International Journal of Computer Aided Manufacturing
Vol. 2: Issue 1
www.journalspub.com

Optimizing Work Scheduling in Shop Floor Using Ant-Colony Algorithm

F. Gilbert Jerald, K.M. Senthil Kumar
M.E. Industrial Engineering, Department of Mechanical Engineering, Kumaraguru College of Technology, Coimbatore, India

Abstract

A job shop manufacturing system is specifically designed to simultaneously produce different types of products in a shop floor. Job shop scheduling problems (JSSPs) have been studied extensively and most instances of JSSP are NP-hard, which implies that there is no polynomial time algorithm to solve them. As a result, many approximation methods have been explored to find near-optimal solutions within reasonable computational efforts. Furthermore, in a real world, JSSP is generally dynamic with continuous incoming jobs and providing schedules dynamically within constrained computational times in order to optimize the system performance becomes a great challenge. In this paper, Ant Colony Optimization Algorithm (ACO) is proposed to reduce the total completion time of shop floor projects. In this scheduling, the time is considered as a main factor. The works are scheduled as based on this time factor. Less time consumption of works is scheduled as first order and others are outsourced in this system. The experimental result shows that ACO the average percentage of reduction in makes span is up to 21.12%.

Keyword: task scheduling, ant colony optimization, completion time, shop floor projects

****Corresponding Authors***
E-mail: gilbertjerald1993gmail.com, Senthilkumar.km.mec@kct.ac.in

INTRODUCTION

The process of scheduling is the decision-making process that deals with the resources allocation to tasks overtime. In the current competitive market environment, companies have delivered the products on the date which is committed to the costumers, using the available resources in the most efficient manner. To achieve these goals, an optimized scheduling, possibly with antagonic goals, is needed.[1]

The methods, that is scheduling, are proposed in the beginning of last century by Henry Gantt, have developed into very sophisticated algorithms, focusing both on stochastic systems and on deterministic. Now a days, we can find out the analytical solutions for different scheduling problems, as well as heuristic optimization methods like genetic algorithms (Bean 1994), or market-based-approaches (Wellman et al. 2001), to find out the NP-hard problems. Although, there are still many topics, both practical and theoretical that have to be further studied in the near future (McKay et al. 2001). One of these topics is the divided scheduling in manufacturing.

The system scheduling of large-scale processes along with and constrains and complex goals can hardly be define using a centralized scheduler. The decay of the problem that is distributed by different interacting agents in the process, that is, tasks and resources, can lead to an improved solution since every presenting part contributes with suggestions and information to the final decision. This type of disseminated methodology is even more important since the future scheduling strategies will have to include online and reactive rearranging, in order to face different phenomena like production failures or changes in the production planning, imposed by clients or by the market.[2–4]

In a distributed arranging problem, the number of managers involved and the amount of information that has to be replaced is very large. Various managers algorithms based on social pests, can avoid this difficulty. Social pests, e.g. ants, have seized the attention of scientists because of the high structuration level that the associations can complete, especially when equated to the relative easiness of the individuals. Artificial ants are offered in detail in (Dorigo et al. 1996), as the result of initial works, where ants were used to solve different types of NP-hard difficulties. Colorni et al. (1994) and more recently, Cicirello and Smith (2001), have projected the application of the ant associations algorithm to solve the jobshop floor problematic. Here, we propose a new method for decentralized arranging in a parallel machine environment.

In this paper, the completion time (C_i) of job i is the time at which processing of the last operation of the job is completed is focused. The operational production management, which decides the processing of those orders on the shop floor in order to fulfill the order requirements, and at the same time, optimizes the performance of the manufacturing system. It needs proper scheduling strategies to meet those requirements. After scheduling, the schedule is transferred to the shop floor and the implementation of a schedule is often referred to as dispatching. The paper is organized as follows, the next section defines a distinctive production system and the restrictions of the schedulers in this type of processes. Then, the principles of the optimization algorithm using ant colonies are introduced, as well as the new agenda for the application to arranging of production processes. Finally, some simulation examples and the analysis of the results are presented. The closing section concludes this paper and defines the future research work.[5–9]

SCHEDULING IN MANUFACTURING PRODUCTION MANAGEMENT

In many industrial and assembly services, every job can be administered in the same type of machines. This kind of environment, where the machines are set up in parallel is usually stated to as parallel machines environment (Pinedo 2002). The study of this environment is very significant from both a practical and theoretical point of view: the existence of resources in parallel is same in the real world, e.g. in the flexible flow shop outlines, and the techniques used for machines in parallel are often used in decay procedures for multistage structures.

Manufacturing Production Management

The control activities and production management in a manufacturing system can be categorized as tactical, strategic, and operational activities, depending on the long, medium or short term nature of their tasks (Hopp and Spearman, 2000; Chryssolouris, 2006). In this proposed system, the short term of their task completion is scheduled using ACO. The scheduling problem is discussed given below (Figure 1).

Arranging deals with the distribution of scarce resources to jobs over time. It is a decision-making procedure with the goal of optimizing one or more objects (Pinedo, 2002). The result of a arranging procedure generates one or several schedules, which are cleared as plans with reference to the

International Journal of Computer Aided Manufacturing
Vol. 2: Issue 1
www.journalspub.com

sequence of and time allocated for each item or operation necessary to complete the item (Vollmann *et al*. 1992). Here, the dynamic scheduling problem is mainly focused.[10–16]

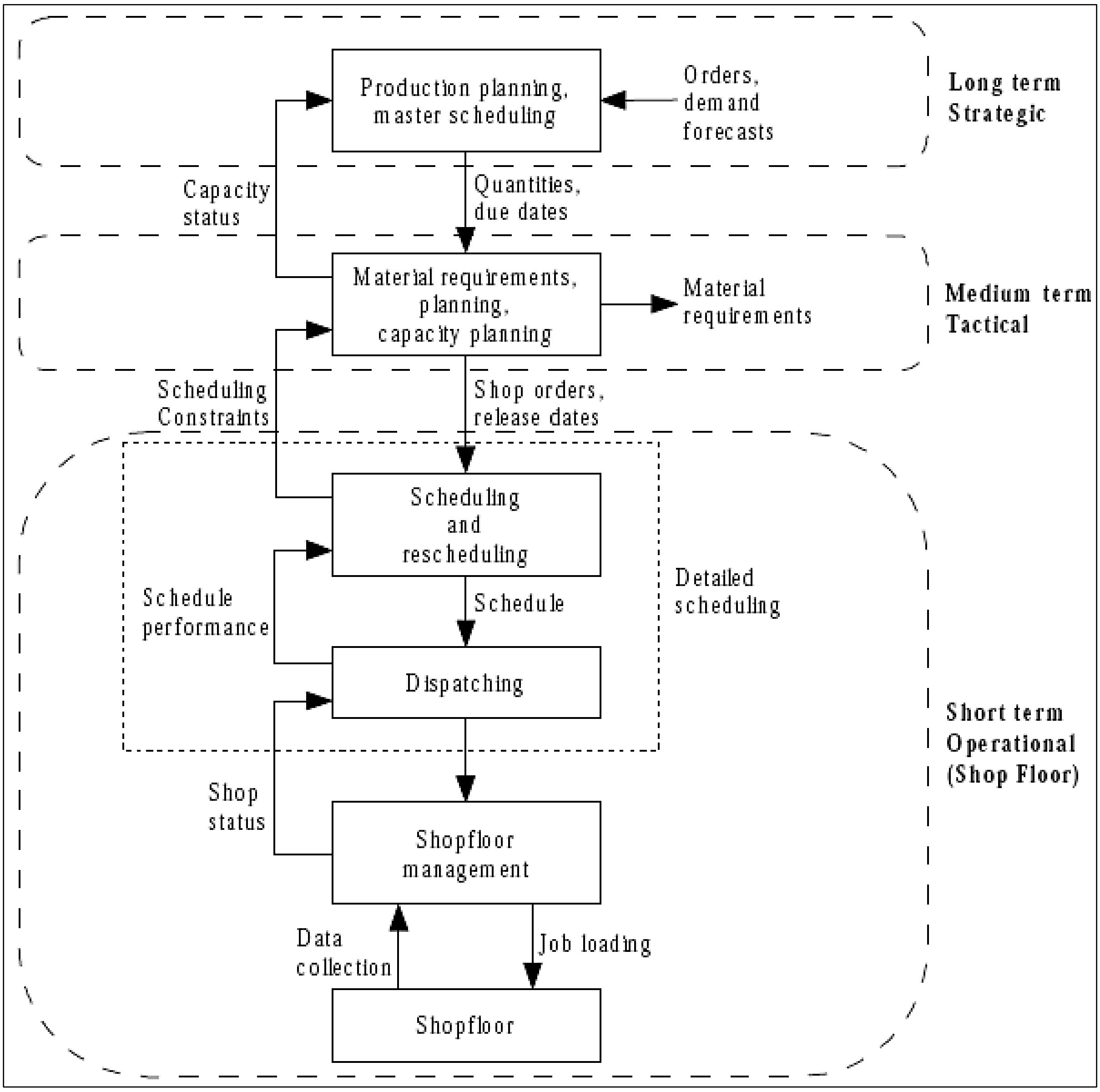

Fig. 1. The Information Flow Diagram in a Manufacturing System (RONG 2008).

Dynamic Scheduling Problems

Scheduling in the real world is dynamic and stochastic in nature. A scheduling problem is dynamic if there are continuous arrivals of new jobs and stochastic if uncertain events like machine breakdowns or variant processing times are considered. Those events are introduced into the system due to two factors. Quantities may either have inherent variability or they cannot be measured exactly (Ovacik and Uzsoy, 1994, 1997). The main consequence of those uncertainties for a scheduling system is that a predetermined schedule can become obsolete immediately. In stochastic/dynamic industrial environments, production planners, managers, and supervisors must not only generate high-quality schedules but also react promptly to unexpected events in order to revise schedules in a cost-effective manner. In an attempt to construct an effective reactive scheduling system, various approaches have been proposed and they can be categorized as industrial and academic studies.

The essential motivation of the current study is to develop a scheduling system that can keep on optimizing the performance of a job shop manufacturing system in real time in the face of dynamic events.ant colony, has found that autonomous agents like ants can find the shortest route from their nest to a food source based on the pheromone strength on their ways. Each ant affects the environment by leaving behind itself some amount of pheromone. This type of optimization mechanism is a collective effect of the interactions between the ants and the pheromone environment.

Furthermore, it is also found that an alternative shortest path can soon be formed by foraging ants if the current one is not available. Both features are of great research interests in the view of their presentations in stochastic/dynamic arranging environments. In order to realize this mechanism for the optimization purpose in scheduling problems, two implementations had been proposed. One is ACO based scheduling and outsourcing.[6–10]

SCHEDULING USING ANT COLONIES

Ants are societal insects. They used to live in groups and all their actions are toward the existence of the colony as a whole, rather than the advantage of a single specific of the society. The individual ants have no distinct abilities. The communication between each other using chemical constituents, the pheromones. This indirect communication allows the entire colony to accomplish complex jobs, such as forming the shortest route paths from their nests to feeding sources.

In (Dorigo *et al.* 1996) an optimization algorithm was planned that tries to mimic the foraging performance of real ants, i.e. the behavior of wandering in the search for food. This algorithm has already been effectively used to solve the TSP (Gambardella and Dorigo 1996), and other NP hard optimization difficulties (Silva *et al.* 2002). The next subsections describe the ant colonies algorithm and a new presentation in scheduling of production systems.

Data Collection

In this system, fifteen types of manufacturing tools are considered. The quantity of all tools set to as 5000. Four types of working time are considered for Computer Numerical Control (CNC). Finally the total completion time is calculated. Based on this time factor only scheduling is processed (Tables 1, 2).[11,15]

General Description of the Ant Colonies Algorithm

When an ant is examining for the nearby food source and comes across with several probable trails, it tends to choose the trail with the biggest concentration of pheromone, with a definite probability p. After choosing the trail, it deposits a definite quantity of pheromone, growing the concentration of pheromones in this trail. The ants return to the nest using always the similar path, depositing alternative portion of pheromone in the way back.

Imagine then, that two ants at the similar location select two different trails at the same time. The pheromone concentration on the shortest path will increase faster than the other: the ant, which is chooses this way, will deposit more pheromones in a lesser period of time, because it backs earlier. If a whole group of thousands of ants follows this behavior, quickly the concentration of pheromone in the shortest

path will be much higher than the concentration in other ways.
Then the possibility of choosing any other path will be very small, and only very few ants among the colony will fail to track the shortest path. There is another occurrence related with the pheromone concentration. Since it is a chemical substance, it tends to dissolve in the air, so the concentration of pheromones disappears along the time. In this way, the concentration of the less used ways will be much lower than that on the most used ones, not only because the concentration rises in the other paths, but also because their own concentration declines.

Table 1. *Data Collection of Shop Floor.*

S.No	Part Name	Quantity	CNC 1 Time (min)	CNC 2 Time (min)	CNC 3 Time (min)	CNC 4 Time (min)	Total CNC Time (min)	Profit per piece	Loss per piece when out source	Due Date
1	Bypass Screw Cap	5000	1.45	0.3	0	0	1.75	4.6682	-1.35	5
2	End Cap Separator Tube Variable Orifice	5000	1	0.5	2.1	0	3.6	6.4077	-2.48	5
3	Throw Out Ring	5000	1	0.5	0	0	1.5	1.9	-0.90	6
4	Piston Air Separator Variable Orifice	5000	2	0	0	0	2	1.4853	-0.34	8
5	Cap Nut - Check Valve Shaft	5000	1.1	0.2	0	0	1.3	0.8705	-0.33	10
6	G 1/4" Plug	5000	2.5	1	0	0	3.5	1.9642	-6.36	2
7	Adjustable Screw - Bypass Valve	5000	3.5	1.5	0	0	5	3.3671	-8.54	4
8	Check Valve Shaft	5000	1.4	1	1.5	0	3.9	2.4193	-9.11	6
9	Needle - Air Valve	5000	2.5	1	0	0	3.5	1.7309	-4.54	10
10	Retainer Check Valve Seal	5000	2.05	1.2	0	0	3.25	1.4174	-5.79	12
11	Bypass Valve Shaft	5000	1	1.5	1.5	0	4	3.3901	-11.29	14
12	Housing Air Valve	5000	1.2	4	2	0	7.2	6.6228	-29.35	5
13	Rotor Shaft-gpu 90	5000	3.25	0.5	1	1.25	6	9.9484	-6.93	6
14	Body - Control Valve Coated - Lee Valve	5000	1	2.3	1	3.2	7.5	8.9494	3.06	4
15	Air Separator Tube Var. Orifice GPU90	5000	4.15	1.4	0	0	5.55	5.3879	-9.27	8

In general, the ant colony performance can be defined formally using the following mathematical framework. Let the nest and the food source be associated by several different ways, connecting n intermediate nodes. The ant k in node i chooses one of the probable trails (i,j) connecting the real node to one of other probable positions $j \in \{1,\dots,n\}$, with probability

$$p_{ij}^{k} = f(\tau_{ij}) \quad (1)$$

where τ_{ij} is the pheromone concentration on the way connecting i to j, in the way to the food source. The pheromone in this trail will vary in time due to:

$$\tau_{ij}(t+1) = \tau_{ij}(t) \times \rho + \delta_{ij}^{k} \quad (2)$$

where δ_{ij}^{k} pheromone free by the ant k on the trail (i,j) and $\rho \in [0,1]$ is the evaporation coefficient. The classification is continuous, so the time deeds as the performance index, since the shortest ways will have the pheromone concentration enlarged in a shorter period of time.

This is the mathematical explanation of a real colony of ants. However, the artificial ants that mimic this performance, can be uploaded with more features, e.g. memory and ability to see. If the pheromone states the *experience* of the colony in the work of finding the shortest path, memory and skill to see, express useful *knowledge* about the problem the ants are resolving. In this way, (1) can be extended to:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha} . \eta_{ij}^{\beta}}{\sum_{r \notin \Gamma}^{n} \tau_{ij}^{\alpha} . \eta_{ij}^{\beta}} if\ j \notin \Gamma \\ 0\ otherwise \end{cases} \quad (3)$$

where η_{ij} a perceptibility is *function* and Γ is a *work list*. In this case, the visibility states the capability of seeing that is the nearest node j to travel toward the food place. Γ is a list that comprises all the trails that the ant has previously passed and must not be chosen again (artificial ants can go back prior achieving the food source). This acts as the retention of an ant.

If the TSP is the problem to be resolved, the perceptibility function can be the opposite of the distance from city i to city j expressed in a matrix d_{ij}. Then η_{ij} = 1/ d_{ij} and the work list Γ is the list of municipalities that the ant has previously visited. The parameters α and β express the relative weight between the importance of pheromone concentration τ and the visibility η. Finally, each ant deposits a pheromone δ_{ij}^{k}on the chosen trail:

$$\delta_{ij}^{k} = \tau_c \quad (4)$$

where τ_c is a constant.

In the artificial ants framework, equation (2) is not sufficient to mimic the growing pheromone concentration in the shortest path. With actual ants, time acts as a presentation index, but the artificial ants use all the same time to perform the task, whether they choose a short way or not. For the artificial ants, it is the length l of the ways they have accepted that will determine if the solution is good or not. Thus the best solution should rise even more the pheromone concentration on the direct trail. To do so, (2) is changed to:

$$\tau_{ij}(t+n) = \tau_{ij}(t) + \rho + \Delta\tau_{ij} \quad (5)$$

where $\Delta\tau_{ij}$ are pheromones deposited in the trails (i, j) followed by all the q ants,

$$\Delta\tau_{ij} = \sum_{k=1}^{q} \delta_{ij}^{k} \times f\left(\frac{1}{z_k}\right) \quad (6)$$

And z_k is the performance index. In the CNC case the z_k can be the length $l_k = \sum d_{ij}$ of the path chosen by the k ant. In this way, the global inform is biased by the solution originate by each distinct ant. The ways followed by the ants that achieved the shortest paths have their pheromone concentration increased. Notice that the time interval engaged by the q ants to do a thorough tour is t + n iterations. A *tour* is a complete route between the nest and the food source and iteration is a step from i to j done by all the ants. The algorithm runs N_{max} times, where in every N^{th} tour, a new ant colony is released. The total number of iterations is $N_{max} \times n$. The general algorithm for the ant colonies is described in algorithm 1.

Algorithm 1: Ant Colonies Optimization
Initialization
Set for every pair (i,j): $\tau_{ij} = \tau_o$
Set N = 1 and define a N_{max}
Place the q ants
While N <= N_{max}
Build a complete tour
For i = 1 to n
For k = 1 to q
Choose the next node using $p_{ij}^{k} = (3)$
Update locally τ_{ij} (t) using (4)
Update the work list Γ
Analyze solutions
For k = 1 to q
Compute performance index $z = (l_k)$
Update globally τ_{ij} (t + n) using (5)

Ants in the Scheduling

To apply the ant colonies in the scheduling of a production system, we use the similar mathematical framework defined in the prior section. The definition of the matrices for this particular problem is:

1) The matrix d_{ij} is not the distance between the towns. When there are no sequence dependent setup-times, matrix d_{ij} is given by $d_{ij} = p_j, v_i \leq n$. When there are sequence dependent setup-times, the matrix d_{ij} is given by $d_{ij} = s_{ij} + p_j, v_i \leq n$.. Since there happens m machines, there are m matrices d_{oij}, which are the setup

times and processing times in each machine o. All the m matrices are equal if the machines are identical.

2) There exist also m matrices $\eta_{oij} = \frac{i}{d_{oij}}$ and m matrices τ_{oij}. The work lists Γ_k for each ant k will now be a matrix with dimension $(n \times m)$, keeping the information about the jobs already executed and the machine o where they were executed.

3) The cost function z is now the make span C_{max}.

The algorithm is the one presented in algorithm 1, except for the fact that in every iteration n, the ant k has to change between the m machines that means, it has to see what will be the next machine o to be available, and

Table 2. *Processing Time of Job j in Machine o*

Job j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
poj	734 8.25	1595 1.60	750 0.00	938 2.00	224 6.40	882 3.50	1709 5.00	186 3.10	1075 2.00	161 62.5	2000 0.00	3600 0.00	2998 8.00	1322 2.50	2775 0.00

```
------------------------------------------------------------
  Item |      Seq1      |       Seq2      |      Seq3    |      Seq4      |
--------------------------------------------------------------
  1        6088.55       1259.70        0.00        0.00
  8        6690.60       4779.00        7168.50        0.00
  4        9382.00       0.00        0.00        0.00
  7        11966.50       5128.50        0.00        0.00
  2        4431.00       2215.50        9305.10        0.00
  15        20750.00       7000.00        0.00        0.00
  12        6000.00       20000.00        10000.00        0.00
  5        1900.80       345.60        0.00        0.00
  3        5000.00       2500.00        0.00        0.00
  6        6302.50       2521.00        0.00        0.00
  13        16243.50       2499.00        4998.00        6247.50
  14        1763.00       4054.90        1763.00        5641.60
  11        5000.00       7500.00        7500.00        0.00
  9        7680.00       3072.00        0.00        0.00
  10        6410.35       3752.40        0.00        0.00


 Total profit before scheduling = 201590.00


 Total profit after scheduling = 302646.00


 Total time taken before scheduling = 297750.00


 Total time taken after scheduling = 234860.00>>
```

Fig. 2. *Scheduling Orders and Total Completion Time.*

RESULTS AND DISCUSSION

In this section, the proposed scheduling performances are analyzed. This is made by writing a coding in MATLAB with ant colony algorithm. The product work scheduling orders and their completion is derived from ACO fitness function (Figure 2).

The work scheduling of items and their time taken are derived in Table 3. Based on the time completion the outsourced items are scheduled using this ACO.

Table 3. *Individual Time and Product Outsourced.*

Item	Time taken	Outsourced
1	7348.25	801
8	1863.10	221
4	9382.00	309
7	17,095.00	1581
2	15,951.60	569
15	27,750.00	0
12	36,000.00	0
5	2246.40	3272
3	7500.00	0
6	8823.50	2479
13	29,988.00	2
14	13,222.50	3237
11	20,000.00	0
9	10,752.00	1928
10	16,162.5	1873

Table 4. *Earliness Timing for Individual Product.*

Item	Before	After
1	197.5	158.06
2	192.57	154.06
3	172.06	137.64
4	63.42	50.3
5	223.81	179.04
6	148.56	118.84
7	7.03	5.62
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0

Table 4 will represent the earliness timing for each product. The Table 4 will explain the timing how early that each job has been finished from due date when scheduling is not done, and how this timing has be reduced when ant colony is used for scheduling. These timing are represent in hours.

Table 5 will represent the tardiness timing for each product. The Table 5 will explain the timing about how later that each job has been finished from due date when scheduling is not done, and how this timing has be reduced when ant colony is used for scheduling. These timing are represent in hours.

Table 5. *Tardiness Timing for Individual Product.*

Item	Before	After
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	208.42	106.73
9	648.28	518.62
10	1464.21	1171.37
11	3009.46	2407.57
12	6239.36	4991.49
13	12,549.06	10039.24
14	25,428.19	20342.56
15	51,109.68	4088.74

Figure 3 shows that the graphical representation of profit vs. iterations. In this proposed system, the number of iteration increases the profit also increased. The time management improves the profit in CNC management.

Figure 4 shows that the graphical representation of objective functions vs. iterations. In this proposed system, the objective function comprises the problem of minimizing schedule time. Further it observed that the objective function decreases as their iteration increases.

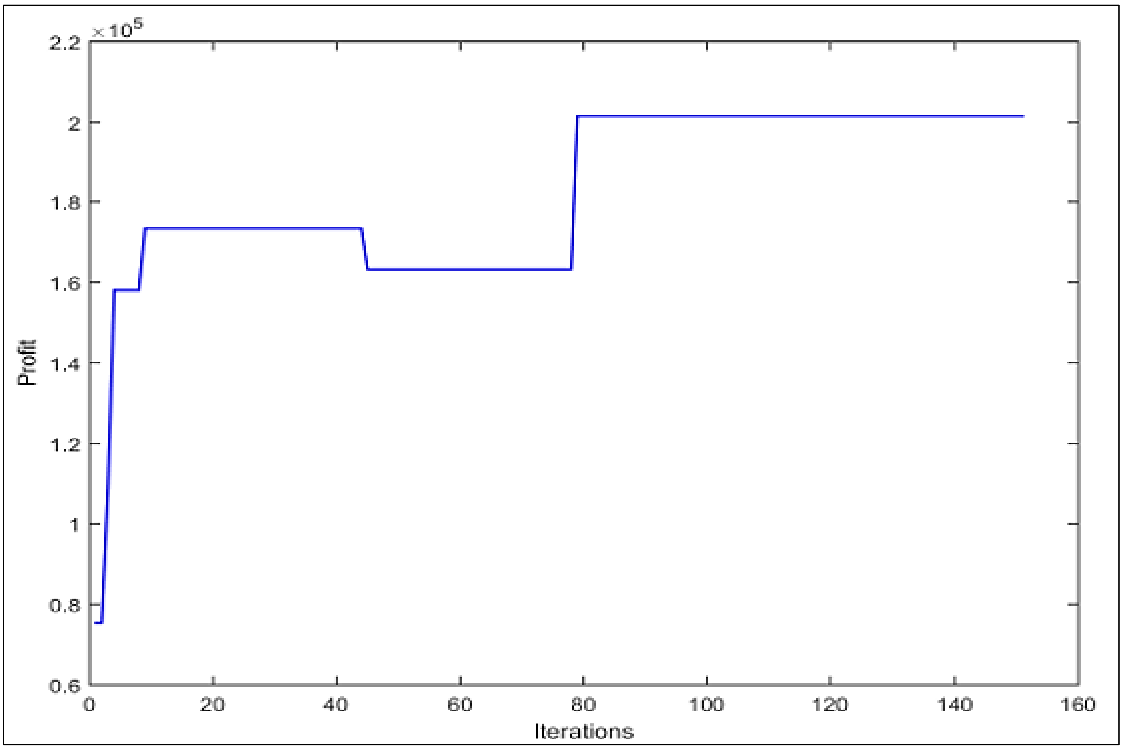

Fig. 3. *Graph with Profit Vs Iterations.*

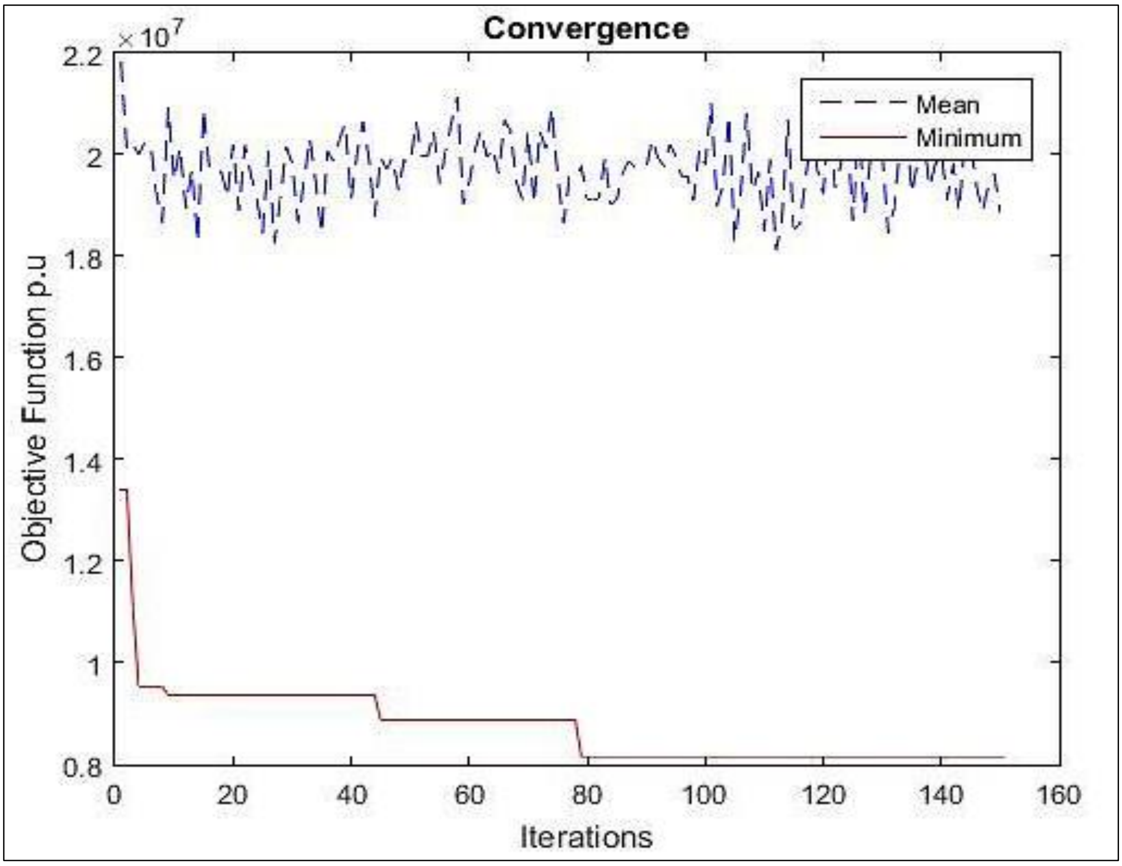

Fig. 4. *Objective Function Graph vs. Iterations.*

CONCLUSION

This paper defines a new algorithm to optimize the arranging in a parallel machine manufacturing environment. This algorithm has extends the optimization algorithm to find a solution for this new completion time problem. The ant algorithm explores both the knowledge and the knowledge of independent agents in the examine for the best solution. The ants are able to communicate between each other by means of pheromones, which have embedded the results of before attempts from other ants, enabling the merging to an optimum solution. The examples resolved in this paper showed that the algorithm is tremendously versatile. Changes in the environment are easily presented in a single matrix.

In this way, in a production method accomplished by this algorithm, variations in the manufacturing procedures do not imply a change in the scheduling method. Generally, when the environments circumstances change, the arranging method has to change also, since the schedulers are generally grow for a single issue. As the next step, we will spread the algorithm to the flow shop problem, a serial sequence of parallel machines problem.

REFERENCES

1. Bean J. Genetics and random keys for sequencing and optimization, *ORSA J Comput.* 1994; 6: 154–60p.
2. Wellman M.P., Walsh W.E., Wurman P., *et al*. Auction protocols for decentralized scheduling, *Games Econ Behav.* 2001; 35: 271–303p.
3. McKay K., Pinedo M., Webster S. A practice-focused agenda for production scheduling research, *Prod Oper Manag*. 2001.
4. Dorigo M., Maniezzo V., Colorni A. Ant system: Optimization by a colony of cooperating agents, *IEEE Trans Syst Man Cybernet-Part B.* 1996; 26(1): 29–41p.
5. Colorni A., Dorigo M., Maniezzo V., *et al*. Ant system for job-shop scheduling, *JORBEL-Belgiuan J Oper Res Stat Comput Sci.* 1994; 34(1): 39–53p.
6. Cicirello V.A. Smith S.F. Ant colony for autonomous decentralized shop floor routing, In: *Proceedings of ISADS-2001, Fifth International Symposium on Autonomous Decentralized Systems*. 2001
7. Pinedo M.. *Scheduling: Theory, Algorithms, and Systems*. 2nd Edn. Prentice Hall; 2002.
8. Hopp W.J., Spearman M.L. *Factory physics: Foundations of Manufacturing Management.* 2nd Edn, Irwin McGraw-Hill; 2000.
9. Chryssolouris G. *Manufacturing Systems, Theory and Practice.* New York: Springer- Verlag; 2006.
10. Rong Z. Dynamic Job Shop Scheduling Using Ant Colony Optimization Algorithm Based On A Multi-Agent System, *Doctoral dissertation*, 2008.
11. Pinedo M. *Scheduling Theory, Algorithms and Systems*. 2nd Edn., Upper Saddle River, New Jersey: Prentice Hall; 2002.
12. Vollmann T.E., Berry W.L., Whybark D.C. *Manufacturing Planning and Control Systems*. Irwin, Homewood, IL; 1992.
13. Ovacik I.M., Uzsoy R. Exploiting shop floor status information to schedule complex job shop, *J Manuf Syst*. 1994; 13(2): 73–84p.
14. Ovacik I.M., Uzsoy R. *Decomposition Methods for Complex Factory Scheduling Problem*. Kluwer, Dordrecht, 1997.

15. Gambardella L.M., Dorigo M. Solving symmetric and asymmetric tsps by ant colonies, In: *IEEE Conf Evolut Comp (ICEC'96)*. 1996.
16. Silva C.A., Runkler T., Sousa J.M., *et al*. Optimization of logistic processes using ant colonies, In: *Proceedings of Agent-Based Simulation 3*. 2020.