Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.5, No.6, 2015 www.iiste.org

Minimizing Multiple Objective Function for Scheduling Machine Problems

Dr. Saad Talib Hasson (Prof.)
Deputy dean, Al-Musaib Engineering College University of Babylon, Iraq.

Shahbaa Mohammed Yousif
Math. Dept, College of Education, University of Al-Mustansiriya, Baghdad, Iraq.

Abstract

The multi-criteria problem of scheduling n jobs on a single machine was considered in this paper. The criteria belong to minimize total completion times, total tardiness and total late work and minimize total completion times, total tardiness and maximum late work by using some exact and local search methods. The proposed methods for solving these minimization problems were seemed to be helpful in finding the set of all efficient solutions. This set of all efficient solutions is not easy to find, therefore, it could be preferable to have an approximation to that set in a reasonable amount of time. Therefore branch and bound (BAB) technique was proposed as an exact approach, while the Genetic algorithm (GA) and Particle Swarm Optimization (PSO) methods were also proposed as local search methods.

Keywords: Multiple objective Scheduling, Branch and Bound, Pareto Optimal Solutions, Genetic Algorithm, Particle Swarm Optimization.

1. Introduction

Scheduling problems are branch of a large field of combinatorial optimization (CO). It was defined as a decision making process that is used on a regular basis in many manufacturing and services industries. Scheduling problems deals with the allocation of resources to tasks over given time periods. Its goal is to minimize one or more objectives (Pinedo 2008). Scheduling theory has been of concern in production problems, manufacturing models, computer science, industrial management, transportation, agriculture, hospitals and many other applications (Agin 1996). Resources and tasks were called **machines** and **jobs** respectively. A machine can perform at most one job at a time. The multi-criteria scheduling problem can be stated as follows: There are n jobs to be processed on a single machine, each job i has processing time p_i and due date d_i at which ideally should be completed. Penalties are incurred whenever a job i is completed earlier or later than its due date d_i. Multi-criteria optimization with conflicting objective functions provides a set of **Pareto optimal solutions**, rather than one optimal solution. This set of solutions includes the solution that no other solution is better than with respect to all objective functions (Abdul_Razaq).

Although the importance of multi-criteria scheduling has been founded for many years (French 1982); (Nelson et al. 1986); (George and Paul 2007); (Mahmood 2014), little attention has been given in the literature to this topic.

The branch and bound (BAB) methods have been first applied to scheduling by Lomnicki 1965 and Ignall and Schrage (1965), these methods accurately solve **machine scheduling problems** (MSP).

Genetic Algorithm (GA) can be considered as a class of optimization algorithms. GA 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 (Sabah 2004).

The Particle Swarm Optimization **(PSO)** considered a new family of algorithms that may be used to find optimal, near optimal or approximated solutions to optimization problems. PSO is an extremely simple algorithm that seems to be effective for optimizing a very wide range of applications (Shi 2004).

The organization of this paper is as follows: in section2 we present multiple objective problems. Section3 and 4 presents the mathematical model and discusses the BAB, PSO and GA methods. Implementation, experimental results, analysis and conclusions are given in the last sections.

1.1 Problem Statement

The problem of this work is to propose different models to find the efficient solution (Pareto optimal solutions) for $1//(\Sigma C_i,\Sigma T_i,\Sigma V_i)$ and $1//(\Sigma C_i,\Sigma T_i,V_{max})$ problems. **Complete Enumeration Method (CEM) and Branch And Bound (BAB)** method can be used. GA and PSO as two local search methods (LSM) may also applied for multi-criteria scheduling problem.

The following notation will be used in this paper:

n : number of jobs.
N : Set of n jobs.
p_j : processing time of job j.
d_j : due date of job j.
C_j : completion time of job j.
T_j : the tardiness of job j.
V_j : the late work of job j.
V_{max} : The maximum late work

The following basic concepts sequence and rules were used in this paper:

Definition (1): Shortest Processing Time (SPT) rule; Jobs are sequencing in non-decreasing order of p_i, this rule was used to minimize ΣC_i for $1/ /\Sigma C_i$ problem (Smith 1956).

Definition (2): Earliest Due Date (EDD) rule: Jobs are sequencing in non-decreasing order of d_i, this rule was used to minimize T_{max} for $1/ /T_{max}$ (Jouni 2000).

Definition (3): The term "**Optimize**" in a multi-criteria decision making problem refers to a solution around which there is no way to improve any objective without worsening the other objective (Jouni 2000).

Definition (4): A feasible schedule σ is a **Pareto optimal** (PO), or non-dominated (efficient) with respect to the performance criteria f and g if there is no feasible schedule π such that both $\mathbf{f(\pi)\leq f(\sigma)}$ **and** $\mathbf{g(\pi)\leq g(\sigma)}$, where at least one of the inequalities is strict (Hoogeveen 2005).

Lemma (1): (**Al-Magraby's lemma**) If $d_j\geq\Sigma p_i$, then there exists an optimal sequence in which job j sequencing last, for the $1/ /\Sigma T_i$ problem (Chen et al. 1997).

Theorem (1): (**Emmon's theorem**)If $\mathbf{p_i\leq p_j}$ **and** $\mathbf{d_i\leq d_j}$ then there exists an optimal sequencing in which job i sequencing before job j, for the $1/ /\Sigma T_i$ problem (Chen et al. 1997).

2. Mathematical Formulation with Analysis

The MCP deals with scheduling the set N={1,2,…,n} of n jobs which are processed on a single machine to minimize the multi-criteria. Each job $i\in N$ has is to be processed on a single machine which can handle only one job at a time. The job i has a processing time p_i and due date d_i, all jobs are available for processing at a time zero.

If a schedule σ = (1,2,…,n) is given, then the earliest completion time $C_i = \sum_{j=1}^{i} p_j$ for each job i can be computed and consequently the tardiness of job i$T_i = \max\{C_i-d_i,0\}$ and

$$V_i = \begin{cases} 0 & \text{if} \quad C_i \le d_i, \qquad i = 1,2,...,n \\ C_i - d_i & \text{if} \quad d_i < C_i < d_i + p_i, \quad i = 1,2,...,n \\ p_i & \text{if} \quad d_i + p_i \le C_i, \qquad i = 2,3,...,n \end{cases}.$$

2.1 Main Concepts

Theorem (2): If the composite objective function F is linear, then there exists an extreme schedule that minimizes F (Hoogeveen 2005).

Definition (5): The function F(f,g) is said to be non-decreasing in both arguments if for any pair of outcome values (x,y) of the functions f and g, we have F(x,y) ≤ F(x+a, y+b), for each pair of non-negative values a and b (Hoogeveen 2005).

Theorem (3): If the composite objective function F(f,g) is non-decreasing in both arguments, then there exists a Pareto optimal schedule that minimizes F (Hoogeveen 2005).

In this work, the following algorithms were required:

Lawler's algorithm (LA) which solves the $1//V_{max}$ problem [1], to find minimum V_{max}.

Lawler's algorithm (LA) (Mahmood 2014) was described by the following steps:

Step (1): Let N={1,2,..,n}, F is the set of all jobs with no successors and $\pi = \phi$.

Step (2): Let j* be a job such that $V_{j^*}(\sum_{i\in N} p_i) = \min_{j\in F}\{V_j(\sum_{i\in N} p_i)\}$.

Step (3): Set N=N-{j*} and sequence job j* in last position of π, i.e. $\pi=(j^*,\pi)$.

Step (4): Modify F with respect to the new set of schedulable jobs.

Step (5): If $\pi = \phi$ stop, otherwise go to step (2).

In this subsection we shall try to find efficient (Pareto optimal) solutions for multi-criteria simultaneous problems. Definition (5) can be extended for three objectives as:

Definition (6): The function F (f_1, f_2, f_3) is said to be non-decreasing in its arguments if for any outcome value (x,y,z) of the functions f_1, f_2 and f_3, we have F(x,y,z) ≤ F(x+a, y+b, z+c) for each of non-negative value a, b and c.

Theorem (4): Consider the composite objective function $F(f_1, f_2,...,f_k)$ where F is non-decreasing in all performance criteria f_i, i=1, 2,…, k, then there is a **Pareto optimal schedule** with respect to $f_1,f_2,\dots,f_k$ that minimizes F (Mahmood 2014).

Let us first consider the formulations of the multi-criteria simultaneous problems say $1//(f_1,f_2,f_3)$. The formulation was as follows:

Min Multi-criteria $\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$

S.t.

Constraints for the problem of object f_1

Constraints for the problem of object f_2

Constraints for the problem of object f_3

In multi-criteria scheduling problems, the optimal solutions were generally called Pareto optimal solutions.

This means that in this type of optimization, we generate all efficient (non-dominated) solutions, then allow the decision maker to make explicit trade-offs among these solutions.

For the problem $1//(f_1, f_2, f_3)$ a schedule (solution) σ_1 dominates another schedule σ_2 if and only if $f_1(\sigma_1) \leq f_1(\sigma_2)$, $f_2(\sigma_1) \leq f_2(\sigma_2)$ and $f_3(\sigma_1) \leq f_3(\sigma_2)$, with at least one strict inequality.

Suppose that $f_1 = \sum C_j$, $f_2 = \sum T_j$ and $f_3 \in \{\sum V_j, V_{max}\}$. Our objective is to find a schedule $\sigma \in S$ (where S is the set of all feasible schedule) that minimizes the multi-criteria problems: $1//(\sum C_j, \sum T_j, \sum V_j)$ and $1//(\sum C_j, \sum T_j, V_{max})$, these problems belong to simultaneous optimization.

Problem (P_1) can be formulated as follows:

$$\text{Min } \{ \Sigma C_i, \Sigma T_i, \Sigma V_i \}$$

Subject to

$$\left.\begin{array}{ll} C_i \geq p_i, & i=1,2,\dots,n. \\ T_i \geq C_i - d_i, & i=1,2,\dots,n. \\ T_i, V_i \geq 0, & i=1,2,\dots,n. \end{array}\right\} \qquad \dots(P_1)$$

The $1//\sum_{j=1}^{n} V_j$ problem is NP-hard since its non-preemptive total late work problem is NP-hard (Jouni 2000).

While problem (P_2) can be formulated as follows:

$$\text{Min } \{ \Sigma C_i, \Sigma T_i, V_{max} \}$$

Subject to

$$\left.\begin{array}{ll} C_i \geq p_i, & i=1,2,\dots,n. \\ T_i \geq C_i - d_i, & i=1,2,\dots,n. \\ T_i, V_i \geq 0, & i=1,2,\dots,n. \end{array}\right\} \qquad \dots(P_2)$$

The only chance to minimize ΣT_i is to use the special cases for the jobs with the same processing times (see section 2.2).

From problems (P_1) and (P_2) we can derive two subproblems, say (SP_1) and (SP_2) respectively.

Problem (SP_1) is $1//\Sigma C_i + \Sigma T_i + \Sigma V_i$ which can be written as:

$$\text{Min } \{ \Sigma C_i + \Sigma T_i + \Sigma V_i \}$$

Subject to

$$\left.\begin{array}{ll} C_i \geq p_i, & i=1,2,\dots,n. \\ T_i \geq C_i - d_i, & i=1,2,\dots,n. \\ T_i, V_i \geq 0, & i=1,2,\dots,n. \end{array}\right\} \qquad \dots(SP_1)$$

While problem (SP_2) is $1//\Sigma C_i + \Sigma T_i + V_{max}$ which can be written as:

$$\text{Min } \{ \Sigma C_i + \Sigma T_i + V_{max} \}$$

Subject to

$$\left.\begin{array}{ll} C_i \geq p_i, & i=1,2,\dots,n. \\ T_i \geq C_i - d_i, & i=1,2,\dots,n. \\ T_i, V_i \geq 0, & i=1,2,\dots,n. \end{array}\right\} \qquad \dots(SP_2)$$

The aim for the (P_1) and (P_2) problems is to find a processing order of the jobs on a single machine to minimize the sum of total completion times and the total tardiness and the total of late work, or maximum late work, which are a single object and can be minimized by BAB method.

2.2 Special Cases for the Problems (P_1) and (P_2)

For the multi-criteria, if the objectives can be optimized individually, then one can deduce that the set of efficient solutions have no more elements only one with extreme values of the individul objective functions. The above fact can be seen in the following special cases:

Case (1): A schedule σ obtained by ordering the jobs in a non-decreasing order of thier processing times (SPT-rule) is an effeceint solution for both problems (P_1) and (P_2) if $d_{\sigma(i)}+p_{\sigma(i)} \le C_{\sigma(i+1)}$ for all i = 1,2,…,n-1.

Case (2): From Emmon's theorem, if the SPT and EDD rules are identical then there exist an effeceint solution for both problems (P_1) and (P_2).

Case (3): If $p_i = p$, $\forall i$, p is positive integer and a schedule σ obtained by ordering the jobs in a non-decreasing order of due dates (EDD-rule) is an effeceint solution for both problems (P_1) and (P_2).

Case (4): If $d_i = d$, $\forall i$, d is positive integer and a schedule σ obtained by ordering the jobs in a non-decreasing order of processing times (SPT-rule) is an effeceint solution for both problems (P_1) and (P_2).

Note that case (3) and case (4) are special case of case (2).

Case (5): From Al-Magrapy lemma, if $d_j \ge \sum_{i=1}^{n} p_i$, and $p_j = \max_{i\in N}\{p_i\}$, and this also satisfies for each job $k \in N-\{j\}$, then there exists an efficient solution for (P_1)and (P_2).

Case (6): If σ satisifes Lawler's algorithm (LA),then there exist an efficient solution for problem (P_2).

Case (7): If for any scheduleσ, $C_{\sigma(j)} \le d_{\sigma(j)}$, $\forall j$, j=1,2,…,n, and σ satisfies SPT-rule, then the schedule σgives $\Sigma V_i=0$ and $V_{max}=0$ then there exists an efficient solution for both (P_1) and (P_2).

3. MSP Exact Solving Methods

3.1 Complete Enumeration Method (Jouni 2000).

Complete enumeration methods generate one by one, all feasible schedules and then pick the best one. For a single machine problem of n jobs there are n! different sequences. Hence for the corresponding m machines problem, there are $(n!)^m$ different sequences. This method may take considerable time as the number $(n!)^m$ is very large even for relatively small values of n and m.

3.2 Branch and Bound Method for (P_1) and (P_2)

This method depends on the techniques of branch and bound (BAB) algorithm with some modifications. The BAB method is characterized by its branching procedure, upper and lower bounding procedures and search strategy.

We present a constructive BAB algorithm to find all or some of the efficient solutions (**Pareto optimal points** (**POP**)) when the criteria ΣC_i, ΣT_i and ΣV_i (V_{max}) are of simultaneous interest in problem P_1 (or P_2). The main idea of this BAB algorithm is depending on properties of BAB algorithm and some modifications such as using the definition of efficient solutions and without reset the upper bound (UB) at the last level of BAB method.

Let $LB1=\max(\Sigma C_i-\Sigma d_i, T_{max}(EDD))$

The main steps of the BAB algorithm are as follows:

Step(1): Find the proposed UB by SPT rule, that is sequencing the job in non-decreasing order of their processing time p_i, i=1,2,...,n, for this order σ calculate $\Sigma C_i(\sigma)$,$\Sigma T_i(\sigma)$ and $\Sigma V_i(\sigma)$ ($V_{max}(\sigma)$) and set

UB=($\Sigma C_i(\sigma)$,$\Sigma T_i(\sigma)$,$\Sigma V_i(\sigma)$) (or UB=($\Sigma C_i(\sigma)$,$\Sigma T_i(\sigma)$,$V_{max}(\sigma)$)) at the parent node of the search tree. UB is efficient by proposition (1) and add this efficient solution to the set of POP. If T_{max}(EDD)=0, then there exists an efficient sequence obtained by proposition (2), and also add this efficient solution to the set of POP.

Step(2):For each partial sequence of jobs σ (i.e., for each node in the search tree), compute the lower bound LB(σ) as follows:

$LB_{P1}(\sigma)$=(SPT(σ),LB1,T_{max}(EDD)) for (P_1) and $LB_{P2}(\sigma)$=(SPT(σ),T_{max}(EDD),Lawler(σ))for (P_2).

Step(3):Branch from each node with LB(σ) ≤ UB.

Step(4):At each node of the last level of the BAB method, if (ΣC_i,ΣT_i,ΣV_i) (or (ΣC_i,ΣT_i,V_{max})) denote the outcome, then add this outcome to the set of POP, unless it is dominated by the previously obtained POP.

Step(5): Stop.

3.3 Branch and Bound Method for (SP_1) and (SP_2)

In this section we will apply the above BAB steps for the (SP_1) and (SP_2) subproblems with different lower bounds. We still use the sequence σ satisfies SPT-rule as an UB for both subproblems. While, the lower bound for the same subproblems is as follows:

$LB_{SP1}(\sigma)$=(SPT(σ)+LB1+T_{max}(EDD)) for (SP_1) and $LB_{SP2}(\sigma)$=(SPT(σ)+LB1+ Lawler(σ)) for (SP_2).

3.4 Experimental Results of Applying CEM and BAB for (P_1) and (P_2)

For the problems (P_1) and (P_2), and for the subproblems (SP_1) and (SP_2), a simulation has been constructed using MATLAB10.0 in order to apply CEM and BAB. The following notations were used:

EV: Efficient Value(s).

NE: Number of Efficient Values.

T/s: Time in seconds.

Table (1) and (2) shows the CPU time results after applying BAB method compared with results obtained from CEM in order to get a set of efficient solutions, on samples of different jobs. The results of CPU time, which generate all solutions for n≤10.

In table (3) we will show the performance of BAB method for the problems (P_1) and (P_2) for n=11,…,25.

In table (4) we will show the results accuracy when applying BAB compared with CEM for the subproblem(SP_1) and (SP_2)for n=3,…,10.

Table (5) describes the results of applying BAB method for the subproblem (SP_1) and (SP_2) for n=11,…,30.

4. MSP Local Search Solving Methods

There are many methods that can be used to solve multi-criteria scheduling problems, which are to find the set of all (some) efficient solutions or at least approximation to it. It is known that the set of all efficient solutions is difficult to find especially for large n (≥25). Therefore, it could be acceptable to find an approximation to the Pareto set in a reasonable time.

In this paper, we will introduce two local search methods to solve multi-criteria scheduling for both problem (P_1) and (P_2) (and subproblem (SP_1) and (SP_2)) to find the set of efficient solutions.

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

1. **Problem Definition and Representation**

   The most important problem of the set of combinatorial optimization problems is undoubtedly the Machine Scheduling Problem (MSP). In order to find the set of POP, we solve the problems (P_1) and (P_2) of minimizing (ΣC_i,ΣT_i,ΣV_i) and (ΣC_i,ΣT_i,V_{max}) respectively. Obviously, this scheduling problem is example of

NP-complete, the work area to be explored grows exponentially according with number of jobs, and so does. The general complexity is n!, such that n jobs were must be arranged in a single machine. The solution representation should be an integer vector. In this particular approach we accept schedule representation which is described as a sequence of jobs.

2. 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 N={1,2,…,n}.

4.1 Genetic Algorithms

Genetic Algorithms (GA's) are search algorithms based on the mechanics of natural selection and natural genetics. 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 (Mitchell 1998).

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**

The selection method is the **roulette wheel**. Copying string according to their fitness value means strings with higher value have higher probability of contributing one or more offspring in the next generation (Mitchell 1998).

- **Crossover Operator**

Order Crossover (OX) (Davis 1985) builds offspring by choosing a subsequence of a tour from one parent and preserving the relative order of cities from the other parent. For example, two parents (with two cut points marked by '|')

Individual1	1	2	3 | 4	5	6	7 | 8	9	and
Individual2	4	5	2 | 1	8	7	6 | 9	3	

Would produce the, offspring in the following way; First, the segments between cut points are copied into offspring:

Offspring1	x	x	x | 4	5	6	7 | x	x	and
Offspring2	x	x	x | 1	8	7	6 | x	x	

This sequence is placed in the first offspring (starting from the second cut point), then the Offspring1 and Offspring2 are:

Offspring1	2	1	8 | 4	5	6	7 | 9	3
Offspring2	3	4	5 | 1	8	7	6 | 9	2

- **Mutation Operator**

After the new generation has been determined, the chromosomes are subjected to a low rate mutation process. For our problems we apply the two point mutation operator to introduce genetic diversity into the evolving population of permutation, which randomly selects two elements in the chromosome and swap them (Abdil_Razaq 2013).

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 number of generations.

4.2 Particle Swarm Optimization

PSO is an extremely very simple concept, which it 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 (Ribeiro 2003).

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

$v_{id} = w * v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id})$ …(a)

$x_{id} = x_{id} + v_{id}$ …(b)

where w is the inertia weight, c_1 and c_2 are positive constants, r_1 and r_2 are random function in the range [0,1], $x_i=(x_{i1},x_{i2},\dots,x_{id})$ represents the i^{th} particle; $p_i=(p_{i1},p_{i2},\dots,p_{id})$ represents the (pbest) 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, $v_i=(v_{i1},v_{i2},\dots,v_{id})$ represents the rate of the position change (velocity) for particle i (Ribeiro 2003).

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 pa_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 (a) and (b).
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 (Abdul_Razaq 2013).

A number of factors will affect the performance of the PSO. These factors are called **PSO parameters**, these parameters are (Kennedy 1995):

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 (3a), 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 (3a), 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].

5. Experimental Results of Applying GA and PSO for (P_1) and (P_2)

For the problems (P_1) and (P_2), and for the subproblems (SP_1) and (SP_2), a simulation has been constructed using Delph10.0 in order to apply GA and PSO for n=3,…,1000 jobs. We use the following notations:

MBV: Mean of Best Value(s).

MEV: Mean of Efficient Value(s).

MAE: Mean Absolute Error between the MBV and MEV.

AAE: Average of Absolute Error comparing MBV with EV obtained from SPT rule.

5.1 Experimental Results for Problems (P_1) and (P_2)

For the problems (P_1) and (P_2), a simulation has been constructed using Delph10.0 in order to apply GA and PSO.

Table (6) and (7) shown the experimental results of applying the local search methods; GA and PSO for problems (P_1) and (P_2) respectively compared with CEM for n=3,…,10.

Table (8) and (9) shown the experimental results of applying the local search methods; GA and PSO for problems (P_1) and (P_2) respectively compared with BAB for n=11,…,25.

Figure (1) and (2) shows the MAE behavior for both PSO and GA applied for (P_1) and (P_2) respectively, with n=11,…,25.

Table (10) and (11) shown the experimental results of applying the local search methods; GA and PSO (compared with other) for problems (P_1) and (P_2) respectively for n=(30,(10),90) and (100,(100),1000).

Figure (3) and (4) show the AAE behavior for both PSO and GA applied for (P_1) and (P_2) respectively, while figure (5) shows the time comparison for the two problems with n=30,…,100,…,1000.

5.2 Experimental Results for Subproblems (SP_1) and (SP_2)

Table (12) and (13) shown the experimental results of applying the local search methods; GA and PSO for subproblems (SP_1) and (SP_2) respectively compared with CEM for n=3,…,10. We use the following notations:

BV: Best Value obtained from GA and PSO.

OV: Optimal Value obtained from CEM and BAB.

AE: Absolute Error between the BV and OV.

AES: Absolute Error between the BV, of GA and PSO, and BV obtained from SPT.

NI: Number of Iterations.

Table (14) and (15) shown the experimental results of applying the local search methods; GA and PSO for subproblems (SP_1) and (SP_2) respectively compared with BAB for n=11,…,30.

Table (16) and (17) shown the experimental results of applying the local search methods; GA and PSO (compared with other) for subproblems (SP_1) and (SP_2) respectively for n=(40,(10),90), (100,(100),1000) and 2000.

Figure (6) and (7) show the AES behavior for both PSO and GA applied for (SP_1) and (SP_2) respectively, while figure (8) shows the time comparison for the two subproblems with n=40,…,100,…,1000,2000.

6. Analysis of the Experimental Results

1. For this paper, a different number of jobs (n) are used for single machine, starting from n=3(1)10, n=30(10)100, n=200(100)1000 and n=2000, with number of iterations which is suitable with n to solve the problems (P_1) and (P_2).
2. The criteria of testing the efficiency of local search method (GA and PSO) are calculated, these criteria represented by, the value of the triple-objective function of exact efficient solutions for the problems (P_1)

and (P_2) of this paper (which are calculated from CEM for n≤10 and from BAB for 11≤n≤25), the approximated local search best efficient solutions (BV) and their average (MBV), the MAE and AAE, while for subproblems (SP_1) and (SP_2) we calculate the BV, AE and AES, and the time which complete the single experiment, lastly, the iteration which found the corresponding BV.

3. For problems (P_1) and (P_2):
   - from tables (6) and (7) for n=3,…,10, GA and PSO gives approximated results from EV and from each other and that is clear from MAE values.
   - from tables (8) and (9) for n=11,…,25, GA and PSO gives approximated results from EV but GA serves better from PSO and that is clear from MAE values (see figures (1) and (2)).
   - from tables (10) and (11) for n=30,…,1000, GA gives better results from PSO and that is clear from AAE values (see figures (3) and (4)).
   - from figure (5), for n=30,…,1000, PSO gives better performance in time from GA.
4. For subproblems (SP_1) and (SP_2):
   - from tables (12) and (13) for n=3,…,10, GA and PSO gives approximated results from OV and from each other and that is clear from AE values.
   - from tables (14) and (15) for n=11,…,30, GA and PSO gives approximated results from OV and from each other and that is clear from AE values.
   - from tables (16) and (17) for n=40,…,2000, PSO gives better results from GA and that is clear from AE values (see figures (5) and (6)).
   - from figure (8), for n=40,…,2000, PSO gives better performance in time from GA.

7. Conclusions and future Works

1. In this paper, we applied exact (CEM and BAB) and local search methods (GA and PSO) to solve the problems (P_1) and (P_2) and the subproblems (SP_1) and (SP_2).
2. It's not easy to find an efficient solution to problem of multi-objective functions, especially for more than double-objective functions.
3. From this paper, we see, in general, that the GA gives accuracy results better than PSO, while PSO gives better performance time than GA.
4. As special case, its not hard to solve hierarchical subproblems derived from (P_1) and (P_2).
5. A hybrid can be done to improve the performance of local search methods, e.g. using simulated annealing or tabu search.

References

Abdul-Razaq T. S. and H. Ali F. H., (2013). Algorithms for Scheduling a Single Machine to Minimize Total Completion Time and Total Tardiness. The 2nd International Conference on Mathematical Sciences-ICMA, 23-24 Oct.

Agin N., (1996). Optimal Seeking with Branch and Bound. Mgmt. Sci. 13, 176-185

Chen T, Qi X. and Tu F., (1997). Single Machine Scheduling to Minimize Weight Earliness Subject to Maximum Tardiness. Computer of Operation Research, 24, 147-152

Davis, L., (1985) “Applying Adaptive Algorithms to Epistatic Domains”, Proceedings of the International Joint Conference on Artificial Intelligence pp. 162-164.

French S., (1982). Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop. John Wiley & Sons, New York.

George S., Paul S., (2007). Pareto optima for total weighted completion time and maximum lateness on a

single machine. Discrete Applied Mathematics 155, 2341 – 2354.
Hoogeveen J. A., (2005). Invited Review Multicriteria Scheduling. European Journal of Operation Research 167, 592-623.
Ignall I. and Schrage L., (1965). Applications of the branch and bound technique to some flow shop scheduling problems. Oper. Res. 13, 400-412.
Jouni L.,(2000). Multi-Objective Nonlinear Pareto-Optimization. Lappeenranta University of Technology,
Kennedy J. and Eberhart R. C.(1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on NN, Piscataway, pp. 1942-1948.
Lomnicki Z., (1965). A branch and bound algorithm for the exact solution of the three machine scheduling problem. Oper. Res. 16, 89-100.
Mahmood A. A., (2014). Multicriteria Scheduling: Mathematical Models, Exact and Approximation Algorithms", Ph.D. Thesis, University of Al-Mustansiriya, College of Science, Dept. of Mathematics.
Mitchell M., (1998). An Introduction of Genetic Algorithms .Abroad Book.
Nelson R.T., Sarin R.K., and Daniels R.L., (1986). Scheduling with multiple performance measures: The one-machine case. Management Science 32, 464-479.
Pinedo M. L., (2008). Scheduling: Theory, Algorithms and Systems. New York: Springer.
Ribeiro P. F. and Kyle W. S., (2003). A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control. Member. http://engr.calvin.edu/…/engir302/Reactivepower-PSO-wks.pdf.
Sabah M. S., (2004). A Comparative Study between Traditional Genetic Algorithms and Breeder Genetic Algorithms. M. Sc. Thesis, AL-Nahrain University.
Shi Y., (2004) .Particle Swarm Optimization", Electronic Data Systems, Inc. Kokomo, IN 46902, USA Feature Article, IEEE Neural Networks Society, February.
Smith W. E., (1956).Various Optimizer for Single-Stage Production. Naval Res. Logist. Quart.3, 59-66.

Table (1) Applying CEM and BAB methods on (P_1) for n=3,..,10.

n	Values of (P_1)					
	CEM			**BAB**		
	NE	**EV**	**T/s**	**NE**	**EV**	**T/s**
3	2	(16,1,1),(17,0,0)	0.1	2	(16, 1,1),(17,0,0)	0.09
4	2	(32,13,9),(33,12,8)	0.1	2	(32,13,9),(33,12,8)	0.1
5	5	(37,9,7),(38,8,7)…	0.19	5	(37,9,7),(38,8,7)…	0.1
6	2	(61,20,15),(64,23,13)	0.15	2	(61,20,15),(64,23,13)	0.09
7	3	(83,35,19),(86,34, 21)..	0.46	2	(83,35,19),(91,43,17)	0.1
8	3	(117,61,26),(120,60,28)..	3.15	2	(117,61,26),(127,71,24)	0.12
9	4	(134,73,30),(135,71,28)..	26.4	4	(134,73,30),(135,71,28)..	0.2
10	4	(170,101,35),(171,99,33)..	281	4	(170,101,35),(171,99,33)..	0.22

Table (2) Applying CEM and BAB methods on (P_2) for n=3,..,10.

n	Values of (P_2)					
	CEM			BAB		
	NE	EV	T/s	NE	EV	T/s
3	2	(16,1,1),(17,0,0)	0.05	2	(16,1,1),(17,0,0)	0.08
4	3	(32,13,5),(33,12,5)..	0.1	2	(32,13,5),(33,12,5)	0.086
5	7	(37,9,4),(38,8,4)…	0.11	6	(37,9,4),(38,8,4)….	0.097
6	4	(61,20,9),(64,23,7) …	0.15	3	(61,20,9),(64,23,7)..	0.095
7	6	(83,35,9),(86,34,9)…	0.46	3	(83,35,9),(91,43,7)…	0.11
8	5	(117,61,9),(120,60,9)…	3.07	2	(117,61,9),(127,71,7)	0.112
9	6	(134,73,9),(135,71,9)…	26.7	5	(134,73,9),(135,71,9)…	0.195
10	6	(170,101,9),(171,99,9)…	246.25	99	(170,101,9),(171,99,9)..	0.236

Table (3) Applying BAB method for (P_1) and (P_2) for n=11,…,25.

n	Values of (P_1)			Values of (P_2)		
	BAB			BAB		
	NE	EV	T/s	NE	EV	T/s
11	12	(159,59,25),(160,56,25),(161,53, 25)…	1	16	(159,59,9),(160,56,9),(161,53, 9)….	1
12	12	(198,85,30),(199,82,30),(200,79,30)…	1	16	(198,85,9),(199,82,9),(200,79 ,9)…	1
13	18	(211,91,31),(211,94,30),(212,89,32)…	4	15	(211,91,9),(212,89,9),(213,86 ,9)…	4
14	18	(266,126,40),(266,129,39), (267,124, 41)…	6	7	(266,126,9),(267,124,9),(268,121 ,9)…	2
15	15	(281,133,41),(281,135,39),(282,132, 43)…	3	6	(281,133, 9),(282,132,9),(283, 130,9)…	1
16	15	(318,154,40),(319,151,43),(319,153, 39)…	4	7	(318,154, 9),(319, 151,9),(320,150,9)…	2
17	9	(421,291,59),(422,289,57), (423,288,56)…	2	5	(421,291, 9),(422,289, 9),(423, 288, 9)…	3
18	12	(439,234,52),(440,231,55),(440,233,51)…	18	18	(439,234,10),(440,231,10),(441,230,10)…	38
19	12	(711,527,93),(712,528,92),(713,529,90)…	16	1	(711,527,10)	6
20	16	(895,664,95),(896,663,97),(898,662,101)…	165	14	(895,664,10),(896,663,10),(898,662,10)…	210
21	14	(992,746,101),(993,745,103),(995,744,107)…	114	10	(992,746,10),(993,745,10),(995,744,10)…	230
22	14	(1106,848,108),(1107,847,110),(1109,846,114)..	511	10	(1106,848,10),(1107,847,10),(1109,846,10)…	907
23	15	(1151,880,112),(1152,883,110),(1159,879,113)..	968	6	(1151,880,10),(1159,879,10),(1160,877,10)..	966
24	6	(840,568,86),(841,567,85),(842,565,83)…	279	9	(840,568,10),(841,567,10),(842,565,10)…	4343
25	6	(934,648,92),(935,647,91),(936,645,89)…	310	9	(934,648,10),(935,647,10),(936,645,10)…	4666

Table (4) Applying CEM and BAB methods for subproblem (SP_1) and (SP_2) for n=3,...,10.

n	SP_1				SP_2			
	CEM	T/s	BAB	T/s	CEM	T/s	BAB	T/s
3	17+0+0=17	0.01	17+0+0=17	0.08	17+0+0=17	0.01	17+0+0=17	0.07
4	33+12+8=53	0.01	33+12+8=53	0.07	33+12+5=50	0.01	33+12+5=50	0.07
5	39+6+5=50	0.02	39+6+5=50	0.08	39+6+4=49	0.02	39+6+4=49	0.07
6	61+20+15=96	0.06	61+20+15=96	0.08	61+20+9=90	0.06	61+20+9=90	0.07
7	83+35+19=137	0.33	83+35+19=137	0.09	83+35+9=127	0.3	83+35+9=127	0.09
8	117+61+26=204	2.6	117+61+26=204	0.1	117+61+9=187	2.7	117+61+9=187	0.08
9	135+71+28=234	23.59	135+71+28=234	0.2	135+71+9=215	22.8	135+71+9=215	0.1
10	171+99+33=303	251.9	171+99+33=303	0.32	171+99+9=279	245.5	171+99+9=279	0.1

Table (5) Applying BAB methods for the subproblem(SP_1) and (SP_2) for n=11,..,30.

n	SP_1		SP_2	
	BAB	T/s	BAB	T/s
11	163+51+24=238	1	163+51+9=223	1
12	202+77+29=308	1	202+77+9=288	1
13	215+84+32=331	1	215+84+9=308	2
14	270+119+41=430	2	270+119+9=398	2
15	281+133+41=455	1	285+128+9=422	1
16	319+153+39=511	3	323+146+9=478	4
17	423+288+56=767	5	422+289+9=720	2
18	440+233+51=724	64	444+226+10=680	35
19	711+527+93=1331	130	711+527+10=1248	4
20	895+664+95=1654	58	895+664+10=1569	10
21	992+746+101=1839	150	992+746+10=1748	10
22	1106+848+108=2062	769	1106+848+10=1964	29
23	1151+880+112=2143	389	1151+880+10=2041	33
24	843+564+82=1489	514	842+565+10=1417	41
25	937+644+88=1669	583	936+645+10=1591	47
26	1312+1002+117=2431	906	1310+1000+10=2320	59
27	1358+1031+118 =2507	984	1353+1034+10=2397	83
28	1489+1148+125=2762	4609	1484+1151+10=2645	299
29	1713+1361+139=3213	6027	1711+1359+10=3080	311
30	1881+1511+148=3540	9220	1879+1509+10=3398	324

Table (6) Applying GA & PSO compared with CEM for(P_1) for n=3,…,10.

n	GA		PSO		CEM	MAE	
	MBV	T/s	MBV	T/s	MEV	PSO	GA
3	(16.5,0.5,0.5)	1,1	(16.5,0.5,0.5)	1,1	(16.5,0.5,0.5)	(0,0,0)	(0,0,0)
4	(32.5,12.5,8.5)	0,2	(32.5,12.5,8,5)	0,2	(32.5,12.5,8.5)	(0,0,0)	(0,0,0)
5	(38.6,8,5.4)	0,5	(38.6,8,5.4)	0,5	(38.6,8,5.4)	(0,0,0)	(0,0,0)
6	(62.5,21.5,14)	0,2	(62.5,21.5,14)	1,2	(62.5,21.5,14)	(0,0,0)	(0,0,0)
7	(87,35,20.5)	1,2	(87,37.3,19)	1,3	(86.7,37.3,19)	(0.004,0,0)	(0.004,0.06,0.08)
8	(96.4,45.4,19.6)	0,5	(94.7,44.3,19.67)	0,3	(96.4,45.4,19.6)	(0.02,0.02,0.004)	(0,0,0)
9	(137.5,70.5,29)	0,2	(137.4,7,29.6)	1,5	(139.8,75,28.3)	(0.02,0.03,0.05)	(0.02,0.06,0.03)
10	(176,101,34.5)	1,2	(175.3,103,34.75)	0,4	(176.8,104,33.3)	(0.008,0.01,0.05)	(0.004,0.03,0.04)

Table (7) Applying GA & PSO compared with CEM for(P_2) for n=3,…,10.

n	GA		PSO		CEM	MAE	
	MBV	T/s	MBV	T/s	MEV	PSO	GA
3	(16.5,0.5,0.5)	ε	(16.5,0.5,0.5)	ε	(16.5,0.5,0.5)	(0,0,0)	(0,0,0)
4	(32.5,12.5,9.5)	ε	(32.5,12.5,9.5)	ε	(32.5,12.5,8.5)	(0,0,0.1)	(0,0,0.1)
5	(38.6,8,9.5)	ε	(38.6,8,9.5)	1	(38.6,8,5.4)	(0,0,0.8)	(0,0,0.8)
6	(62.5,21.5,9.5)	1	(62.5,21.5,9.5)	ε	(62.5,21.5,14)	(0,0,0.3)	(0,0,0.3)
7	(86.67,37.33,9.5)	1	(87.67,37.33,9.5)	ε	(86.67,37.33,19)	(0.1,0,0.5)	(0,0,0.5)
8	(96.4,45.4,9.5)	1	(96.4,45.4,19.6)	1	(96.4,45.4,19.6)	(0,0,0)	(0,0,0.5)
9	(139.75,75,9.5)	1	(139.71,75.29,9.5)	1	(139.75,75,28.25)	(0.0003,0.004,0.7)	(0,0,0.7)
10	(176.83,104.67,9.5)	1	(174,101.75,9.5)	1	(176.75,104,33.25)	(0.02,0.02,0.7)	(0.0005,0.006,0.7)

Where $0<\varepsilon<1$.

Table (8) Applying GA & PSO compared with BAB for(P_1) for n=11,…,25.

N	GA		PSO		BAB	MAE	
	MBV	T/s	MBV	T/s	MEV	PSO	GA
11	(163.1,55.1,23.9)	1	(171.7,59.7,21.7)	1	(166,56,22)	(0.034,0.065,0.015)	(0.017,0.015,0.085)
12	(202.6,83.6,28.4)	1	(214,95.7,26.7)	1	(207,84,27)	(0.034,0.861,0.012)	(0.021,0.005,0.052)
13	(214.8,87.2,30.8)	1	(238.8,112.8,26.5)	1	(221.6,92.9,28.8)	(0.078,0.213,0.079)	(0.031,0.061,0.07)
14	(275,133.8,38.3)	1	(293.7,146.3,37)	1	(276.6,127.9,37.8)	(0.062,0.144,0.021)	(0.006,0.046,0.012)
15	(285.5,131,42.3)	1	(316.5,166.3,38)	1	(291,134.2,40.7)	(0.088,0.239,0.066)	(0.019,0.024,0.039)
16	(322,149,44.5)	1	(382.2,219.6,39.4)	2	(325.2,152.2,42.5)	(0.175,0.443,0.072)	(0.01,0.021,0.048)
17	(425.3,292.7,57)	1	(444,311.5,57.5)	1	(424.8,290.2,55.7)	(0.045,0.073,0.033)	(0.001,0.008,0.024)
18	(444.3,234.5,55)	2	(490,278.3,54.7)	1	(447.6,232.1,54.8)	(0.095,0.199,0.001)	(0.007,0.01,0.005)
19	(727.6,543.8,91.8)	2	(811.3,628.2,89.7)	2	(734,552.3,86.6)	(0.105,0.137,0.036)	(0.009,0.016,0.06)
20	(916.8,682.6,99.6)	1	(942.5,706.5,97.5)	2	(908.4,664.2,95.8)	(0.038,0.064,0.018)	(0.009,0.028,0.04)
21	(1008.5,756.8,105)	2	(1105.3,851.5,102.3)	2	(998.8,740.3,102.5)	(0.107,0.15,0.002)	(0.01,0.022,0.024)
22	(1116.8,854.4,112.2)	2	(1223.8,952.5,110.5)	2	(1112.8,842.3,109.5)	(0.1,0.131,0.009)	(0.004,0.014,0.025)
23	(1153.8,879.8,115)	2	(1235,959.5,113)	2	(1163.5,880.6,111.5)	(0.061,0.09,0.013)	(0.008,0.001,0.031)
24	(842,569.3,86.3)	2	(1004.8,724.6,86)	2	(848.2,570.8,83.5)	(0.185,0.269,0.03)	(0.007,0.003,0.033)
25	(941,651.7,92.3)	2	(1077.7,800.7,90)	2	(942.7,651.3,89.5)	(0.142,0.229,0.006)	(0.002,0.0005,0.032)

Table (9) Applying GA & PSO compared with BAB for(P_2) for n=11,…,25.

n	GA		PSO		BAB	MAE	
	MBV	T/s	MBV	T/s	MEV	PSO	GA
11	(166,58,9.5)	2	(173.7,61,9.5)	1	(177.3,67.4,7.4)	(0.021,0.095,0.277)	(0.063,0.139,0.277)
12	(203.8,86.6,9.5)	1	(218,98.6,9.5)	2	(219.3,96.4,7.4)	(0.006,0.023,0.277)	(0.071,0.101,0.277)
13	(219.3,90.7,9.5)	2	(231.8,108.5,9.5)	2	(177.3,67.4,7.4)	(0.307,0.61,0.277)	(0.237,0.346,0.277)
14	(271.1,130.1,9.5)	1	(292.5,143.5,9.5)	1	(271.3,120.1,9)	(0.078,0.194,0.056)	(0.0006,0.083,0.056)
15	(283,132,9.5)	1	(318,173.8,9.5)	1	(285,129.3,9)	(0.116,0.343,0.056)	(0.007,0.021,0.056)
16	(324,149.17,9.5)	1	(378.6,211.2,9.5)	2	(322.3,148.3,9)	(0.175,0.424,0.056)	(0.005,0.006,0.056)
17	(426.8,295.4,9.5)	2	(458.5,325.5,9.5)	2	(423.6,288.2,9)	(0.082,0.129,0.056)	(0.008,0.025,0.056)
18	(445,233,9.5)	1	(493.3,289,9.5)	1	(462.7,243.5,9.5)	(0.066,0.187,0)	(0.038,0.043,0)
19	(742,560.1,9.5)	1	(796.2,609.8,9.5)	1	(711,527,10)	(0.12,0.157,0.05)	(0.044,0.063,0.05)
20	(911.2,659.8,9.5)	2	(953,722.7,9.5)	2	(902,657.9,10)	(0.057,0.098,0.05)	(0.01,0.003,0.05)
21	(1020.8,762,9.5)	1	(1075.8,817,9.5)	2	(999,739.9,10)	(0.077, 0.104,0.05)	(0.022,0.03,0.05)
22	(1117.8,855.4,9.5)	1	(1216.3,940.3,9.5)	2	(1113,841.9,10)	(0.092,0.117,0.05)	(0.004,0.016,0.05)
23	(1153,883.7,9.5)	2	(1257.3,978.3,9.5)	2	(1159.3,876,10)	(0.085,0.117,0.05)	(0.005,0.009,0.05)
24	(842.5,568,9.5)	2	(999.8,729,9.5)	2	(865.6,588.8,9.6)	(0.155,0.238,0.006)	(0.027,0.035,0.006)
25	(935.7,648.3,9.5)	2	(1028,749.5,9.5)	2	(961.3,670.6,9.6)	(0.069,0.118,0.006)	(0.027,0.033,0.006)

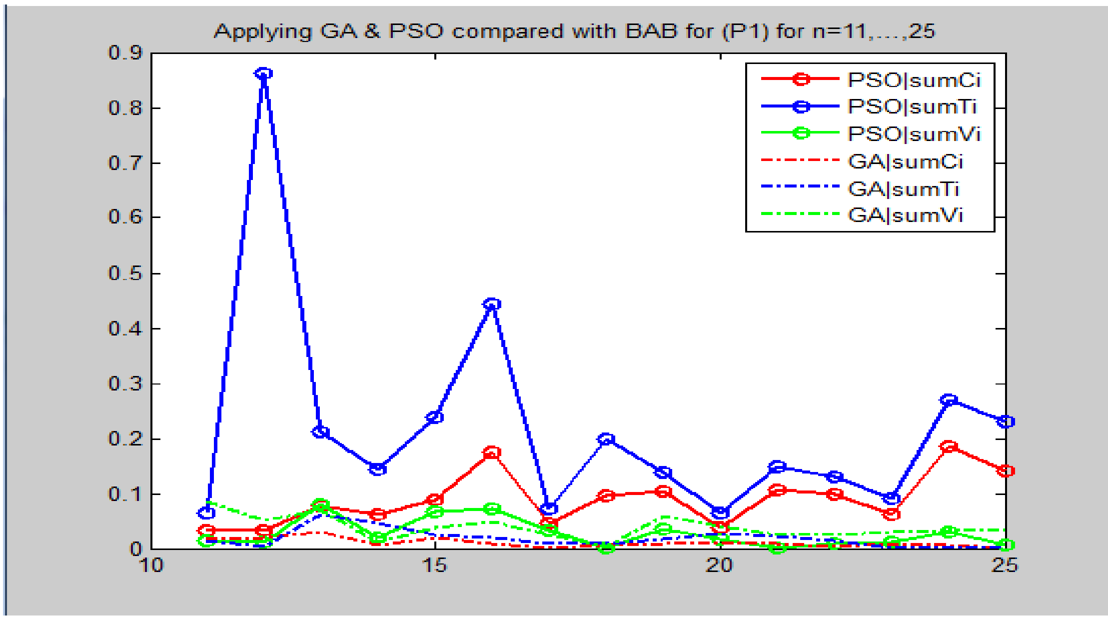

Figure (1): MAE behavior for PSO and GA applied for (P_1) with n=11,…,25.

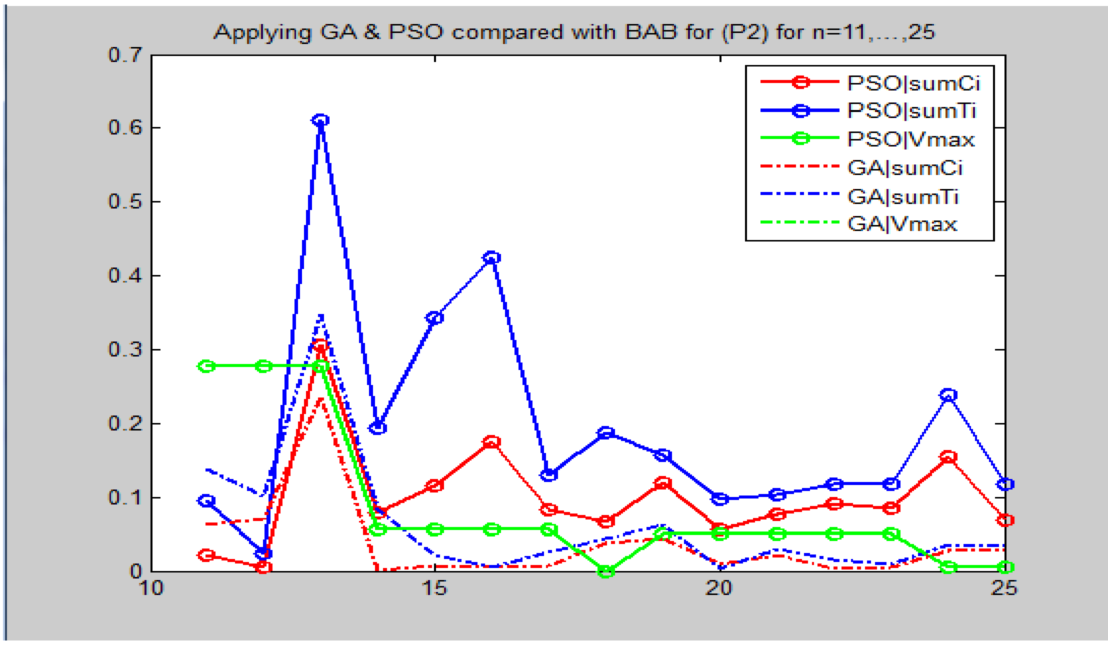

Figure (2): MAE behavior for PSO and GA applied for (P_2) with n=11,…,25.

Table (10) Applying GA & PSO for(P_1) for n=(30,(10),90) and (100,(100),1000).

n	GA			PSO		
	MBV	MAE	T/s	MBV	AAE	T/s
30	(1457.3,1130.5,126.5)	(0.0058,0.0094,0.0417)	2	(1658.5,1334.3,125.5)	(0.1446,0.1914,0.0492)	2
40	(3545.5,3114,208.5)	(0.0016,0.0013,0.0024)	2	(3540,3110,209)	(0.0000,0.0000,0.0000)	2
50	(4287.6,3746.2,239.3)	(0.0086,0.0100,0.0191)	2	(5124,4578.3,235)	(0.2054,0.2344,0.0369)	3
60	(6907.8,6242.5,305.8)	(0.0048,0.0052,0.0105)	3	(7974.5,7297.5,304.8)	(0.1599,0.1751,0.0138)	4
70	(9391.6,8628.8,365.4)	(0.0134,0.0149,0.0124)	4	(10983.3,10222,363)	(0.1852,0.2023,0.0189)	4
80	(10978.8,10031.7,387.8)	(0.0309,0.0349,0.0206)	4	(12704.7,11746.3,389.7)	(0.1929,0.2118,0.0160)	4
90	(17210.8,16158.6,519.4)	(0.0061,0.0064,0.0088)	4	(19846,18785.33,519.67)	(0.1602,0.1700,0.0083)	6
100	(18681.8,17557.3,521.3)	(0.0160,0.0172,0.0126)	4	(22697,21580.4,515.6)	(0.2344,0.2503,0.0235)	5
200	(74651.7,72358.2,1059.8)	(0.0066,0.0069,0.0067)	11	(90736,88447,1059.7)	(0.2235,0.2307,0.0069)	9
300	(182738.5,179175.8,1673.2)	(0.0102,0.0105,0.0041)	17	(223546.8,219991.8,1670)	(0.2358,0.2407,0.0060)	14
400	(314330.3,309387.8,2199.3)	(0.0160,0.0163,0.0031)	27	(394853.8,389919.3,2191.3)	0.2763,0.2808,0.0067)	17
500	(477428.6,471285.8,2743.4)	(0.0043,0.0043,0.0024)	40	(605573,599435.3,2739.5)	(0.2738,0.2774,0.0038)	21
600	(733344.8,726320.8,3264.4)	(0.0590,0.0596,0.0038)	55	(808538,801508.5,3266.5)	(0.1676,0.1693,0.0032)	25
700	(972606.7,964164.4,3830.6)	(0.0422,0.0426,0.0022)	88	(1108341.5,1099892.5,3829.5)	(0.1877,0.1894,0.0025)	36
800	(1276770.8,1266766.2,4505.1)	(0.0093,0.0094,0.0013)	111	(1578415,1568419.3,4496.7)	(0.2478,0.2498,0.0032)	42
900	(1592743.5,1581596.3,4934.3)	(0.0164,0.0165,0.0014)	121	(2032643.8,2021508.6,4926.6)	(0.2971,0.2992,0.0029)	49
1000	(1980003.8,1968100.2,5489)	(0.0252,0.0254,0.0015)	164	(2288728.5,2276824,5488)	(0.1850,0.1862,0.0016)	52

Table (11) Applying GA & PSO for(P_2) for n=(30,(10),90) and (100,(100),1000).

n	GA			PSO		
	MBV	MAE	T/s	MBV	AAE	T/s
30	(1452.4,1124.2,9.5)	(0.0023,0.0038,0.0610)	2	(1613.3,1287.3,9.5)	(0.1134,0.1494,0.0063)	2
40	(3540,3110,9.5)	(0.0000,0.0000,0.0442)	2	(3540,3110,9.5)	(0.0000,0.0000,0.0152)	2
50	(4348,3806.8,9.5)	(0.0228,0.0264,0.0721)	2	(4716,4176.5,9.5)	(0.1094,0.1260,0.0816)	3
60	(6908.6,6240,9.5)	(0.0049,0.0048,0.0691)	4	(8050.4,7380,9.5)	(0.1710,0.1884,0.0302)	4
70	(9337,8579,9.5)	(0.0076,0.0091,0.0231)	3	(11219.8,10461.6,9.5)	(0.2107,0.2305,0.0656)	4
80	(10728,977,9.5)	(0.0073,0.0085,0.0984)	5	(12096.5,11149,9.5)	(0.1358,0.1502,0.0759)	5
90	(17129.8,16075,9.5)	(0.0014,0.0012,0.0695)	4	(20805.5,19738.7,9.5)	(0.2163,0.2294,0.0765)	6
100	(18685.1,17561.6,9.5)	(0.0162,0.0175,0.0693)	5	(22091.3,20972.7,9.5)	(0.2015,0.2151,0.0553)	5
200	(74554.3,72264.3,9.5)	(0.0053,0.0056,0.0712)	11	(93489,91201.7,9.5)	(0.2606,0.2691,0.0813)	5
300	(181574.8,178007.8,9.5)	(0.0038,0.0039,0.0587)	17	(218534.3,214969.7,9.5)	(0.2081,0.2123,0.0355)	16
400	(310531.6,305588.6,9.5)	(0.0037,0.0038,0.0003)	26	(382810,377869.7,9.5)	(0.2374,0.2412,0.0472)	20
500	(481734.7,475591.9,9.5)	(0.0133,0.0135,0.0894)	36	(606223.3,600089.5,9.5)	(0.2752,0.2788,0.0779)	26
600	(710843.8,703815.9,9.5)	(0.0265,0.0268,0.0315)	50	(860058.3,853033,9.5)	(0.2420,0.2445,0.0545)	32
700	(974262.3,965820.2,9.5)	(0.0440,0.0444,0.0824)	78	(1167998.7,1159566.7,9.5)	(0.2516,0.2539,0.0493)	35
800	(1389754.7,1379764.2,9.5)	(0.0987,0.0995,0.0153)	92	(1577220,1567215.3,9.5)	(0.2469,0.2488,0.0884)	37
900	(1620158.3,1609012.3,9.5)	(0.0339,0.0341,0.0050)	116	(1952230.3,1941094,9.5)	(0.2458,0.2476,0.0204)	43
1000	(1932368.5,1920464.3,9.5)	(0.0005,0.0005,0.0887)	146	(2518362.8,2506468.6,9.5)	(0.3039,0.3058,0.0651)	50

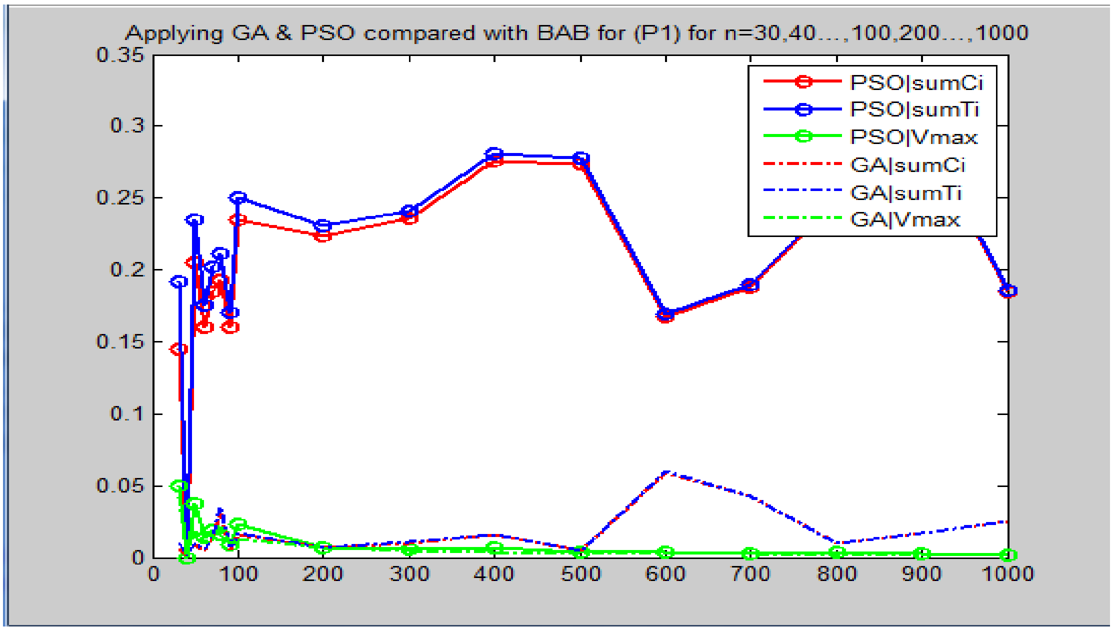

Figure (3): AAE behavior for PSO & GA applied for (P_1) with n=30,…,100,…,1000.

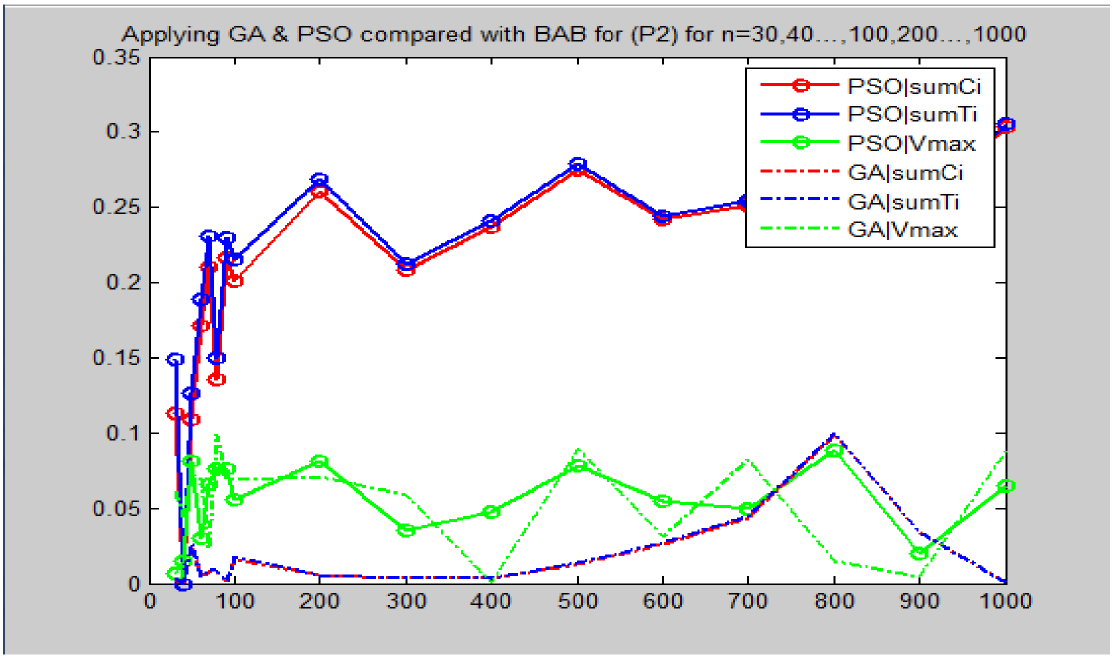

Figure (4): AAE behavior for PSO & GA applied for (P_2) with n=30,…,100,…,1000.

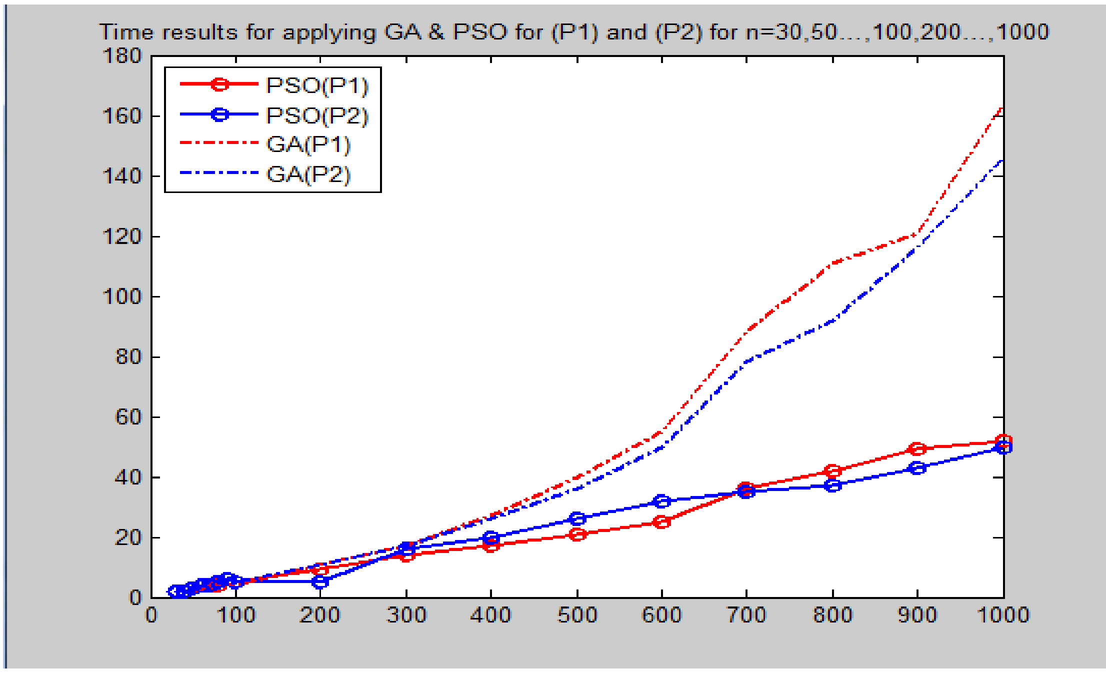

Figure (5): Time comparison for GA & PSO for (P_1) and (P_2) with n=30,…,100,…,1000

Table (12) Applying GA & PSO compared with CEM for(SP_1) for n=3,…,10.

n	GA			PSO			CEM	AE	
	BV	NI	T/s	BV	NI	T/s	OV	GA	PSO
3	17+0+0=17	2	1	17+0+0 =17	1	ε	17+0+0=17	0	0
4	33+12+8=53	12	1	33+12+8 =53	2	ε	33+12+8=53	0	0
5	39+6+5=50	27	1	39+6+5 =50	23	ε	39+6+5=50	0	0
6	61+20+15=96	23	1	61+20+15 =96	57	ε	61+20+15=96	0	0
7	83+35+19=137	36	1	83+35+19 =137	40	1	83+35+19=137	0	0
8	117+61+26=204	40	1	117+61+26 =204	378	1	117+61+26=204	0	0
9	135+71+28=234	57	1	136+71+28 =235	373	1	135+71+28=234	0	0.004
10	171+99+33=303	89	1	173+100+33 =306	79	ε	171+99+33=303	0	0.01

Table (13) Applying GA & PSO compared with CEM for(SP_2) for n=3,…,10.

n	GA			PSO			CEM	AE	
	BV	NI	T/s.	BV	NI	T/s.	OV	GA	PSO
3	17+0+0=17	2	ε	17+0+0=17	1	ε	17+0+0=17	0	0
4	32+13+5=50	5	1	32+13+5=50	1	ε	33+12+5=50	0	0
5	39+6+4=49	88	1	39+6+4=49	7	ε	39+6+4=49	0	0
6	61+20+9=90	116	2	61+20+9=90	104	ε	61+20+9=90	0	0
7	83+35+ 9=127	379	1	83+35+9=127	5	1	83+35+9=127	0	0
8	119+62+9=190	300	1	117+61+9=187	256	ε	117+61+9=187	0.02	0
9	142+74+9=225	29	1	136+71+9=216	205	ε	135+71+9=215	0.05	0.005
10	186+116+9=311	226	1	171+102+9=282	472	ε	171+99+9=279	0.1	0.01

Table (14) Applying GA & PSO compared with BAB for(SP_1) for n=11,…,30.

n	GA			PSO			BAB	AE	
	BV	NI	T/s	BV	NI	T/s	OV	GA	PSO
11	163+51+24=238	101	2	173+51+22=246	796	1	163+51+24=238	0	0.03
12	202+77+29=308	128	2	205+88+33=326	465	1	202+77+29=308	0	0.06
13	215+84+32=331	133	2	219+94+37=350	225	1	215+84+32=331	0	0.06
14	268+121+41=430	206	2	274+134+38=446	580	1	270+119+41=430	0	0.04
15	281+135+39=455	134	2	298+150+40=488	280	1	281+133+41=455	0	0.07
16	319+153+39=511	205	2	341+174+43=558	115	1	319+153+39=511	0	0.09
17	423+288+56=767	181	2	461+320+59=840	107	1	423+288+56=767	0	0.1
18	440+233+51=724	231	3	484+276+62=822	525	1	440+233+51=724	0	0.1
19	715+524+92=1331	365	2	777+580+93=1450	352	0	711+527+93=1331	0	0.09
20	905+652+93=1650	345	3	957+693+105=1755	446	0	895+664+95=1654	0.002	0.06
21	1002+734+99=1835	654	3	1041+771+108=1920	1072	1	992+746+101=1839	0.002	0.04
22	1106+848+108=2062	372	3	1153+878+122=2153	1957	1	1106+848+108=2062	0	0.04
23	1155+881+108=2142	448	3	1219+924+117=2260	1577	2	1151+880+112=2143	0.0005	0.05
24	843+564+82=1489	626	3	951+680+89=1720	407	2	843+564+82=1489	0	0.2
25	937+644+88=1669	943	3	1067+794+100=1961	1188	2	937+644+88=1669	0	0.2

26	1313+998+119=2430	556	3	1403+1109+128=2640	1961	2	1312+1002+117=2431	0.0004	0.09
27	1357+1030+121=2508	740	4	1488+1135+124=2747	1195	2	1358+1031+118=2507	0.0004	0.1
28	1489+1148+125=2762	1141	4	1678+1347+136=3161	1647	2	1489+1148+125=2762	0	0.1
29	1711+1359+143=3213	466	4	1835+1482+155=3472	1687	2	1713+1361+139=3213	0	0.8
30	1884+1506+149=3539	1088	4	2075+1666+158=3899	1629	3	1881+1511+148=3540	0.0003	0.9

Table (15) Applying GA & PSO compared with BAB for(SP_2) for n=11,…,30.

n	GA			PSO			BAB	AE	
	BV	NI	T/s	BV	NI	T/s	OV	GA	PSO
11	168+58+9=235	603	2	165+52+9=226	498	1	163+51+9=223	0.05	0.01
12	207+91+9=307	419	2	208+76+9=293	974	1	202+77+9=288	0.07	0.02
13	224+96+9=329	576	2	221+95+9=325	357	1	215+84+9=308	0.07	0.06
14	281+135+9=425	422	2	285+131+9=425	688	1	270+119+9=398	0.07	0.07
15	294+149+9=452	40	2	301+148+9=458	807	1	285+128+9=422	0.07	0.09
16	349+198+9=556	628	2	340+171+9=520	460	1	323+146+9=478	0.2	0.09
17	463+333+9=805	458	2	464+332+9=805	399	0	422+289+9=720	0.12	0.12
18	508+283+10=801	148	1	489+274+10=773	502	1	444+226+10=680	0.2	0.14
19	769+576+10=1355	309	3	791+607+10=1408	580	1	711+527+10=1248	0.09	0.13
20	977+742+10=1729	1465	3	951+702+10=1663	537	1	895+664+10=1569	0.1	0.06
21	1066+788+10=1864	319	4	1062+805+10=1877	1378	1	992+746+10=1748	0.07	0.07
22	1225+955+10=2190	629	4	1168+899+10=2077	1058	1	1106+848+10=1964	0.12	0.06
23	1265+991+10=2266	351	4	1235+952+10=2197	150	2	1151+880+10=2041	0.1	0.08
24	1012+733+10=1755	88	4	963+693+10=1666	1022	2	842+565+10=1417	0.2	0.2
25	1090+794+10=1894	2310	4	1081+784+10=1875	1165	3	936+645+10=1591	0.2	0.2
26	1495+1184+10=2689	652	5	1452+1118+10=2580	1652	3	1310+1000+10=2320	0.2	0.1
27	1579+1273+10=2862	1930	5	1507+1177+10=2694	322	2	1353+1034+10=2397	0.2	0.1
28	1619+1274+10=2903	887	5	1662+1324+10=2996	1604	2	1484+1151+10=2645	0.09	0.1
29	2024+1632+10=3666	2496	5	1894+1502+10=3406	505	2	1711+1359+10=3080	0.2	0.1
30	2133+1742+10=3885	2410	5	2042+1659+10=3711	2239	3	1879+1509+10=3398	0.14	0.09

Table (16) Applying GA & PSO for(SP_1) for n=(40,(10),90), (100,(100),1000), 2000.

n	GA			PSO		
	BV	AES	T/s	BV	AES	T/s
40	3540+3110+209=6859	2071	6	3980+3517+219=7716	1627	3
50	4251+3709+238=8198	1590	6	4964+4430+236=9630	2510	4
60	6882+6201+301=13384	2769	7	8251+7561+314=16126	528	3
70	9267+8504+368=18139	2024	6	11335+10579+366=22280	160	4
80	10657+9697+384=20738	2899	8	13179+12223+392=25794	2398	5
90	17121+16055+516=33692	2760	8	20372+19299+522=40193	150	5
100	18390+17263+520=36173	2834	9	23132+22008+515=45655	108	5
200	74488+72209+1060=147757	2471	13	97732+95459+1055=194246	1051	9
300	182256+178701+1672=362629	2982	24	235682+232142+1669=469493	296	16

400	313547+308626+2201=624374	2992	36	416959+412014+2195=831168	1894	18
500	483281+477141+2746=963168	2993	47	650002+643859+2739=1296600	816	10
600	711064+704060+3273=1418397	2992	66	935656+928634+3269=1867559	687	30
700	965873+957440+3832=1927145	2994	87	1265625+1257182+3828=2526635	294	41
800	1313764+1303764+4500=2622028	2998	108	1723742+1713749+4502=3441993	769	50
900	1633756+1622623+4931=3261310	2999	136	2141634+2130518+4932=4277084	1251	59
1000	2015722+2003836+5491=4025049	3498	186	2627840+2615950+5481=5249271	639	52
2000	8382378+8358222+10910=16751510	3500	665	10626466+10602271+10911=21239648	2449	275

Table (17) Applying GA & PSO for(SP_2) for n=(40,(10),90), (100,(100),1000), 2000.

n	GA			PSO		
	BV	**AES**	**T/s**	**BV**	**AES**	**T/s**
40	3995+3538+10=7543	124	6	3960+3502+10=7472	1679	3
50	5449+4911+10=10370	166	5	5132+4600+10=9742	455	4
60	8651+7954+10=16615	1148	7	8070+7392+10=15472	687	4
70	11806+11050+10=22866	1069	7	11334+10577+10=21921	1654	4
80	14296+13364+10=27670	2324	7	13200+12247+10=25457	1372	5
90	22275+21188+10=43473	2633	7	20887+19823+10=40720	1649	5
100	24528+23416+10=47954	609	8	23436+22333+10=45779	615	5
200	100530+98251+10=198791	845	13	97903+95620+10=193533	1546	11
300	247036+243483+10=490529	1030	19	235775+232224+10=468009	1283	16
400	427003+422073+10=849086	2715	32	415911+410984+10=826905	289	21
500	674549+668424+10=1342983	1308	40	650181+644055+10=1294246	561	33
600	969093+962072+10=1931175	2615	65	935320+928297+10=1863627	2307	33
700	1317690+1309243+10=2626943	1440	76	1278391+1269956+10=2548357	755	50
800	1782192+1772207+10=3554409	2420	105	1725979+1715980+10=3441969	1102	60
900	2195113+2183998+10=4379121	2316	128	2130866+2119740+10=4250616	2374	74
1000	2714532+2702626+10=5417168	2156	134	2632796+2620899+10=5253705	1047	87
2000	10742308+10718122+10=21460440	1875	540	10641846+10617650+10=21259506	685	252

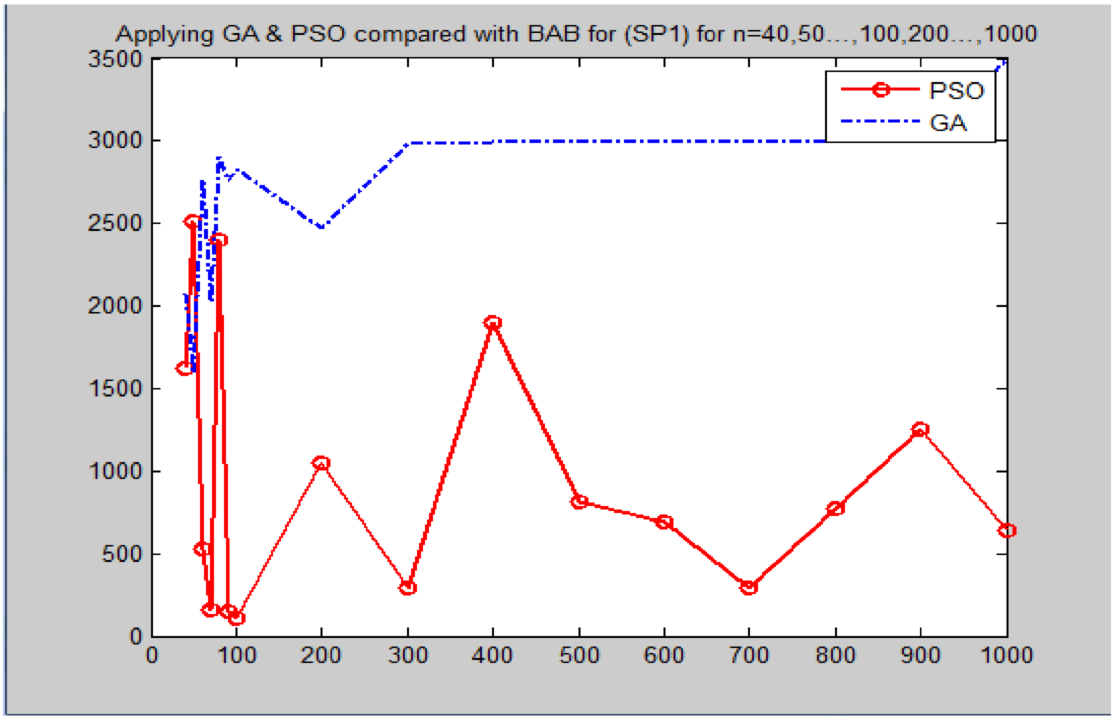

Figure (6): AES behavior for PSO & GA applied for (SP_1) with n=40,…,100,…,1000.

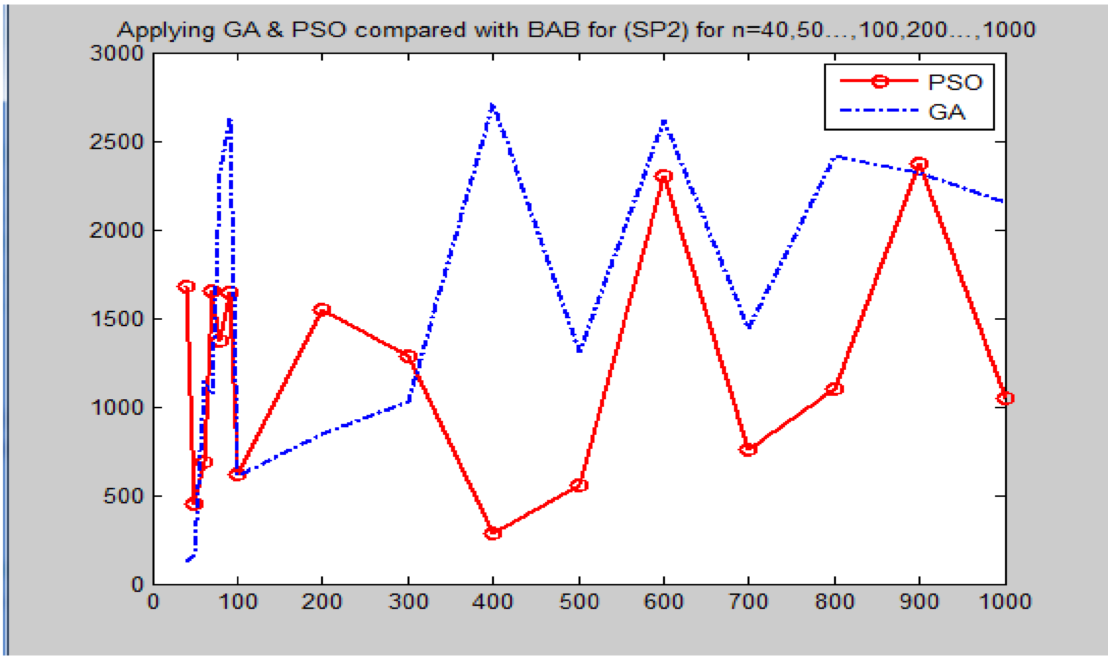

Figure (7): AES behavior for PSO & GA applied for (SP_2) with n=40,…,100,…,1000.

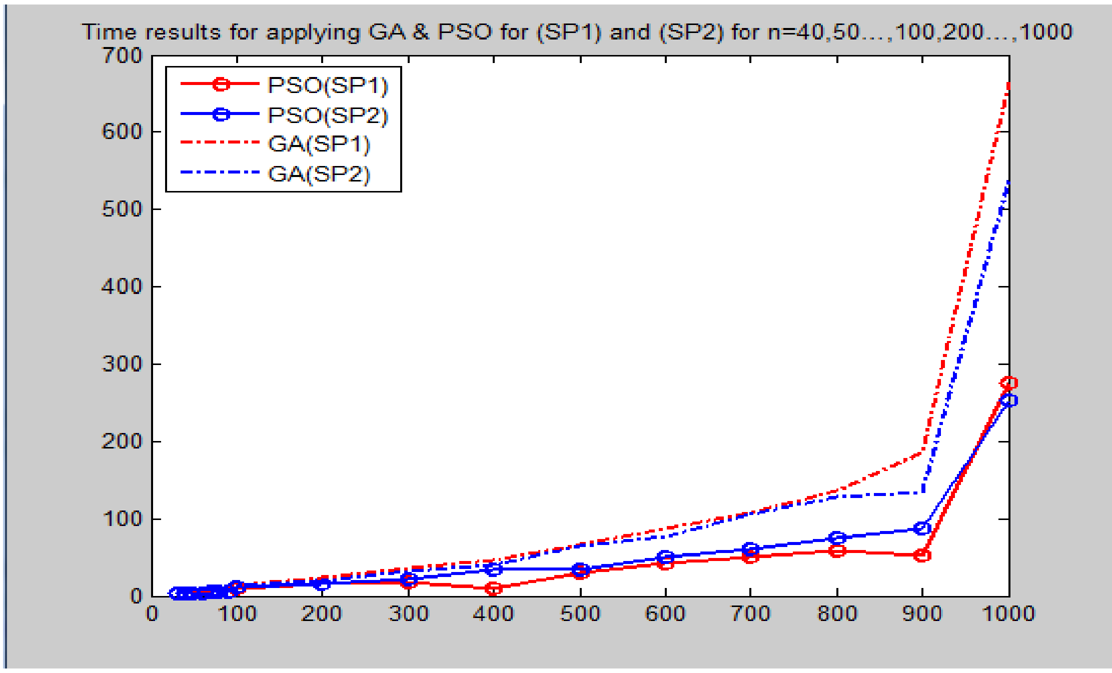

Figure (8): Time comparison for GA & PSO for (SP_1) and (SP_2) with n=40,…,100,…,1000

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar