2008 IEEE Swarm Intelligence Symposium
St. Louis MO USA, September 21-23, 2008

Power System Optimization under Uncertainties: A PSO Approach

V.S. Pappala and I. Erlich

Abstract—**Most power systems optimization problems have to be solved under uncertainty. The scenarios used for modeling the uncertainties should be able to represent their stochastic nature. If this requires huge sampling, particle swarm optimization (PSO) based scenario reduction technique can be a good option to approximate the initial scenario distribution. This paper proposes a multi-stage model for the optimal operation of a wind integrated power system. A parameter free self learning particle swarm optimization algorithm has been used to solve the deterministic and stochastic models. The robustness of the solution procedure has been verified by the effective utilization of the various generation units.**

I. INTRODUCTION

OPTIMIZATION problems in power systems are usually huge, complex and highly nonlinear. In many occasions they have to be solved under uncertainties, i.e. without the complete information of the stochastic quantities like electrical demand and wind generation. By using stochastic programming approach the size of the problem will increase significantly. However, in order to simulate realistic power systems, the optimization problems need to be solved without considerable reduction or approximation of the models. Besides, in practical application, aspects of easy adaptation and extension are of particular importance. This is made possible with the evolution of new robust computational intelligence tools which uses only the evaluations of objective and constrains functions and has no obligation on the characteristic of these functions.

The stochastic quantities such as wind power in feed, photovoltaic generation, electrical demand etc. associated with the optimization model can not be forecasted accurately and involves forecast error. Ignoring the influence of these uncertainties on the optimization models may lead to unrealistic results. These variables should be modelled so that they can be easily incorporated in the cost model. One way to model these uncertainties is by the use of scenario trees [1]. A scenario represents an instance of the evolution of the stochastic variable over a period of time. A huge number of scenarios are required to completely describe the randomness of these variables. Solving the stochastic optimization model with huge set of scenarios is computationally expensive. So the initial huge set of scenarios should be approximated to a reduced set of scenarios. The scenario reduction techniques presented so far use distance metrics [2] to select the best scenarios. The techniques perform one-to-one comparisons to select a set of distinct scenarios. This comparison strategy limits the number of initial scenarios. The use of computational Intelligence (CI) techniques may overcome these disadvantages. The CI based scenario reduction can perform few intelligent and strategic comparisons to decide the best set of scenarios.

Heuristic methods like particle swarm optimization (PSO), ant colony optimization, genetic algorithms, harmony search, bees algorithm, tabu search, simulated annealing, diffusion search etc. have been successfully applied to effectively solve large-scale nonlinear optimization problems. However these global searching algorithms require the selection of optimal population size, suitable parameters for the evolutionary update strategy and appropriate penalty coefficients for handling constrained optimization. The algorithms are therefore problem dependent and require intense parameter tuning for a specific application. This paper presents a PSO based black box optimization tool which is problem independent and can also overcome the major demerits of evolutionary algorithms.

The main objectives of this paper are to present a multistage stochastic model to solve optimization problems under uncertainty, to introduce computational intelligence tools like PSO for scenario reduction technique and finally to present an adaptive self tuning PSO that can be used as a black box and suitable for solving large scale stochastic optimization problems in power systems.

II. OPTIMIZATION UNDER UNCERTAINTY

Many decisions in power system planning, operation and control have to be made without the complete information of the uncertainties such as generator outages, loss of load, wind power fluctuations etc. prevailing in the system model. So these uncertainties should be modeled in a suitable form to be used in the optimization model. The evolution of these uncertainties is usually represented by a scenario tree as shown in Fig.1. A scenario tree is a discrete approximation of the evolution of the uncertainties over time. The scenario tree consists of finite decision making points or nodes. The probability of happening of node n is π_n. Each node, n_i of the scenario tree has a unique predecessor node, n_j and a transition probability $\pi_{ni/nj}$, i.e. probability that n_i being the successor of n_j. The parent node, n_1 (π_{n1}=1) branches of into two successive node n_2 and n_3 with transition probabilities $\pi_{n2/n1}$ and $\pi_{n3/n1}$ respectively. These nodes further branches until the planning horizon. The path from the parent node to a node at the final time stage constitutes a scenario.

Manuscript received June 13, 2008. This Project is supported by the German Federal Ministry of Education and Research (BMBF) under the grant 03SF0312A.

Pappala Venkata Swaroop (venkata.pappala@uni-due.de) is with the institute of electrical power systems, Universität Duisburg-Essen.

Istvan Erlich (istvan.erlich@uni-due.de) is with the institute of electrical power systems, Universität Duisburg-Essen, 47057- Duisburg, Germany).

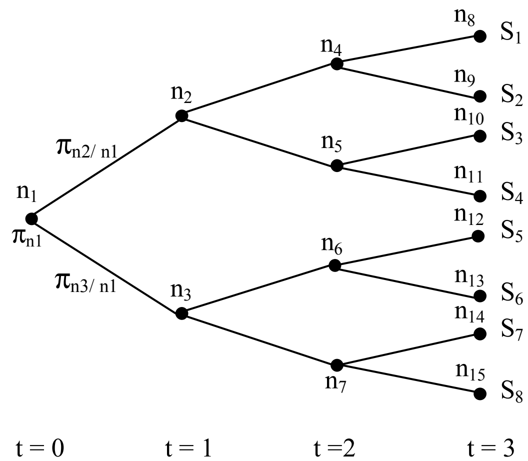

Fig.1. Binary scenario tree with three branching stages

A well defined scenario tree therefore represents the stochastic nature of the uncertainties. This tree can be incorporated into the cost model using the multi-stage stochastic programming approach [3]-[6]. Scenario generation and reduction process poses many challenges: This paper deals with day ahead planning of the power system. The daily demand and renewable generation can not be forecasted accurately. The stochastic nature at each time hour of the day is assumed to be a normal distribution [7]. The statistical information of these distributions can be obtained from the forecasting tool. A huge of sampling or scenarios may be required to cover these distributions at each of the day. This is practically infeasible. So the Gaussian distributions are approximated to five discrete samples of equal probability. This implies that each node of the scenario tree has five equiprobable successors. But even this approximation explodes the scenario set. A suitable scenario reduction technique should be used to filter the huge scenario set.

A. Scenario Generation with PSO

The currently existing scenario reduction techniques [8]-[2] can handle only a small initial scenario set. The reduction is based on a certain distance measure. The scenario to be deleted is selected by rigorous comparison among the scenarios. These one-to-one distance comparisons restrict their use for huge scenario set. One way to handle huge initial scenario set is using computational intelligence methods. A PSO based scenario generation technique is proposed in this paper. This process always has a fixed number of scenarios or particles. During the iterative process, the scenario tree represented by the swarm adjusts its branches to trace the best scenario tree. So the process is a scenario generation and not a reduction process. The generation process is formulated as an optimization problem. The objective is to select the best set of scenarios from an initial set as huge as a million scenarios. The algorithm always has a fixed set of particles or scenarios. These particles will strategically explore the entire search space to position them at prominent areas so as to improve their individual fitness. This is not a global optimization. The swarm therefore does not look for a single optimal solution but instead try to improve the fitness of each and every particle. The generation process is explained by the following algorithm.

Step1: Search space definition and swarm initialization: The search space represents the entire set of scenarios. The swarm size which represents the number of preserved scenarios is fixed before the start of the generation process. The swarm is initialized by randomly selecting scenarios from the search space.

Step2: Calculate particle/scenario probability: The particles have dynamic probability. The swarm represents the reduced scenario set. After each iteration the swarm is compared to the initial scenario set. The probability of the missing scenarios is added to its nearest existing scenarios in the swarm.

Step3: Fitness evaluation: Each particle is assigned a fitness value based on the Euclidean distance with its neighboring particles as shown below:

$$fitness\,(particle^{\,j}) = \min_{q \in N_p} \pi_j \left\{ \frac{1}{N_D} \sum_{l=1}^{N_D} \frac{\left(\sum_{r=1}^{N_T} (A_{l,r}^j - A_{l,r}^q)^2 \right)^{\frac{1}{2}}}{N_l} \right\} \qquad (1)$$

where N_P is the swarm size, N_D is the number of random variables, N_T is the total number of branching stages and $A_{l,r}^j$ is the value of the random variable l corresponding to particle j at r[th] dimension. Since the random variables are measured to different scales, the distance has to be normalized by a factor N_l. The aggregated distance is weighted by the scenario or particle probability π_j. The fitness of the particle will be the aggregated multivariate Euclidean distance with its nearest neighbor weighted by its probability.

Step4: Evaluate pbest and gmin: Each particle in the swarm keeps the traces of its own best performance (pbest) and also the performance of its nearest neighbor (gmin).

Step5: Update position and velocity: The particles update strategy is a modified version of standard PSO update equations [9] as shown below:

$$v_{jk}^{m+1} = \chi(wv_{jk}^m + \phi_1(pbest_{jk}^m - x_{jk}^m) - \phi_2(gmin_k^m - x_{jk}^m)) \qquad (2)$$

The particle's exploration at iteration m+1 is influenced by its previous velocity, its previous best performance and also by the performance of its nearest neighbor. Each particle acknowledges the flight of its neighbors, gmin and therefore is smart enough to move away from them and improve its personal performance. The calculated velocity is then added to the current position to obtain the new position for the particle. Each dimension of the real valued position vector is rounded of to its nearest discrete value.

Step6: Turbulence: For simplicity, the scenario generation process is implemented using the classical version of PSO. So this algorithm might end up with a local minimum. Whenever a particles update deteriorates the fitness of its nearest neighbor, turbulence is added to its velocity to avoid premature convergence.

Step7: Termination criterion: The stopping criterion is usually the maximum number of iterations. Where, iteration includes the complete update of the swarm. Steps 2-6 are repeated until the stop condition is reached.

The optimization process will try to improve the fitness of each particle. Since the particles represent scenarios, the end process results in scenarios with prominent probability and distinct distance from the rest of the scenarios. The PSO based scenario generation process is able to explore the entire set of scenarios and able to make few intelligent comparisons to decide on the best set of distinct scenarios.

B. Formulation of stochastic optimization problem

The optimization problem under uncertainty [10]-[11] is implemented as a multi-stage stochastic model. The aim of the model is to minimize the expectation of the objective function, f over all possible scenarios of uncertainty set ω. Vector $\mathbf{x}$ is the set of all decision variables and vector $\mathbf{x}_{t,s}$ is the set of all decision variables corresponding to scenario s at time t.

$$\text{Minimize} \quad \mathrm{E}_{\omega} f(\mathbf{x}, \omega) \tag{3}$$

Assume that the uncertainties are represented by a scenario tree with S scenarios and the probability of happening of scenario s is π_s. A general formulation of such a stochastic model is as shown below:

$$\text{Minimize} \quad \sum_{s \in S} \pi_s f_s(\mathbf{x}_s) \tag{4}$$

$$\text{Subject to:} \quad g_{i,t,s}(\mathbf{x}_{t,s}) = 0, \ \forall s \in S, \text{ for } i=1,..,y \tag{5}$$

$$h_{j,t,s}(\mathbf{x}_{t,s}) \leq 0, \ \forall s \in S, \text{ for } j=1,..,z \tag{6}$$

The stochastic model has to be optimized at each node of the scenario tree. So the problem size increases with the size of the scenario tree. A self adaptive PSO algorithm is used to solve this stochastic model.

III. Self Adaptive Particle Swarm Optimization

Particle swarm optimization is an evolutionary computational technique developed by Kennedy and Eberhart [12]-[14]. PSO tracks the optimal solution through an evolutionary process inspired by fish schooling and birds flocking in search of food. PSO algorithm is initialized by a population or swarm of random potential solutions called particles. The particles explore the search space at a certain velocity. The particles adjust their flight based on their own previous flying experience and also on the performance of the best particle in the swarm. PSO solves a highly constrained problem as an unconstrained one by adding a suitable penalty function to the main objective function. This paper presents a self-adaptive particle swarm optimization algorithm.

Self-adaptive particle swarm optimization (APSO) is a parameter free optimization tool. The basic structure of the algorithm is derived from the tribe concept introduced by Maurice Clerc [15]. The updates to the original version of this algorithm include a self learning penalty function to handle constrained optimization. This approach eliminates penalty coefficients tuning. A special provision to efficiently handle unit commitment (UC) variables is incorporated in this algorithm. This new variable modelling considerably reduces the total number of UC variables.

The particles of this algorithm have the capability to modify their search strategy based on their personal performances. The particles and the swarm adapt to the situations to find a global optimal solution. The prominent merit of this algorithm is that the optimization process is modelled as a black box where only the search space, the objective function to be minimized along with the constraints and a stopping criterion for the algorithm are to be given. It is totally free from parameter tuning and also from the burden of selecting the most appropriate swarm size. The algorithm is inspired from the nomad community. Nomads are groups of people who move from place to place following the seasonal availability in search for a better living. This algorithm simulates the moving strategy of different sized groups of nomads called "Tribes". Each such tribe succeeds in finding a local minimum. Information linkages between the tribes as shown in Fig. 2 assist the swarm to predict the global solution.

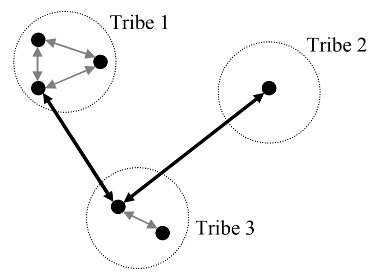

Fig.2. Tribes of different sizes and the information linkages

A single tribe consisting of a single particle ignites the search process. If this tribe can not converge on an optimal region, it will add more information in the swarm by generating new tribes. New particles will be born only when they are required. Particles which do not contribute to the search process are eliminated. The swarm therefore has only significant particles, which helps the swarm to trace the optimal solution. This substantially helps the algorithm to find the solution with few iterations. The evaluation and evolution of the swarm should not be done at every iteration. So the swarm is given enough time to exploit the solution space before they are evaluated. The search process is controlled by two iterative loops.

The constrained optimization is converted to an unconstrained optimization by using the penalty function approach. Many static and dynamic penalty functions [16] are in use. But all these approaches require optimal coefficient tuning so a self learning parameter free penalty function [17] approach is used to handle the constraints. The penalized objective function is the sum of distance term, d($\mathbf{x}$) and penalty term, penalty($\mathbf{x}$). The objective function (4) is modified to incorporate penalty function as shown below:

$$\text{Minimize} \quad d(\mathbf{x}) + \text{penalty}(\mathbf{x}) \tag{7}$$

The distance value is defined as follows:

$$d(\mathbf{x}) = \begin{cases} b'(\mathbf{x}) & \text{if } r_f = 0 \\ \sqrt{(\sum_{n \in N} \pi_n f_n'(\mathbf{x}_n))^2 + b'(\mathbf{x})^2} & \text{otherwise} \end{cases} \quad (8)$$

Where

$$r_f = \frac{number\ of\ feasible\ particle}{swarm\ size} \quad (9)$$

Where f'($\mathbf{x}$) is the normalized objective function value and b'($\mathbf{x}$) is the sum of the normalized violation of each constraint divided by the total number of constraints. The penalties and thereby the penalized objective function adapt to the requirement and necessity of the swarm. For instance when the swarm has no feasible particles, the new objective function has to minimize only the constraint violation. On the other hand when the swarm is in search of promising areas, then the penalized objective function has to minimize both the objective function and the constraint violation. This function is performed by the distance term, d(x).

The prominent feature of this approach is that it identifies the right set of infeasible particles which can help the search process. Not all the information carried by the infeasible particles is important. Different set of infeasible particles are important at different stages of the search process. For example when the swarm is in need of feasible particles, low penalties are assigned to infeasible particles with low constraint violation irrespective of their objective value. When the swarm has enough feasible particles and is exploring the optimal region, low penalties are assigned to infeasible particles with low objective value irrespective of their constraint violation. The penalty term in (7) performs this task. The penalties are therefore adaptive to the progress of the search process and independent of the problem.

The optimization process can be explained by the following steps.

Step 1: Swarm Initialization: A particle represents a possible solution to a given problem. The position of particle *j* is determined by vector $\mathbf{x}_j$ comprising of the set of decision variables at each node of the scenario tree. The velocity of the particle is given by $\mathbf{V}_j$. The particles are randomly generated within the stipulated bounds on each variable

A group of particles represent a tribe. A group of tribes form the swarm. Initially the swarm has a single tribe with one particle. Each particle is associated with a fitness value calculated using the new objective function in (7). Each particle has a set of informers to help the particle in its search process. The size of this group is adaptive.

Step 2: Update procedure: The particles utilize their previous two individual performances to decide the type of update strategy to be used for their search process. The performances can be an improvement (+), status quo (=) or a deterioration (-). A bad particle (--,- =, = =) requires global exploration and therefore prefers pivot strategy. Whereas good particles (=+, ++) require more local search and hence Gaussian update strategy is used to provide a good balance between local and global exploration. These strategies are explained below.

1) Pivot strategy: This moving strategy randomly allocates the position of a particle within the space defined by two hyperspheres. The two hyperspheres H_p and H_g are defined around the individual best performance, p and best informer of the particle, g. The new position vector, $\mathbf{x}_j^{m+1}$ of the particle at iteration m+1 is given by the following equation.

$$\mathbf{x}_j^{m+1} = c_1 random(H_p) + c_2 random(H_g) \quad (10)$$

Where, c_1 and c_2 are the weight factors calculated using the penalized objective function corresponding to p and g respectively.

2) *Gaussian Update strategy*: The selection of optimal acceleration coefficients controlling the influence of local and global best is avoided by defining a set of Gaussian distributions as shown below:

$$\delta_{pk} = x_{pk}^m - x_{jk}^m \quad (11)$$

$$\delta_{gk} = x_{gk}^m - x_{jk}^m \quad (12)$$

The velocity of the particle j at k^{th} dimension and iteration m+1 is influenced by its previous velocity (v_{jk}^m), its previous best performance (x_{pk}^m) and also by the performance of the best informer (x_{gk}^m).

$$v_{jK}^{m+1} = v_{jk}^m + gauss\_rand(\delta_{pk}, |\delta_{pk}|/2) + gauss\_rand(\delta_{gk}, |\delta_{gk}|/2) \quad (13)$$

$$x_{jk}^{m+1} = x_{jk}^m + \chi(v_{jk}^{m+1}) \quad (14)$$

Where gauss_rand(μ, σ) generates a normal distributed numbers with mean μ and standard deviation σ. The good particle to which this strategy is applied requires a high proportion of local exploration compared to global exploration. The Gaussian distribution will perfectly match this requirement and therefore helps the particle to provide an optimal search strategy. The continuous and discrete variables are treated alike during the update process. After the position update, the discrete variables are rounded to the nearest integer. The binary variables are handled using sigmoid function approach.

The update procedure for the particles of all tribes constitute one cycle. If the variables cross the bounds, they are made to stay on the boundary.

Step 3: Inner loop termination criterion: The inner loop consists of step 2. A finite number of cycles control the inner loop.

Step4: Evaluation: Outside the inner loop the swarm is evaluated. The swarm is considered good if it has more good tribes and vice versa. The information lists of the bad swarm are reorganized so as to allow the tribes to explore better. The tribe is considered good if it has more number of good particles. If the good tribe has more than one particle then it sacrifices its worst particle whereas a bad tribe will add a new particle.

Step 5: Outer loop termination criterion: The outer loops consist of inner loop and step 4. A finite number of function evaluations limit the outer loop.

This algorithm has been used for solving the deterministic and stochastic optimization model.

IV. Day Ahead Planning of Power Systems Taking into Account Load and Wind Power Uncertainties

The intermittent nature of the wind power poses serious problems to power system operators in planning and operation of the wind integrated power system. Irrespective of the unpredictable nature of wind energy, the generation resources have to be optimally utilized to cut down the overall operating cost. Storage devices such as pumped storage can be of tremendous help in this regard. The pumped storage is used to level the mismatch between power generation and demand. They store the excess generation from wind farms and also the excess generation by the base load generation plants during off-peak periods for later use. This will enable efficient utilization of the base-load generation units and to smooth the peak loads. The pumped storage can also be used to provide reserve during off-peak period so that no other unit is committed just for providing the reserve.

The test system considered in this paper consists of four thermal (2*1000 MW and 2*500 MW units), nuclear base generation unit (5000MW), wind farm (326 MW) and six pumped storage units (6*190 MW and 6*175 MW). The optimal daily operation of this power system is formulated as an optimization problem to reduce the daily operating costs subjected to a set of unit and system constraints.

Minimize Σ Operating costs
Subject to the following constraints:
(1) Unit constraints
(2) Minimum up/down time limits
(3) Pond level dynamic constraints
(4) Pond level storage limits
(5) Initial and final pond levels
(6) Generation and pumping level constraints

The optimization process has to trace the optimal UC schedule for the thermal power plants and decide the generation levels of the committed units at each hour of the day. The pump storage units should be effectively utilized to reduce the peak demand supplied by the generation units and support reserve when necessary to reduce the overall operating costs. For UC problem, the binary variables representing the on/off of the generating units are represented as positive or negative integers [18] as shown below:

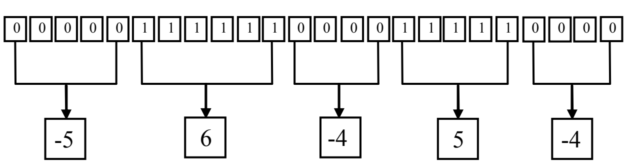

Fig.3. Binary UC variables transformed to integers

The binary string in Fig. 3 represents the hourly scheduling of a generating unit for a period of 24 hours. The positive or negative integers can be interpreted as aggregated binary UC variables. A positive integer characterize the duration of period when the unit is 'on' and negative integer indicates the period when the unit is in the 'off' state. The scheduling variables I_k^c of unit k are therefore represented by N_c integers. The UC integer variables have to fulfill the following constraints:

$$\sum_{c=1}^{N_c} \left| I_k^c \right| = T \quad \text{for k=1,2,3,4} \tag{15}$$

$$-T \le I_k^c \le T \quad \text{for k=1,2,3,4} \tag{16}$$

Where T is the total planning period (24 hours)

A. Deterministic Cost Model

In deterministic optimization forecasted data is used for the uncertainties. The model therefore optimizes one single scenario representing the forecast data. The operation mode and scheduling of the pump storage units is shown in Fig.4. The positive scheduling represents the generation mode of the pump storage and the negative scheduling indicates pumping mode. The pump storage is in pumping mode during the off peak period during which the pond is refilled. During the peak hours, the pump storage functions in generation mode to support the peak demand. In Fig.5 the net electrical demand with and without pump storage is shown. The energy required for pumping is added as an additional demand whereas generation by pump storage is considered as a negative load. The use of pump storage reduces the peak electrical demand by 8%. The overall demand profile is smoothened and hence the generation from the short term expensive generation units can be reduced and this is reflected in the overall operating costs. This is shown by the curve "electrical demand with PS". In Fig.6 the reserve requirement and reserve contribution by pump unit and thermal plant are shown. The pump storage also contributes the spinning reserve. The unused generation capacity of the pump storage units can be used to support the reserve requirement. During the peak load period, no additional thermal unit is switched on only to supply the

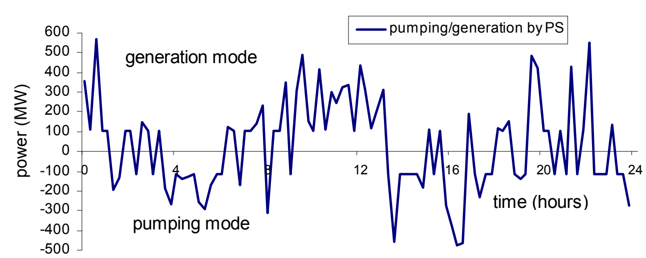

Fig.4. Optimal schedule of the pump storage units by APSO

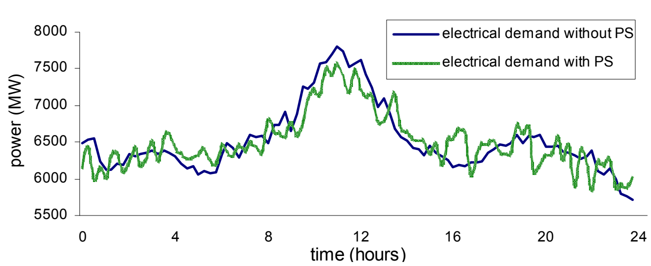

Fig.5. The net electrical demand with and without pump storage units

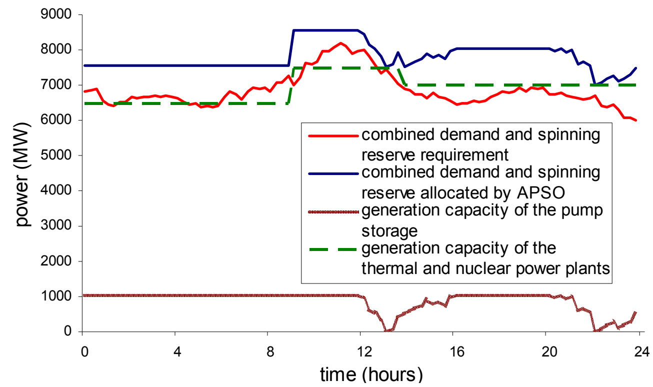

Fig.6. Reserve contributions from thermal power plant and pump storage unit

reserve. This reduces the operating costs of the whole system. The overall operating cost for the deterministic model amounts to 1,546,701 Euro. Planning the operation of a power system based on the deterministic cost model involves huge risk. The model completely relies on the forecasted data but to what extent can forecast information be trusted? The most sophisticated forecast tool can estimate the day-ahead wind generation with an error of 60-70%. This high forecast error indicates the necessity for a stochastic model.

B. *Scenario Analysis*

The increase in the stochastic nature of the forecasted electrical demand and wind generation are shown in Fig. 7(a) and Fig. 8(a) respectively. The forecasted profile is shown by the bold curve. The shaded region indicates the confidence interval. This region is expected to cover the likely outcomes of the uncertain variables.

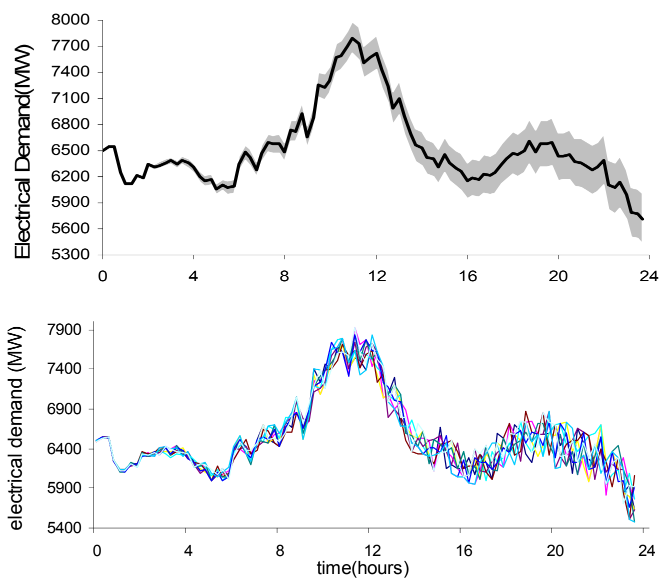

Fig.7. (a) Load forecast with confidential interval (b) five best load scenarios generated by PSO

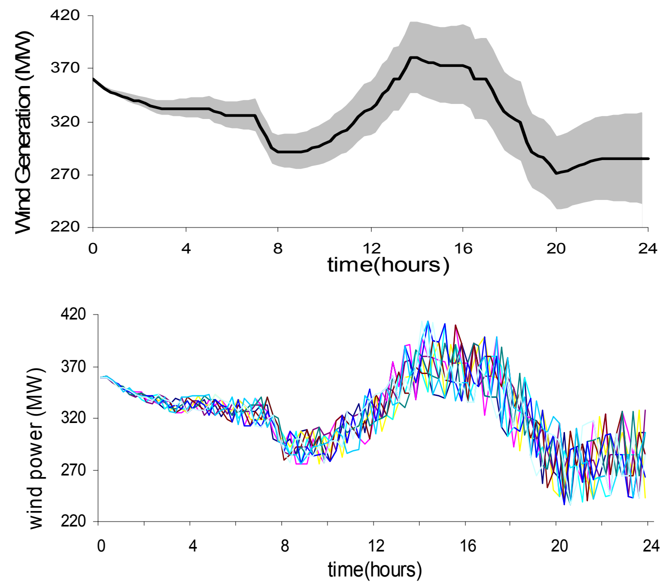

Fig.8 (a) wind generation forecast with confidential interval (b) five best wind scenarios generated by PSO

The performance of the scenario generation algorithm was evaluated by comparing the stochastic solution corresponding to the initial tree and the reduced tree. Since it is difficult to solve the stochastic model with huge initial tree, only three branching stages are assumed in wind and the load is assumed to be constant. Hence the initial scenario tree consists of 125 scenarios representing the wind power variations at three time stages. The stochastic cost model was solved for different set of scenarios and the corresponding operating costs are projected in Fig.9. As the number of scenarios increase, the stochastic solution almost converges to the stochastic solution corresponding to the initial scenario tree with 125 scenarios. This indicates that the reduced set of scenarios generated by PSO algorithm is a good approximation to the initial set of scenarios.

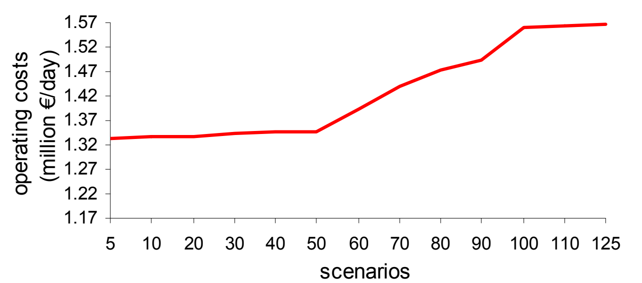

Fig.9. Change in total operating cost with number of scenarios representing three branching stages in wind uncertainties.

The performance of the swarm consisting of five particles/scenarios during the evolutionary process is depicted in Fig.10. Particles 2 and 5 start with a very low fitness value. This implies that these particles are very close to each other but the difference in fitness value is due to their individual probability. These particles therefore try to explore promising areas to improve their fitness without degrading its nearest neighbors. However the improvements in these particles cause a decrease in the fitness of the remaining particles. After a few iterations all the particles are positioned

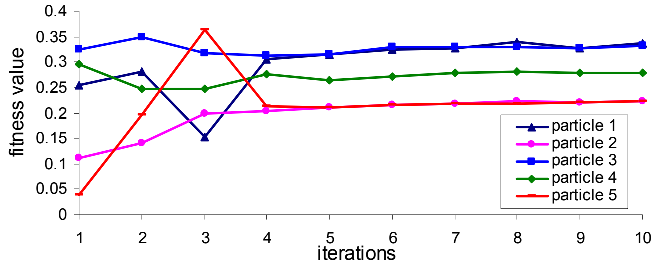

Fig.10. Performance of the 5 particle swarm in scenario generation algorithm by PSO

at optimal locations with promising distance and probability. The strength of the algorithms rests in its ability to ensure that no particle coincide with another particle in the swarm, in which case the distance with its nearest neighbor is zero and hence its fitness is also zero. It means that the new velocity update equation is able to provide enough information to the particle regarding the flight of the other particles in the swarm. The particles are intelligent enough to position themselves optimally in the entire search space with distinct probabilities. The algorithm therefore has the ability to select the optimal set of scenarios from an extremely huge initial set.

C. Stochastic Cost model

The stochastic cost model for the considered power system is given by:

$$\min \sum_{s \in S} \pi_s \left\{ \sum_{t=1}^{24} \sum_{i=1}^{4} \left(FC_{i,t}^{s} u_{i,t} + SC_{i,t}^{s} u_{i,t} \left(1 - u_{i,t-1} \right) \right) \right\} \quad (17)$$

The aim of this stochastic model is to minimize the operating cost overall possible scenarios. The significant feature of this modeling is that the unit commitment schedule of all the four generating units is fixed for all the scenarios. The solution to this model yields a robust UC schedule which is suitable for any scenario that might possibly happen.

The stochastic nature of the wind farm generation output and the uncertain demand affects the scheduling and operation of various generation units in the power system. The stochastic nature of the two uncertainties is analyzed at each hour of the 24 hour planning period. Hence there are 23 branching stages for the scenario tree. Each node of the scenario tree is assumed to have five equally probable successors. The resulting scenarios are reduced by the proposed PSO based scenario generation technique. The cost model solved over this reduced scenario set. This model tries to optimize the total operating costs over all scenarios. The resulting solution is therefore optimal for any scenario that might happen in the future. The operating costs of the stochastic model with a common unit commitment schedule for the whole set of scenarios amounts to 1,807,717 Euro.

The solution provided by the deterministic model is optimal only to a particular scenario and is not optimal for the scenario that may actually occur. The solution obtained by the stochastic model is optimal over all the possible scenarios. The stochastic solution may not be a global optimal solution to the individual scenarios but it a robust solution over all the scenarios. Hence the expectation of the operating costs corresponding to all these scenarios is high compared to the deterministic cost model. However the risk involved in using this model for the operation and planning is quite low and is therefore more preferred than the deterministic model. By defining a common unit commitment schedule for all possible realizations of the uncertainties will help the power system operator to decide the operation of the generators one day in advance irrespective of the evolution of the uncertainties. This enables better planning and operation of the power system.

V. Conclusion

Many decision problems in planning, operation and control of power systems require a nonlinear optimization problem to be solved under uncertainty. PSO based scenario analysis can be used to model the complete stochastic nature of the uncertainties. The proposed scenario generation technique will be a potential alternative to the conventional methods.

The huge stochastic optimization problems can be readily formulated and addressed by APSO. The adaptive swarm size, parameter free velocity update equations and self learning penalty functions frees the user from the burden of parameter tuning. In general, APSO seems an efficient alternative for solving nonlinear mixed-integer optimization problems. The effective use of the power system resources in the illustrated application indicate the robustness of the optimal solutions obtained by APSO.

References

[1] Kjetil Hoyland and Stein W. Wallace, "Generating Scenario Trees for Multistage Decision Problems" Management Science 2001 INFORMS Vol. 47,No 2,February 2001 pp. 295-307.

[2] Diana Barro, Elio Canestrelli and Pierangelo Ciurlia, "Optimal Scenario Tree Reduction for Financial Optimization Problems, " AURORA TR 1998-22, Univ. Wien, Vienna, Austria, 1998.

[3] Ruszczynski, A., Shapiro, A., eds.: Stochastic Programming. Volume 10 of Hand books in Operations Research and Management Science. Elsevier (2003).

[4] Andy B. Philpott, Antonio J. Conejo, and José M. Arroyo. "A Stochastic Programming Approach to Electric Energy Procurement for Large Consumers," IEEE Trans. Power Systems, vol. 22, No. 2, pp. 744–754, May 2007

[5] J. R. Birge and F. Louveaux, Introduction to Stochastic Programming. New York: Springer-Verlag, 1997

[6] John R. Birge "Stochastic Programming Computation and Applications," INFORMS Journal on Computing, Vol. 9, No. 2, Spring 1997.

[7] V.S. Pappala, I. Erlich, "Management of distributed generation units under stochastic load demands using particle swarm optimization," IEEE General Meeting , Tampa, Florida 2007.

[8] Jitka Dupacova, Nicole Growe-Kuska, and Werner Römisch. "Scenario reduction in stochastic programming," Mathematical Programming, 95(3):493–511, 2003.

[9] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995

[10] Shabbir Ahmed, Alan J. King, and Gyana Parija. A multi-stage stochastic integer programming approach for capacity expansion

under uncertainty. Stochastic Programming E-Print Series, http://www.speps.org, 2001
[11] Diana Barro and Elio Canestrelli. A decomposition approach in multistage stochastic programming. Rend. Studi Econ. Quant., pages 73-88, 2005
[12] Kennedy, J. and Mendes, R. “Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms,” IEEE International Workshop on Soft Computing in Industrial Applications 2003 (SMCia/03), pp. 45-50, 2003.
[13] Hu, X., Eberhart, R. C., and Shi, Y. “Engineering optimization with particle swarm,” Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003), Indianapolis, Indiana, USA. pp. 53-57, 2003
[14] J.Kennedy and R.C. Eberhart. Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
[15] Maurice Clerc, "TRIBES, a Parameter Free Particle Swarm Optimizer" French version: 2003-10-02.Presented at OEP’03, Paris, France.
[16] M. Gen, R. Cheng, “A survey of penalty techniques in genetic algorithms,” Proceedings of IEEE International Conference on Evolutionary Computation, 1996, 20-22 May 1996 Page(s):804 – 809.
[17] B. Tessema, G.G. Yen, “A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization,” IEEE Congress on Evolutionary Computation, 2006. 16-21 July 2006 Page(s):246 – 253.
[18] V.S. Pappala, I. Erlich, “A new approach to solve the unit commitment problem by adaptive particle swarm optimization,” IEEE General Meeting , Pittsburgh, PA 2008.