Power Generation Loading Optimization using a Multi-Objective Constraint-Handling Method via PSO Algorithm

Lily D Li[1], Xiaodong Li[2], and Xinghuo Yu[3]

[1]School of Computing Sciences, Central Queensland University
Rockhampton, QLD 4701, Australia
l.li@cqu.edu.au
[2]School of Computer Science and Information Technology, RMIT University
Melbourne,VIC 3001, Australia,
xiaodong@cs.rmit.edu.au
[3]School of Electrical and Computer Engineering, RMIT University,
Melbourne,VIC 3001, Australia
x.yu@rmit.edu.au

Abstract — Power generation loading optimization problem will be of practical importance in the coming carbon constrained power industry. A major objective for the coal-fired power generation loading optimization is to minimize fuel consumption to achieve output demand and to maintain NO_x emissions within the environmental license limit. This paper presents a multi-objective constraint-handling method incorporating the Particle Swarm Optimization (PSO) algorithm for the power generation loading optimization application. The proposed approach adopts the concept of Pareto dominance from multi-objective optimization, and uses several selection rules to determine particles' behaviors to guide the search direction. The simulation results of the power generation loading optimization based on a coal-fired power plant demonstrates the capability, effectiveness and efficiency of using a multi-objective constraint-handling method with PSO algorithm in solving significant industrial problems.

I.   INTRODUCTION

Most power generation plants have a number of generating units. How to make the best use of the units directly affects a company's business bottom line. Increased pressures from environmental regulations, rising fuel costs, and green house gas emissions demand power generators to be more efficient and effective. For a typical power utility with a number of units, the unit thermal efficiencies (or unit heat rates) alter all the time. A unit's thermal efficiency is determined by many factors such as design, construction, fuel and ambient conditions, level of maintenance and operation skills etc. Monitoring and continuously adjusting operational strategies to optimize unit operation is of significant value to industry. For a large power company with different kinds of units, optimizing load distribution is of practical importance in terms of fuel saving and minimizing environmental harm [1].

Generally, a power generation company has a *m*-year (or *m*-month) overhaul system, i.e. each time, a unit is through a major overhaul in turn and every *m* years (or months) the plant completes an overhaul cycle. The unit which was overhauled the most recently would have highest thermal efficiency and the one close to an overhaul will have lowest thermal efficiency. Units with higher thermal efficiency will consume less fuel and cause less environmental harm while units with lower thermal efficiency will consume more fuel and lead to higher environmental harm. In the normal operation range, unit thermal efficiency increases (or heat rate decreases) as load increases. The thermal efficiency for each unit is different depending on when the unit is last overhauled, what kind of problems it developed, what modifications it went through, and what operation mode a unit is operating under (such as mill pattern etc). The optimized loading can be achieved based on the units' thermal efficiency and NO_x emission characteristics, i.e., heat rate/NO_x vs. load, for a given plant condition.

There are two objectives for the power generation loading optimization problem. One is to minimize the total heat consumption (fuel consumption) and another one is to minimize the total NO_x emission. The second objective, however, can be treated as a constraint since there is an environmental license limit in practice. It is desirable that the unit with higher thermal efficiency (lower heat rate) receives higher workload and the unit with lower thermal efficiency (higher heat rate) receives lower workload.

Over the last decade, evolutionary algorithms have been extensively studied as search and optimization tools in various problems domains[2]. Although evolutionary algorithms have been successful in many applications, their uses in solving constrained optimization problems remain problematic because

their original versions lack of a mechanism to incorporate constraints into the fitness function [3-5]. There has been little work on handling constraints by the Particle Swarm Optimization (PSO) algorithm [5]. PSO is a relative new stochastic method for optimizing hard numerical functions based on metaphor of social behavior of flocks of birds and schools of fish [6]. The PSO technique has proven to be effective and efficient for solving real-valued global unconstrained optimization problems [7, 8]. For constrained optimization problems, there have been only a few attempts. Hu and Eberhart [9] proposed a preserving feasibility strategy to handle constraints with PSO. This strategy has been adopted by our previous work [1]. The drawback of this model is that the initialization process may be impractically long or almost impossible for those CNOPs (Constrained Nonlinear Optimization Problems) that have extremely small feasible spaces [10] . Parsopoulos and Vrahatis [11] adopted the penalty functions in a PSO. However, in this model the penalty factors need to be carefully fine-tuned [4] and they are problem-dependent [12, 13]. Pulido and Coello introduced a selection rule based on feasibility checking and constraint violation measurement to handle constraints with PSO [14]. However, it appears that it is not convenient to measure the constraint violation units, and the performance is not consistent. Wei and Wang [15] integrated the multi-objective constraint handling mechanism with PSO, where a selection strategy similar to [14] is used for constraint handling. The approach is tested by four lower-constrained optimization problems with promising results.

In this paper, based on the units' performance, a mathematical formulation is firstly carried out. We then propose a new method to integrate the multi-objective constraint handling mechanism with the PSO algorithm. By converting a single objective constrained optimization problem into a bi-objective unconstrained optimization problem, the proposed approach aims to minimize the original objective function (heat consumption) and the total amount of constraint violations. The proposed approach adopts the concept of Pareto dominance from multi-objective optimization, and uses a few selection rules to determine particles' behaviors to guide the search direction.

The rest of the paper is organized as follows: Section II presents the problem formulation; Section III describes the proposed multi-objective constraint-handling method incorporating with the PSO algorithm. Section IV presents the simulation results for the power generation loading optimization problem. Section V concludes the paper and indicates some future work.

II. PROBLEM FORMULATION

It is necessary to introduce some terms and notations for the power generation loading optimization problem.

- Total load demand, denoted as M_{total} (MW), is the total power demand by the market;

- Unit load, denoted as x_i (MW), the workload allocated to unit i;
- NO_x emission license limit, denoted as P (g/m³), is the maximum emission allowed for each unit;
- Unit heat rate, denoted as f_i (KJ / KW. h), is the heat consumption for generating per unit (KW. h) electricity. For a given condition, the heat rate is a function of unit load and can be expressed by a polynomial format, which is obtained from field testing and unit modelling. The general expression for the heat rate function for unit i is

$$f_i(x_i) = a_{ik}x_i^{k} + a_{i(k-1)}x_i^{(k-1)} + \ldots + a_{i1}x_i + a_{i0}$$

where these a_i are the coefficients of the polynomial, k is the order of polynomial function;
- Heat consumption, denoted as h (MJ / h), is the unit heat consumption per hour at a given load.

$$h_i = x_i f_i(x_i)$$

- Unit NO_x emission level, denoted as q (g/m³), is the amount of emission for a given load. Each unit has its own emission curve. It is generally a linear function in the normal operation range, which is obtained from the field testing and unit modelling.

$$q_i(x_i) = b_{i1}x_i + b_{i0}$$

where b_i are the coefficients.

The major objective for the loading optimization is to determine the optimal unit load so as to minimize the total heat consumption. The total heat consumption is the sum of all units' heat consumption, which can be expressed as the following

$$F(\mathbf{X}) = \sum_{i=1}^{n} h_i = \sum_{i=1}^{n} x_i f_i(x_i)$$

where n is the number of units, x_i is the workload allocated to unit number i.
There are several constraints need to be identified:
- The total load demand must be achieved at a given time. The constraint can be expressed as

$$\sum_{i=1}^{n} x_i = M_{total}$$

Considering the data type that will be implemented in double precision, this constraint can be modified as

$$|\sum_{i=1}^{n} x_i - M_{total}| < \varepsilon$$

1633

where ε is a minimum error criterion for equality constraint .
- The NO$_x$ gas emission for each unit has to be restricted within a license limit P. This constraint can be expresses as

$$q_i(x_i) - P \leq 0 \quad (i = 1,2,...n)$$

- Unit capacity constraints. For stable operation, the workload for each unit must be restricted within its lower and upper limits. This is the range where a unit load can be readily adjusted without excessive human intervention, for example, a unit is operating between 60% to 100% load without the need of mill change. Let M_{imin} and M_{imax} represent the lowest and highest limits for unit number i respectively, the constraint then can be expressed as

$$M_{imin} \leq x_i \leq M_{imax} \quad (i = 1, 2, ...n)$$

The unit capacity constraints can be modelled as the boundary constraints in the optimization.

The optimization problem can be stated as follows:
Minimize

$$F(\mathbf{X}) = \sum_{i=1}^{n} x_i f_i(x_i)$$

subject to

$$g_1(\mathbf{X}) = |\sum_{i=1}^{n} x_i - M_{total}| - \varepsilon \leq 0$$

$$r_i(\mathbf{X}) = q_i(x_i) - P \leq 0 \quad (i = 1,2,...n)$$

where

$$f_i(x_i) = a_{ik} x_i^{k} + a_{i(k-1)} x_i^{(k-1)} + ... + a_{i1} x_i + a_{i0}$$

$$M_{imin} \leq x_i \leq M_{imax} \quad (i = 1, 2, ...n)$$

This is a single objective constrained optimization problem.

III. THE PROPOSED APPROACH

A. The PSO Algorithm

A PSO algorithm consists of individuals, called particles that form a swarm. Each particle represents a candidate solution to the problem. Particles change their positions by flying in a multi-dimensional search space looking for the optimal position. During flight, each particle adjusts its position according to its own experience and the experience its neighbouring particles, making use of the best position encountered by itself and the best position in the entire population (or its local neighbourhood). The performance of each particle is measured by a predefined fitness function (objective function) which is problem-dependent.

Let i-th particle in a D-dimensional search space be represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$. The best previous position of the i-th particle in the flight history is $pBest_i = (p_{i1}, p_{i2}, ... ,$

$p_{iD})$. The position of the best particle of the neighborhood is $lBest_i = (p_{g1}, p_{g2}, ..., p_{gD})$. The velocity for particle i is $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. In the PSO algorithm, the next position $(t+1)$ of particle i on the dimension d is manipulated by the following equations (t denote the iteration):

$$
\left.
\begin{aligned}
v_{id}^{(t+1)} &= \chi[w v_{id}^{(t)} + c_1 r_1 (pBest_{id}^{(t)} - x_{id}^{(t)}) \\
&\quad + c_2 r_2 (lBest_{id}^{(t)} - x_{id}^{(t)})] \\
v_{id}^{(t+1)} &= V_{max} \quad if \quad v_{id}^{(t+1)} > V_{max} \\
v_{id}^{(t+1)} &= -V_{max} \quad if \quad v_{id}^{(t+1)} < -V_{max}
\end{aligned}
\right\} \quad (a)
$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (b)$$

where $d = 1,2,... D$, D is the search dimension; $i = 1,2,...N$, and N is the number of particles in the swarm; w is the inertia weight; c_1 and c_2 are two positive constants, called the cognitive and social parameters respectively; r_1 and r_2 are two random numbers within the range [0, 1].

Clerc and Kennedy [16] suggested the use of a constriction coefficient χ to the velocity formula and shows that the constriction coefficient can converge without using V_{max}. In order to ensure convergence and explore a wider area, in this research, the χ, V_{ma} and w will be used together.

As mentioned, the original PSO algorithm and its variations have no mechanism to handle constraints. In order to integrate constraints handling with PSO, we introduce a constraint handling method and several selection rules in the following sections.

B. Constraint Handling Mechanism

Multi-objective constraint-handling method has been studied in genetic algorithms [4, 17], in which a global optimization problem can be transformed into a bi-objective problem where the first objective is to optimize the original objective function and the second is to minimize

$$\Phi(\mathbf{X}) = \sum_{i=1}^{m} \max(0, g_i(\mathbf{X}))$$

where $\Phi(X)$ is a total amount of constraint violations; m is number of constraints and $g_i(X)$ is the i-th constraint function. If a solution is feasible (total constraint satisfied), the second objective function Φ should be zero. By adopting this idea, the power generation loading optimization problem can be transformed into

$$minimize \quad F_i(X) = (F(X), \Phi(X))$$
where

$$F(\mathbf{X}) = \sum_{i=1}^{n} x_i f_i(x_i)$$

$$\Phi(\mathbf{X}) = \max(0, g_1(\mathbf{X})) + \sum_{i=1}^{n} \max(0, r_i(\mathbf{X}))$$

$$g_1(\mathbf{X}) = |\sum_{i=1}^{n} x_i - M_{total}| - \varepsilon$$

$$r_i(\mathbf{X}) = q_i(x_i) - P \quad (i = 1,2,...,n)$$

The original single objective constrained optimization problem is transformed into a bi-objective unconstrained optimization problem.

For a general multi-objective optimization problem, the ideal procedure is to find a set of Pareto-optimal solutions first and then to choose one solution from the set by using some other higher-level information [12]. However, the second objective should be given higher priority because constraint satisfaction is a must and it is more important than the real objective function minimization. That is, if a solution is not feasible, no matter how fit its objective function is, it is of little use. In other words, if a solution is feasible, even if it is not fit enough, it can be still considered as a candidate solution. Therefore, the second objective $\Phi = 0$ (totally constraint satisfied) or $\Phi \leq \delta$ (total constraint nearly satisfied), can be used as higher-level information to guide decision making during the search. The δ is a small positive number which indicates the feasibility tolerance.

Most multi-objective optimization methods use a dominance concept to search for non-dominated solution, since this concept allows a way to compare solutions with multiple objectives. The definition for dominance as stated below.

Definition: A solution $X^{(1)}$ is said to dominate the other solution $X^{(2)}$, if both conditions 1 and 2 are true:
1. *The solution $X^{(1)}$ is no worse than $X^{(2)}$ in all objectives, (for all $j = 1, 2, ..., m$).*
2. *The solution $X^{(1)}$ is strictly better than $X^{(2)}$ in at least one objective, for at least one $j \in \{1, 2, ..., m\}$ "* [12].

C. Selection Rules

In the PSO algorithm, a particle's best past experience and its group's best experience play a key role in guiding its search direction. For a multi-objective optimization problem, due to many objectives involved, the notion of dominance comparison is adopted [12]. The following selection rules are defined:
- Non-dominated particles are better than dominated ones.
- A particle with lower Φ (constraint violations) is better than a particle with higher Φ.

These two rules will be used in deciding which particle is better than the others.

D. The Modified PSO Algorithm

Fig. 1 illustrates the proposed modified PSO algorithm. It integrates the multi-objective constraint-handling method and the selection rules into PSO algorithm. Comparing with the original PSO, the algorithm has the following features:
- When calculating fitness, both objectives F(X) and Φ(X) need to be evaluated;
- If a particle's new location is better than its best past location, the *pBest* is updated (decided by selection rules);

```
GlobalF = POSITIVE_INFINITY;
P₀ = URand (Lᵢ, Uᵢ)
V₀ = 0
F₀ = Fitness_F (P₀)
Φ₀ = Fitness_Φ(P₀)
pBest₀ = P₀
For i = 0 To N
    lBestᵢ = LocalBest (Pᵢ₋₁, Pᵢ, Pᵢ₊₁)        (Selection rules)
End for
Do
    For i = 0 To N
        Vᵢ₊₁ = Speed ( Pᵢ, Vᵢ, pBestᵢ, lBestᵢ)   (Equation (a))
        Pᵢ₊₁ = Pᵢ+ Vᵢ₊₁                          (Equation (b))
        Fᵢ₊₁ = Fitness_F(Pᵢ₊₁)
        Φᵢ₊₁ = Fitness_Φ(Pᵢ₊₁)
        If (Pᵢ₊₁ isBetterThan pBestᵢ )           (Selection rules)
            pBestᵢ = Pᵢ₊₁
        If (Φᵢ₊₁ ≤ δ)
            If (Fᵢ₊₁ < GlobalF)
                GlobalF = Fᵢ₊₁
    End For
    For i = 0 To N
        lBestᵢ = LocalBest (Pᵢ₋₁, Pᵢ, Pᵢ₊₁)    (Selection rules)
    End for
While (iteration < max_iteration)
```

Fig.1. Pseudo code of the proposed multi-objective constraint-handling method with PSO algorithm

- A particle's best neighbouring particle is determined by the two steps:
  a) Find all the non-dominated particles in the neighborhood (by comparing two objective function);
  b) If there is only one non-dominated particle in the neighborhood, select it as *lBest*; otherwise select one with the lowest Φ as *lBest* (the lower Φ means closer to the feasible region).

IV. SIMULATION RESULTS

A. Unit Heat Rates and Unit Gas Emission Curves

A local power plant has four 360MW and a total generation capacity of 1440MW. It has a four-year overhaul system, i.e. each year, a unit is through a major overhaul in turn and every four year the plant completes an overhaul cycle.

The boundary constraints M_{min} and M_{max} for each unit are 220 (MW) and 360 (MW). The total load output of the power station ranges from $4*220 = 880$ (MW) as the minimum to $4*360 =1440$ (MW) as the maximum. It would be better to simulate a series of output (a dynamic M_{total}) so that to allow the power plant to choose from the optimal results according to the market demand.

The heat rate functions and the NO_x emission functions for the four generator units are provided from a local power plant setting. The heat rate functions are in the polynomial format with the power of two. The NO_x emission functions are linear. Table I lists the sample functions. These functions can be modified when the units' performance are changed. Due to commercial reasons, the functions have been slightly modified.

1635

TABLE I
UNIT HEAT RATE AND NO_x EMISSION FUNCTIONS

Unit No.	Type	Function
1	Heat Rate	$f(x_1) = 0.0023x_1^2 - 3.7835x_1 + 9021.7$
1	NO_x Emission	$q(x_1) = 0.0036x_1 - 0.1717$
2	Heat Rate	$f(x_2) = 0.0238x_2^2 - 9.7773x_2 + 9432.6$
2	NO_x Emission	$q(x_2) = 0.0031x_2 - 0.0226$
3	Heat Rate	$f(x_3) = 0.0187x_3^2 - 5.3678x_3 + 10240.0$
3	NO_x Emission	$q(x_3) = 0.0036x_3 - 0.1252$
4	Heat Rate	$f(x_4) = 0.0120x_4^2 - 5.7450x_4 + 9231.7$
4	NO_x Emission	$q(x_4) = 0.0039x_4 - 0.1706$

B. Parameter Setting

The minimum error criterion for equality constraint is selected as $\varepsilon = 1.0E\text{-}3$. The NO_x license limits P is 1.3 g/m^3. The PSO neighbourhood topology is set to ring topology with the neighbour size of 2. For example, if the neighbour size is 2, a particle with index i will have the particle index $i\text{-}1$ and particle $i+1$ as its neighbours. The PSO parameters are: $w = 0$; (Generally, w decreases from 1 to 0 gradually as iteration processes. In our experiment, it is found $w = 0$ works fine when V_{max} and χ to be used for restricting velocity together); $c_1=c_2=2$; $\chi=0.63$; $V_{max} = 0.5*$(decision variable range); number of particles is 40; the maximum iteration is set to 10,000. The feasibility tolerance allowed $\delta = 1.0E\text{-}8$, that is, if a solution's total amount of constraint violation $\Phi \le \delta$, the solution is considered feasible.

For each total load output M_{total}, the program runs ten times with the lowest heat consumption recorded as results.

C. Results and Discussion

Table II and Fig. 2 present the simulation results to the whole range of the generation capacity. For each total output demand, the optimal workloads to the four generators have been found based on their efficiency functions as listed in the Table I. After optimization, the unit with higher thermal efficiency will receive higher workload (such as Unit 1) while the unit with lower thermal efficiency will receive lower workload (such as Unit 3). In practice, when the total output load changes, the optimal load allocation can be found from these data. For the minimum (880MW) and maximum (1440MW) loading conditions, there is no gain from the optimization since no options for loading at both ends.

In previous work [1], the constraints was handled by adopting a preserving feasibility strategy. In order to see the difference between these two approaches, we have conducted an experiment to evaluate the computation time for each

individual run. The PSO parameters for both approaches are the same. The 40 particles, 10000 maximum iterations have been used for both experiments. Based on ten independent runs, the minimum time, maximum time and the average time spent for the $M_{total} = 1000$ MW is listed in Table III.

TABLE II
OPTIMIZED WORKLOAD DISTRIBUTION

| M_{total} (MW)	Unit 1 (MW)	Unit 2 (MW)	Unit 3 (MW)	Unit 4 (MW)
880	220.0000	220.0000	220.0000	220.000
900	235.1297	222.3830	220.0131	222.4746
950	273.1694	220.7023	220.0023	236.1254
1000	326.7896	230.9750	220.0002	222.2353
1050	359.6199	224.8249	221.3988	244.1572
1100	359.9975	227.8653	221.2147	290.9222
1150	359.8885	235.5493	221.1815	333.3800
1200	359.9999	269.2535	247.5140	323.2334
1250	359.9937	326.3954	221.1032	342.5064
1300	358.9863	339.9221	241.0938	359.9969
1350	359.9939	325.3112	305.7268	358.9679
1400	358.4181	353.5454	328.0362	359.9999
1440	360.0000	360.0000	360.0000	360.0000
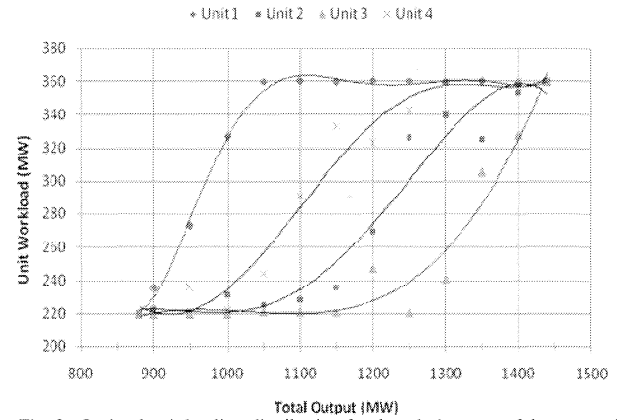
Fig. 2. Optimal unit loading distribution for the whole range of the generation capacity

Table III demonstrates the proposed multi-objective based constraint-handling method is much faster than the preserving feasibility method with PSO. The main reason is that the preserving feasibility approach assumes all particles starting at the feasible space which require a long initialization process. In other words, the iteration won't start until all particles are in the feasible space. It may be impractically too long or impossible for the problems that have large search spaces and with small feasible spaces. The multi-objective constraint-handling approach, however, doesn't require the particles to be in feasible space at beginning. The initialization doesn't need to check if the particles satisfy all constraints which make the initialization easier and faster.

It is worth mentioning that the current optimization process is in a static environment. That is, the objective function and constraints function are static for a specific case. In the real

1636

world application, however, the output demand constraint can be time-varying. The objective function can be considered static or dynamic. It would be interesting to study methods to optimize such challenging problems in the dynamic environment.

TABLE III
TIME SPENT FOR TWO APPROACHES BASED ON 10 INDEPENDENT RUNS FOR M_{total}=1000 MW

CPU time spent	*Approach in this paper (ms)	**Previous Approach [1] (ms)
Minimum	31	3016
Maximum	156	4204
Average	68.9	3925.3

* Multi-Objective Constraint-Handling with PSO
** Preserving Feasibility Constraint-handling method with PSO

V. CONCLUSIONS AND FUTURE WORK

This paper has presented a multi-objective constraint-handling method with the PSO algorithm for tackling power generation unit loading optimization problem. The proposed approach adopts the concept of dominance from multi-objective optimization, and uses a few selection rules to guide the search direction. A four-unit loading optimization for a local power plant has been simulated. The result reveals the capability, effectiveness and efficiency of applying the proposed approach in the power industry. The methodology can be readily applied to a broad range of applications such as grid optimization.

Comparing with the preserving feasibility constraint-handling approach, the multi-objective constraint-handling based approach is faster (consume less CPU time). Furthermore, the multi-objective constraint-handling method has no problem-dependent parameters like those applied in the penalty function based constraint-handling approach. This will make the approach applicable to a wide variety of applications.

Our future work will include an extensive study of multi-objective constraint handling methods with the PSO algorithms and application of the proposed approach to more challenging real-world problems. Optimization in dynamic environments is also in our agenda.

ACKNOWLEDGEMENT

REFERENCES

[1] L. D. Li, J. Zhou, X. Yu, and X. Li, "Constrained Power Plants Unit Loading Optimization using Particle Swarm Optimization Algorithm," WSEAS Transactions on Information Science and Applications, vol. 4, pp. 296-302, 2007.

[2] D. E. Goldberg, Genetic Algorithms for Search, Optimization, and Machine Learning: Addison-Wesley, 1989.

[3] C. A. C. Coello, "A Survey of Constraint Handling Techniques used with Evolutionary Algorithms, Technical Report Lania-RI-9904, Laboratorio, Nacional de Informtica Avanzada," 1999.

[4] E. Mezura-Montes and C. A. C. Coello, "Multiobjective-Based Concepts to Handle Constraints in Evolutionary Algorithms," in Proceedings of the Fourth Mexican International Conference on Computer Science Apizaco, MEXICO, 2003, pp. 192-199.

[5] G. T. Pulido and C. A. C. Coello, "A Constraint-Handling Mechanism for Particle Swarm Optimization," in Proceedings of the 2004 Congress on Evolutionary Computation. vol. 2 Portland, OR, USA, 2004, pp. 1396 - 1403.

[6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proceedins of IEEE International Conference on Neural Networks. vol. 4 Perth, Australia: IEEE Service Centerr, 1995, pp. 1942-1948.

[7] J. Kennedy, R. Eberhart, and Y. Shi, Swarm Intelligence: San Francisco: Morgan Kaufmann Publisher, 2001.

[8] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in The 7th Ann. Conf. on Evolutionary Programming, San Diego, CA, 1998.

[9] X. Hu and R. Eberhart, "Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization," in Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics, 2002.

[10] G. Coath and S. K. Halgamuge, "A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems," in Proceedings of the 2003 Congress on Evolutionary Computation Newport Beach, CA, USA, 2003, pp. 2419-2425.

[11] K. E. Parsopoulos and M. N. Vrahatis, "Paricle Swarm Optimization Method for Constrained Optimization Problem," Intelligent Technologies--Theory and Application: New Trends in Intelligent Technologies, vol. 76 of Frontiers in Artificial Intelligence and Applications, pp. 214-220, IOS Press, pp. 214-220, 2002.

[12] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms: John Wiley & Sons, 2001.

[13] Ruhul Sarker, Masoud Mohammadian, and X. Yao, "Evolutionary Optimization," in International Series in Operations Research & Management Science, F. S. Hillier, Ed., 2002.

[14] G. T. Pulido and C. A. C. Coello, "A Constraint-Handling Mechanish for Particle Swarm Optimization," in proceeding of CEC 2004 San Diego, CA, USA, 2004, pp. 1396-1403.

[15] J. Wei and Y. Wang, "A Novel Multi-objective PSO Algorithm for Constrained Optimization Problems," in Proceeding of 6th Interantional Conference on Simulated Evolution and Learning. vol. 4247 Hefei: Springer, Berlin / Heidelberg, 2006, pp. 174-180.

[16] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space," IEEE Transactions on Evolutionary Computation, vol. VOL. 6, NO.1, pp. 58-73, 2002.

[17] E. Camponogara and S. N. Talukdar, "A Genetic Algorithn For Constrained and Multiobjective optimization," in Proceeding of 3rd Nordic Workshop on Genetic Algorithms and Their Applications Vaasa, Finland, 1995, pp. 49-62.