Colony Algorithm

Ahmed Hameed Ismail

Mechanical engineering dept. University of Baghdad

Ahmad.hameedqa@gmail.com

Prof. Dr. Zuheir Isaa ҩ Ahmed

ziaa12@ icloud.com

Al-Bani University College

Abstract: -

The Job shop scheduling problem is considered as an important industrial activity, especially in production planning. The objective of minimizing the maximum of the completion time is called (Makespan). The job shop scheduling considers as strongly nondeterministic polynomial-time hard named as (NP-HARD). Many methods were developed, including the Ant Algorithm to solve this problem. In this paper is used Multiple Ant colony Algorithm to solve Job Shop Scheduling, many specific features are introduced in the algorithm in order to improve the efficiency of the search. Local Pheromone update with max and min updated are used for all Ant colonies. The proposed algorithm is tested over a set of benchmark instances available in the OR-Library and give the optimal or near to optimal solution.

Keywords: Job-shop scheduling problem; Multiple Ant Colony algorithm; Local Pheromone update; Max & Min Pheromone update.

1-Introduction

Production management is a managerial work in which a very high number of decisions are chosen over time in order to guarantee the delivery of product with, minimum cost, maximum quality, and minimum lead time. These decisions different in their effect on the company, their range, and on the duration time to take them. They range from strategic, high effect, very long-range decisions like deciding whether a main factory will manufacture or not a new type of product, of short-range, low-level, small-effect decisions such as which operation is next to be made in a available machine in the shop ground. [5] In any manufacturing system in industry world cannot solve the Job scheduling problem without some assumptions. These assumptions not found in the real industry world.

a) Number of jobs are finite and fixed.

b) Each job is independent from other jobs.

c) Each job must be processed to complete and cannot be canceled.

d) All jobs arrive at the Shop at the same time.

e) Jobs must wait for the next machine to be available.

f) Number of machines are finite and fixed.

g) There is only one of each type of machine.

h) For one machine cannot process more than one operation at the same time.

i) The operation belongs to a job cannot be started until its previous operations of the same job are completed.

j) Once an operation is begun for processing, it will not be interrupted until its completion.

k) The machine doesn't break or maintained during processing time.

l) Machines may be idle within the schedule period.

m) Process time of each operation is known and fixed.

n) Set up time for each operation equal to zero or the process time include set up time.

o) Time needed to move the operations between machines is zero. [11]

In the last years an Ant Colony received high concern because of its successful applications for many optimization problems. The history of Ant Colony Algorithm (ACO) starts by applying the Ant System to solve Job-shop Scheduling Problem by Marco Dorigo and Alberto Colorni in (1994). [1] , also (ACO) effectively applied to the single machine weighted tardiness problem by Gaene´ et al in (2002), Later Blum and Sampels [3] developed the ant algorithm for shop scheduling. [14]

2. Job Shop scheduling representation

There are two types of Job Shop scheduling representation

a. Gantt chart.

Henry Laurence Gantt (1861–1919) He innovates his famous charts that still using until now during the (First World War) to get modify for production schedules. Gantt designed his charts so that employ or other supervisors could easy to know whether the production was on schedule, ahead of schedule or behind schedule. [13] A Gantt chart, or parllel chart are the same name as shown in **Fig.1** , measures activities by the value of time needed to complete them and use the space on the chart to define the value of the activity that have been done in that time. The earliest and famous kind of control chart mainly designed to display graphically the link between planned performance and actual performance. [12]

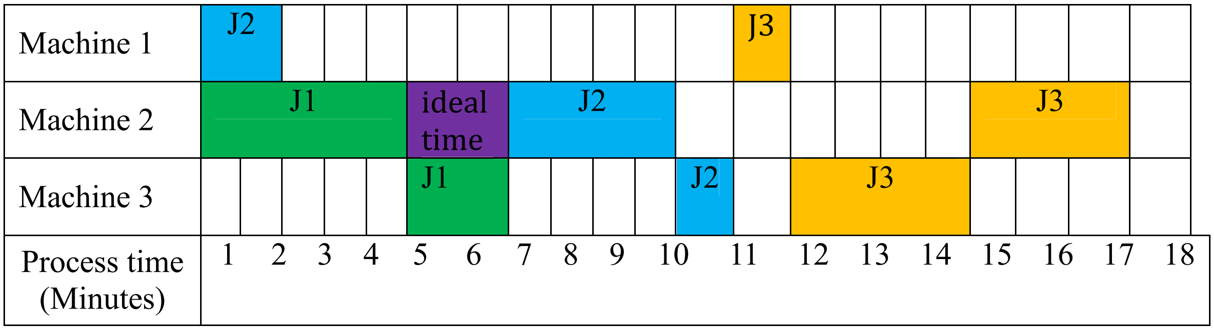


Fig. 1 Gantt chart 3 machine 3 jobs. [8]

b. The Disjunctive Graph model.

Schedules for the Job Shop Problem show as the Disjunctive Graph design and write as G= (V, C, D.) is consist of a set of nodes denoted as (V). Set of conjunctions (directed arcs) denoted as (C), the disjunctions (undirected arcs) denoted as (D) as shown in **Fig. 2**. And first submitted by Sussman and Roy in (1964). The example for a disjunctive graph of the Job Shop Problem (FT03) consist of 3 jobs and 3 machines with a specific sequence of operations and process time for each job submitted by Fisher and Thomson.[11]

$$\mathbf{G} = \begin{bmatrix} M2 & M3 & M1 \\ M1 & M2 & M3 \\ M1 & M3 & M2 \end{bmatrix} \text{Sequence order,}$$

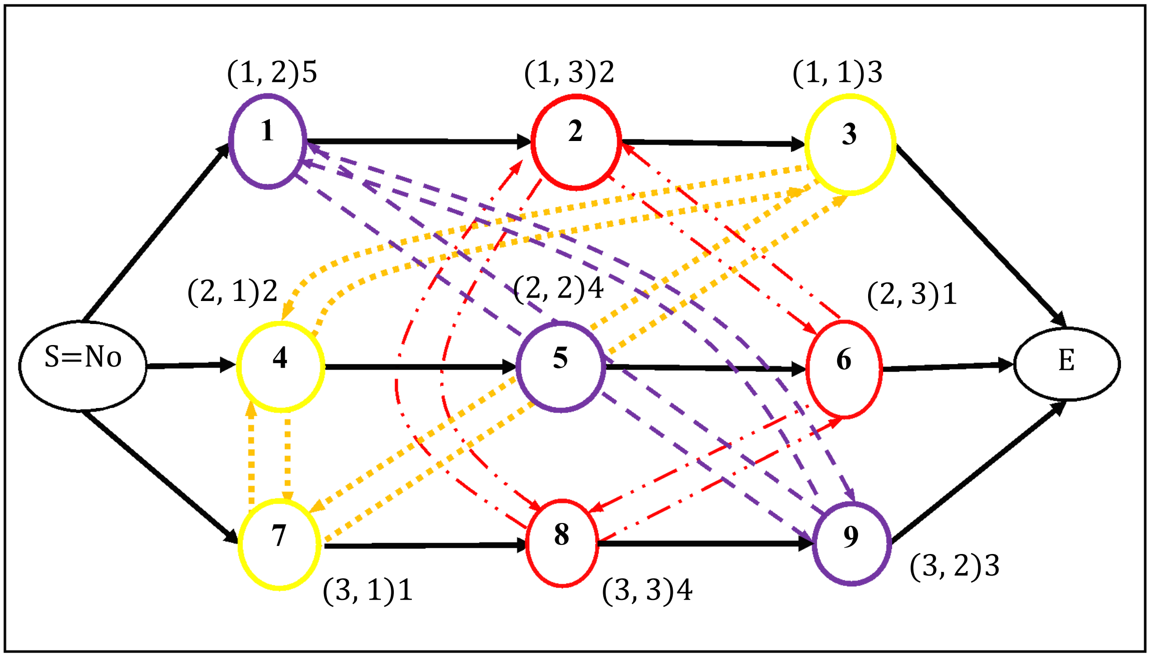
$$\mathbf{P} = \begin{bmatrix} 5 & 2 & 3 \\ 2 & 4 & 1 \\ 1 & 4 & 3 \end{bmatrix} \text{Process time minutes.}$$

a-The set [V] of nodes refer to the set of all operations. Each node (i) had weight with the conforming of processing time (P). An addition, two dummy nodes, the first node (N_O) for start node represent source this means the direct job predecessor of the first operation of each job and the second node (N+1) for end represent sink this mean the node came after the last complete operation, these two nodes has zero processing time are added to the set (with zero).

b- The set (C) of conjunctions arcs (direct arcs) refers to the precedence constraints between sequence operations of one job, the concept of a conjunction arc between (i) and (j), that, operation (i) process before the operation (j).

[Start time of operation (i) + process time of operation (i)] ≤ [Start time of operation (j)]

$S_i + P_i \leq S_j$, and conjunctions arcs between N_o the source node and first operation of each job with weight of zero c-The set [D] of disjunctions arcs (undirected arcs) refer to the operations from varied jobs had to be processed on the same machine (the machine constraint). (Pair-wise connected in both directions). [12]



(I,j) p (i)= Job number, (j) = Machine Number, p=Process time(mint)

Fig. 2 The disjunctive graph. [8]

3. Type of Job Shop Scheduling.

The diversity of scheduling theory is always finding between a sequence, a schedule and a scheduling strategy. The sequence has always dealt with the permutation of the order in which the jobs are should be accomplished on the given machine. The schedule indicates the position of jobs within a more complicated setting of machines, for many types of scheduling only sequences of the jobs are important. [5], **Fig.3** shows the respective relationships. Specific

a). Inadmissible schedules.

All feasible number of scheduling and they have excessive idle time. There is $(n!)^m$ Feasible scheduling most of these scheduling had idle time as appear in Fig.1 the operation (2) for job (2) had 2 mints of idle time that don't need it.[8]

b). Semi -active schedules.

A schedule is called semi-active when no operation can be started earlier without altering the operation sequences on any machine, the schedule cannot make improve in terms of minimizing the makespan without making changes in the operation sequence of machines. [6]

c). Active Schedules.

Schedule is called active when No operation can start earlier without delaying another operation or without violating the precedence constraints. [7] Any permissible left shift of an operation in Gantt chart did not make improve the makespan in the active schedule.

d). Non-Delay Schedules.

No machine remains idle if a job is available for processing or no machine can be kept idle if there is at least one operation which is ready for processing. Non-delay schedules build a subset of active schedules. [13]

e). Optimal schedules.

Feasible schedule in which the maximum schedule interval between any two operations is minimized. The schedule have minimum Makespan. Problems of small size, examine all active schedules and pick the optimal schedules directly from this set. But it's difficult to find optimal solution for large-size problems. To solve these problems several approximation algorithms can be used to give solution in an acceptable time have been developed.

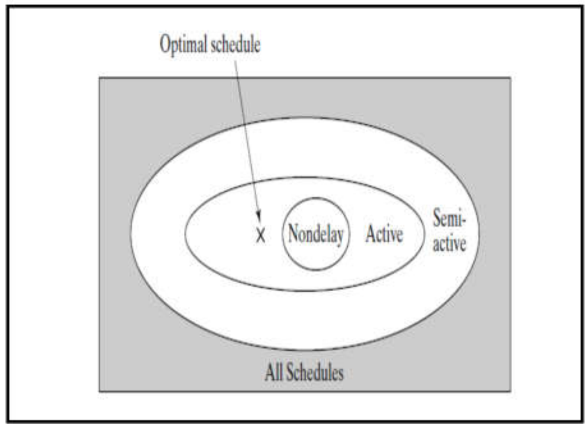


Fig.3 Type of job shop schedules. [10]

4. Definitions belonging to Job Shop Scheduling.

a) Critical path

A critical path define as a sequence of the operations that beginning from the initial operation on the initial machine (M_1) and ending with the last operation on the last machine(M_m). The finishing time of every operation on the critical path is same to the launch time of its next operation, except for the first operation that start with time equal zero, idle time along this path equal to zero, The critical path length is same to the total of the processing times of all the operations on this path and equals to C_{max}. Also sometime there are two or more critical paths in a schedule. [2]

b) Critical Operation:

Operations in which earliest start time equal to last start time. If start time is delayed, makespan will increase. [6]

c) Critical Blocks:

The all operations processed on the same machines belong to critical path it's subsequences of critical paths [7].

5. Job shop scheduling problem formulation.

$$minmize\ \boldsymbol{C_{max}} \geq S_{ij} + P_{ij} \qquad (1)$$

For all (i, j) $\in N$

Where

$\boldsymbol{C_{max}}$ The maximum of the completion times is called (Makespan).

N Set of all operations.

S_{ij} Launch time of the operation (i) belong to job (j).

P_{ij} Process time of the operation (i) belong to job (j).

Subject to

(Job constraints)

$$S_{ij} + P_{ij} \leq S_{kj} \qquad (2)$$

For all (i,j) (k, j) $\epsilon\ A$

Where (A) is set of sequence constraints $(i, j) \rightarrow (k, j)$

The Job (J) had operation on machine (i) before machine (k)

Expresses the operation precedence constraints on the job chains

S_{ij} Launch time of the operation (i) belong to job (j)

P_{ij} Process time of the operation (i) belong to job (j)

S_{kj} Launch time of the operation (k) belong to job (j)

(Machines constrains)

$$S_{il} + P_{il} \leq S_{ij} \quad \text{or} \quad S_{ij} + P_{ij} \leq S_{il} \qquad (3)$$

For all (i,j) and (i,l) ,,,i =1,…,m

Disjunctive constraints; there are some ordering happens between operations of different jobs that have to be processed on the same machine

S_{il} Launch time of the operation (i) belong to job (l)

P_{il} Process time of the operation (i) belongs to job (l)

S_{ij} Launch time of operation (i) belong to job (j)

P_{ij} Process time of operation (i) belong to job (j)

$$S_{ij}, P_{ij} \geq 0 \quad \text{For all (i, j)} \in N \qquad (4)$$

These four equation describe the Job Shop Scheduling problem. [11].

5. Ant Colony Algorithm (ACO).

In Job Shop Scheduling Problem (JSSP) main idea is proposed for how to define the job shop into a graph, then put all ants in source Node (start Node) as shown in **Fig.4** this graph had conjunctions (directional) arcs represents the precedence constraints between consecutive operations of the same job and disjunctions (bi-directional) arcs between all operations from other jobs. Arcs edge has associated of value τ_{ij} representing the amount of Pheromone trail and value of η_{ij} represent the heuristic distance between its nodes. For (JSSP) use processor time for operation instead of distance that use in travel salesman problem. The algorithm must verify that each artificial ant chooses the nodes in an order that does not violate the technological order of jobs. For example ant allowable to go from Node (2) to node (6) if the nodes (4, 5) have completed their processing as shown in **Fig.4**.

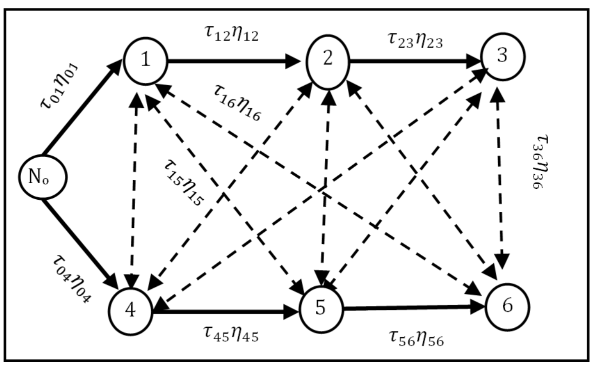


Fig.4 defines the job shop into graph. [8]

Number of nodes = (n × m) +1 (5)

n= number of Jobs,

m= number of machine

(N_0) Is dummy node represent start node or sours node.

Number of arcs

Direct arcs= n+$[n \times (m-1)]$ (6)

Indirect arcs=2$\{ \sum_{i=1}^{n} m^2\,(n-i)\}$ (7)

All arcs= n+$[n \times (m-1)] +$

$2\{ \sum_{i=1}^{n} m^2\,(n-i)\}$ (8)

6. Number of Ants

For each iteration there are K ants construct the solution. [3]

Number of ant =$\mathbf{max}\left[\mathbf{10}, \frac{[O]}{\mathbf{10}}\right]$ (16)

$[O] = n \times m$

7. Ant movement.

All artificial ants at node [N_0] first Ant move and select the next node from the space [S]. As shown in **Fig5**. Where [S] Equal the nodes that are allowed to choose and had (2) constrain, first one the technological path for the same job and the second for the candidate list. This problem small so that there is no candidate list. At Node N_0 the space [S] equal to {N1, N4, N7}

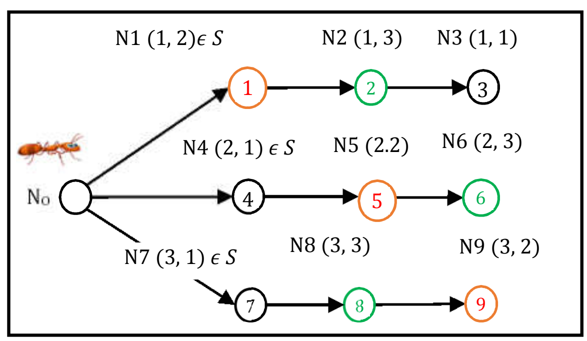


Fig. 5 Ant at start node.

After using transition rule equations and choosing N7 then [S] will be changed to {N1, N4, N8} by deleting N7 from [S] and adding N8. as shown in **Fig.6.**

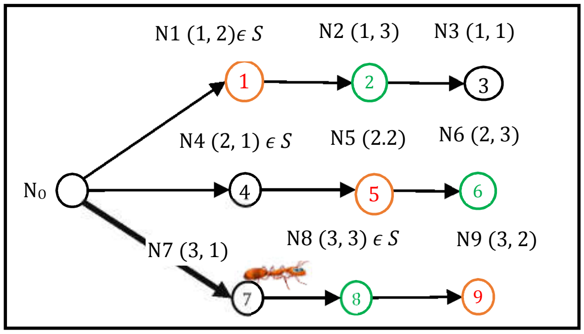


Fig.6 Ant select first node.

If an ant move to N4 then (S) will be changed again to {N1, N5.N8} by adding N5 and removing N4 from [S]. As shown in **Fig.7**.

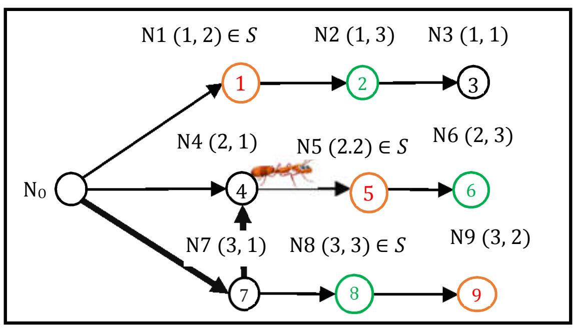


Fig.7 Ant select second node.

Notice that the machine constraints are deleted. This mean artificial ant move from N7 to N4 even this two node is for different jobs and process in different machines. If ant moving to N8 then [S] will be changed again to {N1, N5.N9} by adding N9 and remove N8 from (S) as shown in **Fig 8.**

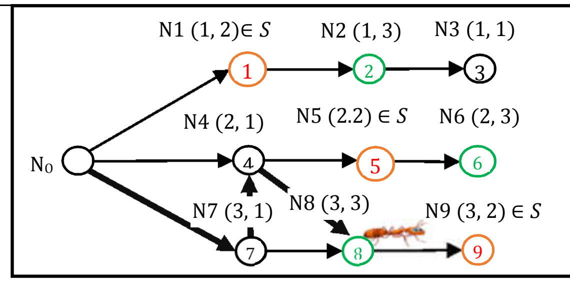


Fig.8 Ant select third node.

These steps will continue until all operations visited by ant and the tour complete and (S) is empty as shown in **Fig.9**.

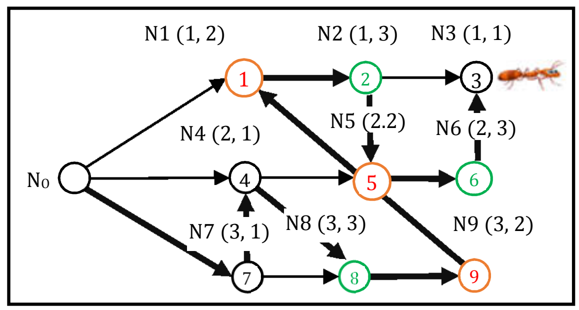


Fig. 9 Ant completes the first tour.

The Path had the sequence (N_0+N7+N4+N8+N9+N1+N2+N5+N6+N3). After completing one artificial ant tour, The Pheromone trial updated depends on the type of Ant colony Algorithm, the next ant will find the next tour and reputed the Pheromone trail update. Candidate list defined as limit the quantity of available choices, which is found at each movement step. Candidate lists have a number of the best available choices for next move according to some heuristic criteria.

8. Steps to constriction solution

i. First step:

The artificial ants put in start node, there are two equations (9, 10) are used to transition between the nodes, first one pseudorandom proportional rule (exploiting the learned knowledge). [4]

$$j = \begin{cases} \boldsymbol{argmax}_{i \in N_i^k}\{[\tau]^{\alpha}[\eta]^{\beta}\} & \text{if } q \le q_o \\ J & q > q_o \end{cases} \quad (9)$$

Where

q = Is a random variable uniformly distributed in (0, 1)

q_o = Parameter $0 \le q_o \le 1$

And the second equation equal to J Random variable selected according to the probability distribution by equation. Random' proportional rule (biased exploration of the arcs)

[Proportional transition rule (random proportional rule)]

Represent exploration. [4]

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{u \in N_i^k}[\tau_{iu}(t)]^{\alpha}[\eta_{iu}]^{\beta}} \quad \forall (i, j) \in N_i^k \ (10)$$

$P_{ij}^k(t)$ Probability to select (j) from N_i^k.

(i) It's the operation at the current step.

(j) The operation that the ant will select in the next step.

(u) The operations that belong to candidate list N_i^k

(N_i^k) Neighboring nodes to the node (i) when ant K is being at node (i) same as for (G, J, K) in **Fig.10.**

(τ_{ij}) Pheromone trail between the operation (I) and (j)

(t) Iteration

(α) Parameters that control the relative importance of the pheromone trail $(\alpha > 0)$

(β)The parameters that control the relative importance of the heuristic information$(\beta > 0)$

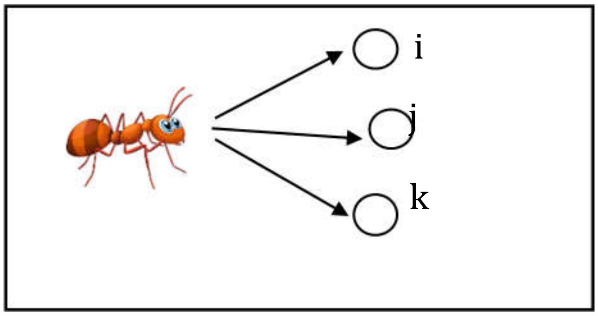


Fig.10 Ant selects the next node. [4]

If $q_o = 0$ (this convert Ant colony to Ant System).

Finding a suitable parameter $(\boldsymbol{q_o})$ Allows determining of the degree of exploration and the choice of whether to concentrate the search on the system around the best-so-far solution or to explore other tour. This selection strategy named (roulette wheel selection) because of the mechanism uses same to the roulette wheel gambler game. Each node (operation) has its percentage in the roulette wheel and the bigger this percentage had the larger width of the slot of the wheel so that the probability of choosing that node becomes larger. After a random spinning of the wheel, which is performed by generating a random number represent by (q), a slot is chosen and the next node the ant will move to be determined.

ii.Second step: Local Pheromone Trail Update.

As soon as ant crossed an arc (i,j) pheromone trail update, the rule should apply. At every time an artificial ant crossed an arc (i,j) its pheromone trail decrease, this make the arc becomes unwanted for the tracking by next ants. This make increase for the exploration of other arcs that have not been crossed yet and. Also prevent the algorithm from show a stagnation behavior. [4]

$$\boldsymbol{\tau_{ij} \leftarrow (1-\varepsilon)\tau_{ij} + \varepsilon\, \tau_o} \qquad (11)$$

Where

ε Parameter (o < ε < 1)

$\varepsilon = 0.1$ The best value. [4]

$\tau_o = {}^{1}/_{nC^m}$

n = Number of Node

C^m= Length of a nearest-neighbor tour

Pheromone update should be between the interval (0, 1)

$$\tau_{min} \leq \tau_{ij} \leq \tau_{max}$$

And $\tau_{max} = 0.999$ and $\tau_{min} = 0{,}001$ the best value of Pheromone. [3]

iii. Third step:
Global Pheromone Trail update

Best only offline Pheromone trail update after construction.

The pheromone trail update, use evaporation and new Pheromone trail deposit at same time, only use to the arcs belong to best tour. [4]

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \qquad (12)$$

ρ= evaporation rate

$$\Delta\tau_{ij}^{bs} = {}^{1}/_{C^{bs}} \qquad (13)$$

C^{bs} = Length of best path in current iteration or best iteration so far.

After Pheromone update use equation (13) to make Pheromone update between Max and Min value.

Fourth step: Feedback information.

This mean after each colony find the solution, the colony that find the best solution send information to update Master pheromone matrix then the Master pheromone Matrix send information to each colony and update the global pheromone matrix by this equation. [15]

$$\boldsymbol{\tau_{ij}^{k} = (1-w)\tau_{global\,(i,j)}^{k} + w\, \tau_{master(i,j)}^{k}} \qquad (14)$$

Where

w = is the pheromone trail important weight of master (0.7).

$\tau^{k}_{global\ (i,j)}$= pheromone trail between (i,j) in Global pheromone trail matrix

$\tau^{k}_{master(i,j)}$ = pheromone trail between (i,j) in Master pheromone trail matrix

9. Multiple ant colony.

The idea of using multiple ant colony is the used more than one colony to update master pheromone matrix. There are two type of pheromone matrix, for or each colony had owns pheromone matrix and named by global pheromone matrix, as shown in **Fig.11**, The colony find the best solution send the information to the second type of pheromone matrix named by Master pheromone matrix, this global pheromone matrix use to update the local pheromone matrix for next iteration. [14] The different in each colony either in heuristic information use or in the initial different parameter like α, β, ρ, q_o.

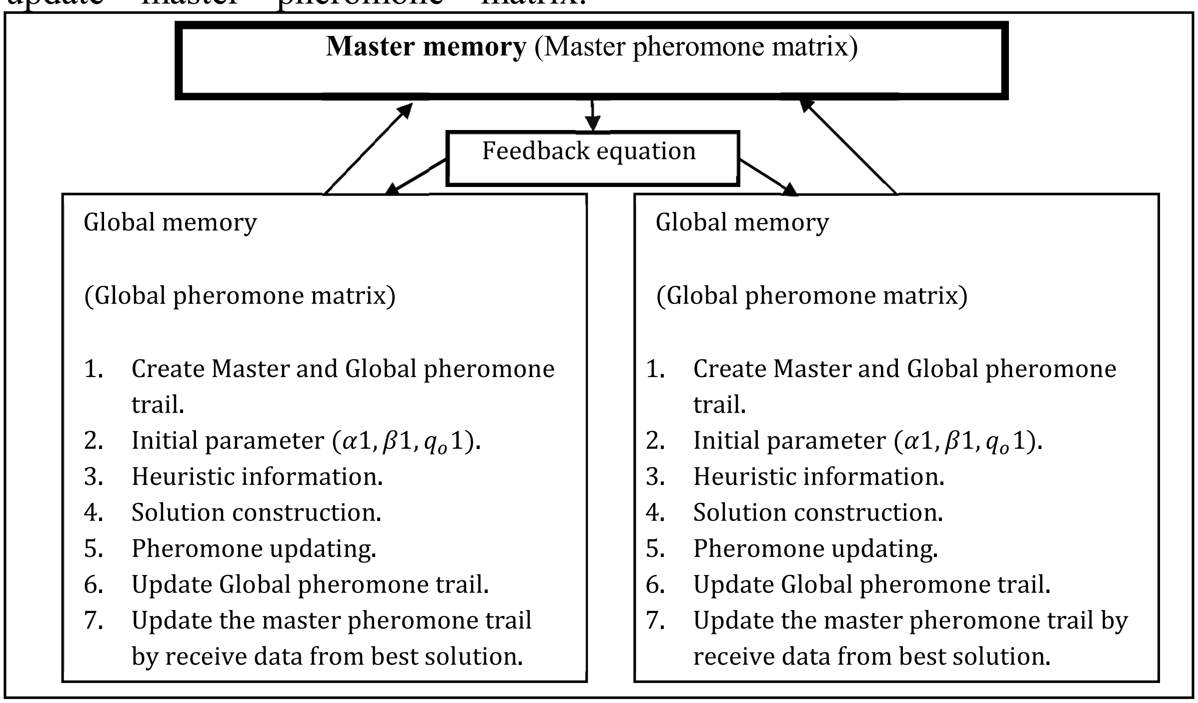


Fig. 11 Multiple Ant Colony Algorithm. [14]

10. Pseudocode for multiple ant colony.

a) For 1 to i iteration

b) For each colony from 1 to C

c) Define the variables β ρ q_o w

d) Define each arcs (i,j), $\tau_{ij} = \tau_o$,
Define arcs (I, j) $\eta_{ij} = {}^{1}/_{p_{ij}}$

e) For k=1 to All Ants

f) For N=1 to (n×m) +1

g) Find number of q and select the transition rule
h) If q > q_o chose exploration
i) If q < q_o chose exploitation
j) Determine allowable list
k) Determine candidate list
l) Determine the tour path and update the pheromone trail (local Pheromone Update)
m) Check Pheromone between Max &Min

n) Determine the tour found and convert the sequence of node by add machine constrain
o) Determine the makespan
p) End all Ants K
q) Find the best tour that find the best solution
r) Update the global pheromone trail
s) Check Pheromone between Max &Min
t) End all colony
u) Find the colony that found the best solution
v) Update the Master pheromone trail
w) Update the global pheromone by the equation of important weight of master
x) . End iteration

The code show in **Fig.12**.

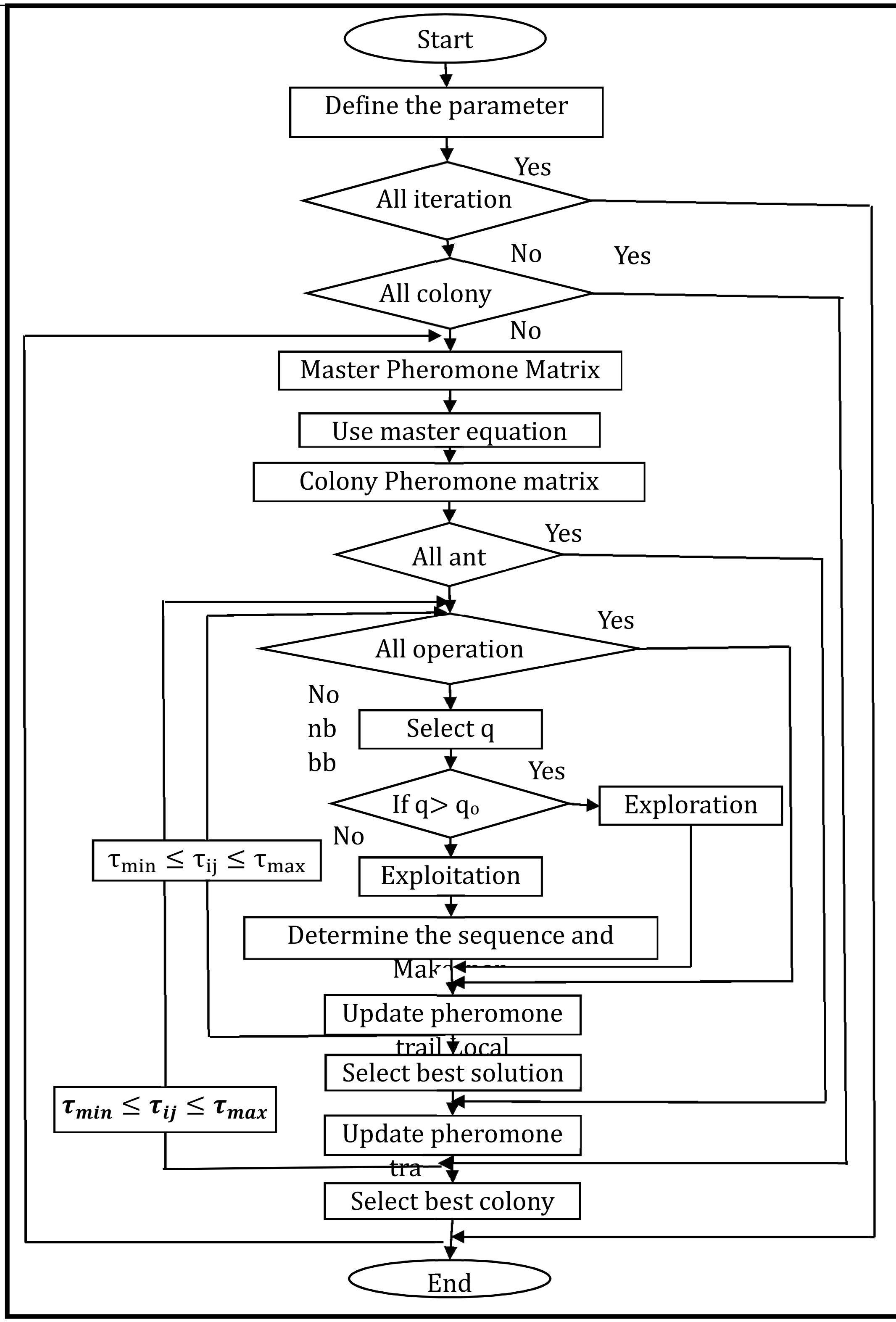


Fig. 12 Multi Ant Colony.

11. Case study

Five classic JSP problems have been tested by using the Ant Colony and Multiple Ant Colony Algorithm. In this paper, which are, Ft06, Ft10, LA01, LA16, and LA21. Available in OR-Library. [9] , FT equal to 4 problems of 4 sizes by Fisher and Thomson: FT 06 of size (6×6) with processing times generated from the interval [1, 10], FT10 and of sizes 10× with processing times generated from the interval [1, 99] , LA equal to 40 problems of 8 sizes by Lawrence (1984) with processing times generated from the interval [5, 99]: problems LA 01 to LA 05 are of size (10×5) ,And the Figures below **Fig.13,Fig.14,Fig.15** show the behavior of the algorithm, At start of Algorithm the Makespan had high value then after repeat construction solution by update Pheromone matrix the value of Makespan will decrease until get near to optimal or optimal solution. **Table 1** shows the results and the best results computed by the multiple Ant Colony Algorithm in 3 runs of the program.

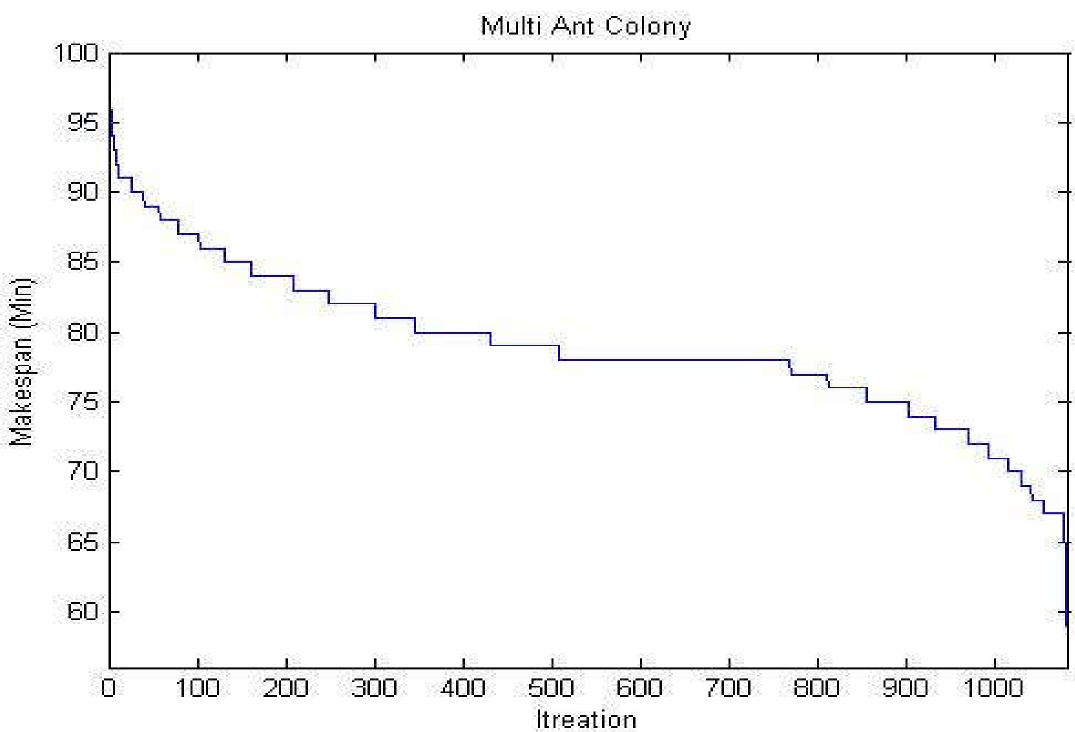
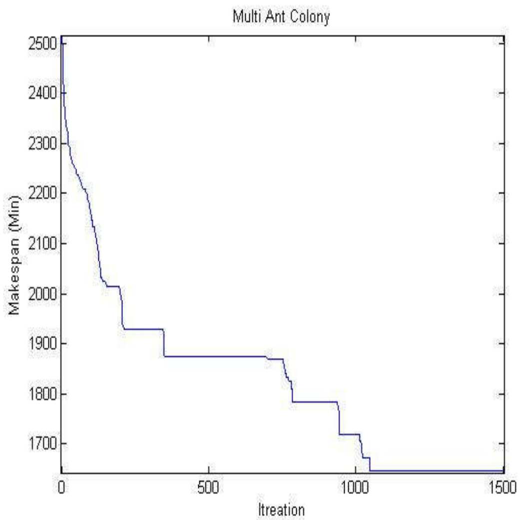


Fig.13 The result of Multi Ant Colony for Ft06



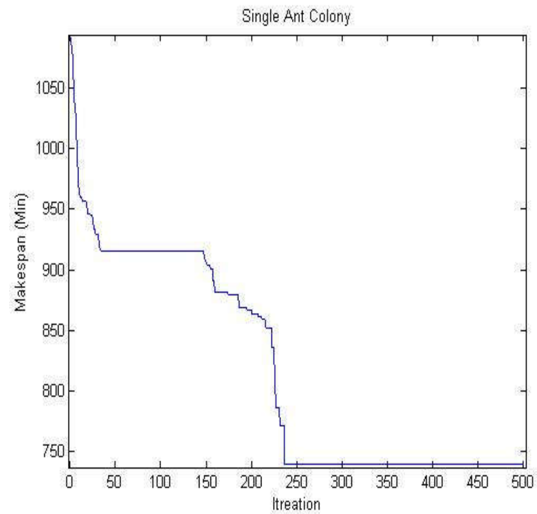


Fig. 14 The result of Multi Ant Colony for Ft10.

Fig.15 The result of Multi Ant Colony for LA01.

Table. 1 results of five problems.

Instance	Number of Jobs	Number of machines	Optimal value of Makespan (mint)	Makespan with Single Ant colony (mint)	Makespan with Multi Ant Colony (mint)
Ft06	6	6	55	63	58
Ft10	10	10	1165	1400	1300
LA01	10	5	666	738	704
LA16	10	10	949	1207	1109
LA21	15	10	1046	1640	1600

Conclusions.

It was shown that Multiple Ant colony algorithm is the best method for the optimization and solving the Job Shop Scheduling Problem. In this paper Max and Min Pheromone update and with the local Pheromone update are representing the soul of main algorithm addition to the two transmission rule. The three levels of update for pheromone with Max and Min update for pheromone use together for the first time in the Ant colony optimization and find these features have a high effect on the behavior of the algorithm. The algorithm depend only on these features of ant colony (use Local Pheromone update with Max and Min update addition to use Max and Min with Global Pheromone update) without any improve by any other algorithm. Without the effect of MAX and MIN the solution never converge to optimal solution, and the range of Max and Min Pheromone was between (0, 1) and the suitable value for Max was (0.9) and for Min was (0.01). When using vales less than (0.9) the solution doesn't reach the optimal solution. This means the Pheromone trail is direct to the optimal solution. The effect of local search also was high, for each iteration there is a local pheromone update guide the ants to explore more than one path and this technique used only in the single iteration if the use the local pheromone update throw continuous iteration the solution go away from optimal. The values of parameters control the search have high effect on the quality of the solutions finding that the evaporation rate (ρ) had strong effect and the suitable of (ρ) was (0.1). So that we change the value of the parameter for each problem to get the best solution. Because of the random use in the start of algorithm. It's important to restart the algorithm five to ten times to find best solution. For small problem the get optimal solution quality by algorithm. For multi algorithm the exchange of information between colonies allows them to advantage from the experience learned by the other colonies. And the result of the multi Ant colony better than using single ant colony. Equation number (6, 7 and 8) used to get the idea of how implement the Ant colony to Job Shop scheduling and which was used for the first time and depended by the researcher.

References.

[1] Alberto Colorni, et al (1994)' Ant system for Job-shop Scheduling' Belgian J. Oper. Res. Stat. Computer. Sci,

[2] Arno Sprecher, Rainer Kolisch, (1995)' theory and Methodology Semi- active, active and non-delay Schedules for the Resource-Constrained Project Scheduling Problem' Germany, European Journal of Operational Research.

[3] Christian, B. and Michael, S. (2004) 'An Ant Colony Optimization Algorithm for Shop Scheduling Problems' Belgium, Journal of Mathematical Modelling and Algorithms.

[4] Dorigo, M. and Stützle, T. (2004)'Ant Colony Optimization' London, England: Institute of Technology.

[5] Foramina, J. M., Leisten, R. and Ruiz García, R. (2014) 'Manufacturing Scheduling Systems' London, Springer.

[6] Giffler, A. B. and Thompson, G. L. (2013) 'Algorithms for Solving Production-Scheduling Problems - scheduling' New York. INFORMS institute for Operations Research and Management Sciences.

[7] Job, M. (2014) 'Job Shop Scheduling Solution 'Artificial Intelligence

[8] Jun Zhang, Xiaomin Hu(2006)' Implementation of an Ant Colony Optimization technique for job shop scheduling problem 'China, The Institute of Measurement and control.

[9] J.E. Beasley (1996) Obtaining test problems via Internet. Journal of Global Optimization

[10] Michael, l. P. (2009) 'Planning and scheduling in Manufacturing and Service' New York, Springer.

[11] Michael, l. P. (2016)' Scheduling Theory, Algorithms, and Systems' New York, Springer

[12] Peter Brucker (2006)' Scheduling Algorithms' Germany, Springer.

[13] Robert H.Storer, S.David Wu. (1992) 'New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling Authors' USA, INFORMS.

[14] Udomsakdigool, A. and Kachitvichyanukul, (2008) 'Multiple colony ant algorithm for job-shop scheduling problem' Thailand, International Journal of Production Research.

[15] Udomsakdigool, A. and Kachitvichyanukul,(2008) 'Multiple-Colony Ant Algorithm with Forward – Backward Scheduling Approach for Job- Shop Scheduling Problem' Thailand, Research Gate

استخدام خوارزمية مستعمرات النمل في حل مشكلة جدولة الانتاج حسب الطلب

احمد حميد اسماعيل
كلية الهندسة، قسم الهندسة الميكانيكية، جامعة بغداد
Ahmd_hameed@yahoo.com
زهير عيسى احمد
ziaa12@ icloud.com
كلية الباني الجامعة

الخلاصة

تعتبر جدولة الاعمال حسب الطلب من الأنشطة الصناعية المهمة ، خاصة في تخطيط التصنيع. وهدفها الحصول على اقل وقت ممكن لوقت الانهاء الاكبر المسمى ب(makespan) يتم من خلاله توزيع العمليات التابعة لكل منتج داخل الوحدة الانتاجية المكونة من الالات ومجموعة من العاملين او الاثنين معا ولكل منتج يوجد تسلسل خاص به من العمليات يسمى المسلك تكنولوجي وكل عملية تتطلب وقت محدد لاكمال انجازها 0 تعتبر مشكلة جدولة الاعمال حسب الطلب من المشاكل المعقدة والتي تسمى (NP–HARD) وذلك لصعوبة الحصول على الحصول على الحل الامثل خلال وقت محدد ,ولذلك هناك العديد من الاساليب التي تم تطويرها بما في ذلك خوارزمية النمل لحل هذه المشكلة. ان خوارزمية النمل التي تعتمد على السلوك الذي يستخدمه النمل الحقيقي في التموين وكيفية الحصول على اقصر طريق بين مصدر الغذاء والمستعمرة الخاصة به. وفي خوارزمية مستعمرات النمل كل المستعمرات تتعاون فيما بينها من خلال استخدام مصفوفة (Pheromone) رئيسة لتبادل المعلومات لايجاد اقصر طريق ولاستكشاف اكثر طرق ممكنة تعتمد كل مستعمرة على معلومات خاصة بها في هذا البحث المقدم تم استخدام تطويرات جديدة باعتماد على تحديث (Pheromone) بثلاث مراحل واستخدام اعلى واقل قيمة في تحديث (Pheromone). تم تصميم برنامج متكامل باستخدام الماتلاب لتنفيذ هذه الخوارزمية. اثبتت العمليات الحسابية ان الخوارزمية المقترحة ذات اداء جيد واالحصول على وقت انهاء اكبر(Makespan) اقل وهذا يودي الى توفير الوقت وتقليل كلفة الانتاج.

الكلمات المفتاحية: خوارزمية مستعمرات النمل , استخدام الية التحديث الداخلي (Pheromone) **, اعلى واقل قيمة لتحديث الفيرومون , جدولة الانتاج حسب الطلب**