Solving Machine Scheduling Problem Using Particle Swarm Optimization Method

Hanan A. Chachan Mathematical Dept./College of Science/Al-Mustansiriyah University

Abstract

In this paper the problem of scheduling n jobs in a single machine is considered to minimize the total cost of sum weighted completion time, maximum weighted lateness and maximum penalty earliness (i,e to minimize the multiple objective functions ($\sum W_i C_i + L^w_{max} + E^h_{max}$)). The Particle Swarm Optimization (PSO) methods are applied as new local search method on a set of randomly generated problems to solve machine scheduling problem with multiple objective functions. Comparison studies are made between PSO and Genetic Algorithm (GA) to show which one is the better method in applications. In addition, tuning the parameters of every method has been suggested in order to improve the application of every method. A new style of development steps has been proposed to achieve good convergence in application. Since our problem is NP-hard, we propose a new heuristic method like particle swarm optimization to find near optimal solutions specially when the number of jobs exceed the ability of some exact methods like Branch and Bound Methods (BAB) in solving such problems.

Last, the two proposed local search methods results are compared with complete search method in solving problem like machines scheduling problem. Computational experience found that these local search algorithms solve problem to '2000 'jobs with reasonable time.

1. Introduction

The function to be maximized or minimized is called the objective function. A vector, x for the standard maximum problem or y for the standard minimum problem, is said to be feasible if it satisfies the corresponding constraints.

The set of feasible vectors is called the constraint set. A linear programming problem is said to be feasible if the constraint set is not empty; otherwise it is said to be infeasible.

A feasible maximum (resp. minimum) problem is said to be unbounded if the objective function can assume arbitrarily large positive (resp. negative) values at feasible vectors; otherwise, it is said to be bounded.

The value of a bounded feasible maximum (resp, minimum) problem is the maximum (resp. minimum) value of the objective function as the variables range over the constraint set. A feasible vector at which the objective function achieves the value is called optimal [20].

The meaning of the Particle Swarm Optimization (PSO) refers to a relatively new family of algorithms that may be used to find optimal (or near optimal) solutions to numerical and qualitative problems. PSO is an extremely simple algorithm that seems to be effective for optimizing a wide range of Applications [17].

Genetic Algorithms (GA's) are a class of optimization algorithms. GA's attempts to solve problems through modeling a simplified version of genetic process. There are many problems for which a GA approach is useful. It is, however, untraditional if assignment is such a problem [16].

2. Multiple Objective Problems

The Machine Scheduling Problems (MSP) plays a very important role in most manufacturing and production systems as well as in most information processing environment. Scheduling theory has been developed to solve problems occurring in for instance production facilities. The basic scheduling problem can be described as finding for each of the tasks, which are also called jobs, an execution interval on one of the machines that are able to execute it, such that all side-constraints are met; obviously, this should be done in such away that the resulting solution, which is called a schedule, is best possible, that is, it minimizes the given objective function [8].

For many years, scheduling researchers focused on single regular performance measures that are non-decreasing in job completion time.

Typically, each criterion has been studied separately, even though most real life scheduling problems involve multiple criteria [4]. However few studies considered multiple criteria together. Three types of multiple criteria problem can be identified. The first of these types of problems involves identifying all sequence that minimizes the first objective. One of these sequences that minimize a second objective is chosen as the optimal sequence for that problem, this approach is called hierarchical approach [4]. The second of these multiple criteria problems, when the criteria are weighted differently, an objective functions and transform the problem into a single criterion scheduling problem. This approach is called simultaneous optimization along with the third type of multiple criteria problems [4].

The third one of these multiple criteria problems is going to consider both criteria as equally importance. The problem now is to find a sequence that does well on both objectives.

Scheduling problem is specific case of the multiple objective (multi-criteria) scheduling problems can be formulated as follows: minimize or maximize $F(s)=(f_1(s),(f_2(s),\ldots,f_k(s))$ s.t. $s\in S$ where s is a solution, S is the set of feasible solution, k is number of objectives in problem, F(s) is the image of s in the k-objective space and each $f_i(s)$, $i=1,\ldots,k$ represents one (minimization or maximization) objective.

In many problems, the aim is to obtain the optimal arrangement of group of discrete entities in such a way that the additional requirements and constraints (if they exist) are satisfied. If the problem is a multi objective one, various criteria exist to evaluate the equality of solution and there is an objective (Min. or Max.) attached to each of these criteria [4].

The literature on multiple objective problems for single machine problems is summarized by Dileepan and Sen [3], Fry et al [5], Hoogeveen [6], Lee and Vairaktarakis [11] and Nagar et al [12] provide a detailed for MSP's.

In this paper we consider the problem of scheduling n jobs on a single machine to minimize total cost of sum of weighted completion time, maximum weighted lateness and maximum penalty earliness (i.e. to minimize the multiple objective functions ($\sum W_i C_i + L^w_{max} + E^h_{max}$)).

This MSP can be described as follows: a set of n jobs $N=\{1,2,\ldots,n\}$ are available for processing at time zero and each j requires processing during an uninterrupted period of given length p_j and ideally should be completed at its due-date d_j. Our objective is to find a sequence that minimizes the multiple objective functions, the

sum weighted completion time, maximum weighted lateness and maximum penalty earliness. This problem denoted by $1/\ /\sum W_i C_i + L^w_{max} + E^h_{max}$, our main object is to find the optimal and near optimal schedules that minimize the multiple objective functions.

3. Problem Formulation

We consider the problem of scheduling n jobs on a single machine to minimize the sum weighted completion time, maximum weighted lateness and maximum penalty earliness (i.e. to minimize the multiple objective functions (MOF)). Let N={1,2,…,n} be the set of independent jobs (i.e. no precedence constraints are on jobs) which are processed on a single machine. The jobs have to be scheduled without preemption on the machine, where the machine can process one job at a time (i.e. the machine cannot process two jobs at the same time).

Given a schedule (1,2,…,n) then for each job j we calculate the completion time $C_j = \sum_{k=1}^{i} p_k$ s.t. no two jobs overlap in their execution, the earliness and tardiness of job j are defined by:

$E_j = \max_{1\le j\le n}\{d_j - C_j, 0\}$ **and $L_j=C_j-d_j$ respectively corresponding a job is called early if it is completed before its due-date and tardy if it's completed after its due-date. If a job is completed exactly at its due-date, then it is called just-in-time, the objective is to find a:**

$$\text{Min}\quad F(\sigma) = \min_{\sigma\in S}\{\sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)} + L^w_{max} + E^h_{max}\}$$

Subject to

$$\left.\begin{array}{ll} C_{\sigma(j)} \ge p_{\sigma(j)}, & j=1,2,\ldots,n. \\ C_{\sigma(j)} \ge C_{\sigma(j-1)} + p_{\sigma(j)}, & j=2,3,\ldots,n. \\ L^w_{max} = \max\{w_{\sigma(j)} L_{\sigma(j)}\}, & j=1,2,\ldots,n. \\ L_{\sigma(j)} = C_{\sigma(j)} - d_{\sigma(j)}, & j=1,2,\ldots,n. \\ L_{\sigma(j)} \ge 0, & j=1,2,\ldots,n. \\ E^h_{max} = \max\{h_{\sigma(j)} E_{\sigma(j)}\}, & j=1,2,\ldots,n. \\ E_{\sigma(j)} = \max\{ d_{\sigma(j)} - C_{\sigma(j)}, 0\}, & j=1,2,\ldots,n. \\ E_{\sigma(j)} \ge 0, & j=1,2,\ldots,n. \\ w_{\sigma(j)} \ge 0,\ h_{\sigma(j)} \ge 0, & j=1,2,\ldots,n. \end{array}\right\} \quad \ldots(p)$$

processing order of jobs σ, σ=(σ(1),σ(2),...,σ(n)) which minimizes the multiple objective functions (MOF) defined by:

$$\text{Min}\quad F(\sigma) = \min_{\sigma\in S}\{\sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)} + L^w_{max} + E^h_{max}\} \qquad \ldots(1)$$

where $w_{\sigma(j)}$ is positive weighted for completion time of job j, as well as, it is positive number (penalty for tardiness of job j), S is the set of all feasible solutions, σ is a schedule in S.

4. Derivation of Lower and Upper Bound

In this section of the paper we will derive the lower and upper bound of machine scheduling problem using the objective function (19).

4.1 Derivation of Lower Bound

Consider the formulation of the problem (p), the problem can be decomposed into three subproblems with a simple structure. Then the lower bound of the problem (p) is the sum of the minimum value for each subproblem. Consider the three subproblems (p_1),(p_2) and (p_3) as follows:

$$Z_1 = \min_{\sigma \in S}\{\sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)}\} \quad \text{…(2)}$$

Subject to

$$\left.\begin{array}{ll} C_{\sigma(j)} \geq p_{\sigma(j)}, & j=1,2,\ldots,n. \\ C_{\sigma(j)} \geq C_{\sigma(j-1)} + p_{\sigma(j)}, & j=2,3,\ldots,n. \end{array}\right\} \quad \text{…}(p_1)$$

$$Z_2 = \min_{\sigma \in S}\{L^{w}_{max}\} \quad \text{…(3)}$$

Subject to

$$\left.\begin{array}{ll} L^{w}_{max} = \max\{w_{\sigma(j)} L_{\sigma(j)}\}, & j=1,2,\ldots,n. \\ L_{\sigma(j)} = C_{\sigma(j)} - d_{\sigma(j)}, & j=1,2,\ldots,n. \\ L_{\sigma(j)} \geq 0, & j=1,2,\ldots,n. \\ w_{\sigma(j)} \geq 0, & j=1,2,\ldots,n. \end{array}\right\} \quad \text{…}(p_2)$$

$$Z_3 = \min_{\sigma \in S}\{E^{h}_{max}\} \quad \text{…(4)}$$

Subject to

$$\left.\begin{array}{ll} E^{h}_{max} = \max\{h_{\sigma(j)} E_{\sigma(j)}\}, & j=1,2,\ldots,n. \\ E_{\sigma(j)} = \max\{ d_{\sigma(j)} - C_{\sigma(j)}, 0\}, & j=1,2,\ldots,n \\ E_{\sigma(j)} \geq 0, & j=1,2,\ldots,n. \\ h_{\sigma(j)} \geq 0, & j=1,2,\ldots,n. \end{array}\right\} \quad \text{...}(p_3)$$

Its clear that for the decomposition, (p_1),(p_2) and p($_3$) have simpler structure than (p), and thus appear easily first to solve optimality for (p_1) to get z_1 by applying shortest weighted processing time (SWPT) rule. Second, to solve optimality for (p_2) to get z_2 by applying Lawler algorithm [10], Lawler algorithm says that, the 1/ /f_{max} problem is minimized as follows: while there are unassigned jobs, assign the job that has minimum cost when scheduled in that last unassigned position in that position. Third, to get minimum value z_3 for (p_3) by using large maximum slack time (LST) rule, with set $h_{\sigma(j)}$=1 for all j, j=1,2,…,n. the LST rule [7] says that, the 1/ /E_{max} problem is solved by sequencing the jobs according to the SMT rule, that is in order non-increasing d_j-p_j. Last, hence LB=z_1+z_2+z_3 as a lower bound for the problem, since:

$$\min_{\sigma \in S}\{\sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)} + L^{w}_{max} + E^{h}_{max}\} \geq z_1+z_2+z_3 = LB \quad \text{…(5)}$$

4.2 Derivation of Upper Bound

We propose to use a simple heuristic solution which is obtained by ordering the jobs in SWPT rule to provide an initial upper bound (UB) on the MOF. Let the σ, σ=(σ(1),σ(2),...,σ(n)) be such ordered, then:

$$\mathbf{UB=}\sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)} + L_{max}^{w} + E_{max}^{h}$$

5. Particle Swarm Optimization (PSO)

One of the important new learning methods is a Particle Swarm Optimization (PSO), which is simple in concept, has few parameters to adjust and easy to implement. PSO has found applications in a lot of areas. In general, all the application areas that the other evolutionary techniques are good at are good application areas for PSO [17].

PSO was originally developed by a social-psychologist J. Kennedy and an electrical engineer R. Eberhart in 1995 and emerged from earlier experiments with algorithms that modeled the "flocking behavior" seen in many species of birds. Where birds are attracted to a roosting area in simulations they would begin by flying around with no particular destination and in spontaneously formed flocks until one of the birds flew over the roosting area [9]. PSO has been an increasingly hot topic in the area of computational intelligence. It is yet another optimization algorithm that falls under the soft computing umbrella that covers genetic and evolutionary computing algorithms as well [15].

The evolution of several paradigms outlined, and an implementation of one of the paradigms had been discussed. In 1999, Eberhart R.C. and Hu X. [8], arranged a new method for the analysis of human tremor using PSO which is used to evolve an NN that distinguishes between normal subject and those with tremor. In 2004, Shi Y. [17], surveyed the research and development of PSO in five categories: algorithms, topology, parameters, hybrid PSO algorithms, and applications. There are certainly other research works on PSO which are not included due to the space limitation.

PSO is an extremely simple concept, and can be implemented without complex data structure. No complex or costly mathematical functions are used, and it doesn't require a great amount of memory [17]. The facts of PSO has fast convergence, only a small number of control parameters, very simple computations, good performance, and the lack of derivative computations made it an attractive option for solving the problems.

5.1 Fitness Criterion

One of these stopping criteria is the fitness function value. The fitness value is related by the kind of the objective function, the PSO can be applied to minimize or maximize this function, in this paper we focused in minimizing the objective function in order to improve the results.

5.2 PSO Algorithm

The PSO algorithm depends on its implementation in the following two relations:

$$v_{id}^{k} = w^{k-1}v_{id}^{k-1} + c_1 * r_1 * (p^{k-1}{}_{id} - x^{k-1}{}_{id}) + c_2 * r_2 (p^{k-1}{}_{gd} - x^{k-1}{}_{id}) \quad (5a)$$

$$x_{id}^{k} = x_{id}^{k-1} + v_{id}^{k} \quad (5b)$$

where c_1 and c_2 are positive constants, r_1 and r_2 are random functions in the range [0,1], $x_i=(x_{i1},x_{i2},\ldots,x_{id})$ represents the i^{th} particle; $p_i=(p_{i1},p_{i2},\ldots,p_{id})$ represents the best previous position (the position giving the best fitness value) of the i^{th} particle; the symbol g represents the index of the best particle among all the particles in the population, $v=(v_{i1},v_{i2},\ldots,v_{id})$ represents the rate of the position change (velocity) for particle i [2].

The original procedure for implementing PSO is as follows:

1. **Initialize a population of particles with random positions and velocities on d-dimensions in the problem space.**
2. **PSO operation includes:**
   a. **For each particle, evaluate the desired optimization fitness function in d variables.**
   b. **Compare particle's fitness evaluation with its pbest. If current value is better than pbest, then set pbest equal to the current value, and p_i equals to the current location x_i.**
   c. **Identify the particle in the neighborhood with the best success so far, and assign it index to the variable g.**
   d. **Change the velocity and position of the particle according to equation (5a) and (5b).**
3. **Loop to step (2) until a criterion is met.**

Like the other evolutionary algorithms, a PSO algorithm is a population based on search algorithm with random initialization, and there is an interaction among population members. Unlike the other evolutionary algorithms, in PSO, each particle flies through the solution space, and has the ability to remember its previous best position, survives from generation to another. The flow chart of PSO algorithm is shown in figure (1) [20].

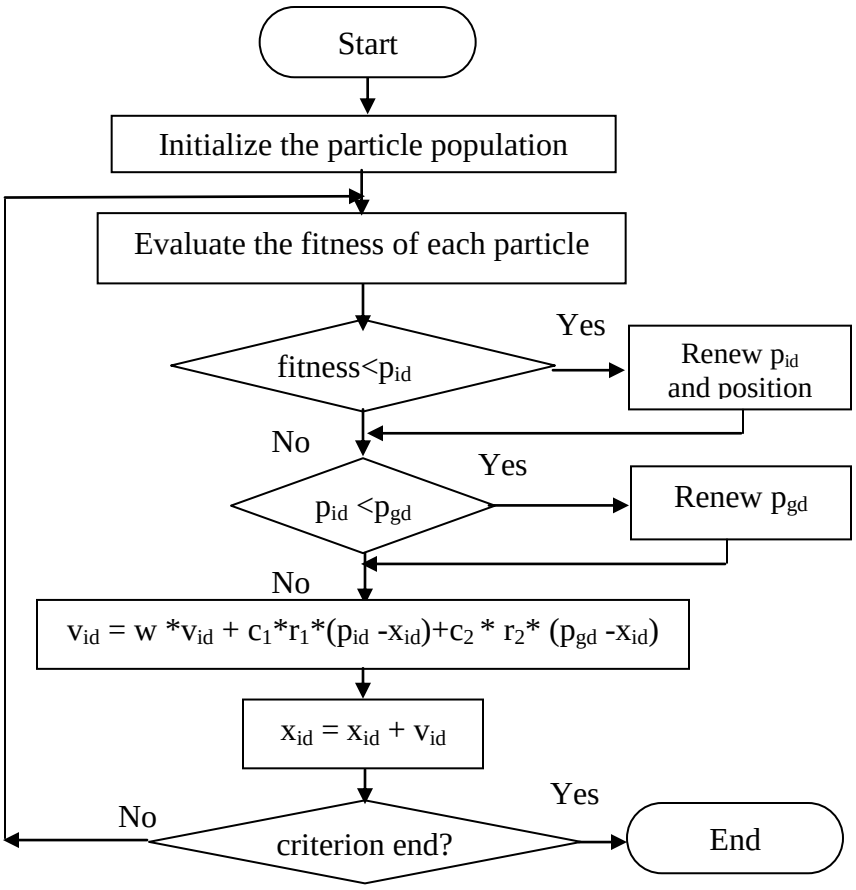

Figure (1) Flowchart of PSO Algorithm [16].

5.3 The Parameters of PSO [18],[13]

A number of factors will affect the performance of the PSO. These factors are called PSO parameters, these parameters are:

1. **Number of particles in the swarm affects the run-time significantly, thus a balance between variety (more particles) and speed (less particles) must be sought.**
2. **Maximum velocity (v_{max}) parameter. This parameter limits the maximum jump that a particle can make in one step.**
3. **The role of the inertia weight w, in equation (5a), is considered critical for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current one.**
4. **The parameters c_1 and c_2, in equation (5a), are not critical for PSO's convergence. However, proper fine-tuning may result in faster convergence and alleviation of local minima, c_1 than a social parameter c_2 but with $c_1 + c_2 = 4$.**
5. **The parameters r_1 and r_2 are used to maintain the diversity of the population, and they are uniformly distributed in the range [0,1].**

6. Genetic Algorithms (GA's)

Genetic Algorithms (GA's) are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a

search algorithm with some of the innovative flair of human search [2]. GA is an iterative procedure, which maintains a constant size population of candidate solution. During each iteration step (Generation) the structures in the current population are evaluated, and, on the basic of those evaluations, a new population of candidate solutions formed. The basic GA cycle shown in figure (2) [1].

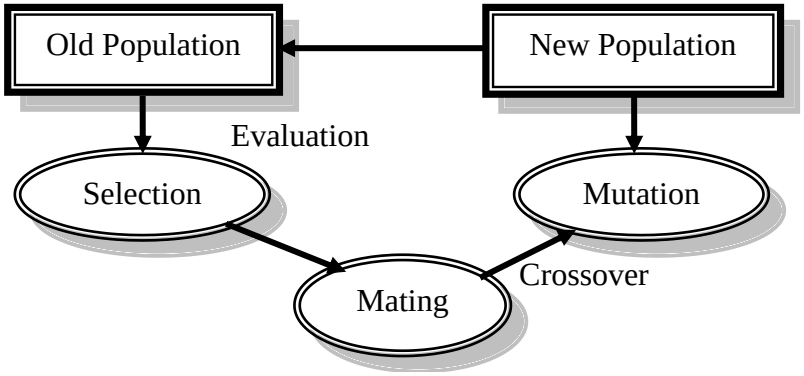

Figure (2) Basic cycle of GA [1].

An abstract view of the GA is:
Generation=0;
Initialize G(P); {G=Generation ; P=Population}
Evaluate G(P);
While (GA has not converged or terminated)
Generation = Generation + 1;
Select G(P) from G(P-1);
Crossover G(P);
Mutate G(P);
Evaluate G(P);
End (While)
Terminate the GA [20].

7. Implementation of Evolving Methods in MSP

Obviously the problems including more than two criteria are more difficult. So there is a need for local search methods to treat a large size instances problem. This is the main aim of the present paper.

Effectively, evolving methods or can be called Local Search methods like PSO and GA have demonstrated their ability to solve multi objective problems to find near optimal solution to the problem (p).

In this section, we are going to describe the two methods of local search In this section we will implement two First, is the PSO as the main new method, and the second, is GA as comparative method to compare the results obtained from the two methods in order to find which is better.

Before we discuss each of method, we have to talk about the common basics between the two methods, these basics are:

1. **Problem Definition**

   The most prominent member of the rich set of combinatorial optimization problems is undoubtedly the Machine Scheduling Problem (MSP). In this problem, n jobs want to be executed in a single machine in some arrangement

which gives minimum objective function (19). Obviously, a single machine scheduling problem sequencing example of NP-complete, the work area to be explored grows exponentially according with number of jobs, and so does. In general, if n jobs were must be arranged in a single machine, then the general complexity is n!.

2. **Problem Representation**

   The chromosome representation should be an integer vector. In this particular approach we accept schedule representation which is described as a list of jobs. For example of (10) jobs numbered from 1 to 10. Table (1) shows the $1 \le p_i \le 10$, $1 \le d_i \le 100$, $1 \le w_i \le 10$ and $1 \le h_i \le 10$ of the MSP all generated randomly.

Table (1) Example of (10) jobs generated randomly for the MSP.

	1	2	3	4	5	6	7	8	9	10
p_i	3	9	8	8	6	1	7	9	5	10
d_i	15	9	14	29	32	9	5	1	5	1
w_i	4	8	5	6	3	3	2	4	1	9
h_i	4	8	5	6	3	3	2	4	1	9

3. **Initial Population**

   For the initialization process we can either use some heuristics starting from different jobs, or we can initialize the population by a random sample of permutation of {1,2,…,n}.

7.1 Use of GA in MSP

Now we will discuss the Use of GA first since it has been used before in MSP for many times.

1. **Genetic Operators**

   - **Selection Operator**

     This method uses the roulette wheel selection method. The string with low fitness has a higher probability of contributing one or more offspring to the next generation.

   - **Crossover Operator**

     The strength of genetic algorithms arises from the structured information exchange of crossover combinations of highly fit individuals. So what we need is a crossover-like operator that would exploit important similarities between chromosomes. For that purpose the crossover used in this algorithm is the Order Crossover (OX) [21], given two parents, builds offspring by choosing a subsequence of a tour from one parent and preserving the relative order of jobs from the other parent.

     For example, if the parents are:

     v_1= (1 2 3 4 5 6 7 8 9 10)

     v_2= (4 5 2 1 10 8 7 6 9 3)

     The resulting offspring is:

     o_1= (1 10 8 4 5 6 7 9 3 2)

     o_2= (4 5 6 1 10 8 7 9 2 3)

   - **Mutation Operator**

     After the new generation has been determined, the chromosomes are subjected to a low rate mutation process. For this example applies two

mutation operators to introduce genetic diversity into the evolving population of permutation. The first operator is a simple two point mutation, which randomly selects two elements in the chromosome and swap them (1 10 8 4 5 6 7 9 3 2) becomes (1 10 3 4 5 6 7 9 8 2). The second operator is a shuffle mutation, which shunts the permutations forward by a random number of places; thus (1 10 3 4 5 6 7 9 8 2) shuffled forward six places becomes (6 7 9 3 2 1 10 8 4 5).

2. **Genetic Parameters**

   For MSP, from our experience, the following parameters are preferred to be used: population size (pop_size=20), probability of crossover (Pc=0.7), probability of mutation Pm =0.1 and some hundreds of number of generations.

7.2 Use of PSO in MSP

For MSP, from our experience, the following parameters are preferred to be used: Number of Particles (N_Par=20,30), Maximum velocity (v_{max}=Number of Jobs (J)), Minimum velocity (v_{min}=1), Inertia Weight ($w \in [0.4,0.9]$), First acceleration parameter ($c_1 \in [0.5,2]$), Second acceleration parameter ($c_2=c_1$), Diversity of the population Maintenance (random $r_1, r_2 \in [0,1]$) and some hundreds of generations.

8. Experimental Results of PSO and GA Implementation in MSP

For this problem, a simulation has been constructed in order to apply the PSO and GA, when using the parameters of PSO and GA mentioned above, the value of MOF, time and number of iterations for best value of MOF results are showed, in table (2) and table (3) which are obtained when applying PSO and GA methods respectively, from number of jobs=3…10, with 1000 generations, for 5 experiments for each number of jobs, using the following abbreviations:

1. **J: Number of Jobs.**
2. **Values of MOF:**
   - **Ex: Experiment number i.**
   - **Max: Maximum value of MOF of experiment i.**
   - **Opt: Optimal value of MOF of experiment i using complete search.**
   - **UB: Upper Bound value of MOF of experiment i using complete search.**
   - **BV: Best Value of MOF of experiment i i.**
   - **ABV: Average of Best Values of MOF for all experiments.**
3. **Values of Time:**
   - **CT: Complete Time of finish experiment i.**
   - **BT: Best Time of best value of MOF of experiment i.**
   - **ABT: Average of Best Time of MOF of all experiments.**
4. **NI: Number of Iteration of best value of MOF of experiment i.**

Note: the shaded cell represents the minimum best value of MOF, the minimum time, the iteration of best value of MOF in all experiments.

Table (2) Applying PSO method on MSP for J=3..10.

J	Ex	Value of MOF					Time			NI
		Max	Opt	UB	BV	ABV	CT	BT	ABT	
3	1	341	323	341	323	252	1	0	0	1
	2	309	241	241	241		0	0		1
	3	355	345	345	345		0	0		1

	4	331	256	264	256		0	0		1
	5	112	94	112	94		0	0		1
4	1	434	431	461	431	431	1	0	0	1
	2	294	217	229	217		0	0		1
	3	723	664	672	644		0	0		12
	4	877	456	639	456		0	0		5
	5	621	407	453	407		1	0		1
5	1	490	282	293	282	524	0	0	0	9
	2	697	647	697	647		1	0		3
	3	286	236	298	236		0	0		21
	4	1382	1045	1162	1045		0	0		10
	5	435	409	578	409		0	0		3
6	1	737	648	674	648	672	1	0	1	51
	2	851	716	840	716		0	0		16
	3	491	402	417	402		1	0		111
	4	842	726	735	726		0	0		4
	5	1300	868	914	868		1	0		38
7	1	1266	1113	1185	1113	897	1	0	1	31
	2	937	468	468	468		1	0		54
	3	1125	724	799	724		0	0		88
	4	1574	1032	1039	1032		1	0		58
	5	1689	1146	1268	1146		1	0		94
8	1	1043	557	664	577	1114	2	0	2	167
	2	1278	914	930	914		2	0		109
	3	2235	1731	1927	1734		2	0		303
	4	1743	1297	1313	1297		2	0		198
	5	1585	1048	1072	1048		2	0		310
9	1	1269	780	780	809	930	2	1	2	640
	2	1552	987	1084	1008		2	1		586
	3	1576	1045	1114	1045		2	1		298
	4	1285	600	643	602		2	1		364
	5	2170	1184	1378	1184		2	0		187
10	1	3671	2260	2380	2343	1436	2	1	2	369
	2	2430	1074	1084	1106		4	2		439
	3	1805	1006	1099	1044		2	0		78
	4	1868	1095	1258	1181		1	0		270
	5	2725	1505	1543	1505		1	0		126

It's important to note that the optimal and maximum value of MOF for each experiment obtained by using complete search method. The complete search, of course, difficult to be applied for jobs more than 10 jobs. For this reason the results of the optimal and maximum value of MOF are not mentioned in the tables included jobs more than 10 jobs.

Table (3) Applying GA method on MSP for J=3..10.

J	Ex	Value of MOF					Time			BNI
		Max	Opt	UB	BV	ABV	CT	BT	ABT	
3	1	341	323	341	323	252	12	0	0	4
	2	309	241	241	261		11	1		25
	3	355	345	345	345		12	0		6
	4	331	256	264	256		12	0		2
	5	112	94	112	94		12	0		2
4	1	434	431	461	431	431	12	1	1	74
	2	294	217	229	217		12	1		11

	3	723	664	672	664		12	0		2
	4	877	456	639	456		11	1		93
	5	621	407	453	407		11	0		18
5	1	490	282	293	282	524	14	6	0	444
	2	697	647	697	647		20	0		23
	3	286	236	298	236		18	2		132
	4	1382	1045	1162	1045		17	17		199
	5	435	409	578	409		14	4		243
6	1	737	648	674	665	673	13	5	9	410
	2	851	716	840	716		17	17		975
	3	491	402	417	402		17	13		762
	4	842	726	735	726		18	5		265
	5	1300	868	914	868		19	3		164
7	1	1266	1113	1185	1129	911	13	8	11	585
	2	937	468	468	501		17	11		567
	3	1125	724	799	726		21	1		73
	4	1574	1032	1039	1049		28	26		959
	5	1689	1146	1268	1152		15	9		592
8	1	1043	557	664	608	1139	12	1	7	101
	2	1278	914	930	928		19	13		693
	3	2235	1731	1927	1754		13	12		986
	4	1743	1297	1313	1313		17	7		362
	5	1585	1048	1072	1093		18	4		286
9	1	1269	780	780	830	1017	13	9	15	772
	2	1552	987	1084	1142		17	14		720
	3	1576	1045	1114	1148		13	11		821
	4	1285	600	643	600		12	11		951
	5	2170	1184	1378	1367		30	28		961
10	1	3671	2260	2380	2350	1535	12	2	4	144
	2	2430	1074	1084	1266		12	3		207
	3	1805	1006	1099	1157		14	2		118
	4	1868	1095	1258	1208		12	9		704
	5	2725	1505	1543	1693		13	4		326

In table (4) a comparison has been made between the results of applying PSO (P) obtained from table (2) and the results of applying GA (G) obtained from table (3) for value of MOF, Time and number of iterations from number of jobs=3…10.

Table (4) Comparison results between GA and PSO methods on MSP for J=3..10.

J	Ex	Value of MOF					Time						NI	
		UB	BV		ABV		CT		BT		ABT			
			G	P	G	P	G	P	G	P	G	P	G	P
3	1	341	323	323	252	252	12	1	0	0	0	0	4	1
	2	241	261	241			11	0	1	0			25	1
	3	345	345	345			12	0	0	0			6	1
	4	264	256	256			12	0	0	0			2	1
	5	112	94	94			12	0	0	0			2	1
4	1	461	431	431	431	431	12	1	1	0	1	0	74	1
	2	229	217	217			12	0	1	0			11	1
	3	672	664	644			12	0	0	0			2	12
	4	639	456	456			11	0	1	0			93	5
	5	453	407	407			11	1	0	0			18	1
5	1	293	282	282	524	524	14	0	6	0	0	0	444	9
	2	697	647	647			20	1	0	0			23	3
	3	298	236	236			18	0	2	0			132	21

	4	1162	1045	1045			17	0	17	0			199	10
	5	578	409	409			14	0	4	0			243	3
6	1	674	655	648	673	672	13	1	5	0	9	1	410	51
	2	840	716	716			17	0	17	0			975	16
	3	417	402	402			17	1	13	0			762	111
	4	735	726	726			18	0	5	0			265	4
	5	914	868	868			19	1	3	0			164	38
7	1	1185	1129	1113	911	897	13	1	8	0	11	1	585	31
	2	468	501	468			17	1	11	0			567	54
	3	799	726	724			21	0	1	0			73	88
	4	1039	1049	1032			28	1	26	0			959	58
	5	1268	1152	1146			15	1	9	0			592	94
8	1	664	608	577	1139	1114	12	2	1	0	7	2	101	167
	2	930	928	914			19	2	13	0			693	109
	3	1927	1754	1734			13	2	12	0			986	303
	4	1313	1313	1297			17	2	7	0			362	198
	5	1072	1093	1048			18	2	4	0			286	310
9	1	780	830	809	1111	930	13	2	9	1	15	2	772	640
	2	1084	1142	1008			17	2	14	1			720	586
	3	1114	1148	1045			13	2	11	1			821	298
	4	643	600	602			12	2	11	1			951	364
	5	1378	1367	1184			30	2	28	0			961	187
10	1	2380	2350	2343	1535	1436	12	2	2	1	4	2	144	369
	2	1084	1266	1106			12	4	3	2			207	439
	3	1099	1157	1044			14	2	2	0			118	78
	4	1258	1208	1181			12	1	9	0			704	270
	5	1543	1693	1505			13	1	4	0			326	126

The average values of MOF, time and number of iterations for best value of MOF results are showed, in table (5) and table (6) which are obtained when applying PSO and GA methods respectively, from chosen number of jobs=20(10)100,100(100)1000 and 2000 with 1000 generations, for 10 experiments for each number of jobs.

Table (5) Applying PSO method on MSP for chosen J=20..2000.

J	Value of MOF			Time			NI	
	AUB	MBV	ABV	MBT	ABT	ACT	MNI	ANI
20	4761	3333	5251	0	1	2	7	374
30	8161	9023	10167	0	1	2	61	335
40	14516	14266	19083	0	1	3	26	339
50	22779	28373	31462	0	1	3	102	362
60	35510	36551	47768	0	1	3	15	336
70	41182	49401	59036	0	1	3	16	300
80	60106	73455	85875	0	1	4	4	294
90	74061	90705	110252	0	0	5	1	38
100	90965	115286	134557	0	2	5	50	316
200	344725	490990	541607	1	3	8	41	333
300	812917	1186399	1278446	0	6	13	26	425
400	1377378	2114556	2245164	0	5	16	5	314
500	2175457	3427849	3557075	0	8	27	1	317
600	3102997	4823224	5099394	0	12	32	2	391
700	4302238	6641942	7066573	0	13	39	20	368
800	5480747	8522990	9087555	4	24	51	134	514
900	6764360	10962441	11409550	2	22	72	40	346
1000	8663102	14354308	14580899	1	28	77	25	366

2000	34438851	56827363	58368948	1	71	142	12	340

Table (6) Applying GA method on MSP for chosen J=20..2000.

J	Value of MOF			Time			NI	
	AUB	MBV	ABV	MBT	ABT	ACT	MNI	ANI
20	4761	3229	5700	0	5	12	31	393
30	8161	9556	11074	3	9	14	232	703
40	14516	15531	21343	0	8	14	129	576
50	22779	29672	35941	0	8	15	48	508
60	35510	43253	53565	0	8	15	28	533
70	41182	57000	66964	1	9	15	24	585
80	60106	83286	94706	0	6	15	53	458
90	74061	101928	121413	0	6	15	104	433
100	90965	128813	149324	0	5	15	4	369
200	344725	523888	582203	1	9	17	35	555
300	812917	1283386	1358119	1	10	24	15	396
400	1377378	2187841	2393128	1	10	28	2	349
500	2175457	3668737	3760479	2	16	30	74	546
600	3102997	5191795	5351359	1	13	38	7	370
700	4302238	7029639	7390968	2	28	60	28	480
800	5480747	9045534	9646061	2	31	65	3	594
900	6764360	11401897	11959531	4	50	79	57	610
1000	8663102	14919013	15202279	10	38	85	119	455
2000	34438851	57831833	59948136	15	101	275	60	384

In table (7) a comparison has been made between the results of applying PSO obtained from table (5) and the results of applying GA obtained from table (6) for value of MOF, Time and number of iterations from chosen number of jobs=20(10)100,100(100)1000 and 2000.

Table (7) Comparison results between GA and PSO methods on MSP for chosen J=20..2000.

J	Value of MOF				Time						NI			
	MBV		ABV		MBT		ABT		ACT		MNI		ANI	
	G	P	G	P	G	P	G	P	G	P	G	P	G	P
20	3229	3333	5700	5251	1	0	5	0	12	2	31	7	393	374
30	9556	9023	11074	10167	3	0	9	0	14	2	232	61	703	335
40	15531	14266	21343	19083	1	0	8	0	14	3	129	26	576	339
50	29672	28373	35941	31462	1	0	8	0	15	3	48	102	508	362
60	43253	36551	53565	47768	1	0	8	0	15	3	28	15	533	336
70	57000	49401	66964	59036	0	0	9	0	15	3	24	16	585	300
80	83286	73455	94706	85875	1	0	6	0	15	4	53	4	458	294
90	101928	90705	121413	110252	1	0	6	0	15	5	104	1	433	38
100	128813	115286	149324	134557	0	0	5	0	15	5	4	50	369	316
200	523888	490990	582203	541607	0	1	9	1	17	8	35	41	555	333
300	1283386	1186399	1358119	1278446	0	0	10	0	24	13	15	26	396	425
400	2187841	2114556	2393128	2245164	1	0	10	0	28	16	2	5	349	314
500	3668737	3427849	3760479	3557075	2	0	16	0	30	27	74	1	546	317
600	5191795	4823224	5351359	5099394	1	0	13	0	38	32	7	2	370	391
700	7029639	6641942	7390968	7066573	1	0	28	0	60	39	28	20	480	368
800	9045534	8522990	9646061	9087555	1	4	31	4	65	51	3	134	594	514
900	11401897	10962441	11959531	11409550	4	2	50	2	79	72	57	40	610	346
1000	14919013	14354308	15202279	14580899	10	1	38	1	85	77	119	25	455	366
2000	57831833	56827363	59948136	58368948	15	3	101	71	275	142	60	12	384	340

Figure (3) describes comparison chart which is shows the relation between value of MOF and number of iterations when applying PSO and GA on MSP consists of 10 jobs.

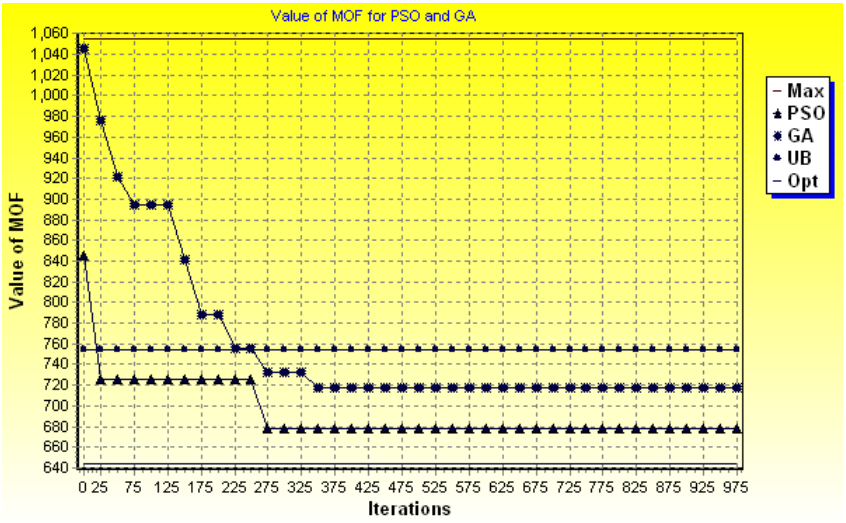

Figure (3) comparison chart of applying PSO and GA on MSP consists of 10 jobs.

Figure (4) describes comparison chart which is shows the relation between value of MOF and number of iterations when applying PSO and GA on MSP consists of 150 jobs.

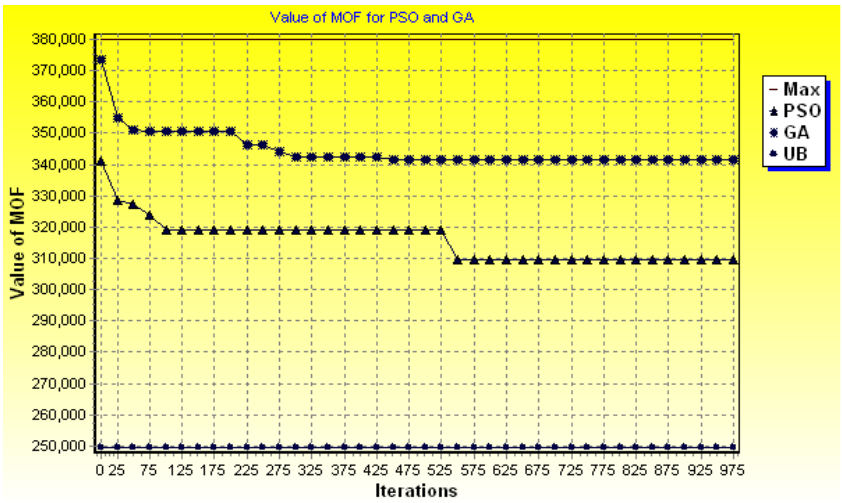

Figure (4) comparison chart of applying PSO and GA on MSP consists of 150 jobs.

The PSO and GA methods were tested by a programming them using version 6 of Delphi Language, and running on Pentium IV at 1.4 GHz, with Ram 128 MB computer.

9. Conclusions

In this paper, near optimal approaches have been developed for one machine scheduling problem to minimize a multiple objective function for the

1/ $/\sum W_i C_i + L_{max}^w + E_{max}^h$ **, this problem is considered to be NP-hard.**

1. **The local search methods that are used to solve all of the large problems in this paper, the results show the robustness and flexibility of local search heuristics.**

2. **The main conclusion to be drawn from our computation results is that the PSO method is more effective than GA method.**
3. **an interesting future research topic would involve the experimentations with the following machine scheduling problems:**
   - $1/\ /\mathrm{Lex}(\sum W_i C_i + L^w_{max} + E^h_{max})$.
   - $1/\ /\sum W_i C_i + L^w_{max} + E^w_{max}$.

References

[1]. Ali F. H., "*Cryptanalysis of the Stream Cipher Systems Using the Genetic Algorithm*", Information Technology & National Security Conference, Al-Riyadh-KSA, 1-4/Dec./2007.

[2]. Abbas S. A. and Ali F. H., "*Cryptanalysis of Polyalphabetic Substitution Cipher Using Genetic Algorithm*", The 1st Conference of Iraqi Association of Information Technology-Iraq, Jen./2009.

[3]. Dileepan P. and Sent T., "*Bicriterion Static Scheduling Research for a Single Machine*", OMEGA; 16(1):53-59, 1988.

[4]. Evans, G. W., "*An Over View of Techniques for solving Multi-Objective Mathematical Programs*", Management Science, Vol. 30, pp.1268-1282, 1984.

[5]. Fry T. D., Armstrong R. D. and Lewis H., "*A Framework for Single Machine Multiple Objective Sequencing Research*", OMEGA; 17 (6): pp.595-607, 1989.

[6]. Hoogeveen H., "*Multicriteia Scheduling*", Department of Computer, Utrecht University, P.O. Box 80089, Utrecht 3508TB, Netherlands, 2005.

[7]. Hoogeveen H., "*Single bi-Criteria Scheduling*", Ph. D. Dissertation, Center for Mathematics and Computer Science, Amsterdam, Netherlands, 2005.

[8]. Hoogeveen H. and Van de Velde S. L., "*Polynomial-Time Algorithms for Single-Machine Multicritera Scheduling*", Centre for Mathematics and Computer Science P.O. Box 4079, 1009 AB, Amsterdam, Netherlands, 1990.

[9]. Kennedy J. and Eberhart R. C. "*Particle Swarm Optimization*", Proceedings of IEEE International Conference on NN, Piscataway, pp. 1942-1948, 1995.

[10]. Lawler E. L., "*Optimal Sequencing of a Single Machine Subject to Precedence Constraints*", Management Science, 19, 544-546, 1973.

[11]. Lee C. V. Vairaktarakis G. L., "*Complexity of Single Machine Hierarchical Scheduling: A Survey*", Report No. 95-10, Department of Industrial and Systems Engineering, University of Florida, Gainesville FL, USA, 1993.

[12]. Nagar A., Haddock J. Heragu S., "*Multiple and bicriteria Scheduling a Literature Survey*", Eur. J. OPI. Res. 1:88-104, 1995.

[13]. Parsopoulos K. E. and Vrahatis M.N., "*Recent Approaches to Global Optimization Problems through Particle Swarm Optimization*", Kluwer Academic Publishers, Netherlands, Natural Computing 1, pp 235–306, 2002.

[14]. Pindeo, M., "*Scheduling: Theory; Algorithms and Systems*", Printice-Hill, Inc., Englewood Diffs, New Jersey 2nd Edition, 2002.

[15]. Ribeiro P. F. and Kyle W. S., "*A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control*", Member, 2003. http://engr.calvin.edu/WEBPAGE/courses/engir302/Reactivepower-PSO-wks.pdf

[16]. Sabah M. Salmo, "*A Comparative Study between Traditional Genetic Algorithms and Breeder Genetic Algorithms*", M. Sc., Thesis, AL-Nahrain University, 2004.

[17]. Settles M. and Rylander B., "*Neural Network Learning using Particle Swarm Optimizers*", Advances in Information Science and Soft Computing, pp. 224-226, 2002.

[18]. Shi Y., "*Particle Swarm Optimization*", Electronic Data Systems, Inc. Kokomo, IN 46902, USA Feature Article, IEEE Neural Networks Society, February 2004.

[19]. Thomas S. Ferguson, "*Linear Programming*", A Concise Introduction, 2008.

[20]. Zhou Y., and et al, "*Particle Swarm Optimization Based Approach for Optical Finite Impulse Response Filter Design*", Optical Society of America, 2003.

[21]. Woodruft D. L., "*Advanced in Computational and Stochastic Optimization Logic*" Programming, and Heuristic search 1998.