Chapter 2

State of art

This chapter contains brief about parameter optimization of 2DOF controller. The state of art survey on evolutionary and swarm intelligence algorithms including its methodological issues from the perspective of multiobjective optimization is provided. Justification for selection of evolutionary (NSGA-II and NSGA-III) and swarm (MOPSO) algorithms of 2DOF controller parameter optimization is provided at the end.

2.1 Parameter optimization of 2DOF controller

Two degree of freedom(2DOF) structure can attain two control system objectives simultaneously i.e. set point tracking and disturbance rejections. 2DOF controller consists of two compensator $C(s)$ and $C_f(s)$. Where, $C(s)$ is called the serial (or main) compensator and $C_f(s)$ the feed forward compensator. Three parameters of $C(s)$ i.e., the proportional gain K_p , the integral time T_i , and the derivative time T_D , are referred to as "basic parameters", and two parameters of $C_f(s)$ i.e., α and β are referred to as "2DOF parameters". The disturbance response is completely determined by serial compensator $C(s)$, while set point response depends on both serial compensator $C(s)$ and feed forward compensator $C_f(s)$. So, set point response can be still adjusted by feed forward compensator $C_f(s)$ even after serial compensator $C(s)$ is fixed [3]. Considering above conditions M Araki et. al have proposed following two step tuning methods.

Step 1: Optimize disturbance response by tuning serial compensator $C(s)$ (i.e. adjust PID parameters K_p, T_i, and T_D).

Step 2: Fixed serial compensator C(s) and optimize the set point response by tuning

feed forward compensator $C_f(s)$(i.e. adjust 2DOF parameters α and β).
The problem with above method is that optimizing disturbance response only deteriorates set point response such that tuning of feed forward compensator $C_f(s)$ does not guarantee the overall optimal response. In order to avoid this problem following two different strategies have been proposed.
Strategy 1: Optimize serial compensator C(s) and feed forward compensator $C_f(s)$ by assigning weight to objective functions.
Strategy 2: Instead of assigning weights to objective functions for tuning (C(s) and $C_f(s)$) a problem of multiobjective optimization of 2DOF controller will be formed to tune simultaneously all the five parameters of 2DOF controller.
The weight based approach (strategy 1) can be selected only if we know the exact trade-off among objectives. This weight based approach results in a single optimal solution. The problem of multiobjective optimization required multiple optimal solutions, instead of single optimal solution. The multiobjective optimization algorithms required modifications in simple evolutionary or swarm based algorithm. Hence, outperformed evolutionary (NSGA-II and NSGA-III) and swarm(MOPSO) based multiobjective optimization algorithms (with strategy 2) are selected for the optimization of five parameters of 2DOF controller.

2.2 Multiobjective optimization: State of art

The problem of multiobjective optimization consists of two or more objectives that required to be optimized concurrently. The constrained may be imposed on the objective functions and objective functions are in conflict. If objective functions are not in conflict then, single solution exist for multiobjective optimization problem (MOOP). The majority of natural world MOOPs consists of set of solutions, instead of single solution. This set of solutions are in trade-offs among distinct objectives. In order to obtain these trade-off solutions, an old concept of Pareto optimality is commonly used. This concept of optimality was formerly proposed by Francis Ysidro Edgeworth [19] and later established by Vilfredo Pareto [20]. The set of solutions is called Edgeworth-Pareto optimal solutions or Pareto optimal solutions.
Evolutionary and swarm based optimization has become very prevailing research field in the past few years due to their advantage to work based on population of search instead

of single search. Also, evolutionary and swarm based algorithms are less sensitive to the pattern of Pareto front [21].

Multiobjective optimization algorithm gives number of Pareto optimal solutions (Non-dominated set of solution) in a single simulation run. Each Pareto optimal solution is significant with reference to trade-off relations among the objectives. Practically, user needs only one solution from the set of Pareto optimal solutions for a particular problem. So, the question is which solution to select from these available multiple Pareto optimal solutions? It is easy to answer the question if user has obtained set of trade-off solutions but, not so easy to answer in the absence of set of trade-off solutions. If user is aware of exact trade-off among objective functions then there is no need to obtain multiple solutions. Hence, weight based classical method is enough to find single optimal solution. This weight based classical method is known as a priori method. Generally, user is not aware of exact trade-off among objective functions. Therefore, it is preferred to first attain a set of Pareto optimal solutions and select best one using some higher level information. This method of obtaining single Pareto optimal solution is known as an ideal approach for multiobjective optimization. One can still use a priori approach of classical method to attain set of Pareto optimal solutions in following way. First of all, arbitrary weight vector is chosen and for each weight vector formulate single objective optimization problem. Thereafter, corresponding to each weight vector obtain optimum solution. Here, Pareto optimal solution depends on arbitrary weight vector selected for single objective optimization (MOOP is converted into single objective optimization problem). In the case of ideal approach for multiobjective optimization the original MOOP is not transformed into single objective optimization problem. An ideal approach for multiobjective optimization can preserve diversity along with domination. An ideal method for the solution of MOOP gives better result compared to a priori approach of solution.

2.2.1 Advances in multiobjective optimization of evolutionary algorithms

The first idea of using evolutionary algorithm for the solution of multiobjective optimization problem was found by Rosenberg [22]. Rosenberg converted multiobjective optimization problem to single objective optimization and solved using GA. Afterwards, there is hardly any trail was observed till 1983 to use an evolutionary or swarm based algorithm for the solution of multiobjective optimization problem.

David Schaffer is known to be the first to have devised Multiobjective Evolutionary Algorithm (MOEA) called Vector Evaluated Genetic Algorithm (VEGA) in 1984. GA is used to evaluate vector of objective functions instead of single objective function. Here, number of objectives M is divided in to q=M/N (N total number of population) equal subpopulation of GA at random. Individual subpopulation is assigned a fitness based on its partition. Perform proportionate selection on all q solutions to form mating pool. Crossover and mutation is performed on populations in mating pool to create new population. Each objective function is used to evaluate members in the population. This method insists on solutions which are good for individual objective function only. Schaffer allowed crossover between two good solutions in entire populations, which may find offspring that are satisfactory trade-off solution between the objectives. Mutation operator is enforced on each individual as it is. The prime benefit of VEGA is, it requires minor changes in the simple GA to convert algorithm for solution of multiobjective optimization. The limitation of VEGA is that, it is assessed only with one objective function hence; each solution is not evaluated for remaining M-1 objectives which are important from the perspective of multiobjective optimization. VEGA fails to provide diversity in the population [23], [24].

Subsequently to VEGA, investigators opted for several years other elementary techniques. The most accepted was linear and non linear aggregating function. In this case all the objective functions were added and considered single objective function, which was evaluated as fitness of an evolutionary algorithm [25], [26]. Another approach was the Lexicographic ordering approach. In this approach, first single objective is optimized without considering other objectives. Then, the second objective is optimized without affecting the characteristic of the result attained for the first objective. This mechanism is replicated for rest of the objectives [27].

David E. Goldberg was the first to introduce notion of Pareto optimality using evolutionary algorithm which is mentioned in his seminal book on genetic algorithms [28]. Here, Goldberg criticized Schaffers VEGA and recommended use of nondominated ranking and selection to move solutions towards Pareto front. The main objective was to obtain Pareto set of nondominated solutions based on ranking mechanism from the population. He also proposed use of niching mechanism to converge algorithm at single point on the Pareto front. Goldberg did not practically employ his strategy, but all the MOEAs matured were persuaded by his theory. MOEAs are classified in two categories first and second generation algorithms. First of all, we will discuss algorithms which are known as first generation which emphasis on simplicity. These algorithms are also known as non-elitist multiobjective genetic algorithms as they do not use any elite preserving operator. The second generation of multiobjective optimization algorithms that are emphasizing on efficiency and uses elitism mechanism (popularly known as elitist multiobjective genetic algorithms) are discussed.

2.2.1.1 First Generation or Non elitist multiobjective genetic algorithms

Following are the most commonly used non elitist MOEAs.

1. Multiobjective Genetic Algorithm (MOGA)

Multiobjective Genetic algorithm called (MOGA) which used nondominated classifications of a GA population was first proposed by Fonessa and Fleming [29]. Here, individual result is tested for its domination in the population. To a solution i, rank equals to one plus number of solution n_i that dominates solution i is $r_i = 1 + n_i$. The rank 1 is assigned to nondominated set of solutions as there is no solution in a population which dominates to nondominated set of solutions. From this, it can be concluded that in any population of size N, there must be at least one solution with rank equals to 1 and utmost rank of any population member cannot be greater than N. Populations are sorted in ascending order of magnitude of its rank. Once the process of assigning rank is completed, a raw fitness to a solution is assigned based on its rank using linear mapping function. The concept of niching is introduced in order to maintain diversity among nondominated set of solutions. The fitness sharing function is calculated on objective function value instead of parameter value. Stochastic Universal selection is applied on shared fitness value, then single point crossover and bitwise mutation is applied to create new populations. Niching is performed in objective space; MOGA is suitable choice if spread of Pareto optimal solutions are required in objective space. MOGA is sensitive to the shape of the Pareto optimal front and density of solutions in the search space. The shared fitness is small for crowded solutions with better rank as niche count for these solutions is large. Due to this, required selection pressure may not exist to all solutions in a better rank there by leading to delay convergence or incapable to obtain better spread in the Pareto optimal front. The concept of nondomination is used to assign fitness; all solutions in particular nondominated front does not have the same assign fitness. This may present undesired bias towards some solutions in a search space. Shared fitness calculation does not assure that a solution in poorer rank will always have worse scaled fitness than every solution in better rank. If this happens its unable to found good spread on Pareto front. This problem is avoided by assigning fitness value front wise in Nondominated Sorting Genetic Algorithm (NSGA).

2. Nondominated Sorting Genetic Algorithm (NSGA)

The concept of nondominated sorting suggested by Goldberg was implemented by Srinivas and Deb (1994). First step in NSGA is to sort population P according to nondomination. This classifies the population into mutually exclusive equivalent classes. The population is ranked on the basis of nondomination before selection operation is performed. The nondominated individuals from the current populations are first obtained which constitute first nondominated front in the population. A substantial dummy fitness value of N (Population size) is assigned to all these nondominated individuals. The fitness assignment process starts from the first nondominated set and advances towards dominated sets. An identical fitness value is designated to these nondominated individuals in particular front for providing an identical reproductive potential. To preserve the diversity of the population, these individuals are shared with their dummy fitness values. Then, this group of individuals is omitted and another layer of nondominated individuals is taken. The procedure prolongs until all individuals in the population are analyzed [30]. The main advantage of NSGA is the assignment of fitness front wise. It is noted that, performance of NSGA is sensitive to the sharing function parameter σ_{share} ; hence, proper selection of parameter σ_{share} is required to have good spread of solution. Selection of sharing function parameter σ_{share} is the challenge in NSGA. It has been observed in the simulation studies that mutation operator seems to be destructive. Means instead of obtaining diverse set of solutions on the Pareto front number of solutions are reduced. To avoid that problem concept of elitism was introduced to preserve better solutions obtained during previous mutation operator [6].

3. Niched Pareto Genetic Algorithm (NPGA)

The NPGA proposed by J. Horn et al. used binary tournament selection scheme based on Pareto dominance, unlike proportionate selection method used in VEGA, NSGA and MOGA. This is the first proposed multiobjective optimization algorithm which uses the tournament selection operator. Here, two solutions are arbitrary selected and analyzed against a subset from the whole population (approximately 10% of the population). If one of them is dominated (by the individuals arbitrary chosen from the population) and the other is not, then the nondominated individual wins. All the other conditions are treated a tie (i.e., both opponents are either dominated or nondominated). When there is a tie, the outcome of the tournament is determined through fitness sharing [31]. The prime benefit of NPGA is no definite fitness allocation is required unlike VEGA, NSGA and MOGA. As the complexity of the NPGA does not depend on number of objectives M thus, it is found to be computationally efficient in solving problems having multiple objectives [32].

2.2.1.2 Second Generation or Elitist multiobjective genetic algorithms

Second generation of MOEAs commenced when concept of elitism was introduced. Eckart Zitzler considered being the first to introduce concept of elitism in his Strength Pareto Evolutionary Algorithm (SPEA) work published in a specialized journal [13]. After the publication of this paper, researchers begun to assimilate external populations in MOEAs and utility of elitism became a prevalent practice [33]. In connection to multiobjective optimization, elitism refers to the usage of an external population or secondary population to preserve the nondominated individuals obtained during evolution. The prime objective of elitism mechanism is to know that, a set of nondominated solutions obtained for its current population is nondominated with respect to all the populations that are produced by an evolutionary algorithm or not. The most instinctual way of doing this is by saving all the nondominated solutions in memory and compare newly generated nondominated solution with saved nondominated set of solutions. If newly generated nondominated solution dominates anyone saved in the memory, then the dominated solution from the memory must be removed [29].

Following are the most commonly used elitist MOEA:

1. Strength Pareto Evolutionary Algorithm (SPEA)

Strength Pareto Evolutionary Algorithm (SPEA) was presented by Zitzler and Thiele [34], [35]. SPEA uses an archive containing the individuals that represent a nondominated front among all solutions obtained till now (called external nondominated set). After every iteration, nondominated values are saved into an archive containing nondominated set. For each nondominated values in the archive, a strength value is obtained. SPEA uses the notion of Pareto dominance in order to assign scalar fitness value to individuals in the present population. The process begins with designating a real value s ϵ [0, 1) known as strength for each individual in the Pareto-optimal set. The strength of an individual is equal to the number of solutions covered by it. The strength of a Pareto solution is same as its fitness. The fitness of each solution in the population is the sum of the strengths of all external Pareto solutions obtained. The fitness assignment mechanism of SPEA takes into account both convergence and distributions to the real Pareto front. Hence, in place of using niche based on distance, Pareto dominance is applied to make sure that the outcomes are perfectly distributed along the Pareto front. Despite the fact that, this mechanism does not demand a niche radius, its effectiveness depends on the size of the external nondominated set. An external nondominated set is used in the selection mechanism of SPEA, if its size becomes too large, it might reduce the selection pressure, thus sluggish the exploration. Due to this, the researchers required to use a mechanism that curtails external nondominated set so that its size remains under a certain threshold.

2. Strength Pareto Evolutionary Algorithm 2 (SPEA2)

Strength Pareto Evolutionary Algorithm 2 (SPEA2) was presented by Zitzler and Thiele has following main differences with respect to SPEA [33].

(1) It takes into account a method of fine-grained fitness assignment which considers for each individual the number of individuals that dominate it and the number of individuals by which it is dominated.

(2) In order to direct search more efficiently it uses a nearest neighbor density estimation technique.

(3) It uses a mechanism that curtails external nondominated set so; it ensures safeguarding the boundary solutions.

3. Pareto Archived Evolution Strategy (PAES)

Pareto Archived Evolution Strategy (PAES) was presented by Joshua D. Knowles and David W. Corne [13]. PAES comprises of a (1 + 1) evolution tactics (i.e., a single parent that generates a single offspring) in association with an archive that stores the nondominated outcomes already obtained in the past. Each mutated individuals is compared with every member of archive. PAES algorithm comprises of a crowding technique that divides objective space in a repetitive way to maintain diversity among solutions. The solution is placed in an assured position of grid based on its objective values (known as "coordinates"or "geographical location"). A map of similar grid is preserved, indicating the number of solutions that located in each grid position. Since the process is versatile, no additional parameters are needed (except for the number of divisions of the objective space). This concept of adaptive grid proposed by Joshua D. Knowles has been accepted by several modern MOEAs.

4. Nondominated Sorting Genetic Algorithm II (NSGA-II)

Nondominated Sorting Genetic Algorithm II (NSGA-II) was presented by Kalyanmoy Deb and his students in the year 2000 [14]. This algorithm does not have much resemblance with the formerly developed NSGA, but authors kept the name NSGA-II to highlight its inception and place of origin. Unlike previously discussed algorithms which consist of using only elite preservation strategy, NSGA-II also uses explicitly diversity maintaining system. Here, offspring population Q_t is created by using crossover and mutation using the parent population P_t . Two populations Q_t and P_t are merged to form R_t of size 2N then, nondominated sorting is used to classify the entire population R_t. The consolidated population R_t is sorted according to various nondomination levels (F_1, F_2, and so on). Then, each nondomination level is chosen one at a time to form a new population S_t, beginning from F_1, until the size of S_t is up to N or for the first time surpasses N. An objective in NSGA-II is to derive number of dominated and nondominated set of solutions. The NSGA-II computes density of solution in the population based on crowding distance. In the process of selection, the NSGA-II uses a crowded comparison operator which takes into account both the nondomination rank of an individual in the population and its crowding distance (means, nondominated solutions are selected over dominated solutions, but between two solutions with the same nondomination rank, the one that exist in the less crowded region is selected). Diversity among nondominated set of solutions is obtained by using crowding comparison mechanism along with tournament selection procedure. Hence, no extra niching operator is required such as needed in MOGA, NPGA and NSGA. The NSGA-II does not use an archive as the other formerly mentioned MOEAs used. On the other hand, the elitist technique of the NSGA-II comprises of joining the best parents with the best offspring achieved (i.e., a (μ + λ) selection). Due to this intelligent process, the NSGA-II is much more competent (computationally) than its antecedent. Performance of NSGA-II is so good, that it has become very prevalent in the last few years. It has become a benchmark with which other multiobjective evolutionary algorithms required to be compared. As long as size of first nondominated set is not larger than the population size (N), this algorithm preserves all of them. Later on crowded distance operator tries to restrict the population size and hence, it loses its convergence property. When more than N members belong to the first nondominated set, then some closely packed Pareto optimal solutions may give their space to nondominated solutions. Therefore, it loses convergence property [36].

5. Nondominated Sorting Genetic Algorithm III (NSGA-III)

This algorithm is the extension of the NSGA-II algorithm for optimization of many objectives (more than four objectives) problems. This was presented by Kalyanmoy Deb and his students in the year 2013 [15]. The basic framework of NSGA-III is identical to the NSGA-II though there is a sound change in its selection procedure to maximize diversity. In the case of many objective optimization problems, proportion of nondominated solutions gets exponentially immense due to rise in number of objectives. Since, nondominated solutions seize most of the population space, any elite preserving algorithm faces problem in containing sufficient number of novel outcomes in the populations. This retards the exploration mechanism. Diversity among the population members is maintained by providing and adaptively updating reference points. An objective in NSGA-II is to derive number of dominated and nondominated set of solutions. The NSGA-II computes density of solution in the population using crowding distance operator. NSGA-II considers rank of nondomination of an individual in the population and it's crowding distance to select solution. If two solutions have same nondomination rank then solution having larger crowding distance value is selected. The crowding distance operator is replaced in NSGA-III described as under.

(1)Classification of population into nondominated levels.

The number of nondominated fronts is identified using prevailing nondominated theorem. First of all, population members from the nondominated front level '1'to level 'l'are incorporated in set S_t . If $|S_t| = N$, no further calculations are required and subsequent generation is begun with P_{t+1} =R_t . For $|S_t| > N$, members from '1'to 'l-1'fronts are selected. The remaining members K=N-$|P_{t+1}|$ are chosen from last front F_l.

(2) Deciding reference points on a hyper plane.

The set of reference points are used in order to maintain diversity among solutions. The set of reference points (denoted as H) can be predefined in a structured way or determined by the user. The algorithm is likely to obtain near Pareto optimal solutions corresponding to the provided reference points. This feature of algorithm is used for two combine application one is decision making and second is optimization. Das and Denniss systematic

approach is applied to locate points on a normalized hyper plane.
(3) Adaptive normalization of population members.
The main aim of normalization of an objective functions is to solve problems with Pareto optimal front whose objective function values are scaled distinctly. The mechanism used for normalization of objective function includes creation of hyper plane on objective space. The normalization of objective functions and generation of hyper plane is carried out every iteration to maintain diversity among population. Hence, it is known as adaptive normalization of population.
(4) Association operation.
Once normalized objective functions adaptively, it requires to associate each population members with a reference point. The objective of this operation is to find population member is associated with reference point or not? For this objective, we define a reference line corresponding to each reference point on the hyper plane by connecting the reference point with the origin. Then, determine perpendicular distance of each population member from each of the reference lines. The reference point whose reference line is nearest to a population member in the normalized objective space is considered to be associated with the population member.
(5) Niche preservation operation.
The association operation provides number of population members associated with reference points. The number of population members affiliated with particular reference points are denoted as niche count for that reference point only. The niching procedure is used to obtain niche count from the population members.

From the process mentioned for NSGA-III, it can be concluded that this algorithm does not require setting any new parameters other than usual GA parameters, similar to NSGA-II. The parameter number of reference point (H) is not an algorithmic parameter; it is directly related to the aspired number of trade-off points. The overall computational complexity of NSGA-III algorithm is higher than that of NSGA-II algorithm.

2.2.2 Advances in multiobjective optimization of PSO algorithms

Particle Swarm Optimization (PSO) algorithm falls under the category of swarm intelligence. PSO is proposed by Kennedy and Eberhart based on studies of social behavior of insects and animals [37]. In swarm intelligence an intelligent behavior is created by particles (like bird, fish, ant etc.) in the swarm. The level of an intelligent achieved by co-operations of all members of the swarm is so high that, it is not possible to reach at that level by individual member of swarm. Working of PSO algorithm is mentioned as under. First of all assume swarm of particle, where every particle is a candidate solution of an optimization problem. Particle has position and velocity in the search space of an optimization problem, for a particle 'i' at time 't' position of particle is denoted as $\vec{x_i}(t)$ and velocity of particle is denoted as $\vec{v_i}(t)$ both are member of search space and having same dimensions. Velocity of particle describes movement of the particle in terms of direction and distance traverse. In each generation, particle is updated based on two 'best' values called personal best 'pbest' $\vec{p_i}(t)$ and global best 'gbest' $\vec{g_i}(t)$ respectively. The first best value 'pbest' is the best solution value individual particle has achieved so far (its corresponding fitness value will be stored in memory). The second best value 'gbest' is the best solution value achieved by any particle in the population so far. Particle updates its velocity and position based on above two values using following two equations.

$$v_{ij}(t+1) = w * v_{ij}(t+1) + r_1 * c_1(p_{ij}(t) - x_{ij}(t)) + r_2 * c_2(g_j(t) - x_{ij}(t)) \tag{2.1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2.2}$$

Where,
r_1 and r_2 are uniformly distributed random numbers between 0 to 1.
$w * v_{ij}(t+1)$ is inertia term and w is inertia coefficient.
$r_1 * c_1(p_{ij}(t) - x_{ij}(t))$ is cognitive component.
$r_2 * c_2(g_j(t) - x_{ij}(t))$ is social component.
c_1 and c_2 are acceleration coefficients.
PSO has resemblance with evolutionary algorithms; common points for both the algorithms are mentioned as under [37].
1. They are initialized with a population of randomly generated values.
2. Individual solutions in the populations are evaluated based on fitness value.
3. In both the cases update of population and search for the optimum value are based on random techniques.
However, working of PSO differs compared to evolutionary algorithms as under.
1. PSO does not have any evolution operators like crossover and mutation here; particles update values with its internal velocity.

2. Information sharing mechanism is different in GA & PSO. In GA chromosome exchanges information with each other hence, whole population moves like a one group towards an optimal area. In PSO, only 'gbest' (global best) exchanges information. Therefore, PSO has a one way information sharing mechanism.
PSO is very simple to implement and has high speed of convergence hence, its proposed for multiobjective optimization problems [10]. Following are the widely used PSO based multiobjective optimization algorithms.

2.2.2.1 Multiobjective optimization based on Pareto dominance by Moore and Chapman[10]

The theorem based on Pareto dominance of multiobjective optimization of PSO was proposed by Moore and Chapman in an unpublished document. Here, algorithm begins by randomly initializing $\vec{x}$ and $\vec{v}$ vectors. Every time, position of particle is updated, it's compared with the solution in the Pareto dominance list(p-list) to determine nondominated solution or not. The p-list is constantly updated to maintain values of nondominated set only. In this algorithm no method for maintaining diversity is mentioned.

2.2.2.2 A swarm metaphor for multiobjective design by Ray and Liew [38]

The behavioral process of real swarm is employed like, identification of leaders from participants and information exchange among participants to attain goal of an optimization. The multilevel sieve is used to generate set of leaders and probability based crowding distance approach for leader selection (which maintains diversity). This algorithm is compared with evolutionary nondominated sorting genetic algorithm (NSGA).

2.2.2.3 The algorithm by Parsopoulos and Vrahatis [11]

The algorithm on Particle Swarm Optimization mechanism in multiobjective problems by Parsopoulos and Vrahatis claims to be first study of PSO approach in multiobjective optimization problems. Here, VEGA approach of multiobjective optimization using genetic algorithm is adapted for PSO frame work. This result in multi-swarm PSO called VEPSO algorithm. Three forms of approaches were realized and tested for two objective problems: a conventional linear aggregating function (CWA), a dynamic aggregating function (DWA) and bang-bang weighted aggregating approach (BWA).

2.2.2.4 MOPSO using dynamic neighborhood by Hu et al.[39]

The concept of dynamic neighborhood proposed by Hu and Eberhart is based on lexicographic ordering where, only one objective is optimized at a time. The algorithm is tested only for two objective optimization problems without constraints. In this approach, after every iteration particle find its new neighbor based on calculating distance to other particle. Here, distance to other particle is calculated based on fitness value of objective function. Idea of dynamic neighborhood is said to be novel in the context of MOPSO.

2.2.2.5 MOPSO based on dominated tree data structure and turbulence by Fieldsend and Singh[40]

In this study, authors have introduced concept of dominated tree data structure and turbulence variable. Dominated tree data structure consists of nondominated individuals found during search process called as unconstrained elite archive. The archive interacts with the primary populations and provides local guides. Concept of turbulence variable which is also known as mutation operator that acts on velocity variable of PSO algorithm similar to stochastic variable called as craziness. Authors have experienced that if there is no closeness between parameter space and objective space this method may experience multifrontal problems.

2.2.2.6 MOPSO based on extended memory dynamic neighborhood by Hui, Eberhart, and Shi[4]

This algorithm is an extension of the previously published work on dynamic neighborhood by Hu and Eberhart [39]. Here, concept of extended memory is introduced for storing global Pareto optimal solutions obtained in PSO to reduce computation time. It has been observed that this approach does not give satisfactory results in generating true Pareto front in more complex problems.

2.2.2.7 Strategies for finding good local guides in MOPSO by Mostaghim and Teich [7]

The sigma method was proposed by authors for obtaining best local guides of particles in the population. The turbulence operator is also applied on the population to make sudden changes in the population. The technique was compared with SPEA2 and dominated tree data structure and turbulence by Fieldsend and Singh [40]. It has been observed that use of sigma value raises the selection pressure on PSO and may result early convergence in some cases.

2.2.2.8 Nondominated sorting PSO for multiobjective optimization by Li et al.[12]

In this algorithm, concept of NSGA-II of nondominated sorting was applied to PSO. In this case instead of single comparison between particles' personal best with its offspring this method correlates all particles' personal bests and their offspring in the entire population. This method outperforms NSGA-II in some of the cases.

2.2.2.9 Handling multiple objectives with PSO by Carlos, Gregorio and Maximino [13]

In this paper, concept of Pareto dominance is incorporated with PSO to optimize multiple objective functions. Algorithm uses an external repository of particles similar to adaptive grid to direct their own flight. A special mutation operator is also used to enhance search efficacy of algorithm. The proposed method is compared with standard evolutionary algorithms like NSGA-II, PAES and Microgentic algorithm of multiobjective optimization. It has been observed that this algorithm has faster convergence rate compared to NSGA-II, PAES and Microgentic algorithm of multiobjective optimization.

2.3 Conclusion

Here, it is tried to give an overview of work that has been carried out in the field of multiobjective optimization based on evolutionary (GA) and swarm intelligence (PSO). The discussion focuses on implementation of algorithm, issues, and discussion on it's results. The survey is restricted only to the specific method of implementation i.e. genetic algorithm and particle swarm optimization only. From the literature survey it is concluded that, NSGA-II, NSGA-III, and MOPSO algorithms are the most representative multiobjective optimization algorithms in the field of evolutionary and swarm intelligence. It is observed that NSGA-II has become a benchmark with which other multiobjective evolutionary algorithms required to be compared [14]. NSGA-III is the extension of the NSGA-II algorithm for optimization of many objectives (more than four objectives) problems[15]. MOPSO algorithm has faster convergence rate compared to other algorithms in the category of PSO based multiobjective optimization algorithms [13]. Hence, NSGA-II and NSGA-III algorithms from the evolutionary category and MOPSO algorithm from the swarm intelligence category are used to compare the performance of 2DOF controller parameter optimization for the problem of shell and tube heat exchanger system.