International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

Multi Objective Particle Swarm Optimization: A Survey

Richa Agnihotri
Department of Computer Science Engineering
UIT, RGPV
Bhopal, India

Dr. Shikha Agrawal
Department of Computer Science Engineering
UIT, RGPV
Bhopal, India

Dr. Rajeev Pandey
Department of Computer Science Engineering
UIT, RGPV
Bhopal, India

***Abstract*— In this current scenario, choosing any one choice among multiple conflicting objective is became the common problem. These problems are considered to be solved through the decision being made for the given objective is best compromising solution i.e. the solution satisfying all the objectives. Particle swarm optimization is one of meta-heuristic mechanism being used to find solution from the solution space. It belongs to evolutionary algorithm as it is population based optimization technique which figured out to be efficient, effective, flexible and easy implementation. Changes have been made in original particle swarm optimization techniques result in better solutions for multi objective optimization problems. This paper provides the basic known concepts of multi objective optimization as well as of particle swarm optimization. This results in better understanding of the concept of multi objective particle swarm optimization. Here, we also discussed the concepts of multi objective particle swarm optimization, techniques used in multi objective particle swarm optimization, approaches applied in multi objective particle swarm optimization and some of the future related work directions are also being included.**

Keywords—Multi objective particle swarm optimization(MOPSO); Multi objective optimization problems(MOOPs); Particle swarm optimization(PSO); Pareto optimality; Pareto front.

I. INTRODUCTION

In the real world scenario, optimization plays a vital role. It is the technique by which maximum or minimum value of the function is being calculated. This is being used to optimize efficiency, reliability, profit and other measures etc. [1] Optimization can be defined as the either maximization or minimization such as maximization of one function is equivalent to minimization of opposite function. In optimization, problem may be defined with various objectives, and these objectives are normally conflicting in nature and have to satisfy simultaneously. Optimization problem belong to area of Multi criteria decision making. This deals with mathematical optimization problems. Therefore, in this method, decision has been taken by best tradeoff condition between two or more conflicting objective. [21]

Particle Swarm Optimization (PSO) is meta-heuristic optimization technique. It belongs to the field of swarm intelligence and collective intelligence. It is also one of the sub field of computational intelligence. It is being used to consider the social behavior of birds. The approach defined here can be viewed as distributed behavior algorithm that operates in multi- dimensional search. In particle swarm optimization, individuals perform best from their past experiences as normally the population has memory which is being used by the individuals. This approach is been successfully implemented for continuous non-linear and binary discrete optimization. Particle swarm optimization is resulted as efficient, simple and flexible population based approach. Therefore, it can be extended to deal with multi objective optimization problems. [3][11]

In the multi objective optimization problem, particle swarm optimization can be extended in two ways. Firstly, every objective function is computed separately and secondly, all objective functions are calculated for every particle. Normally, a non-dominant is being used to guide the particles is known as leader. For each iteration, non-dominant solutions are stored in order to detect pareto optimal solution that are then been stored in memory called external archive. The key issues that are considered for better design of Multi-Objective Particle Swarm Optimization (MOPSO) are defined below: [7] [3]

(1) Evaluation of objective functions

(2) Selection of leader from archive

(3) Promotion of diversity in external archive

(4) Maintaining the archive

(5) Exchange of information through neighborhood topology.

The above mention issues are discussed later on details. The remaining part of this paper is organized as follows. The concepts of PSO are described in Section II. The fundamentals of MOOPs are explained in Section III. The work till date being done is being discussed in Section IV extending with concepts and approaches of MOPSO. Finally, future scope and conclusion are drawn in Section V and VI respectively.

II. PARTICLE SWARM OPTIMZATION

James Kennedy and Russell Eberhart [1] introduced the term particle swarm optimization in 1995. It is based on swarm algorithms which depicts the social behavior of organism such as birds and fishes. PSO can be defined as the extension of social behavior of animals which follows the principle of population based meta-heuristic technique for optimization of problems. It includes the acceleration by distance being covered while velocity is being matched from nearest neighbor, hence belongs to technique known as nearest neighbor principle. Originally, PSO is been utilized for weight balancing in neural networks [2]. The basic characteristics of PSO are being given below:

Sociality [3]: Social interaction is the basic for human intelligence as they learn from their past experiences which make them to adapt the environment and also provide the attitude to behave in social environment.

Other concepts of PSO indicate that due to mutual learning individuals behave in similar manner which later being called as Culture. There are certain principles which every swarm group follows to study their intelligence behavior. Thus, fundamental principles are stated below [1, 3, 5, 56]:

(1) Proximity: Population must have time computation and simple space.

(2) Diversity: For the avoidance of narrow channels population must have diversity.

(3) Stability: Change in environment should not affect population.

(4) Quality: Population should consider quality factors in the environment.

(5) Adaptability: Population should be ease in changing with the computation.

In PSO, the usage of swarm is being slightly different from the evolutionary algorithms, because it follows cooperative behavior rather than competitive behavior. PSO use adaptable velocity vector, which monitors as well as changes the position of the particles after iteration in algorithm. It exploits information springing from their past experience which makes them to move towards the safe regions of search space [9]. To remember the particles past experience the separate region is present in search space which stores their best position being obtained in the search space. In single-objective optimization problem, PSO can be described in more formal manner is being given below:

Let f : S→R be defined objective function, where S is the search space of d-dimension and n is the number of particles, and where S is given as $S = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$. Hence, the i^{th} particle can be represented as $X_i = (x_{i.1}, x_{i.2}, x_{i.3} \ldots x_{i.d}) \in$ S and the past position obtained by swarm in search space is $pbest_i = (pbest_{i.1}, pbest_{i.2}, pbest_{i.3} \ldots pbest_{i.d}) \in$ S. The velocity of the i^{th} particle is given by $V_i = (v_{i.1}, v_{i.2}, v_{i.3} \ldots v_{i.d})$. Thus, the particle movement for $(t+1)^{th}$ iteration is computed as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (1)$$

$$V_i(t+1) = V_i(t) + c_1 r_{i,1}(t) \times (pbest_i(t) - X_i(t)) + c_2 r_{i,2}(t) \times (gbest(t) - Xi(t)) \qquad (2)$$

where i = 1, 2,3,…..,n. the position of the ith particle and the velocity at the i^{th} iteration is denoted with X_i (t) and V_i (t) respectively. The best position achieved by entire swarm and particle itself at ith iteration is given by gbest (t) and pbest (t) respectively. There are two acceleration coefficient c_1 and c_2 which define cognitive and social parameter respectively, while $r_{i,1}$ and $r_{i,2}$ are random values within range of [0,1]. Velocity of the particles can be updated with the three components [10]:

(1) V_i: makes particle to continue in same direction.

(2) Pbest: makes particle to move to personnel best position found in past, which is being computed using random weight $c_1 r_{i,1}$.

(3) Gbest: makes particle to move to global best position found by any particle of swarm, which is being computed using random weight $c_2 r_{i,2}$.

The overall procedure of PSO is given in Table I:

TABLE I. PROCEDURE OF PSO [67]

```
begin
for each particle of the swarm
Initialize particles position and velocity randomly
end for
do
for each particle of the swarm
Evaluate the fitness function
if the objective fitness value is better than the personal
best objective fitness value ($pbest$) in history
Current fitness value set as the new personal best ($pbest$)
end if
end for
From all the particles or neighborhood, choose the particle
with best fitness value as the $gbest$ or $lbest$
for each particle of the swarm
Update the particle velocity according to Eq. 2
Update the particle position according to Eq. 1
end for
until stopping criteria is not satisfied
end begin
```

A. Particle Swarm Topologies

The components being described above shows that the PSO performance is influenced by the personnel best position (pbest) and global best position (gbest). Thus, the best position is being depended on the information been exchanged among the neighbor particles. The particles can be connected to each other in any of the way hence some most common neighborhood topologies are being described below [3]: Figure.1 shows all the topologies.

Empty Graph: Every particle is connected to itself and compared with its own current value, which is found as pbest. [8] Thus, Eq.2 is computed using $c_2 = 0$.

Local best: In this, the k-immediate neighbor's performance affects every particle, and also particle being affected by its own best (pbest). In this case, Eq.2 has leader = lbest. [8] If k=2, then the structure obtained is ring topology.

Fully connected Graph: It describe the topology which provides the interconnection of particles with one another. Every particle makes use of its own best (pbest) and also the best position of the swarm (gbest). The structure being obtained is star topology [11]. Hence in Eq.2 calculation is done as leader = gbest.

Star Network: In this type of topology, one particle is known as focal particle which is being connect to the all the particle remaining in the swarm. In PSO, this is being called as wheel topology. [8] In this case Eq.2 is calculated using leader = focal.

Tree Network: In this topology, all particles are being arranged in form of tree i.e. each node has only one particle. In PSO, this is being called as hierarchical topology. [61] In this case Eq.2 is calculated using leader = $\text{pbest}_{\text{parent}}$.

Ring topology and wheel topology are the commonly defined topologies. Kennedy [12] described that fully connected graph makes fast convergence but being trapped in local minima.

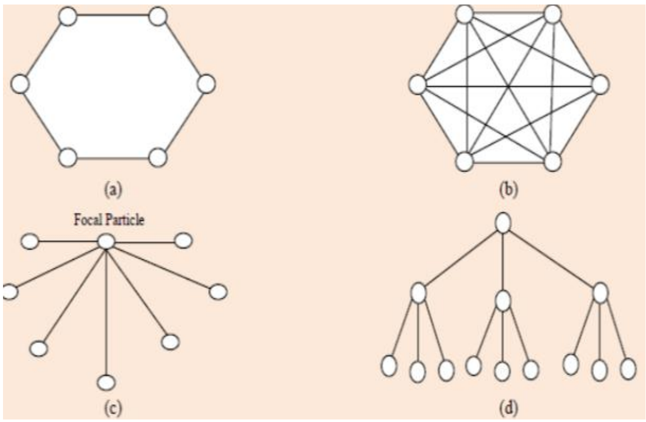

Fig.1 Topologies of swarm: (a) Ring topology (b) fully connected topology (c) star network topology (d) tree network topology [67]

B. *Selection of parameters for Particle Swarm Optimization*

During the implementation of PSO many of the consideration is being made to avoid swarm exploitation and convergence. The consideration includes such as selection of acceleration constant, limitation of maximum velocity and inertia constant or constriction factor. [4]

(a) Selection of maximum velocity: The velocity of the particle is the stochastic variable. Thus, it makes an uncontrolled trajectory to go beyond the wider cycles of the problem. [3, 13] Thus the limits of the velocity are being defined to avoid this problem as given: [3]

$$\text{If } V_{id} > V_{max} \text{ then } V_{id} = V_{max}$$

$$\text{Else if } V_{id} < -\,V_{max} \text{ then } V_{id} = -\,V_{max}$$

If considering velocity as V_{max}, which is very large and by go beyond the solution space. While, if velocity $-V_{max}$, which is very small and limit the movement of the particle. In both of the condition optimal results is not obtained. H. Fan [14] proposed the formulation to calculate maximum velocity which remains uniform throughout the search space. It is being depicted in Eq.3

$$V_{max} = (X_{max} - X_{min}) / N \qquad (3)$$

X_{max} and X_{min} are the maximum and minimum values of the position obtained by the particles and N is the number of intervals being defined by the user in k-dimensional search space.

(b) Selection of acceleration constants: Acceleration of pbest and gbest are monitored through the acceleration constant c1 and c2 respectively. The values of these constant should not be very large or small because it may either diverge the particles or limit their movement respectively. Ozcan and Mohan [15] performed several experiments to proposed the effects of deterministic acceleration constant as c = c1 + c2. The author found that particle's trajectory goes to infinity if c > 4. So, the highest value of acceleration constant will be limited using Vmax. Ozcan and Mohan [] also suggested that for better results the value of acceleration constant should be c1 = c2 = 2. Note that value of c1 and c2 alter according to the problem characteristics.

(c) Selection of inertia constant and constricting factor: According to several literatures, the acceleration constant and maximum velocity is being defined properly still there is a chance for particle to go to infinity, thus this is known as exploitation of the swarm. Therefore, two mechanism is being proposed in the literature to control the exploitation, they are inertia constant [16, 17] and constriction factor [18, 58].

Inertia constant: Inertia (w) is being multiplied to the velocity V_i (t) of the previous time t. Thus in Eq.2, velocity can be computed again as [3]:

$$V_i\,(t+1) = w\,(t) \text{ x } V_i\,(t) + c_1\, r_{i,\,1}(t) \text{ x } (\text{pbest}_i\,(t) - X_i\,(t)) + c_2\, r_{i,2}(t) \text{ x } (\text{gbest}\,(t) - Xi\,(t)) \qquad (4)$$

The value of inertia constant can be static or be dynamic []. Dynamically inertia value can be computed as defined below:

$$w\,(t) = (T - t) \text{ x } (w_s - w_e) / T \qquad (5)$$

where T is maximum number of time required to search, w_s is weight of starting inertia and w_e is weight of ending inertia. As learnt from literature w_s can be 0.9 as it allows finding the global optimum quickly. The shifting from exploration to exploitation nature reaches until w_e decreased up to 0.4. [16,17]

Constriction factor: Clerc and Kennedy [58] proposed a constriction factor to control the exploitation of the swarm. If particles being present in multi-dimensional search space then velocity can be calculated as [3]:

$$V_i\,(t+1) = \chi\ \{V_i\,(t) + c_1\, r_{i,\,1}(t) \text{ x } (\text{pbest}_i\,(t) - X_i\,(t)) + c_2\, r_{i,2}(t) \text{ x } (\text{gbest}\,(t) - Xi\,(t))\} \qquad (6)$$

.where

$$\chi \;=\; 2k / |\, 2- \varphi - \sqrt{\varphi^2 - 4\varphi}| \qquad (7)$$

as $\varphi = c_1 + c_2$ and $\varphi > 4$. If k=1 and $\varphi = 4.1$[20] then value of constriction factor χ is computed as 0.729 and thus being multiplied to previous velocity.

III. Multi Objective optimization problem

In the real world various problems are available that have more than two conflicting objectives. These problems are known as multi-objective problems. The solution to these problems is to find best tradeoff condition among the conflicting objectives [21]. Multi-objective optimization problem (MOOP) can be simply defined as the minimization or maximization of the multiple objective functions simultaneously. MOOP is also known as vector optimization, multi-criteria optimization or pareto-optimization. [65] Mathematically, MOOP can be defined as [23]:

$$\text{Minimize or Maximize } f_n\,(x) \qquad (8)$$

where n = 1,2,3,…..,N being subjected to :

$$\left.\begin{array}{l} g_i\,(x) \geq 0 \text{ where } i = 1,2,3,\ldots,m \\ h_k\,(x) = 0 \text{ where } k = 1,2,3,\ldots..p \\ x_i^{(L)} \leq x_i \leq x_i^{(U)} \text{ where } i = 1,2,3,\ldots..,q \end{array}\right\} \qquad (9)$$

Here, x is being defined as the vector of n decision variables which can be represented as:

$$x = [x_1, x_2, x_3 \ldots x_n]^T \qquad (10)$$

where, T is the transposition of column vector into row vector. Thus decision variable are numeric values that being chosen for optimization. In MOOP, constraints are imposed in-order to apply restriction over the resources and environment. They are dependent on the parameters and decision variables being involved in problem. The set of constraints, $x_i^{(L)} \leq x_i \leq x_i^{(U)}$, impose restriction over variable x_i to take value within the lower bound and upper bound. These bounds are known as decision space. g_i (x) and h_k (x) are defined as inequality and equality constraints respectively. The solution x which does not satisfy all the constraints as well as bounds of the problem is called infeasible solution. On the other hand, the solution x which satisfy all the constraints as well as bounds of the problem is called feasible solution. Set of all feasible solution are called as search space and being denoted as S.

To evaluate the solution it is being important to have certain criteria. The criteria has been used for computational function of the decision variable is known as objective function [22]. Thus, N- objective function can be represented as:

$$f(x) = [f_1(x), f_2(x), f_3(x), \ldots, f_n(x)]^T \qquad (11)$$

The multi-objective optimization includes multi-dimensional space is known as objective space. Euclid describes two spaces which is being considered in MOOP are decision variable and objective space. For every decision variable, there exists a point in objective space which is denoted by O and being defined as:

$$f(x) = [f_1(x), f_2(x), f_3(x), \ldots, f_n(x)]^T = O = [O_1, O_2, O_3, \ldots, O_n]^T \qquad (12)$$

The mapping of n-decision variable and N-objective space is being shown in figure. 2

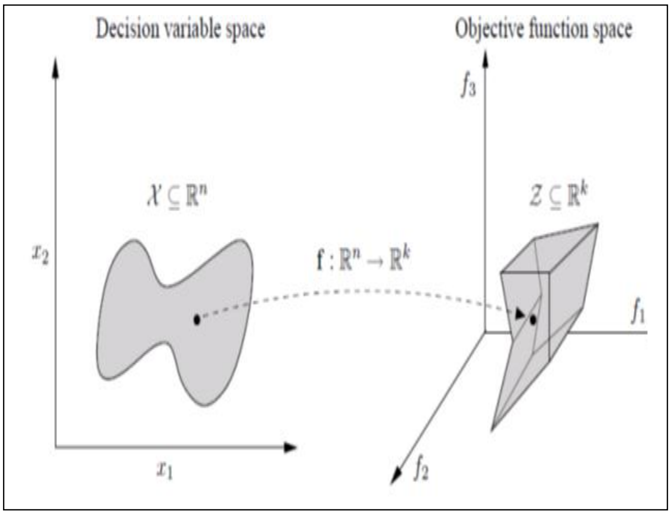
Decision variable space
Objective function space
f_3
$X \subseteq \mathbb{R}^n$
$Z \subseteq \mathbb{R}^k$
x_2
$f : \mathbb{R}^n \rightarrow \mathbb{R}^k$
f_1
x_1
f_2

Fig.2 Mapping of decision varible and objective space [66]

MOOP satisfies multiple objective functions, which does not produce a unique solution, but the set of solutions. Pareto-optimality theory [25] is being applied to find the set of solutions. The objective function can be either continuous or discrete and be linear or non-linear. The decision variable can only be either discrete or continuous.

Linear and Non-linear MOOPs: MOOPs are classified as linear and non-linear on the basis of the objective function and the constraints being used. Therefore, if all the objective function and constraints of the problem defined are linear then the MOOP is known as Multi-objective linear program while if anyone of the objective function or constraints is found to be non-linear, then MOOP is called as Non-linear Multi-objective program[24]. Likewise MOOP can also be convex and non-convex in nature. They can be defined as follows [24]:

Convex function: A function f: $R^n \rightarrow R$ can be termed as convex for any two pair of solution $x_1, x_2 \in R^n$ if the following condition is true:

$$F(\alpha x_1 + (1 - \alpha) x_2) \leq \alpha F(x_1) + (1 - \alpha) F(x_2); \forall\, 0 \leq \alpha \leq 1$$

Thus, MOOP can be convex if all its functions are convex as well as feasible region is also convex in nature.

A. Ideal Objective vector and Concept of Dominance

For obtaining the solution from the search space for MOOP, the concept of dominance is being used. The multi-objective algorithm is being bounded by two objective vectors such as [24,67]:

(1) Nadir objective vector: Nadir objective vector is always considered approximately because the whole set of pareto-optimal is unknown.

(2) Ideal objective vector: Ideal objective vector is being defined through individual optimal objective values. Mathematically it is denoted as [24, 67]:

Let $x_i^o = [x_{i.1}^o, x_{i.2}^o \ldots x_{i.n}^o]^T$ can be defined as vector of variables that minimizes or maximizes the i^{th} conflicting objective f_1 (x) where $x_i^o \in S$ can be represented as:

$$\left.\begin{array}{l}\text{Minimize/maximize } f_n(x) \\ \text{Subject to } x \in S\end{array}\right\} \qquad (13)$$

Hence, it describes that $f_i(x_i^o) = \text{optimim}_{x \in S} f_i(x)$ and ideal objective function is denoted as O^* and being given as follows:

$$O^* = f^* = [f_1^o, f_2^o \ldots f_n^o]^T \qquad (14)$$

Every MOOP does not have same solution as well as ideal vector. In general, ideal vector is non-existing solution because it is only possible if and only if all objectives are non-conflicting and have same values to MOOP. Therefore, it is only being used for reference.

Concept of Dominance: Multi-objective optimization algorithms uses dominance concept. As MOOP produces set of solution for the objective functions that are conflicting in nature. If solution x_i is better than x_j solution, then x_i is said to dominate x_j. Dominance can be defined as [24]:

Solution x_i is said to be dominating to solution x_j if and only if:

(1) x_i is better than x_j, ($f_k(x_i) > f_k(x_j)$; for all k = 1,2,3… N

(2) x_i is strictly better than x_j, ($f_k(x_i) \geq f_k(x_j)$; for all k = 1,2,3… N

Dominance relation follows certain properties such as non-reflexive, non-symmetric, non-anti-symmetric and transitive.

B. *Pareto Optimality*

In multi-objective optimization problems, the set of solution are obtained and these solutions are been compared to get non-dominant set of solutions from the given set of feasible solution. Non-dominant set can be defined as follows [21]:

Vector decision variable x ϵ S C R^n is termed as non-dominant with respect to S, if there does not exists another x' ϵ S such as f(x') < f(x).

On the contrary, we can also say that if P^* C S is set of non-dominating solutions i.e. there does not exist any another solution to dominate it, is called as pareto-optimal set. Formally, it is being defined as [60]:

Pareto optimal: vector decision variable x^* ϵ F C R^n is called as pareto-optimal if it is non-dominant in nature with respect to feasible region F.

Pareto optimal set: Pareto optimal set P^* can be stated as

$$P^* = \{x \epsilon F \mid x \text{ is pareto-optimal}\}$$

Pareto optimal solutions can also be known as non-inferior, pareto-efficient, or pareto-admissible solutions. Therefore, the set of pareto-optimal outcomes are termed as pareto-front. It is being defined as [60]:

For given MOOP, let P^* be pareto-optimal set and f(x) be objective function, then pareto-front is represented as:

$$PF^* = \{f(x) \epsilon R^k \mid x \epsilon P^*\}$$

Figure.3 represents the pareto-optimal solutions and pareto-curve.

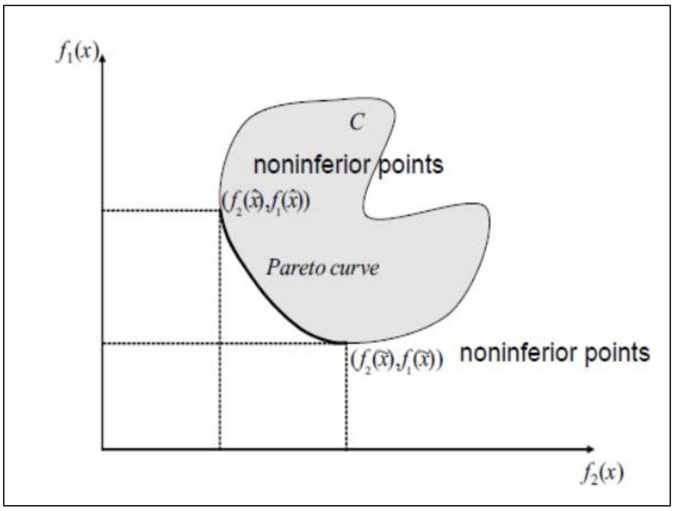

Fig.3 Paretooptimal solutions and Pareto front [66]

Pareto optimal set has two variants for MOOP, namely locally pareto-optimal set and globally pareto-optimal set. Both are defined below [22]:

Locally pareto-optimal set: For the given set P, if there exists no solution y to dominate other member x of the set, where x ϵ P have a neighbor such that $\| y - x \| \leq \varepsilon$ and $\varepsilon > 0$ is positive value, hence the solution belong to set P being included in locally pareto-optimal set.

Globally pareto-optimal set: for the given function f: S̲ C $R^n \rightarrow R^k$, S ≠ ϕ, k ≥ 2, for all x ϵ S then pareto-front $PF^* \triangleq f(x_i^*) >$ (-∞,..., ∞) is known as global minimum if and only if:

$$\forall x \in S : f(x_i^*) \leq f(x)$$

where, $x_i^* = 1,2,\ldots, k$ is called as globally pareto-optimal set, where f is multiple objective function and S is the set of feasible regions.

The above defined definitions describes that Pareto front PF^* is being formed by non-dominating vectors. In general, many solutions are calculated to obtain objective function f(x). Then corresponding to them non-dominant solution are produced to obtain pareto-front.

Pareto optimality can be classified as weak pareto-optimality and strict pareto-optimality. A point $x^* \in S$ is called weak pareto-optimality if there exist no x ∈ S such that $f_i(x) < f_i(x^*)$ ∀ i=1, 2, 3…..N []. On the other hand, a point $x^* \in S$ is called strict pareto-optimality if there exist no x ∈ S, $x \neq x^*$ such that $f_i(x) < f_i(x^*)$ ∀ i=1, 2, 3…..N [22].

There are two essential conditions being proposed by Fritz-John and Krush-Kuhn-Tucker for pareto-optimality, namely the Fritz-John necessary condition and Krush-Kuhn-Tucker sufficient condition. These conditions are being defined below assuming the objective function and constraints as being described in Eq. 8 [24]:

Fritz-John Necessary condition for Pareto optimality: The necessary condition for x^* to be pareto-optimal is that there exist vectors λ ≥ 0 and u ≥ 0; where $\lambda \in R^L$, $u \in R^J$ and λ, u ≠ 0; such that if following conditions are true:

$$\sum_{l=1}^{L} \lambda l \nabla f l(x) - \sum_{j=1}^{J} uj \nabla gj(x) = 0 \text{ and}$$

$$U_j g_j(x) = 0\ \forall\ j = 1,2,3 \ldots J$$

Krush-Kuhn-Tucker sufficient condition for Pareto optimality: The sufficient condition for x^* to be pareto-optimal is that there exist vectors λ >0 and u ≥ 0; where $\lambda \in R^L$, $u \in R^J$ and λ, u ≠ 0; such that if following conditions are true:

$$\sum_{l=1}^{L} \lambda l \nabla f i(x) - \sum_{j=1}^{J} uj \nabla gj(x) = 0 \text{ and}$$

$$U_j g_j(x) = 0\ \forall\ j = 1,2,3 \ldots J$$

IV. MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION PROBLEM

Multi-objective optimization problems have pareto-optimal sets; therefore it is being required to modify the original PSO to solve MOOP. Eckart Zitler [27] defines three general goals to archive:

(1) Maximum number of solutions in Pareto optimal set.

(2) Minimum distance between true pareto-front and pareto-front produced by an algorithm.

(3) Maximum diversity in the solution set being obtained.

There are two methods by which we can find non-dominant set; firstly, make PSO to make many runs, each run of PSO produces single solution. Therefore, after all runs of PSO, solution set is produced. Secondly, PSO is the population based algorithm, thus its run produces non-dominant solutions. The fundamental issues which are considered for designing of PSO for MOOP are discussed below [28]:

(1) Strategy to choose leaders for non-dominant solutions to give preference over dominant solution.
(2) Strategy to maintain non-dominant set of solution in the process of searching with all previous as well as current population.
(3) Strategy to retain diversity in solution set to avoid convergence to a single solution.

PSO follows two basic approaches for MOOP [29]:

Approach1: In this approach, algorithm considers every objective function separately. One function at a time is computed for each particle. This method aims to communicate modified from each objective function to guide towards a pareto-optimal front.

Approach 2: for each particle, all objective functions are computed which consider the pareto-optimality concept. This method produces non dominant set known as leaders.

Leader is the one who guides particle towards the true pareto-optimal front. Storing their best position as Pareto may result in exceeding the size of swarm. Therefore, the solutions are being stored in an external archive during search of non-dominant solution. External archive is maintained by the replacement of old solution by the new solution. Pseudo Code for the multi-objective particle swarm optimization is given in Table II:

The defined code includes variation which used to optimize MOOP. Firstly the velocity and position of the particle are initialized randomly. Then leader set is initialized with non-dominant particles. Leader is selected for each particle to update the velocity as well as position. Some mechanisms are applied to select leader from the archive. Then the particle in swarm updates pbest if the previous value is not better than the current one. This process is being carried out for fixed number of iteration. For iteration, leader in external archive is being updated and recomputed the quality of the leader.

TABLE II. PSEUDO CODE OF MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION [67]

```
begin
  for each particle of the swarm
    Initialize particles position and velocity randomly
  end for
  Initialized external archive (initially empty)
  Quality (leader)
  do
    for each particle of the swarm
      select a particle (leader) from external archive
      Evaluate the fitness function
      if the fitness value of the objective is the better than the
      best fitness value of the objective (pbest) in history
      then
        Current fitness value of the objective function is set
        as the new pbest
      end if
      update the velocity of the particle according to the
      equation-2
      update the position of the particle according to the
      equation-1
    end for
    update leader in external archive
    Quality (leader)
  until stopping criteria is not satisfied
  Report the results of external archive
end begin
```

Gregorio Toscano Pulido [30] presented some of the issues in algorithm for PSO in multi-objective scenario. They are selection and updation of leader and creation of new solution [30, 64]:

(1) Selection and updating of the leader: How selection of leader is being made from non-dominant solution set, what must be the criteria for selection of leader, what strategy must be planned for the selection of particle which remain in external archive.

(2) Creation of new solution: what must be the criteria for diversity promotion so as to create solution as position updation and mutation operator?

The above described issues indicate that the selection of leader is very important concept in multi-objective particle swarm optimization (MOPSO) design. All non-dominant solution can become leaders, so any one from them is elected. The importance of leader is being evaluated by the criteria of quality. In several literatures, quality measures being defined in many ways. The common way being proposed is density based leader selection. Several authors proposed basically two important density based measures such as, nearest neighbor density estimator [57] and kernel density estimator [31].

(1) Nearest neighbor density estimator: In this mechanism, the density of the particle is being calculated by the closet neighbor in the objective function space.

(2) Kernel density estimator: In this mechanism, parameter called as σ_{share} is defined, which is the radius of the neighborhood of the particle. The neighborhood being defined for each particle is called as niche. Hence, fewer numbers of particles are preferred.

In MOPSO, the most complicated task is to update external archive. MOPSO use three types of archives. Firstly to store global best solutions, second to store personal best solutions and third to store local best if required in problem being defined [60]. The solution obtained will be kept in external archive if the solution is non-dominant from all the member of the archive. If the solution fails to follow the above defined condition, then it is usually being removed. To avoid the complexity in updation of archive, bound archive size is used [28]. Suppose the solution obtained is non-dominant by the member of archive but the archive is full, then diversity is important factor for the insertion of solution into the external archive. Thus, archive is being updated to retain the maximum diversity of the archive.

A. MOPSO Related work

There are several proposals that have been used recently to implement PSO for multi-objective. Here, all have been reviewed in the paper [39] and some are discusses below:

(1) Moore and Chapman Algorithm [64]: It is based on Pareto dominance. The author focused over the importance of performance of both individual as well as group search which includes cognitive component as well as social component. In this method, the pbest of the each particle is the list of the non-dominated solutions which is being found in its trajectory. While the pbest is selected for the particle, it is being randomly selected.

Since the author used the ring topology, in this when the best particle is being chosen of the neighborhood the solutions present in pbest lists are compared and the non-dominant solution of the neighborhood is chosen. However, the authors haven't adopted any mechanism for diversity maintenance. They also haven't provided description about the selection of lbest of the particle when more than one non-dominating solution is obtained in the neighborhood.

(2) Ray and Liew proposal swarm metaphor [38]: This algorithm is also based on the Pareto dominance but it also combines particle swarm with evolutionary approach. They use crowding method to maintain diversity and multi-level sieve to handle constraints. In this the same constraints and objective functions are used which they had used in their previous paper [38].

The approach also uses nearest neighbor density estimator for the promotion of the diversity. For this, they adopted roulette selection scheme of leaders on the problem values. Therefore, the set of leaders computed by the authors is being grouped as external archive.

(3) Parsopoulas and Vrahatis Algorithm [32]: In this algorithm author uses the aggregating approach to handle multi-objective function using particle swarm. The uses basically three types of aggregating function such as:

(1) Conventional linear aggregating function: In this function uses weights whose values are fixed during the run.
(2) Dynamic aggregating function: In this function, the weights used are changed gradually during the run.
(3) Bang-bang weighted aggregation function: In this function, the weights used are changed abruptly during the run.

In the all cases, the author uses the fully connected topology and in that way concave portion of the Pareto front is being generated.

The author studied about the parallel version of the Vector Evaluated Particle Swarm (VEPSO) method [55] for optimization of multi objectives. VEPSO is the one of the variant of PSO being developed on the inspiration of Vector Evaluated Genetic algorithm (VEGA) [56, 57]. In method, only one objective function of the problem for the swarm is computed under consideration and this information is being exchanged to the other swarms using their best experiences as gbest. The author defines that this approach leads to pareto-optimal solutions.

(4) Hu and Eberhart proposed Dynamic Neighborhood PSO [69]: In this algorithm, single objective is optimized at a time using simple scheme similar to lexicographic ordering [34]. It tends to be beneficial when there are only few function say two or three and it may also be sensitive towards ordering of objective function.

Hu [69]: The approach proposed is the extension of dynamic neighborhood PSO. As author uses the secondary population for solving the problem but this approach fails to produce true pareto-front for some multi-objective problem. The author makes their results to compared with strength Pareto evolutionary algorithm [37] using the set coverage metric [36]

(5) Fieldsend and Singh Approach [35]: They proposed an approach in which author used unconstrained elite archive where the special data structure is adopted called as "dominated tree" to store non-dominant individual being found during search process. The archive makes an interaction for the selection of leaders with the primary population. The selection criteria for the selection of gbest of the particle are being defined by the dominated tree.

Firstly, a composite point is being located on the tree on the basis of dominance relations and then the member closet to the point in the objective function space is declared as leader. While on the other hand, the personal best set is being maintained for each member and the selection is uniformly performed. The approach also uses turbulence operator that acts over the velocity value used by PSO. But this approach faces the multi frontal problems.

(6) Mostaghim and Teich [62]: The author proposed sigma method in which best local guides is being adopted by each particle to improvise convergence and diversity of PSO. This approach is being found similar to compromise programming [13]. For the selection of the leader the sigma value is being assigned to the each particle of the external archive. Particle in the swarm selects its leader to the particle whose value is close to the sigma value of the particle from external archive. They also uses turbulence operator which is being applied over decision variable space. The usage of sigma values increases the pressure of selection in PSO which causes the premature convergence. This method is being applied to the molecular force field parameterization problem [39].

In their further work, Mostaghim and Teich [65] discussed about the influence of ϵ-dominance on the multi-objective problems. ϵ-dominance is being compared with techniques existing for fixation of the external archive and the solution is being compared in terms of time of computation, convergence and diversity. The results obtained by the author describes that ϵ-dominance mechanism can find the solution faster than the clustering techniques with comparable convergence and diversity.

The author proposed a new density measure is inspired from their previous work [62]. Based on the idea that initial external archive from which the particles make the selection of leader has effect on the diversity of solution, therefore the author propose the new mechanism which provide successive improvements in external archive of the solution.

Mostaghim and Teich [58] proposed another method called as coveringMOPSO (cvMOPSO). It works on the above discussed idea. It works in two phases. In phase 1, MOPSO algorithm executes with the restricted size of external archive to find good approximation of the Pareto front. In the phase 2, the solution computed in phase 1 act as an input to the external archive to cvMOPSO. The particles in cvMOPSO are divided into sub-swarms around each non-dominated solution after the first generation. The task of these sub-swarms is to cover the gaps being present in the non-dominated solution obtained from the phase 1. In phase 2, no restrictions are imposed on the size of the external archive.

(7) Li [40]: The author proposed on approach where PSO is being combined with main method of NSGA II [40]. It uses the fully connected topology. In this method, once the position of the particle updated rather than being comparing the new position only against the pbest

position of the particle. All pbest positions of the swarm and all the new positions obtained recently combined in one set say as 2N where N is the number of the swarms. The method selects the best solution among them to conform the next swarm this is being performed by the means of non-dominated sorting. The author hasn't specified the values that are assigned to the velocity of pbest positions to consider them as particles. The approach selects leaders randomly from set of leader being stored in external archive from the best of them on the basis of the two mechanisms, namely niche count and nearest neighbor density estimator. The author uses a mutation operator which is applied at every step of iteration to the particle with either smallest density estimator value or the largest niche count value.

In further work, the author proposed the maximinPSO, which uses fitness function which is being derived from the maximin strategy defined by Balling [68] to determine the Pareto dominance. The author shows advantage to this approach that it doesn't need additional clustering or niching techniques. Since the maximin function only is incapable to define that solution is dominant or non-dominant but if it is clustered with other solution i.e. the method also provide diversity information. In this mechanism, each particle, different leader is selected for each of the decision variables to conform a single global best. Leaders are randomly selected on the basis of maximin fitness from the external archive.

B. *Approaches of MOPSO*

Several MOPSO approaches been proposed in many literature. Therefore, in this section some of the common approaches are explained below:

(1) Weighted objective function aggregation approach

(2) Lexicographic ordering approach

(3) Pareto based approach

(4) Combined approach

Weighted objective function aggregation approach: In MOOP, the approach proposed by Petropoulos and Vrahatis [32]. The author uses a weighted sum of the objective, being shown in given Equation:

$$f\,(x) = \sum_{i=1}^{k} wifi(x) \qquad (15)$$

where i = 1, 2, … k and w_i is non-negative weight which can be represented as:

$$\sum_{i=1}^{k} wi = 1 \qquad (16)$$

The function weight can either be fixed or dynamic during the process of optimization. In case of fixed weights conventional weighted aggregation is used. In this method, for each run of optimization pareto-optimal solution is obtained. Therefore, to choose weight according to objective function it requires pre-known knowledge about search space [33].

Most of the time during process of optimization set of pareto-optimal solutions has to be obtained. Thus, this method is not efficient because of heavy computation of optimization and also because of inability to find solution in concave regions [34]. To overcome the defined problem, two methods have been proposed namely, Bang-Bang weighted aggregation (BWA) and Dynamic weighted aggregation (DWA).

In BWA, weights of bi-objective function are modified during optimization process as shown in Equation given below:

$$w_1\,(t) = \text{sign}\,(\sin\,(2\Pi t\,/\,f_w)) \qquad (17)$$

$$w_2\,(t) = 1 - w_1\,(t) \qquad (18)$$

where, f_w and t are weight change frequency and iteration index respectively. The sign function heretically changes the weight of the objective function. On the other hand, DWA provides alternative weight modification method. It allows changing of weight in slow manner. It tries to move closer to true pareto-optimal front. The weight is being modified according to the equations shown below:

$$w_1(t) = |\sin\,(2\Pi t\,/\,f_w)| \qquad (19)$$

$$w_2\,(t) = 1 - w_1\,(t) \qquad (20)$$

However, the performance of BWA and DWA is equal for concave front but for convex pareto-front BWA is better [].

Lexicographic ordering approach: Lexicographic ordering method is based on ranking system. Each objective function is being ranked by the user according to their importance. Then their optimization begins in serial manner. Suppose $f_i(x)$ be i^{th} ranked function where i = 1, 2 … n which indicates the rank of the objective functions. $f_1(x)$ and $f_n(x)$ defines the most important and least important objective function respectively [34]. In optimization process, firstly function $f_1(x)$ is formulated either to minimizes or maximizes the function and solution x_1^* is obtained. These procedures are continued till last objective function. Hence problem can be formulated as [Carlos A. Coello Coello 1999]:

$$\text{Minimize } f_i(x) \qquad (21)$$

$$\text{Subject to } g_j\,(x) \le 0;\ j = 1, 2 \ldots m \qquad (22)$$

$$f_k\,(x) = f_k^*\ \ k = 1, 2 \ldots i\text{-}1 \qquad (23)$$

x_n^* is the solution to last objective function and that is the desired solution to the defined problem.

Pareto based approaches: Pareto based approaches are based on pareto-dominance i.e. it uses the concept of leader selection. The leader guides the swarm during search. As we have learned in above section that all non-dominant solution are equally good. Thus, several techniques have being proposed for leader selection in literature. Some addition criteria can also be added so as to avoid random search and fast convergence and pareto-front spread, swarm diversity and exchange of information being given by density estimator. Some of the proposed methods have been categorized in Table III for leader selection on pareto-based approach.

Other approaches also proposed some methods with selection of leader, external archive and neighborhood topologies are discussed in Table IV:

TABLE III. LEADER SELECTION TECHNIQUES IN PARETO BASED APPROACHES [67]

S.N.	***Leader Selection Techniques***	***External Archive***	***Proposed by*** *(used fully connected neighborhood topology)*
1.	Dominance	Yes No Yes	Fieldsend and Singh [35] Srinivasan and Hou [36] Alvarez-Benitez et al. [37]
2.	Density Estimator	Yes Yes Yes Yes Yes Yes	Ray and Liew[38] Coello et al. [55] Bartz et al. [39] Li [40] Reyes and Coello [41] Raquel and Naval [42]
3.	Randomly	No Yes	Toscano and Coello[43] Janson and Merkle [44]
4.	Niche Count	Yes Yes Yes	Srinivasan and Hou [36] Li [40] Salazar-Lechuga and Rowe [45]
5.	Sigma Value	Yes	Mostaghim and Teich [46, 47, 48]
6.	Fuzzy Membership	Yes	Zhao and Cao [49]
7.	Stripes	Yes	Vallalabos-Arias et al. [50]

TABLE IV. LEADER SELECTION TECHNIQUES IN OTHER APPROACHES [67]

S.N.	***Leader Selection Techniques***	***External Archive***	***Proposed by*** *(used fully connected neighbourhood topology)*
1.	Single Objective	No	Mahfouf et al. [51]
2.	Energy value	Yes	Xiao-hua et al. [52]
3.	Maximum fitness	Yes	Li [53]
4.	Composite leader	No	Zhang et al. [54]

V. Future Research Scope

It can be observed that during last decade, this area is highly developed but still provide large scope to find direction of research. Therefore, some of the scope for future being discussed here:

(1) In weighted objective function aggregation approach, the function weight is dependent on the user, thus it is essential to address function weight according to problem characteristics.

(2) In lexicographic ordering, function is being ranked on their performance basis. Thus, to rank the objective function it requires higher knowledge about the problem. Therefore, some any other parameter can be found to rank function on basis of problem characteristics.

(3) In pareto-based method, it includes three steps. Firstly, leader selection, secondly, diversity promotion and thirdly, archive maintenance. Thus, these three of them form a great scope to find new parameters through which true pareto-optimal front is obtained efficiently and also need less computational.

Thus, MOPSO can also be further extended to non-pareto algorithm which includes direction for exchange of information as well as for frequency.

VI. Conclusion

The paper basically discussed about Multi-objective particle swarm optimization. It also includes the concepts and functionality of Particle swarm optimization and Multi-objective optimization problems. Finally, issues and work related to MOPSO is being discussed. It also described the methods to extend particle swarm optimization for multi-objective problem. Some common approaches of multi-objective particle swarm optimization also being described. And finally, some of the future directions for research in multi-objective particle swarm optimization are addressed briefly.

References

[1] J. Kennedy, R. C. Eberhart, “Particle Swarm Optimization,” in *Proc. of the IEEE international Conference on Neural Networks*, pp. 1942-1948, Piscataway, New Jersey, USA, 1995.

[2] R. C. Eberhart, R. Dobbins, P. K. Simpson, “Computational intelligence PC Tools,” *Morgan Kaufmann Publishers*, 1996.

[3] R. Eberhart, Y. Shi, “Particle swarm optimization: Developments, applications and resources,” in *Proc. of the IEEE Congr. Evol. Comput.*, vol. 1, pp. 81-86, 2001.

[4] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, R. G. Harley, “Particle Swarm Optimization: Basic Concepts, Varients and Applications in Power Systems,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, 2008. Article (CrossRef Link)

[5] S. Yang, M. Wang, L. Jiao, “A quantum particle swarm optimization,” in *Proc. of the IEEE Congr. Evol. Comput.*, vol. 1, pp. 320-324, 2004.

[6] M. Millonas, “Swarms, phase transitions, and collective intelligence,” in *Proc. of Santa Fe Institute Studies in the Sciences of Complexity*, pp. 417-445, 1994.

[7] A. P. Engelbrecht, “Particle swarm optimization: Where does it belong?,” in *Proc. of the IEEE Swarm Intell. Symp.*, pp. 48-54, 2006.

[8] A. P. Engelbrecht, “Computational Intelligence: An Introduction,” *John Wiley& Sons*, England, 2002.

[9] K. E. Parsopoulos, M. N. Vrahatis, “Multi-Objective Particles Swarm Optimization Approaches,” *IGI global*, Chapter 2, 2008.

[10] D. Boeringer, D. Werner, “Particle swarm optimization versus genetic algorithms for phased array synthesis,” IEEE Trans. Antennas Propagat., vol. 52, no. 3, pp: 771-779, 2004. Article (CrossRef Link)

[11] A. P. Engelbrecht, “Fundamentals of Computational Swarm Intelligence,” *John Wiley & Sons*, 2005.

[12] J. Kennedy, “Small worlds and mega-minds: Effects of neighbourhood topology on particle swarm performance,” in Proc. of IEEE Congr. Evol. Comput., vol. 3, pp. 1931-1938, Jul. 1999.

[13] X. Hu, Y. Shi, R. Eberhart, “Recent advances in particle swarm,” in *Proc. of IEEE Congr. Evol. Comput.*, vol. 1, pp. 90-97, 2004.

[14] H. Fan, Y. Shi, “Study on Vmax of particle swarm optimization,” in *Proc. of Workshop on Particle Swarm Optimization, Purdue School of Engineering and Technology*, Indianapolis, IN, USA, 2001.

[15] E. Ozcan, C. Mohan, “Particle swarm optimization: Surfing the waves,” in *Proc. of IEEE Congress Evol. Comput.*, vol. 3, pp. 1939-1944, 1999.

[16] Y. Shi, R. Eberhart, “Parameter selection in particle swarm optimization,” in *Proc. of 7th Int. Conf. Evol. Program*, pp. 591-600, 1998.

[17] Y. Shi, R. Eberhart, “Empirical study of particle swarm optimization,” in *Proc. of IEEE Congr. Evol. Comput.*, vol. 3, pp. 1945-1950, 1999.

[18] F. Bergh, A. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, 2004. Article (CrossRef Link)

[19] A. Abido, “Particle swarm optimization for multi-machine power system stabilizer design,” in *Proc. of Power Engineering Society Summer Meeting (PES)*, vol. 3, pp. 1346-1351, 2001.

[20] M. Clerc, “The swarm and the queen: Towards a determininistic and adaptive particle swarm optimization,” in *Proc. of Congress on Evolutionary Computation (CEC99)*, pp. 1951-1957, 1999.

[21] C. A. C. Coello, "An Introduction to Multi-Objective Particle Swarm Optimizers," *Soft Computing in Industrial Application, Advances in Intelligent and Soft Computing*, vol. 96, pp. 3-12, 2011. Article (CrossRef Link)

[22] C. A. C. Coello, G. B. Lamont, D. A. V. Veldhuizen, "Evolutionary Algorithms for Solving Multi-Objective Problems," *Genetic and Evolutionary Computation Series*, 2nd edition, Springer, 2007.

[23] A. Osyczka, "Multicriteria optimization for engineering design," *Design Optimization*, Academic Press, pp. 193-227, 1985. Article (CrossRef Link)

[24] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms," *WILEY*, 2002.

[25] M. Ehrgott, "Multicriteria Optimization," *Springer*, Berlin, 2005.

[26] H. T. Kung, F. Luccio, F. P. Preparata, "On Finding the maxima of a set of vectors," *Journal of the Association for Computing Machinary*, vol. 22, no. 4, pp. 469-479, 1975. Article (CrossRef Link)

[27] E. Zitzler, K. Deb, L. Thiele, "Comparison of Multi-objective Evolutionary algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp.173-195, 2000. Article (CrossRef Link)

[28] C. A. C. Coello, D. A. V. Veldhuizen, G, B. Lamont, "Evolutionary algorithms for Solving Multi-Objective Problems," *Kluwer Academic Publishers*, New York, 2002. Article (CrossRef Link)

[29] M. R. Sierra, C. A. C. Coello, "Improving PSO-based Multi-objective Optimization using crowding, mutation and ∈-Dominance," *Lecture Notes in Computer Science*, vol. 3410, pp. 505-519, 2005. Article (CrossRef Link)

[30] G. T. Pulido, "On the Use of Self-Adapteation and Elitism for Multiobjective Particle Swarm Optimization," Ph.D Thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV_IPN, Mexico, 2005.

[31] K. Deb, D, E. Goldberg, "An Investigation of niche abd species formation in genetic Function Optimization," *in Proc. of the Third International Conference on Genetic Algorithms*, pp. 42-50, San Mateo, California, USA, 1989.

[32] K. E. Parsopoulos, M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2-3, pp. 235-306, 2002. Article (CrossRef Link)

[33] Y. Jin, M. Olhofer, B. Sendhoff, "Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?," in *Proc. of GECCO-2001*, 2001.

[34] C. A. C. Coello, "A Comprehensive Survey of Evolutonary-Based Multiobjective optimization Techniques," *Knowledge and Information Systems*, pp. 269-308, 1999. Article (CrossRef Link)

[35] J. E. Fieldsend, S, Singh, "A multiobjective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence," in *Proc. of the 2002 U.K. Workshop on Computational Intelligence*, Birmingham, UK. pp. 37-44, 2002.

[36] D. Srinivasan, T. H. Seow, "Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problem," in *Proc. of Congress on Evolutionary Computation (CEC'2003)*, vol. 3, pp. 2292-2297, Canberra, Australia, 2003.

[37] J. E. Alvarez-Benitez, R. M. Everson, J. E. Fieldsend, "A MOPSO algorithm based exclusively on pareto dominance concepts," in *Proc. of 3rd International Conference on Evolutionary Multi-Criterion Optimization*, pp. 459-473, Guanajuato, Mexico, 2005. Article (CrossRef Link)

[38] T. Ray, K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering Optimization*, vol. 34, no. 2, pp. 141-153, 2002. Article (CrossRef Link)

[39] T. Bartz-Beielstein, P. Limbourg, K. E. Parsopoulos, M. N. Vrahatis, J. Mehnen, K. Schmitt, "Particle swarm optimizers for pareto optimization with enhanced archiving techniques," in *Proc. of Congress on Evolutionary Computation (CEC'2003)*, vol. 3, pp. 1780-1787, Canberra, Australia, 2003.

[40] X. Li, A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'2003)*, pp. 37-48, 2003. Article (CrossRef Link)

[41] M. R. Sierra, C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and ²-dominance," in *Proc. of 3rd International Conference on Evolutionary MultiCriterion Optimization*, pp 505–519, 2005.

[42] C. R. Raquel, Jr. P. C. Naval, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO2005)*, pp. 257-264, Washington, DC, USA, 2005.

[43] G. T. Pulido, C. A. C. Coello, "Using clustering techniques to improve the performance of a particle swarm optimizer," in Proc. of the Genetic and Evolutionary Computation Conference (GECCO'2004) , pp. 225-237, Seattle, Washington, USA, 2004. Article (CrossRef Link)

[44] S. Janson, D. Merkle, "A new multiobjective particle swarm optimization algorithm using clustering applied to automated docking," in Proc. of Hybrid Metaheuristics, Second International Workshop, pp. 128-142, Barcelona, Spain, 2005.

[45] M. Salazar-Lechuga, J. Rowe, "Particle swarm optimization and fitness sharing to solve multi-objective optimization problems," in *Proc. of Congress on Evolutionary Computation (CEC'2005)*, pp. 1204-1211, Edinburgh, Scotland, UK, 2005. Article (CrossRef Link)

[46] S. Mostaghim, J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proc. of the 2003 IEEE Swarm Intelligence Symposium*, pp. 26-33, Indianapolis, Indiana, USA, 2003. Article (CrossRef Link)

[47] S. Mostaghim, Jurgen Teich, "The role of dominance in multi objective particle swarm optimization methods," in *Proc. of Congress on Evolutionary Computation (CEC'2003)*, pp. 1764-1771, Canberra, Australia, 2003.

[48] S. Mostaghim, J. Teich, "Covering paretooptimal fronts by subswarms in multi-objective particle swarm optimization," in *Proc. of Congress on Evolutionary Computation (CEC'2004)*, pp. 1404-1411, Portland, Oregon, USA, 2004.

[49] B. Zhao, Y. Cao, "Multiple objective particle swarm optimization technique for economic load dispatch," *Journal of Zhejiang University SCIENCE*, vol. 6, no. A(5), pp.420-427, 2005.

[50] M. A. Villalobos-Arias, G. T. Pulido, C.s A. C. Coello, "A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer," in Proc. of the 2005 IEEE Swarm Intelligence Symposium, pp. 22-29, Pasadena, California, USA, 2005.

[51] M. Mahfouf, M.-Y. Chen, D. A. Linkens, "Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels," *Lecture Notes in Computer Science*, vol. 3242, pp. 762-771, 2004.

[52] X.-H. Zhang, H.-Y. Meng, L.-C. Jiao, "Intelligent particle swarm optimization in multiobjective optimization," in *Proc. of Congress on Evolutionary Computation (CEC'2005)*, pp. 714–719, Edinburgh, Scotland, UK, 2005.

[53] X. Li, "Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'2004)*, pp. 117-128, Seattle, Washington, USA, 2004. Article (CrossRef Link)

[54] L. B. Zhang, C. G. Zhou, X. H. Liu, Z. Q. Ma, Y. C. Liang, "Solving multi objective optimization problems using particle swarm optimization," in *Proc. of Congress on Evolutionary Computation (CEC'2003)*, pp. 2400-2405, Canberra, Australia, 2003.

[55] C. A. C.o Coello, M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in Proc. of Congress on Evolutionary Computation (CEC'2002), pp. 1051-1056, Piscataway, New Jersey, USA, 2002.

[56] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. of 6th Int. Symp. Micro Machine and Human Science (MHS)*, pp. 39-43, 1995. Article (CrossRef Link)

[57] K. Deb, A. Pratap, S. Agrarwal, T. Meyarivan, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no .2, pp. 182-197, 2002.

[58] M. Clerc, J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol.Comput.*, vol. 6, no. 1, pp. 58-73, 2002. Article (CrossRef Link)

[59] R. Eberhart, Y. Shi, J. Kennedy, "Swarm Intelligence," *Morgan Kaufmann*, San Mateo, CA, 2001.

[60] M. R. Seirra, C. A. C. Coello, "Multi-Objective particle Swarm optimizers: A Survey of the State-of-the Art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308, 2006.

[61] S. Janson, M. Middendorf, "A hierarchical particle Swarm Optimizer," in *Proc. of Congress on Evolutionary Computation (CEC' 2003)*, pp. 770-776, Camberra, Australia, 2003.

[62] A. A. Kadkol and G. G. Yen, "A cultural-based particle swarm optimization framework for dynamic, constrained multi-objective optimization,"*Int. J. Swarm Intell. Res.*, vol. 3, no. 1, pp. 1–29, Mar. 2012.

[63] M. Daneshyari and G. G. Yen, “Cultural based particle swarm for dynamic optimization problems,” *Int. J. Sys. Sci*., vol. 43, no. 7, pp. 1284–1304, Jul. 2012.

[64] C. K. Chow and S. Y. Yuen, “A multiobjective evolutionary algorithm that diversifies population by its density,” *IEEE Trans. Evol. Comput*.,vol. 16, no. 2, pp. 149–172, Apr. 2012.

[65] S. Z. Mart´ınez and C. A. Coello Coello, “A multiobjective particle swarm optimizer based on decomposition,” in *Proc. Genetic Evol. Comput*., 2011, pp. 69–76.

[66] V. Kumar and S.Minz, “A multiobjective particle swarm Optimization: Introduction,” in *Proc. IEEE Congr.Evol. Comput.*,Oct.2014, pp. 345-352.

[67] G. G. Yen and Z. He, “Performance Metrics Ensemble for Multiobjective Evolutionary Algorithms,” *IEEE Trans. Evol. Comput*., vol. 18, no. 1, pp. 131–144, Feb. 2014.

[68] Z. He and G. Yen, “Ranking many-objective evolutionary algorithms using performance metrics ensemble,” in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2480–2487.

[69] W. Hu and G. G. Yen, “Density estimation for selecting leaders and maintaining archive in MOSPO,” in *Proc. IEEE Congr. Evol. Comput*.,Jun. 2013, pp. 181–188.