SCITECH
RESEARCH ORGANISATION
Volume 13, Issue 3
Published online: August 15, 2018|

Journal of Progressive Research in Mathematics
www.scitecresearch.com/journals

Local search Methods to Solve The Sum of Two Objective Functions

Al- Zuwaini Mohammed Kadhim[1], *Sarah Kamil Hanoon*[2]
[1]Mkzz50@ yahoo.com [2]ssaa.kamal@ yahoo.com
Department of Mathematics, College of Computer Science and Mathematics
Thi-Qar University, Thi-Qar, Iraq

Abstract

In this paper, the problem of sequencing a set of n jobs on single machine was considered to minimize theobjective function. The aim is to find the optimal or near optimal solution (scheduling) for the objective function consists of a sum of total late work and maximum lateness. This problem is strongly NP-hard. Simulated Annealing, Ant colony Algorithm, and usagea hybridization as a tool to solved the problem approximatelywith up to 100000 jobs in a reasonable time 10 minutes.

Keywords

Single machine, Scheduling, Simulated Annealing, Ant colony,Hybridization**.**

1. Introduction

In this paper, we tackle a problemof scheduling n independent jobs on a single machine to minimize the sum of total late work and maximum lateness which is denoted by $1 \,//\, \sum_{j=1}^{n} V_j + L_{max}$. Clearly that our problem consist of two sub problems. The first subproblem $1 \,//\, \sum_{j=1}^{n} V_j$ which is NP-hard see[1, 2, 3, 4, 5]; Manal (2012)[6] also solved single machine scheduling problem to minimize the sum of total completion times and total late works; Asmaa (2015) [7] solved scheduling jobs on a single machine to minimize the sum of tardy jobs and total late work. And the second subproblem the maximum lateness ($1 \,//\, L_{max}$) which is p-type and studied by Jackson (1955) [8], Jackson proved that L_{max} was solved by EDD rule. Many researchers have studied this problem see [9, 10, 11, 12, 13, 14, 15].

2. Problem Formulation

A set of n independent job $\{j_1, j_2, \dots\}$ be scheduled on a single machine with the conditions only one job j can be processed at a time. All jobs are available

for processing at time zero, the precedence relation among the jobs are not supposed and preemption is not allowed. The goal is to find a processing order of the jobs that minimize the sum of the total late work and maximum lateness, i.e solve the $1//\sum_{i=1}^{n} V_i + L_{max}$ problem.

Each job j has positive integer processing time (p_j) and due date (d_j). Fora given schedule the completion time $(C_j = C_{j-1} + p_j, and\ C_0 = 0)$,the late work$V_j = \min\{T_j, p_j\}$ and maximum lateness$L_{max} = \max\{L_j\} = \max\{C_j - d_j\}$, $j = 1,2, \dots n$were computed. The objective is to find the schedule $\sigma = (\sigma(1), \dots, \sigma(n))$ of the job that minimize the total cost which is formulated in mathematicsforms as:

$$\left.\begin{aligned} & R = min_{\sigma\in\delta}\left\{\sum_{j=1}^{n} V_{\sigma(j)} + L_{max\,(\sigma)}\right\} \\ & subject\ to: \\ & C_{\sigma(j)} \geq p_{\sigma(j)} \qquad j = 1, \dots, n \\ & V_{\sigma(j)} \leq T_{\sigma(j)} \qquad j = 1, \dots, n \\ & V_{\sigma(j)} \leq p_{\sigma(j)} \qquad j = 1, \dots, n \\ & L_j = C_j - d_j \qquad j = 1, \dots, n \\ & p_{\sigma(j)} > 0 \quad , \quad d_{\sigma(j)} > 0 \quad , V_{\sigma(j)} \geq 0 \end{aligned}\right\} \dots (P)$$

where δtheset of all feasible solutions and $\sigma(j)$ denotes the position of job j in the ordering σ.

3. Problem Decomposition

In this section, the problem (P) is decomposed into two sub problem (SP_1) and (SP_2) which are simple structure of the original problem as follows :

(i) The total late work $(1//\sum V_j)$ is NP-hard [1]

$$\left.\begin{aligned} & R_1 = min_{\sigma\in\delta}\left\{\sum_{j=1}^{n} V_{\sigma(j)}\right\} \\ & subject\ to \\ & C_{\sigma(j)} \geq p_{\sigma(j)} \qquad j = 1, \dots, n \\ & V_{\sigma(j)} \leq T_{\sigma(j)} \qquad j = 1, \dots, n \\ & V_{\sigma(j)} \leq p_{\sigma(j)} \qquad j = 1, \dots, n \\ & p_{\sigma(j)} > 0, \ d_{\sigma(j)} > 0, \ V_{\sigma(j)} \geq 0 \end{aligned}\right\} \dots\dots\dots (SP_1)$$

ii. The maximum lateness (1 \\ L_{max}) which is solved by the EED rule [8] and the formulation form can be given as:

$$\left.\begin{array}{ll} R_2 = min_{\sigma(j)}\{L_{max}\} & \\ subject\ to & \\ C_{\sigma(j)} \geq p_{\sigma(j)} & j = 1, \dots, n \\ L_{\sigma(j)} = C_{\sigma(j)} - d_{\sigma(j)} & j = 1, \dots, n \\ d_{\sigma(j)} > 0, \quad p_{\sigma(j)} > 0, & j = 1, \dots, n \end{array}\right\} \dots\dots (SP_2)$$

Theorem (1) [16]

R_1+R_2 ≤ R where R_1, R_2 and R are the minimum objective function values of SP_1, SP_2 and P respectively.

4. Special Cases

A special case means finding an optimal solution without using mathematical programming techniques.

Case (1)

If EDD rule satisfies$c_j < d_j$ for each job j in EDD then EDD rule gives the optimal solution for problem 1/ /$\sum_{j=1}^{n} V_j + L_{max}$.

Proof:-

Since $c_j < d_j \forall$ j in EDD Hence, no job will be tardy i.e. T_j=0 ∀ j and $\sum_{j=1}^{n} V_j$ =0 then the problem 1/ $c_j < d_j$ / $\sum_{j=1}^{n} V_j + L_{max}$ depend on 1/ $c_j < d_j / L_{max}$ which was already solved by EDD [8]. So EDD gives an optimal solution for 1/ /$\sum_{j=1}^{n} V_j + L_{max}$.

Case (2)

If p_j=p , d_j=d , j=1,….,n then the problem 1/ p_j=p, d_j=d /$\sum_{j=1}^{n} V_j + L_{max}$ is independent on a schedule.

Proof :-

Since p_j=p , thenc_j=jp ∀ j for any schedule, andV_j=min{$max(jp - d, 0), p$} and since d_j=d then any order gives the minimumL_{max}, hence the problem 1/ p_j=p, d_j=d /$\sum_{j=1}^{n} V_j + L_{max}$ is independent on a schedule.

Case (3)

If there is a schedule π satisfyingp_j=p and $d_j = jp$ $\forall j \in \pi$, π=(1,……,n) then π is optimal solution for problem 1/ / $\sum_{j=1}^{n} V_j + L_{max}$.

Proof:-

Since $d_j = jp$ and $p_j = p$ therefor $c_j = d_j$ then π gives just in time for each job $j \in \pi$ and $\sum_{j\in\pi}^{n} V_j + L_{max} = 0$.

Case (4)

. If $d_j = d \ \forall$ j, j=1,…, n then any schedule give an optimal solution for the problem $1//d_j = d / \sum_{j=1}^{n} V_j + L_{max}$.

Proof: -

First, From the condition of due date, then any order of jobs gives an optimal solution for problem $1/ d_j = d / L_{max}$

Second, since $d_j = d \ \forall j$ then for any ordering of the jobs the total late work equal to $\max\{\sum_{j=1}^{n} p_j - d \ , 0\}$[1]. Hence, any schedule gives an optimal solution for the problem $1/ d_j = d / \sum_{j=1}^{n} V_j + L_{max}$.

Case(5)

If $L_{max}(EDD) = \sum_{j=1}^{n} V_j$(EDD) then EDD rule gives an optimal solution for the problem $1/ /\sum_{j=1}^{n} V_j + L_{max}$.

proof:-

Since $L_{max}(EDD) = \sum_{j=1}^{n} V_j$(EDD) and a schedule which is optimal with respect to L_{max} is also optimal with respect to T_{max} [18], then $T_{max}(EDD) = \sum_{j=1}^{n} V_j$, and $T_{max}(EDD)$ is a lower bound for total late work $\sum_{j=1}^{n} V_j$ [1] i.e $T_{max}(EDD) \leq \sum_{j=1}^{n} V_j$. Then (EDD) rule is optimal for $1/ /\sum_{j=1}^{n} V_j$.Then (EDD) rule is an optimal solution for the $1/ / \sum_{j=1}^{n} V_j + L_{max}$.

5. Derivation of Lower Bound (LB)

The lower bound for the problem (P) is based on decomposing (P) in to two sub problems $(SP_1)$and (SP_2) as shown in section (2.3). R_1 and R_2 was calculated to be the lower bound for SP_1 and SP_2 respectively, and thus applying theorem (1) gives lower bound (LB) for the problem (P).

For subproblem (SP_1), $R_1 = T_{max}(EDD)$ is a lower bound [1] while for subproblem (SP_2), $R_2 = L_{max}(EDD)$ is a lower bound [8].

and LB = $(R_1 + R_2) = T_{max}(EDD) + L_{max}(EDD)$

Theorem(2)

If i and j unscheduled jobs with $d_i < d_j$ and p_i=p for any job then job i≺ j in optimal solution for problem(P).

Proof:-

Let S_k partial sequence which its schedule, $k \subset N$ for $i,j \in \bar{k} = N - k$, and (S_k ,i ,j) be a schedule which is obtained by interchanging job i and j in (S_k ,j ,i), then all job other than i and j has the same completion time in (S_k ,i ,j) as in (S_k ,j ,i). Since potts and van wessanhove [1] prove that EDD rule gives as optimal for 1/ /$\sum_{j=1}^{n} V_j$ proved that p_i=p $\forall i$ then $\sum_{j \in S_k ,i ,j} V_j \leq \sum_{j \in S_k ,j ,i} V_j$ since $d_i < d_j$.

Also by EDD rule $L_{max}(S_k , i , j) < L_{max}(S_k , j , i)$, hence i≺j in optimal solution for problem (P).

Theorem(3)

If $d_j = \max\{d_i\}$, i=1,…,n and $d_j \geq$ t where t= c_{max} then there exists an optimal sequence such that job j is ordered last job.

Proof:

since $d_j \geq c_{max}$ then job j is early job, and V_j=0 and by condition $d_j = \max\{d_i\}$, and EDD rule gives an optimal for 1/ / L_{max} hence there exists an optimal sequence such that job j is ordered last job.

Theorem(4)

If δ_k partial sequence which it is job are schedule K ⊂ N for i , j ∈N - k and let c be a completion time of the last job in δ_k if $d_i \leq d_j$ and c+p_i+p_j $\leq d_i$ then i ≺ j inan optimal solution fortheproblem (P).

Proof:

Since $d_i \leq d_j$ and c +p_i+p_j $\leq d_i$ then jobs i and j are be early and $V_i = V_j = 0$, since EDD rule gives an optimal for L_{max} then i ≺ j in an optimal solution for the problem (P).

6. Ants Colony optimization (ACO) Algorithm

In the early 1990s, ant colony optimization (ACO) [19] was introduced by M. Dorigo and his colleagues as a novel nature-inspired meta heuristic for the solution of hard combinatorial optimization (CO) problems Ant Colony Optimization is a part of the larger field of a swarm intelligence technique through which scientists studied the behavior patterns of termites, "ants and other social insects [20] and inspired by the behavior of ants in their search for food also. The ants colony algorithm is an experimental and error-based search algorithm that gives an acceptable solution (may

be the best solution and may not be) so it is used to solve long-time problems using a computer such as NP-Complete or questions that need to be tried all the possibilities even reach the desired solution if found. Also the ant colony studied by Ali and Farouk in 2013 [21],Araibi 2017 [22]andHussametal[23].

6.1Algorithm Idea

The idea of algorithm came from simulating the process of searching of food in the ants are as follows:

1. A group of ants start from the cell in several random directions (this process is done only in the first time in subsequent times, each track is tested and a certain path chosen as we will see later).

2. During the passage of the ant it secrete a material called pheromone in a certain percentage.

3. When it find a source of food, it takes a quantity of it and returns to the cell by choosing a certain path (the path that contains the largest amount of pheromone). During it return it secrete the same amount of the pheromone.

4. When the ant starts from the cell again, it will test the amount of pheromone in each path and choose the path that contains the largest amount of pheromone. (during the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food, the pheromone trails will guide other ants to the food source [24]).

5. The amount of pheromone is updated every time period (the concentration of the pheromone fades over time. the ants' life is millions of years .

Note that the shortest path will always contain the largest amount of pheromone and therefore all the ants will pass it. The advantage of this algorithm is that it is dynamic in the sense that if there is an obstacle in the shortest path, the ants will choose a new path in the same way.

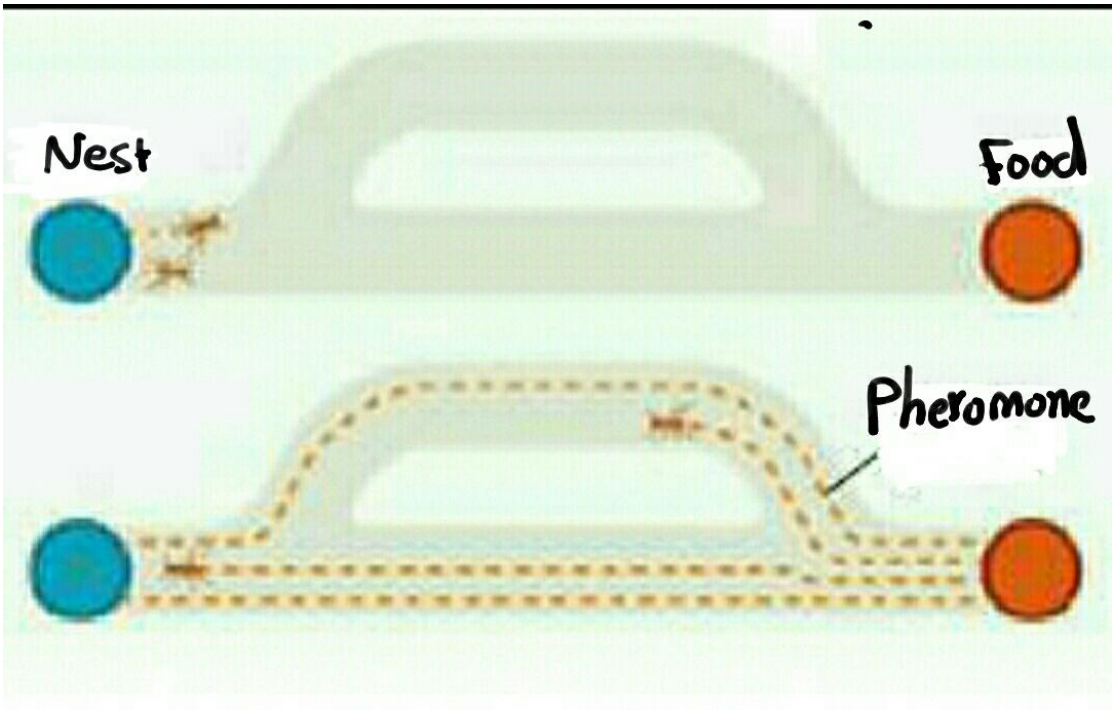

Fig. (1) Find the shortest path by ants

Steps of the ants algorithm [25]:

Step1. Set pheromone trails to be small constant.

Step2. While (termination condition not met).

Step3. Place each ant on initial node (its index usually).

Step4. Repeat.

Step5. For each ant do.

Step6. Choose next node with Apply State Transition Rule.

Step7. End for.

Step8. Until "each ant builds one solution".

Step9. Choose the best solution.

Step10. Apply Local Update pheromone.

Step11. Apply Global Update.

Step12. End White.

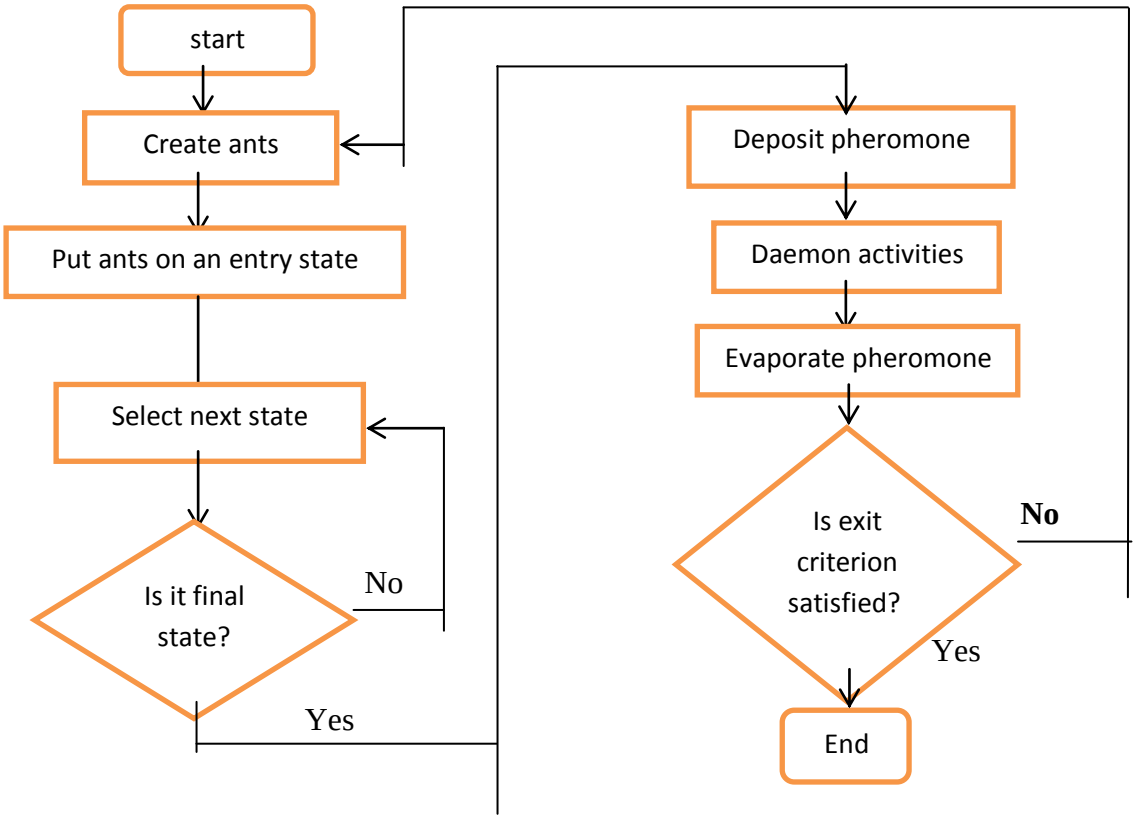

Fig. (2): Flowchart of ACO algorithm

6.2 Actual Ant

In the experiment "Linepithema humile", Deneubourg et al. (1990) on the argentine ant, which provides a clear demonstration of how to organize the ants by the effects of pheromones and called this experiment, the double bridge is two paths, one of which twice the length of the other and both used for crossing between the ant site and food source in minutes, the ant Choose in the most cases of the shorter branch. This has been achieved by the ants that leave marks of pheromone and followed by other ants in their search for food.This has been achieved by the ants that leave marks of pheromone and followed by other ants in their search for food. Obviously the faster ants that go back to the nest are which followed the shorter road back and forth. Although ant species are almost blind, they can still communicate with the environment and with each other by means of substances they release

7. Simulated Annealing (SA) Heuristics

Simulated annealing algorithm is a well-known neighborhood search approach, deriving its acceptance mechanism from annealing process in order to let the current solution escape from the local optima. It starts with an initial solution and navigates

around it by any kind of neighborhood search structure. In order to escape from local optima, leading ultimately to the optimum solution. In generating this path, solutions are chosen from the locality of the preceding solution by a probabilistic function of the improvement gained by this move. So, steps are not strictly required to produce improved solutions, but each step has a certain probability of leading to improvement; at the start all steps all equally likely, but as the algorithm progresses, the tolerance for solutions worse than the current one decreases, eventually to the point where only improvements are accepted. In this way, the algorithm can attain the optimum solution without becoming trapped in local optimal.

This tells us that we can have some confidence in the answer generated by a well formulated simulated annealing implementation, but says nothing about how long it might take to find that answer. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities).

Hence, one finds a few references in the literature that addresses Simulated Annealing algorithm. (2009) Seyed, Hashemi and Khatibi using the simulated annealing method to solve an problem on more than one machine is a matter of type NP-hard in a reasonable computation time[26],(2013)Sergio, Fulvio, Antonio and Alberto the simulated annealing algorithm was used to solve an problem of the NP-hard type with release dates [27], (2014) Nakandhrakumar and Balachandar They found approximate solutions to solve the Job Shop Scheduling Problem by simulated annealing method [28].

The following structure gives the outline of (SA):

Algorithm[29]**:**

Let 's term

S_c : Candidate schedule

S_o : Best schedule found so far

S_k : schedule constructed at K iteration (K= iteration counter)

$G(S_o)$: Value of best schedule

$G(S_k)$: Value of schedule constructed at K iteration

$G(S_c)$: Value of candidate schedule

$$rand(x) < \exp(\frac{G(S_c) - G(S_k)}{\beta_k})$$

Where, x is a random variable having uniform distribution U[0; 1].

β_k is called cooling parameter in annealing terminology usually $\beta_k = a^k$, where $a \in [0,1]$.

Step (1): Initialize

Set K=1

Let S_k be a randomly create sequence

Let $S_k = S_o$

Then $G(S_k) = G(S_o)$

Step (2): Generate a perturbed sequence $\boldsymbol{S_c}$with one of the neighborhoods operators and set $k = k + 1$.

Step (3): Evaluate the $G(S_c)$ values.

Step (4): If $G(S_c) < G(S_o)$, then let $S_k = S_c$, $G(S_o) = G(S_c)$. Set $k = k + 1$ and go to step **7**.

Step (5): Generate a random number $\boldsymbol{x}$.

Step (6): If $rand(x) < \exp(\frac{G(S_c)-G(S_k)}{\beta_k})$, then let $S_k = S_c$, $K = N$.

Step (7): If $k < N$, then go to step **2**.

Else

Let $\beta_{k+1} = a\beta_k$ and $k = 0$.

Step(8): Stop best value of S_o stored in best.

8. Hybridization algorithms

Meta-heuristics are used to solve with the computationally hard optimization problems. Meta-heuristics consist of a high level algorithm that guides the search using other particular methods. Meta-heuristics can be used as a standalone approach for solving hard combinatorial optimization problems. But now the standalone approach is drastically changed and attention of researchers has shifted to consider another type of high level algorithms, namely hybrid algorithms. There are at least two issues having to be considered while combining more than one meta-heuristics: (a) how to choose the meta-heuristic methods and (b) how to combine the chosen heuristic methods into new hybrid approaches. Unfortunately, there are no theoretical foundations for these issues. For the former, different classes of search algorithms can be considered for the purposes of hybridization, such as exact methods, simple heuristic methods and meta-heuristics. Moreover, meta-heuristics themselves are classified into local search based methods, population based methods and other classes of nature inspired meta-heuristics. Therefore, in principle, one could combine any methods from the same class or methods from different classes [30], Hence, one finds a few references in the literature that addresses Hybridization algorithms, (2006)

Mohammad Kadhim He studied the hybridization of genetic with Simulated to solve the problem [31], (2010) A. Jamili, M.A. Shafia, R. Tavakkoli-Moghaddam They hybridized Simulated with electromagnetism-like mechanism to solve a periodic job shop scheduling problem [32], (2016) KashifAkram etal. are using simulated annealing hybridized with quenching for solving job shop scheduling problem [33]. In this chapter, we will study hybridization for the two approximate methods (Ants Colony, Simulated Annealing) as follows:

- Hybridization the Ants Colony with Simulated Annealing(HASA).
- Hybridization the Simulated Annealing with Ants Colony(HSAC).

8.1 Hybridization the Ants Colony with Simulated Annealing (HASA)

The use of the ants algorithm in order to find a preliminary solution or candidate schedule of the problem and then this solution is improved by introducing it in the simulated algorithm, i.e (Hybridization of ants algorithm with simulated algorithm), An efficient scheduler is to be obtained at a reasonable time.

8.2 Hybridization the Simulated Annealing with Ants Colony (HSAC)

Hybridization of the Ants algorithm by sequential taking as an initial solution from the simulated algorithm and then inserted with the Ants algorithm to improve the first solution.

9.Computational Results of Local Search Algorithms and Comparison

9.1 Test Problems

The data were generated in this chapter in the same way as in [34]that generates as follows:

- The processing time p_j is uniformly distributed in the interval [1,10].
- The due date d_i is uniformly distributed in the interval

[P(1-TF-RDD/2),P(1-TF+RDD/2)]; where $P = \sum_{j=1}^{n} p_j$

depending on the relative range of due date (RDD) and on the average tardiness factor (TF).

For both parameters, the values 0.2, 0.4, 0.6, 0.8 and 1.0 are considered. For each selected value of n (where n is the number of jobs), ten problems were generated.

9.2 Computational Results

The local search algorithms in this chapter Ants Colony Algorithm, Simulated Annealing Heuristics andCats Swarm Algorithm, are coded in MATLAB 8.3.0 (R2014a) and implemented on Intel (R) Core(TM) i3-6100U CPU @ 2.30 GHZ, with

RAM 4.00 GB personal computer. In our computation, we use the condition that: if the solution of an example with " n " jobs for any algorithm does not appear after 600 seconds, i.e. (10 minutes) from its run; then this example is unsolved and this algorithm is active until the problem of size " n"

9.2.1 Comparative Effective of Local Search Algorithms

Table (1) shows for each algorithm the value of the objective function and how many it can catch the optimal value for each value of " n " (problem size). In addition, it describes the deviation of local search methods fro the optimal solution. The optimal solution for examples in table (1) was found by using complete enumeration (CE).

Table (2) shows the values of each local search algorithms and how many times that each of them catch the best value. where:

Opt= the optimal value.

ACO= the value found by Ants Colony Algorithm.

SA= the value found by Simulated Annealing.

HASA = the value found by HASA algorithm.

HSAC = the value found by HSAC algorithm.

No of opt.= number of examples that catch the optimal value.

Av. of Time = the average of time for (10) examples for each size.

Av. of Cost = the average of cost for (10) examples for each size.

/ = refer to the unsolved example.

Table(1): The performance of local search methods and the optimal solution for $n \in \{5, 10\}$

n	Ex	Opt	SA	ACO
5	1	30	31	34
	2	41	41	41
	3	22	22	22
	4	32	32	32
	5	25	25	25
	6	31	31	33
	7	29	30	30
	8	57	57	59
	9	40	40	47
	10	15	15	24
No. of opt.			8	4
Av. of time			0.1153	0.0012
Av. of cost			32.4	34.6
10	1	121	121	122
	2	67	67	79
	3	33	33	70
	4	76	76	76
	5	22	22	43
	6	70	70	76
	7	27	27	36
	8	122	122	125
	9	57	58	68
	10	54	54	65

No. of opt.	9	1
Av. Time	0.1285	0.0018
Av. of cost	65	76.6

Table (2): The performance of local search methods for problem (P) withn ∈ {50,100,200,500,1000,2000,5000,10000,15000,20000,30000,50000,100000}

n	SA		ACO		HASA		HSAC	
	Av. of cost	Av. of time	Av. of cost	Av. of time	Av. of cost	Av. of time	Av. of cost	Av. of time
50	263.4	0.1797	391.3	0.0221	263.9	0.2181	265.4	0.3915
100	470.9	0.2513	782.9	0.1084	471	0.4747	473.2	0.6099
200	1051.3	0.4042	1686.8	0.6430	1056.4	1.6799	1058.3	1.4534
400	1763	0.7304	3331.4	4.8082	1764.4	10.3751	1755.4	6.2230
500	2534.6	0.8836	4175.3	8.9643	2534.8	18.8087	2529	10.6705
1000	5781.8	1.7305	8782.1	72.3901	5796.4	146.3028	5765.8	75.7601
2000	11013	3.5265	/	/	/	/	/	/
5000	25899	9.0975	/	/	/	/	/	/
10000	67369	18.4783	/	/	/	/	/	/
15000	93634	23.0555	/	/	/	/	/	/
20000	117930	30.9836	/	/	/	/	/	/
30000	142000	49.2750	/	/	/	/	/	/
50000	374290	86.9954	/	/	/	/	/	/
100000	660240	213.4520	/	/	/	/	/	/

9.2.2 Summary of Experimental Evaluation of Local Search Methods

In the table (3), we summarize the results of table (1) by viewing how many times that the algorithm catch the optimal value only, and their sum, for each number of jobs and for the two local search methods.

Table (3): summary of results of table (1)

n	SA	ACO
5	8	4
10	9	1
Sum	17/20	5/20

In the table (4), we summarize the results of table (2) by viewing how many times that the algorithm catch the best value only, and their sum. For each number of jobs for the four local search algorithms.

Table (4): summary of results of table (2)

n	SA	ACO	HASA	HSAC
50	9	0	7	7
100	4	0	4	3
200	5	0	3	2
400	3	0	5	5
500	5	0	1	4
1000	6	0	0	5
2000	10	/	/	/
5000	10	/	/	/
10000	10	/	/	/
15000	10	/	/	/
20000	10	/	/	/
30000	10	/	/	/
50000	10	/	/	/
100000	10	/	/	/
Sum	112/140	0	20/140	26/140

In the table (5), we give the activity of local search algorithms, (i.e. give the maximum number of jobs " n ") that the local search algorithms can solve the (1/ / $\sum_{i=1}^{n} V_i + L_{max}$) problem with reasonable time, (i. e. according to the condition that had been given in subsection (9.2)).

Table (5): shows activity of the local search methods

Algorithm	Active until (maximum no. of jobs)
SA	100000
ACO	1000
HSAC	1000
HASA	1000

From a comparative study of the introduced local search methods and precise vision of all the above tables, we found that for the (1/ / $\sum_{i=1}^{n} V_i + L_{max}$) problem the SA algorithm has obtained the best results and used hybridization methods to try to improve solutions, we note that the hybridization algorithms (HASA, HSAC) have improved the results of the ACO algorithm, whereas the hybridization algorithm (HSAC) has improved the results of the SA algorithm when n $\geq$ 400 see figure (3).

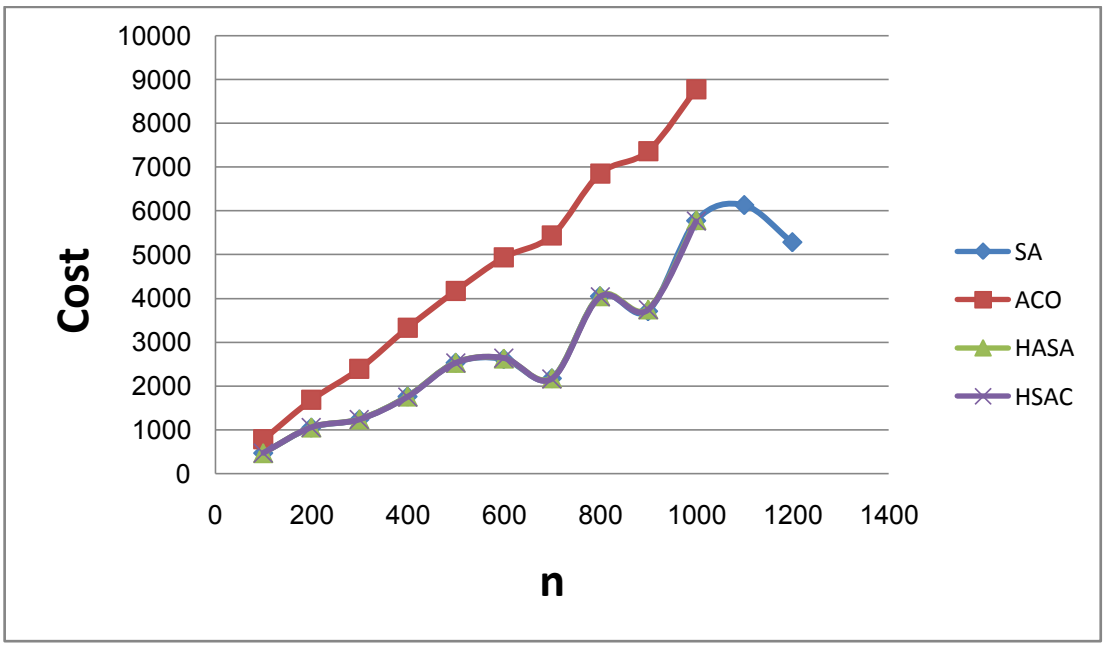

Fig.(3): the results with n large than 50 jobs

Where:

SA: Cost averages of the results of the simulated annealing.

ACO: Cost averages of the results of the ants colony.

HASA: Cost averages of the results of the Hybridization the Ants Colony with Simulated Annealing.

HSAC: Cost averages of the results of the Hybridization the Simulated Annealing with Ants Colony.

REFERENCES

[1] Potts C. N. and Van Wassenhove L. N., "Single Machine To Minimize Total Late Work "Operations Research 40-3(1991)586-595.

[2] Held, M., and Karp ,R.M., "A dynamic programming approach to sequencing problems", Journal of the Society for Industrial and Applied Mathematics 10, 196-210 (1962).

[3] Hariri, A. M. A., Potts, C. N., & Van Wassenhove, L. N. Single machine scheduling to minimize total weighted late work. ORSA Journal on Computing, 7, 232–242. doi:10.1287/ ijoc.7.2.232.

[4] Kethley, R. B., & Alidaee, B. (2002). Single machine scheduling to minimize total weighted late work: A comparison of scheduling rules and search algorithms. Computers & Industrial Engineering, 43, 509–528. doi:10.1016/S0360-8352(02)00123-7.

[5] Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2005a). The two machine flow shop problem with weighted late work criterion and common due date. European Journal of Operational Research, 165,408–415. doi:10.1016/j.ejor.2004.04.011.

[6] Manal G. Ahmed Al-Ayoubi "A Single Machine Scheduling Problem to Minimize the Sum of Total Completion Times and Total Late Works" Al-Mustansiriyah J. Sci. Vol. 23, No 7, 2012.

[7] Asmaa Ali Zeyad " Scheduling Jobs on a Single Machine to Minimize Bi-criteria" M.Sc thesis, Dept. of mathematics, college of Education for Pure Sciences, Thi-Qar University (2016).

[8] Jackson, J.R.,"Scheduling a production line to minimize maximum tardiness", Research Report 43, Management Sciences Research Project, UCLA (1955).

[9] Horn, W.A., "Some simple scheduling algorithms", Naval Research Logistics Quarterly, 21(1):177-185 (1974).

[10] Hariri, A.M.A., and Potts, C.N., "Single machine scheduling with batch set-up times to minimize maximum lateness", Annals of Ops. Res. O, 75-92 (1997).

[11] Christos, K., and George J.K., "Single-machine scheduling problems with past-sequence-dependent delivery times", Int. J. Production Economics 126. 264–266 (2010).

[12] Habibeh, N., and Lai, S.L., "Solving Single Machine Scheduling Problem with Maximum Lateness Using a Genetic Algorithm", Journal of Mathematics Research Vol. 2, No. 3; August (2010).

[13] Radostaw, R., " Minimizing maximum lateness in a single machine scheduling problem with processing time based aging effects", European J. Industrial Engineering Vol. x, No. x,xxxx Accepted 12 September (2011).

[14] Suh-Jenq, Y., Jia-Yuarn, G., Hsin-Tao, L., and Dar-Li, Y.,"Single machine scheduling problem past sequence dependent delivery times and deterioration and learning effects simultaneously", ICIC International c ISSN 1349-4198 (2013).

[15] Morteza, S., and Mehde , S., "Minimizing Maximum Lateness in a Single Machine Scheduling Problem with a Fixed Availability Constraint", Scientific Research Volume: 4, Issue: 1, Pages: 155-165 (January 2015).

[16] Araibi, S.M., "Machine Scheduling Problem to Minimize Two and Three Objectives Function", M.Sc thesis, Dept. of mathematics, college of Education for Pur Sciences,Thi-QarUniversity (2012).

[17] Husein, N.A.," Machine Scheduling Problem to Minimize Multiple Objective Function", M.Sc thesis, Dept. of Mathematics College of Education (Ibn AL-Haitham), Baghdad University (2012).

[18] S. French, Sequencing and scheduling: An introduction to the mathematics of the job-Shop, John Wiley & Sons, New York (1982).

[19] Dorigo, M., Maniezzo V., and Colorni ,A., "A positive feedback as a search strategy", Milan, Italy. tech. Rep. 91-016. (1991).

[20] John E.B., and Patrick R. M., " Ant colony optimization techniques for the vehicle routing problem", Advanced Engineering Informatics 18 41–48, (2004).

[21] Ali Berrichi, Farouk Yalaoui " Efficient bi-objective ant colony approach to minimize total tardiness and system unavailability for a parallel machine scheduling problem " Int J Adv Manuf Technol. Accepted: 15 February 2013

[22] Sami. M Araibi "Ant Colony Method to Minimize Single Machine Scheduling problem "Journal of Thi-Qar Science Vol. 6(3), Dec./2017**.**

[23] Hussam A.A. Mohammed, Fathalh A. Cheachan, Hanan A. Cheachanand Faria A. Cheachan"On Fuzzy Distances and Their Applications on Cost Function $\tilde{L}(\tilde{C}_j, \tilde{D}_j)$" Transactions on Engineering and Sciences**.**Vol.3, Issue 2, February 2015.

[24] Christian Blum., " Ant colony optimization: Introduction and recent trends", Physics of Life Reviews 2353–373.(2005).

[25] Hussein J.M.," Flow shop scheduling problems Using Exact and Local search methods", M.Sc. thesis University of Al-mustansiriyah, College of science, Dep. Of Mathematics (2008).

[26] S.A. Seyed-Alagheband, H. Davoudpour, S.H. Hashemi Doulabi, M. Khatibi"Using a Modified Simulated Annealing Algorithm to Minimize Makespan in a Permutation Flow-shop Scheduling Problem with Job Deterioration"Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II WCECS 2009, October 20-22, 2009, San Francisco, USA.ISBN:978-988-18210-2-7.

[27] Sergio Fichera, Fulvio Cappadonna, Antonio Costa, Alberto Fichera"A Simulated Annealing Algorithm for Single Machine Scheduling Problem with Release Dates, Learning and Deteriorating Effects"Proceedings of the World Congress on Engineering 2013 Vol I, WCE 2013, July 3 - 5, 2013, London, U.K.

[28] Nakandhrakumar R.S and Balachandar M"Implementation of Simulated Annealing Technique for Optimizing Job Shop Scheduling Problem" International Journal of Advanced Mechanical Engineering. ISSN 2250-3234 Volume 4, Number 2 (2014), pp. 169-174

[29] Sergio, F. Fulvio, C. Antonio, C. Alberto, F. " A Simulated Annealing Algorithm for Single Machine Scheduling Problem with Release Dates, Learning and Deteriorating Effects" Proceedings of the World Congress on Engineering Vol I, WCE 2013, July 3 - 5, London, U.K.

[30] Thamilselvan Rakkiannan, Balasubramanie Palanisamy "Hybridization of Genetic Algorithm with Parallel Implementation of Simulated Annealing for Job Shop Scheduling" American Journal of Applied Sciences 9 (10): 1694-1705, 2012.

[31] AL Zuwaini M.K," A comparative Study of Local Search Methods for Some Machine Scheduling Problems with the Usage of Hybridization As A Tool"., Ph. D. thesis, University of Baghdad, College of Education (Ibn Al-Haitham), Dep. of Mathematics (2006).

[32] A. Jamili, M.A. Shafia, R. Tavakkoli-Moghaddam "A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem" journal homepage: www.elsevier.com/locate/eswa (2010).

[33] Kashif Akram , Khurram Kamal and Alam Zeb "Fast simulated annealing hybridized with quenching for solving job shop scheduling problem" Applied Soft Computing 49 (2016) 510–523, journal ho me page: www.elsevier.com/locate /asoc.

[34] Abdul-Razaq, T.S., Potts C.N. and Van Wassenhove, "A survey of algorithms for the single machine total weighted tardiness scheduling problem", Discrete App. Math. 26235-253(1990).