Performance Based Unit Loading Optimization using Particle Swarm Optimization Approach

LILY D LI
School of Information Technology
Central Queensland University, Rockhampton, QLD 4701, AUSTRALIA
l.li@cqu.edu.au http://www.cqu.edu.au

JIPING ZHOU
Stanwell Power Station
Rockhampton, QLD 4701, AUSTRALIA
jiping.zhou@stanwell.com http://www.stanwell.com.au

YINGHUO YU
School of Electrical and Computer Engineering
RMIT University, Melbourne, VIC3001, AUSTRALIA
x.yu@rmit.edu.au http://www.rmit.edu.au

Abstract: - This paper presents a Particle Swarm Optimization (PSO) based approach for economically dispatching generation load among different generators based on the unit performance. A modified PSO algorithm with preserving feasibility and repairing infeasibility strategies is adopted for handling constraints. A four-unit loading optimization for an Australian power plant is successfully implemented by using the modified PSO algorithm. The result reveals the capability, effectiveness and efficiency of using evolutionary algorithms such as PSO in solving significant industrial problems in the power industry.

Key-words: - Evolutionary Computing, PSO algorithm, Optimization, Loading dispatching, Application

1 Introduction

Most power companies have a number of generating units and how to make the best of each unit directly affects a company's bottom line. Increased pressures from environmental regulations, rising fuel costs, and green house gas emission 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 rate) change all the time. The unit thermal efficiency is determined by many factors such as design, construction, level of maintenance and operation skills etc. Monitoring and continuously adjusting operational strategies to optimize unit operation is of practical use. To a large scale power company with different kinds of units adopting a total load bidding system, optimizing load distribution is of practical importance in terms of fuel saving and minimizing environmental harm [1],[2].

A major objective of the loading optimization is to minimize the heat consumption (fuel consumption) for a given generating output or bidding at a given time. The heat consumption is dependent of each unit's thermal efficiency and its workload. 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. Meanwhile, the constraint on output demand and the unit capacity must be maintained.

The methods to tackle constrained optimization problem have been categorized in two groups – deterministic and stochastic. The deterministic methods find the optimum up to certain accuracy while the stochastic methods find the optimum up to a certain probability. In other words, the deterministic methods find solution more accurate but sometimes they cannot find the solution if the objective functions are not well defined. The stochastic methods, however, do not have specific requirements of the objective functions and can find solution with a certain probability. Since the deterministic methods impose strong assumptions on the continuity and differentiability of the objective function, the stochastic methods such as evolutionary algorithms have been increasingly becoming an alternative approach to address the complicated optimization problems [3],[4],[5]. PSO is a relative new stochastic method for optimizing hard numerical functions on metaphor of social behavior of flocks of birds and schools of fish [6],[7],[8]. The PSO technique has proven to be effective and efficient for solving real valued global unconstrained optimization problem [9],[10],[11]. However, PSO approach has not been fully used for constrained optimization problems. In [12], four categories for handling constraints in evolutionary optimization approaches are summarized, i.e. preservation of feasibility, penalty functions, searching for feasibility and other hybrids. Promising results have been reported by using these methods in evolutionary optimization [13],[14],[15]. These constraints handling techniques have potential to be adopted in PSO because PSO possesses similar characteristics as evolutionary optimization. For example, they are both stochastic, population based, evolving from generation to generation. The only difference is, instead using crossover and mutation, a PSO system uses each individual's best past experience and its neighbors' best experience. Some researches have reported the results by using these techniques in PSO [3],[10],[16].

This research presents a PSO based algorithm for the unit loading optimization problem for electricity utilities. A main reason for choosing PSO is that PSO appears to be able to find the global optimum effectively while Genetic Algorithms (GA) can sometimes easily fall into the local optimum [7]. In addition, research with PSO indicates that by properly setting the parameters, the global optimum can be found more quickly on average [11], [17].

In this paper, based on the units' performance, a mathematical formulation is firstly carried out. The original PSO algorithm is modified by adopting the preserved feasibility and repaired infeasibility for handling the constraints. A four-unit loading optimization for a local power plant is successfully implemented by using the modified PSO algorithm. The result reveals the capability, effectiveness and efficiency of applying evolutionary algorithm such as PSO algorithm in the power industry.

In the next section, the problem formulation is presented. The PSO algorithm and the constraints handling strategy are then described in section 3. A performance based unit loading optimization simulation is reported in section 4. Section 5 concludes the paper.

2 Problem Formulation

Before the problem formulation, some definitions are first introduced.

a. Plant total load demand, denoted as M_{total} (MW), is the total plant load bid.

b. Unit load, denoted as x (MW), the workload allocated to each unit.

c. Unit heat rate, denoted as f (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 is

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

where i is unit number, these a_{ix} are the coefficients of the polynomial, k is the order of polynomial function.

d. Heat consumption, denoted as hc (MJ / H), is the unit heat consumption per hour at a given load.

$$hc = xf(x)$$

The objective for the loading dispatching 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

$$F(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} hc_i = \sum_{i=1}^{n} x_i f(x_i)$$

where n is the number of units, x_i is the workload allocated to unit number *i*.

There are several constraints:

I. The total load constraint must be maintained and adjustable according to the demand. The constraint can be expressed as

$$\sum_{i=1}^{n} x_i = M_{total} \quad \text{(MW)}$$

Considering the data type will be implemented in double precision, it is difficult to maintain an exact equality. The above constraint can be modified as

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

where ε is a minimum error criterion.

II. Unit capacity constraints. For stable operation, the workload for each unit must be restricted within its lower and upper limits. Let $M_{i\min}$ and

$M_{i\max}$ represent the lowest and highest limits for unit number i respectively, n is the number of units, the constraint then can be expressed as

$$M_{i\min} \le x_i \le M_{i\max}$$
$$(i = 1, 2, \ldots n)$$

The optimization problem is stated as follows:

Minimize $F(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i f(x_i)$

where

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

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

$$M_{i\min} \le x_i \le M_{i\max} \quad (i = 1, 2, \ldots n)$$

3 PSO Algorithm and Constraint Handling

Particle Swarm Optimization, originally developed by Kennedy and Eberhart in 1995 [6] , is a method for optimizing hard numerical functions on metaphor of social behaviour of flocks of birds and schools of fish [6],[8]. A swarm consists of individuals, called particles. Each particle represents a candidate solution to the problem. Particles change their position 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 of its neighbouring particles, making use of the best position encountered by itself and its neighbours. 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 $X_i = (x_{i1}, x_{i2}, \ldots x_{iD})$. The best previous position of the *i*-th particle in the fly history is $pBest_i = (p_{i1}, p_{i2}, \ldots p_{iD})$. The best particle of the swarm, e.g. the particle with the most desired objective function value, is $gBest = (g_1, g_2, \ldots g_D)$. The velocity for particle i is $V_i = (v_{i1}, v_{i2}, \ldots v_{iD})$. In the PSO algorithm, the next position of particle i on the dimension d is manipulated by the following equations (the superscripts denote the iteration):

$$V_{id}^{t+1} = wV_{id}^{t} + c_1 r_{1id}^{t}(pBest_{id}^{t} - x_{id}^{t}) + c_2 r_{2id}^{t}(gBest_d^{t} - x_{id}^{t}) \quad \text{(a)}$$

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

where w is the *inertia* weight. The c_1 and c_2 are two positive constants, called the *cognitive* and *social* parameters respectively. These two constants are used to determine particles' individuality weight and sociality weight. The r_{1id} and r_{2id} are two random numbers within the range [0, 1].

To determine who is and isn't in a particle's "neighbourhood", Kennedy and Eberhart discovered that using smaller, overlapping neighbourhoods was often more effective than using a global neighbourhood topology (i.e. all the particles as neighbours) [8]. Therefore, it is a common practice to construct particles into different topology styles with a certain size of neighbours.

The preserving feasibility method introduced in GENOCOP system [14] assumes that the constraints are all linear and the start points are all feasible. When initializing, particles can be generated within the entire search space but only those who are in feasible space (satisfy all the constraints) are kept for processing. However, although initial particles are all in the feasible space, during flying, they may get out of the feasible space to become infeasible due to improper parameter settings. In order to maintain the population diversity and to keep the population size for next generation, it would be better to get these infeasible particles repaired rather than rejecting them. Unfortunately, there are no standard repairing algorithms for every situation. The repairing infeasibility methods lie in their problem dependence [18]. In this research, an infeasible particle is to be repaired by replacing the infeasible particles with a closer, first-found feasible particle. The algorithms are illustrated Fig.2 (a) and (b). Since the loading optimization problem has one liner constraint, intuitionally, this constraint handling method will satisfy.

Fig. 1 is the modified PSO algorithm. Compare with the original PSO algorithm, two modifications have been made:

1. All particles are repeatedly initialized until they are feasible. The initial particles can be generated randomly.
2. During flying (iteration), if particles are not feasible, repair them to be feasible. Then calculate the fitness.

```
For each particle {
   Do {
     Initialize particle
   } While particle in the feasible space
}
Do {
   For each particle {
       If (the particle is NOT in the feasible space)
        {
          Repair particle to be feasible, call
                 repairing algorithm
        }
        Calculate fitness
        If (the fitness value is better than the best
          fitness value (pBest) in history {
          Set current value as the new pBest
        }
     }
     Choose the particle with the best fitness
value of all neighbourhood particles as the gBest
   For each particle {
       Calculate particle velocity according
            equation (a)
       Update particle position according
            equation (b)
   }
} While maximum iteration is not attained or
  minimum error criteria is not attained
```

Fig. 1. The modified PSO algorithm

Fig. 2 (a) is a graphic illustration of the repairing algorithm. Ps is a infeasible particle, Pr is a feasible reference particle, Z1, Z2… are those attempt particles between Ps and Pr, Zn is the first-found feasible particle between Ps and Pr. Zn will be used as a repaired partible of Ps. Fig. 2 (b) is the repairing algorithm.

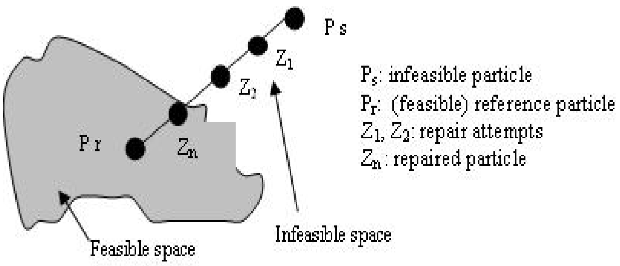

Fig. 2 (a). The graphic illustration of the repairing algorithm

1. Select a reference particle Pr.
2. Create a sequence of candidate particles Z_i between Pr and Ps:

$$Z_i = a_i P_s + (1 - a_i) P_r$$

   Where $0 < a_i < 1$, is generated at random

   The process stops when the first feasible Zn is found.
3. With Zn the repaired particle of Ps, go back to the PSO algorithm for fitness calculation

Fig. 2 (b). The infeasibility repairing algorithm

4 Performance Based Unit Loading Optimization

4.1 Unit Heat Rates

An Australian 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 unit recently completed an overhaul will have a highest efficiency and the one close to overhaul will have a 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 increase. The slop for each unit is different depending on when the unit is last overhauled and what kind of problems it developed and what modifications it went through. The optimized loading can be achieved based on the units' thermal efficiency characteristics, i.e., heat rate vs. load.

The heat rate curve for the four generator units are provided in a local power plant setting, which is in the polynomial format with the power order of 2. The functions are listed in the table 1. These functions can be modified when the units' performance are changed.

Table 1 Unit Heat Rate Functions

Unit No.	*Unit Heat Rate Function
1	$f(x_1) = 0.0023x_1^2 - 3.7835x_1 + 9021.7$
2	$f(x_2) = 0.0238x_2^2 - 9.7773x_2 + 9432.6$
3	$f(x_3) = 0.0187x_3^2 - 5.3678x_3 + 10240.0$
4	$f(x_4) = 0.0120x_4^2 - 5.7450x_4 + 9231.7$

* For simulation purposes only; due to commercial reasons the figures have been modified.

4.2 Parameter Setting

The total load output of the power station ranges from 880 MW as the minimum to 1440 MW as the maximum. This will cover units' whole range of the capability and allow user to choose according to the demand. The minimum error criterion ε is defined as 1.0E-7. For each total load output, the program runs ten times with the lowest heat consumption recorded as result.

In the infeasibility repairing algorithm, the reference particle *Pr* is determined by using the average load, i.e. the reference particle is defined as the following:

$$P_r = (M_{total}/4, M_{total}/4, M_{total}/4, M_{total}/4)$$

The *Pr* is the unit load allocation before the optimization. It will be used to in the heat consumption saving comparison.

The population size of PSO is set to 40. The generation (iteration) is set to 10000. The neighbourhood topology is selected as a CIRCLE type. The neighbour size is 5. The velocities are restricted in [-4, 4]. The boundary constraint type is set to be "Stick", i.e., if the velocity value great than the boundary value, it will be sticked to equal to the boundary value. The individuality weight *c1* and the sociality weight *c2* are set to 2 respectively. The inertia weight *w* is set to 1.

4.3 Results and Discussion

For each output load demand, four generators have been optimized allocated based on their efficiency curves. Meanwhile, the heat consumption for an average load allocation is also calculated which can be used for the optimization benefit comparison.

After optimization, the unit with higher thermal efficiency will receive higher workload while the unit with lower thermal efficiency will receive lower workload. In practice, when the total output load changes, the optimal load allocation can be found from the optimization results. The PSO system should be executed again if any unit's performance changes.

Heat consumption can be obtained from the objective function for an average allocation (before optimization applied) and an optimized allocation (after the optimization). The heat consumption saving is calculated for comparing the difference between the two. The formula is.

$$\text{Heat Consumption Saving} = \sum_{i=1}^{4} x_{avg} f_i(x_{avg}) - \sum_{i=1}^{4} x_i f_i(x_i)$$

where the $x_{avg} = M_{total} / 4$, the f_i are the heat rate curve listed in the Table 1.

From the heat consumption saving, the fuel savings based on fuel heating value or the calorific value and the price of fuel can be calculated. The result is illustrated in Fig.3.

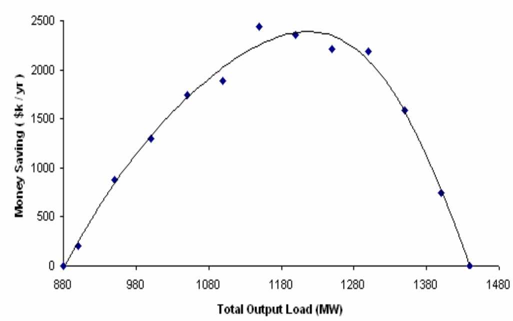

Fig. 3. Annual Money Saving from Loading Optimization (Calorific value = 26 MJ / kg, fuel price = $28 /per ton)

The curve in Fig.3 indicates that most benefits from load optimisation are made around 1200MW in excess of annual fuel saving of two million dollars while no gain is obtained on minimum and maximum loading conditions, which is logical as no options for loading at both ends. In reality, it is impossible to always operate the plant in such a desirable way, i.e., cannot guarantee all four units keep running for a whole year without stoping. Assume there is a 50% chance of possible loading optimisation, the benefits will be halved and fuel savings will be around one million dollars per year.

5 Conclusion

Loading dispatching optimization problem is a widely recognized problem in power industry. A number of researches suggested that PSO is one of the most effective, efficient and robust search methods in optimization practice. However, constraint handling is still a key issue. A modified

PSO approach has been proposed in this paper for economically dispatching generation load among different generators based on the unit performance, which adopts preserving feasibility and repairing infeasibility strategies for handling constraints. A four-unit loading optimization for a local power plant is successfully implemented by using the modified PSO algorithm. The result reveals the capability, effectiveness and efficiency of applying evolutionary algorithm such as PSO algorithm in the power industry. The methodology can be readily applied to greater application such as grid optimization.

References

[1] Diesendorf, M., Australia's Polluting Power: Coal-fired electricity and its impact on global warming, in WWF Australia. 2003.

[2] National Statement on Coal Power Stations Proposals, http://www.cana.net.au/nomorepollutingpower/NoCoalNationalStatement.pdf.

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

[4] Wu, B., Evolutionary strategies for function optimization, fuzzy modeling and control. 2001, phd thesis, Central Queensland University: Rockhampton.

[5] Yao, X., Evolutionay Computation, A Gentle Introduction, in Evolutionary Optimization, R. Sarker, M. Mohammadian, and X. Yao, Editors. 2002, Kluwer Academic Publishers. p. 27-53.

[6] Kennedy, J. and Eberhart, R., PSO optimization. Proceedins of IEEE International Conference on Neural Networks. Vol. 4. 1995, Perth, Australia: IEEE Service Centerr. 1942-1948.

[7] Fan, H., A modification to particle swarm optimization algorithm. Engineering Computations, 2002. vol.19: p. 970-989.

[8] adaptiveview.com, An Introduction to Particle Swarm Optimization, http://www.adaptiveview.com/articles/ipsop1.html.

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

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

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

[12] Koziel, S. and Michalewicz, Z., Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. Evolutionary Computation, 1999. vol. 7, no.1: p. 19-44.

[13] Runarsson, T. and Yao, X., Constrained Evolutionary Optimization, in Evolutionary Optimization, R. Sarker, M. Mohammadian, and X. Yao, Editors. 2002, Kluwer Academic Publishers. p. 87-113.

[14] Michalewicz, Z. and Janikow, C.C., GENOCOP: A Generic Algorithm for Numerical Optimization Problems with Linear Constraints. Communication of the ACM, 1996.

[15] Yu, X. and Wu, B., An Adaptive Penalty Function Method for Constrained Optimization with Evolutionary Programming. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2000. vol 4(2): p. 164-170.

[16] Zavala, A.E.M., Aguirre, A.H., and Diharce, E.R.V. Constrained Optimization via Particle Evolutionary Swarm Optimization Algorithm (PESO). in Proceedings of the 2005 conference on Genetic and evolutionary computation. 2005.

[17] Eberhart, R.C. and Shi, Y. Comparison between Genetic Algorithms and Particle Swarm Optimization. in Proceedings of the 7th International Conference on Evolutionary Programming VII. 1998: Springer-Verlag, Berlin.

[18] Zbigniew, M. and Schmidt, M., Evolutionary Algorithms and Constrained Optimization, in Evolutionary Optimization, R. Sarker, M. Mohammadian, and X. Yao, Editors. 2002, Kluwer Academic Publishers. p. 57-86.