Constrained Power Plants Unit Loading Optimization using Particle Swarm Optimization Algorithm

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

XIAODONG LI
School of Computer Science and Information Technology
RMIT University, Melbourne, VIC3001, AUSTRALIA
xiaodong@cs.rmit.edu.au http://www.rmit.edu.au

Abstract: - Power plants unit loading optimization problem is of practical importance in the power industry. It generally involves minimizing the total operating cost subject to satisfy a series of constraints. Minimizing fuel consumption while achieve output demand and maintain emissions within the environmental license limits is a major objective for the loading optimization. This paper presents a Particle Swarm Optimization (PSO) based approach for economically dispatching generation load among different generators based on the units’ performance. Constraints have been handled by a proposed modified PSO algorithm which adopting preserving feasibility and repairing infeasibility strategies. A simulation of an Australia power plant implementing the modified algorithm is reported. 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 use 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, provided that each unit’s emission levels are within the environmental license limits. In the power station
In this paper, based on the units' performance, a mathematical formulation is firstly carried out. The original PSG algorithm is modified by adopting the preserved feasibility and repaired infeasibility for handling the constraints. A simulation of an Australia power plant implementing the modified algorithm is reported. The result reveals the capability, effectiveness and efficiency of using evolutionary algorithms such as PSG in solving significant industrial problems in the power industry.

In the next section, the problem formulation is presented. The PSG 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 \( M_{\text{total}} \) (MW), is the total plant load bid.

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

c. NOx emission license limit \( P \) (g/m^3).

d. 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 for unit \( i \) is

\[
f_i(x_i) = a_{i_k} x_i^k + a_{i(k-1)} x_i^{(k-1)} + \ldots + a_{i1} x_i + a_{i0}
\]

where these \( a_{ik} \) are the coefficients of the polynomial, \( k \) is the order of polynomial function.

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

\[
h_c = x_i f_i(x_i)
\]

f. Unit NOx emission emission level, denoted as \( g \) (\( \mu g/m^2 \)), is the amount of emission for a given power output. 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.

\[
g_i(x_i) = b_{i1} x_i + b_{i0}
\]

where \( b_{i1} \) and \( b_{i0} \) are the coefficients.

The 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

\[ F(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} h c_i = \sum_{i=1}^{n} x_i f_i(x_i) \]

where \( n \) is the number of units, \( x_i \) is the workload allocated to unit number \( i \).

There are several constraints:
I. The total load constraint must be maintained and adjustable according to the demand. The constraint can be expressed as

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

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

\[ \left| \sum_{i=1}^{n} x_i - M_{\text{total}} \right| < \epsilon \]

where \( \epsilon \) is a minimum error criterion.

II. Each unit's NOx gas emission has to be restricted within a license limit \( P \). This constraint can be expressed as

\[ g_i(x_i) < P \quad (i = 1, 2, \ldots n) \]

III. 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 \( M_{\text{imin}} \) and \( M_{\text{imax}} \) represent the lowest and highest limits for unit number \( i \) respectively, \( n \) is the number of units, the constraint then can be expressed as

\[ M_{\text{imin}} \leq x_i \leq M_{\text{imax}} \quad (i = 1, 2, \ldots n) \]

The optimization problem is stated as follows:

Minimize

\[ F(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i f_i(x_i) \]

where

\[ f_i(x_i) = a_{i0} x^4 + a_{i(k-1)} x^{(k-1)} + a_{i1} x + a_{i0} \]

subject to

\[ \left| \sum_{i=1}^{n} x_i - M_{\text{total}} \right| < \epsilon \]

\[ g_i(x_i) < P \quad (i = 1, 2, \ldots n) \]

\[ M_{\text{imin}} \leq x_i \leq M_{\text{imax}} \quad (i = 1, 2, \ldots n) \]

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 \( p\text{Best}_i = (p_{i1}, p_{i2}, \ldots , p_{id}) \). The best particle of the swarm, e.g. the particle with the most desired objective function value, is \( g\text{Best} = (g_1, g_2, \ldots , g_D) \).

The velocity for particle \( i \) is \( V_{id} = (v_{i1}, v_{i2}, \ldots , v_{id}) \). 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):

\[ V_{id}'^{t+1} = w V_{id}'^t + c_1 r_{id}' (p\text{Best}_i'^t - x_{id}'^t) + c_2 r_{id} (g\text{Best} - x_{id}'^t) \]

\[ x_{id}'^{t+1} = x_{id}'^t + V_{id}'^{t+1} \]

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 \( r_{id}' \) and \( r_{id} \) 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 two linear constraints, intuitively, 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, i.e. to satisfy all constraints. 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.

Fig. 2 (a) is a graphic illustration of the repairing algorithm. \( P_s \) is a infeasible particle, \( P_r \) is a feasible reference particle, \( Z_1, Z_2 \ldots \) are those attempt particles between \( P_s \) and \( P_r \), \( Z_n \) is the first-found feasible particle between \( P_s \) and \( P_r \). \( Z_n \) will be used as a repaired particle of \( P_s \). Fig. 2 (b) is the repairing algorithm.

For each particle {
  Do {
    Initialize particle
  } While particle in the feasible space (i.e. satisfy all constraints)
}

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
optimized loading can be achieved based on the units’ thermal efficiency and NOx emission characteristics, i.e., heat rate/NOx vs. load, for a given plant condition.

The heat rate curves and the NOx emission functions for the four generator units are provided in a local power plant setting. The heat rate curves are in the polynomial format with the power of 2. The NOx emission functions are linear. Table 1 and Table 2 list the sample functions. These functions can be modified when the units’ performance are changed. Due to commercial reasons, the functions have been modified.

<table>
<thead>
<tr>
<th>Unit No.</th>
<th>Unit Heat Rate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( f(x) = 0.0023x^3 - 3.7835x + 9021.7 )</td>
</tr>
<tr>
<td>2</td>
<td>( f(x) = 0.0238x^3 - 9.7773x + 9432.6 )</td>
</tr>
<tr>
<td>3</td>
<td>( f(x) = 0.0187x^3 - 5.3678x + 10240.0 )</td>
</tr>
<tr>
<td>4</td>
<td>( f(x) = 0.0120x^3 - 5.7450x + 9231.7 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit No.</th>
<th>Unit NOx Emission Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( g(x) = 0.0036x - 0.1717 )</td>
</tr>
<tr>
<td>2</td>
<td>( g(x) = 0.0031x - 0.0226 )</td>
</tr>
<tr>
<td>3</td>
<td>( g(x) = 0.0036x - 0.1252 )</td>
</tr>
<tr>
<td>4</td>
<td>( g(x) = 0.0039x - 0.1706 )</td>
</tr>
</tbody>
</table>

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 \( \varepsilon \) is defined as 1.0E-7. The NOx license limits \( P \) for each unit is 1.3 g/m³. 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 \( P_r \) is determined by using the average load, i.e. the reference particle is defined as the following:

\[
P_r = (M_{\text{aver}}/4, M_{\text{aver}}/4, M_{\text{aver}}/4, M_{\text{aver}}/4)
\]
The $P_r$ 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 stucked to equal to the boundary value. The individuality weight $c_1$ and the sociality weight $c_2$ 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.

Fig 3 illustrates the optimization results. 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 this figure. The PSO system should be executed again if any unit’s performance changes.

Fig. 3. Optimization results

In order to see the benefit gained from the optimization, heat consumption can be calculated 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.

\[
\text{Heat Consumption Saving} = \sum_{i=1}^{4} \frac{\mathcal{H}_i(x_{avg}) - \mathcal{H}_i(x_{\text{opt}})}{\mathcal{H}_i(x_{avg})}
\]

where the $x_{avg} = \frac{M_{load}}{4}$, the $\mathcal{H}_i$ 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.4.

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 stopping. In practice it is also not ideal to always moving unit around due to its negative impact on dynamic losses and plant life. 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 of practical usage in contemporary 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 simulated by implementing 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