailability constraints:

- *1st level:* Construction architecture which respects sequence constraints: the HNN representation is repeated between the operations of each job and it is shown by Fig.5 (a).
- *2nd level:* Construction architecture which respects resource constraints: it is repeated between the starting times of two different jobs that belong the same machine and it is shown by Fig.5 (b).
- *3rd level:* Construction architecture which respects unavailability constraints: the HNN representation is repeated for all starting times and for each interval maintenance task [I_{rk} F_{rk}] on each machine. It is shown by Fig.5 (c).

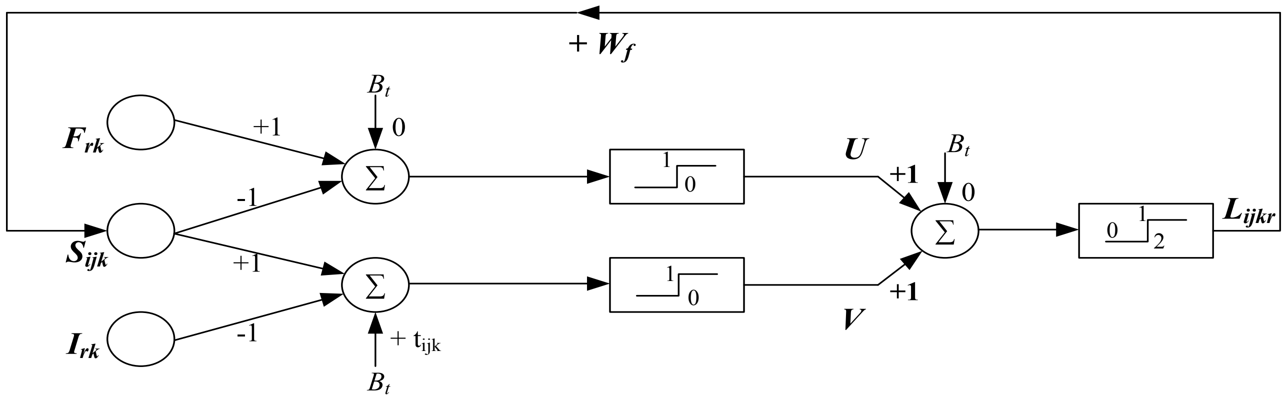

Fig. 4.(a) Hopfield Neural Network for solving an overlapping between a production operation and a maintenance task on machine k

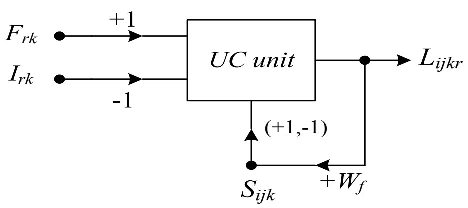

Fig. 4.(b) Simplified symbol of Unavailability Constraint

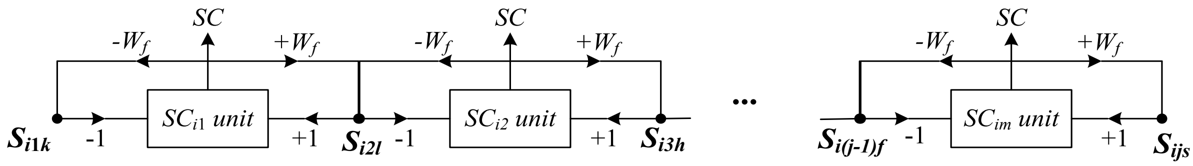

Fig. 5. (a) Construction architecture of sequence constraints

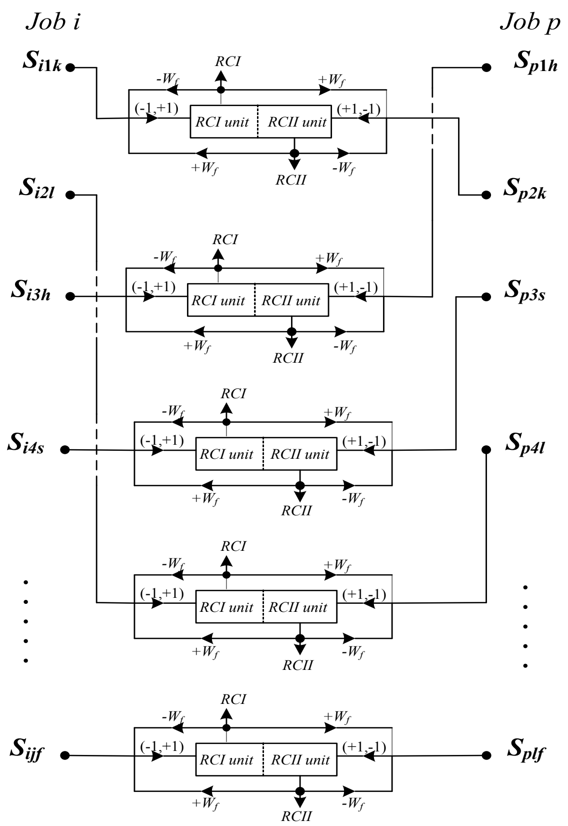

Fig. 5. (b) Construction architecture of resource constraints

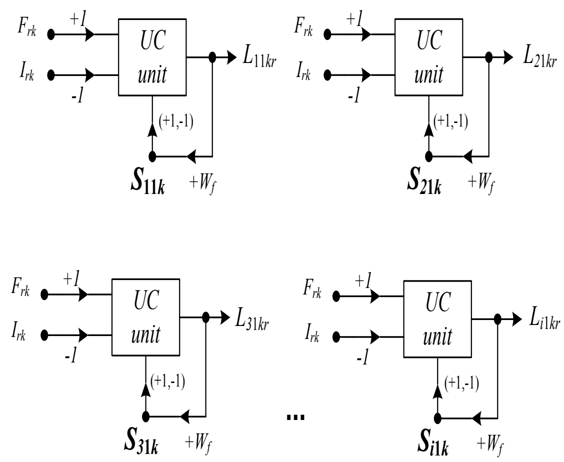

Fig. 5. (c) Construction architecture of unavailability constraints

C. General algorithm of the proposed HNN

The basic steps of the general algorithm and running mechanism for solving joint job-shop scheduling of production and maintenance are shown as follows:

Step 1) Initialize the different starting times S_{ijk} of the job shop problem in a general way of sequencing order using the initialization procedure proposed by Yahyaoui et *al.* [4].

Step 2) Run all the SC-units and RC-units in automatic way. If there is only one unit different to zero (RCI≠ 0, RCII≠ 0 or SC≠0) means the corresponding constraint is disabled. In this case, the adjustment of all the S_{ijk} by the feedback connections (+W_f) or (–W_f) is done. This running is repeated automatically until all units are equal to zero.

Step 3) Run all the UC-units, if there is only one unit different to zero (L_{ijkr} ≠0) means that the unavailability constraint is not satisfied. In this case, the adjustment of all the S_{ijk} by the feedback connections (+W_f) is done. This running is repeated automatically until all L_{ijkr} units are equal to zero.

Step 4) Test again the obtained starting times S_{ijk} by repeating the running mechanism of the SC-units and RC-units to ensure that the sequence constraints and resource constraints remained satisfied. If all the units are equal to zero (RCI=0, RCII=0 and SC=0) then go to step 5) otherwise go back to step 2).

Step 5) Display the starting times obtained to build Gantt chart in order to compute after the makespan C_{max}.

The simulation stops automatically when all the constraints of the problem are satisfied (RCI=0, RCII=0, SC=0 and L_{ijkr}=0). These steps lead to find a set of starting times S_{ijk} which respect the three kinds of constraints by considering that the minimization of the makespan C_{max} is our main objective.

V. EXPERIMENTAL STUDY

A. Simulation example

To evaluate the effectiveness of the proposed model, three small-sized problems are tested. Table I represents the machines allocations and the processing time of the first example with 2 jobs and 3 machines. Its corresponding distribution of their maintenance tasks are given in Table II randomly generated.

TABLE I
Machine allocations
and processing times of *2/3/J/C_{max}*

Machine allocation (Processing Time)			
	Operation		
Job	1	2	3
1	1(5)	2(8)	3(2)
2	3(7)	1(3)	2(9)

TABLE II
Distribution of maintenance tasks on the machines

Machine M_k	$[I_{1k}\ \ F_{1k}]$
M_1	[10 13]
M_2	[14 16]
M_3	[20 23]

After running the global architecture of HNN with fixed maintenance activities, the obtained results are given in Table III and a Gantt diagram is depicted on Fig. 6.

TABLE III
RESULTING FINAL VALUES OF STARTING TIME OF THE PROBLEM *2/3/J/C_{max}* WITH UNAVAILABILITY CONSTRAINTS OF THE MACHINES

Job i	*Starting time*	*Initial value*	***Final value***	*Job i*	*Starting time*	*Initial value*	***Final value***
Job 1	S_{111}	0	**0**	*Job 2*	S_{213}	0	**0**
	S_{122}	5	**14**		S_{221}	7	**12**
	S_{133}	13	**22**		S_{232}	10	**22**
$\boldsymbol{C_{max}} = \mathbf{31}$							

The corresponding makespan of the example *2/3/C_{max}* is equal to 31 when each machine has one maintenance task to be done during different data.

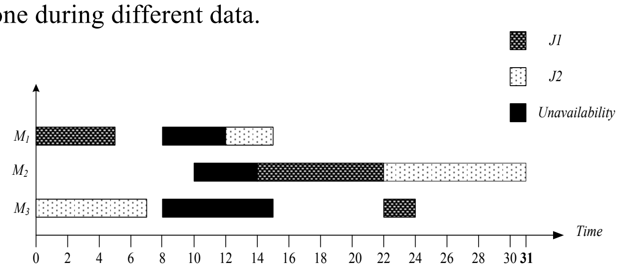

Fig.6. Gantt chart of the example *2/3/Cmax*

The second example was generated with 4 jobs and 3 machines which is given in Table IV. The maintenance tasks are given in Table V.

TABLE IV
Machine allocations and processing times of *4/3/J/C_{max}*

Machine allocation (Processing Time)			
	Operation		
Job	1	2	3
1	1(4)	2(3)	3(2)
2	2(1)	1(4)	3(4)
3	3(3)	2(2)	1(3)
4	2(3)	3(3)	1(1)

TABLE V
DISTRIBUTION OF MAINTENANCE TASKS ON THE MACHINES

Machine M_k	$[I_{1k}\ \ F_{1k}]$	$[I_{2k}\ \ F_{2k}]$
M_1	[6 10]	[15 19]
M_2	[12 17]	_
M_3	[5 9]	[17 21]

The obtained results of the definitive starting times are given in Table VI and the Gantt chart is represented by Fig.7.

TABLE VI
RESULTING FINAL VALUES OF STARTING TIME OF THE PROBLEM *4/3/J/C_{max}* WITH UNAVAILABILITY CONSTRAINTS OF THE MACHINES

Job i	*Starting time*	*Initial value*	***Final value***	*Job i*	*Starting time*	*Initial value*	***Final value***
Job 1	S_{111}	0	**0**	*Job 2*	S_{212}	0	**0**
	S_{122}	4	**6**		S_{221}	1	**10**
	S_{133}	7	**25**		S_{233}	5	**21**
Job 3	S_{313}	0	**0**	*Job 4*	S_{412}	0	**1**
	S_{322}	3	**4**		S_{423}	3	**9**
	S_{331}	5	**19**		S_{431}	6	**22**
$\boldsymbol{C_{max}} = \mathbf{27}$							

The corresponding makespan of the example *4/3/C_{max}* is equal to 27 when the first and the third machines have two maintenance tasks to be executed during different data; the second machine has only a single maintenance task.

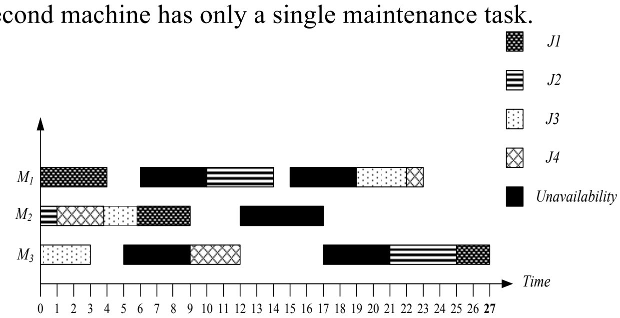

Fig.7. Gantt chart of the example *4/3/Cmax*

B. Comments of the simulation results

According to the two Gantt charts obtained, one can conclude that the HNN converges to an optimal solution for the example *2/3/C_{max}* and near optimal solution for the example *4/3/C_{max}*. In fact, the visualization of the solution presented in Fig.7 shows that the operation O_{313} can be realized before the second maintenance task [I_{23} F_{23}] of the third machine which minimize even more the makespan C_{max}.

Our HNN always converges to a feasable solution satisfying all the constraints of a job shop, however, this solution can not be optimal in some cases. This depends on the distribution of the fixed maintenance tasks of the problem.

VI. CONCLUSION AND PERSPECTIVES

In this paper, a new Hopfield Neural Network model for solving joint JSSP of production and preventive maintenance is proposed. It is seen as an extension of the Willems et *al.* [1], [2], [3] algorithm combined with the heuristic initialization procedure proposed by Yahyaoui et *al.* [4]. This proposed HNN model is able to solve the scheduling problem with unavailability of the machines. Our main objective is to determine different starting times which satisfy sequence constraints, resource constraints and unavailability constraints in order to minimize C_{max}.

Experimental examples show that the new HNN model provides good results for small and large problems; however the obtained solutions can not be optimal in some cases. This can be improved by implementing other heuristics to converge on a better solution.

Extension of this approach could be done for flexible maintenance tasks (tasks which are not fixed but are able to start inside a given interval). Another interesting future research is to investigate random failures of the machines where considering Job shop scheduling problem for joint production and corrective maintenance (random failures of the machines).

REFERENCES

[1]. T. M. Willems and L. E. M. W. Brandts, "Implementing heuristics as an optimization criterion in neural networks for job-shop scheduling," *J. Intell. Manufacturing*, vol. 6, pp. 377–387, 1995.

[2]. T. M. Willems, and J. E. Rod, "Neural nets for job-shop scheduling will they do the job?," *in Proc. IFAC 12th World Conference*, vol. 3, pp. 53-56, 1993.

[3]. T. M. Willems, and J. E. Rooda, "Neural networks for job-shop scheduling," *Contr. Eng. Practice*, vol. 2, no. 1, pp. 31–39, 1994.

[4]. A. Yahyaoui, N. Fnaiech, and F. Fnaiech, A Suitable Initialization Procedure for Speeding a Neural Network Job-Shop Scheduling," *IEEE Transactions on Industrial Electronics*, *Vol. 58* (Issue 3)*: 1052-1060, March 2011.*

[5]. O. Morandin, D. S. Sanches, A. C. Deriz, E. Kato, R. H. Tsunaki, "An Adaptive Genetic Algorithm Based Approach for Production Reactive Scheduling of Manufacturing Systems," *34th Annual Conference of Industrial Electronics of IEEE IECON*08, *Orlando* 10-13, pp. 1461-1466, Nov. 2008.

[6]. O. Morandin, E. R. R. Kato, A. C. Deriz, D. S. Sanches, "A Search Method using Genetic Algorithm for Production Reactive Scheduling of Manufacturing Systems," *34th Annual Conference of Industrial Electronics of IEEE IECON*08, *Orlando* 10-13, pp. 1843-1848, Nov. 2008.

[7]. C. Yong, Z. Pei, G. Li, Z. Lin, G. Yun, Q. Rao, "A tabu search algorithm with a new neighbourhood structure for the job shop scheduling problem," *Computers and Operations Research,* vol. 34 , Issue 11, pp.3229-3242, November 2007.

[8]. R. A. Zaied, K. Abhary, "A Design of an Intelligent Maintenance Integrated System into Manufacturing Systems," *IEEE International Conference on Industrial Technology, ICIT,* pp.1-6, 2009.

[9]. A. Ferrolho, M. Crisostomo, "Intelligent Control and Integration Software for Flexible Manufacturing Cells," *IEEE Trans on Industrial Electronics*, vol. 3, no. 1, pp.3-11, February 2007.

[10]. S. Y-P. Foo and Y. Takefuji, "Stochastic neural networks for solving job-shop scheduling: Parts I. Problem representation," *in Proc. of the IEEE Inter. Conference on Neural Networks,* vol. 11, pp. 275-282, 1988.

[11]. S. Yang and D. Wang "Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job Shop Scheduling," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp.474-485, March 2000.

[12]. S. Y-P. Foo and Y. Takefuji, "Integer-linear programming neural networks for job-shop scheduling," in *Proc. IEEE IJCNN'88,* pp. 341-348, 1988.

[13]. S. Y. Foo, Y. Takefuji and H. Szu, "Scaling properties of neural networks for job-shop scheduling," *Neurocomputing*, *Optimization and Combinatorics*, Part I, vol. 8, pp. 79-91, May 1995.

[14]. T. Lust, O. Roux, and F. Riane, "Exact and heuristic methods for the selective maintenance problem", *International Journal of Production Economics, Vol. 197: 1166–1177, 2009.*

[15]. M. Kans, and A. Ingwald, Common database for cost-effective improvement of maintenance performance. *International Journal of Production Economics, Vol. 113:734–747, 2008*.

[16]. O. Roux, D. Duvivier, V. Dhaevers, N. Meskens, and A. Artiba, "Multicriteria approach to rank scheduling strategies", *International Journal of Production Economics, Vol. 112: 192–201, 2008*.

[17]. A. Yahyaoui, N.Fnaiech, and F. Fnaiech, "New Shifting Method for Job Shop Scheduling Subject to Invariant Constraints of Resources Availability", *Annual Conference of the IEEE Industrial Electronics Society IECON*, *2009*.

[18]. N. Fnaiech, A. Yahyaoui, F. Fnaiech, "New Shifting Method for Combined Production Scheduling and Maintenance Cost in Job Shop", *International Review on Modelling and Simulations (I.RE.MO.S)*, *Vol.4, N.2, April 2011*.

[19]. Y. Harrath, B. Chebel-Morello, N. Zerhouni "Lower Bounds and Multiobjective Evolutionary Optimization for Combined Maintenance and Production Scheduling in Job Shop" *IEEE Conference on Emerging Technologies and Factory Automation, 2003. Proceedings.* 16-19 Sept. 2003 Page(s): 95 – 100 vol.2.

[20]. Ma, Y.; Chu, C. & Zuo, C. "A survey of scheduling with deterministic machine availability constraints", *Computers & Industrial Engineering,* 2010, 58, 199 – 211.