Received: December 19, 2020. Revised: March 9, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Solving Max-Min Separable Problem Using Hybrid Particle Swarm Optimization

Hanaa Khater[1]* **Ahmed ElSawy[2]** **Assem Tharwat[3]** **Ihab El-Khodary[1]**

[1] *Faculty of Computers and Artificial Intelligence, Operations Research and Decision Support Department, Cairo University, Egypt*
[2] *Faculty of Computers and Artificial Intelligence, Computer Science Department, Benha University, Egypt*
[3] *College of Business Administration, The American University in the Emirates, Emirates*
* Corresponding author's Email: hanaa_or@yahoo.com

Abstract: The max-min separable optimization problem is depended on max-min algebra. This optimization problem focuses on the worst case for minimization of objective function. Additionally, this problem has separable max-min constraints. It was solved in another article by using difficult and complicated algebraic method for finding one solution only. In this paper, the Max-Min Separable problem under Max-Min Separable Linear Constraints (MMSMMLC) is solved by hybrid PSO. The main idea of the presented algorithms is combining GA processes with PSO algorithm to be suitable for solving MMSMMLC optimization problem under all types of constraints (equality and/or inequality). Finally, the new algorithms applied through several benchmark functions show that the new algorithms present all optimal solutions in best result with 100% and improve the optimal solution in some cases rather than algebraic method.

Keywords: Hybrid-PSO, Max-min separable function, Max-Min linear constraint, Genetic algorithm (GA).

1. Introduction

The extremely algebra has many branches such as max, min denoted by $\vee$, $\wedge$ respectively, max-min and min-max ...etc. The two main operators max or min are usually followed by plus (+) or minus (–) operator. Many authors applied max or min operator followed by min or max operation. The merger between max-min or min-max shows the worst case of max-min or the best case of min-max optimization problems.

Cuninghame-Green introduced the main definition of max-plus algebra and min-max algebra. He introduced all its algebraic properties. He also calculated the exact period value for a given matrix [1]. The extremel algebra has a special case called max-separable function. The max-separable constraint was introduced as an optimization problem with unimodel function. This has been solved and implemented as an application in service-point-location problems [2]. Zimmermann solved the same disjunctive restrictions [3]. Tharwat and Zimmermann proposed a method for solving optimization problems with a max-separable objective function and min-separable inequality constraints, and implemented that in location problems [4]. Moreover, Gavalec proposed a duality for max-separable problem of optimization, which would occur max-separable in both objective function and constraint without gap in duality [5]. Gavalec et. al presented algorithms for solving optimization problems under (max, min)-linear equations and/or inequality constraints and determined the feasible set of solution of each system of inequality constraint and system of equation [6].

Cimler et al. designed an algorithm for the unsolvability of (max, min) separable linear equation systems with coefficients in the unit interval (for fuzzy systems), or in an arbitrary real interval (general case). If the system has no solution, then the proposed algorithm computed the nearest vector to the right-hand side of the system for which the system is solvable [7].

A new definition of approximate solution was introduced by Xiao et al for the inconsistent system of max-min fuzzy relationship equations. They defined approximate solution that could minimize the equations biggest degree of dissatisfaction. They suggested a linear search algorithm on the basis of an auxiliary system with a parameter to find an approximate solution to an inconsistent max-min system [8].

In the last years, many scholars solved the max-plus or max-min function using heuristics searches and executed a comparison between PSO and genetic Algorithm for solving max-separable problem [9]. Other scholar applied max-algebra in location problem and using particle swarm algorithm for solving that problem [10]. Guo et al. solved constrained min–max optimization problems using constraint-activated differential evolution algorithm. They proved that a min–max algorithm could be used to solve a max–min problem without any algorithmic changes. Based on the theorems, they proposed a constraint-activated differential evolution to solve constrained min–max problems. The presented method consists of three components, propagation, constraint activation, and inner level evolution [11]. Khan and Rice suggested an alternative approach based on Max-Min algebra, where ternary logic functions are interpreted and realized using multiple-controlled unary gates as Max-Min expressions. They also suggested realizations of several regulated unary gates on the quantum level. Using K-maps, they implemented minimization of Max-Min expressions and then suggested a system for minimizing and synthesizing ternary reversible circuits based on a hybrid genetic algorithm [12].

From the previous literature, it clear that the direction of modern methods for optimization is the hybrid algorithm. The concept of hybrid algorithm is mixing the two heuristic algorithms or meta-heuristic algorithm with another one to improve the performance of the algorithm reaching the optimal solution. Garg presented hybrid PSO_GA to solve optimization problem with constraint by mixing the operator of GA such as selection, crossover and mutation in population in PSO [13]. Also, Barroso et al. presented a hybrid PSO with mixed GA operator (mutation) for solving optimization of laminated composite structures [14]. Ahmed et al. solved a convex economic load dispatch (ELD) problem by a hybrid PSO-GSA which associated particle swarm optimization PSO and Gravitational Search Algorithm (GSA) to find global optimal solution of that problem [15]. Yan et. al presented a modern type of the hybrid particle swarm optimization (PSO) and Quasi-Newton (QN) algorithm on CPU-GPU platform using OpenCL to accelerate the training process of (artificial neural network _ANN) [16]. Other scholars presented a modern hybrid PSO with artificial Neural Network in Bankruptcy Prediction [17]. Mitras and Anwar applied hybrid PSO algorithm with modified conjugate gradient method for some image processing [18]. Xu et al. solved vehicle routing problem with time windows by applying hybrid PSO and GA by using one operator of GA as crossover to solve that problem by obtaining a more optimal result [19]. Niazy et al. solved the capacitated vehicle routing problem using hybrid chicken swarm optimization algorithm with Tabu Search [20]. Moreover, they solved the same problem using hybrid chicken swarm optimization with genetic algorithm [21].

In real situations, we resort to the worst case of (max or min) optimization to guarantee the optimality of objective function (target) and achieve the quality of optimizing the resources. Actually, if it can achieve the target in the worst case, it will achieve the target in other cases. So the ordinary use of that type of problem is mainly in economic application or to control the resources such as the capacity of roads, electricity networks, water networks …etc.

From the previous literature we find that all researches didn't solve the max-min separable optimization problem with meta-heuristics algorithm. It solved only max-plus separable function [14, 17].

The previous work focuses on using algebraic method and not using the meta-heuristic algorithms to benefit from the power of those methods for finding the optimal solution without complexity computational.

The aim of this paper is to solve (MMSMMLC) problem by using meta heuristic Algorithms to find the optimal solution specially the (MMSMMLC) problem which is NP-hard [3]. So, the new algorithms solve (MMSMMLC) problem without converting the equality constraint into (two inequality constraint ($\leq$ and $\geq$) as the algebraic method [22]. This paper shows the abilities of meta-heuristics algorithm to achieve the optimal solution of each separable objective functions and finally compute the maximum of all.

This paper is organized as follows :- Section 2: presents the theoretical background about the Max-Min-separable optimization problem under max-min constraints and presents the algebraic method for solving that problem depending on lemmas were presented by Gavalec et.al [6] . Section 3: presents back ground about uses algorithm such as Partial Swarm Optimization algorithm and genetic algorithm. Section 4: presents an algorithm for determine maximum point in feasible set of solution for the max-min separable problem. Section 5 presents a hybrid PSO with genetic Algorithm for solving the presented problem. Section 6: tests the modified algorithms through numerical examples and presents the result. Section 7: presents the conclusion from the results and suggests the future work to be presented.

2. Theoretical background

Gavalec et al. presented the general form of an optimization problem with max-min separable objective function subject to max–min linear equality/inequality constraints, and solved that problem by algebraic method with real number. Actually, it is not easy to solve the max-min separable objective variable under max-min constraints because max-min separable constraints must be satisfied by vector of decision variables.

The algebraic method solved that problem with two separated algorithms. One, for solving that problem under inequality constraints (greater than or equal and less than or equal) in [22], and another for solving the same optimization problem under equality constraints only in [6]. The algebraic method for solving the presented optimization problem under mix constraint between equality and inequality constraint must be converted equality constraint into double inequality constraint with type greater than or equal and less than or equal.

2.1 Max-Min separable objective function under max-min linear constraint (MMSMMLC)

optimization problem with mixed (inequality and/or equality) constraint a following:

$$F(X) = \max(\min(f_1(x_1), \min(f_2(x_2), \dots, min(f_{nvar}(x_{nvar})) \tag{1}$$

Subject to:

$$max(a_{ij} \wedge x_j) = b_i, \quad i \in I1 \tag{2}$$

$$max(a_{ij} \wedge x_j) \leq b_i, \quad i \in I2 \tag{3}$$

$$max(a_{ij} \wedge x_j) \geq b_i, \quad i \in I3 \tag{4}$$

$$lx_j \leq x_j \leq ux_j \tag{5}$$

Where $f_j: R^{nvar} \to R^{nvar}$ are continuous functions, $x_j \in \boldsymbol{M}$ are decision variables, $\boldsymbol{M}$ is a feasible set of solution, $I1 = \{1,2,\dots,k\}$, $I2 = \{k+1,\dots,m\}$, $I3 = \{m+1,\dots,n\}$ $I = I1 \cup I2 \cup I3$, $I1$ for $=$ constraints and $I2$ for $\leq$ constraints , $I3$ for $\geq$ constraints, $j = \{1,2,\dots,nvar\}$, $I = \{1,2,\dots,n\}$, $x_j = (x_1, x_2, \dots, x_{nvar}) \in R$, $a_{ij} \in R$, $b_i \in R \forall i \in I, j \in J$.

2.2 Algebraic Method.

The algebraic method was investigated by [22] solved the (MMSMMLC) optimization problem without equality constraints. The algorithm depends on determining the feasible set of solution of each decision variable by closed interval using the following *Lemma 1* and *Lemma 2* after that select the minimum of objective function and set that as optimal solution. That algorithm is manipulating an inequality constraint to find one optimal solution only even if the objective function has multi-optimal solutions. The algebraic method is valid in case of linear objective and uni-modal nonlinear objective function, but in case of multimodal nonlinear objective function all possible optimal solution can't be founded. Moreover, it can't solve problem which has inequality constraint and equality constraint in the same problem without converting each equality constraint into twice inequality constraints. On the other hand, the same authors investigated an algebraic algorithm for solving that type of problem under equality constraint only and started search for optimal solution with feasible point which it was determined by the following lemmas 3 and 4

The following four lemmas are presented and proved by Gavalec et al [6] in details.

Lemma 1:

$$V_{ij}^{\leq} = \{x_j; (a_{ij} \wedge x_j) \leq b_i \text{ and } lx_j \leq x_j \leq ux_j \} \tag{6}$$

Therefore $V_{ij}^{\leq}$ is the feasible set of the system and its upper bound represents the maximum element for the feasible set M satisfying Eq. (3) and Eq. (5).

For arbitrary $i \text{ and } j$, $V_{ij}^{\leq}$ can be reformulated as follows:

$$V_{ij}^{\leq} = \begin{cases} [lx_j, ux_j] & if a_{ij} \leq b_i \\ [lx_j, ux_j \wedge b_i] & if a_{ij} > b_i \text{ and } b_i \geq lx_j \\ \emptyset & if a_{ij} > b_i \text{ and } b_i < lx_j \end{cases} \tag{7}$$

Since $x_j \in \boldsymbol{M} \Rightarrow x_j \in \bigcap_{i \in I_2} V_{ij}^{\leq} \forall j \in J$,

Therefore $V_{ij}^{\leq} = \emptyset \Leftrightarrow \boldsymbol{M} = \emptyset$,

Received: December 19, 2020. Revised: March 9, 2021.

So, ux_{ij} can be defined as an upper bound of $V_{ij}^{\leq} \neq \emptyset$ as a following:

$$ux_{ij} = \begin{cases} ux_j & a_{ij} \leq b_i \\ ux_j \wedge b_i & a_{ij} > b_i \ and \ b_i \geq lx_j \end{cases}$$

Therefore the upper bound $ux_j^{max} = \min_{i \in I2} ux_{ij}$ the element ux_j^{max} is the maximal element of feasible set of solution $\boldsymbol{M}$ satisfying Eq.(3) and Eq. (5) and replacing by new upper bound ux_j^{max} ■

Remark: For any $x_j \in [lx_j, ux_j]$ if $x_j \leq ux_j^{max}$ we will have to find the solution for the system of Eq. (2) and Eq.(4) otherwise there is no solution for MMSMMLC problem.

Lemma 2:

The following lemma determines the set of feasible solution for greater than or equal constraint Eq. (4) with regard of satisfying the system of equation Eq. (3) and Eq. (5) by defining $V_{ij}^{\geq}$ as follows:

$$\boldsymbol{V_{ij}^{\geq}} = \{\boldsymbol{x_j} : (\boldsymbol{a_{ij}} \wedge \boldsymbol{x_j}) \geq \boldsymbol{b_i} \& \ \boldsymbol{lx_j} \leq \boldsymbol{x_j} \leq \boldsymbol{ux_j}\} \quad (8)$$

Therefore $\boldsymbol{V_{ij}^{\geq}}$ is the feasible set of the system and its upper bound represents the maximum element for the feasible set $\boldsymbol{M}$ satisfying (2) and (4)

For arbitrary i and j, $V_{ij}^{\geq}$ can be reformulated as follows:

$$V_{ij}^{\geq} = \begin{cases} [b_i \vee lx_j, ux_j] & if \ a_{ij} \geq b_i \ \& \ b_i \leq ux_j \\ \emptyset & if \ a_{ij} < b_i \ or \ b_i > ux_j \end{cases} \quad (9)$$

Where $x_j \in M = \bigcap_{i \in I3} \bigcup_{j \in J} V_{ij}^{\geq}, \forall j \in J$

Therefore $V_{ij}^{\geq} = \emptyset \Leftrightarrow M = \emptyset$ ■

The following lemmas (3) and (4) present the properties of set $\boldsymbol{M}$ satisfying Eq.(2) and Eq.(5)

Let us define:

$$I_j = \{i \in I1 \ and \ a_{ij} > b_i\}$$

Which represent the position of constraint i for any fixed $j \in J$ satisfying $a_{ij} > b_i$;

$S_j(x_j) = \{k \in I1 \ ; a_{kj} \wedge x_j = b_k$

Which represent the position of constraint i for any fixed $j \in J$ satisfying $a_{kj} \wedge x_j = b_k$;

$M = \{x_j \in M \ and \ x \leq ux_j\}$

Lemma 3:

$$V_{ij}^{=} = \{x_j ; (a_{ij} \wedge x_j) = b_i \ and \ lx_j \leq x_j \leq ux_j\} \quad (10)$$

For arbitrary i and j, $V_{ij}^{=}$ can be reformulated as follows:

$$V_{ij}^{=} \begin{cases} \{b_i\} & if a_{ij} > b_i \ and \ b_i \leq ux_j \\ [b_i, ux_j] & if a_{ij} = b_i \ and \ b_i \leq ux_j \\ \emptyset & if a_{ij} < b_i \ and b_i > ux_j \end{cases} \quad (11)$$

$V_{ij}^{=}$ represents the new feasible set of solution including the optimal solution.

Lemma 4:

Set for all $i \in I1$ and $j \in J$ which $x_j^{(i)}$ is a maximum element in $\boldsymbol{M}$ for arbitrary i

$$x_j^{(i)} = \begin{cases} b_i & if \ a_{ij} > b_i \ and \ b_i \leq ux_j \\ ux_j & if \ a_{ij} = b_i \ and \ b_i \leq ux_j \ or \ V_{ij}^{=} = \emptyset \end{cases} \quad (12)$$

$$I_j = \{i \in \ I1 \ \ and \ a_{ij} \geq \ b_i\} \forall \ j \in J$$

And let $\hat{x}_j$ is a maximum element in $\boldsymbol{M}$

$$\hat{x}_j = \begin{cases} \min_{k \in I_j} x_j^{(k)} & if \ I_j \neq \emptyset \\ ux_j & if \ I_j = \emptyset \end{cases} \quad (13)$$

Let $S_j(\hat{x}_j) = \{k \in I1 \ ; x_j^{(i)} = \hat{x}_j\} \ \forall j \in J$

And the following statements hold:

$$\hat{x}_j \in \boldsymbol{M} \Leftrightarrow x_j \in \bigcup_{J \in j} S_j(\hat{x}_j) = I1$$

If $\boldsymbol{M} \neq \emptyset$, then $\boldsymbol{M}$ and $\forall \ x \in \boldsymbol{M} \Leftrightarrow x \leq \hat{x}$,

So $\hat{x}$ is maximum element in M

3. Background about PSO and GA algorithms

Particle swarm optimization (PSO) was investigated in 1995 as a stochastic technique of global search. PSO is a method inspired by the behaviour of birds. The following sub-sections presents some types of meta-heuristics algorithm which will be used in this paper such as Particle Swarm Optimization algorithm (PSO) and Genetic Algorithm (GA). These algorithms will be modified to solve the MMSMMLC. Moreover, this paper will present a hybrid particle swarm optimization and genetic algorithm to solve MMSMMLC problem.

3.1 Particle swarm optimization (PSO)

The PSO algorithm is stimulated from swarm performance such as bird flocking. PSO has been widely used in optimization problem as a type of meta-heuristics algorithm for solving NP-hard problem. The parameters used in PSO algorithm are defined in Table 1 to explain the idea of PSO algorithm.

The PSO Algorithm consists of $npop$ particle $i \in npop$ fly around d dimension to search in feasible region to find the optimal solution. Initialize the position of each particle $x_i \in [x_{min}, x_{max}]$ and $v_i \in [v_{min}, v_{max}]$ randomly. The velocity v_i and particles x_i must be updated according to the following Eq. (14) and Eq. (15) in each iteration.

$$v_i(t+1) = w.v_i(t) + c_1.r_1\left(x_i^{best}(t) - x_i(t)\right) + c_2.r_2\left(x_i^{global}(t) - x_i(t)\right) \quad (14)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (15)$$

Where r_1 and r_2 are uniform random variable in range [0, 1], c_1 and c_2 are constant which was known as accelerations coefficient and w was known as inertia weight within range [0, 1]. Inertia weight was affecting the type of solution. If inertia weight is small, it will fall on local optimal. Also if inertia weight is large, the global optimal will be found.

After each iteration select the best particle of each swarm x_i^{best} according to the nature of problem (Max or Min) and after each iteration select the global best particle x_i^{global}. The benefits of two variable x_i^{best} and x_i^{global} are played as memory of algorithm to improve the solution in the next iteration till reaching the optimal solution. Algorithm 1 is briefing the pseudo code of Particle swarm Optimization (PSO) Algorithm.

Algorithm 1 : pseudo code Particle Swarm Optimization

Initialize randomly position and velocity of all particles.
Do
For t to t_{max}
For i to $npop$
Set x_i^{best} and x_i^{global}
Calculate particle velocity according to Eq.
Update particle position according to Eq.(15)
Evaluation the objective function value (fitness value)
End
End
While A satisfactory optimal solution has been found

Table 1. Parameters that using in different PSO Algorithms

$I1$	Number of equality = constraints
$I2$	Number of less than or equal ≤ constraints
$I3$	Number of greater than or equal ≥ constraints
a_{ij}	Right hand side variable
b_i	Left hand side variable
$x_j \in [lx_j, ux_j]$	Decision variable x_j must be between lower bound lx_j and upper bound ux_j
$v_j \in [lv_j, uv_j]$	Velocity v_j of particle must be between lower bound lv_j and upper bound uv_j
t_{max}	Maximum number of iterations
$npop$	Number of particle
$w \in [wmin, wmax]$	Inertia weight w must be between minimum weight $wmin$ and maximum weight $wmax$
PC rate	Crossover rate, $PC \in [0,1]$
PM rate	Mutation rate, $PM \in [0,1]$
x_i^{best}	In each iteration select the best particle of each swarm x_i^{best} according to natural of problem (Max or Min) and after each iteration select
x_i^{global}	global best particle x_i^{global} for all iterations
r_1 and r_2	are uniform random variables in range [0,1]
c_1 and c_2	are constant

3.2 Genetic algorithm

The genetic algorithm (GA) is an important stochastic search algorithm used in the last century to solve optimization problems. Different problems such as control problem, image processing, and route planning at construction sites have been commonly and effectively implemented [23]. GA varies in the following characteristics from most traditional calculus-based search algorithms: no constraint on the continuity or discreteness of the search space, parallel population computation. The idea of GA comes from the Evolutionary Algorithms (EA) family. So, it produces solutions through the use of natural influenced evolution techniques in optimization problems [24].

Genetic algorithm starts with initializing the offspring randomly then, for each iteration executing several processes such as selection process which selects the good parents according to selection method, crossover process and mutation process. All of those processes are executed by different strategies and then the result of these processes tested fitness of objective function. Finally, select the accepted solution according to objective function (maximize or minimize) and repeating the previous steps till reaching the maximum number of iterations. The following pseudo code explains the (GA) algorithm, see Algorithm 2

Algorithm 2: pseudo code Genetic Algorithm

Initialize offspring
Do
For $t \le t_{max}$
  Selection
  Crossover
  Mutation
  Test fitness of objective function
  Accepted and produced new offspring
End
While A satisfactory optimal solution has been found

4. Proposed algorithms.

The MMSMMLC problem will be solved by using algebraic method which is described in section (2.1) then we will present Max-Point Algorithm to determine the feasible set of solution for all types of constraints greater than or equal, less than or equal and equality constraints. This algorithm helps us to solve MMSMMLC problem by new Hybrid-PSO with genetic algorithms with (crossover and mutation operation or both) which are called MMS-PSO-Cross, MMS-PSO-Mut and MMS-PSO-GA for solving max-min separable optimization under max-min linear constraints which will be listed in the next section. Moreover modify PSO algorithm and LAPSO algorithm to be suitable for solving MMSMMLC problem.

5. Max-point algorithm (MPA)

The idea of this algorithm (MPA) depends on the previous *Lemma 1* and *Lemma 3* and *Lemma 4*. That algorithm split all constraint into three parts I1 for equality constraint, I2 for less than or equal constraint and $I3$ for greater than or equal constraint. So, the following algorithm starts with $I2$ constraint and determines maximum point of the feasible region according to *Lemma (1)* then from the produced new maximum point tested of satisfaction for all constraint $I1$ by decreasing that maximum point to produce feasible region for $I1$ and $I2$ constraint. Finally, examine that new maximum point for satisfying of $I3$ else if the new maximum point doesn't satisfy all constraints then the set of feasible solution M is empty, in the other words if $a_{ij} < b_i \ or \ b_i > ux_j$ then $\mathrm{M} = \emptyset$ for all $i \in I3$ and $j \in nvar$ according to Lemma 2.

Note, The following Algorithm obtains maximum point in a set of feasible solution M for decision variables x_j . That algorithm whose efficient in case of equality constraints and less than or equal ($\le$) constraints but that is not effective in case of greater than or equal ($\ge$) constraints to change the upper limit of decision variable because the nature of stochastic search can reach optimal solution without need to start search with feasible point. The Max-Point-Algorithm is explained in details in the following steps and Algorithm 3 for its pseudo code.

Step 1: Define $I1, I2, I3, a_{ij}, b_i, j \in nvar\ , lx_j\ , ux_j$ and $i\ \in\ I1, I2\ \ and\ I3$

Step 2: Start with less than or equal constraints and determine the maximum element of feasible set of solution which satisfies all less than or equal constraints defined as $Xmax$, for all $i \in I2$ and $j \in nvar$ then set $xu = \mathrm{Xmax}$ according to *Lemma 1*, if $V_{ij}^{\le} = \emptyset$ for any $i \in I2$ and j ∈ nvar then $\mathrm{M} = \emptyset$. So, there is no feasible solution, Stop.

Step 3: Determine the maximum point $ux(new)$ of equality constraints according to *Lemma 3* and *Lemma 4* and set $ux = ux(new)$ then check if $a_{ij} > b_i$ and $b_i < lx_j$ for all $i \in I2$ and $j \in nvar$ then $\mathrm{M} = \emptyset$. So, there is no feasible solution, Stop.

Step 4: Check the feasibility of new maximum point xu in greater than or equal constraints by checking if $a_{ij} < b_i \ or\ \ b_i > ux_j$ for all $i \in I3$ and $j \in nvar$ then $\mathrm{M} = \emptyset$ so there is no feasible solution, Stop. Otherwise, return ux Stop.

Algorithm 3: pseudo code **Max-Point Algorithm**
Initialize $I1, I2, I3$, a_{ij} , b_i, $j \in nvar, i \in I1\ or\ I2\ or\ I3,\ lx_j, ux_j$ **If** $I2 \neq 0$ **For** all $i \in I2$ and $j \in nvar$ Define $V_{ij}^{\le}$ to find maximum point ($Xmax$) of feasible region according to lemma (1) then $ux == Xmax$ **If** $a_{ij} > b_i \& b_i < lx_j$

Then $M = \emptyset$,so there is no feasible solution , **Stop**
End
End
End
If *I1≠0*
For all $i \in I1$ and $j \in nvar$
Find the maximum point of feasible region according to *Lemma* 3 and *Lemma* 4 defined as $ux(new)$
If $a_{ij} > b_i$ & $b_i \leq xu_j$
then $x_{ij} = b_i$
End
If $a_{ij} == b_i$ && $b_i \leq xu_j$
Then $x_{ij} \leq ux_j$
End
End
For $i \in I_j$ & $I_j = \{i \in I1$ &$a_{ij} > b_i\}$
$ux(new)_j$ == min(x_{ij})
End
End
$ux == ux(new)$
End
For all $i \in I1$ and $j \in nvar$
If $a_{ij} > b_i$&$b_i < lx_j$
Then M = ∅, so there is no feasible solution, **Stop**
End
End
End
if *I3* ≠ 0
For all $i \in I3$ and $j \in nvar$
If $a_{ij} < b_i \ or \ b_i > ux_j$ test the satisfaction of xu for all constraint $I3$;
then M = ∅ , so there is no feasible solution, **Stop**
Else
Return xu , **Stop**
End
End
End

6. Max-min separable PSO-GA algorithm (MNS-PSO-GA)

The MMSMMLC problem was solved by two separated algorithms. The first algorithm which solved that problem with only inequality constraints [22] and the second algorithm which solved the optimization problem including equality constraints only using algebraic method [6]. The proposal algorithm solved MMSMMLC problem with all types of constraint (≤, ≥ and =) by using hybrid particle swarm optimization with genetic algorithm as a stochastic search of all optimal solutions.

Actually, any type of PSO algorithms can't solve the MMSMMLC optimization problem without modification for many reasons: the first reason is PSO was designed as an algorithm optimization problem without any constraints and treated this problem by a penalty function [25] to guarantee satisfying all constraints. The second reason is the max-min separable constraint in case of equality constraint can't be suitable without transferring equality constraint into pair of inequality constraints. The third reason is the nature of max-min separable objective function which requires finding the optimal of the separated objective functions $f_j(x_j)$. Finally, find the maximum of all separable functions. So, the PSO optimizes only *F(x)* and neglected optimizing each separating objective function $f_j(x_j)$ included in *F(x)* due to the previous reasons the next modified algorithms to optimize separates $f_j\left(x_j\right)$.

The proposed algorithm finds the optimal solution of MMSMMLC problem without transforming the equality constraint into pair of inequality constraints as in algebraic method. The MNS-PSO-GA Algorithm is considered as hybrid-PSO which mixed PSO with some process of GA to achieve the benefits of PSO and GA for solving optimization problem. The MNS-PSO-GA Algorithm will start with determining the set of feasible solution by applying Max-point-Algorithm (MPA) to decrease the chance of search outside the feasible region. So, it doesn't need penalty function because the search of solution applies within feasible region, then determine the dynamic weight Eq. (22) recommended by literature [26]. Next, it implements PSO algorithm for each population and executes crossover process and mutation process. Finally, test the fitness of objective function to find optimal solution. The following steps explain the next steps in details.

Step 1: This algorithm starts by entering the parameter of PSO Algorithm, t_{max} , w_{max} ,w_{min} , $c1, c2, r1, r2$

Step 2: Enter the parameters of constraints a_{ij} ,b_i where $i \in I , d \in nvar, lx_i, ux_i$ and objective function $F(X) = max(min(f_1(x_1), minf_2(x_2), \dots, minf_d(x_d)$.

Step 3: Apply Algorithm Max-Point-Algorithm to determine the new upper limit of decision variable according to feasible set of solution, see Algorithm (3).

Step 4: Initialize randomly the decision variable $x_i \in [lx_i, ux_i]$, $v_i \in [lv_i, uv_i]$, For all $i \in npop$.

Step 5: Set x_i^{best} for each particle swarm, Set $x^{global} = ux$ for first iteration to guarantee start search with feasible point especially in case of equality constraints.

Step 6: Apply the crossover process in particle position $hild1(x_t)$, $Pchild2(x_t)$ and velocity as a following

$$Pchild1(x_t) = PC \times PParent1(x_t) + (1 - PC) \times PParent2(x_t) \quad (16)$$

$$Pchild2(x_t) = PC \times PParent2(x_t) + (1 - PC) \times PParent1(x_t) \quad (17)$$

Which PC rate of crossover $PC \in [0,1]$, update velocity $Pchild1(v_t)$, $Pchild2(v_t)$

$$Pchild1(v_t) = \frac{PParent1(v_t)+PParent2(v_t)}{|PParent1(v_t)+PParent2(v_t)|} |PParent1(v_t)| \quad (18)$$

$$Pchild2(v_t) = \frac{PParent1(v_t)+PParent2(v_t)}{|PParent1(v_t)+PParent2(v_t)|} |PParent2(v_t)| \quad (19)$$

Step 7: Apply the mutation process with given mutation rate and update velocity v_i and position x_i of particle x_i for all npop as following equations:

$$x_i = Mu \times (ux_i - lx_i) + lx_i \quad (20)$$

$$v_i = Mu \times (uv_i - lv_i) + lv_i \quad (21)$$

Mu is a uniform random variable [0,1]

Step 8: Calculate dynamic weight w is weight linear decreasing function which was presented [26] and proved the efficiency of that weight to find solution.

$$w = (w_{max} - w_{min}) \times \frac{t_{max}-t}{t_{max}} + w_{min} \quad (22)$$

for all t_{max}

Step 9: update v_i and x_i according Eq. (14) and Es. (15) for all $i \in npop$

Find the optimal value of each separable objective function separately

Step 10: Evaluate fitness of separable objective function for $i \in npop$

$F(X_i) = (minf_1(x_{i1}), minf_2(x_{i2}), \dots, minf_d(x_{id}))$, $SwapBest = x_i$

Step 11: Update x_i^{best} $\forall d \in nvar$ which $nvar$ is a dimension of decision variable

If $f_d(x_{id}) <= f_d(x_{id}^{best})$;
then $SwapBest^{(d)} = x_{id}$; **End**
If SwapBest satisfy all constraint,
then $x_i^{best} = SwapBest$; End
Set $SwapGlobal = x_i^{global}$

Step 12: Update x_i^{global} $\forall d \in nvar$ which $nvar$ is a dimension of decision variable

If $f_d(x_{id}^{best}) <= f_d(x_i^{global})$,
then $SwapGlobal^{(d)} = x_{id}^{best}$; **End**
If $SwapGlobal$ satisfy all constraint,
then $x_i^{global} = SwapGlobal$; **End**

Repeat Step 6 to Step 12 till to end number of maximum iteration t_{max}

Step 13: maximum of $F(X_i^{global})$ is objective function with optimal solution x_i^{global}. Stop

Algorithm 4: Pseudo code Max-Min Separable optimization problem with max-min separable constraint **Hybrid-PSO-GA Algorithm (MNS-PSO-GA)**

```
Read parameter of algorithm
    npop, t_max, w_max, w_min, c1, c2, r1, r2
Read parameter of optimization problem a_ij, b_i
where I ∈ I1 ∪ I2 ∪ I3, nvar and objective
    function
F(X)
= max(minf_1(x_1), minf_2(x_2), ..., minf_nvar(x_nvar))
lx = {lx_1, lx_2, ..., lx_nvar}, ux =
    {ux_1, ux_2, ..., ux_nvar}.
Determine maximum point of set of solution and set
  as ux by applying(MPA) algorithm
Initialize randomly position x_0 and velocity v_0 of all
particles.
Do
For t ≤ t_max
Applying crossover process in position according to
Eq. (16) and Eq. (17) and velocity according
to Eq. (18) and Eq. (19).
  And/Or
Applying mutation process in position according to
Eq. (20)and velocity according to Eq.
    (21)
Set weight w according Eq. (22)
For i ≤ npop
  Set x_i^best , and x_i^global
  Evaluation the objective function value (fitness
value)
  Set SwapGlobal = x_i^global
  For d = 1:nvar
    If minf_1(x_(i)d) ≤ minf_1(x_(i)d^best)
      SwapGlobal(d) = x_(i)d
      If SwapGlobal satisfy all constraint I
     Then
       x_i^global = SwapGlobal
     End
    End
  End
End
End
```

While A satisfactory optimal solution x_i^{global} has been found
Calculate maximum of F(x^{global}) , **Stop**.

The processes of genetic algorithm (crossover and mutation) are applied into three Hybrid-PSO algorithms, the first algorithm applies only crossover process which is called MNS-PSO-GA-Cross and the second algorithm applies only mutation process which is called MNS-PSO-GA-Mut and the third algorithm applies two processes (crossover–mutation) which are presented in the previous algorithm called MNS-PSO-GA

Similarly, in PSO algorithm can be modified to be suitable for solving MMSMMLC by applying the Max-Point Algorithm and continuing the steps of PSO algorithm, then testing the fitness of objective function to reach optimal solution which can be called as MNS-PSO. On the other hand, LOAPSO algorithm [27] is a type of PSO which can be modified to solve MMSMMLC in the same way and called as MNS-LOAPSO. In this paper, all presented algorithms are compared with algebraic method [22] and will observe the efficiency of the proposal algorithms for solving MMSMMLC problem through numerical examples.

7. Numerical examples and results

This section presents benchmark P1 to P5 were presented in [22] and the new numerical examples P6 to P10 present to test the proposal algorithms. The benchmark functions and numerical examples present all types of optimization problem to test the proposal algorithms such as multimodal objective function in P3, P7, P9 and P10. Moreover, they presented all types of constraints such as equality constraints only through P1 to P3, mixed constraints (equality and inequality in the same problem) through P4,P6,P8 and P9, inequality constraint (greater than or equal and less than or equal in the same problem) through P4 and P10. Finally, the numerical examples present one type of inequality constraints greater than or equal in P5 and less than or equal constraints in P7. The result presents the difference between algebraic method [22] and the presented Algorithms. The following numerical examples denote the capacity x_j as decision variables that affect objective function (cost function) such as delivery cost in benchmark P2 and P3 and taxes in benchmark P1, P4… etc. The new proposal algorithms obtain the value of x_j to optimize the objective function in the pessimistic case (maximum of minimum cost function). If the cost function value is less than zero, the cost is considered equal zero.

P1:

$F(X) = \max_{j\in J}(\min f_j(x_j))$ Where $j = \{1, 2,\ldots,7\}$

$f_1(x_1)$ = -0.2057 x_1 +1.451,
$f_2(x_2)$ = 4.8742x_2+1.5346,
$f_3(x_3)$ = 2.8848x_3 + (-3.6121),
$f_4(x_4)$ = 0.9861x_4 + (-0.9143),
$f_5(x_5)$ = 1.7238x_5 + (-2.0145).
$f_6(x_6)$ = 1.1737x_6 + 1.9373
$f_7(x_7)$ = -3.3199x_7 + (-4.8467)

Subject to:

max(6.1221∧x_1,9.0983∧ x_2,9.5032∧ x_3, 6.0123∧ x_4, 6.1112∧ x_5, 4.1221∧ x_6, 5.5776∧ x_7) = 6.1221
max(8.2984∧x_1,3.392∧ x_2,2.5185∧ x_3, 1.1925∧ x_4, 8.9742∧ x_5, 6.7594∧ x_6, 8.6777∧ x_7) = 7.0955
max(2.0115 ∧x_1 ,6.3539 ∧x_2 , 4.4317 ∧x_3 , 7.7452∧x_4, 0.6465∧x_5, 9.4098∧x_6, 1.3576∧x_7) = 6.3539
max(6.4355∧x_1, 1.6404∧x_2 , 3.185∧x_3 , 3.7361∧x_4, 7.2605∧x_5, 3.0201∧x_6, 5.3808∧x_7) = 6.4355
max(8.5668∧x_1, 5.831∧x_2, 2.5146∧x_3, 8.7804∧x_4, 3.7709∧x_5, 4.477∧x_6, 2.3007∧x_7) = 6.5712
max(5.2690∧x_1, 9.69∧x_2, 5.1598∧x_3, 9.2889∧x_4, 6.1585∧x_5, 1.0786∧x_6, 7.0121∧x_7) = 7.0121
$Xl \le X \le Xu$;
lX = (0, 0, 0, 0, 0, 0, 0), uX = (10, 10, 10, 10, 10, 10, 10)

P2:

$F(X) = \max_{j\in J}(\min f_j(x_j))$ where $j = \{1,2,\ldots,6\}$

$f_1(x_1) = (x_1 - 3.3529)^2$
$f_2(x_2) = (x_2 - 1.4656)^2$
$f_3(x_3) = (x_3 - 5.6084)^2$
$f_4(x_4) = (x_4 - 5.6532)^2$
$f_5(x_5) = (x_5 - 6.1536)^2$
$f_6(x_6) = (x_6 - 6.5893)^2$

Subject to:

max(3.694∧x_1,0.874∧ x_2,0.5518∧ x_3, 4.6963 ∧ x_4, 2.123∧ x_5,1.4673 ∧ x_6) = 4.0195
max(1.9585∧x_1,8.347∧ x_2, 5.815∧ x_3, 8.5545∧ x_4, 8.9532∧ x_5, 8.7031∧ x_6)= 7.2296
max(1.3207∧x_1,8.961∧x_2 ,1.5718∧x_3 , 3.7155∧x_4, 0.1555∧x_5,4.3611∧x_6) = 4.2766
max(8.4664∧x_1,9.1324∧x_2 ,6.6594∧x_3 , 2.5637∧x_4, 6.0204∧x_5, 6.0846∧x_6) = 6.6594
max(2.4219∧x_1,9.6081∧x_2,1.9312 ∧x_3, 2.5218∧x_4, 1.3976∧x_5, 4.1969∧x_6) = 4.1969
max(1.1172∧x_1, 3.6992∧x_2, 7.5108∧x_3, 4.7686∧x_4, 4.4845∧x_5,4.3301∧x_6) =6.9784
$Xl \le X \le Xu$;
lX = (0, 0, 0, 0, 0, 0), uX = (10, 10, 10, 10, 10, 10)

P3:
$F(X) = \max_{j\in J}(\min f_j(x_j))$ where j = {1,2,…,6}
$f_1(x_1)$ = | (x_1- 3.3529) (x_1-0.7399)|,
$f_2(x_2)$ = | (x_2- 1.4656) (x_2+0.1385)|,
$f_3(x_3)$ = | (x_3-5.6084) (x_3+4.1585)|,
$f_4(x_4)$ = | (x_4-5.6532) (x_4-1.1625)|,
$f_5(x_5)$ = | (x_5-6.1536) (x_5+2.0188)|,
$f_6(x_6)$ = | (x_6-6.5893) (x_6-1.2852)|.

Subject to:
max(3.694∧x_1, 0.874∧ x_2, 0.5518∧ x_3, 4.6963∧ x_4, 2.123∧ x_5,1.4673 ∧ x_6) = 4.0195
max(1.958∧x_1, 8.347∧ x_2, 5.815∧ x_3, 8.5545∧ x_4, 8.9532∧ x_5, 8.7031∧ x_6) = 7.2296
max(1.3207∧x_1,8.961∧x_2 ,1.5718∧x_3 , 3.7155∧ x_4, 0.1555∧x_5,4.3611∧x_6) = 4.2766
max(8.4664∧x_1,9.1324∧x_2 ,6.6594∧x_3 , 2.5637∧x_4, 6.6594∧x_5, 6.0846∧x_6) = 6.6594
max(2.4219∧x_1,9.6081∧x_2,1.9312 ∧x_3, 2.5218∧x_4, 1.3976∧x_5, 4.1969∧x_6) = 4.1969
max(1.1172∧x_1, 3.6992∧x_2, 7.5108∧x_3, 4.7686∧x_4, 6.9874∧x_5,4.3301∧x_6)=6.9784
$Xl \leq X \leq Xu$;
lX = (0, 0, 0, 0, 0, 0), uX = (10, 10, 10, 10, 10, 10)

P4:
$F(X) = \max_{j\in J}(\min f_j(x_j))$
Where j = {1, 2, 3}
$f_1(x_1)$ = 0.5x_1 + 1.4,
$f_2(x_2)$ = 0.8x_2 +5.2,
$f_3(x_3)$ = 0.7 x_3+ 3.1.

Subject to:
max (7∧ x_1,5∧ x_2,6∧ x_3) ≥ 6
max (6∧ x_1,7∧ x_2,8∧ x_3) ≥ 8
max (8∧ x_1,5∧ x_2,4∧ x_3) ≥ 4
$lX \leq X \leq uX$,
lX = (0, 0, 0), uX = (10, 10, 10)

P5:
$F(X) = \max_{j\in J}(\min f_j(x_j))$ Where j = {1, 2, 3}
$f_1(x_1)$ = | x_1-7.2|
$f_2(x_2)$ = | 0.3x_2-1.9|
$f_3(x_3)$ = | 1.5x_3 -3.2|

Subject to:
max(5∧x_1,16∧ x_2,9∧ x_3) ≤ 6
max(3∧x_1,5∧ x_2,8∧ x_3) ≤ 5
max(15∧x_1,7∧x_2 , 11∧x_3) ≤ 7
max(9∧x_1, 8∧x_2 , 13∧x_3) ≤ 8
max(11∧x_1, 3∧x_2, 3∧x_3) ≥ 4
max(5∧x_1, 6∧x_2, 9∧x_3) ≥ 5
max(5∧x_1, 3∧x_2, 12∧x_3) ≥ 4
max(3∧x_1, 5∧x_2, 9∧x_3) ≥ 2
max(5∧x_1, 5∧x_2, 6∧x_3) ≥ 5
$lX \leq X \leq uX$, lX = (0, 0, 0), uX = (10, 10, 10)

P6:
$F(X) = \max_{j\in J}(\min f_j(x_j))$ Where j = {1, 2, 3}
$f_1(x_1) = (x_1 - 4)^2 + 1.5$
$f_2(x_2) = (x_2 - 9)^2 + 2.5$
$f_3(x_3) = (x_3 - 16)^2 + 5.5$

Subject to
max(9 ∧x_1, 14∧ x_2,5 ∧ x_3) = 9
max(3 ∧x_1, 4 ∧ x_2,7 ∧ x_3) = 7
max(15∧x_1,8 ∧x_2, 7 ∧x_3) = 8
max(11∧x_1, 25∧x_2, 10∧x_3) ≤ 20
max(15∧x_1, 12∧x_2, 5 ∧x_3) ≤ 15
max(7 ∧x_1, 6 ∧x_2, 5 ∧x_3) ≤ 10
max(32∧x_1, 5 ∧x_2, 4 ∧x_3) ≥ 5
max(12∧x_1, 10∧x_2, 5 ∧x_3) ≥ 7
max(20∧x_1, 17∧x_2, 30∧x_3) ≥ 6
$lX \leq X \leq uX$, lX = (0, 0, 0), uX = (30, 30, 30)

P7:
$F(X) = \max_{j\in J}(\min f_j(x_j))$
Where j = {1, 2, 3, 4, 5}
$f_1(x_1) = |(x_1 - 5)(x_1 - 7)|$,
$f_2(x_2) = |(x_2 - 10)(x_2 - 12)|$,
$f_3(x_3) = |(x_3 - 1)(x_3 - 2)|$,
$f_4(x_4) = |(x_4 - 2.5)(x_4 - 5.5)|$,
$f_5(x_5) = |(x_5 - 1)(x_5 - 14)|$.

Subject to
max (55.1∧ x_1,10.5∧ x_2,20.3∧ x_3,51.1∧ x_4,18.2∧ x_5) ≤ 12.7
max (26.2∧ x_1,22.2∧ x_2,19.3∧ x_3,65.5∧ x_4,55.2∧ x_5) ≤18.8
max (2.5∧ x_1,17.3∧ x_2,11∧ x_3,5.1∧ x_4,18.3∧ x_5) ≤11.2
$lX \leq X \leq uX$
lX = (0, 0, 0, 0, 0),uX = (20, 20, 20, 20, 20)

P8:
$F(X) = \max_{j\in J}(\min f_j(x_j))$
Where j = {1, 2, …, 6}
$f_1(x_1) = |(x_1 - 4)|$,
$f_2(x_2) = |(x_2 - 2.5)|$,
$f_3(x_3) = |(x_3 - 3)|$,
$f_4(x_4) = |(x_4 - 1)|$,
$f_5(x_5) = |(x_5 - 1.5)|$,

$f_6(x_6) = |(x_6 - 6)|$.

Subject to:

max(15∧x_1,20∧ x_2,22∧ x_3,7∧ x_4,15∧ x_5, 30 ∧ x_6) = 7
max(6∧x_1,25∧x_2,22∧x_3,30∧ x_4,35∧x_5,31∧x_6) = 6
max(10∧x_1,20∧x_2,15∧x_3,18∧x_4,12∧x_5, 9∧x_6) ≥ 5
max(10∧x_1,8∧x_2,20 ∧x_3,12∧x_4,11∧x_5,17∧x_6) ≤ 9
$lX \leq X \leq uX$,
lX = (0, 0, 0, 0, 0, 0),uX = (10, 10, 10, 10, 10, 10)

P9:

$$F(X) = \max_{j\in J}(\min f_j(x_j))$$

Where j = {1, 2,…, 5}

$f_1(x_1) = |(x_1 - 2)(x_1 - 5)|$,
$f_2(x_2) = |(x_2^2 - 4)| + 0.5$,
$f_3(x_3) = |(x_3 - 2.5)(x_3 - 4.25)|$,
$f_4(x_4) = |(x_4^2 - 6.5)| + 0.5$,
$f_5(x_5) = |(x_5 - 9)(x_5 - 10)|$.

Subject to

max (14∧ x_1,20∧ x_2,6∧ x_3,12∧ x_4,2∧ x_5) = 12.7
max (30∧ x_1,13∧ x_2,12∧ x_3,10∧ x_4,8∧ x_5) = 8
max (3∧ x_1,10∧ x_2,13∧ x_3,21∧ x_4,15∧ x_5) = 10
max (15∧ x_1,21∧ x_2,18∧ x_3,25∧ x_4,30∧ x_5) ≤ 20
max (30∧ x_1,21∧ x_2,38∧ x_3,40∧ x_4,15∧ x_5) ≤ 35
max (15∧ x_1,19∧ x_2,11∧ x_3,18∧ x_4,20∧ x_5) ≥ 10
max (22∧ x_1,30∧ x_2,15∧ x_3,25∧ x_4,30∧ x_5) ≥ 8
$lX \leq X \leq uX$
lX = (0, 0, 0, 0, 0),uX = (20, 20, 20, 20, 20)

P10:

$$F(X) = \max_{j\in J}(\min f_j(x_j))$$

Where j = {1, 2, …, 5}

$f_1(x_1) = |sin(x_1) + 0.2| + 0.2$,
$f_2(x_2) = |sin(x_2 + 1) + 0.3| + 0.3$,
$f_3(x_3) = |sin(x_3 + 2) + 0.4| + 0.4$,
$f_4(x_4) = |sin(x_4 + 3) + 0.5| + 0.5$,
$f_5(x_5) = |sin(x_5 + 4) + 0.6| + 0.6$;

Subject to:

max (2∧ x_1,20∧ x_2,5∧ x_3,11∧ x_4,7.5∧ x_5) ≤ 15
max (23∧ x_1,13∧ x_2,20∧ x_3,1∧ x_4,30∧ x_5) ≤ 5
max (30∧ x_1,17∧ x_2,2∧ x_3,15∧ x_4,8∧ x_5) ≤ 11
max (15∧ x_1,21∧ x_2,32∧ x_3,11∧ x_4,3∧ x_5) ≤ 6
max (20∧ x_1,30∧ x_2,23∧ x_3,15∧ x_4,18∧ x_5)≥ 4
$lX \leq X \leq uX$
lX = (0, 0, 0, 0, 0),uX = (7, 7, 7, 7, 7)

The propose algorithms are applied by software Matlab R2013a and executed by hardware Intel (R) core™i5 2520M CPU. All algorithms were repeated (t) number of iteration = 100. After examining several numbers of iteration we get the suitable number of iterations = 100 to reach the optimal solutions. Moreover, we tested the number of particle swarm $npop$ = 20, 30 and 40 to find the suitable number of particle swarm npop= 30 with parameter(c1, c2) = (2,2), the inertia weight w ∈ [0.1,1]. The algorithms MNS-PSO-GA and MNS-PSO-GA-cross were applied by crossover rate 75% according to recommended [28] and the MNS-PSO-GA and MNS-PSO-GA-Mut were applied by mutation rate 5% [28,29] which is suitable according to the size of particle swarm$npop$. The benchmark functions were tested through P1 to P5 and investigated numerical example P6 which has all types of constraints such as Eq.(2), Eq.(3) and Eq.(4). The results are presented after 100 feasible trials and calculated the best result which is considered as the minimum result of feasible trials, average results and standard deviation (S.D). The starting of all proposal algorithms is applied the MPA algorithm to determine the feasible set of solution then search the optimal solutions with in the feasible region, see Table. 2.

The MNS-PSO algorithm and MNS-LAPSO algorithm could reach the optimal solution as an algebraic method in regards to the best solution for all tested numerical examples except P2 which isbetter than the Algebraic Method, see Table 3. and Fig. 1.

On the other hand, the (MNS-PSO-GA-Mut) algorithm can reach optimal solution the same as algebraic Method in regard to the best result for all numerical examples except P2 which is better than Algebraic Method. Also, the better result can be obtained for P6 in regard to the best result, see Table 3. and Fig. 1.

The three types of hybrid PSO are: - the first is (MNS-PSO-GA) algorithm which has two processes crossover and mutation. The second is (MNS-PSO-GA-Cross) algorithm which has crossover process only and the third is (MNS-PSO-GA-Mut) algorithm which has mutation process only. These algorithms improve algorithms improve the solution. The (MNS-PSO-GA) and (MNS-PSO-GA-Cross) Algorithms can reach the optimal solution the same as algebraic method in regard of the best result for all numerical examples except P1, P2 and P3 which are better than the Algebraic Method. Moreover, the (MNS-PSO-GA) only improved the solution in regard of the best result of P8 and P9 see see Table 3. and Fig. 1.

Received: December 19, 2020. Revised: March 9, 2021.

Table 2. The new upper limit of numerical examples which are found by Max-Point Algorithm

P1	$ux(new)$ = [6.5712,6.1221,6.1221,6.3539,6.4355,6.3539,7.0955]
P2	$ux(new)$= [6.6594,4.1969,6.9874,4.0195,7.2296,4.2766]
P3	$ux(new)$ = [6.6594,4.1969,6.9874,4.0195,7.2296,4.2766]
P4	$ux(new)$ = [10,10,10]
P5	$ux(new)$= [7,6,5]
P6	$ux(new)$ = [8, 9, 30]
P7	$ux(new)$= [12.7, 11.2, 12.7, 12.7, 11.2]
P8	$ux(new)$ = [7, 6, 6, 6, 6, 6]
P9	$ux(new)$ = [6, 6, 8, 6, 10]
P10	$ux(new)$= [5,5 ,5,6,5]

All proposal Algorithms could obtain multi-optimal solutions and multi-values of decision variables in P3 are $x^*_{(1)}$ and $x^*_{(2)}$ for multi-model objective
functions such as P3 are
(3.3529,1.4656,5.6084,4.0195,7.2296,4.2766),
(0.7399,1.4656,5.6084,4.0195,7.2296,4.2766)
respectively. The algebraic method obtained one solution only $x^* =$ (3.3297, 1.4689, 6.9874, 4.0195, 7.2296, 4.2766). Also, in P7 the proposal algorithms obtain 8 optimal solutions $x^*_{(1)}$ to $x^*_{(8)}$ and in P10 the proposal algorithms obtain 4 optimal solutions $x^*_{(1)}$ to $x^*_{(4)}$ which are presented the same result of objective function but the algebraic method presents only one solution [22], see Table 2.

The MNS-PSO-GA algorithm got a better optimal result of separation objective functions $min\, f_j(x_j)$ where $j = \{1,2,3\}$ are (1.4,5.2,8.7) and (0.2, 0.1, 0) for P4 and P5 respectively with corresponding decision variables are $x^* =$ (0,0,8) and $x^* =$ (7, 6, 2.1333) respectively with the same value of F(X) of P4 and P5 are 8.7 and 0.2, respectively. On the other hand the algebraic method get the same values of objective function of $F(X)$ for P4 and P5 with different values of $min\, f_j(x_j)$ are (4.4,5.2, 8.7) and (0.2,0.1,0.02) respectively with corresponding decision variables $x^* =$ (6,0,8) and $x^* =$ (7,6,2.12) respectively, see Table 2. The results of presented algorithms optimize the separated objective functions $min\, f_j(x_j)$, then maximize the result. Moreover, the presented algorithms will become suitable for solving min-min separating problem or the need of decision maker to calculate the range of objective function result in case of max-min separate problem (pessimistic case) and min-min separate problem (Optimistic case).

Generally, the (MNS-PSO-GA) Algorithm achieves the best average of all testing algorithms for all benchmark functions except P7, P9 and P10 the best average of P7 and P10 was found by (MNS-LOAPSO) and the best average of P9 was found by MNS-PSO algorithm, see Fig. 2.

8. Conclusion

The (MMSMMLC) problem included the max-min Algebraic in objective function and constraints to solve economics problem and control the recourses in the huge and complex network such as transportation network, water network, electricity network, etc. This paper presents other methods for solving

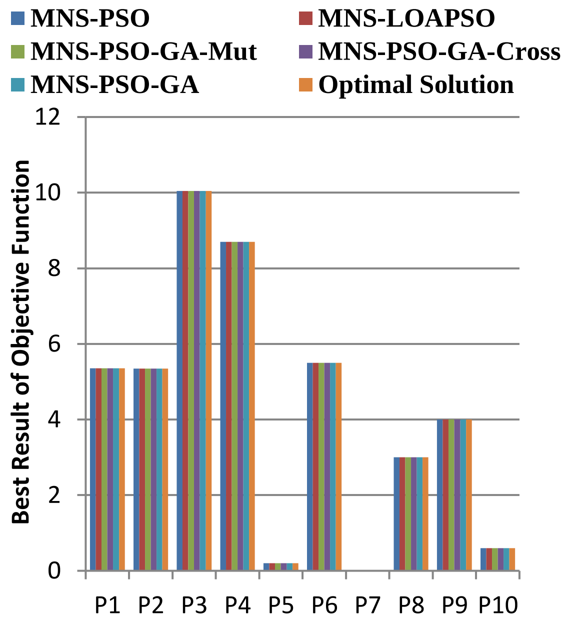

Figure. 1 The Best Result of F(X) for each numerical example and optimal solution of Algebraic method

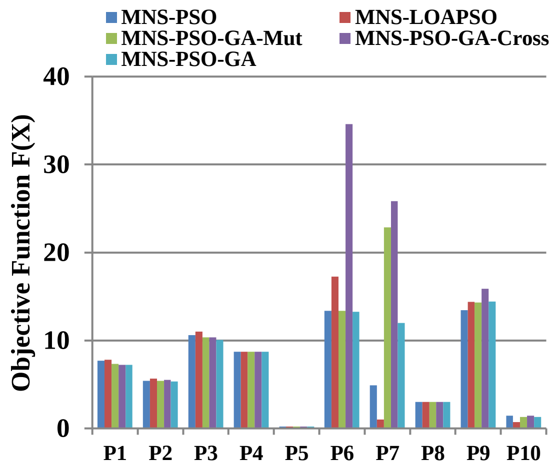

Figure. 2 The average F(X) of each numerical example

Table 3. Comparison between Algebraic Method, MNS-PSO, MNS-LOAPSO, MNS-PSO-GA-Mut, MNS-PSO-GA-Cross and MNS-PSO-GA after 100 independent feasible runs where SD is the standard deviation

Bench mark	Type of Constraints	Objective Function	Algorithms	Best result	Average	S.D	Optimal
P1	Equality	Uni-modal	Algebraic Method				5.3513
			MNS-PSO	**5.35128**	7.69721	3.14	
			MNS-LOAPSO	**5.35128**	7.78874	3.12	
			MNS-PSO-GA-Mut	**5.35128**	7.33264	2.03	
			MNS-PSO-GA-Cross	**5.35118**	**7.21128**	2.03	
			MNS-PSO-GA	**5.35118**	**7.21128**	2.02	
P2	Equality	Uni-modal	Algebraic Method				5.3488
			MNS-PSO	**5.34858**	5.404424	0.56	
			MNS-LOAPSO	**5.34858**	5.656414	1.16	
			MNS-PSO-GA-Mut	**5.34858**	5.40442	0.56	
			MNS-PSO-GA-Cross	**5.34858**	5.51611	0.96	
			MNS-PSO-GA	**5.34858**	**5.34858**	1.4×10^{-14}	
P3	Equality	Multi-modal	Algebraic Method				10.0481
			MNS-PSO	10.04812	10.61116	1.87	
			MNS-LOAPSO	10.04812	11.00449	2.42	
			MNS-PSO-GA-Mut	10.04812	10.35625	1.4	
			MNS-PSO-GA-Cross	**10.04708**	10.33334	1.63	
			MNS-PSO-GA	**10.04708**	**10.04756**	5.2×10^{-4}	
P4	Inequality (≥ ≤)	Uni-modal	Algebraic Method				8.7
			MNS-PSO	**8.7**	**8.7**	2.8×10^{-9}	
			MNS-LOAPSO	**8.7**	**8.7**	9.5×10^{-12}	
			MNS-PSO-GA-Mut	**8.7**	**8.7**	1.6×10^{-14}	
			MNS-PSO-GA-Cross	**8.7**	**8.7**	1.9×10^{-12}	
			MNS-PSO-GA	**8.7**	**8.7**	2.1×10^{-12}	
P5	Inequality (≥) only	Uni-modal	Algebraic Method				0.2
			MNS-PSO	**0.2**	**0.2**	5.3×10^{-16}	
			MNS-LOAPSO	**0.2**	**0.2**	5.3×10^{-16}	
			MNS-PSO-GA-Mut	**0.2**	**0.2**	5.3×10^{-16}	
			MNS-PSO-GA-Cross	**0.2**	**0.2**	5.3×10^{-16}	
			MNS-PSO-GA	**0.2**	**0.2**	5.3×10^{-16}	
P6	Inequality (≥, ≤) and equality	Uni-modal	Algebraic Method				5.5
			MNS-PSO	**5.5**	13.38	38.2	
			MNS-LOAPSO	**5.5**	17.24	42.26	
			MNS-PSO-GA-Mut	**5.5**	13.38	38.2	
			MNS-PSO-GA-Cross	**5.5**	34.6	69.62	
			MNS-PSO-GA	**5.5**	**13.26**	38.2	
P7	Inequality (≤) only	Multi-modal	Algebraic Method				0
			MNS-PSO	2.887×10^{-15}	4.884	10.77	
			MNS-LOAPSO	9.814×10^{-14}	1.001	4.883	
			MNS-PSO-GA-Mut	$2.887\times 10^{-15} \cong 0$	22.86	11.46	
			MNS-PSO-GA-Cross	$8.156\times 10^{-11} \cong 0$	25.84	34.96	
			MNS-PSO-GA	$2.664\times 10^{-11} \cong 0$	12	14.17	
P8	Inequality (≥, ≤) and equality	Uni-modal	Algebraic Method				3
			MNS-PSO	3	3	0	
			MNS-LOAPSO	3	3	0	

			MNS-PSO-GA-Mut	3	3	0	
			MNS-PSO-GA-Cross	3	3.015	0.15	
			MNS-PSO-GA	**2.99999**	3	3.015×10^{-5}	
P9	Inequality (≥, ≤) and equality	Multi-modal	Algebraic Method				4
			MNS-PSO	4	13.45	12.01	
			MNS-LOAPSO	4	14.38	12.45	
			MNS-PSO-GA-Mut	4	14.3	12.36	
			MNS-PSO-GA-Cross	4	15.86	12.99	
			MNS-PSO-GA	**3.9995**	14.44	12.47	
P10	Inequality (≥ ≤)	Multi-modal	Algebraic Method				0.6
			MNS-PSO	**0.6**	1.456	0.3093	
			MNS-LOAPSO	**0.6**	0.7179	0.2923	
			MNS-PSO-GA-Mut	**0.6**	1.295	0.4263	
			MNS-PSO-GA-Cross	**0.6**	1.449	0.3251	
			MNS-PSO-GA	**0.6**	1.309	0.426	

Table 2. the results of objective function F(X) and decision variable x^* using Algebraic Method and MNS-PSO-GA

	Algebraic Method	MNS-PSO-GA
P1	$F(X)$ = max(0.1039, 1.5346,-3.6121, **5.3513**, -2.0145, 1.9373,-28.4031) x^* =6.5712,0,0,6.3539,0,0,7.0955)	$F(X)$ = max(0.10398, 1.5346,-3.6121,**5.35118**, -2.0145, 1.9373,-28.40305) x^* = (6.5712 , 0, 0 ,6.3538 ,0 , 0 , 7.0955)
P2	$F(X)$ = max(0,0,1.90164,2.66898,1.15778,**5.3488**) x^* = (3.3297,1.4689,6.9874,4.0195,7.2296,4.2766	$F(X)$ = max(0, ≅0,1.901641,2.66898,1.15778,**5.348588**) x^* = (3.3529,1.4656,6.9874,4.0195,7.2296,4.2766)
P3	$F(X)$ = max(0,0,0,4.667,**10.0481**,6.9182) x^* =(0.7325,1.4689,5.5899,4.0195,7.2296,4.2766)	$F(X)$ = max(≅0, ≅0, ≅0,4.667488, ≅**10.04812** ,6.918218) $x^*_{(1)}$ = (3.3529,1.4656,5.6084,4.0195,7.2296,4.2766) $x^*_{(2)}$ = (0.7399,1.4656,5.6084,4.0195,7.2296,4.2766)
P4	$F(X)$ = max(4.4 ,5.2 ,**8.7**) x^* = (6,0,8)	$F(X)$ = max(1.4 ,5.2 ,**8.7**) x^* = (0,0,8)
P5	$F(X)$ = max(**0.2**, 0.1, 0.02) x^* = (7 , 6 , 2.12)	$F(X)$ = max(**0.2** , 0.1, 0) x^* = (7 , 6 , 2.133)
P6	$F(X)$ = max(1.5 , 2.5 , **5.5**) x^* = (4,9,16)	$F(X)$ = max(1.5 , 2.5 , **5.5**) x^* = (4,9,16)
P7	$F(X)$ = max(**0**,0,0,0,0) x^* =[5, 10,1,2.5,1] ;	$F(X)$ = max(**0**,0,0,0,0) $x^*_{(1)}$ =[5, 10,1,2.5,1] ; $x^*_{(2)}$ = [7, 10,1,2.5,1] $x^*_{(3)}$ = [7, 10,1,5.5,1] $x^*_{(4)}$ = [5, 10,1,5.5,1] $x^*_{(5)}$ = [7, 10,2,2.5,1] $x^*_{(6)}$ = [7, 10,2,5.5,1] $x^*_{(7)}$ = [5, 10,2,5.5,1] $x^*_{(8)}$ = [5, 10,2,2.5,1]
P8	$F(X)$ = max(**3**, 0, 0, 0, 0, 0) x^* = [7, 2.5, 3, 1, 1.5,6]	$F(X)$ = max(**2.9999**, 0, 0, 0, 0, 0) x^* = [6.9999, 2.5, 3, 1, 1.5,6]
P9	$F(X)$ = max(**4**, 0.5, 0, 0.5, 0) x^* = [6, 2,2.5,2.5,10]	$F(X)$ = max(**3.9995**, 0.5, 0, 0.5, 0) $x^*_{(1)}$ = [6, 2,2.5,2.5,10] ; $x^*_{(2)}$ = [6, 2,4.25,2.5,10]
P10	$F(X)$ = max(0.2, 0.3, 0.4, 0.5, **0.6**) x^*= [3.343, 4.978, 1.553, 2.76, 1.64]	$F(X)$ = max(0.2, 0.3, 0.4, 0.5, **0.6**) $x^*_{(1)}$= [3.343, 4.978, 1.553, 2.76, 1.64] $x^*_{(2)}$= [3.343, 4.978, 1.553, 0.665, 1.64] $x^*_{(3)}$= [3.343, 4.978, 3.8717, 2.76, 1.64] $x^*_{(4)}$= [3.343, 4.978, 3.8717, 0.665, 1.64]

(MMSMMLC) problem using Hybrid-PSO algorithm under any type of constraints (inequality and/or equality). The Hybrid-PSO algorithms are (MNS-PSO-GA-Cross, MNS-PSO-GA-Mut and MNS-PSO-GA) and the modified algorithms are MNS-PSO and MNS-LAPSO compared with algebraic method [22] and are tested these algorithms through several benchmark functions. From the previous results we observed that the performance of Hybrid-PSO (MNS-PSO-GA, MNS-PSO-GA-Cross and MNS-PSO-GA—Mut) algorithms are improved the solution better than the algebraic method. All meta-heuristics algorithms get multi-solutions and multi values of decision variable especially in case of multi-model objective functions but the algebraic method lead to only one optimal solution with only value of decision variable. So, the proposal algorithms avoided that shortcoming.

In future work other meta-heuristic methods can be applied to solve MMSMMLC problem with stochastic variable in right hand side or left hand side to suit the real life situations. Moreover, other meta-heuristics methods can be applied on the MSMMLC problem under nonlinear max-min constraints

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing have been done by 1st author. The supervision, methodology , validation and reviewing have been done by 2nd author, The conceptualization, writing review have been done by 3rd author and the writing-review and editing, supervision and validation have been done by 4th author.

References

[1] R. A. Cuninghame-Green, *Minimax algebra, Lecture notes in economics and mathematical systems*, Vol 166. Berlin: Springer, 1979.

[2] K. Zimmermann, "Optimization problems with unimodal fnctions in max-separable constraints", *Optimization*, Vol. 24, No. 1–2, pp. 31–41, Jan. 1992.

[3] K.Zimmermann, "Disjunctive optimization, max-separable problems and extremal algebras", *Theoretical Computer Science*, Vol. 293, No. 1, pp. 45–54, 2003.

[4] A.Tharwat, and K. Zimmermann, "One class of separable optimization problems :solution method, application", *Optimization*, Vol. 59, No. 5, pp. 619–625, Jul. 2010.

[5] M. Gavalec, and K. Zimmermann, "Duality for max-separable problems", *Central European Journal of Operations Research*, Vol. 20, No. 3, pp. 409–419, Sep. 2012.

[6] M. Gavalec, M. Gad, and K. Zimmermann, "Optimization problems under (max, min)-linear equations and/or inequality constraints", *Journal of Mathematical Sciences*, Vol. 193, No. 5, pp. 645–658, Sep. 2013.

[7] R. Cimler, M. Gavalec, and K. Zimmermann, "An optimization problem on the image set of a (max, min) fuzzy operator", *Fuzzy Sets and Systems*, Vol. 341, pp. 113–122, 2018.

[8] G. Xiao, T. Zhu, Y. Chen, and X. Yang, "Linear searching method for solving approximate solution to system of max-min fuzzy relation equations with application in the instructional information resources allocation", *IEEE Access*, Vol. 7, pp. 65019–65028, 2019.

[9] A. El-sawy, and A. Tharwat, "Comparison of Particle SWARM Optimization, Genetic Algorithm and Max separable Technique for Machine Time Scheduling Problem", In: *Proc. of The International Conf. on Mathematics and Engineering Physics*, Vol. 5, No. 5, pp. 1–10, 2010.

[10] H. M. Rabie, I. A. El-Khodary, and A. A. Tharwat, "Particle swarm optimization algorithm for the continuous p-median location Problems", In: *Proc. of 10th International Computer Engineering Conf.: Today Information Society What's Next?, ICENCO 2014*, pp. 81–86, 2015.

[11] S.-M. Guo, P.-H. Hsu, C.-C. Yang, and J. S.-H. Tsai, "Constrained min–max optimization via the improved constraint-activated differential evolution with escape vectors", *Expert Systems with Applications*, Vol. 46, pp. 336–345, Mar. 2016.

[12] M. Khan, and J. E. Rice, "Hybrid GA synthesis of ternary reversible circuits using max-min algebra", *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 32, No. 1–2, pp. 27–55, 2019.

[13] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems", *Applied Mathematics and Computation*, Vol. 274, pp. 292–305, Feb. 2016.

[14] E. S. Barroso, E. Parente, and A. M. Cartaxo de Melo, "A hybrid PSO-GA algorithm for optimization of laminated composites", *Structural and Multidisciplinary Optimization*, Vol. 55, No. 6, pp. 2111–2130, Jun. 2017.

[15] M. Ahmad, W. Ali, H. Farooq, M. Jamil, M. Ali, and A. U. Rehman, "Solving the Problem of Economic Load Dispatch for a Small Scale Power System Using a Novel Hybrid PSO-GSA Algorithm", In: *Proc. of 2018 International Symposium on Recent Advances in Electrical Engineering (RAEE)*, pp. 1–6, 2018.

[16] S. Yan, Q. Liu, J. Li, and L. Han, "Heterogeneous Acceleration of Hybrid PSO-QN Algorithm for Neural Network Training",

IEEE Access, Vol. 7, pp. 161499–161509, 2019.

[17] A. Ansari, I. S. Ahmad, A. A. Bakar, and M. R. Yaakub, "A Hybrid Metaheuristic Method in Training Artificial Neural Network for Bankruptcy Prediction", *IEEE Access*, Vol. 8, pp. 176640–176650, Sep. 2020.

[18] B. A. Mitras, and D. A. Anwar, "Using hybrid PSO algorithm with modified conjugate gradient method for some image processing", In: *Proc of AIP Conf. Proceedings*, pp. 1–16, 2019.

[19] S.-H. Xu, J.-P. Liu, F.-H. Zhang, L. Wang, and L.-J. Sun, "A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows", *Sensors*, Vol. 15, No. 9, pp. 21033–21053, Aug. 2015.

[20] N. Niazy, A. El-Sawy, and M. Gadallah, "A hybrid chicken swarm optimization with tabu search algorithm for solving capacitated vehicle routing problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 237–247, 2020.

[21] N. Niazy, A. El-Sawy, and M. Gadallah, "Solving Capacitated Vehicle Routing Problem Using Chicken Swarm Optimization with Genetic Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 502–513, Oct. 2020.

[22] M. A. M. Gad, "Optimization Problems Under (max, min)-Linear Constraints and Some Related Topics", PhD thesis, Faculty of Mathematics and Physics, Charles University Prague, 2015.

[23] M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic Algorithm: Review and Application", *International Journal of Information Technology and Knowledge Management*, Vol. 2, No. 2, pp. 451–454, 2010.

[24] K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems", *Applied Mathematics and Computation*, Vol. 212, No. 2, pp. 505–518, Jun. 2009.

[25] K. Masuda, K. Kurihara, and E. Aiyoshi, "A penalty approach to handle inequality constraints in particle swarm optimization", In: *Proc. of 2010 IEEE International Conf. on Systems, Man and Cybernetics*, pp. 2520–2525 Oct. 2010.

[26] M. A. Arasomwan, and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization", *The Scientific World Journal*,Vol. 2013, pp.1-12, 2013.

[27] S. M. A. Salehizadeh, P. Yadmellat, and M. B. Menhaj, "Local optima avoidable particle swarm optimization," In: *Proc. of 2009 IEEE Swarm Intelligence Symposium, SIS 2009 - Proceedings*, 2009, pp. 16–21.

[28] W. Chinnasri, "Adaptive Probability of Crossover and Mutation in Genetic Algorithm on University Course Timetabling Problem Adaptive Probability of Crossover and Mutation in Genetic Algorithm on University Course Timetabling Problem", In: *Proc. of CSAE 2013 : 2013 IEEE International Conf. on Computer Science and Automation Engineering*, No. March, pp. 725–728, 2015.

[29] M. Kuehn, T. Severin, J. Lippold, H. Salzwedel, and V. Nissen, "Adaptive Mutation Rate at Genetic Algorithms with Multi-Chromosome Representation in Multi-department Hospital Process Optimization", *International Journal of Computer Applications*, Vol. 182, pp. 41–52, Oct. 2018.