

# **Application of PSO for Optimization of Power Systems under Uncertainty**

Der Fakultät für Ingenieurwissenschaften der  
Universität Duisburg-Essen  
zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr. -Ing.)

genehmigte Dissertation

von

**Pappala Venkata Swaroop**

aus  
Indien

Referent: **Prof. Dr.-Ing. habil. István Erlich**  
Korreferent: **Dr. Ganesh Kumar Venayagamoorthy**

Tag der mündlichen Prüfung: 27.11.2009



## Acknowledgement

I would like to express my sincere gratitude to my advisor Prof. Dr.-Ing. habil. Istvan Erlich, whose encouragement, supervision and support from the initial to the concluding level enabled me to produce this thesis. His perpetual energy and enthusiasm in research had motivated me to do this Ph.D. I could not have imagined having a better advisor and mentor for my Ph.D study. I am privileged and proud to have worked under his esteemed supervision. I sincerely dedicate this thesis to him.

I especially want to thank Prof. Dr.-Ing. habil. Gerhard Krost for his valuable guidance during my research. His great efforts to explain things clearly and simply have helped a lot of young researchers, including me to successfully carry out my study at the university. I would also like to express my sincere thanks to my co-advisor Dr. Ganesh Kumar Venayagamoorthy for his valuable feedback, comments and suggestions to my dissertation.

My heartiest thanks to Hannelore Treutler, for helping me with the administrative issues especially with my visa and Ralf Dominik, for providing all the software resources. I never started my day without the warm greetings from Rüdiger Reißig. Special thanks to him for making my stay at 'Fachgebiet Elektrische Anlagen und Netze', the most memorable part of my life.

My deepest appreciation to my colleagues Koch, Jens, Shewarega, Befekadu, Teeuwsen, Feltes, Robert, Nakawirow, Hashmani, Ayman, Stark, Wilch and Zamri for providing a stimulating and fun environment at the institute. I am indebted to my friends Jai, Naveen, Kiran, Sagar, Murali, Anka Rao and Subash who were always with me in bliss and despair.

Most importantly, I wish to thank my parents, Chalapathi rao and Chandra Kumari, my sisters Malathi and Srilatha, my brother-in-laws Siva Kumar and Dorababu for helping me get through the tough times, and for all the emotional support, love and caring they provided me throughout my life. Finally, I would like to thank all people who have helped and inspired me during my doctoral study.



## Abstract

The primary objective of this dissertation is to develop a black box optimization tool. The algorithm should be able to solve complex nonlinear, multimodal, discontinuous and mixed-integer power system optimization problems without any model reduction. Although there are many computational intelligence (CI) based algorithms which can handle these problems, they require intense human intervention in the form of parameter tuning, selection of a suitable algorithm for a given problem etc. The idea here is to develop an algorithm that works relatively well on a variety of problems with minimum human effort. An adaptive particle swarm optimization algorithm (PSO) is presented in this thesis. The algorithm has special features like adaptive swarm size, parameter free update strategies, progressive neighbourhood topologies, self learning parameter free penalty approach etc.

The most significant optimization task in the power system operation is the scheduling of various generation resources (Unit Commitment, UC). The current practice used in UC modelling is the binary approach. This modelling results in a high dimension problem. This in turn leads to increased computational effort and decreased efficiency of the algorithm. A duty cycle based modelling proposed in this thesis results in 80 percent reduction in the problem dimension. The stern up-time and downtime requirements are also included in the modelling. Therefore, the search process mostly starts in a feasible solution space. From the investigations on a benchmark problem, it was found that the new modelling results in high quality solutions along with improved convergence.

The final focus of this thesis is to investigate the impact of unpredictable nature of demand and renewable generation on the power system operation. These quantities should be treated as a stochastic processes evolving over time. A new PSO based uncertainty modelling technique is used to abolish the restrictions imposed by the conventional modelling algorithms. The stochastic models are able to incorporate the information regarding the uncertainties and generate day ahead UC schedule that are optimal to not just the forecasted scenario for the demand and renewable generation in feed but also to all possible set of scenarios. These models will assist the operator to plan the operation of the power system considering the stochastic nature of the uncertainties. The power system can therefore optimally handle huge penetration of renewable generation to provide economic operation maintaining the same reliability as it was before the introduction of uncertainty.



## Table of Contents

<b>Acknowledgement .....</b>	<b>i</b>
<b>Abstract .....</b>	<b>iii</b>
<b>Table of Contents.....</b>	<b>v</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Objective.....	3
1.3 Outline .....	4
<b>2 Particle Swarm Optimization .....</b>	<b>5</b>
2.1 Introduction.....	5
2.2 Simple PSO Algorithm .....	6
2.3 Algorithm Settings .....	7
2.3.1 Population Size.....	7
2.3.2 Maximum Velocity .....	7
2.3.3 Acceleration coefficients .....	8
2.3.4 Inertia weight.....	8
2.3.5 Constriction factor.....	8
2.3.6 Neighbourhood Topology .....	9
2.4 PSO Versions.....	9
2.4.1 Binary PSO.....	9
2.4.2 Discrete PSO .....	10
2.4.3 Adaptive PSO.....	10
2.4.4 Multiobjective PSO .....	12
2.5 Applications.....	12
2.6 Novel Particle Swarm Optimization Algorithm.....	13
2.6.1 Introduction .....	13
2.6.2 Who can benefit?.....	14
2.6.3 The solution strategy .....	14
2.6.4 Description of the Proposed Approach.....	18
2.6.5 Performance of APSO .....	20
2.7 Summary .....	28
<b>3 Constrained Optimization .....</b>	<b>31</b>
3.1 Introduction.....	31
3.2 Penalty Functions.....	33
3.2.1 Static Penalty .....	34

3.2.2 Dynamic Penalty .....	35
3.2.3 Adaptive Penalty Function .....	35
3.2.4 Self Learning Penalty Function (AP) .....	36
3.3 Novel Approach (Rule Based Penalty Approach, RBP ) .....	41
3.3.1 Why do we need a fitness function? .....	41
3.3.2 The Idea.....	42
3.3.3 Who decides the camouflages? .....	42
3.3.4 The Procedure .....	43
3.3.5 The Benefits .....	44
3.4 Test Problems .....	44
3.5 Summary.....	60
<b>4 Unit Commitment Problem .....</b>	<b>63</b>
4.1 Introduction .....	63
4.2 Problem Formulation .....	64
4.3 Particle Formulation .....	66
4.3.1 Binary Coding .....	66
4.3.2 Integer Coding.....	66
4.4 Binary Programming .....	67
4.4.1 Repair Strategy .....	67
4.4.2 Mutation Operator .....	68
4.4.3 Generation .....	68
4.5 APSO Approach .....	68
4.5.1 Swarm Initialization .....	68
4.5.2 Velocity Update .....	69
4.5.3 Additional Constraints.....	70
4.6 Special Convergence Operators.....	71
4.6.1 Demand Equalizer .....	71
4.6.2 Reserve Manager.....	71
4.7 Numerical Example .....	71
4.8 Conclusion .....	75
<b>5 Optimization under Uncertainty .....</b>	<b>77</b>
5.1 Introduction .....	77
5.2 Stochastic programming .....	77
5.2.1 Types of SP with recourse.....	78
5.2.2 Advantages and disadvantages of SP .....	79
5.3 Uncertainty Modeling .....	80
5.4 Scenario reduction algorithm.....	83
5.5 Applications.....	87
<b>6 Selected Applications of PSO in Power Systems.....</b>	<b>89</b>
6.1 Introduction .....	89
6.2 Reactive Power Management in Offshore Wind Farms .....	89



6.2.1 Offshore Wind Farm .....	90
6.2.2 Reactive Power Capability of Wind Energy System.....	91
6.2.3 Grid Requirements .....	92
6.2.4 Reactive Power Dispatch Problem .....	92
6.2.5 Solution Procedure .....	95
6.2.6 Test Network .....	96
6.2.7 Results .....	97
6.2.8 Conclusion.....	100
6.3. Optimal Operation of a Wind-Thermal Power System.....	101
6.3.1 Introduction .....	101
6.3.2 Wind Power Forecast Methodology .....	103
6.3.3 Uncertainty Modeling .....	108
6.3.4 Problem Formulation.....	109
6.3.5 Case Study.....	111
6.3.6 Conclusion.....	123
6.3 Hybrid Power Systems for Residential Loads .....	124
6.3.1 Introduction .....	124
6.3.2 Problem formulation.....	125
6.3.3 Scenarios .....	129
6.3.4 Solution procedure .....	131
6.3.5 Numerical results.....	131
6.3.6 Conclusion.....	133
<b>7 Conclusions .....</b>	<b>135</b>
7.1 Summary .....	135
7.1.1 Optimization algorithm .....	135
7.1.2 Power system operation.....	136
7.1.3 Uncertainty modelling .....	136
7.1.4 Optimization under uncertainty .....	137
<b>References .....</b>	<b>139</b>
<b>Resume .....</b>	<b>147</b>
<b>List of Publications .....</b>	<b>149</b>



# Chapter 1

## Introduction

### 1.1 Motivation

Optimization problems in power systems are usually huge, complex and highly nonlinear. They are usually formulated as discontinuous and multi-modal problems and have to be solved in continuous, discrete, combinatorial or mixed integer parameter space. In some instances the optimization problem cannot be mathematically modeled and there is no information about the gradient or derivatives except the objective function value. The gradient search optimization methods are efficient for continuous and uni-modal problem but cannot handle complex multi-modal problems. These optimization techniques have a deterministic search procedure and are highly dependent on the starting operating point. Moreover, in order to simulate realistic power systems, the optimization problems need to be solved without considerable reduction or approximation of the models. In practical application, aspects of easy adaptation and extension of the optimization algorithm are of particular importance. This is made possible with the evolution of new robust computational intelligence tools which use only the evaluations of objective and constraint functions and has no obligation on the characteristic of these functions.

Evolutionary algorithms (EAs) are population based stochastic global search algorithms. They provide robust solutions to highly complex optimization problems in engineering with minimum human effort. They ignite the evolutionary process with a population of random individuals (arbitrary operating points) representing the potential solutions to the given problem. The quality of the individuals is evaluated by the fitness function. Each evolutionary algorithm has its own distinctive search procedure to refine the solution during the iterative process. The credibility of the algorithms depends on its global exploration of the search space and local exploitation of the optimal solution space. EAs like Particle Swarm Optimization, 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.

Despite the huge success of the global search algorithms in tracking good solution to real-world applications, they are subjected to rigorous parameter tuning. These parameters which influence the search procedure are problem dependent. Most EAs also has the demerit of premature convergence. So the converged solution may not even be a local minimum. The selection of optimal population size influences the quality of the solution and computational time. The constrained optimization problems require the selection of appropriate penalty function and well

tuned penalty coefficients. The algorithms are therefore problem dependent and require intense trial and error procedure to track the optimal parameters of the algorithm for a specific application.

The above mentioned demerits of evolutionary algorithms have challenged researchers to develop new variants to improve the performance of the EAs. For this special reason, there is much research underway that is aiming to develop a robust algorithm applicable to a wide variety of optimization problems with less human intervention and also reduce the burden of parameter tuning. In this thesis, a parameter free particle swarm optimization algorithm capable of handling complex optimization problems in power systems is presented. PSO developed by Kennedy and Eberhart simulates the social behaviour of fish and birds. The individuals or particles are associated with a certain velocity to explore the search space. The magnitude and direction of the particle's velocity is guided by its own previous flying experience and also by the performance of the best particle in the swarm. PSO has high priority among the class of EAs because of its simple evolutionary procedure, few algorithm dependent parameters, faster convergence and easy adaptability to different problems.

Most decisions in real-time optimizations have to be made under uncertainty i.e. the data regarding all the variables associated with the optimization models is not known with certainty. Solving the optimization models by fixing the uncertain variables with their nominal values may provide poor solutions. These solutions are not robust to perturbations in the uncertain variables. The stochastic nature of the uncertainties can be described by statistical information such as mean and standard deviation from the historical data. The stochasticity of the random variables is described by probabilistic distribution. These continuous distributions are approximated to discrete probabilistic distribution. A large number of scenarios resulting from these distributions are then used to represent the randomness of the uncertainties. Stochastic programming approach can be used to incorporate these scenarios in the optimization model. In these models decisions are made before the uncertainty is disclosed and the corrective actions are taken after the uncertainty is revealed. Since stochastic models generate optimal decisions by minimizing the future consequences, it results in reliable and robust optimal solutions. Solving the optimization model with these huge set of scenarios is computationally expensive. So these bulky scenario set should be reduced with minimum loss of information. The conventional scenario reduction technique use heuristic approach to perform one-to-one comparisons in order to find the scenarios to be deleted. This limits their use for extremely huge number of initial scenarios. This thesis proposes the use of computational intelligence in scenario reduction algorithm so as to liberalize the restrictions posed by the currently available reduction techniques and to reduce the modeling error. Solving the stochastic model involves optimization over a set of scenarios which drastically increases the dimension of the control variables. Therefore an efficient algorithm is required to solve these high dimensional models. Particle swarm optimization algorithms can be a good choice to handle these high dimensional multi-modal problems.

## 1.2 Objective

The main contributions of this thesis are to develop a parameter free particle swarm optimization algorithm and provide robust solutions to various issues in power system planning and operation. The research described herein will address the following objectives:

1. **Develop a robust global searching algorithm:** Real-time power systems optimization problems should be solved without considerable model reduction. This is made possible with the global search algorithms. A new algorithm is therefore necessary to overcome the major drawbacks of evolutionary algorithms and provide robust solutions in any complex search environment.
2. **Penalty functions:** The most efficient and easiest way to handle constraints in optimization problems is by the use of penalty functions. The direction of the search process and thus, the quality of the optimal solution are hugely impacted by these functions. A suitable penalty function has to be chosen in order to solve a particular problem. These penalty functions are associated with numerous user defined coefficients which have to be rigorously tuned to suit the given problem. Therefore, the objective is to develop a general parameter free penalty technique which can handle most optimization problems with out any human intervention.
3. **Unit Commitment Problem:** UC is the most significant optimization task in the operation of the power systems. The complexity of the UC problems grows exponentially to the number of generating units. The currently available UC algorithms suffer from the curse of dimensionality. The increased problem size adversely effects the computational time and the quality of the solutions. So a new UC problem formulation is necessary to reduce the problem size.
4. **Reduce uncertainty modelling error:** Many decisions in power system planning, and operation have to be made without the complete information of the uncertainties such as generator outages, electrical and thermal load, renewable generation capacity fluctuations etc. prevailing in the optimization model. So these uncertainties should be modeled in a suitable form to be used in the model. The evolution of uncertainties is usually represented by a scenario tree. Computational Intelligence techniques have to be introduced in scenario generation and reduction techniques for better uncertainty modeling.
5. **Optimization under uncertainty:** Planning and operation of the power system often require decisions to be made in the presence of uncertain information. All the uncertainties should be considered in the decision making process. Therefore suitable optimization models have to be developed which results in cost optimal and reliable operation of the power system.

### 1.3 Outline

Chapter 2 describes the fundamentals of particle swarm optimization algorithm. The developments by various other authors are explained through the PSO variants. The chapter then proposes the unconstrained adaptive particle swarm optimization algorithm. The performance of the algorithm is tested on various benchmark problems and the results obtained through the probabilistic comparison with other PSO variants are listed in this chapter.

Chapter 3 presents the basic concepts of constrained optimization. A complete description of the various penalty techniques is provided. The chapter then states the main contributions to constrained optimization. First, the improvements made to the self learning penalty function approach. Second, the new proposed rule based penalty technique is introduced. Third, the new penalty techniques are compared with other approaches. The results are compared using the statistical measures and also through the non-parametric Mann-Whitney U hypothesis.

The most significant optimization task in power system operation, the unit commitment problem (UCP) is described in chapter 4. This chapter presents the new proposed duty cycle based UCP with the advanced convergence operators. A standard ten machine system is used to check the performance of this new approach.

Chapter 5 deals with the optimization under uncertainty. Uncertainty modeling using scenario analysis is explained in detail. The new swarm intelligence based scenario reduction algorithm is presented in this chapter. The chapter provides a rich literature on the applications of stochastic programming.

Chapter 6 presents three applications of APSO in power systems. The first application deals with reactive power management in offshore wind farms, the second application describes a two-stage stochastic programming approach for stochastic UCP applied to wind-thermal power plant and the final application is a multi-stage stochastic programming model describing the optimal operation of a residential hybrid power system with unpredictable PV generation and load.

## Chapter 2

### Particle Swarm Optimization

#### 2.1 Introduction

Particle swarm optimization is an efficient evolutionary computational technique developed by Kennedy and Eberhart [1], [2]. Unlike the other population based search algorithms, PSO tracks the optimal solution not by survival of the fittest but by a process motivated by the personal and social behaviour of a flock of birds. Analogous to other optimization algorithms, PSO does not need the gradient information of the objective function or an appropriate operating point for initiating the search process.

PSO has been successfully applied to solve complex global optimization problems. Its simple evolutionary process, less problem dependent parameters and faster convergence properties makes it a leading competitor among all the evolutionary algorithms. Its solution methodology is hardly affected by the dimension, complexity and nonlinearity of the problem. PSO is therefore a smart alternative to solve large scale power system optimization problems such as optimal power flow, power systems planning and operation, reactive power management, controller parameter estimation etc.

PSO performs the search process by a population of particles called a swarm. The particle is characterised by  $D$ -dimensional vector representing the position of the particle in the search space. The position vector represents a potential solution to an optimization problem. During the evolutionary process, the particles traverse the entire solution space with a certain velocity. Each particle is associated with a fitness value evaluated using the objective function at the particle's current position. Each particle memorizes its individual best position encountered by it during its exploration and the swarm remembers the position of the best performer among the population. At each iteration the particles update their position by adding a certain velocity. The velocity of each particle is influenced by its previous velocity, the distance from its individual best position (cognitive) and the distance from the best particle in the swarm (social). A weighted combination of these three parameters gives the new velocity.

The particle therefore appends its previous flying experiences to control the speed and direction of its journey. Apart from its own performances, the particle also interacts with its neighbours and share information regarding their previous experiences. The particle also utilizes this social information to build their future searching trajectory. During the iterative procedure the particles update their velocity so as to stochastically move towards its local and global best positions. The particle therefore tracks the optimal solution by cooperation and competition among the particles in the swarm.

Consider a swarm of  $s$  particles in a  $D$  dimensional search space  $S \subset \Re^D$  designed to minimize a function  $f(\mathbf{x})$ . A particle  $i$  is defined by its current position  $\mathbf{x}_i$ , velocity  $\mathbf{v}_i$  and personal best position  $\mathbf{x}_{pi}$  as shown below:

$$\begin{aligned}\mathbf{x}_i^T &= [x_{i1}, x_{i2}, \dots, x_{iD}] \\ \mathbf{v}_i^T &= [v_{i1}, v_{i2}, \dots, v_{iD}] \\ \mathbf{x}_{pi}^T &= [x_{pi1}, x_{pi2}, \dots, x_{piD}]\end{aligned}\tag{2.1}$$

The global best position in the swarm is considered to be  $\mathbf{x}_g$ . The standard PSO update equations for each particle in the swarm at iteration  $t$  are as given below:

$$\mathbf{v}_i(t+1) = w\mathbf{x}_i(t) + c_1r_1(\mathbf{x}_p(t) - \mathbf{x}_i(t)) + c_2r_2(\mathbf{x}_g(t) - \mathbf{x}_i(t))\tag{2.2}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)\tag{2.3}$$

where  $\mathbf{x}_g \in \{\mathbf{x}_{p0}, \mathbf{x}_{p1}, \dots, \mathbf{x}_{ps}\}$  such that:

$$f(\mathbf{x}_g) = \min(f(\mathbf{x}_{p0}), f(\mathbf{x}_{p1}), \dots, f(\mathbf{x}_{ps}))\tag{2.4}$$

The randomness in the search procedure is introduced by two independent uniform random sequences,  $r_1$  and  $r_2$  in the range (0,1). The weighting coefficients  $c_1$  and  $c_2$  are the acceleration coefficients which control the influence of cognitive and social terms on the particle's velocity. The inertia weight  $w$  regulates the global and local exploration capability of the particles.

## 2.2 Simple PSO Algorithm

**Step 1:** Initialize a random population of particles. Each dimension  $x_{ij}$  of the position vector,  $\mathbf{x}_i$  is generated by a random distribution on the interval  $[x_{\min}, x_{\max}]$ . Each coordinate of velocity vector,  $\mathbf{v}_i$  is similarly initialized on the interval  $[v_{\min}, v_{\max}]$ . Where  $v_{\max} = k \cdot x_{\max}$  and  $v_{\min} = x_{\min}$  and  $0.1 \leq k \leq 1.0$ . The individual best position vector  $\mathbf{x}_{pi}$  replicates the initial position vector,  $\mathbf{x}_i$ .

**Step 2:** Evaluate the fitness,  $f(\mathbf{x}_i)$  of each particle.

**Step 3:** Update the local best position,  $\mathbf{x}_{pi}$  as shown below:

$$\begin{aligned}\mathbf{x}_{pi} &:= \mathbf{x}_{pi} \text{ if } f(\mathbf{x}_i) \geq f(\mathbf{x}_{pi}) \\ &= \mathbf{x}_i \text{ if } f(\mathbf{x}_i) \leq f(\mathbf{x}_{pi})\end{aligned}\tag{2.5}$$



The global best,  $\mathbf{x}_g$  is estimated by (2.4).

**Step 4:** Update the velocity and position of the particles according to equations (2.2) and (2.3) respectively.

**Step 5:** Repeat steps 2, 3 and 4 until the stopping criterion is met. The exit condition is usually the maximum number of function evaluations, maximum number of iterations or a tolerance value pertaining to the fitness value.

## 2.3 Algorithm Settings

Like any other evolutionary algorithm, the performance of PSO is also influenced by several settings like parameters and topology. The selection of these settings depends on the optimization problem and greatly affects the convergence behaviour and optimal solution.

### 2.3.1 Population Size

Eberhart and Kennedy [3] have analyzed that PSO requires a smaller population compared to other evolutionary algorithms like Genetic algorithm. Complex multimodal problems require large swarms to fruitfully exploit several promising areas (local minima) and provide reliable global solution. A smaller population in this case would end up in a local minimum. Whereas a larger swarm for simple problems would be computationally expensive. The optimal choice of the swarm size should be a good balance between computational time and reliability [4].

### 2.3.2 Maximum Velocity

This parameter limits the maximum exploration capability of the particles [5]. A large value of  $v_{\max}$  may explode the search space. The particles may unnecessarily exploit the infeasible space and the algorithm may not find a feasible solution. Whereas a smaller value of  $v_{\max}$  will not facilitate global exploration and the particles may prematurely converge on a local minimum. Moreover the inertia weight directly controls the global exploration whereas  $v_{\max}$  can only monitor the exploration indirectly. Therefore a cap of the maximum velocity should be carefully carried out by inertia weight rather than by  $v_{\max}$ .

### 2.3.3 Acceleration coefficients

The acceleration coefficients govern the relative velocity of the particle towards its local and global best position. These parameters have to be tuned based on the complexity of the problem. A suitable constriction factor calculated from these parameters will ensure cyclic behaviour for the particles.

### 2.3.4 Inertia weight

The choice of  $v_{\max}$  to curtail the maximum allowable velocity for a particle is also problem dependent. There is no generic rule to either estimate or control this parameter. This disadvantage has been overcome by using inertia weight [6]. This parameter is so designed that the particles have good balance between global and local exploration. Therefore a linearly decreasing function of iterations between 0.9 and 0.4 is chosen. An initial larger value of inertia weight allows particles to search new promising areas efficiently and a lower value during the termination facilitates fine tuning the optimal solution.

$$w = (w_{\max} - w_{\min}) \frac{(iter_{\max} - iter)}{iter_{\max}} + w_{\min} \quad (2.6)$$

### 2.3.5 Constriction factor

The constriction factor based PSO algorithm proposed by Clerc can converge without using  $v_{\max}$ . The particle's oscillations can be effectively damped using this factor. This phenomenon is well explained in [7], [8] using the eigen value analysis considering the velocity and position update equations as state space equations. The velocity update equation with the constriction factor can be expressed as follows:

$$\mathbf{v}_i(t+1) = \chi (\mathbf{v}_i(t) + c_1 r_1 (\mathbf{x}_{pi}(t) - \mathbf{x}_i(t)) + c_2 r_2 (\mathbf{x}_g(t) - \mathbf{x}_i(t))) \quad (2.7)$$

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \text{ where } \varphi = c_1 + c_2, \varphi > 4 \quad (2.8)$$

The convergence factor will allow the particles to cycle around the randomly defined regions around  $\mathbf{x}_p$  and  $\mathbf{x}_g$ . Regardless of the distance between the two regions, the constriction factor ensures that PSO can efficiently exploit the defined space and can finally converge.

### 2.3.6 Neighbourhood Topology

In the standard PSO algorithm described by equations (2.2) and (2.3), each particle is completely informed about the performance of all other particles in the swarm. The particle is attracted towards the global best performer in the swarm. Another set of algorithms have topologies where each particle has access only to a certain set of neighbours such as ring, wheel topologies [9], [10]. Several other network structures are proposed to solve different set of problems. The choice of the topology is problem dependent and there is no general network which is suitable for all classes of optimization problems. The stability and the convergence properties of the PSO algorithm are proved in [11].

## 2.4 PSO Versions

### 2.4.1 Binary PSO

The standard PSO was designed to handle real valued vector space. However PSO can easily be adapted to handle binary space with ease. There are several applications like unit commitment, lot sizing problem etc. where the control variables can be either YES or NO represented by the binary bits 1 and 0 respectively. In binary PSO, the particle is modelled to represent the binary bits. The velocity,  $v_{id}$  represents the probability of flipping the bit,  $x_{id}$ . The velocity update equation in (2.2) remains unchanged except that  $\mathbf{x}_p$  and  $\mathbf{x}_g$  are vectors of binary bits. The velocity is transformed to probability by using the sigmoid function [12].

$$s(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \quad (2.9)$$

The maximum allowable velocity,  $v_{\max}$  is used to control the mutation of the binary bits.  $v_{\max}$  is usually set to 4.0. Curtailing the velocity to  $[-4.0, 4.0]$  means that the probability,  $s(v_{id})$  is limited between 0.982 to 0.018. Unlike real coded PSO where high  $v_{\max}$  allows increased exploration, in binary PSO high  $v_{\max}$  allows a lower rate of altering the binary bits. The modified position update equation can be expressed as:

$$\begin{aligned} x_{id} &= 1, \text{ if } \text{rand}() < s(v_{id}) \\ x_{id} &= 0, \text{ otherwise} \end{aligned} \quad (2.10)$$

### **2.4.2 Discrete PSO**

PSO can be easily adaptable to solve mixed inter optimization problems [13]-[15]. The particle can be modelled to accommodate binary, integer and continuous control variables. The evolution and update strategy is very similar to the standard PSO except that after the update process, the position,  $x_{id}$  corresponding to the integer variables is ceiled to the nearest integer. The rounding can be deterministic or probabilistic process. In deterministic process,  $x_{id}$  is rounded based on its distance to its adjoining integers whereas in probabilistic rounding, a probability function governs the ceiling process.

### **2.4.3 Adaptive PSO**

The particles flying trajectory and convergence rate can be optimally defined by rigorously tuning the inertia weight, acceleration coefficients and constriction factor. The amplitude and damping rate of the particle's oscillations during the iterative process can be controlled by these parameters. But a deterministic set of these parameters for the entire search process may lead to bad results. For example, if the population of particles is concentrated near a local minimum, the velocity guided by its own performance and global best particle is not sufficient enough to support exploration. The particles therefore converge on the best position discovered so far by the swarm which might not be even a local minimum. Such dangerous situations can be avoided by energizing the particles by giving them enough velocity to explore. Several adaptive strategies were proposed to dynamically adjust the particles trajectories during the evolutionary process and improve the solution. Three such variants adopting the fuzzy control for dynamically adjusting the velocity and inertia weight are describes below.

The parameters can be dynamically adjusted during the iterative process as per the requirements of the particles. The fuzzy logic controller is a tool that generates directive control action from a given set of inputs through a strong knowledge base generated by simple logical operators. The knowledge base is built based on the experience gained during the trial and error methods for parameter tuning.

#### **2.4.3.1 Fuzzy Inertia weight**

Shi and Eberhart [16] proposed a fuzzy controller to estimate the changes in the inertia weight. The inputs to the fuzzy controller are the current inertia weight and the normalized fitness corresponding to the global best position. Three membership functions are used to define how the input space is mapped to the three fuzzy sets (low, medium, high). The output of the controller predicts the new inertia weight.

The results indicate that the fuzzy control is very effective for unimodal problems. The controller suggests a larger inertia weight until the particle traces a good solution space. Once the particle finds the optimal space, the controller automatically reduces the inertia weight to fine tune the optimal space. Adjusting the inertia weight based on the particles performance rather than by iteration number will remove unwanted iterative steps and hence assert faster convergence. However for multimodal problems, the controller may identify the solution space close to a local minimum but has no information whether the traced local minimum is also the global optimum. The particles therefore may be trapped in a local minimum.

#### 2.4.3.2 Fuzzy Velocity

In [17], the authors proposed an adaptive fuzzy control to dynamically control the velocity of the particles. The particles velocity is scrutinized at every iteration and the poor performing particles are motivated to search better and compete with the best particles. If the velocity is less than a predefined threshold, the particle is given a sudden boost determined by a set of fuzzy rules.

$$v_{id}^{new} = \begin{cases} v_{id} & \text{if } |v_{id}| \geq v_c \\ \text{rand}(-1,1) * v_{\max} / \rho & \text{if } |v_{id}| < v_c \end{cases} \quad (2.11)$$

$v_c$  is the threshold velocity and  $\rho$  is the scaling factor which determines the magnitude of the turbulence. A large  $v_c$  will rejuvenate the particles to discover new promising areas and a smaller value allows a local fine search. A small value of  $\rho$  increase the amplitude of the particle's oscillations and this might help the particle to jump even a strong local minimum. The entire search process is therefore divided into three stages. In the first stage,  $v_c$  and  $\rho$  are set at large and small values respectively to ensure that the particles identify the best local minimum.  $v_c$  and  $\rho$  are set at medium values in the second stage and in the final stage  $v_c$  is small and  $\rho$  is large. In the second and third stages, the swarm will exploit the discovered optimal space. These characteristics are used to frame the rules of a fuzzy controller. At every iteration, the controller monitors the particle's personal performance and velocity to decide the threshold and scaling factor.

#### 2.4.3.3 Adaptive Constriction factor

In this algorithm, the particle's dynamics and convergence are controlled by optimally adjusting the constriction factor during the search process. In the standard PSO, the constriction factor is constant throughout the search process. In [18], the authors proposed an adaptive constriction factor to enhance the performance of PSO to complex multi-modal problems. The influence of social and cognitive pa-

rameters on the particle's velocity can be controlled by carefully regulating the constriction factor. This adaptive policy results in faster convergence. A special index called location related ratio (LR) is defined for each dimension of the particle. LR is a normalized distance of the particle from a core position (average performance of all the particles in the swarm). The particles tries to adjust their velocity based on their average distance from the other particles in the swarm. This avoids premature convergence and also regulates velocity for effective global exploration and local exploitation.

#### **2.4.4 Multiobjective PSO**

Multiobjective optimization problems have two or more objectives to be optimized simultaneously. The Pareto front concept describes the optimal trade off possibilities between the objectives. A potential solution on the Pareto front cannot improve any objectives without degrading at least one of the other objectives. The algorithms for multiobjective optimization problems have to identify the true Pareto front. Evolutionary algorithms and especially PSO can effectively explore different parts of the Pareto front simultaneously. In PSO all the particles are concentrated or directed towards the global best particle in the swarm. Hence PSO can not find multiple points on the Pareto front. The global best and the local best have to be intelligently selected to let the swarm discover different regions of the front.

Hu and Eberhart [19] proposed a dynamic neighbourhood PSO where each particle selects a set of neighbours based on one of the objective. Then a local best is selected from this neighbourhood based upon the other objective. This local performer replaces the global best in the velocity update equations. Although this method is applicable to a wide variety of multiobjective problems, it is strictly restricted to a two dimension function space.

In [20], [21] Coella suggested another approach to incorporate pareto ranking scheme in PSO. The idea is to archive the non-dominant solution found by the particles in the past and use this information in the global centring mechanism of PSO to promote convergence towards the most optimal non-dominant solutions. Each particle chooses a different leader based on the historical data stored in the repository to guide its search trajectory. Furthermore the already explored function space is partitioned into hypercubes. Each hypercube is assigned a fitness based on the number of particles in it. A particle in the less crowded hypercube is more likely to be selected as a leader of a particle. This facilitates the swarm to explore unrepresented areas of the Pareto front.

## **2.5 Applications**

Particle Swarm Optimization outsmarts the other evolutionary algorithms due to its simple search process, few tuneable parameters and its ability to rapidly dis-

cover good solutions. PSO can be an effective tool when the optimization problem can not be mathematically modelled; no expert knowledge is available to exactly define the problem space or when the objective function is complex, non-linear and high-dimensional. These special features enable PSO to be applied to a variety of applications ranging from engineering design, process optimization, to service oriented applications in finance, healthcare, bioinformatics and entertainment. In medical field PSO is used in edge detection of medical images, optimize drug formulas, and to identify the right sample in medical diagnostic tests involving huge samples. With the rise of AI based computer games PSO is used to generate smarter computer players; create artificial human-like interaction to a human player by strategically adapting the AI and to improve video and sound quality. In Robotics, PSO can be used to generate fast motion trajectories for robots with high degree of freedom. In dynamic environment, robots store information about the changing topologies and obstructions. PSO is used to decide the shortest path in the ever changing environment. It can also be used machine- learning applications, including classification and object prediction. In financial sector PSO is used for optimal asset allocation, index tracking, risk analysis and search for the optimal trading indicators and rules for the stock market using high frequency data. The finance models and bidding strategies can also be improved. PSO has also been used to model immune systems, ecological and social systems.

In power systems PSO has been successfully applied to optimal power flow, unit commitment problem, economic dispatch, reactive power and voltage control, generation expansion planning, reliability assessment, controller design, machine modelling, neural network training and forecast models [22], [23].

## **2.6 Novel Particle Swarm Optimization Algorithm**

### ***2.6.1 Introduction***

The advent of new evolutionary based global search algorithms has almost filled the vacuum created by the conventional optimization techniques in solving large complex problems. These stochastic algorithms are capable of solving a wide variety of real-world problems which may or may not have an explicit description. However, they too have some limitations. They have several problem dependent parameters which need to be rigorously tuned to obtain good solutions. For example, the population size will depend on the complexity of the problem. As such, there is no standard fixed swarm size which suits a range of problems. The iterative search process ensures that all particles converge at the global optima. Extra energy in the form of iterations is devoted to improve the bad performing particles. What is the contribution of these bad particles in searching the optimal space? The particles in the swarm may have different exploration capabilities. How fair is it to have the same flying strategies for both good and bad performing

particle. The bad particle may call for additional iterations to converge. There are several adaptive algorithms which adapt the inertia weight, cognitive and social parameters with the iterative search process. It is a common practice to have high inertia weight to rejuvenate the particles. This helps the particles to explore new areas efficiently. What will happen to the swarm, if no good region is discovered during this stage? The particles will be exhausted and may only facilitate a local search. The swarm may end up with no good solution. The selection of local and global best parameters also involves computational effort. The user should either have expert knowledge on the problem being solved or repeat the optimization process several times to decide on the right set of these parameters. The standard PSO velocity update strategy demands all particles in the swarm to follow the global best performer. The best performer might catch up local minima and all the particles might follow its path and therefore the whole swarm might converge prematurely on this local minimum. There are many such challenging issues which have to be addressed to take the evolutionary algorithm to the next level. Most of these issues are addressed in the proposed version of PSO.

### ***2.6.2 Who can benefit?***

Almost any optimization problem can be solved by the proposed algorithm without much human intervention. One has to define the right set of decision variables represented as a particle and a suitable measure (fitness function) to compare the relative performance of the particles. The ability to discover good solutions at a rapid pace is the key feature for its success. The algorithm can be easily adapted to solve any new application. The new version of PSO is a good alternative to other optimization techniques when:

- (1) Search space is complex, nonlinear, non-differentiable and multimodal
- (2) Search space is widely spread
- (3) No expert knowledge or mathematical analysis of the problem is available
- (4) Problem space or objective function can not be approximated
- (5) Applications can return only the numerical evaluations for objective function and constraint violations and no other information is available.

### ***2.6.3 The solution strategy***

The motive behind the new version of PSO is to develop a black box optimization tool. The algorithm should involve minimum human assistance and should provide good solutions to a wide variety of real-time applications. PSO should be used like a built in function call. The function should only have inputs like problem dimension, lower and upper bounds of the decision variables, number of equality and inequality constraints and a stopping criteria. This function



call should be associated with an external function which can take an instance of decision variables and return the corresponding fitness value and constraint violations. No other information about the problem can be exchanged. The particles should be able to adjust their flight and the swarm should maintain a suitable population diversity to automatically self tune their search process and discover good solutions.

The above requirements are fulfilled by the new version of PSO called Self-adaptive particle swarm optimization, here after referred to as APSO. It is a parameter free optimization tool. 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 the global optimal solution. It is free 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". The basic structure of the algorithm is derived from the TRIBE-PSO introduced by Maurice Clerc [24].

### 2.6.3.1 Swarm Evolution

The search process is ignited by minimal set of  $N_T$  tribes. Each consisting of fixed set of particles,  $N_p$ . Each particle is associated with a certain velocity and fitness. The particles try to memorize its previous two performances and also its best performance. At the end of  $N_G$  generations, the tribes are evaluated. The particle is judged based on its two previous performances. The performances can be an improvement (+), status quo (=) or a deterioration (-). A bad particle is one which deteriorates or shows no progress(--, - =, = -, +-). On the other hand, a good particle is one whose performances are improvements(=+, ++, +=). The TRIBE is also labelled as good or bad based on the majority of its good or bad particles. At the first iteration, the previous two performances of the particles are initialized to their current position. If the TRIBE happens to be a bad performer, it indicates that its current information about the search space is not enough to find good solution. At this instant, this tribe will add more information by generating a new tribe with  $N_p$  particles. Two-third of the new particles are randomly generated while the remaining one-third particles are generated in the close proximity of the best particle in the current TRIBE. The second bad TRIBE will add another  $N_p$  particles to the newly generated TRIBE. Whereas the good TRIBE has majority of good particles. It means that the TRIBE has enough information about the good solution. If this TRIBE has more than one particle, the worst performing particle is identified. This particle may also be good but its close associates in the same TRIBE are much better and this particle has the same or less information as the rest of its associates. So there is no risk in deleting this particle. The good TRIBE therefore will eliminate one its least performing particle. The updated swarm is

again allowed to explore for  $N_G$  iterations. This process continues until the stopping criteria is reached. The process of evolution indicates that new particles will be born only when they are required. Particles which do not contribute to the search process are eliminated. The swarm always has potential particles enthusiastic enough to search for an optimal solution. This substantially helps the algorithm to find the solution within few iterations.

### 2.6.3.2 Flying Strategies

Different particles in the TRIBEs have different levels of performances. Before each TRIBE evaluation, the particles are given enough time to explore. During the evaluation, the particles are compared based on their performance and least performing particles may be eliminated. But, are all the particles given a fair chance to improve? Both good and bad particles are allowed to explore for the same amount of time. Since all particles have the same flying strategies, the good particles will always perform better and the bad particle will never reach the standards of the good particle. In order to remove this bias, different particles have different flying strategies. Based on their previous performances, the particles automatically judge the right flying strategy. The particles are categorized into three groups. A worst particle is one whose performances are deterioration (--,=-). Bad particles and good particles comprise the following combinations (-=, ==, +-), (+=, =+, -+, ++), respectively. The worst particles follow a random search strategy, the bad particle prefers pivot strategy and the good particles follow Gaussian update strategy. These strategies are explained below.

#### Random Strategy

The flight of the particles totally depends on either the local best or the global best. Two procedures are randomly chosen in this strategy. In the first procedure distance of the local best,  $p$  from the current position and also the global best;  $g$  is calculated. The larger of the two distances serves as the radius of the hypersphere generated around the local best. A position vector is randomly generated on this hypersphere ( $H_p$ ) and this when added to the current position results in the updated position vector.

$$x_{id}^{k+1} = \text{rand}(H_p) + x_{pbest,d}^k \quad (2.12)$$

Similarly the second procedure identifies the maximum of the two distances of global best to local best and current position. With this as the radius and centre at the global best position, a hypersphere ( $H_g$ ) is generated. The new updated position is randomly generated on this hypersphere as shown below:

$$x_{id}^{k+1} = \text{rand}(H_g) + x_{gd}^k \quad (2.13)$$

### Pivot Strategy

In this strategy the particles are forced to follow the direction of its individual best performance,  $p$  and the best informer of the particle,  $g$ . Two hyperspheres,  $H_p$  and  $H_g$  with  $p$  and  $g$  as centers and radius equal to the distance between them are defined. A point is randomly chosen in each of these hyperspheres. A weighted combination of these points gives the new position vector,  $\mathbf{x}_i^{k+1}$  of the particle.

$$\mathbf{x}_i^{k+1} = c_1 \text{rand}(H_p) + c_2 \text{rand}(H_g) \quad (2.14)$$

$$c_1 = \frac{f(\mathbf{x}_p)}{f(\mathbf{x}_p) + f(\mathbf{x}_g)} \quad (2.15)$$

$$c_2 = \frac{f(\mathbf{x}_g)}{f(\mathbf{x}_p) + f(\mathbf{x}_g)} \quad (2.16)$$

Where,  $c_1$  and  $c_2$  are the weight factors. These values are calculated using the objective function,  $f$  corresponding to local and global best.

### Gaussian Update Strategy

A set of Gaussian distributions govern the cognitive and social contribution to the particle's velocity. The velocity of the particle  $i$  at dimension  $d$  and iteration  $k+1$  depend on its previous velocity ( $v_{id}^k$ ), its previous best performance ( $x_{pd}$ ) and performance of the best informer ( $x_{gd}$ ) as shown below.

$$\delta_p = x_{pd}^k - x_{id}^k \quad (2.17)$$

$$\delta_g = x_{gd}^k - x_{id}^k \quad (2.18)$$

$$v_{id}^{k+1} = v_{id}^k + \text{gauss\_rand}(\delta_p, |\delta_p|/2) + \text{gauss\_rand}(\delta_g, |\delta_g|/2) \quad (2.19)$$

$$x_{id}^{k+1} = x_{id}^k + \chi(v_{id}^{k+1}) \quad (2.20)$$

Where  $\text{gauss\_rand}(\mu, \sigma)$  generates a normal distributed numbers with mean  $\mu$  and standard deviation  $\sigma$ . During the early stages of the search process the distance of the particle from its local best and global best will be quite high. A Gaussian distribution generated with this distance as the mean and half the distance as the

standard deviation will have a wide spread. The velocity generated by these distributions will be high enough to provide global exploration. When the search process is in the final stages,  $\bar{\sigma}_p$  and  $\bar{\sigma}_g$  are small and therefore the Gaussian distributions have a very small spread. Hence provide only a local search. The Gaussian distributions will therefore provide both global and local exploration around  $p$  and  $g$ . This approach eliminates the use of acceleration coefficients. Hence the update strategy is totally independent of the tuning parameters.

### 2.6.3.3. Neighborhood Topologies

The standard PSO version employs a star topology where each particle is directly connected to the global best performer of the swarm. When the global best performer catches a local minima, all particles are naturally attracted to it. Since each particle is directly connected to the global best, information is rapidly propagated and the whole swarm may prematurely converge on this local minima. In order to avoid such untoward convergence several information or neighborhood topologies are suggested. In APSO algorithm each particle has an evolving neighborhood. The flight of the particle is consistently monitored and guided by these neighbors. The global best performer is no more common to all the particles. Each particle selects its own global best performers from its neighborhood and not from the whole swarm. The neighborhood of a particle in a TRIBE consists of all its contemporaries in that particular TRIBE and also its parent particle (particle from which it is born). Initially all particles have random parents. When a particle dies due to ill performance, its presence in all neighborhoods is replaced by its best performer. The information regarding the global best solution discovered by the swarm is propagated to all particles. However it takes several iterations to spread this information. During this process the particles will have enough time to explore and easily escape the local minima suggested by the global performer. However this process is at the cost of convergence. The swarm with this topology will be able to avoid premature convergence but the speed of convergence will be less than the swarm with ring topology.

### 2.6.4 Description of the Proposed Approach

The optimization process starts with  $N_T$  tribes and eventually evolves to explore the entire problem space. Each particle in a tribe is assisted by a set of associates in its neighborhood. Each tribe will try to locate a minimum and in the process also communicate with the other tribes to discover the global solution. The algorithm consists of two iterative loops. One loop controls the ultimate termination of the search process. The second loop allows the swarm to explore and exchange information among its neighbors before they are finally evaluated. So this loop con-

trols the evolutionary process. A fixed set of generation,  $N_G(=10)$  is set as the termination criteria for this loop. The algorithm is explained in the following steps:

1) Swarm initialization

FOR  $I=1$  to  $N_T$

FOR  $j=1$  to  $TOTAL\_PARTICLES\_TRIBE[i]$

- (a) Randomly generate a particle with position,  $\mathbf{x}$  and velocity,  $\mathbf{v}$
- (b) Assign a fitness value,  $f(\mathbf{x})$  for each particle
- (c) Initialize the local best ( $\mathbf{x}_p$ ) and previous two performances ( $\mathbf{x}_{p1}$ ,  $\mathbf{x}_{p2}$ ) to the current position,  $\mathbf{x}$ .
- (d) Generate a neighborhood topology for the particle. The neighborhood list of a particle consists of all the particles in the current tribe and a random parent from the other tribe.

END

END

2) WHILE ( $CYCLE < N_G$ )

DO

FOR  $i=1$  to  $TOTAL\_TRIBES$

FOR  $j=1$  to  $TOTAL\_PARTICLES\_TRIBE[i]$

- (a) choose the right flying strategy and update velocity and position
- (b) correct the *BOUND* violations
- (c) evaluate the fitness
- (d) memorize  $\mathbf{x}_{p1}$ ,  $\mathbf{x}_{p2}$
- (e) update  $\mathbf{x}_p$ ,  $\mathbf{x}_g$
- (f)  $COUNT++$
- (g) IF ( $COUNT > MAX-FUNCTION\_EVALUATIONS$ ) GOTO STEP 7

END

END

$CYCLE++$

END WHILE

4) Evaluate the *TRIBES* and perform the necessary adaptations

FOR  $i=1$  to  $TOTAL\_TRIBES$

- (a) Evaluate the *TRIBE* (*GOOD* or *BAD*)

- (b) IF ( $TRIBE == BAD$ )

- (i) Identify the best performer in the *TRIBE*

- (ii)  $TRIBE\_COUNT = TOTAL\_TRIBES + 1$ , for the first *BAD TRIBE* only

- (iii) Generate a new *TRIBE* with  $N_p$  particles

- Two-third particles are randomly generated
- One-third particles are generated close to best performer of  $TRIBE[i]$

- (iv) Generate the neighborhood topologies for these particles. The neighbors of a particle include all the particles in the new *TRIBE* and the best performer of the current *BAD TRIBE*
- (c) IF (*TRIBE*==*GOOD*)
  - (i) Identify the worst particle and remove it.
  - (ii) Update the neighborhood topologies by replacing the deleted particle with its global best performer.
- CYCLE*=0, *TOTAL\_TRIBES*= *TRIBE\_COUNT*
- 6) GOTO STEP 2
- 7) END

### 2.6.5 Performance of APSO

APSO was developed to solve large power system optimization problems. However it is computationally difficult to validate the algorithm using these applications. Therefore a set of standard test problems commonly used in global optimization are used to investigate the effectiveness of the proposed algorithm. The performance of the proposed adaptive PSO algorithm (APSO) is compared with two other variants of PSO (NPSO and SPSO). PSO algorithm with fixed neighborhood topologies is referred to as NPSO (standard PSO 2006). In this algorithm each particle has a fixed neighborhood size. The second variant is the basic PSO algorithm with star neighborhood i.e. each particle is connected to every other particle in the swarm. The parameters for NPSO and SPSO are optimally tuned to obtain quality solutions. The particles which violate the bounds are made to stay on the boundary and their corresponding velocity is initialized to zero. The acceleration coefficients are set to 1.5 each and inertia weight is assumed to be a linearly decreasing function from 0.8 to 0.2. A population size of 200 was considered in all simulations.

#### 2.6.5.1 Test Functions

The performance of APSO was investigated on six benchmark functions. The seven functions describe a wide variety of complexities in unconstrained global optimization problems. The functions are all high-dimensional problems. All the applications are formulated as minimization problems and have a unique global optimal solution. These functions are given by:

1. Sphere Function

$$f_1(\mathbf{x}) = \sum_{i=1}^{N_d} x_i^2 \quad (2.21)$$

where  $\mathbf{x}^* = (0)$  and  $f_1(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

## 2. Step Function

$$f_2(\mathbf{x}) = \sum_{i=1}^{N_d} (\lfloor x_i + 0.5 \rfloor)^2 \quad (2.22)$$

where  $\mathbf{x}^* = (0)$  and  $f_2(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

## 3. Rosenbrock Function

$$f_3(\mathbf{x}) = \sum_{i=1}^{N_d-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2) \quad (2.23)$$

where  $\mathbf{x}^* = (1, 1, \dots, 1)$  and  $f_3(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

## 4. Rastrigin Function

$$f_4(\mathbf{x}) = \sum_{i=1}^{N_d} (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (2.24)$$

where  $\mathbf{x}^* = (0)$  and  $f_4(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

## 5. Ackley's Function

$$f_5(\mathbf{x}) = -20 \exp \left( -0.2 \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} x_i^2} \right) - \exp \left( \frac{1}{N_d} \sum_{i=1}^{N_d} \cos(2\pi x_i) \right) + 20 + e \quad (2.25)$$

where  $\mathbf{x}^* = (0)$  and  $f_5(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

## 6. Griewank Function

$$f_6(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{N_d} x_i^2 - \prod_{i=1}^{N_d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (2.26)$$

where  $\mathbf{x}^* = (0)$  and  $f_6(\mathbf{x}^*) = 0$  for  $-30 \leq x_i \leq 30$

Functions Sphere and Rosenbrock are unimodal problems with continuous (smooth) landscapes. Step function is also unimodal problem but has discontinuous (flat) landscapes. No gradient information is available for this function and therefore the algorithm may converge on any of the flat surfaces. Rastrigin, Ackley and Griewank functions are multimodal problems whose number of local minima increases exponentially with problem dimension. The complex nature of these functions is shown by the 3D view and contour plots in Fig. 2.1-2.4. The objective of these optimization problems is to find the global optimal solution defined as:

$$\begin{aligned}
&\text{Given } f : \mathfrak{R}^{N_d} \rightarrow \mathfrak{R} \\
&\text{Find } \mathbf{x}^* \in \mathfrak{R}^{N_d} \text{ such that } f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in \mathfrak{R}^{N_d}
\end{aligned} \tag{2.27}$$

The performance is analyzed in terms of convergence rate, convergence reliability and the quality of final solution. The convergence rate indicates the reduction of the distance to the final global solution over time. Convergence reliability shows the ability of the algorithm to repeat final solutions within  $\varepsilon$ -tolerance. The dimension of all problems is set to 20. Each function minimization problem is simulated 250 times and the resulting best final solution, worst solution and the mean and standard deviation of all simulations are reported in Table 2.1. The  $\varepsilon$ -tolerance is assumed to be  $10^{-4}$ . The algorithm terminates when the target value or the maximum number of function evaluations (60,000) is reached. The function evaluations required by all the three PSO algorithms for obtaining the best and worst solution along with the mean evaluations required to converge for the 250 simulations are listed in Table 2.2. It is clear from the results that all the algorithms performed well for the Sphere and Step functions. For Rosenbrock function, APSO produced far better results than the other two algorithms. SPSO was able to obtain the best solution of 0.0545 but the average and standard deviation of optimal solutions over several simulations was very far from that obtained by APSO. NPSO was the best performer on Griewank function with a mean value of 0.0053 and standard deviation of 0.0090. The tabulated results do not give a clear picture about the performance of the algorithms. Moreover the results obtained by the 250 simulation might have occurred by chance. So the results should be standardized. It means that the solutions obtained during the various simulations should result from a distribution. Another simulation of the algorithm on the considered test function should produce a result corresponding to this distribution.

Performing a new simulation is similar to sampling the distribution. The algorithms can be compared only when the simulation results follow similar probability distributions. Lilliefors test is performed on the results to verify the goodness of fit to normal distribution. All the algorithms results are observed to fit a normal distribution. Algorithm A is considered to perform better than algorithm B only if the probability of a solution obtained by A is greater than the solution obtained by B.

The results obtained by algorithm A represents a random variable  $X$  whose probability distribution is given by  $D_A$ . Similarly results by algorithm B are represented by random variable  $Y$  and distribution  $D_B$ . A new simulation of the algorithm is similar to drawing a random number from their corresponding distribution



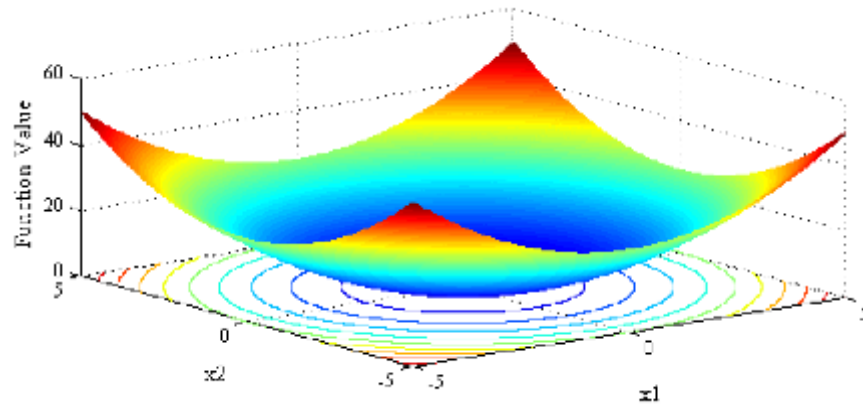


Fig. 2.1 3-D view and contour plots of the sphere function

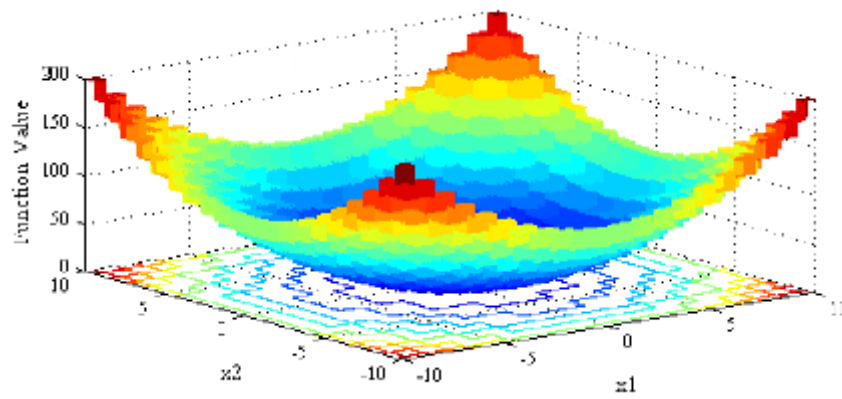


Fig. 2.2 3-D view and contour plots of the step function

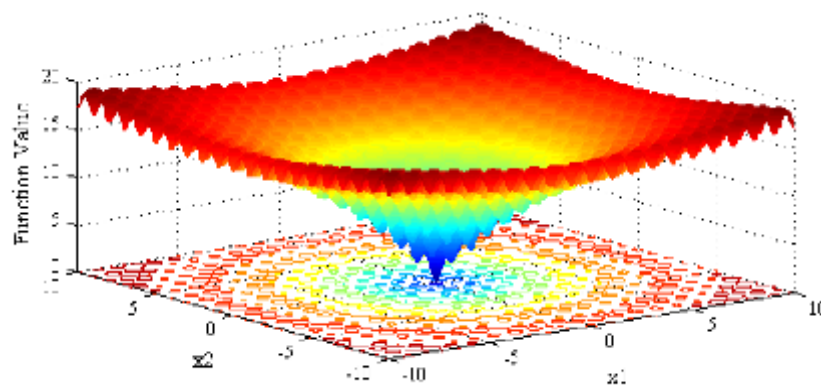
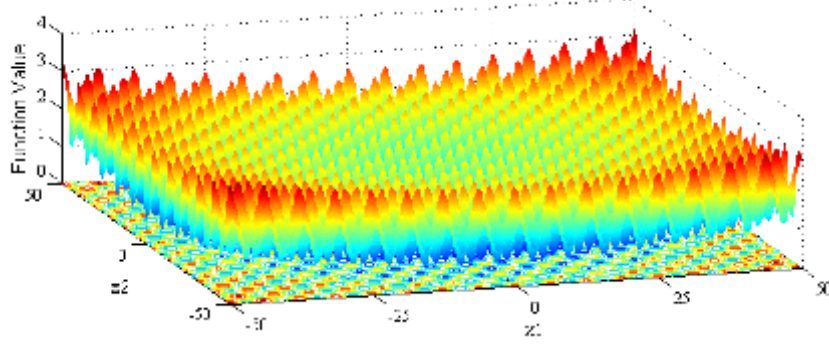


Fig. 2.3 3-D view and contour plots of the Ackley function



**Fig. 2.4** 3-D view and contour plots of the Griewank function

The probability is given by:

$$\Pr[y \leq Y] = \int_y^{+\infty} D_B(v) dv \quad \forall y \in \mathfrak{R} \quad (2.28)$$

In order to compare the two algorithms we have to integrate the above probability over all possible values of random variable  $X$ , weighted by its probability density. The resulting formula is as follows:

$$\Pr[X \prec Y] = \int_{-\infty}^{+\infty} \left( \int_X^{+\infty} D_B(v) dv \right) D_A(x) dx \quad (2.29)$$

The probabilities obtained by comparing APSO with NPSO and SPSO for different set of benchmark problems are listed in Table 2.3. The second column compares the performance of APSO with NPSO. APSO performs better than NPSO for Rosenbrock and Rastrigin functions. But for Griewank and Ackley function NPSO performs better. The average probability of APSO compared to NPSO is

**Table 2.3** Probability comparison of APSO with the other two PSO variants

function	APSO better than NPSO	APSO better than SPSO	NPSO better than SPSO
Sphere	0.5	0.5	0.5
Step	0.5	0.5	0.5
Rosenbrock	0.8271	0.7164	0.434
Rastrigin	0.8450	0.7848	0.2929
Griewank	0.2231	0.3877	0.7171
Ackley	0.4601	0.5278	0.5399
Average	0.5888	0.604	0.4959

given in the last row. This probability indicates that APSO performs better than NPSO by nearly 59% over all considered test functions. Similarly APSO outsmarts SPSO by 60%.

The effectiveness of the algorithm with increased problem dimension is observed with respect to one of the test problem (Ackley function). The average function value and average function evaluations for the algorithms are shown in Fig. 2.5 and 2.6. The results indicate that with the increase in problem dimension, the solution complexity increases, quality of the solution are degraded and the algorithm require high function evaluations to converge. But the increase is prominent in NPSO and SPSO compared to APSO. The quality of the average function value for APSO is similar to NPSO but the number of function evaluations increases drastically in NPSO compared to APSO. The convergence properties of the algorithms for Griewank function are shown in Fig. 2.7. It is evident that APSO convergences much faster than the other two variants. The convergence

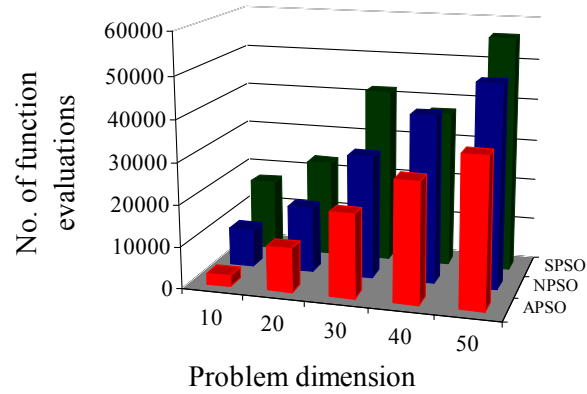


Fig. 2.5 Performance of the algorithms with the increase in problem size

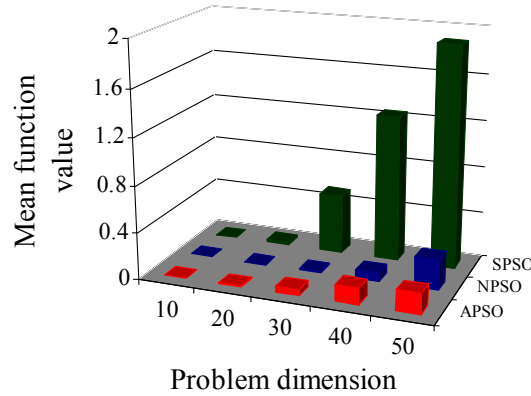
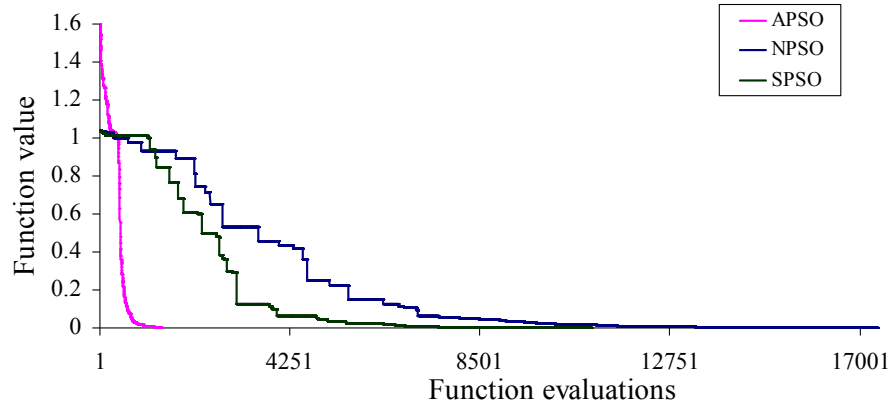
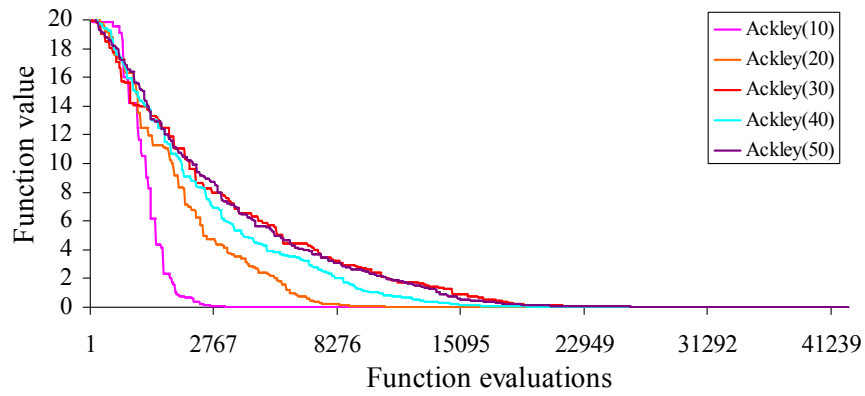


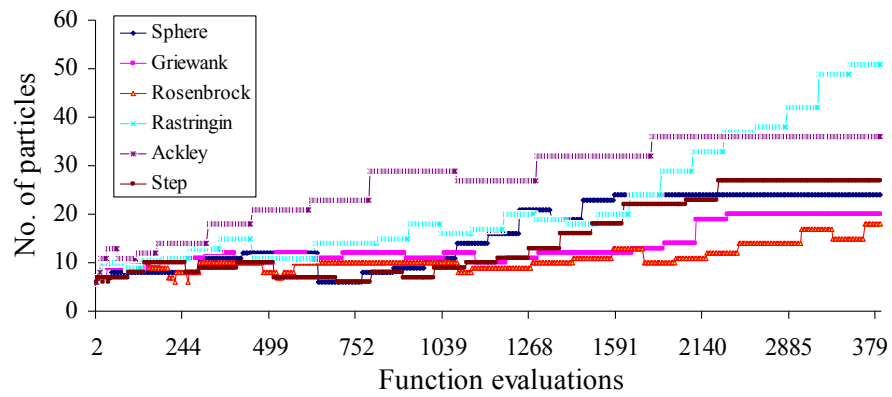
Fig. 2.6 Performance of the algorithms with the increase in problem size



**Fig. 2.7** Convergence of the algorithms for Griewank function



**Fig. 2.8** Convergence of the APSO with increased problem size for Ackley function



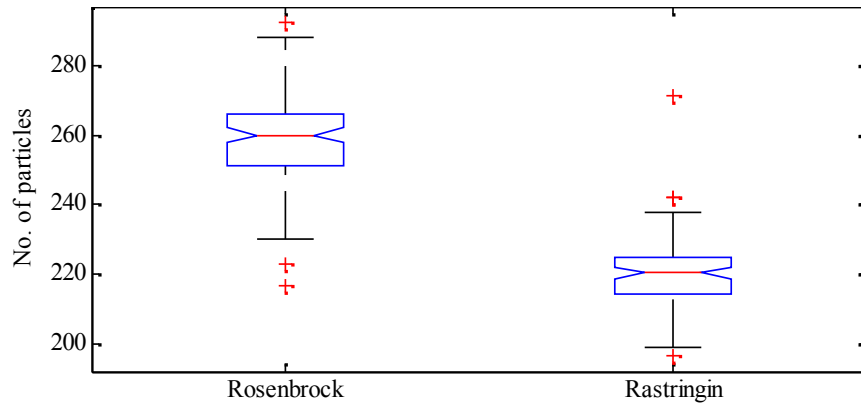
**Fig. 2.9** Evolution of the swarm for different test problems

**Table 2.1** Performance comparison of the three PSO variants with respect to final function values

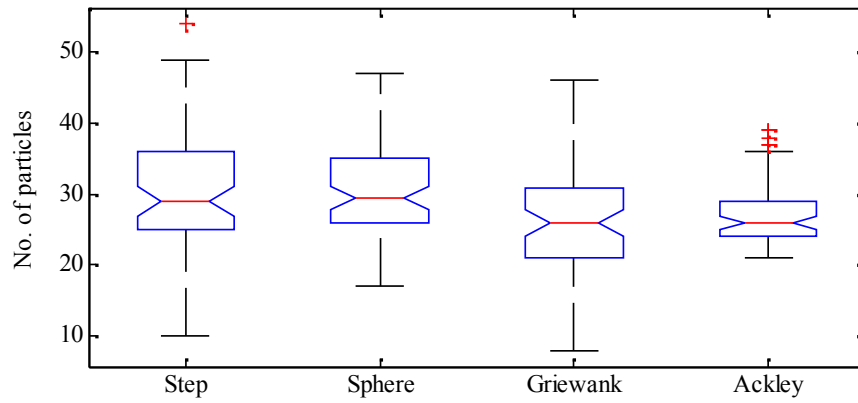
Problem	Best Result			Worst Result			Mean Result			STD		
	APSO	NPSO	SPSO	APSO	NPSO	SPSO	APSO	NPSO	SPSO	APSO	NPSO	SPSO
f1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f3	0.0064	12.2513	0.0545	9.8149	99.09	162.3282	4.4244	17.077	13.6615	3.049	13.068	15.8508
f4	6.9717	9.3454	8.9546	29.9485	98.66	60.6923	20.653	43.61	30.44663	6.407	21.690	10.636
f5	0.0	0.0	0.0	0.9067	0.0	1.6462	0.0085	0.0	0.02161	0.077	0.0	0.1710
f6	0.0	0.0	0.0	0.2943	0.055	0.2446	0.0373	0.005	0.0229	0.041	0.009	0.0293

**Table 2.2** Performance comparison of the three PSO variants with respect to function evaluations

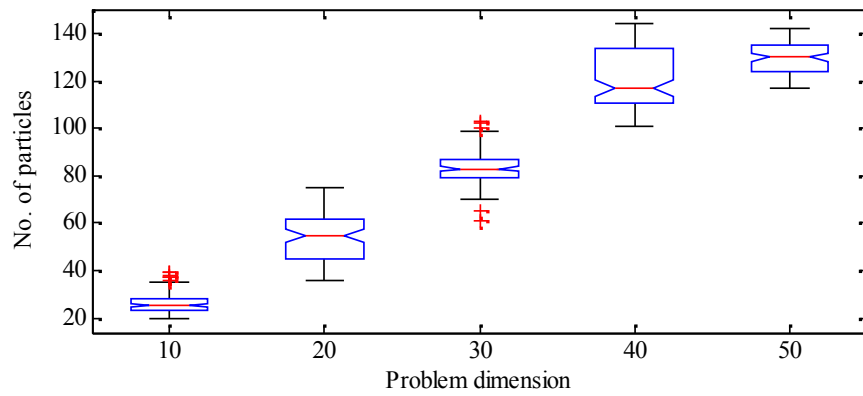
Problem	Best Result			Worst Result			Mean Result		
	APSO	NPSO	SPSO	APSO	NPSO	SPSO	APSO	NPSO	SPSO
f1	4428	9900	13400	7373	11900	16600	5767	10932	14875
f2	4917	2400	2400	7213	5250	7200	5745	3995	3922
f3	60000	60000	60000	60000	60000	60000	60000	60000	60000
f4	60000	60000	60000	60000	60000	60000	60000	60000	60000
f5	10684	15850	23200	60000	18950	60000	24845	17066	25725
f6	3865	15300	12250	10415	60000	60000	8732	45637	451728



**Fig. 2.10** Average swarm size required for rosenbrock and Rastrigin test functions



**Fig. 2.11** Average swarm size required for Step, Sphere, Griewank and Ackley test functions



**Fig. 2.12** Average swarm size required for different dimensionality of Ackley test function

properties of APSO with increase in problem dimension are shown in Fig. 2.8. The problem dimension has a greater impact on the convergence properties of the algorithm. The evolution of the swarm (swarm size) for all test functions is shown

in Fig. 2.9. The swarm adopts different evolutionary strategies for different problems. For Ackley function, the swarm generates a lot of TRIBES at the initial stages whereas very few particles are born during the final stages of the search process. For sphere, Step, Griewank and Rosenbrock functions, the swarm generates numerous particles during the medieval phases of the optimization. A very strange evolution is evident from Rastrigin function where the swarm continuously evolves at all stages of the search process. As shown in Fig. 2.10, extremely huge swarm size of approximately 270-250 particles is required to solve the Rosenbrock function. Whereas the inter-quartile range of the swarm size for Rastrigin function is 230-210. However the rest of the test functions require approximately 30 particles to obtain the optimal solution (Fig. 2.11). The increase in swarm size with problem dimensionality for Ackley function is shown in Fig. 2.12. As the dimension of the problem space increases, the complexity is exponentially increased. The optimization therefore requires more number of particles to assist the search process. The adaptive swarm therefore acknowledges the solution space complexity and accordingly modifies its search technique automatically to obtain the global optimal solutions

## 2.7 Summary

A new PSO variant called APSO is addressed in this chapter. The algorithm has been tested on various standard unconstrained test functions. The results are analyzed with respect to solution quality and reliability. The solutions are probabilistically compared with two other variants of PSO algorithms. The investigation showed that APSO outperforms the other versions of PSO. The convergence is also improved with the new approach. The solution strategies for unimodal and multimodal problems indicate the exploration and exploitation capabilities of the algorithm. Apart from all these performance improvements the algorithm was executed for all test functions without any parameter tuning. Although there was a sacrifice in the performance with regard to certain functions, the overall performance was much better than the rest of the PSO variants.





## Chapter 3

### Constrained Optimization

#### 3.1 Introduction

The prominent advantage of evolutionary algorithms is their ability to handle complex parameter space defined by a set of equality and inequality constraints. In evolutionary algorithms all the information about the objective function value and the constraint violation should be expressed by a single fitness value. The particle's evaluation and evolution depends greatly on their fitness value. The fitness function should be modelled in such a way that the particles can observe all the disjoint feasible space and return quality solutions.

The general constrained optimization problem can be formulated as:

$$\text{minimize } f(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathcal{R}^n \quad (3.1)$$

subject to inequality constraints :

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, I \quad (3.2)$$

and equality constraints :

$$h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, J$$

The objective function,  $f$  and constraint functions,  $g$  and  $h$  are functions on  $\mathcal{R}^n$ . The equality constraints are called active constraints and the inequality constraints that satisfy  $g_i(\mathbf{x}) = 0$  are also called active constraints. A particle that satisfies all the  $I+J$  constraints is called a feasible particle while the particle that does not satisfy at least one of the constraints is an infeasible particle. The entire parameter set  $S \subset \mathcal{R}^n$  defines the search space, the set  $F \subset S$  is the feasible space and the set  $U \subset S$  defines the infeasible space. The search space is limited by the bounds,  $x_l$  and  $x_u$  on each variable.

$$x_{l,k} \leq x_k \leq x_{u,k} \quad k = 1, 2, \dots, n \quad (3.3)$$

The intersection of constraints with the search space  $S$  defines the feasible space  $F$ . In general, the equality constraints are transformed to inequality constraints using the tolerance value,  $\varepsilon$  as follows:

$$|h_j(\mathbf{x})| - \varepsilon \leq 0 \quad (3.4)$$

The major issue in solving the constraint optimization is how to handle the infeasible particles. Several constraint handling techniques are proposed to carefully handle the infeasible particles. They are classified as follows:

**(i) Rejection Strategy**

All the infeasible particles are discarded during the search process. If the initial swarm contains an infeasible particle, it is replaced by randomly generating a feasible particle.

**(ii) Repair Strategy**

A set of problem dependent rules are used to transform an infeasible particle to a feasible one.

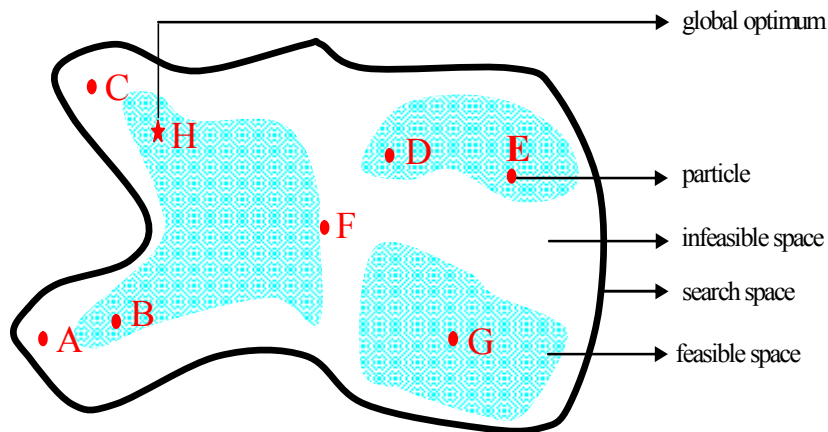
**(iii) Special Operator Strategy**

These operators will ensure that the particles always lie in the feasible space.

**(iv) Penalizing Strategy**

The particles accumulate a penalty to their fitness value for each violation of the constraints.

The advantages and disadvantages of these strategies can be examined by addressing the fitness value of different feasible particles (B, D, E, G, H) and infeasible particles (A, C, F) in Fig. 3.1 with the optimal solution located at 'H'.



**Fig. 3.1** Two dimensional search space

If we were to follow the rejection strategy and remove infeasible particles like C which contains useful information about the optimal location, then the feasible particles D, E and G has to overcome their respective local minima or particle B has to climb down the huge peak to trace the optimal solution. Repair strategies may not exist for all optimization problems. Several researchers investigated that a swarm capable of exploring the entire search space yields fast and better results compared to a swarm constricted to only the feasible space. The most common

way to treat infeasible particles in evolutionary optimization is by using the penalizing strategy. The major issue is how to formulate the penalty function and the resulting penalized objective function. For instance if the penalties are designed based on their distance from the feasible boundary then particle F has better fitness than particle C. But if we happen to know the optimal solution H, then particle C should have had better fitness than the other infeasible particles. Another important concern in using this strategy is how to model the fitness function i.e. how to compare a feasible particle with an infeasible particle. For instance infeasible particle C has more information regarding the optimal solution H than the feasible particle D or G. Since there is no prior knowledge about the optimal solution, great care should be taken to evaluate the penalties for the infeasible particles and suitably model the penalized objective function or the fitness function for the particles.

### 3.2 Penalty Functions

The most common way to handle constraints in evolutionary programming is to use penalty techniques [25], [26]. This approach transforms a constrained problem to an unconstrained problem by adding an additional penalty function to the main objective function as shown in (3.5).

$$\begin{aligned}
 F(\mathbf{c}, \mathbf{x}) &= f(\mathbf{x}) + p(\mathbf{c}, \mathbf{x}) \\
 \text{where} \\
 p(\mathbf{c}, \mathbf{x}) &= 0 \quad \text{if } \mathbf{x} \text{ is feasible} \\
 &= \begin{cases} c_i * |g_i(\mathbf{x})|^2, & i = 1, 2, \dots, I \\ c_j * (|h_j(\mathbf{x})| - \varepsilon)^2, & j = 1, 2, \dots, J \end{cases}
 \end{aligned} \tag{3.5}$$

Here after  $F(\mathbf{c}, \mathbf{x})$  is referred to as the fitness function and  $f(\mathbf{x})$  is the objective function. The penalty function taxes the infeasible particles for each violation of the constraints. While designing the penalty functions, the following aspects of the constraints should be considered: (1) distance from the feasible boundary, (2) soft or hard and (3) difficulty. The severity of the penalties should be weighted based on these characteristics. But what is the most appropriate penalty technique for a given problem? How severe should the penalty terms be for a given set of constraints? There are many more related queries which may not have transparent solutions. Huge penalties for the infeasible particles may completely disregard the information carried by the infeasible particles across the boundary of the feasible region. If the optimal solution lies across the boundary, then the swarm can never identify or guide the particles towards the optimal space. The huge penalties discourage the exploration of the particles which results in poor quality solutions. Where as smaller penalties allow the particles to explore more search space and may stay away from the feasible space for a longer time. The infeasible particles

may not be penalized enough and may dominate the search process. The search process may eventually end without a feasible solution.

If the optimization problem is described by too many constraints, it could be impossible to find a feasible solution. One way to solve this problem is to identify the constraints whose satisfaction is not required but preferred. These constraints are referred to as soft constraints. The penalty technique should be able to identify these soft constraints. The last aspect to be considered is the difficulty in satisfying the constraints. This difficulty arises due to the size of the feasible space being too small compared to the overall search space. The penalty function should accommodate all these features and allot appropriate magnitudes to the penalty terms.

The success of any penalty function depends on the selection of the penalty coefficients,  $\mathbf{c}$ . These parameters are problem dependent and therefore demand huge efforts to repeatedly solve the optimization problem in order to select the optimal set. The definition of these parameters classifies the penalty techniques into several categories. The most general classification includes static penalty, dynamic penalty and adaptive penalty techniques.

### 3.2.1 Static Penalty

The easiest way to implement penalty technique for constrained optimization is to use static penalty functions [27]. In this approach the penalties grow heavier with the increase in constraint violations. The fitness function is given by the objective function plus the penalties for all constraint violation.

The static penalty function approach proposed by Homaifar et al. [28] has the following formulation:

$$F(\mathbf{c}, \mathbf{x}) = f(\mathbf{x}) + \sum_{m=1}^{I+J} c_m d_m^2 \quad (3.6)$$

where

$$d_m = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is feasible} \\ |g_m(\mathbf{x})|, & m = 1, \dots, I \\ |h_m(\mathbf{x})|, & m = I + 1, \dots, I + J \end{cases}$$

The variable  $d$  measures the level of violation and  $\mathbf{c}$  is a vector of penalty coefficients. The selection of  $\mathbf{c}$  is based on the level of violation. This approach defines several penalties for each constraint based on the intensity of the violation. The entire range of violation for each constraint is categorized into intervals. Each interval has a unique penalty coefficient.

The main drawback of this approach is that huge number of penalty coefficients has to be defined and there is no heuristic to determine these coefficients.

### 3.2.2 Dynamic Penalty

The dynamic penalty function [29] models the violation distance along with the dynamics of the search process. The penalties adapt with the level of violation and also with the progress of the algorithm. This approach will levy low penalties to the infeasible particles during the early stages and gradually increase the pressure with the progress of the iterative process. The swarm will be initially assisted by the infeasible particles and with the evolution of the search process; the swarm will impose high penalties on the infeasibles and drive the feasible particles towards the optimum.

The dynamic penalty function proposed by Jong and hauck [30] can be described as follows:

$$F(\mathbf{p}, \mathbf{x}) = f(\mathbf{x}) + \rho_k^\alpha \sum_{m=1}^{I+J} d_m^\beta(\mathbf{x})$$

where

$$d_m(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is feasible} \\ |g_m(\mathbf{x})|, & 1 \leq m \leq I \\ |h_m(\mathbf{x})|, & I+1 \leq m \leq I+J \end{cases} \quad (3.7)$$

$$\rho_k = C * k$$

Variable  $k$  is the iteration number,  $\alpha$  and  $\beta$  control the severity of the penalty values,  $C$  is a constant,  $\rho_k$  regulates the penalties based on the progress of the search process and  $d_m$  is the distance metric. Although this technique performs better than the static penalty, the quality of the solutions is very sensitive to changes in the value of the parameters.

### 3.2.3 Adaptive Penalty Function

The dynamic penalty approach based on the progress of the optimization process has produced mixed results. If the swarm can give life to feasible particles during the preliminary iterative process only then a quality solution is assured. A draught of feasible particles during the early stages will end the search in an infeasible region or in a feasible region far away from the optimal because the later stages of the search process follow a rejection or high penalty for the infeasible particles. In adaptive penalty technique [31], [32] the penalties adapt according to the performance of the swarm. The penalties will reflect the history of previous performance of the particles. The particles will therefore have the ability to reject already visited infeasible areas and move closer to promising feasible space.

Smith, Tate and Coit [33], [34] proposed an adaptive technique by defining a near-feasible threshold (NFT) for each constraint. NFT is the threshold distance

from the feasible region at which the user could consider that the search is reasonably close to the feasible region. The fitness function is as follows:

$$F(\mathbf{x}, t) = f(\mathbf{x}) + (F_{\text{feasible}}(t) - F_{\text{all}}(t)) * \sum_{m=1}^{I+J} \left( \frac{d_m}{\text{NFT}_m} \right)^2 \quad (3.8)$$

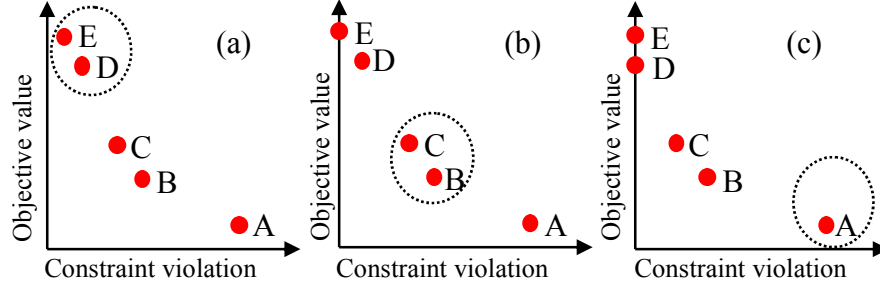
Where  $F_{\text{feasible}}(t)$  is the best feasible solution at generation  $t$ ,  $F_{\text{all}}(t)$  is the best unpenalized solution and  $d$  is the distance metric. The penalties adapt in accordance to the best results available at that generation. The scaling of the term is neither too high nor too lenient. The scaling is tuned to the necessity of the search process. The swarm is always confined to the feasible space or to NFT neighbourhood. Defining the feasible NFT neighbourhood is a very challenging and tedious job. This is the main drawback of this approach. There is also a danger that the infeasible particle may escape the penalty term when  $F_{\text{feasible}}(t) = F_{\text{all}}(t)$ . More over if  $F_{\text{feasible}}(t) \gg F_{\text{all}}(t)$  at early stages then huge penalties will be allocated to all infeasible particles.

### 3.2.4 Self Learning Penalty Function (AP)

Birul Tessema and Gary G. Yen [35] proposed a penalty scheme where useful information from the infeasible particles is effectively used in tracing the optimum. A similar approach with some improvements was successfully implemented in the proposed APSO. The penalties and thereby the fitness of the particles adapt to the requirement and necessity of the swarm. For example if the swarm has no feasible particles, then the penalized objective function has to minimize the constraint violation and on the other hand when the swarm is in search of attractive regions, then the new objective has to minimize the objective as well the constraint violation.

The prominent feature of this approach is that it identifies the right set of infeasible particles which can help the search process. Not all 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 and when the swarm has sufficient feasible particle, low penalties are assigned to infeasible particles with low objective value irrespective of their constraint violations. This concept can be explained by Fig. 3.2. This figure plots the objective function value against constraint violation for a swarm with five infeasible particles (A,B,C,D,E). Particle A has high constraint violation but low objective value while particles D and E are very close to the feasible region but have high objective value. And particles B and C have moderate objective value and constraint violations. Each particle holds significant information and therefore can not be completely ignored. The potentiality of the penalty scheme depends on how and when these infeasible particles

are incorporated in the search process. For instance when the swarm is excavating feasible zones, particles like D and E will furnish the best information about the feasible zone (3.2(a)). On the other hand, if the swarm is tracking the global minimum, then particles like A (Fig. 3.2(c)) will give more information. The intermediate search process can be motivated by particles like B or C (Fig. 3.2(b)).



**Fig. 3.2** Change in the preference for infeasible particles during the evolutionary process

This ideology is mathematically modelled by two terms namely distance value and penalty value. The sum of these terms defines the new fitness function,  $F$ .

$$F(\mathbf{x}) = d(\mathbf{x}) + p(\mathbf{x}) \quad (3.9)$$

The distance value,  $d(\mathbf{x})$  measures the Euclidean distance of the particles from the origin on a two-dimensional space represented by normalized objective value and cumulative normalized constraint violations.

$$d(\mathbf{x}) = \begin{cases} v(\mathbf{x}) & \text{if } r_f = 0 \\ \sqrt{f'(\mathbf{x})^2 + v(\mathbf{x})^2} & \text{otherwise} \end{cases} \quad (3.10)$$

Where  $r_f$  is the ratio of feasible particles in the swarm and  $f'(\mathbf{x})$  represents the normalized objective value given by:

$$f'(\mathbf{x}) = \frac{f(\mathbf{x}) - f_{\min}(\mathbf{x})}{f_{\max}(\mathbf{x}) - f_{\min}(\mathbf{x})} \quad (3.11)$$

$f_{\min}$  and  $f_{\max}$  correspond to the minimum and maximum objective function value of the particles in the swarm. These values can not be static because they are not known a priori. More over if the objective value is greater or lower than the fixed  $f_{\max}$  and  $f_{\min}$  respectively then the normalized objective value is negative and not scaled between zero and one. In this case the motive behind normalization i.e. to scale objective and penalties to the same measure can not be met. The normalized objective value can not be negative because this will deteriorate the penalties in

the penalty term,  $p(\mathbf{x})$ . So these values are not fixed but dynamically adjusted at every iteration.  $f_{\min}$  corresponds to the smallest objective value at the current generation and also from all the past generations. Similarly  $f_{\max}$  is also updated. With this normalization technique, the particle with the least objective value is assigned a zero magnitude. If this particle happens to be in a feasible space, the swarm assumes this as a global optimal solution because this is the smallest optimal solution allowed by this normalization. However this particle is only the best of the swarm at the current iteration and may not even represent a local minimum. Even if there is another particle with a better solution at the next iteration (this particle assumes zero fitness as this is the best at the current iteration), the natural fitness comparison assumes that there is no change in the fitness. The swarm therefore converges on the first feasible solution. In order to avoid this trap, a four fold comparison rules are used. Each particle memorizes not only its normalized objective value but also its objective value in original measure. These rules are quite essential because the fitness of infeasible particle may sometimes be better than a feasible particle. The following rules are used to update  $\mathbf{x}^k$  of each particle at every iteration  $k$ :

- (i) if  $\mathbf{x}_p^k$  and  $\mathbf{x}$  are infeasible, compare normalized fitness values and update accordingly.
- (ii) if  $\mathbf{x}_p^k$  is infeasible and  $\mathbf{x}$  is feasible, update  $\mathbf{x}_p^{k+1} = \mathbf{x}$ .
- (iii) if  $\mathbf{x}_p^k$  is feasible and  $\mathbf{x}$  is infeasible,  $\mathbf{x}_p^{k+1}$  retains its previous best values irrespective of the fitness of current  $\mathbf{x}$ .
- (iv) if  $\mathbf{x}_p^k$  and  $\mathbf{x}$  are feasible, compare the objective values in original measure and update accordingly. Ignore the normalized quantities.

Similar rules are also used to update the global best performer ( $g$ ) of the swarm.  $v(\mathbf{x})$  in (3.10) refers to the total normalized constraint violations.

$$v(\mathbf{x}) = \frac{1}{I+J} \sum_{m=1}^{I+J} \frac{C_m}{C_{m,\max}} \quad (3.12)$$

where

$$C_m = \begin{cases} |g_m(\mathbf{x})|, & m = 1, \dots, I \\ |h_m(\mathbf{x})| - \delta, & m = I+1, \dots, I+J \end{cases} \quad (3.13)$$

$C_{m,\max}$  is the maximum constraint violation for constraint  $m$  among all the particles in the swarm at the current generation. This is also dynamically adjusted in a similar way as  $f_{\max}$ .

The distance value,  $d(\mathbf{x})$  will insist the swarm to initially explore feasible regions and gradually trace the global optimum. It is clear from (3.10) that when



there is a drought of feasible particles, the swarm concentrates only on the constraint violations or the distance of the particles from the feasible space. Once the feasible space has been excavated, the swarm tries to find more feasible space and in parallel traces the optimal region. This ideology is mathematically modelled as a root mean square sum of the objective value and constraint violations. Two feasible particles are compared based on their objective value and two infeasible particles are compared based on either their objective value, constraint violation or both depending on the progress of the swarm. It might happen that the distance value of an infeasible particle is better than a feasible particle. In this case the feasible particle is always given priority to give clear direction to the swarm's search process.

The second term of equation (3.9) is called the penalty value. This term determines which set of infeasible particles can help the exploration at a given generation. So the most useful infeasible particles are given lower penalties compared to other infeasibles. The penalty term is formulated using the rate of feasible particles,  $r_f$  as shown below:

$$p(\mathbf{x}) = (1 - r_f)A(\mathbf{x}) + r_f B(\mathbf{x})$$

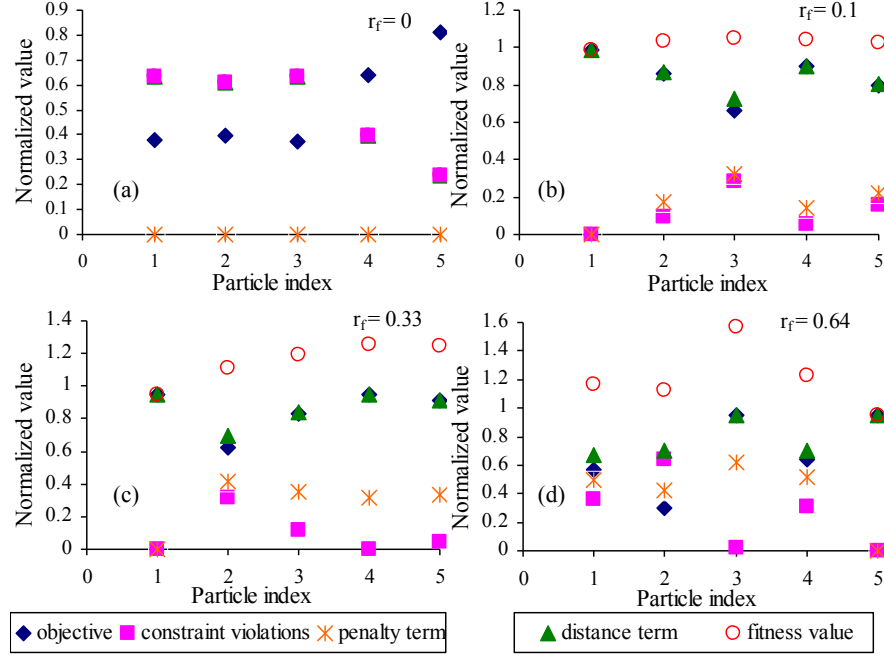
where

$$A(\mathbf{x}) = \begin{cases} 0 & r_f = 0 \\ v(\mathbf{x}) & \text{otherwise} \end{cases} \quad (3.14)$$

$$B(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is infeasible} \\ f'(\mathbf{x}) & \text{if } \mathbf{x} \text{ is feasible} \end{cases}$$

This term makes a clear observation of the swarm progress and analyzes the requirement to further inspire the search process. Based on this, the right set of infeasible particles are selected and made sure that these particles with low penalties are incorporated into the search process. When the ratio of feasible particles is small,  $A(\mathbf{x})$  is the dominating term in equation (3.14). At this stage the swarm requires more feasible particles, so care should be taken to include the infeasible particles close to the feasible boundary. The first term of the penalty value equation implements this idea. The penalties are assigned based on  $A(\mathbf{x})$  i.e. constraint violation. So, particles with low constraint violations are less penalized than particles far away from the feasible region. When  $r_f$  is large,  $B(\mathbf{x})$  is the dominant term. The penalty value is guided by the objective value. So particles with low objective value will be less penalized irrespective of their constraint violation. This supports the search process to identify the optimal regions. In the medieval stage of the search process,  $r_f$  is moderate. Both  $A(\mathbf{x})$  and  $B(\mathbf{x})$  are equally significant. So, lower penalties are assigned to particles with moderate objective value and constraint violations. The adaptive nature of the penalties with  $r_f$  is shown in Fig. 3.3. The objective value, distance term, penalty term and the fitness value are shown for five infeasible particles in the figure. The top figure on the left shows the pen-

ality assignment when  $r_f = 0$ . Since there are no feasible particles, the only aim of the algorithm is to trace the nearest feasible space. So the total constraint violation is reflected as the fitness value. The objective value places no role at this stage. The distance term is the same as the fitness value and is equal to the constraint violations. The penalty term for all the particles is zero. Particle 5 which has the



**Fig. 3.3** Fitness assignment in self learning penalty function

lowest constraint violations is the fittest particle in the swarm. In Fig. 3.3 (b) particle 1 is feasible so no penalties are assigned to it. The distance term is small for particles that have moderate objective as well as the constraint violations. Particles 2, 4, and 5 have low violations but compared to particle 5, they have higher magnitude of objective value. So the distance term is small for particle 5. Since  $r_f = 0.1$ ,  $A(\mathbf{x})$  is dominating term in (3.14). So particle 4 which has the lowest constraint violations has the smallest penalty term. Similar behaviour is seen with  $r_f = 0.33$  in Fig. 3.3 (c). In Fig. 3.3 (d),  $r_f = 0.64$  so  $B(\mathbf{x})$  dominates over  $A(\mathbf{x})$ . Particle 2 which has the lowest objective value also has the smallest penalty term. Although particle 3 has very low constraint violation, it is ignored since it has higher objective value compared to particle 2.

The proposed penalty function technique designs the penalty terms based on the search procedure and therefore no problem dependent information is required to formulate the penalties. The penalties are automatically adapted to the requirements of the swarm. There are as such no penalty coefficients tuning or redefini-

tion of the penalty function for a new application. The penalty function once defined can be used without any restructuring on any optimization problem.

Although the above technique significantly utilizes the information from the infeasible particles, it still suffers from normalization parameter definition. In dynamic normalization the parameters are updated at every generation. So the scaling of the fitness is not equal in all the generations. However they are compared between various generations. This might possibly lead to ambiguous search process.

### 3.3 Novel Approach (Rule Based Penalty Approach, RBP)

All the penalty function approaches discussed so far are problem dependent. Most of them require rigorous tuning of penalty coefficient set,  $\mathbf{c}$ . The infeasible particles can be successfully exploited by strategically adjusting these parameters. The adaptive penalty techniques are developed to dynamically adjust these parameters with the search process. The adaptive penalty function and self learning penalty function fall in this category. These methods are able to eliminate the penalty coefficients but in the process give rise to new parameters such as the normalization parameters in self learning penalty function approach. The major problems in all these methods arise in the definition of the fitness function. The fitness function is built based on the information available from the objective function and constraint violations. The fitness function is therefore a combination of two different measures of different scale basically the objective function value and the total constraint violation. In order to provide a good balance, the two measures are either normalized or scaled accordingly. This process results in new parameters. A rule based penalty function (RBP) approach is proposed in this section.

#### 3.3.1 *Why do we need a fitness function?*

The major objectives in optimization are to find a feasible solution by minimizing the constraint violations and then find a quality solution by minimizing the objective function. The information related to these two targets is collectively defined by the fitness function. But what is the role of the fitness function?

In stochastic optimization the particles evolve with generations. During the evolutionary process, the particles exchange information. As the particles move from one location to the other, it tries to find the right direction to reach the final target. During the evolutionary process the particles evaluate their performances by comparing with a certain measure which characterizes the objectives of the optimization process. This measure is referred to as the fitness function.

Another usage of fitness function is during the comparison of two or more particles. As the particles are social beings, they try to associate with their neighbours and enrich their performances through these interactions. This exchange of infor-

mation is usually about the distance of the particles from the feasible boundary and the objective function value i.e. fitness value.

The above analysis infers that the particles need a suitable measure to validate their flight and evaluate their performance. The required measure need not be a combination of the objective function and constraint violations.

### ***3.3.2 The Idea***

During the search process, the swarm tries to minimize the constraint violations or the objective function. So in the new approach the fitness is either the objective function or the total constraint violations and not a combination of both measures. The motive of the swarm's motion is straight forward i.e. trace the feasible solution space and then identify the optimal solution. So the fitness can be modelled accordingly to meet the swarm's objectives.

In the early stages, the swarm has no information about the feasible space. During this period, the fitness is characterized by the total constraint violations. The particles with low constraint violations are preferred. These particles dominate the search process. When the swarm has explored enough feasible regions, the motive of the swarm is to exploit the discovered regions. The swarm therefore tries to trace the optimal solution. During this period the fitness is characterized by the objective function.

### ***3.3.3 Who decides the camouflages?***

The major shifts in the swarm's motives occur with its progress. In primitive stages, the swarm might be full of infeasible particles, while in the advanced stages the swarm might have considerable feasible particles. So this improvement in the swarm (ratio of feasible particles,  $r_f$ ) can be a good decision maker for the camouflage in the fitness characterization.

A feasible particle is always evaluated based on its objective value. Where as the infeasible particles are assessed based on either their objective value or their constraint violations. The ratio of feasible particles decides which measure should be used. If  $r_f < 0.1$ , the measure of constraint violations is used to judge the infeasible particles. So particles which are closest to the feasible regions are incorporated into the search procedure. Where as when  $r_f > 0.2$ , the measure of objective function is used to select the most significant infeasible particles. So particles with low objective value are preferred irrespective of their constraint violations. The infeasible particles with low objective value have lot more information about the optimal solutions; hence these particles are associated with the feasible particles to trace the global optimum. When  $r_f$  is in the range of (0.1, 0.2), a random choice of the two measures decides the selection procedure.

### 3.3.4 The Procedure

As discussed in the above sections, the fitness function is used to update the particles individual best performance,  $p$  or for selecting the global best performer,  $g$  in its neighbourhood. A set of decision making rules acts as a constitution to perform these actions. The rules decide which measure should be used for updating  $p$  and  $g$ . There are basically three major rules. Let a particle have an objection value  $f1=|f1(\mathbf{x})|$  and constraint violations  $v1=|v1(\mathbf{x})|$  at the current generation. The local best,  $p$  corresponding to this particle has an objective value  $f2=|f2(\mathbf{x})|$  and constraint violations  $v2=|v2(\mathbf{x})|$ . The process for updating  $p$  is as follows:

**RULE 1:**

This rule is applicable only when the two states are feasible i.e. when the particle at the current generation is feasible ( $v1 = 0$ ) and  $p$  is also feasible ( $v2 = 0$ ). In this situation the objective function is the suitable measure for updating  $p$ .

$$\text{if } f1 < f2 \text{ then } p = f1 \quad (3.15)$$

**RULE 2:**

This rule ensures that a feasible state of a particle always wins the race against its infeasible state i.e. the particle at the current generation is feasible ( $v1 = 0$ ) and  $p$  is infeasible ( $v2 \neq 0$ ) or vice-versa. No measure is required for selection. The feasible state of the particles always has the upper hand.

$$\begin{aligned} \text{if } (v1 = 0 \text{ and } v2 \neq 0) \text{ then } p &= f1 \\ \text{if } (v1 \neq 0 \text{ and } v2 = 0) \text{ then } p &= f2 \end{aligned} \quad (3.16)$$

**RULE 3:**

This rule governs two infeasible states of the particle i.e. the current state of the particle is infeasible ( $v1 \neq 0$ ) and  $p$  is also infeasible ( $v2 \neq 0$ ). In this situation, either of the two can be used for comparison. The ratio of feasible particles,  $r_f$  decides which measure should be used for updating the particle. There are three sub divisions in this rule based on the value of  $r_f$ .

$$\text{if } (v1 \neq 0 \text{ and } v2 \neq 0) \text{ then} \quad (3.17)$$

**RULE 3.1:**

The first sub-division of rule 3. This rule is applied when  $r_f < 0.1$ . When the number of feasible particles in the swarm is less than 10% of the population then the constraint violation is used as the assessment tool.

$$\text{if } v1 < v2 \text{ then } p = v1 \quad \forall r_f < 0.1 \quad (3.18)$$

**RULE 3.2:**

The second sub-division of rule 3. This rule is applied when  $0.1 \leq r_f \leq 0.2$ . In this situation a random selection of the measures is used for updating the particle. A uniform random number generated between (0,1) is compared against a specified threshold,  $U$  to decide the choice of the measure for updating the particle.

$$\left. \begin{array}{l} \text{if } v1 < v2 \text{ then } p = v1 \quad \forall \text{ rand}() < U \\ \text{if } f1 < f2 \text{ then } p = f1 \quad \forall \text{ rand}() > U \end{array} \right\} \forall 0.1 \leq r_f \leq 0.2 \quad (3.19)$$

**RULE 3.3:**

The third sub-division of rule 3. This rule is applied when  $r_f > 0.2$ . When the swarm has more than 20% feasible particles, the objective function is used to judge the particles performance.

$$\text{if } f1 < f2 \text{ then } p = f1 \quad \forall r_f > 0.2 \quad (3.20)$$

### 3.3.5 The Benefits

The proposed penalty function approach is very simple to implement. It can be used in any optimization algorithm. The fitness characterization is based on three simple rules. The swarm's trajectory is guided by these rules. The fitness is shifted to either of the two measures based on the progress of the optimization process. The approach is therefore independent of the problem being solved. Based on the performance of the particles, the swarm strategically adapts its fitness function. The penalty technique utilizes either the objective function or the constraint violation. Since these two measures are not combined there is no need for defining the specialized penalty function or penalty coefficients. The two measures need not be to the same scale. So no normalization parameters are required. Both the measures are effectively utilized to guide the swarm to the optimal space.

The self learning penalty function and the new penalty techniques are successfully implemented in APSO algorithm. Both the algorithms are very easy to implement. The penalty technique observes the success of the search process at each generation and adapts its penalties. The penalty function is therefore free from problem dependent information.

## 3.4 Test Problems

The performance of the proposed penalty approach along with the other mentioned techniques were evaluated using the 13 test functions explained in [36]. These test functions describe the various significant characteristics of the objective functions such as complexity, nonlinearity, dimensionality etc associated with constrained optimization problems. A different variety of constraints such as lin-

ear inequalities (LI), nonlinear inequalities (NI) and nonlinear equalities (NE) have been incorporated in these test functions. The general characteristics of the test functions along with their associated constraints are listed in Table 3.1. Test problems g02, g03, g08 and g12 are maximization problems. They are solved as minimization problems using  $-f(\mathbf{x})$ .

**Table 3.1** Summary of test functions

Function	Type	Dimension	LI	NI	NE
g01	Quadratic	13	9	0	0
g02	Nonlinear	20	0	6	0
g03	Polynomial	10	0	0	1
g04	Quadratic	5	0	6	0
g05	Cubic	4	2	0	3
g06	Cubic	2	0	2	0
g07	Quadratic	10	3	5	0
g08	Nonlinear	2	0	2	0
g09	Polynomial	7	0	4	0
g10	Linear	8	3	3	0
g11	Quadratic	2	0	0	1
g12	Quadratic	3	0	1	0
g13	Nonlinear	5	0	0	3

The performance of the self learning penalty technique and the RBP approach were compared with a static and dynamic penalty functions mentioned in the above sessions. The penalty coefficients for static and dynamic penalties were well tuned for each of the benchmark problem. No changes were made to the APSO algorithm. The particle update strategies were unaltered while solving the various benchmark problems. APSO algorithm with self learning penalties or RBP was treated as a parameter free constrained optimization algorithm or a black box optimization tool. Each of the benchmark problems was solved using the four penalty techniques in APSO. One hundred independent runs were performed on each of the benchmark problems. The maximum allowed function evaluations in each of the run were set to 60,000. All equality constraints were treated as inequality constraints with a tolerance value of 0.001.

The statistical comparisons among the four penalty techniques are performed using the mean, median, standard deviation and hypothesis testing. In order to perform these analyses, the simulation results should obey the following assumptions. Let's assume that we are comparing the simulations results obtained from tests A and B.

- (i) The outcomes of the tests A and B should be random
- (ii) The results from each of the tests should follow a similar distribution
- (iii) The simulation results should have a fixed location (deterministic component) and a fixed variance (random component) i.e.

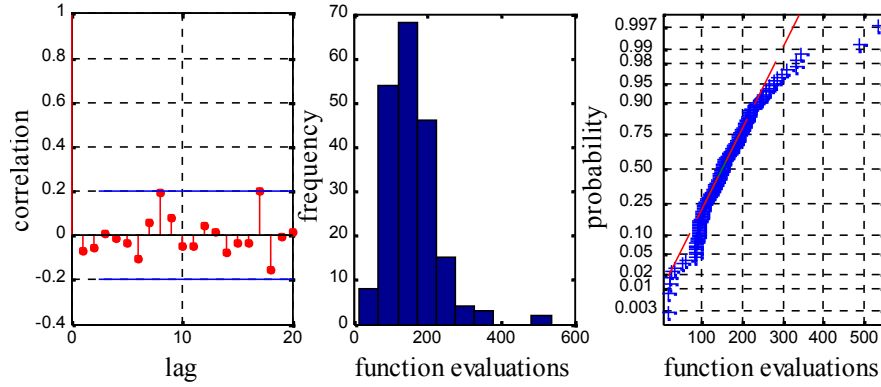
$$\text{test result} = \text{deterministic component} + \text{random component} \quad (3.21)$$

Validity of these assumptions allows one to make probabilistic inference not only on the already performed simulations but also on the outcomes of the future simulations. An autocorrelation plot can be used to check the randomness of a simulation result. The randomness can be judged by evaluating the autocorrelation for the simulation results at different time lags. Random components have autocorrelation coefficients which are either zero or located within the 95% confidence interval. The presence of a deterministic component in a simulation result can be viewed using a histogram. A probabilistic plot can be used to ascertain whether or not a simulation results follow a given distribution. The distance of the plot to a straight line indicates the goodness of fit. The above three plots validate all the assumptions. So these plots are drawn for the hundred simulation results obtained by each of the penalty technique on every test function. The plots for the simulation results obtained using static penalty method for test function g03 are shown in Fig. 3.4.

The mean and standard deviation of the simulation results are listed in Table 3.2. The proposed rule-based penalty approach (RBP) was able to trace the known optimum for g01, g03, g04, g05, g06, g08, g09, g11 and g12. For test functions g03, g05, g06, g09 and g11 the results were better than the best solution reported so far. For g02, g07 and g13, the best value was very close to the known optimum. The best results for g10 were no way near to the global optimum. The improved self learning penalty technique (AP) was able to find very good results for test functions g01, g02, g03, g04, g05, g06, g08, g09, g11 and g12. The best values found by AP are better than the known optimum. The algorithm was not able to find the optimum for g07, g10 and g13. The results for static penalty (SP) and dynamic penalty (DP) reported in this chapter are not similar to those published by other authors because the algorithm related parameters are not altered while solving the various test functions. The DP and SP were very effective in solving g13. The known optimum was found for g01, g03, g04, g05, g06, g08, g09, g11 and g12. In comparison to the best results found, all the four penalty methods were effective for g01, g03, g04, g05, g06, g08, g09, g11 and g12. The SP approach was able to find a far better result than the known optimum for g09. For test function g02, AP was the only approach to find the optimal solution. RBP performed better on g07. All the approaches were able to obtain better solution than the known optimum for g03, g09 and g11. SP was better on g10.

The mean results for g01, g02, g07 and g13 obtained by DP were the best compared to the other techniques, where as it had the worst performance on g03, g05 and g06. The mean results for g04, g08, g11 and g12 were the same for all the four approaches. SP was good on g09 and g10, while it had worst performance on g01, g02 and g05. AP had the best mean results for g06 while it was the worst for g10. The new RBP was the best on g05. Both AP and RBP perform equally on g01, g03, g04, g05, g06, g07, g08, g09, g11, g12 and g13. RBP was the worst performer on g07 and g13.





**Fig. 3.4** Autocorrelation plot, Histogram and Weibull probability plot of SP for first feasible evaluation in test function g03

The standard deviation was either zero or almost equal in all the four methods for g02, g04, g08, g09, g11, g12 and g13. Both DP and RBP had lower standard deviation for g01. Except DP, all the other methods had almost zero standard deviation for g03. RBP had the global solution for g06 in all 100 simulations. RBP was the most effective method for g05. SP performed relatively well on g10.

In terms of the mean and standard deviation of the results (objective value), all the four techniques performed equally well on g03, g04, g08, g09, g11 and g12. RBP, AP and SP were able to produce mean results very close to the optimum for g06. However RBP produced more consistent results. None of the methods were able to find the optimal solution for g02, g07, g10 and g13. The optimal results for g05 using the four methods were never consistent. This can be viewed from the standard deviation. RBP was the best among the three in terms of consistency. Both RBP and AP produced feasible solutions in all the runs for all the test functions. SP has a 100% success rate on all test functions except g06 and g13 (94% and 92 % respectively). DP had a 100% success rate on g02, g04, g05, g08, g09, g11 and g12, whereas for g01, g03, g06, g07, g10 and g13 it had a success rate of 91, 75, 83, 98, 66 and 97% respectively.

Although the best results were close to the known optimum for g01, g05 and g07, the mean is quite far from these results. This indicates the presence of outliers. Similarly the performance of the four penalty techniques on g02, g10 and g13 can be better analyzed using another measure of central tendency called the median. This gives a better picture about the distribution of the results and is less influenced by the presence of few large values. Table 3.3 presents the first (25th percentile), second (50th percentile or median) and third (75th percentile) quartiles of the results obtained using the four methods. For g01, both DP and RBP had more than 75% of the results which are equal to the known optimum, whereas the other two techniques could only reproduce the optimum with a success rate of 50%. According to the results reported in Table 3.2, RBP produced the best mean results for g05. However Table 3.3 indicates that AP produced better results than RBP by nearly 50%, which indicates that AP had outliers. The 25th percentiles of

all the methods are quite far away from the global optimum. This implies that there is very small probability of attaining the global solution for g05. For g07, AP had better quartiles compared to RBP. But the chances of obtaining the known optimum in both the techniques were very less. For g02, 75% of the results in AP are better than the 25% of the results in RBP. For g10, the lower quartiles indicate that RBP produced better results but the upper quartile in RBP was far worst compared to AP. On an average AP performed better than RBP for g13. But the results were quite far from the global optimum.

The quartiles for the number of function evaluations required to solve the various test functions are depicted in Table 3.4. The simulations terminate when they either hit the target fitness value or maximum function evaluations. For g02, g05, g07 and g13, all the penalty techniques required the maximum evaluations. AP was the most successful technique in solving g01 with few function evaluations. AP solved 25% of the simulations within 3148 evaluations and 50% of the simulations in 180000 evaluations. RBP was the most expensive of all the remaining methods. It always hit the maximum evaluations in reaching the target fitness. For g03, all the three quartiles are better in case of AP. SP and RBP had nearly 5000 evaluations more in each quartile compared to AP. Test function g04 was solved by AP and RBP within a maximum limit of 6600 evaluations, whereas the other two methods required a minimum of 6200 evaluations. RBP could solve g06 with an average of 10,000 evaluations less than AP. SP was the best for this test function. The maximum evaluations required to solve g08 using AP was 552 evaluations. RBP required 1219 evaluations while SP and DP couldn't solve the problem with five times this number. For g09, the maximum function evaluations required by AP was 2457 while RBP, SP and DP required 3, 5 and 15 times more evaluations. Both SP and AP solved g11 with similar number of function evaluations, while RBP took double the number. On an average AP was very efficient in solving most of the test functions with few function evaluations. RBP was the winner for g06. Though SP showed major success in solving g06 than AP and RBP, in most of the cases it took 3-5 times more number of evaluations. Dynamic penalty was the worst performer for all the test functions with regard to function evaluations.

As discussed in chapter 2, APSO has an adaptive swarm size. The evolving strategy changes with the progress of the particles. So it is interesting to know which penalty technique best assists the search process and in turn reduces the swarm size. Table 3.5 lists the swarm size required for all the test functions by the four penalty techniques. AP has the smallest swarm size for most of the test functions. RBP is the second in the race. It showed good performance with g13 compared to AP or SP. For g01, g02, g04, g05 and g11, the swarm size was very similar to AP. For the rest of the test functions, it took 10-20 more particles than AP to attain the optimal solution. SP was good on g02 and g03 but it had a huge swarm for g06, g09 and g10. DP had an explosive evolving strategy for most of the test functions.

**Table 3.2** Mean and standard deviation of the results in terms of objective value

	Optimal	Best result				Worst Result				Mean Result				STD			
		RBP	AP	SP	DP	RBP	AP	SP	DP	RBP	AP	SP	DP	RBP	AP	SP	DP
g01	-15.000	-14.999	-14.990	-15.000	-15.000	-11.965	-12.000	-11.400	-11.397	-14.632	-14.182	-13.986	-14.851	0.898	1.198	1.325	0.584
g02	0.804	-0.734	-0.804	-0.571	-0.796	-0.253	-0.306	-0.232	-0.444	-0.475	-0.600	-0.417	-0.626	0.087	0.096	0.078	0.074
g03	1.000	-1.001	-1.002	-1.002	-1.001	-0.987	-0.994	-0.990	0.000	-1.000	-1.000	-1.000	-0.799	0.002	0.001	0.001	0.289
g04	30665.53	-30665.53	-30665.30	-30665.53	-30665.53	-30665.52	-30665.00	-30665.52	-30665.52	-30665.53	-30665.24	-30665.53	-30665.53	0.002	0.069	0.002	0.003
g05	5126.498	5126.495	5126.499	5126.499	5126.498	6112.217	6112.228	6112.214	6112.369	5496.759	5522.475	5640.542	5581.604	388.4	436.1	437.7	440.3
g06	6961.814	-6961.811	-6961.889	-6961.810	-6960.794	-6961.753	-6960.109	-6960.508	-5951.125	-6961.809	-6961.057	-6961.786	-6867.273	0.006	0.383	0.143	187.5
g07	24.306	24.361	24.450	24.501	24.489	32.632	31.270	31.518	28.993	27.641	27.191	26.476	25.763	2.141	1.922	1.814	1.096
g08	0.096	-0.096	-0.096	-0.096	-0.096	-0.096	-0.095	-0.096	-0.094	-0.096	-0.096	-0.096	-0.096	0.000	0.000	0.000	0.000
g09	680.630	679.425	680.001	678.597	679.572	680.922	680.977	679.603	682.075	680.066	680.194	679.018	680.675	0.252	0.226	0.148	0.525
g10	7049.331	7098.718	7181.315	7064.585	7102.788	10746.207	9933.586	8955.160	14217.295	8459.295	8658.972	7761.018	7847.852	852.9	613.9	508.0	936.4
g11	0.750	0.749	0.749	0.749	0.749	0.752	0.750	0.759	0.756	0.750	0.749	0.750	0.750	0.000	0.000	0.001	0.001
g12	1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000	-0.993	-1.000	-1.000	-1.000	-1.000	0.000	0.000	0.000	0.001
g13	0.054	0.084	0.126	0.068	0.063	1.945	1.478	1.265	1.012	0.832	0.772	0.823	0.681	0.312	0.248	0.333	0.264

**Table 3.3** First, second and third quartiles of the results in terms of objective value

Problem	Static Penalty			Dynamic Penalty			Adaptive Penalty			Rule Based Penalty		
	25	50	75	25	50	75	25	50	75	25	50	75
g01	-14.999	-14.999	-12.8	-14.999	-14.999	-14.999	-14.989	-14.987	-13	-14.995	-14.981	-14.956
g02	-0.47247	-0.4215	-0.36321	-0.67868	-0.62938	-0.5808	-0.6756	-0.59704	-0.5345	-0.53747	-0.48423	-0.41686
g03	-1.0002	-1	-0.99988	-0.99436	-0.947	-0.75727	-1.0002	-0.99997	-0.99962	-1.0001	-0.99998	-0.9997
g04	-30666	-30666	-30666	-30666	-30666	-30666	-30665	-30665	-30665	-30666	-30666	-30666
g05	5261.8	5377.3	6112.2	5230	5261.4	6111.3	5165.2	5192.6	6112.2	5218.1	5261.8	5969.7
g06	-6961.8	-6961.8	-6961.8	-6957.8	-6939.2	-6887.9	-6961.3	-6961.2	-6960.8	-6961.8	-6961.8	-6961.8
g07	25.184	25.69	27.328	25.055	25.295	25.926	25.449	26.58	29.173	25.665	27.952	29.758
g08	-	-	-	-	-	-	-	-	-	-	-	-
	0.095824	0.095824	-0.095824	-0.09581	-0.0958	-0.095783	0.095793	0.095776	0.095726	0.095825	-0.095824	0.095824
g09	678.94	679	679.07	680.36	680.62	680.98	680.05	680.1	680.26	679.98	680.02	680.15
g10	7343.7	7649.9	8186.5	7391.7	7637.2	8134.9	8271.1	8850.8	9055.5	7907.6	8858.3	10256
g11	0.74976	0.75	0.75033	0.74962	0.74999	0.75084	0.7491	0.74927	0.74956	0.74992	0.75001	0.75019
g12	-1	-1	-1	-0.99993	-0.9999	-0.99986	-1	-1	-1	-1	-1	-1
g13	0.72314	0.87794	0.96617	0.4858	0.69809	0.95453	0.5345	0.8441	1	0.7003	0.9624	0.9996

**Table 3.4** First, second and third quartiles of the results in terms of total function evaluations

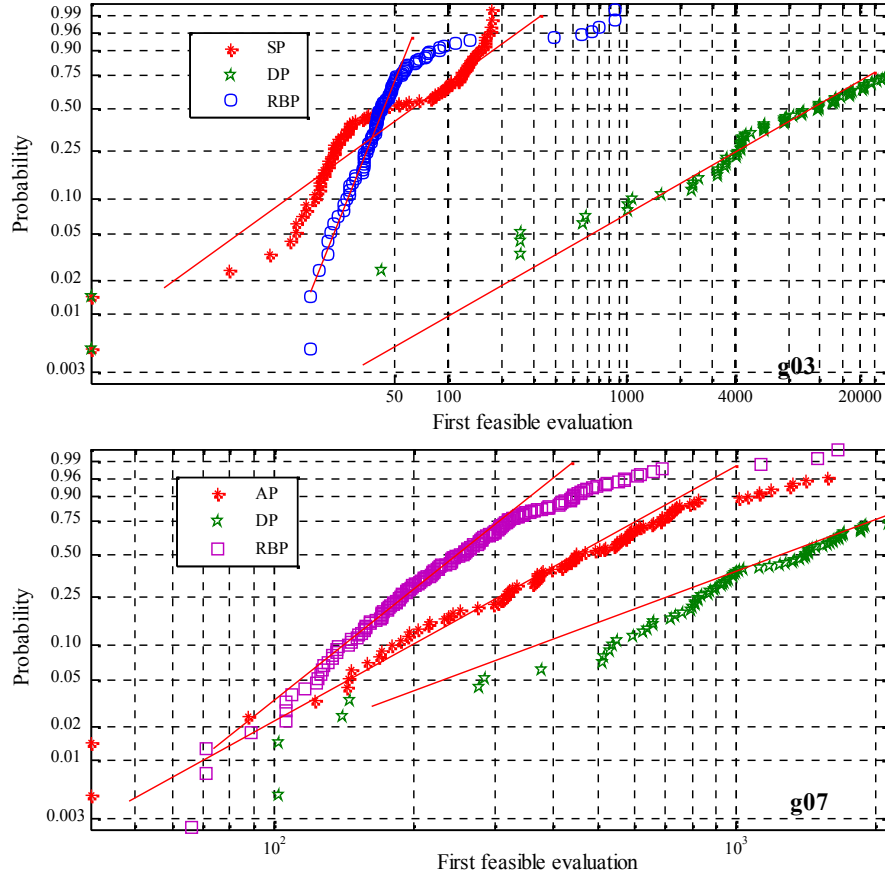
function	Static Penalty			Dynamic Penalty			Adaptive Penalty			Rule Based Penalty		
	25	50	75	25	50	75	25	50	75	25	50	75
g01	11182	39884	60001	25095	32487	53955	3148	18278	60000	60001	60001	60001
g02	60001	60001	60001	60001	60001	60001	60000	60000	60000	60001	60001	60001
g03	15869	18740	25167	60001	60001	60001	10462	15823	23369	18760	24150	31016
g04	6129.5	11331	18670	42023	46206	49426	2474	3761.5	6568	2961.5	4225	6384.5
g05	60001	60001	60001	60001	60001	60001	60000	60000	60000	60001	60001	60001
g06	20022	29242	41202	60001	60001	60001	34705	60000	60000	25069	43247	51424
g07	60001	60001	60001	60001	60001	60001	60000	60000	60000	60001	60001	60001
g08	2199	3711	5393	2147.8	3210	4277.5	245	391.5	551.5	316.5	643.5	1218.5
g09	5714.3	8479	11074	26665	28588	30910	1374	1891	2457	2120	3891	7099.5
g10	60001	60001	60001	60001	60001	60001	60000	60000	60000	60001	60001	60001
g11	3945.3	8252	13197	20515	28035	41896	3692	6432	12092	9409.5	16674	24079
g12	6321.3	8202	10512	1984	3496	4708.8	1078	1604	1976	2047.5	2543.5	3489
g13	60001	60001	60001	60001	60001	60001	60000	60000	60000	60001	60001	60001

**Table 3.5** Mean swarm size, evaluation to first feasible solution and percentage feasible evaluations for all the four penalty methods

	Swarm size				First feasible solution			% feasible evaluations			
	RBP	AP	SP	DP	AP	SP	DP	RBP	AP	SP	DP
g01	35	32	43	326	99.06	81.15	4228.02	0.3254	0.3325	0.5679	0.3782
g02	37	41	29	480	1.00	1.00	1.00	0.0007	0.0005	0.0008	0.0005
g03	36	16	10	478	151.12	69.00	19907.15	0.4306	0.1699	0.2801	0.0174
g04	16	14	20	400	9.65	46.46	15.48	0.4805	0.4149	0.4790	0.3169
g05	41	33	57	400	319.37	7637.31	10211	0.2831	0.0119	0.2060	0.0125
g06	45	20	109	480	1098.81	7837.14	27976.67	0.1433	0.0579	0.0644	0.0095
g07	61	40	41	480	286.73	586.78	2600.88	0.3962	0.1531	0.3602	0.2063
g08	10	5	73	65	81.29	390.38	75.14	0.6530	0.4029	0.2685	0.2268
g09	18	10	111	290	117.98	353.71	151.95	0.5374	0.4073	0.3010	0.2662
g10	39	29	249	480	1558.88	3440.25	7694.50	0.2492	0.0788	0.1189	0.0863
g11	27	21	22	297	91.95	80.15	48.83	0.4481	0.2149	0.1593	0.3653
g12	17	7	19	68	49.36	33.24	32.02	0.6885	0.3875	0.3757	0.0612
g13	27	42	64	480	87.97	4774.45	39220.48	0.1862	0.0730	0.2382	0.0066

The penalty technique assists the swarm to explore promising feasible space and exploit the optimal space. The ability of the algorithm to obtain quality solutions depend on the effectiveness of the penalty technique in directing the swarm motion. Table 3.5 gives an insight into how much percentage of feasible evaluations are generated during the evolutionary process. RBP is the forerunner in all the test functions. For g01, SP produced 5% more feasible evaluations than the rest of the methods. For g02 and g04, all the methods, performed equally good. For g03, RBP generated 43% while AP, SP and DP generated 17, 28 and 1% respectively. For g05, g07, g08, g11 and g12, RBP produced nearly 10-15% more feasible evaluations than the rest of the methods. SP performed relatively well compared to AP and DP. This analysis indicates that RBP has an efficient strategy to move the swarm from an infeasible space to optimal feasible space. This penalty technique can be very useful for highly constrained optimization problems where the feasible space is either small or sparse (scattered).

In constrained optimization, the penalties for the infeasible particles are strategically adjusted to excavate feasible solution space. The following analysis describes the ability of the various penalty techniques in finding the first feasible solution. When the swarm is in an infeasible space both AP and RBP has the same strategy to explore the first feasible space. Hence the statistical results are similar for both the techniques. Hence only one of them is considered in this analysis. The results from 100 independent random runs by the four methods can be compared only if the simulation results from all the methods arise from a similar probability distribution. The results closely fit to Weibull distribution. The Weibull plots for g03 and g07 are shown in Fig. 3.5. The probability of attaining the first feasible



**Fig. 3.5** Weibull probability plots for first feasible evaluation

solution in any given number of evaluations can be evaluated from these graphs. For g07, RBP had a 0.9 probability of finding the first feasible solution in 400 evaluations, whereas SP and DP require 1000 and 5000 respectively. From the figure it is evident that RBP or AP has more probability of finding the first feasible solution in minimum function evaluations. The Weibull parameters  $\alpha$  and  $\beta$  are listed in Table 3.6. The scale parameter,  $\alpha$  defines the number of function evaluations required to obtain the first feasible solution in 63.2% of the trails [37]. For g01, SP performs better i.e. require few function evaluation to reach the first feasible solution. For g02, all the four approaches are able to start the search process with a feasible solution. Though RBP performed better in g03, g04, g11 and g12, SP is not far quite far from the performance of RBP. But for the rest of the test functions, RBP and AP outsmart the other techniques and their performance is completely out of reach by SP or DP. The shape parameter,  $\beta$  indicates the slope of the Weibull plot. It gives the rate at which a particular penalty function can find a feasible solution. RBP is the solo winner on most of the test functions except g05 and g13. This analysis signifies the exploration capabilities of AP and RBP.

**Table3.6** Weibull parameters

function	alpha			beta		
	SP	DP	RBP	SP	DP	RBP
g01	88.737	2646.3	109.5	2.0565	0.53469	3.8569
g02	1	1	1	Inf	Inf	Inf
g03	74.704	17224	71.824	1.2911	0.87505	2.5466
g04	24.062	13.814	8.8894	0.54578	0.82517	1.0556
g05	8410.8	25107	254.45	1.4145	inf	0.88331
g06	7276.7	31051	1056.4	0.88366	1.5513	1.0846
g07	637.55	2423.8	318.66	1.2426	0.90601	2.0395
g08	347.05	82.147	89.468	0.81807	1.3878	1.8597
g09	301.52	154.25	132.84	0.77892	1.0337	1.5797
g10	3780.9	6578.5	1739	1.4414	0.82543	0.92942
g11	84.367	42.795	52.884	1.1628	0.82148	1.326
g12	31.564	34.069	27.343	0.90782	1.1969	1.4181
g13	2931.7	42331	76.101	0.62289	5.5668	1.2515

These techniques can be a good choice in solving complex constrained optimization problems. The convergence properties of the four penalty techniques for various test functions are shown in Fig. 3.6. The high initial objective value (fitness value + penalties) in SP and DP are not shown in the figures. For g01, g02 and g03, AP converges much faster than the rest of the other methods. However RBP always started at a near optimal and had relatively lower objective values compared to the other techniques. In the figure showing the convergence properties for g06, the objective value for RBP has a lower value than the optimum during the oscillatory stage. This is because in all the penalty methods except RBP, there is an additional penalty term in the objective value apart from the fitness value. So the objective in the initial stages is comparatively larger than in the final stages. However in RBP there are no penalties in the objective value. So during the initial infeasible generations, the objective value may be less than the final optimal value. For g05, g07, g09, g10 and g13, the convergence properties of RBP are very similar to that of AP. However for g06, g08 and g12, the oscillations in the global best solution are damped much quickly than the rest of the methods.

In APSO, the swarm evolves gradually with the progress of the search process. The addition or removal of particles depends on the performance of the particles. The penalties alter the direction of the particles and therefore influence the success of the particles. It is of considerable significance to observe whether the proposed penalty technique has the ability to evolve efficiently without an explosion. Fig. 3.7 shows the evolution of the swarm for g04, g05, g06 and g12 using the four penalty methods. The evolving strategy in static penalty for all the four test functions was completely different. Similarly the swarm generation for any test function was totally different in all the four penalty methods. Using SP for g04, the swarm initially builds the swarm rapidly but after a few generations, there is a gradual growth and decay in the swarm size. In DP, the swarm monotonously



grows without any decay. This happened with all the test functions. This implied that APSO with DP require extremely huge population compared to the other methods and therefore not a recommended choice for this algorithm. In AP, the swarm size is relatively small for the entire search process. During the final search process, there was a rapid growth. In RBP, there were some moderate growth and decay in the initial stages. But for most of the medieval stage, there were practically no evolution and the swarm size was minimal. However during the exploitation of the discovered optimal space, the swarm generated huge number of particles. But once the swarm attained the required information, it immediately disposed the entire new population.

Since the simulation results are the outcomes of a stochastic search process, it would be appropriate to relate the comparisons in terms of probability rather than by any measure of central tendency. Some important questions that will be dealt in this section are: Does the use of an efficient penalty technique alter the quality of optimal solutions? Do the four penalty methods differ in performance? Such questions can be addressed by formulating a hypothesis which can be evaluated using the simulation results. A hypothesis is an empirically-testable statement about a relationship involving two or more variables. The performance that is observed using the mean or median might have occurred by chance. Testing of hypothesis answers such questions.

A nonparametric Mann-Whitney U test [38], [39] is used to perform the hypothesis test. This test assesses whether the medians of two random samples from two independent observations are statistically different from each other. The test is based on the following assumptions: (i) the data from each group should be random; (ii) the two groups under study should be independent; and (iii) the groups should represent identical continuous distributions not necessarily a normal distribution. The null hypothesis ( $H_0$ ) states that the median of the objective values in the two groups A and B are equal. The alternative hypothesis ( $H_1$ ) is a statement about the nonequivalence of the medians for the two penalty methods or the mean of the ranks of the two groups are not equal. This is referred to as the non-directional alternative hypothesis and is evaluated using the two tailed test. Another alternative hypothesis ( $H_1$ ) is that the median of group A is less than the median of group B. This is referred to as the directional alternative hypothesis and is evaluated using the one tailed test.

$$\begin{aligned}
 H_0 &: \Pr(A < B) = 0.5 \\
 H_1 &: \Pr(A < B) \neq 0.5 \quad (\text{or}) \\
 H_1 &: \Pr(A < B) > 0.5 \quad (\text{or}) \\
 H_1 &: \Pr(A < B) < 0.5
 \end{aligned}
 \tag{3.22}$$

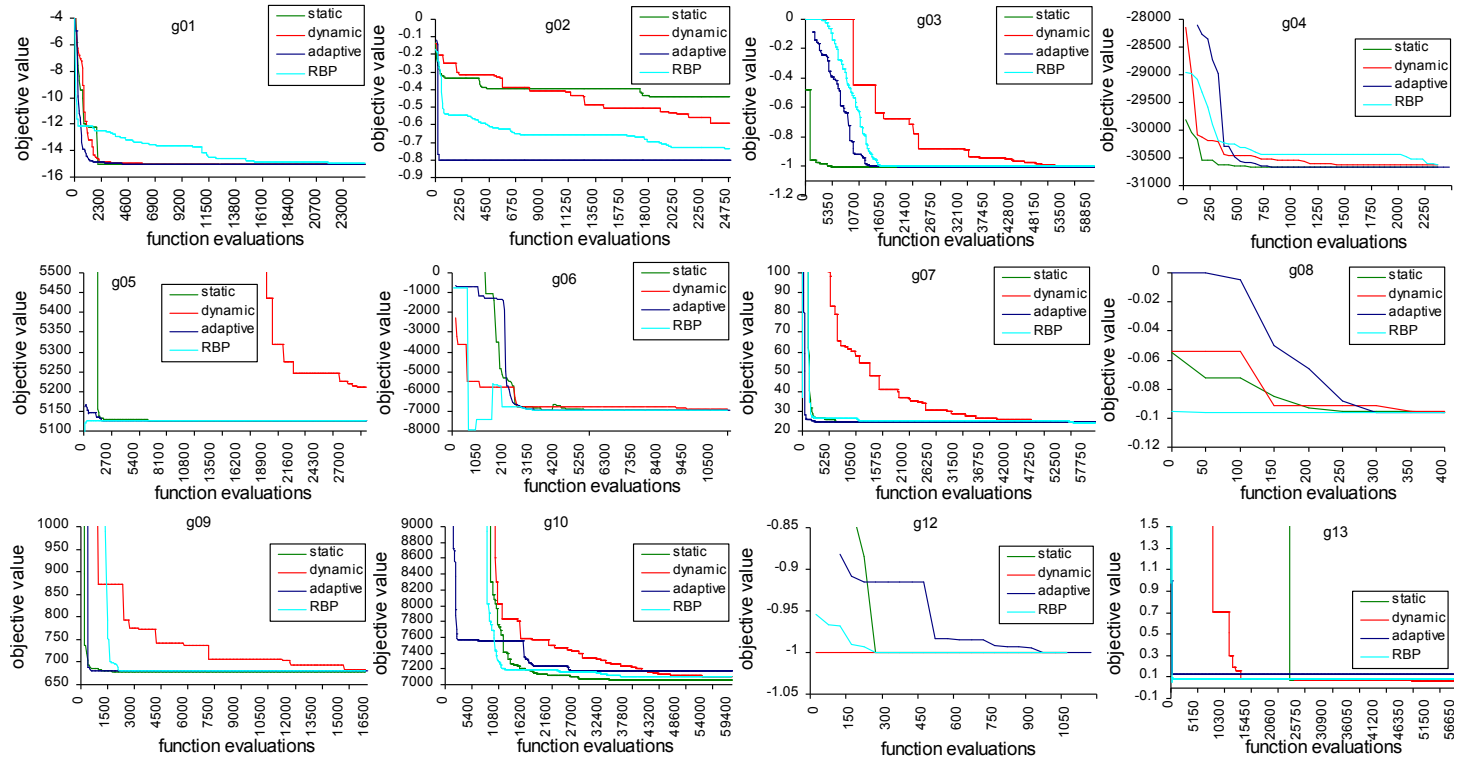


Fig. 3.6 Convergence properties of the four penalty techniques

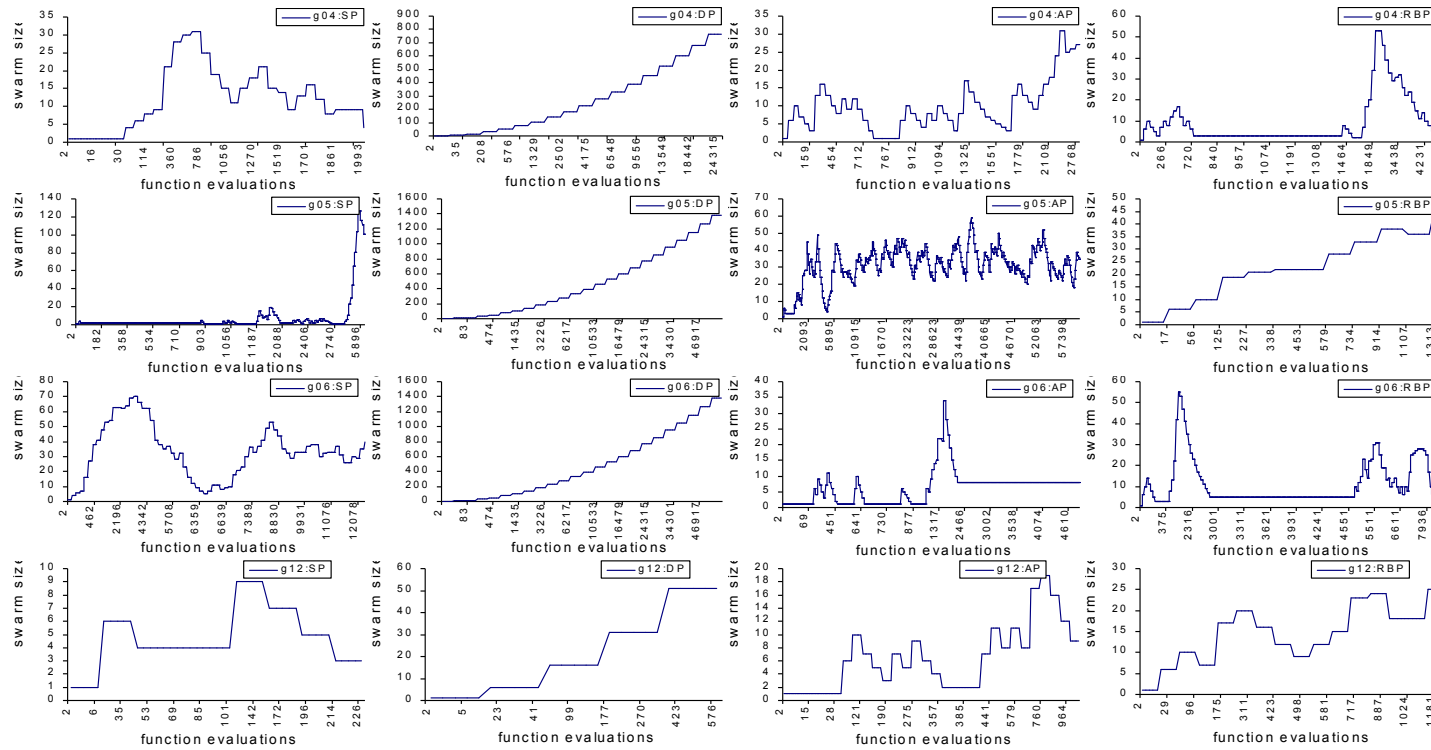


Fig. 3.7 Swarm evolution in various penalty methods for g04, g05, g06 and g12

The scores from the two groups A and B are combined into one list and ranked from lowest to the highest. The sum of the ranks for the scores in each group is computed. The Mann-Whitney  $U$  statistic is then determined for each group using the sample sizes and the rank sums. The smaller of the two  $U$  values is referred to as the  $U$  statistic. Since the sample sizes are large, a normal approximation of the Mann-Whitney  $U$  statistic namely  $z$  is computed. Comparing the calculated absolute  $z$ -value to a tabled critical value at the pre-specified level of significance (alpha value) yields the  $p$ -value. The alpha level is set at 5% i.e. by chance; five out of 100 simulations would have a statistically significant difference in the median. The  $p$ -value indicates the probability that the observed  $z$ -value is as large or larger by chance under the null hypothesis. The tabled critical two-tailed values at 5% and 10% alpha level are  $z_{0.05}=1.96$  and  $z_{0.01}=2.58$ , and the tabled one-tailed critical values are  $z_{0.05}=1.65$  and  $z_{0.01}=2.33$ . The null hypothesis can be rejected if the  $p$ -value is less than the alpha value. The null hypothesis can also be rejected if in the alternative non-directional hypothesis, the absolute  $z$ -value is greater than or equal to the tabled critical two-tailed value at the prescribed alpha level. Similarly the null hypothesis can be rejected if in the alternative directional hypothesis, the absolute  $z$ -value is greater than or equal to the tabled critical one-tailed value at the prescribed alpha level. The results of the Mann-Whitney  $U$  performed on AP against RBP are listed in Table 3.7. The null hypothesis is true for g01, g03, g05, g07, g08, g11 and g12. Both AP and RBP show similar performance on all these test functions which include highly constrained problems such as g01 and g07, and highly discontinuous solution space as in g12. For the rest of the test functions,  $p$ -value is very small indicating that it is unlikely that the results from both the methods will be identical. The performance on the other test functions can be evaluated using the directional alternative hypothesis. For g02 and g13, the directional alternative hypothesis  $H_1$ :  $\Pr(A < B)$  is valid at the 0.05 and 0.01 level because the absolute  $z$ -value is greater than the critical one-tailed value. The data are also consistent with this alternative hypothesis i.e. the mean rank of AP is less than the mean rank of RBP. So AP performs better on these test functions. However for g04, g06, g09 and g10, the directional alternative hypothesis:  $H_1$ :  $\Pr(A > B)$  is valid both at 0.05 and 0.01 level. The absolute  $z$ -value is greater than the critical values and the data were consistent with this alternative hypothesis. Since the optimization is a minimization problem, the smaller the median better is the solution. So RBP performs relatively well on all these test functions. This probabilistic analysis is contradictory to the conclusions drawn from the mean, quartiles and the standard deviation. This analysis proves that on an average RBP performs relatively better than AP. The results of Mann-Whitney  $U$  test performed on DP against AP and RBP are shown in Table 3.8. The  $p$ -values are significant for g05, g08, g11 and g12. So all the three methods, exhibit similar performances on these test functions. From the results shown in Table 3.8, it is evident that the probability of attaining better solution using DP is higher than AP or RBP for g01, g02, g07 and g10. For g03, g06 and g09, both AP and RBP perform better than DP. For g04, DP loses the performance race to RBP. In g13, DP

performs similar to AP but is far advanced compared to RBP. This nonparametric hypothesis testing proves that DP performs far better than AP or RBP in terms of the quality of the final solution. However as discussed before, DP had very slow convergence and explosive swarm evolutionary strategies. Table 3.9 lists the test

**Table 3.7** Results of Mann-Whitney U test performed with AP against RBP

function	mean ranks		RBP	
	AP	RBP	z	p
g01	107.4	93.6	-1.75	0.080392
g02	68.7	132.3	-7.77	7.99E-15
g03	100.0	101.0	-1.00	0.317311
g04	150.5	50.5	-12.83	0
g05	95.2	105.8	-1.29	0.197976
g06	148.5	52.5	-12.47	0
g07	98.7	102.3	-0.44	0.660947
g08	101.0	100.0	-1.00	0.317311
g09	120.8	80.2	-4.96	7.15E-07
g10	112.3	88.7	-2.88	0.003967
g11	101.0	100.0	-1.00	0.317311
g12	100.5	100.5	0.00	1
g13	85.5	115.5	-3.66	0.000248

**Table 3.8** Results of Mann-Whitney U test performed with DP against AP and RBP

	mean ranks		AP		mean ranks		RBP	
	DP	AP	z	p	DP	RBP	z	p
g01	58.1	142.9	-11.01	0	70.1	130.9	-8.24	2.22E-16
g02	89.2	111.8	-2.77	0.00559	59.3	141.7	-10.07	0
g03	138.5	62.5	-10.64	0	138.3	62.7	-10.56	0
g04	50.5	150.5	-12.41	0	121.4	79.6	-6.35	2.22E-10
g05	104.9	96.1	-1.07	0.28429	97.2	103.8	-0.81	0.415585
g06	148.8	52.2	-11.80	0	150.4	50.6	-12.96	0
g07	72.8	128.2	-6.77	1.32E-11	74.6	126.4	-6.33	2.49E-10
g08	101.0	100.0	-0.59	0.55783	101.5	99.5	-1.42	0.15626
g09	131.6	69.4	-7.61	2.8E-14	135.3	65.7	-8.51	0
g10	64.7	136.3	-8.75	0	74.5	126.5	-6.36	2.01E-10
g11	101.5	99.5	-1.42	0.156255	102.0	99.0	-1.73	0.083778
g12	101.0	100.0	-1.00	0.317311	101.0	100.0	-1.00	0.317311
g13	103.3	97.8	-0.67	0.501126	87.9	113.2	-3.10	0.001956

**Table 3.9** Results of Mann-Whitney U test performed with SP against DP, AP and RBP

mean ranks		DP		mean ranks		AP		mean ranks		RBP	
SP	DP	z	p	SP	AP	z	p	SP	RBP	z	p
118.8	82.2	-5.67	1.41E-08	88.9	112.1	-2.89	0.0038	99.4	101.6	-0.28	0.77733
148.0	53.0	-11.60	0	143.7	57.4	-10.55	0	119.4	81.6	-4.62	3.83E-06
62.7	138.3	-10.56	0	101.0	100.0	-1.00	0.3173	100.5	100.5	0.00	1
76.2	124.8	-7.65	2.09E-14	50.5	150.5	-13.00	0	97.0	104.0	-1.98	0.04746
115.9	85.1	-3.76	0.00017	106.7	94.3	-1.53	0.1267	104.2	96.8	-0.91	0.36212
55.4	145.6	-11.47	0	59.2	141.8	-10.50	0	106.6	94.4	-3.00	0.0027
112.7	88.3	-2.99	0.00279	87.1	113.9	-3.29	0.0010	86.1	114.9	-3.51	0.00045
99.5	101.5	-1.42	0.15626	100.0	101.0	-1.00	0.3173	100.5	100.5	0.00	1
50.5	150.5	-12.22	0	50.5	150.5	-12.22	0	50.5	150.5	-12.21	0
101.2	99.8	-0.17	0.86419	65.4	135.6	-8.57	0	75.6	125.4	-6.08	1.17E-09
101.0	100.0	-0.46	0.64495	102.0	99.0	-1.74	0.0817	102.5	98.5	-2.00	0.04549
100.5	100.5	-0.01	0.99434	101.0	100.0	-1.00	0.3173	101.0	100.0	-1.00	0.31731
113.6	87.4	-3.19	0.00140	115.6	85.4	-3.70	0.0002	98.6	102.4	-0.46	0.64645

results conducted on SP against DP, AP and RBP. The null hypothesis is true for g08, g12 and g11. For g01, the null hypothesis is true with RBP but the probability of attaining optimal solution better than DP is low. In g02, SP is inferior to all the methods. SP performs similar to AP and RBP but inferior to DP in g03. SP is the winner for g04, g09 and g10. DP overrules the performance of SP in g05. For g06, SP performs better than DP and AP but RBP is the final winner. Both SP and AP overtake SP in terms of performance in g13. The above analysis proves that SP is not a good alternative to solve the constrained optimization compared to the other three penalty techniques.

### 3.5 Summary

The constrained optimization problems are solved using the penalty function approach. Although it is quite easy to implement these techniques in evolutionary algorithms, it poses the problem of defining the optimal penalty coefficients. The choice of these parameters is problem dependent and significantly controls the quality of the optimal solutions. In order to overcome this problem, a self learning penalty technique was proposed. This penalty technique is completely free from penalty coefficients. However this method requires the normalized quantities for the fitness and constraint violations. The process of dynamic normalization, introduces dead zones. Important information may be lost in the translation. So a completely new penalty technique called the rule based method is introduced. There are no explicit penalty functions in this approach. However a set of rules controls

the search process. It is completely free from parameter tuning and no normalization is required.

The performance of the two proposed penalty technique was compared to two other penalty techniques. Standard statistical measures like the mean, quartile and standard deviation are used for comparison in terms of optimal solution, evaluation required to obtain the first feasible solution, swarm size and maximum function evaluation required for convergence. The ability of the proposed approaches to trace a feasible solution in any environment is verified using the weibull probability analysis. A nonparametric hypothesis testing was performed to validate the probabilistic performance of the new approaches. The results indicate the robustness of the proposed approaches. Both RBP and AP have better ability to find a feasible solution in minimum function evaluations. The mean swarm size was also minimal compared to the other penalty approaches. RBP has the capability to generate higher percentage of feasible evaluations and could therefore return high quality solutions compared to SP, DP and AP. The rule based penalty approach can be an effective technique in designing a black box model for solving the constrained optimization problems.





## Chapter 4

### Unit Commitment Problem

#### 4.1 Introduction

Unit commitment (UC) is the process of selecting the most effective generating units at each hour of the scheduling period to meet the system demand and reserve requirements subjected to a set of constraints. This represents a combinatorial optimization problem. The best combination of the units has to be selected from a huge number of feasible solutions. The economic dispatch problem (EDP) ensures that the demand and reserve is economically dispatched among the committed units. The unit commitment problem (UCP) is the combined operation of UC and EDP. The UCP is therefore a nonlinear mixed integer optimization problem. This is the most significant optimization task in the operation of the power systems. Solving the UCP for large power systems is computationally expensive. The complexity of the UCP grows exponentially with the number of generating units.

Many optimization methods have been proposed to provide quality solutions to the UC problem and increase the potential savings of the power system operator. These include deterministic and stochastic search approaches. Deterministic approaches include the priority list method [40], dynamic programming [41], Lagrangian Relaxation [42] and the branch-and-bound methods [44]. Although these methods are simple and fast, they suffer from numerical convergence and solution quality problems. The stochastic search algorithms such as particle swarm optimization [45]-[48], genetic algorithms [49], evolutionary programming [50], simulated annealing [51], ant colony optimization [52] and tabu search [53] are able to overcome the shortcomings of traditional optimization techniques. These methods can handle complex nonlinear constraints and provide high quality solutions. However, all these algorithms suffer from the curse of dimensionality. The increased problem size adversely effects the computational time and the quality of the solutions.

A new UCP formulation is used to address this dimensionality problem. The UCP formulation so far uses binary variables to represent the hourly generator schedule. However the new approach accumulates the identical successive binary bits and represents it as an integer. This integer describes the operation of the units over a period of time. So each unit has a set of integers to represent its scheduling status at each hour of the planning period. This formulation drastically reduces the number of decision variables and hence can overcome the shortcomings of stochastic search algorithms for UC problems. The merits of variable reduction are reflected in the quality and convergence of the optimal solution. This chapter investigates the effectiveness of the new UCP formulation compared to the conventional binary UCP formulation.

## 4.2 Problem Formulation

The objective of the UC problem is to minimize the total operating costs subjected to a set of system and unit constraints over the scheduling horizon. The generalized objective function of the UCP can be expressed as a summation of the operating costs of all the generating units:

$$\text{Minimize } \sum_{t \in T} \sum_{i \in I} (F_{it}(P_{it})U_{it} + SC_{it}(T_{it})U_{it}(1 - U_{it-1})) \quad (4.1)$$

The operating costs include the cost of fuel for generating power,  $P_{it}$  and start-up cost. The commitment decision of a unit is described by the binary variable  $U$ .  $U_{i,t} = 1$  if unit  $i$  is committed at time  $t$ , otherwise  $U_{i,t} = 0$ . All the generators are assumed to be connected to the same bus supplying the total system demand. Therefore, the network constraints are not taken into account. The power production cost  $F_{it}(P_{it})$  of the committed units is assumed to be a quadratic function of the electrical power generation:

$$F_{it}(P_{it}) = \alpha_i + \beta_i P_{it} + \gamma_i P_{it}^2 \quad (4.2)$$

where  $\alpha_i$ ,  $\beta_i$  and  $\gamma_i$  represent the cost coefficients of generating unit  $i$ ,  $I$  is the set of all generation resources and  $T$  is the planning period. Certain energy is expended to bring the thermal units on-line. The cost of this energy is termed as start-up cost. The thermal units during their down time can be allowed to cool or bank based on the economic aspects. The generator start-up cost for cooling,  $SC_{it}$  depends on the down time of the unit. The start-up costs are assumed to be maximum for cold start and vary exponentially with the down time of the unit. The general form of the start-up cost is as follows:

$$SC_{it} = \sigma_i + \delta_i \left( 1 - \exp\left(\frac{-T_{it}^{off}}{\tau_i}\right) \right) \quad (4.3)$$

where  $\sigma_i$  is the hot start-up cost,  $\delta_i$  the cold start-up cost,  $\tau_i$  is the cooling time constant and  $T_{it}^{off}$  is the downtime of unit  $i$  at time  $t$ .

### 4.2.1 Constraints in Unit Commitment

The power system operator may include several constraints on the generating units depending on his forward contracts and reliability requirements. The constraints that are considered in this work may be classified into two main groups:

#### 4.2.1.1 System Constraints

##### *Power balance*

The total generated power at each hour must balance the load at the corresponding hour,  $D_t$ .

$$\sum_{i=1}^N P_{it} U_{it} = D_t \quad \forall t \in T \quad (4.4)$$

##### *Spinning reserve*

Spinning reserve is the total generation capacity of all the synchronized units in the power system, minus the load and losses being supplied. The reserve capacity helps power systems to overcome the unscheduled generator outages and major load forecast errors without load shedding. The reserve allocation is usually done with respect to certain reliability measures. There are several ways to determine the reserve capacity. The reserve capacity is usually determined as a certain percentage of the forecasted peak demand or large enough to make up for the loss of the largest generating unit. But this reserve comes at a cost. Therefore the reserve allocation should be an optimal compromise between economic and reliability issues.

$$\sum_{i \in I} P_i^{\max} U_{it} \geq D_t + R_t \quad \forall t \in T \quad (4.5)$$

#### 4.2.1.2 Unit Constraints

##### *Generation limit constraint*

Each generator has a certain operating range termed as the minimum and maximum generation states. These levels may vary over time due to maintenance or unscheduled outages of various equipments in the plant.

$$P_i^{\min} \leq P_{it} \leq P_i^{\max} \quad \forall i \in I, \forall t \in T \quad (4.6)$$

##### *Minimum up/down time constraints*

A thermal unit requires certain time either to shut down (cool down) or start from a certain temperature to the on-line mode. These requirements are formulated as minimum up/down time constraints. These constraints indicate that a unit must be on/off for a certain number of hours before it can be shut off or brought online.

$$T_{it}^{on} \geq MUT_i \quad \forall i \in I, \forall t \in T \quad (4.7)$$

$$T_{it}^{off} \geq MDT_i \quad \forall i \in I, \forall t \in T \quad (4.8)$$

$T_i^{on}$  &  $T_i^{off}$  represents the up-time and down-time of the  $i^{th}$  unit respectively and  $MUT_i$  &  $MDT_i$  are the minimum up-time and down-time respectively. Finally the initial unit states at the start of the scheduling period must also be taken into account.

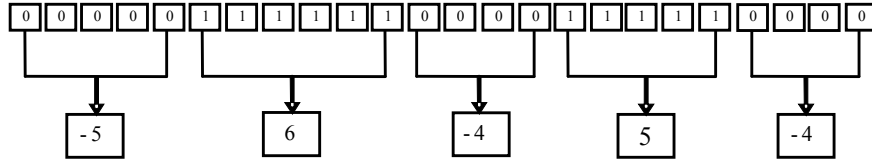
### 4.3 Particle Formulation

#### 4.3.1 Binary Coding

The UC variables representing the on/off status of the generators at each hour of the planning period are represented by binary bits. In order to represent the scheduling of a single generator for 24 hours, 24 binary decision variables [54] are required. So solving UC for large power systems requires extremely huge number of UC binary variables.

#### 4.3.2 Integer Coding

The solution techniques for unit commitment developed so far use binary programming. The particles are encoded as binary strings. There have been advancements in the solution techniques but no attempt so far is made to reduce the number of decision variables. The new approach concatenates the identical successive binary bits. The point of time schedule representation is changed to period of time representation. The UC variables no more has a unique variable to represent the status of the generator at each time step but instead has a variable that's gives information about the generator over a period of time. The constricted strings are represented by integers as shown below in Fig. 4.1. The unit commitment schedule in Fig. 4.1 is represented by five integers. Each integer represents the continuous on/off period of a generating unit. Negative integer indicates the period when the unit is switched off and positive integer indicates the period when the unit is in operation. The number of integers required to represent the scheduling decisions should be defined before the start of the optimization. For a system with  $I$  generating units and  $D$  integers representing the UC schedule of each unit, the particle consists of  $I*T$  continuous variables representing the power generation levels of units at each hour and  $I*D$  integer variables representing UC schedule of the units at each hour, whereas the binary coded particle requires  $I*T$  continuous variables and  $I*T$  binary variables. The new variable formulation would reduce the number



**Fig. 4.1** The new UC variable formulation

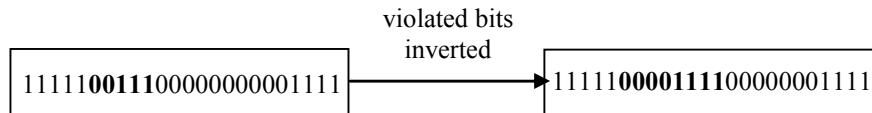
of UC decision variables by  $(T-D)/D\%$ . The five integer encoding scheme for representing UC schedule of generators results in 80% reduction in the number of UC decision variables.

## 4.4 Binary Programming

Evolutionary programming handles the binary variables using the Sigmoid function [55]. This function generates a probability for each binary bit. This probability decides the transition of the binary decision variables. The binary variables are therefore controlled individually and there is no collective strategy to influence a set of binary variables. Therefore satisfying the *MUT/MDT* constraints in UCP which demands that a certain portions of the binary string should be identical is very difficult. These constraints should be handled using special solution techniques. There are many ways to treat a particle that disobeys these constraints. The most popular ones are repair strategy, mutation and generation.

### 4.4.1 Repair Strategy

This strategy repairs the infeasible particle that violates the *MUT/MDT* constraints [56]-[57]. This is done by evaluating the status of the generators at each hour. If the current status of any unit violates the constraints then the corresponding binary variable is inverted. The repair strategy is performed on each unit starting from hour 0 until the planning period. Fig. 4.2 explains this procedure. The UC schedule shown in the figure corresponds to a unit with a *MUT* and *MDT* requirement of four hours. The infeasible solution on the left is evaluated at each time step. The unit is down at hour 6 and 7 and up again at hour 8. However the unit has a minimum down time of four hours. So the up status at hour 8 and 9 are inverted to match the *MDT* constraint. The other bits are similar inverted to generate a feasible solution.



**Fig. 4.2** Repair operator

#### 4.4.2 Mutation Operator

This operator mutates the binary bits on either side of the violated bits to convert an infeasible solution to a feasible one [58]. The choice of left or right mutation is random. This process is shown in Fig. 4.3.

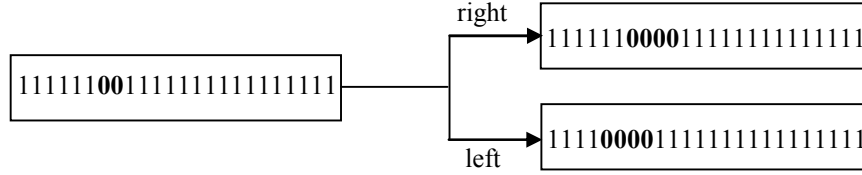


Fig. 4.3 Mutation operator

#### 4.4.3 Generation

In this strategy infeasible solution are replaced by randomly generating a feasible solution. The optimization process deals only with the feasible solutions.

These special operators can successfully repair the infeasible particles that violate the *MUT/MDT* constraints. However these bit inversion might cause the demand or reserve imbalance and thus degrade the performance index of the particle. Random generation of a feasible particle may not always be an easy task. So, better strategies that can collectively satisfy the various constraints are required.

### 4.5 APSO Approach

The Adaptive Particle Swarm Optimization mentioned in the previous chapters is slightly modified to accommodate the new UC formulation. The UC integer variables representing the operating schedule of a unit  $i$  are listed as  $U_i^j, j = 1, 2, \dots, D$ . The adaptations to the original version of APSO are explained in the following steps:

#### 4.5.1 Swarm Initialization

The unit commitment integer variables of each particle are randomly generated within the specified limits. The UC integer variables are generated as shown in 4.9. The function `rand_int` generates random integer numbers. The UC variables can be initialized between  $(-T, T)$ . But generating random numbers within a huge range might lead to a situation where the first set of UC integers may have high operation periods and the last set of variables may be set to zero operation time.

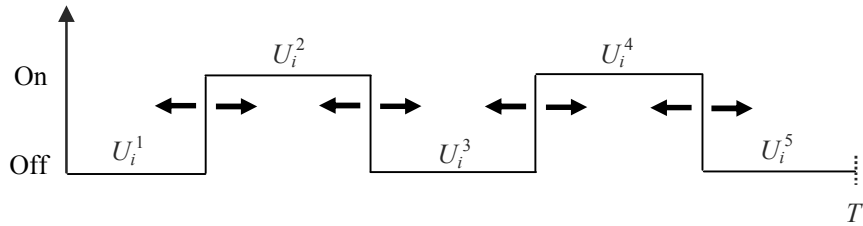
$$U_i^j = \begin{cases} \text{rand\_int}\left(-\frac{T}{D-1}, \frac{T}{D-1}\right), & j=1, \dots, D-1 \\ T - \sum_{j=1}^{D-1} U_i^j, & j=D \end{cases} \quad (4.9)$$

The  $P$  integer set may be underutilized to represent the operation of a unit. The initialization described in (4.9) has constricted interval for random integer generation and this will ensure that none of the UC integers are zero. The randomly initialized UC variables are validated for their  $MUT/MDT$  requirements (4.10). So the  $MUT/MDT$  constraints are included in the particle initialization.

$$U_i^j = \begin{cases} U_i^j & \text{if } U_i^j \text{ is +ve \& } U_i^j \geq MUT \\ -U_i^j & \text{if } U_i^j \text{ is -ve \& } |U_i^j| \geq MDT \\ -MDT & \text{if } U_i^j \text{ is -ve \& } |U_i^j| < MDT \\ MUT & \text{if } U_i^j \text{ is +ve \& } U_i^j < MUT \end{cases} \quad (4.10)$$

#### 4.5.2 Velocity Update

The UC integer variables are updated like any other continuous variables. The selection of a suitable update strategy for a particular particle is similar to that explained in chapter 2. The update procedure for the UC variables can be explained by the following figure.



**Fig. 4.4** Update procedure for UC variables of a single generator with five operating periods

The operation of a unit is represented by five integers. The integers  $U_i^1, U_i^3, U_i^5$  represent the down time operation periods and  $U_i^2, U_i^4$  represent the up time operation period. To each integer or duty cycle a certain velocity is added. The arrows indicate the direction of velocity. The left arrow implicates a negative velocity which causes a reduction in the corresponding integer or operation period.

Where as a right arrow indicates a positive velocity which helps to increase the duty cycle or the corresponding integer. There is no need to assign a sigmoid transition probability or mutation probability for altering the on/off status of the units. The UC integer variables are treated like any other decision variables. The maximum velocity is curtailed to  $(-T/D, T/D)$ . Since the variables describe the operation of the unit over a period of time, it is quite easy to satisfy the  $MUT/MDT$  constraints. This new formulation will assist the swarm to track feasible regions much faster than the binary coding approach. After updating the integer variables they are ceiled to the nearest integers.

#### 4.5.3 Additional Constraints

The new UCP formulation has an additional constraint apart from (4.4) – (4.8) and special rules to assist the search procedure. The variables are free to assume any value between  $(-T, T)$  but the sum of all the integer UC variables corresponding to each unit must be equal to the scheduling period  $T$ .

$$\sum_{j=1}^D |U_i^j| = T \quad \forall i \in \{1, \dots, I\} \quad (4.17)$$

This constraint is implemented as an internal constraint and violation of this constraint is corrected by the following rules. This constraint can also be defined as an external constraint similar to the demand or reserve constraint.

$$\text{if} \left( \sum_{j=1}^D |U_i^j| < T \right) \text{ Rule 1: } U_i^D = \pm \left( T - \sum_{j=1}^{D-1} |U_i^j| \right) \quad (4.18)$$

$$\text{if} \left( \sum_{j=1}^D |U_i^j| > T \right) \left\{ \begin{array}{l} \text{if}(|U_i^1| > T) \quad \text{Rule 2: } U_i^1 = \pm T \\ \text{if}(|U_i^1| + |U_i^2| > T) \quad \text{Rule 3: } U_i^2 = \pm(T - |U_i^1|) \\ \dots\dots\dots \\ \text{if} \left( \sum_{j=1}^D |U_i^j| > T \right) \quad \text{Rule } D+1: U_i^D = \pm \left( T - \sum_{j=1}^{D-1} |U_i^j| \right) \end{array} \right. \quad (4.19)$$

The new approach facilitates considerable reduction in the number of UC variables. This approach will enhance the swarm to trace feasible regions even in complex UCP problems with ease. Since the new framework is able to control the operation states of several hours in a single variables, complex constraints regarding the units operation can easily be handled.



## 4.6 Special Convergence Operators

Certain problem dependent operators can be defined to speed up the convergence of the UCP. These operators help the optimization to satisfy the most significant demand and the reserve constraints. The search algorithm will be assisted in discovering the feasible space. The swarm therefore will concentrate more on the feasible space. In the process these operators contribute to finding quality solutions. Two such operators are explained below:

### 4.6.1 Demand Equalizer

The demand equality constraint is handled by this operator. The demand-generation balance is done in two steps. In the first step the total generation is altered to be either equal or greater than the demand. If there is a generation deficit the most economical non-committed unit is brought into operation. The selection process also ensures that the committed units do not violate the MDT constraints. The excess generation allotted during this process is reduced in the second step. The imbalance is distributed among all the committed units. The economical units share a higher imbalance power. This operator comes into operation only when the swarm is unable to find a suitable feasible space.

### 4.6.2 Reserve Manager

This operator ensures that no unwanted reserve is allocated. The process monitors if the excess reserve is greater the maximum generation capacity of any of the committed units. The units so found are decommitted. The new operation schedule is validated for its feasibility with respect to demand-generation balance and up-time and down-time requirements of the units. In case of infeasibility the reserve manager decisions are reset to the default original schedule. This operator functions during the entire iterative search process.

## 4.7 Numerical Example

The proposed approach was implemented with APSO algorithm. The efficiency of this technique was tested on a bench-test UCP consisting of ten generators [45]. The scheduling horizon was set to 24 hours. The problem data for this test case is given in Table 4.1 and 4.2. The spinning reserve is assumed to be 5% of the load demand. Test results are compared with the binary programming approach presented in [45]. In binary coded UC formulation the particle consists of  $10 \times 24$  continuous variables representing the generation levels of the ten units and  $10 \times 24$

**Table 4.1** Test System Data

Unit	$p_{(MW)}^{\max}$	$p_{(MW)}^{\min}$	Fuel cost			Start-up cost			$MDT$ (hrs)	$MUT$ (hrs)	$INS$ (hrs)
			$a$	$b$	$c$	$\sigma$	$\delta$	$\tau$			
1	455	150	1000	16.19	0.00048	4500	4500	4	5	5	8
2	455	150	970	17.26	0.00031	5000	5000	4	5	5	8
3	130	20	700	16.60	0.00200	550	550	2	2	2	-5
4	130	20	680	16.50	0.00211	560	560	2	2	2	-5
5	162	25	450	19.70	0.00398	900	900	2	2	2	-6
6	80	20	370	22.26	0.00712	170	170	2	2	2	-3
7	85	25	480	27.74	0.00079	260	260	2	1	1	-3
8	55	10	660	25.92	0.00413	30	30	1	0	0	-1
9	55	10	665	27.27	0.00222	30	30	1	0	0	-1
10	55	10	670	27.79	0.00173	30	30	1	0	0	-1

**Table 4.2** Hourly Load Demand

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MW)	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MW)	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

binary variables representing the operation schedule of the units at each hour. Whereas, the new approach, the particle consists of 10\*24 continuous variables and 10\*5 integer variables.

The Optimal schedule obtained by the binary programming approach is listed in Table 4.3. The total operating costs amounts to €565,804 [45]. Whereas using the new approach, the optimal costs are €561,586. The final UC schedule of the new approaches is presented in Table 4.4. The UC schedules imply that the new approach exploits the search space far better than the binary APSO approach. The new approach was able to optimally utilize the various units and reduce their idle operation periods. This was significantly visible in the schedules corresponding to units 3, 4 and 5. In binary approach these units operate for complete 24 hours. Whereas in the new approach the units come into operation, only during the peak operation periods and were shut down during the off-peak periods. This efficient utilization of the resources was reflected in the cost reduction.

The quality of the solution obtained by the new approach can also be examined using the reserve allocation shown in Fig. 4.5. In binary method, there was huge unwanted reserve allocation during the low demand hours. When there was a sudden reduction in demand between hours 15:00 to 19:00, the algorithm was unable to reschedule the units efficiently. The units that are in operation during the first peak demand continue to operate even during this sudden dip. The excess unwanted re-

serve allocated during this period comes at a price and accounts for the difference in the operation costs for the two methods. In the new approach there was optimal reserve allocation. The reserve allocation by the new approach was very close to the required reserve capacity.

The convergence of the new approach is shown in Fig. 4.6. The algorithm converges much faster than binary approach. The swarm was able to trace the global optimal space within a few iterations. This is observed as a steep decline in the costs during the first 10000 function evaluations. However the first feasible solution starts quite far away from the global final solution. This can be improved by using the special convergence operators. The improved convergence is shown in Fig. 4.7. The search process starts in a near optimal region. These operators will

**Table 4.3** Optimal UC Schedule using Binary Programming Approach

Unit	Unit schedule
1	11111111111111111111111111111111
2	11111111111111111111111111111111
3	11111111111111111111111111111111
4	00111111111111111111111111111111
5	11111111111111111111111111111111
6	0000000011111111000011000
7	000000000111100000010000
8	000000000011000000000000
9	000000000001000000000000
10	000000000000000000000000
OC(€)	565,450

**Table 4.4** Optimal UC Schedule using New Approach

Unit	Unit schedule				
	1	2	3	4	5
1	24	0	0	0	0
2	24	0	0	0	0
3	-4	10	-4	4	-2
4	-3	12	-2	6	-1
5	-6	15	-3	0	0
6	-8	8	-3	2	-3
7	-9	4	-6	1	-4
8	-10	2	-12	0	0
9	-11	1	-12	0	0
10	-24	0	0	0	0
OC(€)	561,586				

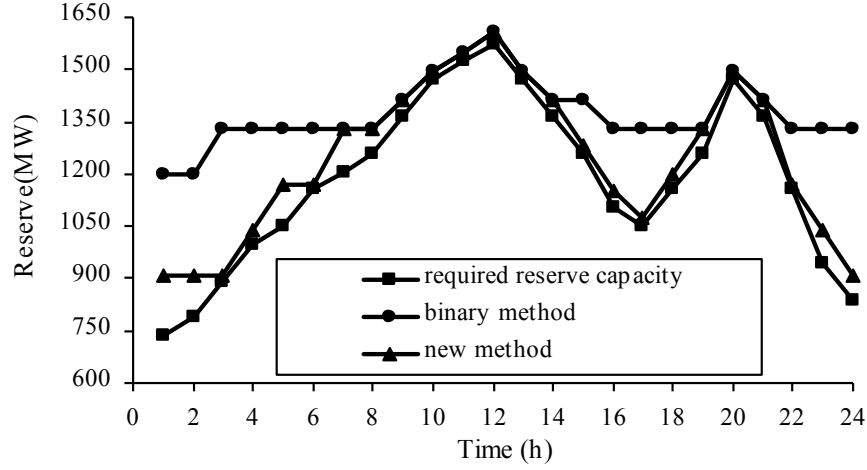


Fig. 4.5 Reserve allocation by binary programming and the new approach

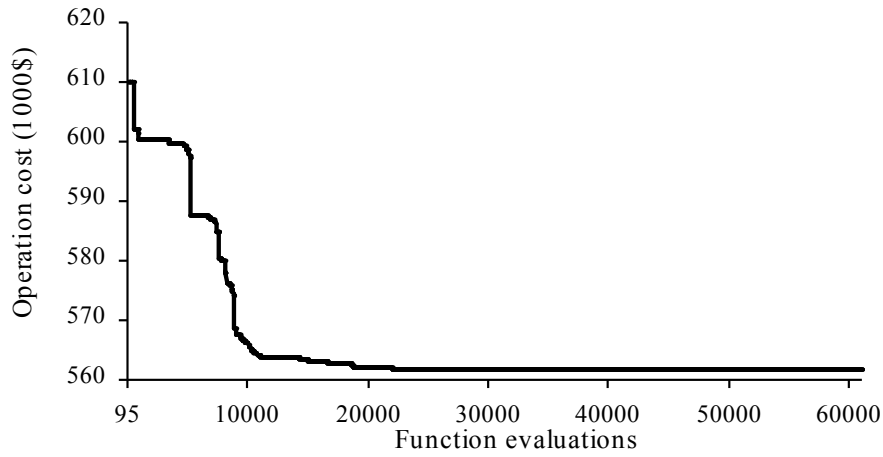
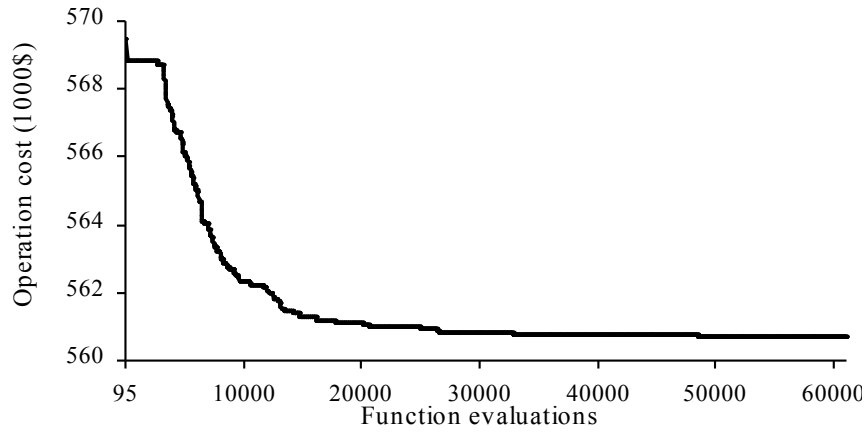


Fig. 4.6 Convergence tendency of the new approach

feed important information to the swarm regarding the optimal space. So the swarm spends more time in the feasible area and is able to perform better. These operators besides assisting the swarm to converge faster also help to fine tune the final solution. The generation schedule obtained by incorporating these operators into the new approach is shown in Table 4.5. These operators further improve the rescheduling process during the sudden demand dips. The operation of the idle generators during the off-peak demand periods is further reduced and this is reflected in the total operating cost which amounts to €560,675. The reserve allocation as shown in Fig. 4.5 is very similar to the previous simulation without the operators. The production and start-up costs and the generation schedule of generating unit at each hour of the 24 hour planning period is given in Table 4.6.



**Fig. 4.7** Convergence tendency of the new approach with convergence operators

**Table 4.5** Optimal UC Schedule using New Approach and special operators

Unit	Unit schedule				
	1	2	3	4	5
1	24	0	0	0	0
2	24	0	0	0	0
3	-6	8	-5	3	-2
4	-5	10	-2	6	-1
5	-3	18	-3	0	0
6	-10	6	-2	3	-3
7	-9	4	-6	1	-4
8	-11	1	-12	0	0
9	-8	4	-12	0	0
10	-24	0	0	0	0
OC(€)	560,675				

## 4.8 Conclusion

The new UC variable formulation causes considerable reduction in the number of decision variables and thus the size of optimization problem. The reduced parameter space enables the swarm to discover new promising feasible regions. This tool will be a good alternative to solve large complex UCP problems. The swarm is able to explore efficiently using the new approach. The fast convergence and high quality solutions obtained indicate the strength of the new proposed approach. Special convergence operators are used to further improve the convergence properties of the algorithm.

**Table 4.6** Optimal Operating costs and Generation Schedule

Hour	Production Cost(€)	Startup Cost(€)	Generation Schedule (MW)											
1	13810.60	0	312.9	387.0	0	0	0	0	0	0	0	0	0	0
2	14632.93	0	366.4	383.5	0	0	0	0	0	0	0	0	0	0
3	16336.85	0	416.5	433.4	0	0	0	0	0	0	0	0	0	0
4	18599.45	1790	454.7	454.6	0	0	40.6	0	0	0	0	0	0	0
5	19623.09	0	451.1	455.0	0	0	93.8	0	0	0	0	0	0	0
6	22230.36	1116.22	397.0	455.0	0	90.9	156.9	0	0	0	0	0	0	0
7	23277.79	1097.75	437.9	428.0	130.0	129.0	25.0	0	0	0	0	0	0	0
8	24153.62	0	454.2	454.9	29.8	130.0	31.0	0	0	0	0	0	0	0
9	26984.27	59.99	438.5	455.0	129.9	129.6	136.8	0	0	0	10.0	0	0	0
10	29904.42	519.35	451.6	455.0	130.0	130.0	161.8	0	25	0	46.4	0	0	0
11	31273.83	339.74	452.4	455.0	129.8	129.9	162.0	80.0	30	0	10.0	0	0	0
12	33296.67	60	452.8	454.9	129.3	130.0	161.4	80.0	60	16	14.9	0	0	0
13	29374.81	0	454.6	455.0	130.0	129.7	159.3	46.2	25	0	0	0	0	0
14	26591.47	0	455.0	455.0	129.8	129.4	110.7	20.0	0	0	0	0	0	0
15	24396.87	0	436.9	454.4	0	129.9	158.5	20.0	0	0	0	0	0	0
16	21077.76	0	446.3	455.0	0	0	128.6	20.0	0	0	0	0	0	0
17	19629.39	0	449.5	455.0	0	0	95.4	0	0	0	0	0	0	0
18	21867.95	913.98	452.9	454.8	0	130.0	62.1	0	0	0	0	0	0	0
19	24374.92	277.46	442.6	453.9	0	129.7	153.6	20.0	0	0	0	0	0	0
20	29471.09	1561.9	438.5	454.9	130.0	130.0	162.0	59.5	25	0	0	0	0	0
21	26763.23	0	451.2	453.8	129.9	129.8	55.2	79.8	0	0	0	0	0	0
22	21935.38	0	399.8	452.7	117.3	129.9	0	0	0	0	0	0	0	0
23	17851.74	0	390.5	433.1	0	76.3	0	0	0	0	0	0	0	0
24	15480.26	0	395.8	404.1	0	0	0	0	0	0	0	0	0	0
Total	552938.85	7736.43												

## Chapter 5

# Optimization under Uncertainty

### 5.1 Introduction

Deterministic optimization model formulation assumes that the problem data can either be measured or forecasted. However, for many real-world problems, this data cannot be known accurately because of the measurement errors or that the data represents information about the future. Since the outcome of any event happening in the future cannot be perfectly forecasted, the data should be considered uncertain. Deterministic models are solved using either the forecasted data or the statistical mean or median for the uncertain quantities. This procedure might result in unsatisfactory solutions. For instance consider the scheduling problem in a thermal power plant. If the power plant is operated based on the solution from a deterministic model with forecasted information for the demand, it might fail to hedge against worst case demand scenarios. During a low demand scenario, the generation units might be spinning but idle. The operation will incur additional spinning and shut-down costs for the inappropriate decisions. Where as, during a high demand scenario, expensive generation units are brought into operation in order to meet the demand. The most appropriate way to handle these uncertainties is by considering them as random variables or random processes. The group of mathematical programming that deals with uncertainty is known as stochastic programming (SP) [59], [60]. SP models can generate robust decisions that can hedge against all possible outcomes of the uncertainties.

### 5.2 Stochastic programming

SP is used to model real-world optimization problems where decisions have to be made in the presence of uncertainty. The main assumption in SP is that the probability distribution describing the nature of the uncertainties is either known with certainty or can be estimated. SP models can be classified into chance constrained problems [61], recourse problems [62], distribution problems [63] and hybrid models [64]. This chapter deals with SP models with recourse. For further information on other SP models refer to [65], [66].

In SP with recourse [67], decisions are made before the uncertainty is realized and then the expectation of all the consequences of the decisions is minimized. These programs have the ability to take corrective actions or recourse actions after the uncertainty is disclosed.

### 5.2.1 Types of SP with recourse

Based on the number of recourse stages, these programs can be classified into two-stage and multistage SP.

#### Two-Stage SP

In these programs, the decisions are made in two stages. The first stage decisions,  $\mathbf{x}$  are made without anticipating the outcome of the uncertainty,  $\omega$ . The second stage decisions,  $\mathbf{y}$  represents the recourse actions [59], [68], [69]. The sequence of events is as follows:

$$\mathbf{x} \rightarrow \zeta(\omega) \rightarrow \mathbf{y}(\mathbf{x}, \omega) \quad (5.1)$$

A standard formulation of the two-stage SP model [70] is as follows:

$$\begin{aligned} & \min \mathbf{c}^T \mathbf{x} + E_{\omega \in \Omega} [Q(\mathbf{x}, \omega)] \\ & \text{subject to} \\ & \mathbf{Ax} = \mathbf{b} \\ & \text{with} \\ & Q(\mathbf{x}, \omega) = \min \mathbf{f}(\omega)^T \mathbf{y} \\ & \text{subject to} \\ & T(\omega)\mathbf{x} + D(\omega)\mathbf{y} \geq h(\omega) \end{aligned} \quad (5.2)$$

Where  $\mathbf{x} \subseteq \mathfrak{R}^{n1}$ ,  $\mathbf{y} \subseteq \mathfrak{R}^{n2}$  and  $\omega$  is a random variable from the probability space  $(\Omega, F, P)$  with  $\Omega \subseteq \mathfrak{R}^k$ ,  $f: \Omega \rightarrow \mathfrak{R}^{n2}$ ,  $h: \Omega \rightarrow \mathfrak{R}^{m2}$ ,  $D: \Omega \rightarrow \mathfrak{R}^{m2 \times n2}$  and

$T: \Omega \rightarrow \mathfrak{R}^{m2 \times n1}$ .  $D(\omega)$  is the recourse matrix and  $h(\omega)$  is the right hand side. The objective of this program is to minimize the direct costs,  $\mathbf{c}^T \mathbf{x}$  incurred by the first stage decisions and the expectation of the recourse costs,  $Q(\mathbf{x}, \omega)$  which is a function of the first and second stage decisions. These programs generate optimal first stage decisions by satisfying the first stage constraints,  $\mathbf{Ax} = \mathbf{b}$  and suitable recourse actions by using the recourse function,  $T(\omega)\mathbf{x} + D(\omega)\mathbf{y} \geq h(\omega)$ .

#### Multi-stage SP

These programs are an extension of the two-stage and has more recourse action stages i.e. consists of several decisions making stages which respond to the evolution of the uncertainty over time [59], [68], [69], [71]. The decisions are non-anticipatory [72]. The decisions for period  $t$  depends only on the data observable



till period  $t$ . The decisions at any stage are independent of the next stage. The sequence of events for a T-stage SP as specified by Duppacova [73] can be listed as follows:

$$\begin{aligned} x_1 \rightarrow \xi(\omega_1) \rightarrow x_2(x_1, \omega_1) \rightarrow \xi(\omega_2) \rightarrow \dots \dots \dots \\ \dots \dots \dots \rightarrow \xi(\omega_{T-1}) \rightarrow x_T(x_1, \omega_1, \dots, x_{T-1}, \omega_{T-1}) \end{aligned} \quad (5.3)$$

Where  $\{x_t\}_{t=1}^T$  are the set of decisions at each stage and  $\{\omega_t\}_{t=1}^T$  is a stochastic process on the probability space  $(\Omega, F, P)$ . The mathematical formulation of the stochastic model as stated by Römisch [74] is as follows:

$$\begin{aligned} \min E \left[ \sum_{t=1}^T \mathbf{c}^T(\omega_t) x_t \right] \\ \text{such that} \\ x_t \text{ measurable w.r.t. } F_t \subseteq F \\ x_t \in X_t \\ B_t(\omega_t) x_t \geq d_t(\omega_t) \quad \forall t = 1, \dots, T \\ \sum_{\tau=1}^t A_{t\tau}(\omega_t) x_\tau \geq g_t(\omega_t) \quad \forall t = 2, \dots, T \end{aligned} \quad (5.4)$$

The model is subjected to three sets of constraints. The measurability constraints on  $x_t$ , constraints describing the relation between decisions at different stages and constraints describing the feasibility of the  $t^{\text{th}}$  stage decisions  $x_t$ .

### 5.2.2 Advantages and disadvantages of SP

Deterministic models provide solutions based on the forecasted scenario with the assumption that this scenario will occur with absolute certainty. But SP models are simulated on a wide range of probable scenarios simultaneously. SP models provide robust solutions by minimizing the ill effects of the random outcomes.

Risk models can be easily incorporated in the SP models. Therefore decision makers can mould their actions to balance their net revenue against their risk targets.

In capacity expansion, there is a lot of information regarding the future outcomes which need to be included in the decision making process. For instance, let us consider the German federal governments plan to install new offshore wind parks. Such projects require decision models which can not only investigate the environmental and economical compatibility but also incorporate factors which involve technical, economical and legal uncertainties. With the SP approach, the

decisions makers can generate optimal plans by including all these information into one decision making model.

Another advantage of SP is that the problem formulation is totally independent of the solution algorithm. This gives the chance for the development of efficient optimization algorithms for complex stochastic models.

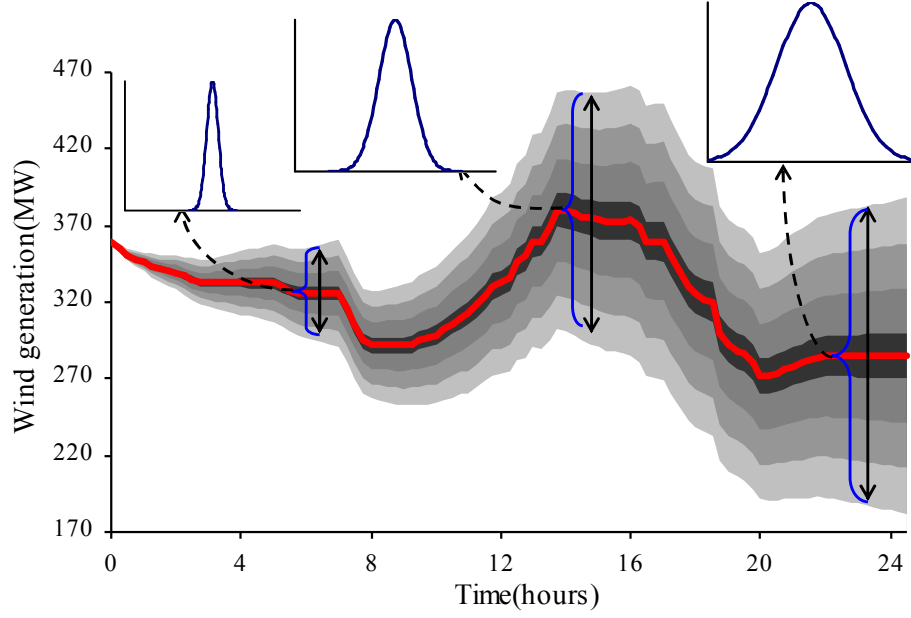
SP involves recourse actions distinct to each scenario. So there are a set of decision variables associated with each scenario. The problem dimension grows in size with the increase in the number of scenarios. Therefore a reduced order stochastic model should be used. This requires that the uncertainty description should be curtailed by suitable approximations. Even with considerable model reduction, SP models demands for an efficient solution algorithm.

### 5.3 Uncertainty Modeling

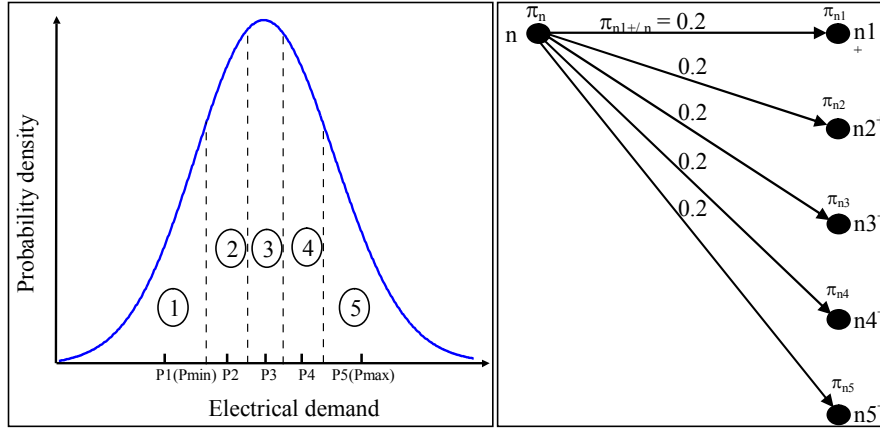
The significant uncertainties associated with planning the operation of the power systems [75] are (i) electrical and thermal demand, (ii) spot and forward price, (iii) unit outages and (iv) renewable generation capacity. Both electrical and thermal demands are characterized by seasonal changes, variation in weather, activity and life style of the end users. Daily load profiles also depend on the type of the day such as weekday/weekend and public holidays and time of the day. These dependencies are more dominant when the load models represent a small residential utilization. The scheduling of the generators is highly influenced by the spot price uncertainty. The variability in spot price is relatively high even within a short time span. Due to the lack of economical energy storages, the generation should match the demand at all times. So the power spot price is highly dependent on the marginal generation cost. So all the factors, which influence the operation cost of a power system also slave the spot price. The price spikes are strongly dependent on the load pattern and the time of the day especially when the demand is high and generation is scarce. In order to operate at a suitable unit commitment risk level, the unit outage rates should be included in the scheduling model. However for short-term UCP, the outage probabilities are very small and can be neglected. The growing concern for environment has asked for rapid development in wind and solar power generation technologies. Both of these energy sources have a special importance in German energy planning. By 2020, 20% of the power consumed in Germany will be supplied by the renewable generation. Due to the stochastic nature of the wind and solar insolation, the output capacity from these units can not be predicted accurately. With the ever increasing penetration of these energy sources, there will be huge fluctuations in the power generation. There are several sophisticated forecasting tools to predict the output from these units using the wind speed and direction, thermodynamic properties of the air masses, like density and temperature. However there is a huge error in these measurements. The prediction error increases at higher time steps due to the decrease in the accuracy of the weather forecast at higher forecast periods.

The uncertainties described above should be modelled in a suitable form so as to be included in the SP models. In SP, the randomness of a stochastic variable at any time stage is described by a continuous probability density function. The difficulty in solving the stochastic models with these continuous functions lies in evaluating the integrals. In order to overcome this issue, the continuous distributions are approximated to discrete distributions. Each possible outcome of the random variable is a scenario. In multi-stage, the uncertainty is treated as a random process. Sampling of this process results in scenarios. This process of uncertainty modelling using the scenario analysis avails the information from the historical data to generate the scenario set,  $S$  or the scenario tree. This scenario tree represents the evolution of the uncertainty over time. The scenario set contains not only the replicates of the historical data but also contain instances that might possibly happen but never happened in the past. Each scenario is assigned a weight,  $\pi_s$  which represent the probability of its happening in the future. The stochasticity of the uncertainties dealt in this thesis namely the electrical and thermal demand, wind and PV generation capacity are described by normal distribution. Fig. 5.1 shows how the randomness in the wind generation unfurls over time. The bold line is the forecasted wind power and the shaded region describes the area covered by 10%, 20% and 30% prediction interval. Although the randomness is described by similar bell shaped distribution, they are statistically different. They have different mean and standard deviation. The variability in the distributions as shown in Fig. 5.1 increases with higher forecast periods due to the decrease in the accuracy of the forecast. The continuous normal distributions are discretized into five samples of probability equal to the area under the corresponding region of the probability density function as shown in Fig. 5.2. All values in one region are approximated to their corresponding P value (mean value of that region). For example all the values in region 2 are approximated to P2. The higher the number of discrete samples, greater is the model accuracy but however the sampling is restricted due to computational effort involved in solving huge stochastic models.

The evolution of this random process over time i.e. all possible future demands is realized in the form of a scenario tree with finite set of nodes as shown below in Fig. 5.3. Each node of the scenario tree is a decision making point. The information regarding the uncertainty at time  $t$  is known only at the beginning of the next time stage  $t+1$ . The decision making process over the whole planning horizon is presented as a scenario tree. Each node ( $n$ ) of the scenario tree has a set of successive nodes ( $n^+$ ) and a transition probability  $\pi_{n^+/n} > 0$ , which is the probability of  $n^+$  being the successor of  $n$ . The probability  $\pi_{n^+}$  of each node  $n^+$  is given recursively by  $\pi_1=1$ ,  $\pi_{n^+/n} * \pi_n$  for  $n^+ \neq 1$ . The probabilities of all the nodes at any given time stage add to one. Scenarios can be sampled from these discrete distributions using the Monte Carlo approach, Importance sampling, Bootstrap sampling, Internal sampling, conditional sampling or stratified sampling procedures [76], [77]. A



**Fig. 5.1** Wind generation forecast with 10%, 20% and 30% prediction interval



**Fig. 5.2** Discrete approximation of the continuous Gaussian pdf

huge set of scenarios are required to completely describe the evolution of the uncertainties. But it is computationally expensive [78], [79] to solve stochastic models with huge number of scenarios. Therefore scenario reduction techniques [80], [81] are used to find a reduced set of scenarios that are as close as possible to the initial set of scenarios. In this thesis a new scenario reduction algorithm based on swarm intelligence is proposed.

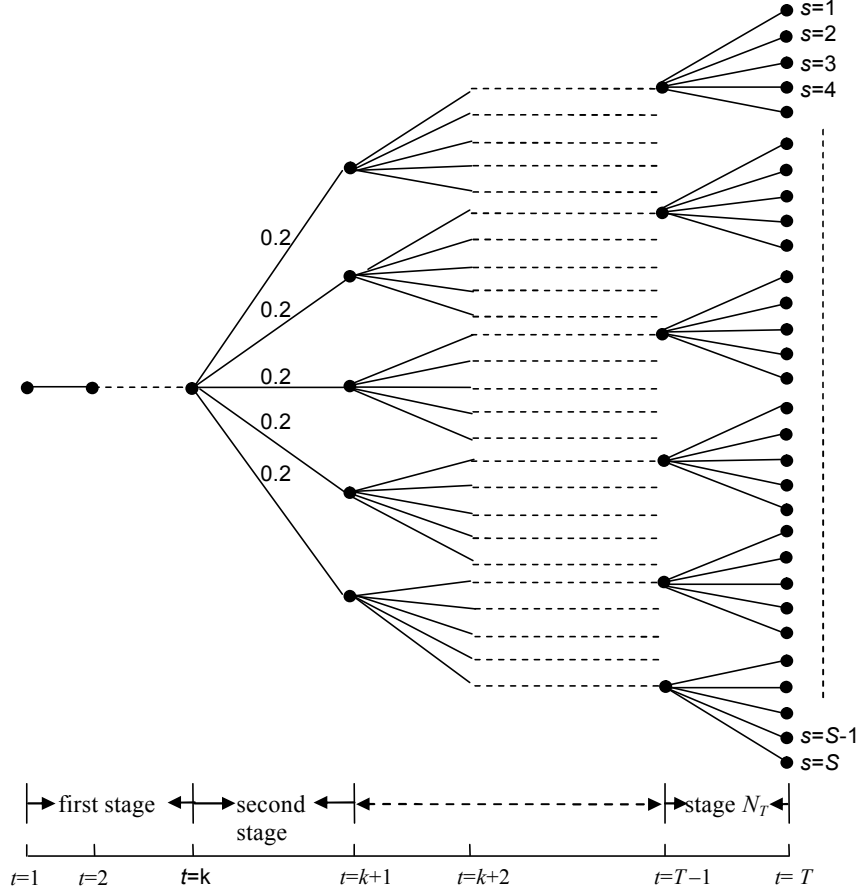


Fig. 5.3 Multi-stage scenario tree with five samples to each node at a branching stage

## 5.4 Scenario reduction algorithm

Michal Kaut and Stein W. Wallace [82] listed different scenario generation and reduction algorithms. All the currently available scenario reduction methods use different probabilistic metrics to select the best set of scenarios. The scenario to be deleted is selected by comparing each scenario with the rest of the scenarios [81]. For deleting  $N_r$  scenarios from an initial set of  $N_s$  scenarios, the process of one-to-one comparison has to be repeated  $N_r \cdot (N_s - 1)$  times. This is computationally expensive and is not suited for huge set of initial scenarios.

The new scenario reduction algorithm proposed in this thesis is formulated as a special optimization problem and is solved using PSO. The scenario reduction method always has a finite set of scenarios. The search space represents the set of all possible scenarios. In global optimization all the particles tries to find an opti-

mal solution space to minimize/maximize a common objective. But in this problem formulation, each particle has a unique objective. The objective is to improve their fitness measure. However the fitness of a particle depends on the position or performance of its neighbors. Each and every particle in the swarm has to observe the flight of its close associates and thereby adjust its velocity and position to find the optimal objective value. The multivariate scenario is a realization of the random variables i.e. each node of the scenario tree represents an  $N_D$  dimension vector corresponding to  $N_D$  random variables. The particle represents a scenario. The swarm representing the reduced tree optimally adjusts its branches during the evolutionary process. The aim of this optimization is to place the scenarios at distinct positions in the search space such that the distance between the scenarios weighted by their probability is maximized. After the optimization process, the swarm returns scenarios that represent the entire probability space of the initial scenarios. The particles in the swarm utilize their previous best performances and follow the best performer in their neighborhood to explore and exploit new areas in the search space. This swarm intelligence technique allows the particles to decide on the optimal set of scenarios within a few iterations. The particles traverse the entire search space to maximize their probability along with the distance from their neighbors.

The objective of particle  $m$  is to maximize its fitness value i.e.

$$\text{objective}(\text{particle}^m) = \text{maximize}(\text{fitness}^m) \quad (5.5)$$

The fitness of each particle is formulated as an aggregated normalized multivariate Euclidean distance weighted by its probability.

$$\text{fitness}(\text{particle}^m) = \inf_{q \in N_p} \pi_m \left\{ \frac{1}{N_D} \sum_{i=1}^{N_D} \frac{\left( \sum_{j=1}^{N_T} (\mathbf{A}_{i,j}^m - \mathbf{A}_{i,j}^q)^2 \right)^{\frac{1}{2}}}{N_i} \right\} \quad (5.6)$$

where  $N_p$  is the swarm size,  $N_D$  is the number of random variables,  $N_T$  is the total number of branching stages,  $\mathbf{A}_{i,j}^m$  is the value of the random variable  $i$  corresponding to particle  $m$  at  $j^{\text{th}}$  stage and  $\pi_m$  is the probability of the scenario/particle  $m$ . Since the random variables are measured to different scales, the distance has to be normalized by a factor  $N_i$ . The distance of particle  $m$  with the rest of the particles in the swarm is evaluated and the infimum distance measure is assigned as the fitness of this particle.

In this optimization problem, the particles try to improve their fitness by moving away from its nearest particle. So they should be aware of their nearest neighbor and try to move away from them. Therefore each particle is informed about its nearest neighbor through its new velocity update equation. The social

contribution term in the standard PSO velocity update equation [83] is modified to accommodate the information regarding the particle's nearest neighbor. The exploration of particle  $m$  at stage  $j$  and iteration  $k+1$  is influenced by its previous velocity ( $v^k$ ), the distance from its previous best performance ( $pbest$ ) and the distance from its nearest neighbor ( $gmin$ ).

$$v_{mj}^{k+1} = \chi(w_{mj}^k + \phi_1(x_{mj}^{pbest} - x_{mj}^k) - \phi_2(x_{mj}^{gmin} - x_{mj}^k)) \quad (5.7)$$

The particles follow the direction of its previous best performance but makes sure that they move opposite to its nearest neighbor. The new velocity update equation will guide the particles to explore prominent areas of the search space so as to maximize their fitness. Since they have the information about their nearest neighbor, they try to move away from them without degrading their fitness. Since the particles represent scenarios, the end process results in prominent scenarios. The algorithm performs one-to-one comparison only among the particles in the swarm and not for the entire initial set of scenarios. The algorithm follows a PSO flying strategy to select the optimal set even from an extremely huge set of scenarios in a very few iterations. The optimization process explores the entire search space or the huge set of initial scenarios to find the best set of scenarios. With the above mentioned uncertainty modeling which involves  $N_T$  branching stages for a  $N_D$  number of random variables results in  $(5^{N_T})^{N_D}$  number of initial scenarios. The process of scenario reduction can be explained by the following steps.

#### Step 1: Initialize the swarm

The swarm size which represents the number of preserved scenarios ( $N_r$ ) is fixed before the start of the reduction process. The position and velocity of each particle at a given dimension or time stage is randomly initialized among the discrete samples of the random variables at that time stage. The particles best performance so far measured,  $pbest$  is initialized to the current position.

#### Step 2: Calculate scenario probability

Each node of the initial scenario tree has  $N_b$  successor nodes. Each successor has a transition probability of migrating from the current time stage to the next. The current swarm is visualized as a reduced scenario tree and is compared to the initial tree. If any of the successors is missing then the transition probability of that missing successor is added to its nearest neighbor. The probability of the successor node is its transition probability multiplied to its parent node probability. The node probability of the leaf node or the successor nodes of the last branching stage gives the scenario or particle probability.

#### Step 3: Fitness evaluation

A fitness value is assigned to each particle based on its distance with its neighboring particles. Calculate the Euclidean distance corresponding to particle  $m$  as shown in (5.6). The fitness of the particle will be the aggregated multivariate

Euclidean distance with its nearest neighbor weighted by its probability. The objective of this particle is to maximize its distance from its nearest neighbor and also improve its probability.

**Step 4: Evaluate  $pbest$  and  $gmin$**

If the current fitness value of a particle is better than the current  $pbest$  of the particle, the  $pbest$  value is replaced by the current value. The position and fitness of the nearest neighbor to a particle is stored in  $gmin$ .

**Step 5: Update position and velocity**

The update equations are similar to the standard PSO update equations. But the social information term in the velocity update equation is modified as shown in (5.6). The particle's velocity is guided in the direction of its previous velocity ( $v^k$ ), its previous best performance ( $x^{pbest}$ ) and in the direction opposite to its nearest neighbor ( $x^{gmin}$ ). By doing so, the particle will move farther away from its nearest neighbor. This will improve the Euclidean distance and also its fitness value. The calculated velocity is then added to the current position to obtain the new position for the particle. The real valued position vector is rounded off to its nearest discrete value.

**Step 6: check for the boundaries**

This process ensures that the particle is in the designated search space. If a particle crosses the boundaries, it is made to stay on the search space boundary.

**Step 7: Turbulence**

During the early process of the optimization, turbulence is added in the form of velocity to each particle to avoid premature convergence. After a particles position has been updated, check if this update has degraded the fitness of the worst member of the swarm. If it does so, then the velocity of that particle is randomly initialized.

**Step 8: Check the exit condition**

If the current iteration number reaches the maximum iteration number, then exit. Otherwise go to Step 2.

The particle's fitness is improved at every iteration and towards the end of the optimization process; the swarm has particles with best fitness values. It means the swarm has best particles or distinct scenarios. The optimization process explores the entire search space or the huge set of initial scenarios to find the best set of  $N_p$  scenarios. Unlike the conventional scenario reduction methods, the proposed method does not go through each and every scenario in picking the best scenarios. The particles use the experience gained from their exploration to select the scenarios. The performance of the proposed approach is discussed in chapter 6. Once the realizations of the uncertainties are modeled as a suitable scenario tree, they are incorporated into the stochastic model. The resulting stochastic model is solved using the adaptive particle swarm optimization algorithm discussed in chapter 3.



## 5.5 Applications

The robustness in decision making has extended the use of SP to a wide variety of applications in power systems, transportation, finance, economy, manufacturing etc. The first application reported back in 1956 was in fleet assignment [84]. In this application, a stochastic model was developed to find the optimal allocation of flights to various routes under uncertain passenger demand. The model resulted in a potential gain of 9% more than the deterministic model. This tremendous success had an epidemic effect on the applications of SP. In energy sector, SP has been tremendously applied in planning and operation of the power system [85]-[94]. The scheduling of various generation resources can be optimally done considering the uncertainties in the inflows to the reservoirs, variability in electrical and heat demand and stochastic generation capability of the renewable units like the wind [95] and photovoltaic [96]. In planning, stochastic models can be used for feasibility analysis which helps investors to make optimal cash inflows to various new attractive renewable options. SP models can be used for capacity expansion of power systems to model the unexpected growth in demand, fuel prices and other financial constraints [97]-[99]. Several SP models are developed for trading electricity in the deregulated power market [100]-[103].

Most of the motivating applications of SP are related to finance. The finance models can not rely on deterministic approximations for they are subjected to the maximum risk. SP can be a useful tool to make optimal investment strategies in a volatile financial market. They can substantially improve the trade-off between risk and reward. Some of the prominent applications in finance include asset allocation for insurance companies [104], security selection for stocks and bonds [105], portfolio management [106], currency hedging for multinational corporations, hedge fund strategies to capitalize on market conditions and risk management for large public corporations [107].

In production and supply chain, SP is used to achieve better capacity planning and resource utilization which in turn can maximize the expected profits subjected to uncertain demand and cost. Major applications include production planning [108], optimal investment and operation planning of offshore gas fields under uncertain gas reserves [109], scheduling and economic analysis [110], manufacturing-to-sale planning [111], selection of optimal supply contracts [112], investment planning [113], process planning [114], managing purchases [115] etc. In environmental and pollution control [116], [117], SP models are used to estimate the type and capacity of water treatment plants without the actual information about the population and industrial growth.

SP has also several applications in agriculture [117], forest management [118], military [119], gaming [120], fisheries [121], telecommunication [122], water management [123], traffic management [124], portfolio selection [125], lake level management [126], timber management [127] etc.



## Chapter 6

### **Selected Applications of PSO in Power Systems**

#### **6.1 Introduction**

APSO was successfully applied to solve complex optimization problems in power systems. This chapter deals with the most significant optimization tasks in power systems such as optimal power flow, unit commitment and economic dispatch. Section 6.2 addresses the optimal power flow in an offshore wind farm connected to the grid via long cables. This problem is formulated as a mixed-integer model. The remaining sections in this chapter deal with optimization under uncertainty. Section 6.3 handles the unit commitment problem for industrial applications. The unpredictable nature of electrical demand and wind generation capacity are considered in the decision making process. The uncertainties are modelled using scenario analysis. The model is formulated as a two-stage stochastic model with recourse. This model generates a unique robust UC schedule which is near optimal to any possible scenario for the uncertainties. The operator can therefore have a reliable and cost optimal operation of the power system. In section 6.4, the optimal operation of a hybrid power system for residential application is described using a multi-stage stochastic model with recourse.

#### **6.2 Reactive Power Management in Offshore Wind Farms**

In order to reduce the dependency on overseas energy suppliers, German government wish to harness as much energy as possible from its renewable natural resource, the wind. Wind power is already a leading renewable energy alongside hydropower. Germany is aiming to increase the amount of power generated by renewable energy sources like the wind and solar power from the current 14 percent to 30 percent by 2020. With the onshore windy areas already crowded, plans to encash the offshore windy areas is in progress. In order to meet the country's renewable energy targets it is planning to build up to 30 offshore wind farms. These wind farms are located far away from the grid and transporting the power in an efficient and economical manner is one of the major concerns of policy makers and system operators [128]. Normally, medium and large capacity wind farms are connected to the grid at medium and high voltage levels. Long radial lines and cables are required to transport power from the remote wind farms. Detailed system planning studies are required for building new transmission lines/cables to provide better stability and reliability, and to avoid the system congestions. Over the past several years, many investigations of wind's impacts on power system operation and operating cost have been carried out [129]. Mainly, the impacts depend on the

wind park location, generation capacity, grid interconnection point, network configurations etc.

Due to the large penetration and continuous improvement in the wind power technology, wind farms must fulfil almost the same requirements as the conventional power plants. According to the German grid code, wind farms have to supply not only active power but also reactive power into the grid [130]. The requirements are defined with respect to the power factor as a function of the voltage at the point of common coupling (PCC) with the main grid. Offshore wind parks are connected to the grid via long AC submarine cables. For secure system operation and to provide variable reactive power generation other reactive power sources like shunt reactors, capacitor banks or even FACTS may also be connected. On-load tap changing (OLTC) transformers impact the reactive power generation indirectly by varying the voltage level. The available reactive power sources must be utilized properly during both steady-state and dynamic conditions for efficient and secure operation of the system. Thus, the reactive power management becomes an integral issue in the grid-connected offshore wind parks and can be formulated as a non-linear mixed integer optimization problem.

This section deals with optimal reactive power management in an offshore wind park using adaptive particle swarm optimization (APSO) technique. The active power losses of the wind power system (up to the main grid) are minimized subject to the grid code requirements. The effectiveness of the proposed technique has been demonstrated on a real offshore wind park energy system and results are discussed. The algorithm is very fast and can be used for on-line application in VAr management. This research could be a guideline for policy makers and system operators to promote wind power in terms of the system reliability and security.

### ***6.2.1 Offshore Wind Farm***

Offshore wind parks are normally connected to the main grid using long cables (at least two cable to increase the reliability of the transmission system) having step-up transformers at both ends as shown in Fig. 6.1. Due to excessive charging currents of cables, line reactors are permanently connected at both ends of the cables. Apart from these, switched reactors may also be connected in the system to take care of voltage increase during low power generation periods and synchronization. To provide fast and continuous VAr control FACTS devices, such as thyristor controlled reactors (TCR), static VAr compensator (SVC), static synchronous compensator (STATCOM) etc. are also proposed. But due to excessive cost, the application of these devices is limited.

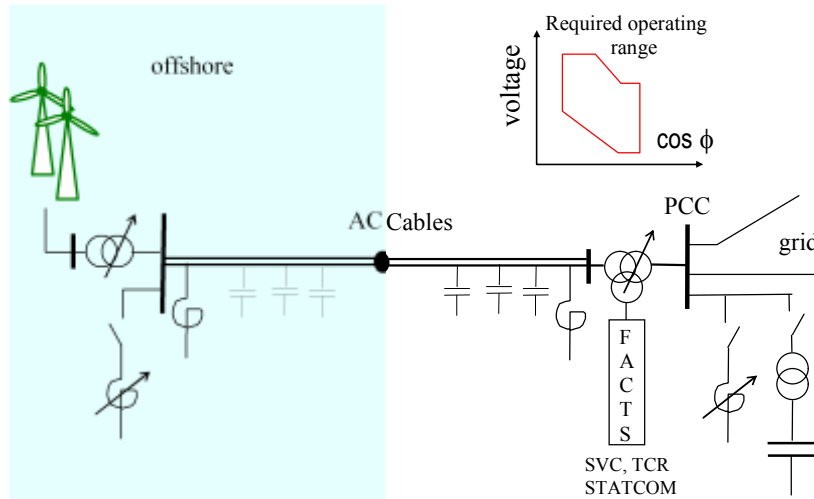


Fig. 6.1 Grid connected offshore wind park

### 6.2.2 Reactive Power Capability of Wind Energy System

Variable speed wind turbines are equipped with voltage source converters. The stator of the doubly-fed induction machines (DFIM) is directly connected to the grid while the rotor winding is connected using a voltage source converters (VSC). Only one third of the rated power flows through these converters. By supplying a voltage with variable frequency and variable amplitude to the rotor circuit, the shaft speed can be optimally adapted to the wind speed. The rotor-side converter (RSC) usually controls active and reactive power of the machine while the line-side converter (LSC) keeps the voltage of the DC-circuit constant. However, the LSC is also able to generate some reactive current until the maximum converter current is reached.

The reactive power capability curve (P-Q characteristic) of doubly-fed induction generators used in wind farms is shown in Fig. 6.2. The reactive power capability (in the first quadrant) of a doubly fed induction generator depends on the LSC and RSC converters capability, as the maximum reactive and active powers of the converter are limited by the maximum absolute current, and the magnetizing current of the induction generator's characteristic. Thus, such P-Q characteristics have two special features: machines can absorb more reactive power in under excited mode than generate reactive power in overexcited operation. Additionally, turbines are able to feed reactive power even if no active power is fed. Following the given characteristic in Fig. 6.2, it is desirable to reduce active power and increase reactive power during fault situations. Also, in low wind speed periods, when the wind turbine is still not running, the full reactive power generation capability is available if the converter can be switched solely to the grid.

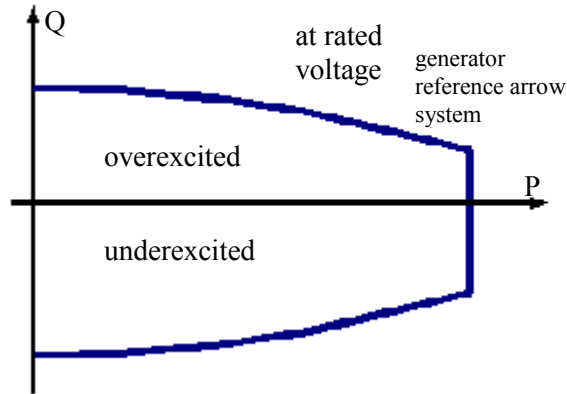


Fig. 6.2 P-Q-characteristic of an exemplary DFIG

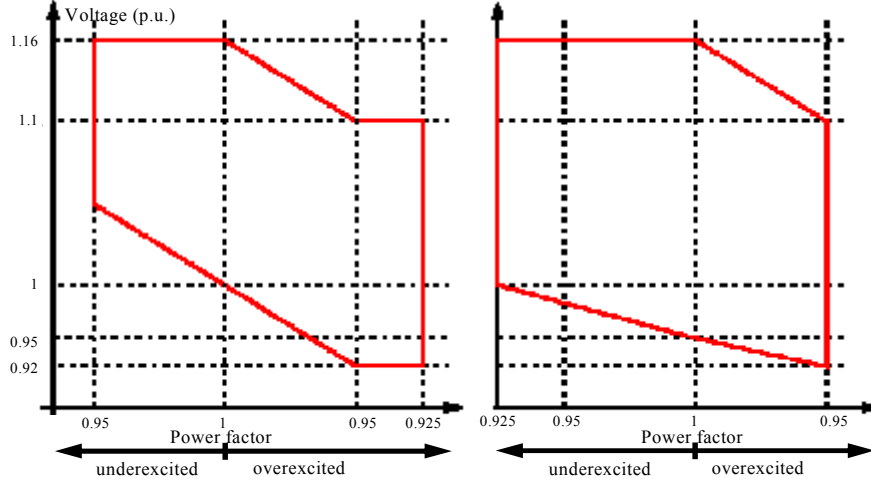
### 6.2.3 Grid Requirements

The wind farms connected to the high voltage grid has to fulfill the same requirements as the conventional power plants. These basic requirements are specified by the transmission system operators (TSO) and have to be fulfilled by every generating plant at the grid connection point. These grid codes depend on the topology and loadings of the TSO's grid and therefore differ from company to company even if they are located in the same country.

The power factor requirements during the normal operation of two German TSOs [130] are illustrated in Fig. 6.3. There is a clear difference between the Fig. 6.3(a) and Fig. 6.3(b). In Fig. 6.3(b), the whole operating range seems to be moved to the under-excited area. The reason may be low-load and reactive power surplus situations. Thus, the generating units need to operate in under-excited mode for limiting the voltage. Other TSO prefers to operate the generators in overexcited mode to support the grid voltage. This may be necessary if the TSO's grid is heavily loaded or supplies a huge reactive power loads. If a transformer, which connects the wind energy system to the grid, is equipped with an OLTC, the voltage on the secondary side can be changed for the optimal operation of the wind generators.

### 6.2.4 Reactive Power Dispatch Problem

The reactive power dispatch problem is an optimization task to manage the various VAR sources in a wind farm system so as to minimize the real power transmission loss and to improve the voltage profile in the system while satisfying the unit and system constraints. The system constraints include the TSO specified grid code requirements. These constraints must be fulfilled by the wind energy system



**Fig. 6.3** Power-factor requirements of two TSO for generating units connected to the HV grid

in order to stay connected to the grid. This objective is realized by optimizing the power flow with regard to system losses by optimally adjusting the reactive power control variables. Some of these variables such as the reactive power output of the wind turbine can be controlled in a continuous range where as some variables like transformer tap settings can be operated only in a discrete range. So the reactive power management problem is formulated as a mixed-integer optimization task. The objective of this task is to identify the right settings of reactive power control variables, which minimizes the real power loss ( $P_{\text{loss}}$ ) of the system. The mathematical model is formulated as shown below:

$$\text{Minimize } P_{\text{loss}}(\mathbf{x}, \mathbf{d}) = \sum_{k=1}^{N_l} P_k \quad (6.1)$$

Where  $P_k$  is the real power losses in line  $k$  and  $N_l$  is the total number of lines including cables:

$$P_{k=(i,j)} = G_{ij} (V_i^2 + V_j^2 - 2V_i V_j \cos \delta_{ij}) \quad (6.2)$$

Subject to:

#### EQUALITY CONSTRAINTS

The equality constraints are the power balance equations which include the active and reactive power balance equations for each load bus and the real power balance equations for each generator bus.

$$P_i - |V_i| \left| \sum_{j=1}^N V_j G_{ij} \right| \cos(\delta_{ij} - \theta_{ij}) = 0 \quad \forall i \in N \quad (6.3)$$

$$Q_i - |V_i| \left| \sum_{j=1}^N V_j G_{ij} \right| \sin(\delta_{ij} - \theta_{ij}) = 0 \quad \forall i \in N \quad (6.4)$$

Where  $P_i/Q_i$  is the net active/reactive power injected at bus  $i$ ,  $G_{ij}$  is the real part of the element in the admittance matrix corresponding to the  $i^{\text{th}}$  row  $j^{\text{th}}$  column,  $B_{ij}$  is the imaginary part of the element in the admittance matrix corresponding to the  $i^{\text{th}}$  row  $j^{\text{th}}$  column and  $\theta_{ij}$  is the difference in the voltage angle between the  $i^{\text{th}}$  and  $j^{\text{th}}$  buses.

### INEQUALITY CONSTRAINTS

The system operating constraints constitute the inequality constraints. These constraints include:

- (i) Reactive power output of generators other than wind turbines ( $Q_g$ )

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max} \quad i \in N_g \quad (6.5)$$

- (ii) The voltage magnitude of the buses other than the PV buses ( $V$ )

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad i \in N_{PV} \quad (6.6)$$

- (iii) Current through the cables, lines and transformers ( $I$ )

$$I_i^{\min} \leq I_i \leq I_i^{\max} \quad i \in N_l + N_{Tr} \quad (6.7)$$

### BOUNDS

- (i) The transformer tap change ratio ( $t_{Tr}$ )

$$t_{Tr,i}^{\min} \leq t_{Tr,i} \leq t_{Tr,i}^{\max} \quad i \in N_{Tr} \quad (6.8)$$

- (ii) The wind turbine VAr settings ( $Q_{WT}$ )

$$Q_{WT,i}^{\min} \leq Q_{WT,i} \leq Q_{WT,i}^{\max} \quad i \in N_{WT} \quad (6.9)$$



(iii) The switchable cable ( $n$ )

$$0 \leq n_i \leq 1 \quad i \in N_{\text{cable}} \quad (6.10)$$

(iv) Switched reactor/capacitor reactive power limits ( $Q_C$ )

$$Q_{C,i}^{\min} \leq Q_{C,i} \leq Q_{C,i}^{\max} \quad i \in N_C \quad (6.11)$$

(v) VAR compensator (SVC) settings ( $Q_{\text{svc}}$ )

$$Q_{\text{SVC},i}^{\min} \leq Q_{\text{SVC},i} \leq Q_{\text{SVC},i}^{\max} \quad i \in N_{\text{SVC}} \quad (6.12)$$

(vi) Transmission line flow limit

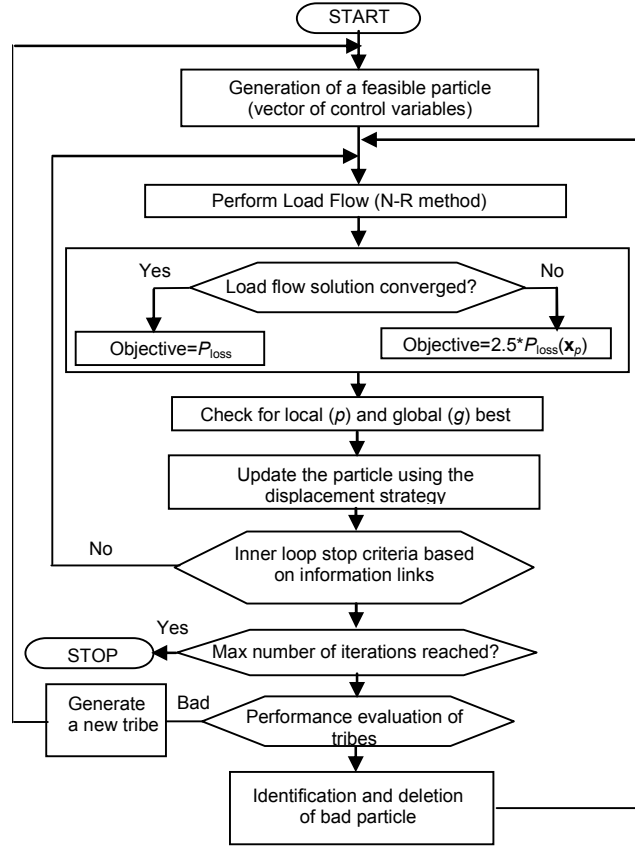
$$S_k \leq S_k^{\max} \quad k \in N_l \quad (6.13)$$

$\mathbf{x}$  is the vector of control variables which include the transformer tap change ratio ( $t_{\text{Tr},k}$ ), wind turbine VAR settings ( $Q_{\text{WT}}$ ), switchable cable ( $n$ ), switched reactor/capacitor ( $Q_C$ ) and VAR compensator (SVC) settings ( $Q_{\text{svc}}$ ). Vector  $\mathbf{d}$  consists of the dependent variables such as the reactive power outputs of generators other than wind turbines ( $Q_g$ ), voltage magnitude of the buses other than the PV buses ( $V$ ) and current through the cables, lines and transformers ( $I$ ). Constants  $N_g$ ,  $N_{\text{WT}}$ ,  $N_{\text{Tr}}$  are the number of conventional generators, wind turbines and transformers respectively.  $N_{\text{cable}}$ ,  $N_{\text{SVC}}$ ,  $N_C$  represent the number of cables, SVC's and shunt elements respectively.

In this work, reactive power generation of wind turbines (continuous variable), transformer tap changers' position (discrete variable) along with cable switching (binary variable) have been considered for wind farms' VAR management. This is a mixed-integer nonlinear optimization problem which is solved using the adaptive particle swarm optimization technique. The problem is formulated in a general way so that any other available reactive power sources (discrete, binary or continuous) such as bus or tertiary reactors, SVC or STATCOM etc. can be easily incorporated.

### 6.2.5 Solution Procedure

The optimization process for the economic reactive power dispatch problem can be explained by the flowchart in Fig. 6.4. The particle represents the vector of all control variables. The reactive power settings suggested by the particles are valid-



**Fig. 6.4** Flowchart of APSO approach for reactive power dispatch problem

dated by performing a load flow. The load flow is performed using Newton-Rampson method without any simplifications. If the load flow doesn't converge, the objective is 2.5 times the losses corresponding to the local best position of that particle ( $x_p$ ). The process of evolution of the swarm is very similar to the algorithm mentioned in chapter 2.

### 6.2.6 Test Network

The test network consists of a German offshore wind park as shown in Fig. 6.1 having 80 wind turbines each rated at 5 MW, 0.95 kV, connected to the main grid (380 kV) with two, 144 km long cables (94 km AC submarine cable and 50 km onshore cable) each rated at 150 kV. One 36 kV/150 kV and two 150kV/380kV step-up transformers on-load tap changing facility are used as shown in Fig. 6.1. Each transformer has 27 steps with 1.23% per step. These discrete points are nor-

malized to integers between  $\pm 13$ . The tap settings of the two 150/380 kV transformer are identical. During full capacity wind generation both the cables are in operation. For low wind generation the lines and cables should be optimally used to reduce the losses. The effectiveness of the optimization algorithm and its response to low wind generation is tested by considering the total generation as 40 MW which is 10% of rated wind farm power output ( $P_N=400$  MW). The power system considered in this research consists of 195 nodes, 4 shunt elements, 97 single lines, 3 three-winding transformers and 80 DFIGs with their machine transformers. For simplicity, the reactive power of all generators of the wind park is kept equal.

### 6.2.7 Results

Adaptive Particle Swarm Optimization is used to solve the reactive power dispatch problem of a wind energy system. The simulations are performed on a P4, 3.4GHz processor. The aim of the optimization is to find the best operational settings of the system so to provide minimum losses. The simulation results corresponding to 10% loading condition are shown in Fig. 6.5-6.9. The decrease in the power losses with the progress of the APSO algorithm is shown in Fig. 6.5. The bullets indicate the objective function value at each function evaluation. The solid line represents the evolution of the global best particle. The points below the solid line indicate the infeasible solutions. The search process starts at a region (10MW) far away from the global solution. But the swarm takes only a few function evaluations to jump to the optimal space. Most of the function evaluations belong to the optimal space. The evaluations outside this region indicate the

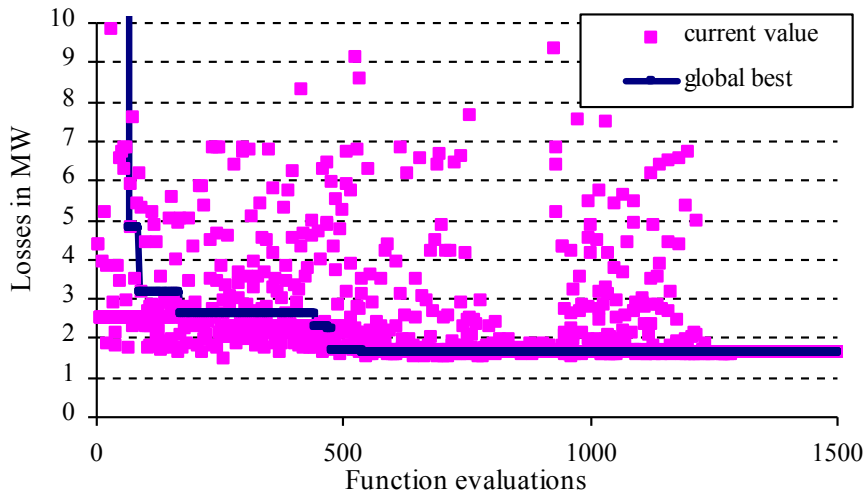
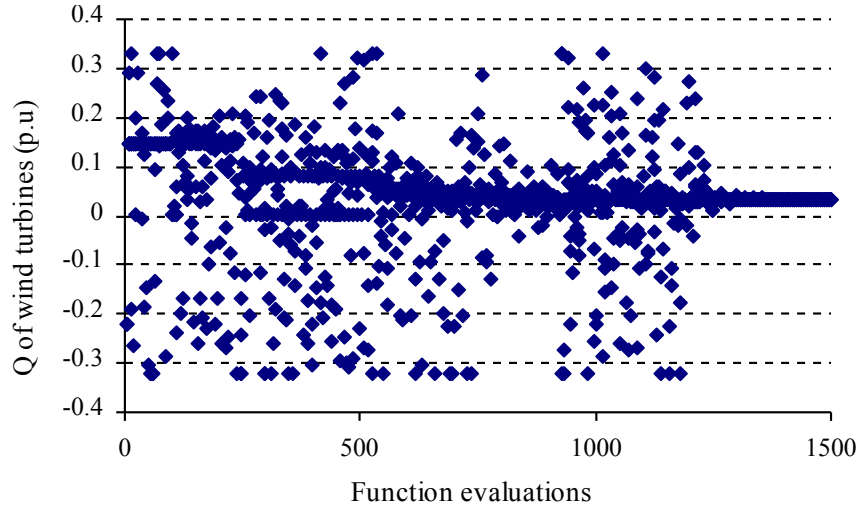
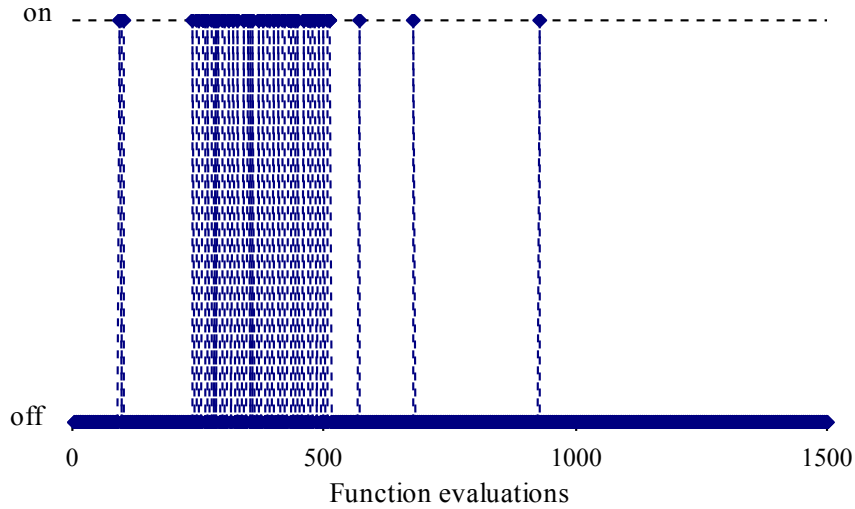


Fig. 6.5 Minimum wind energy system loss during the APSO iterations

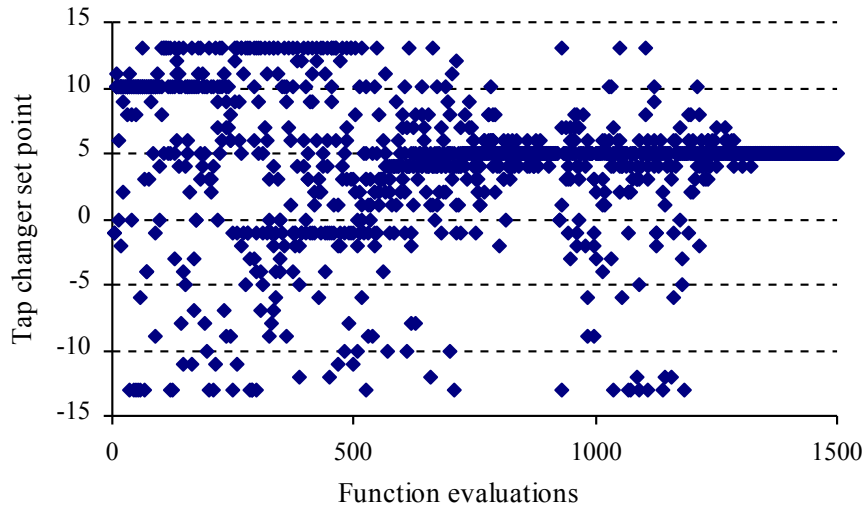


**Fig. 6.6** Reactive power generation of a single wind generator during the APSO iterations

diversity in the search process. After nearly 1200 evaluations the swarm is able to converge to an optimal solution of 1.6152MW. The whole optimization process takes about 3.36 seconds. The reactive power generation of a single wind turbine during the iteration process of PSO is shown in Fig. 6.6. The particles explore a wide area of the parameter space between (-0.33p.u., 0.33p.u.). The swarm has traced four optimal reactive power settings (0.14 p.u., 0.076 p.u., 0.0038 p.u., 0.034 p.u.) for the wind turbine. However the swarm is able to identify the global reactive power setting for each wind generator. The switching of cable is also cru-

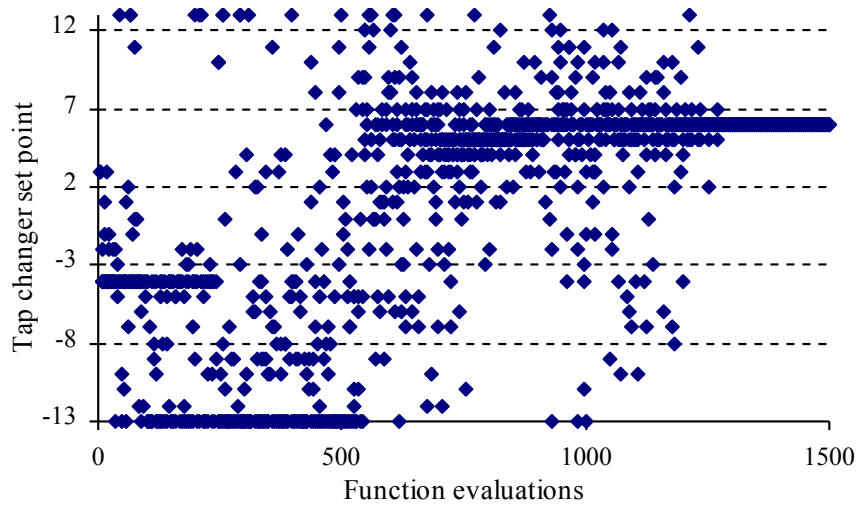


**Fig. 6.7** Cable-2 status during the APSO iterations



**Fig. 6.8** Tap changer set pint of 36/150 kV transformer during the APSO iterations

cial in the reactive power management for required grid code where power factor at the point of common coupling is to be maintained in the desired limit. Fig. 6.7 shows the status of cable-2 during the optimization process. Most of the optimal solutions suggest that cable 2 need to be switched OFF for this loading condition. The tap settings of 36/150 kV and 150/380 kV transformer are shown in Fig. 6.8 and Fig. 6.9. These transformers have a wide range of operation. As evident from the figures, there are many optimal operation states. The swarm however is able to overcome the local optimal settings and converge on the global optimal settings.



**Fig. 6.9** Tap changer set point of 150/380 kV Transformer during the APSSO iterations

**Table 6.1** Real power loss and optimal parameter for different loading

$P_{WT}$ (% of $P_N$ ) (MW)	$P_{loss}$ (MW)	$Q_{WT}$ (Mvar)	Transform tap settings (36/150)      (150/380)		Cable-2 status
40.0 (10%)	1.6152	0.1800	6	5	OFF
80.0 (20%)	2.8264	0.1445	12	0	OFF
160.0 40%)	5.6857	0.2817	3	1	ON
240.0 60%)	10.1757	0.0649	4	1	ON
320.0 80%)	16.4876	-0.1027	5	0	ON
400.0 100%)	24.6559	-0.3921	6	-1	ON

Simulations are performed for different levels of wind generation (10% to 100% of rated power) and their corresponding losses together with the various optimal system settings are listed in Table 6.1. The real power losses show a moderate increase from 10% to 20% loading. Beyond this loading point the second cable is switched ON and this accounts for a steep increase in the losses. The wind turbines operate in the under-excited mode until 60% loading. The wind turbines are in overexcited mode and deliver reactive power beyond 60% loading. The transformer tap settings are adjusted to match the required voltage profile of the network.

### 6.2.8 Conclusion

The optimal reactive power dispatch problem has been successfully solved using APSO algorithm to attain minimum power losses in a grid connected offshore wind farm. As evident from the results the algorithm is able to overcome the local optimal settings and provide global solutions. The optimization process is computationally less expensive and has a very high convergence speed. The algorithm is modelled to incorporate any new reactive elements in the power system. It is recommended to implement and run the algorithms in online mode as part of the wind farm management software so that the setting are always adapted when the wind conditions as well as the grid requirements change. A proper operation of reactive power sources in the wind farms will give a better economy. This research could be a guideline for policy makers and system operators to promote wind power in terms of system reliability and security.

## 6.3. Optimal Operation of a Wind-Thermal Power System

### 6.3.1 Introduction

The rapid development of new efficient, reliable and economical wind turbines has enabled huge penetration of wind energy in the power system. Germany has an installed wind energy capacity of 22.5 GW (May 2008). This represents 5.7% as annual average of all electricity generated in the country. The repowering program can increase the current installed wind capacity by a factor of 2.5 or more. The installation of huge offshore wind farms along the coast of Germany might further increase the penetration level. With the ever increasing wind installations and intermittent nature of the wind, it is essential to investigate the impact of huge wind penetration on the power system operation. The scheduling decisions in the power system are made one forecast period (usually one day) in advance to meet the net load and reserve requirements. The huge variation in wind penetration levels over the planning period (Fig. 6.10) makes the scheduling of the generators a challenging task. For optimal scheduling and utilization of generators, the amount of wind in feed should be estimated with a certain level of accuracy. Moreover the unpredictable nature of the wind requires additional power reserves for operating the wind integrated power system at the required stability margin. The scheduled system reserve therefore has to support the generator outages and also support the unpredictable nature of the wind generation. Due to the competitive power market, the utilities should be able to define the optimal unit commitment schedule and spinning reserve levels with respect to economic and reliability issues.

This section presents a unit commitment formulation that takes into consideration the stochastic nature of both the wind generation and load. Different authors have proposed models to solve unit commitment problem for wind integrated power system. Chun-Lung Chen [131] proposed a method to incorporate wind generators (WEC) into the generation scheduling problem. Special reserve constraints were established to operate the power system within the required stability margin. The author however did not consider the effect of wind fluctuations on system operation. Contaxis and Kabouris [132] modeled the WEC as an n-state unit. A probabilistic model for the WECs is considered to represent the wind uncertainties. The disadvantage of such modeling is its dependency on the statistical information of the individual WECs. In [133] an artificial neural network (ANN) based forecasting technique is used to estimate the probabilistic confidence interval for the forecasted wind generation. This information is utilized to predict the upper level of the distribution curve for the wind generation with a specified risk level.

The major issue in developing the UC problem formulation is the modeling of the uncertainties i.e. wind generation and load. Scenario analysis as describes in chapter 5 is used to model the uncertainties. In this study wind generation forecast is carried out by an ANN tool for all WECs as a single quantity [134]-[137]. This

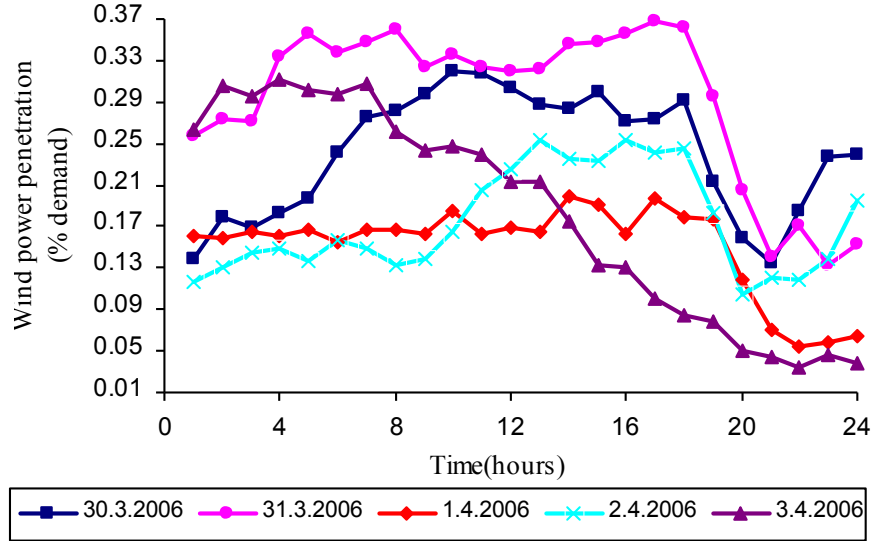


Fig. 6.10 Example for wind power penetration presented for day ahead planning

approach reduces the overall forecast error and also eliminates the dependency on the individual WECs. The wind generation and load are considered as two independent random processes. The sampling of these random processes results in scenarios representing the future realizations of the uncertainties. These scenarios are generated using the forecasted data, mean and standard deviation of the forecast error at every time stage. The scenarios lie within a certain probabilistic confidence interval defined by the forecasting tool. A huge number of scenarios are required to completely describe the stochastic nature of the uncertainties. Solving the stochastic UC problem [138], [139], [140] with these huge set of scenarios is computational too expensive. So an appropriate scenario reduction technique must be used to limit the number of scenarios. The currently available scenario reduction methods can not handle huge number of initial scenarios. Therefore the scenario generation and reduction technique proposed in chapter 5 is used. This method improves the quality of the scenario tree, reduces the uncertainty modeling error and thus improves the stochastic solution.

The stochastic UC problem is solved by two-stage stochastic programming approach. The aim of this optimization process is to determine a robust UC schedule common to all scenarios and to minimize the expectation of the daily operating costs over all possible set of scenarios. This section investigate the influence of forecasted wind and load data on power system planning, improvements in uncertainty modeling using the PSO based scenario reduction algorithm and finally present a mixed-integer nonlinear two-stage stochastic programming model for unit commitment problem accounting for uncertainty of wind power generation and electrical demand. In this case, the operator has an optimal UC schedule to plan



the day-ahead operation of the power system irrespective of the actual realization of wind generation and load uncertainties.

### ***6.3.2 Wind Power Forecast Methodology***

Wind is a fluctuating power source and difficult to control. Integration of large amounts of wind power in the power system requires accurate and reliable forecasts of the electricity generated by wind turbines for the next hours to days ahead. For power plant scheduling and electricity trading the ‘day-ahead’ prediction is used, in contrary to shortest-term forecasts which are essential for grid operation and trading on the intraday market.

There are mainly two different approaches used for wind power prediction. One employs physical models of wind farms which are generally based on the calculation of the wind speed in hub height of the wind turbines. This method requires not only weather data from numerical weather prediction (NWP) models but also special knowledge of the local conditions of each wind farm like orography, surface roughness and wind park losses. The resulting wind speed will be transformed to the power output using the individual power curves of the wind turbines. The second approach focuses on mathematical and statistical modeling methods, like regression analyses, and methods of artificial intelligence mostly realized by Artificial Neural Networks (ANN). Such models are aimed to qualify the coherence between meteorological data from a NWP model and the most probable wind farm power output and are generally based on a training procedure using historical weather and power data. An advantage is that the individual local conditions of each wind farm have been measured empirically during the training phase.

The wind power prediction data processed within this section is adapted from an ANN-model that is used by German transmission system operators and hence represents true-to-life conditions. This model is developed by Dr. Kurt Rohrig at R&D division for Information and Energy Economy, Institut für Solare Energieversorgungstechnik, Kassel, Germany. As a case study, wind power measurements and forecasts of a pre-defined control zone (~ 4GW) were used. Both time-series have been derived by an up-scaling procedure using the power measurements and forecasts from 16 representative wind farms located within the control zone. For the applied day-ahead prediction, local meteorological forecast data from the NWP system of the German weather service (DWD) has been used as input for an ANN for each selected wind farm. The respective ANN structure and the up-scaling methodology are described below.

### 6.3.2.1 Artificial Neural Network Model

The wind power prediction model used for this research consists of an ANN in the form of a multi-layer perceptron (MLP) that was trained by the back propagation algorithm. In this work a MLP with one hidden layer is used. The neurons of the input layer are linear neurons, i.e. within this layer there is no information processing but a distribution of the information of the input parameters onto the neurons of the hidden layer. For ANN training, historical NWP data as well as historical measured power data with discrete time steps of one hour was used. As input vector for the time step  $t$  the NWP point-data for the location of the wind farm, namely values of the wind speed and direction from several successive time steps (e.g.  $t-1$ ,  $t$ ,  $t+1$ ) in different heights (e.g. 30m, 100m) has been selected. Several successive time steps has been utilized as simultaneous training parameters due to the fact that the ANN, and hence the power prediction, is aimed to handle temporal gradients (fluctuations) of the wind speed and direction. Furthermore some parameters describing the thermodynamic properties of the present air masses, like density and temperature, were also included in the input vector. ANN is trained by gradient descent with the back propagation algorithm.

### 6.3.2.2 Up scaling Methodology

The determination of the aggregated wind generation for pre-defined control zones is calculated by transformation of measured and forecasted wind farm power values of a set of selected wind farms. To ensure an adequate representativeness the selected wind farms have to be spatial distributed over the defined control zone. The transformation algorithm is based on the sub-division of the control zone (or sub-grid area) into small sector. The wind power feed-in (measurement or forecast)  $P_i$  of area (sector)  $i$  is calculated by a weighted summation of wind power signals of all representative sites.

$$P_i = k_i \sum_j s_j * A_j * P_j \quad (6.14)$$

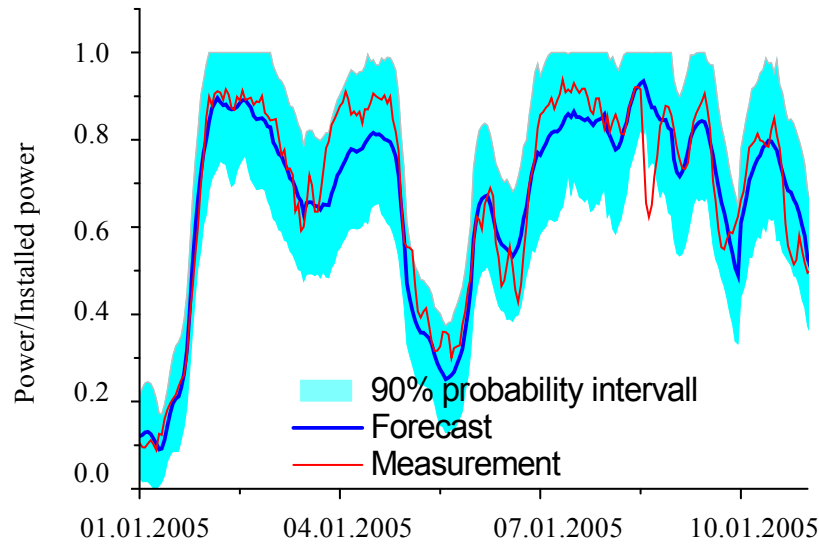
$P_j$  is the standardized measured (or forecasted) wind power of wind farm  $j$  and  $s_j$  is the status of the measurement/forecast (0 := wrong; 1 := o.k.).  $k_i$  is a correction factor, which guarantees that the sum of all weights result in one.  $A_{ij}$  contain the rated power of sector  $i$  and a distance dependent weight factor that present the influence of the (measured or forecasted) power of the representative wind farm  $j$  on the calculation of the power of area  $i$ . The total wind power feed-in is determined by the summation of the wind power feed-in of all sections at the same time.

$$P_{sum}(t) = \sum_i P_i(t) \quad (6.15)$$

### 6.3.2.3 Uncertainty area / Prediction interval

In addition to the wind power forecast itself, it is important to have knowledge about uncertainties of this forecast. A statistical and a MLP based method has been used to predict not only the power output, but also a prediction interval presenting the forecast accuracy for each time step. Fig. 6.11 shows the prediction interval along with the forecast and measured wind data for different days of the week. Regarding the generation of prediction intervals the main objective is to maximize the reliability, i.e. maximize the probability that the real wind power feed-in will be located within the interval while minimizing the interval's width.

The calculation of the prediction interval describing the accuracy of the power forecast of a pre-defined control zone is also based on the up scaling method and hence on the uncertainties of each individual wind farm. As described in the previous section the predicted wind power output of a control zone can be written as a weighted sum over the predicted power outputs of the representative wind farms:



**Fig. 6.11** Prediction interval of the wind power prediction from 30.03.2006 00:00:00 to 05.04.2006 23:00:00 generated with the 90%-model

$$P_{sum}(t) = \sum_j z_j * P_j(t) \quad (6.16)$$

where

$$z_j = \sum_i k_i * s_j * A_{ij} \quad (6.17)$$

The prediction error of the control zone can be calculated by the Gaussian error propagation scaled by a smoothing factor  $\sigma$  taking into account the reduction of the prediction error by the spatial smoothing effect:

$$\Delta P_{\text{sum}}(t) = \sigma \sum_j z_j * \Delta P_j(t) \quad (6.18)$$

with  $\Delta P_j(t)$  as the prediction error i.e. the prediction interval of the representative wind farms  $j$ . The smoothing factor  $\sigma$  is based on the calculation of cross-correlation coefficients between pairs of representative wind farms as described by M.Lange [141].

Using the statistical method the model to generate prediction intervals of an individual wind farm  $j$  was developed by investigating historical weather data and the simultaneous resulting forecast error observed at site  $j$ . The determination of the respective forecast error rest on a simplified classification of the weather situation based on the wind speed  $wg_j$  and wind direction  $wr_j$  adapted from the NWP. Concerning this modeling method the root mean square error (RMSE) is utilized to estimate the forecast error depending on the specified classification

$$\Delta P_j(wr_j, wg_j) = \sqrt{(P_{M,j}(wr_j, wg_j) - P_{P,j}(wr_j, wg_j))^2} \quad (6.19)$$

where  $P_{M,j}$  is the measured and  $P_{P,j}$  the forecasted power at wind farm  $j$ . Thus the total error of the control zone only depends on the individual weather situations and the respective classified forecast error of the  $n$  representative wind farms:

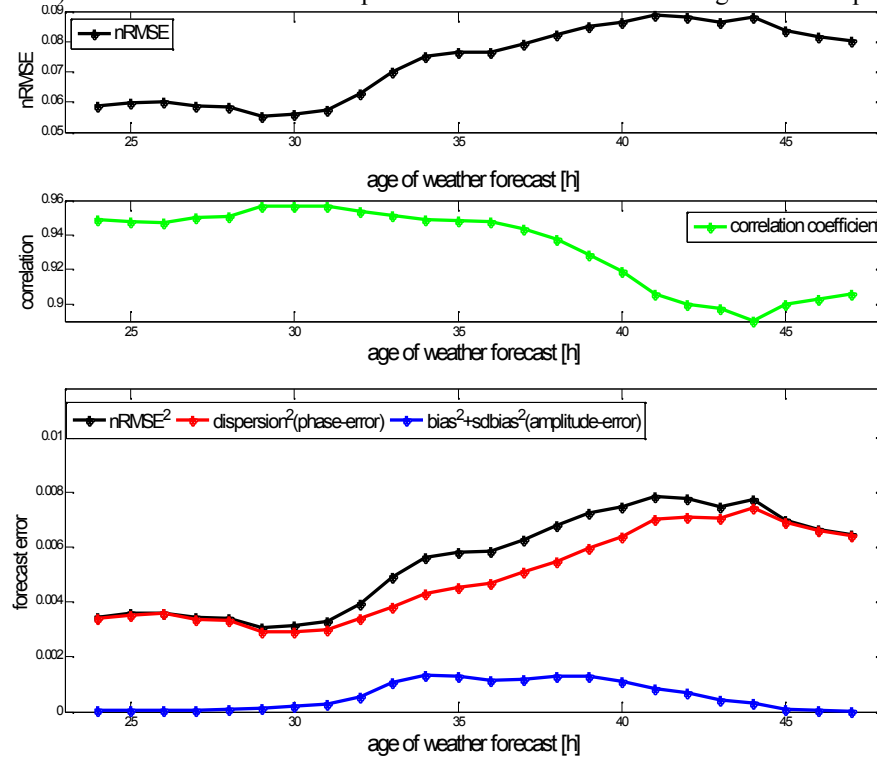
$$\Delta P_{\text{sum}}(wr_1, wg_1, \dots, wr_n, wg_n) = \sum_j z_j * \Delta P_j(wr_j, wg_j) \quad (6.20)$$

The resultant prediction interval of the forecasted total power at time  $t$  is calculated by using the RMSE value of the forecast error depending on the actual wind data of each wind farm. By using the smoothing factor  $\sigma$  from (6.18) one obtains the final prediction interval of the total forecast that ensures an error reduction due to spatial smoothing effects.

The second method to calculate a prediction interval of an individual wind farm requires the adaption of two MLP and an investigation of historical time series of power forecasts. In the first step the respective time series of forecast errors is divided in one time series covering all positive errors and in a second with all negative errors. In addition not only the weather forecasts at the location of the individual wind farm but also at different spatial distributed locations around it are used to create a reflection of a weather situation covering an expanded region. In the last step this weather situation serves as input to train two MLP which are able to reveal the relation between the current weather situation and the respective positive or negative forecast errors. To generate a prediction interval the two

trained MLPs were employed simultaneously to the power-MLP with the only difference that the presented input consists of a forecasted wind field.

In addition to both methods it is to note that generally the prediction error (Fig. 6.12) and hence the width of the prediction interval increases at higher time steps



**Fig. 6.12** Quality of the ‘day-ahead’ power prediction depending on the age of the weather forecast. Top: RMSE normalized on capacity; Center: correlation-coefficient; Bottom: Subdivision of the error in phase and amplitude error

of the current weather forecast when its accuracy decreases. As apparent in the bottom panel of Fig. 6.12 the  $\text{RMSE}^2$  can be split into a part that indicate amplitude errors ( $\text{Bias}^2 + \text{SDBias}^2$ ) that are typically related to site specific effects and random errors ( $\text{Dispersion}^2$ ) that reflect global properties of the predictability or non-predictability of the weather (often demonstrated ad phase shifts).

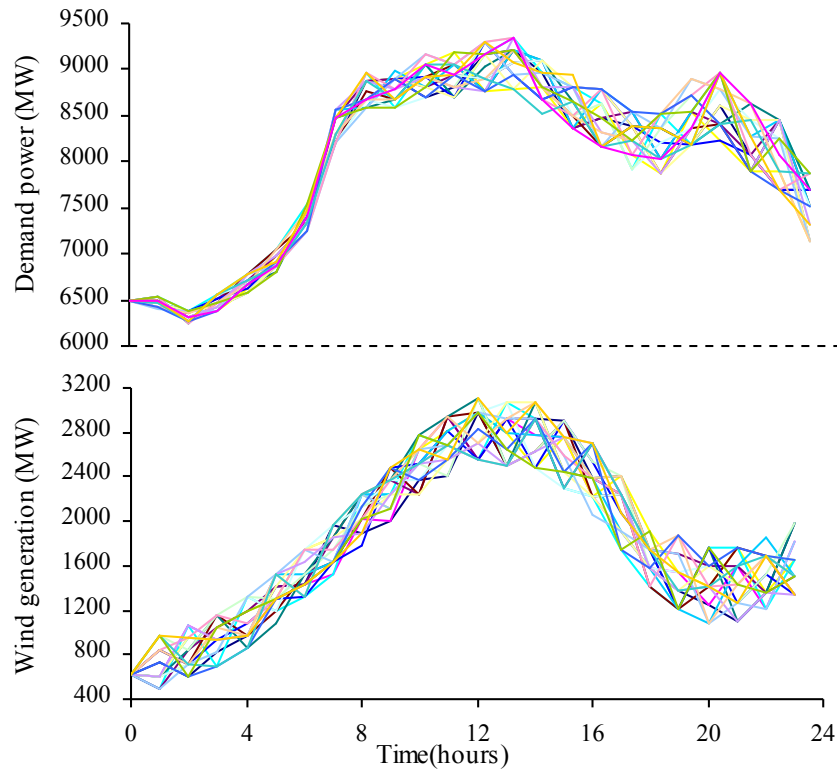
$$\text{RMSE}^2 = \text{Bias}^2 + \text{SDBias}^2 + \text{Dispersion}^2 \quad (6.21)$$

Note, that the forecast error reflects the diurnal cycle and is not monotonously increasing with forecast time, e.g. missing turbulence effects during night-time facilitate an improved weather (wind) forecast.

### 6.3.3 Uncertainty Modeling

#### 6.3.3.1 Scenario Generation

The wind power and demand scenarios representing the future realizations of the uncertainties are generated using the forecast data, the mean and standard deviation of the forecast error which cover the 90% probabilistic confidence interval defined by the forecast tool. The evolution of the two stochastic variables (electrical demand and wind generation) is modeled as two independent random processes (multi-variant random process). The marginal distribution for each of these variables at any time step is assumed to be a normal distribution. The marginal distribution is approximated to five discrete samples. The random process therefore has 25 samples at each time stage. Each of these samples has equal probability. The information regarding the two random variables is completely defined at  $t=1$ . The evolution of the random process for the next 23 hours is a huge set of  $23^{25}$  scenarios i.e. multi-stage scenario tree with 23 branching stages. Each of these scenarios



**Fig. 6.13** Scenario tree for stochastic wind power generation and electrical demand generated by PSO for 23 branching stages and 5 branches at each stage

represents a possible future realization of the random process. This type of uncertainty modeling results in a multistage scenario tree with 23 branching stages and 25 branches/samples at each stage. So, each node ( $n$ ) of the scenario tree has 25 equally probable successor nodes ( $n^+$ ). To solve the stochastic UC model, the random process with huge set of scenarios has to be approximated to a simple random process with finite set of scenarios and should be as close as possible to the original process. This huge set is reduced to a suitable size using the scenario reduction algorithm proposed in chapter 5. The initial scenario tree with  $25^{23}$  scenarios is reduced to 50 scenarios as shown in Fig. 6.13.

### 6.3.4 Problem Formulation

The power system consists of thermal generating units and wind turbines. The planning period is for the next 24 hours. The scheduling period is uniformly discretized into hourly intervals. The decision variables are categorized into first and second stage decisions. The unit commitment decision variables  $u_{it} \in \{0,1\}$ ,  $i=1,\dots,I$  and  $t=1,\dots,T$  are the first stage decisions and the generation levels  $p_{it}^s$ ,  $i=1,\dots,I$  and  $t=1,\dots,T$  of all units at each time interval of the planning period constitute the second stage decisions. Using scenario analysis for uncertainty modeling, the optimization has to be done for all considered scenarios. For a scenario tree with  $N_n$  nodes, there are  $N_n$  control variables for each unit describing the generation level at each node of the scenario tree. The objective of this optimization is to minimize the sum of the direct costs for first stage decisions (start-up costs) and the statistical mean value of the costs incurred by the first and second stage decisions (fuel cost).

Since the operating costs of the WECs are negligible, the total operating cost is given by the sum of the fuel cost and the start-up cost of all thermal units. The fuel cost for the operation of unit  $i$  and scenario  $s$  at time  $t$  is modelled as a quadratic function of the generator power output  $p_{i,t}$ .

$$FC_{i,t}^s = a_i + b_i p_{i,t}^s + c_i (p_{i,t}^s)^2 \quad (6.22)$$

Where  $a_i$ ,  $b_i$ ,  $c_i$  represent the cost coefficients. The generator start-up cost depends on the time the unit has been off prior to start up. The start-up cost is approximated by the following exponential cost function:

$$SC_{i,t}^s = \sigma_i + \delta_i \left( 1 - \exp \left( \frac{-T_{off,i}^s}{\tau_i} \right) \right) \quad (6.23)$$

Where  $\sigma_i$ ,  $\delta_i$  are the hot and cold start-up costs,  $\tau_i$  the unit cooling time constant and  $T_{off,i}$  is the time the unit has been off.

The unit constraints include:

- The minimum and maximum rated unit capacities
- Ramp rates
- Minimum up/down time limits of the units
- The initial states of the units must be taken into account

The overall objective function of the stochastic unit commitment problem is to minimize the total operation cost ( $TC$ ) of the thermal units.

$$TC = \sum_{t=1}^T \sum_{i=1}^I SC_{i,t} u_{i,t} \left(1 - u_{i,t-1}\right) + \sum_{s=1}^{N_s} \pi_s \sum_{t=1}^T \sum_{i=1}^I FC_{i,t}^s u_{i,t} \quad (6.24)$$

Where  $N_s$  represents the total number of scenarios,  $I$  is the total number of thermal units and  $T$  is the planning period. Apart from the unit constraints the cost model is subjected to a set of system constraints as shown below:

**(i) System hourly power balance:**

Total power generation must equal the load demand,  $P_D$ , in all time steps and for all scenarios.

$$\sum_{i=1}^I p_{i,t}^s + P_{W,t}^s = P_{D,t}^s \quad \forall t \in T, s \in N_s \quad (6.25)$$

Where  $P_{w,t}^s$  is the total wind power output of all wind turbines corresponding to scenario  $s$  at time  $t$ .

**(ii) Spinning reserve requirements:**

In many power system the amount of reserve maintained at any time is just sufficient to cover the loss of the largest generator or a certain percentage of electrical demand. But due to the huge penetration of wind power and its intermittent nature, its frequent variation may be greater than the reserve allocated. Although the scenarios capture the hourly variation of the wind generation, their fast variations between two branching stages have to be supported by two additional reserves [142]. The up spinning reserve ( $USR$ ) supports the sudden fall in wind power. During a sudden decrease in wind power, the thermal units should be able to ramp up to support for the reduction in wind power.  $USR$  is supplied by the up ramping capacity of the thermal units. The second reserve is the down spinning reserve ( $DSR$ ). This reserve contributes to the sudden raise in wind power. When ever there is unpredictable increase in wind power, the thermal units should be able to ramp down and should support the after affects of a sudden increase.



$$USR_t^s = \sum_{i=1}^N US_{i,t}^s \geq MSR_t + us * P_{W,t}^s \quad \forall t \in T, s \in N_s \quad (6.26)$$

$$US_{i,t}^s = \min(p_{i,t}^{s,\max} - p_{i,t}^s, US_i^{\max}) \quad (6.27)$$

$$DSR_t^s = \sum_{i=1}^N DS_{i,t}^s \geq ds * P_{W,t}^s \quad \forall t \in T, s \in N_s \quad (6.28)$$

$$DS_{i,t}^s = \min(p_{i,t}^s - p_{i,t}^{s,\min}, DS_i^{\max}) \quad (6.29)$$

Where  $US_{i,t}$ ,  $DS_{i,t}$  are the up and down spinning reserve supplied by unit  $i$  at time  $t$  respectively,  $MSR$  is the minimum reserve level to support generator outages and forecast errors in electrical demand,  $US_i^{\max}$  and  $DS_i^{\max}$  are maximum allowable up and down spinning reserves by unit  $i$  respectively,  $us$  and  $ds$  are the percentage of wind generation contributing to up spin and down spin requirements.

**(iii) Generation limits on the total thermal power output:**

$$\sum_{i=1}^N p_{i,t}^{s,\max} + P_{W,t}^s \geq P_{D,t}^s + MSR_t + us * P_{W,t}^s \quad (6.30)$$

$$p_{i,t}^{s,\max} = \min(p_i^{\max}, p_{i,t-1}^s + UR_i^{\max}) \quad (6.31)$$

$$P_{D,t}^s - P_{W,t}^s \geq \sum_{i=1}^N p_{i,t}^{s,\min} + ds * P_{W,t}^s \quad (6.32)$$

$$p_{i,t}^{s,\min} = \max(p_i^{\min}, p_{i,t-1}^s - DR_i^{\max}) \quad (6.33)$$

Where  $UR_i^{\max}$ ,  $DS_i^{\max}$  are the maximum allowable up and down ramp of unit  $i$  respectively,  $P_i^{\max}$ ,  $P_i^{\min}$  are the maximum and minimum generation limits of unit  $i$  respectively.

### 6.3.5 Case Study

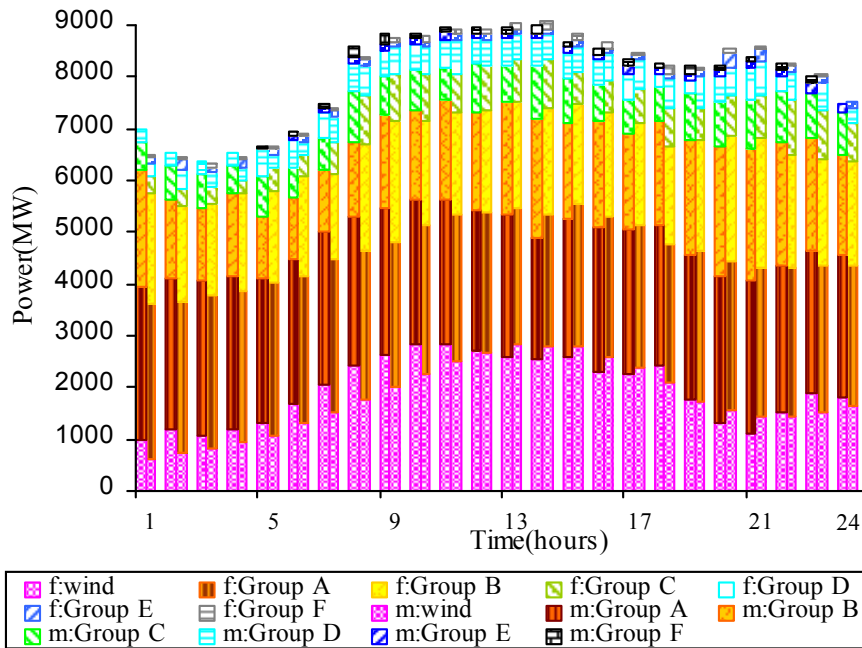
The proposed uncertainty modelling and solution procedure by adaptive particle swarm optimization was applied to a generation system with 12 generators and a wind farm serving a mean load of 8GW. The conventional generation system is divided into six groups. Group A (2\*1500MW) and Group B (3\*1000MW) consists of low cost generating units and act as the base load generation. Group C (2\*750MW) and Group D (2\*500MW) have fast ramping and medium cost generating units. The fast start-up but expensive units are located in Group E (1\*300MW) and Group F (2\*200MW). The installed capacity of the wind farm is

3975MW. The average penetration level of the wind power is about 25% and the distribution is as shown in Fig. 6.10 (30.03.2006). The reliability measure of the power system ensures that the spinning reserve for compensating the forecast errors in load and unforeseen generator outages should be 10% of the load and the reserve for unpredictable fluctuation in wind power to be 20% of the total wind power generation ( $us\%=20\%$ ,  $ds\%=20\%$ ). The maximum up/down spinning reserve contribution by any individual generator should not exceed 20% of its rated capacity. The ramp rates of the units are set at 60% of their rated capacity.

### 6.3.5.1 Deterministic Solution

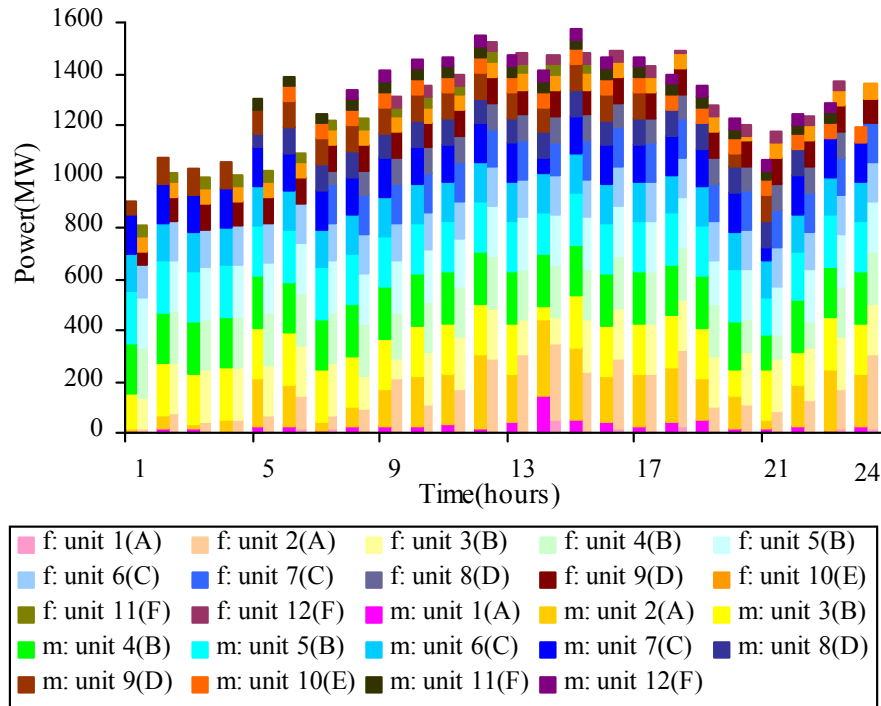
#### (i) Case A: Forecast Vs Measured Data

In this case study (standard unit commitment formulation) the wind generation and load are considered as deterministic quantities. In order to check the influence of forecast errors on system operation, the UC problem was solved for the forecasted data and also for the measured data. The optimal generation levels of different generation areas for the two cases are listed in Fig. 6.14. The total length of



**Fig. 6.14** Generation levels of different generation groups corresponding to measured (m:) and forecasted (f:) data for wind power and load

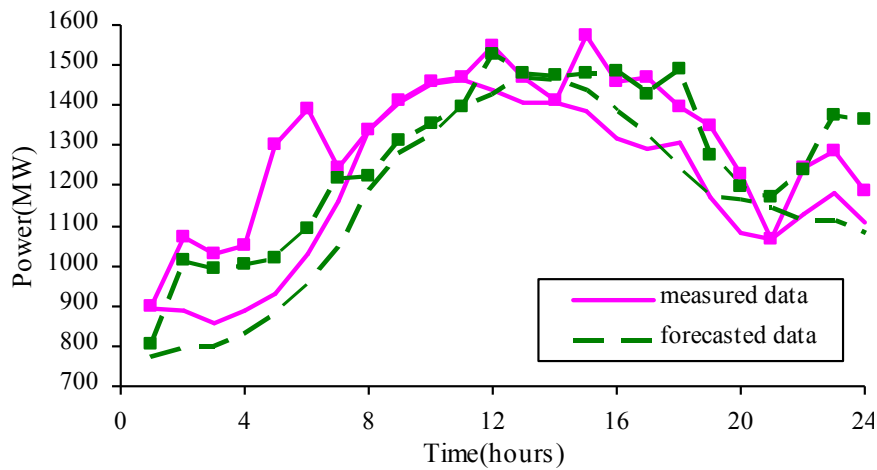
each stack represents the electrical demand at that hour. The contribution of different generation groups is shown by different colors. The difference in the stack lengths corresponding to forecasted and measured data in Fig. 6.14 indicate the forecast errors in electrical demand. The forecast errors cause a considerable change in the output level of different generation areas. The total operating costs of the power system are €3,649,562 for measured data and €3,702,599 for forecasted data. The differences in the reserve capacity allocation and the contributions by different generators are shown in Fig. 6.15. The plain curves in Fig. 6.16 show the changes in the up reserve requirement over time for the forecasted and measured data. The bullet curves show the optimal up spinning reserve allocated by APSO algorithm for the two cases. The optimal allocation almost follows the requirement. The algorithm therefore is capable of optimally scheduling the generators and efficiently dispatching the reserve among the committed units. The down spinning reserve allocation in Fig. 6.17 have a huge deviation from the requirement. This is because all the committed generators are always economically



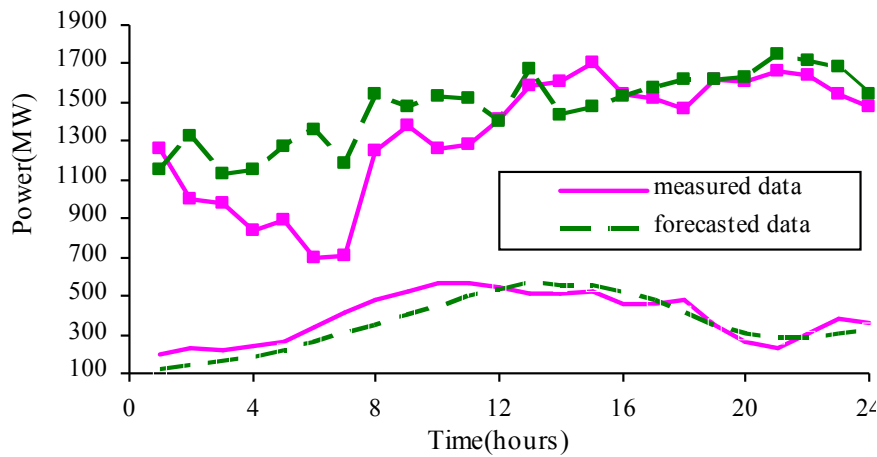
**Fig. 6.15** Reserve contribution by different generators corresponding to measured (m:) and forecasted (f:) data for wind power and load

operating at the nominal power and the down spin,  $DS_{i,t}$  contributed by each generator is therefore high. The optimal UC schedule for the two cases (measured - m:25%, forecasted - f:25%) is listed in Table 6.3. The UC schedule for forecasted

data has unit 7 switched off during the off-peak period and also has extended operation periods for the expensive units. Where as the UC schedule for the measured data has unit 9 and unit 10 operating at a shorter period. So preparing the power system based on the forecasted information is not an optimal planning. The huge difference in the reserve allocation for the two cases indicates the risk involved in planning the power system for a particular scenario.



**Fig. 6.16** Up reserve requirement (plain curve) and optimal allocation (bullet curve) for the forecasted and measured data



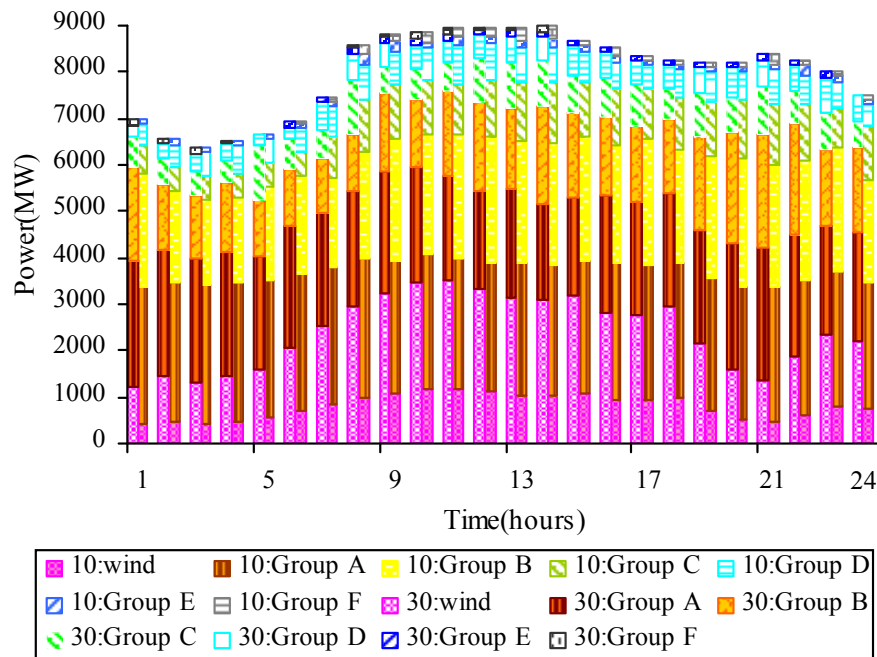
**Fig. 6.17** Down reserve requirement (plain curve) and optimal allocation (bullet curve) for the forecasted and measured data

**(ii) Case B: Different wind penetration levels.**

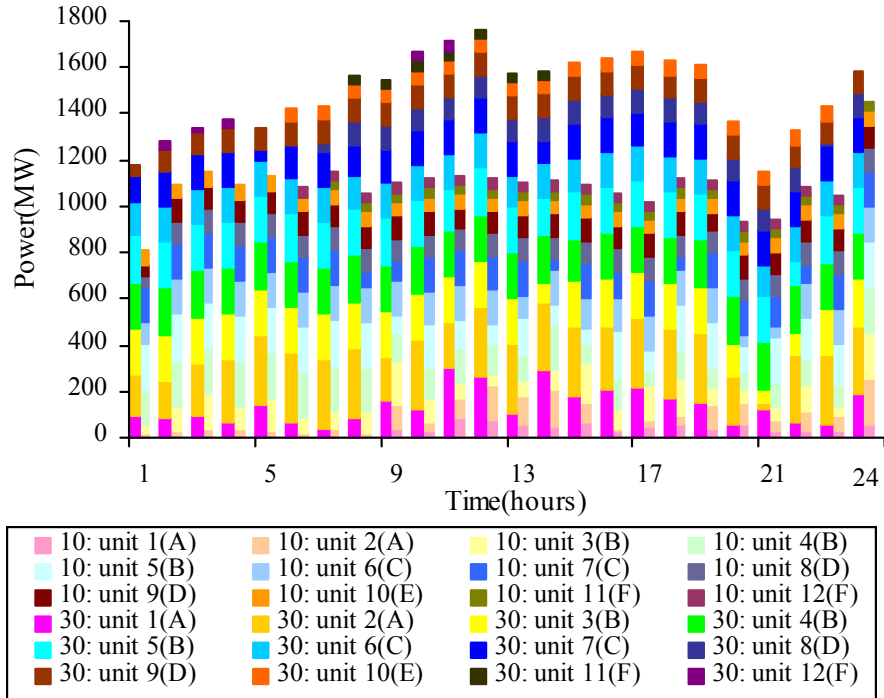
In this case study, measured data is used for wind generation and load. The effect of wind penetration on the power system operation is investigated in case B. The actual wind profile is not altered but is scaled up to meet different average penetration levels. The decline in the operating costs with increased wind penetration is listed in Table 6.2. For wind penetration above 35%, the available resources are able to supply the demand but could not support the up spinning reserve required

**Table 6.2** Total Operating Costs for Different Wind Penetration Levels

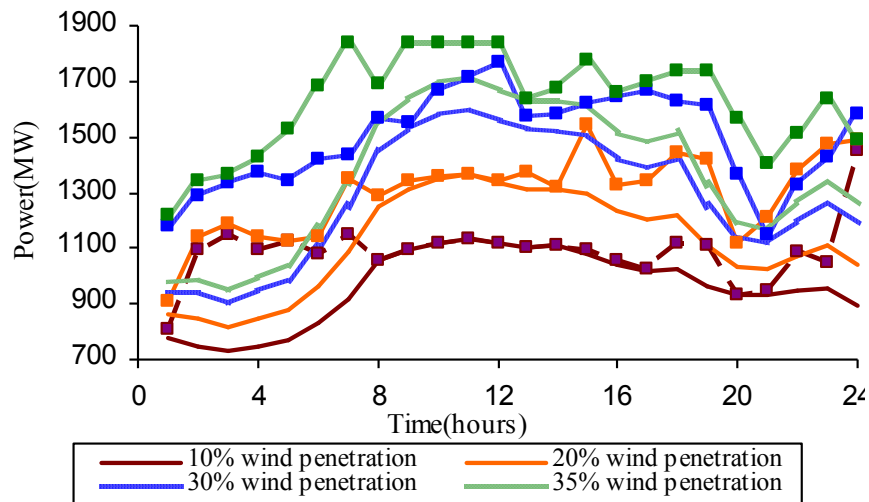
Wind		Total operating cost(€/day)
Average % wind-penetration per day	Average wind power per day(MW)	
10	802	4,263,690
15	1198	4,054,488
20	1600	3,820,804
25	1958	3,649,562
30	2398	3,464,432
35	2798	3,296,060



**Fig. 6.18** Generation levels of different generation groups corresponding to 10% and 30% wind penetration levels

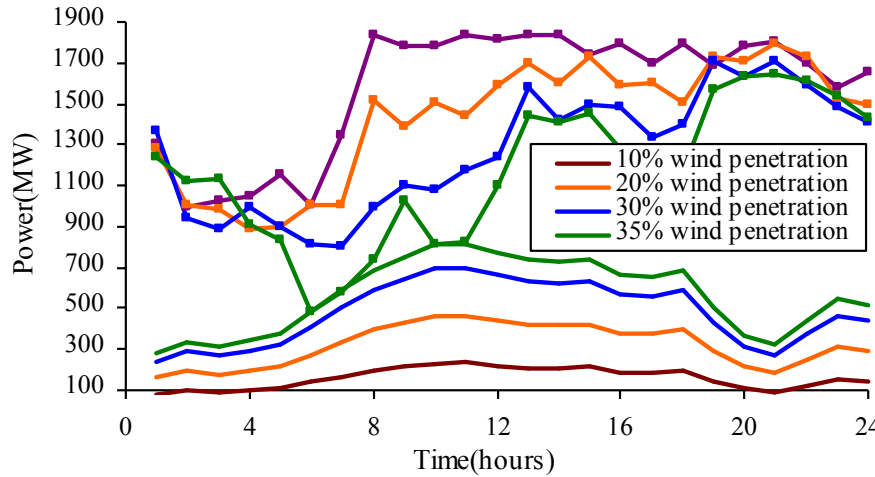


**Fig. 6.19** Reserve contribution by different generators corresponding to 10% and 30% average wind penetration levels and measured data for load



**Fig. 6.20** Up reserve requirement (plain curve) and optimal allocation (bullet curve) for different wind penetration levels

for the reliable operation of the power system. The generation capacity of different groups for 10% and 30% penetration level is shown in Fig. 6.18. The increase in reserve capacity and the contribution by different generators is shown in Fig. 6.19. As the wind level increase, APSO is able to cater the demand among the cost-effective generation groups A, B, C and D and the expensive generation groups E and F are shut down. But when the wind level is greater than 25%, the increase in reserve comes at a higher price than the one at 10%. As shown in Fig. 6.19, the expensive generators of group E and F has to operate at minimal level to support the reserve. This explains the reason why the decrease in the operating cost due to the increase in wind penetration from 10-15% is not the same as the increase from 30-35%. The requirement and allocation of up and down spinning reserve for different wind penetration levels is shown in Fig. 6.20 and Fig. 6.21 respectively. The UC schedule corresponding to different wind penetration levels is presented in Table 6.4.

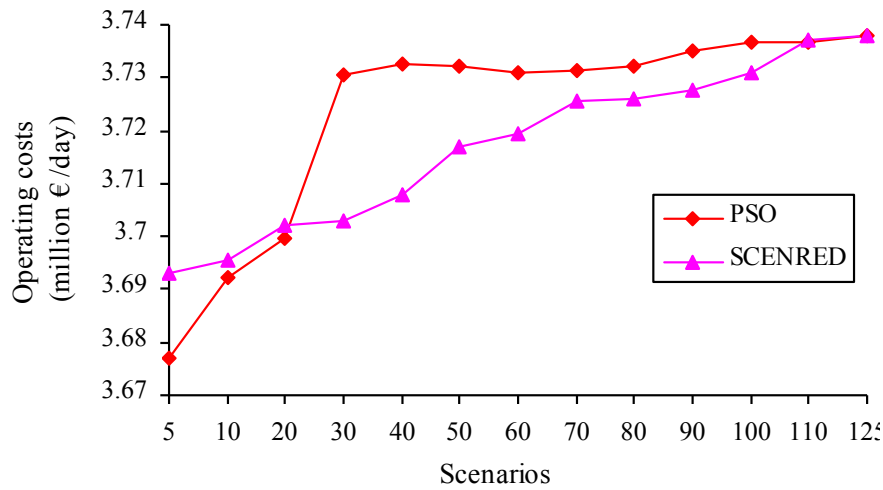


**Fig. 6.21** Down reserve requirement (plain curve) and optimal allocation (bullet curve) for different wind penetration levels

### 6.3.5.2 Performance of Scenario reduction algorithm

The performance of the scenario reduction algorithm was evaluated by comparing the stochastic solution corresponding to the initial bulk tree with the reduced tree generated by the proposed scenario reduction algorithm. 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. 6.22. Similar evaluations were

performed using the reduced scenarios sets obtained using SCENRED software developed by H. Heitsch and W. Römisch [143]. This algorithm is used as reference because its stability is mathematically proved for linear systems. The only drawback is its inability to handle huge number of scenarios. 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. 6.22** Change in total operating cost with number of scenarios representing three branching stages in wind uncertainties

In order to check the evolution of the swarm or the reduced scenario tree, only two branching stages in wind generation uncertainties were considered. The initial scenario tree has two branching stages with five branches at each node, resulting in 25 scenarios. As mentioned earlier, the flight of one particle influences the fitness of the rest of the particles. As the particles move closer to each other, the fitness (Euclidean distance) decreases and ultimately become zero when they converge on any particular minima. It would be interesting to observe if the new update equations are able to guide the particles to distinct positions in the search space. A swarm of 25 particles is allowed to excavate all the available 25 scenarios in the initial search space. The iterative process is shown in Fig. 6.23. At the first iteration, there were only 18 distinct particles. The remaining seven particles were identical to their nearest neighbors. But as the iterations progress, the swarm is able to identify the missing distinct positions and finally at 15<sup>th</sup> iteration, the swarm had traced all the available scenarios in the search space. This is shown in Fig. 6.23. The convergence of the various particles in a swarm of 25 particles is depicted in Fig. 6.24. In Fig. 6.24 (a) particle 4 has a very high fitness value where



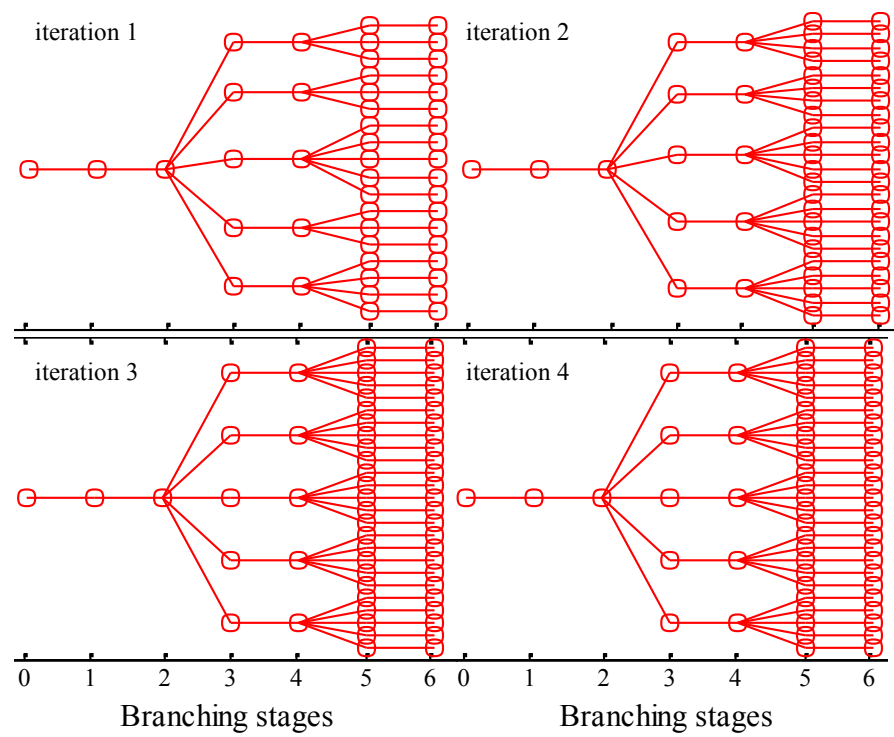


Fig. 6.23 Evolution of the swarm/reduced scenario tree

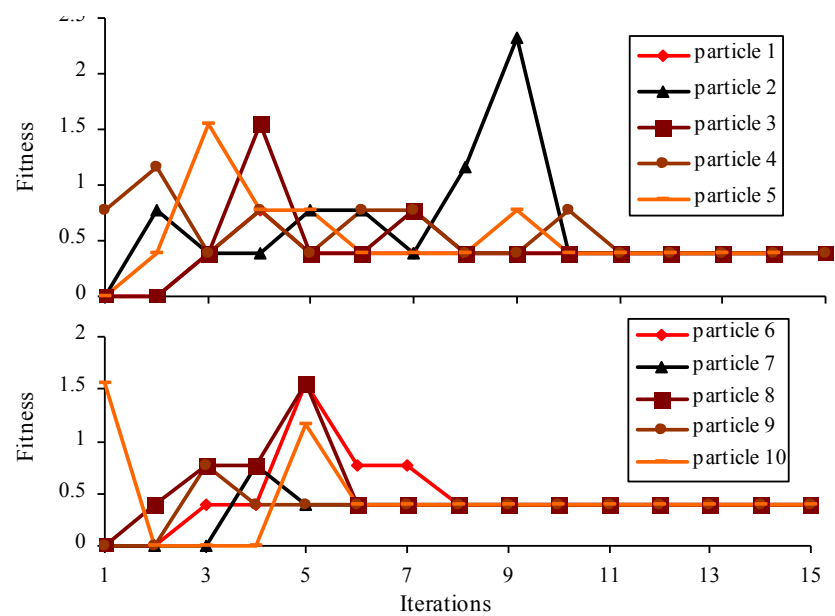


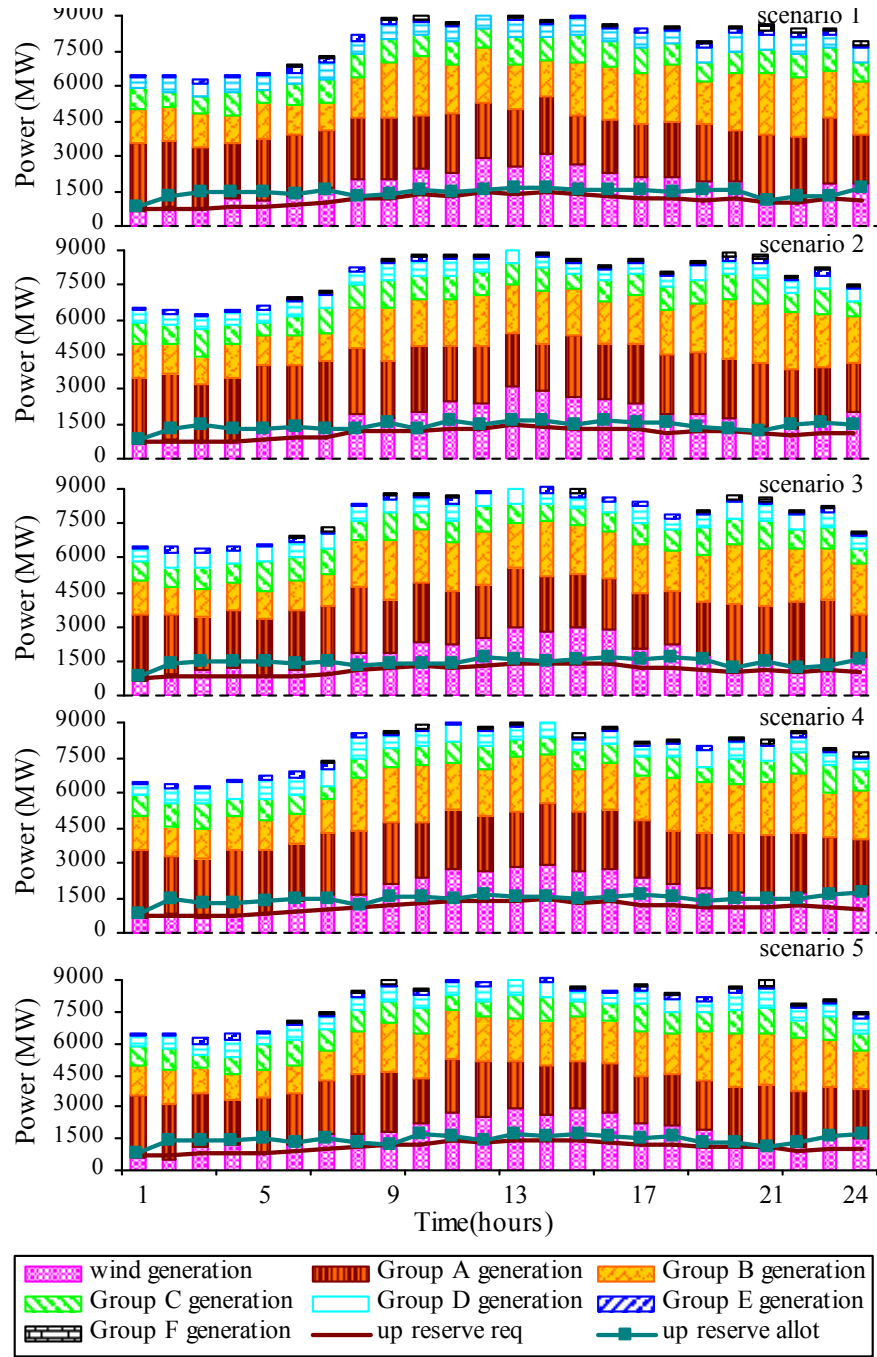
Fig. 6.24 Convergence of the particles in scenario reduction algorithm by PSO

as the other particles 1, 2, 3, and 5 start with zero fitness value. This implies that these particles are very close to each other. These particles therefore try to explore promising areas to improve their fitness without degrading their nearest neighbours. 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 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 neighbour is zero and hence its fitness is also zero. 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.

### 6.3.5.3 Stochastic Approach

For solving the stochastic cost model, the stochastic nature of the uncertainties at each hour of the scheduling period is considered. The evolution of the uncertainties in wind generation and load is modeled as a multistage scenario tree with a branching stage at each hour of the planning period. For a 24 hour planning period, there are 23 branching stages and 25 samples at each node of the scenario tree i.e.  $25^{23}$  scenarios. The initial set of scenarios was reduced to 5, 10, 20, 30, 40 and 50 scenarios. The problem size increases with the increase in the number of scenarios. The results were computed on a 3GHz 1GB RAM Intel Pentium 4 CPU. With the rise of fast and efficient computing tools, huge stochastic model can be easily solved. So the solution methodology and the optimal solution should not be compromised due to inability of the computing devices. Hence it is necessary to develop scenario reduction algorithms that can handle huge set of initial scenarios and stochastic cost model that can be solved over huge set of scenarios.

The UC solution based on forecast information is optimal only to a particular scenario and is not optimal for the scenario that may actually occur. With the huge fluctuations in wind power generation and high forecast errors, it is not advisable to plan power system operation based on deterministic models. The stochastic models try to find a near optimal solution considering all possible scenarios. The stochastic solution may not be a global optimal solution to the individual scenarios but it is a robust solution over all possible realizations of the uncertainties. The power dispatch among the generation groups for a five reduced scenario set is shown in Fig. 6.25 and the corresponding UC schedule in Table 6.5. The wind contribution to the total demand is also shown in the figure. Each scenario has a different load and wind generation profile. The stochastic cost model will deliver a UC schedule which is optimal to all the five scenarios. Based on this schedule, each scenario has its own optimal dispatch and reserve allocation. The costs (RS)



**Fig. 6.25** Optimal generation levels and reserve allocation corresponding to five reduced scenarios with a common unit commitment schedule

**Table 6.3** Duty Cycle based Unit Commitment Schedule for Different Wind Penetration Levels

	Unit 1-6	Unit 7	Unit 8	Unit 9	Unit 10	Unit 11	Unit 12
Measured	24 0 0 0 0	24 0 0 0 0	-4 18 0 0 -2	17 -2 2 0 -3	-5 19 0 0 0	-4 19 0 0 -1	-7 16 0 0 -1
Forecast	24 0 0 0 0	-6 18 0 0 0	-6 17 0 0 -1	24 0 0 0 0	24 0 0 0 0	-15 0 0 0 -9	-8 15 0 0 -1

**Table 6.4** Duty Cycle based Unit Commitment Schedule for Different Wind Penetration Levels

Wind Penetration	Unit 1-6	Unit 7	Unit 8	Unit 9	Unit 10	Unit 11	Unit 12
m:10%	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-6 18 0 0 0	-5 18 0 0 -1
m:15%	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	3 -4 17 0 0	-6 18 0 0 0	-8 14 0 0 -2
m:20%	24 0 0 0 0	-6 18 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	23 0 0 0 -1	24 0 0 0 0
m:25%	24 0 0 0 0	24 0 0 0 0	-4 18 0 0 -2	17 -2 2 0 -3	-5 19 0 0 0	-4 19 0 0 -1	-7 16 0 0 -1
m:30%	24 0 0 0 0	24 0 0 0 0	-6 18 0 0 0	24 0 0 0 0	-5 18 0 0 -1	-7 7 0 0 -10	4 -5 2 0 -13
m:35%	24 0 0 0 0	24 0 0 0 0	-3 18 0 0 -3	-5 18 0 0 -1	-5 8 0 0 -11	-5 19 0 0 0	15 0 0 0 -9

**Table 6.5** Duty Cycle based Unit Commitment Schedule for Different Set of Reduced Scenarios

Scenarios	Unit 1-7	Unit 8	Unit 9	Unit 10	Unit 11	Unit 12
5	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-12 3 -4 5 0	-5 19 0 0 0
10	24 0 0 0 0	-5 19 0 0 0	24 0 0 0 0	24 0 0 0 0	-4 20 0 0 0	-6 18 0 0 0
20	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-4 20 0 0 0	24 0 0 0 0
30	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-5 19 0 0 0	-3 21 0 0 0
40	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-5 19 0 0 0
50	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	24 0 0 0 0	-3 21 0 0 0

**Table 6.6** Total Operating Costs for Different Set of Scenarios

Scenarios	Nodes	control variables	constraints	cost (€/day)
1	24	348	1224	3,702,599
5	116	1,452	3912	3,734,340
10	225	2,760	7,272	3,774,047
20	457	5,544	13,992	3,773,548
30	679	8,208	20,712	3,784,218
40	903	10,896	27,432	3,823,241
50	1,124	13,548	34,152	3,843,547

presented in Table 6.6 are the results obtained by solving the stochastic model with reduced set of scenarios. The stochastic solution is high compared to the model can be extended to risk-averse model and operators can plan the power system operation at certain risk to make promising profits [144]. The importance of stochastic modeling and the influence of the stochastic nature of the variables on the cost models can be estimated by  $VSS$  (value of stochastic solution) [145]. This measure is evaluated using the following equation:

$$VSS = RS - EEV \quad (6.34)$$

Expected value ( $EV$ ) is calculated by replacing the random parameters in the cost model by their expected values. The solution from this expected value problem is applied to all the scenarios. The expectation on all these solutions results in  $EEV$ . The value of stochastic solution ( $VSS$ ) reported in Table 6.6 therefore indicates the expected value of savings that can be obtained by using the stochastic cost model. The unit commitment schedule obtained using the stochastic model 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.

### 6.3.6 Conclusion

The variable nature of the wind penetration over the planning period demands for a sophisticated unit commitment formulation for wind integrated power system. The power system operators need a robust unit commitment schedule that can accommodate the intermittent nature of the wind generation. The stochastic cost model addresses the influence of demand and wind generation uncertainties on the optimal operation of the power system. The performance of the proposed scenario reduction algorithm is verified by already existing technique. The proposed uncertainty modeling with particle swarm optimization is able to handle extremely huge number of scenarios. The selection of optimal scenarios is based on an intelligent

search process. Since the reduction process doesn't require a one-to-one comparison, the process is computationally fast. This scenario reduction technique can handle huge number of branching stages and therefore has the ability to model the complete stochastic nature of the uncertainties. The unit commitment solution proposed in this section will assist the power system operators to plan the scheduling of the units considering the future evolution of the load and wind generation uncertainties. The power system therefore can operate at the same reliability as before the introduction of wind power

## 6.3 Hybrid Power Systems for Residential Loads

### 6.3.1 Introduction

The growing concern about global warming and the huge incentives for renewable energy generation from government policies have prompted residential consumers to opt for the most elegant and environmentally benign energy sources like photovoltaic (PV) and the fuel cell. This section presents an optimization approach to reduce the daily operating costs of an autonomous hybrid PV/Fuel cell system with energy storage device supplying a residential load. The hybrid system has to regulate the imbalances between generation and demand at all times. The generation resources (fuel cell and battery) have to operate efficiently to follow the unpredictable fluctuations in residential demand. PV provides the most economical operation, so it has to be completely utilized. However the stochastic nature of PV poses a serious challenge for the optimal operation of this power system. Decisions regarding the commitment and generation capacity of the fuel cell (current actions) have to be made before the random event (demand and PV generation) is revealed. Later, based on the outcome of the random event, the charging and discharging capacities of the battery have to be adjusted (recourse actions). The current actions should be such that the ill effects of the recourse actions should be minimal. These recourse actions have to be done at each hour of the planning period. Therefore a multi-stage stochastic model is developed to study the cost optimal operation of this hybrid power system.

The generation capacity of the PV depends to a great extent on the insolation and temperature. The electrical & thermal demand is very sensitive to climatic changes. The total daily energy consumption of a single household can be estimated to some extent from historical data. But the daily load profile is highly unpredictable. The average load profile depends on the type of the day, the number of inhabitants and their availability at each hour and also on their life style. Since these dependencies can not be mathematically modeled, the evolution of these variables should be considered as random processes. All possible outcomes of these random processes should be taken care while planning for the optimal operation of the hybrid system. The evolution of these random processes which represents the future realizations of the uncertainties is modeled as a suitable scenario

tree. Each scenario is an instance of the future realizations for the uncertainties. This tree gives the complete information of the uncertainties prevailing in the cost model. The better the scenario tree, the better will be the stochastic solution for the cost model.

The scenario tree modeling transforms the cost model into a multistage nonlinear stochastic cost model. The aim of this stochastic model is to minimize the average operating costs over this scenario tree. The decisions generated by this model at any time step  $t$ , before the random event has been observed at that time step will be cost effective to what ever might be the evolution of the uncertainty at time  $t$  and also at later time steps. The resulting stochastic model is solved without decomposition [146] using adaptive particle swarm optimization technique.

### 6.3.2 Problem formulation

#### 6.3.2.1 Deterministic Model

The deterministic cost model consists of a photovoltaic (PV), fuel cell, gas boiler and a lead acid storage battery supplying both electrical and thermal energy to a single household. The planning horizon is of 24 hours and is split into 24 equal subintervals. The operation cost of the system consists of the daily fuel cost ( $DFC$ ), start-up cost ( $STC$ ), maintenance cost ( $MC$ ) of the fuel cell and the cost of gas ( $CG$ ) for the boiler. The boiler operates only then the fuel cell could not meet the thermal demand. There are no costs involved in the operation of the PV and the battery storage.

The daily fuel cost ( $DFC$ ) includes the natural gas price ( $CFC$ ) for generating active power  $P$  and the cost of the power used by the auxiliary devices ( $P_a$ ) and is formulated as shown below:

$$DFC = C_{FC} U_t \frac{P_t + P_a}{\eta_t} \quad (6.35)$$

Here  $U_t \in \{0, 1\}$  indicates the commitment decision (1 if on, 0 if off) variable of the unit at time  $t$ . The fuel cost [147] is dependent on the efficiency ( $\eta$ ) of the fuel cell which is a polynomial function of the active power generation.

$$\eta = 0.4484 - 0.05359P_t + 0.01267P_t^2 - 0.00182P_t^3 \quad (6.36)$$

The start-up cost,  $STC$  depends on the hot and cold start-up costs ( $\alpha$ ,  $\beta$ ) and the off time of the unit. The maintenance cost,  $MC$  is proportional to the power generation and is therefore constant for kWh.

$$STC = U_t(1 - U_{t-1})(\alpha + \beta(1 - e^{\frac{-T_{t,off}}{\tau}})) \quad (6.37)$$

$$MC = K_{MC}P_t \quad (6.38)$$

The operation cost,  $CG$  of the boiler is modeled as follows:

$$CG = C_G \max(D_{th,t} - P_{th,t}, 0) \quad (6.39)$$

Where  $C_G$  is the gas price,  $D_{th,t}$  is the thermal demand and  $P_{th,t}$  is the thermal power generated by the fuel cell.

$$P_{th,t} = -0.0715 + 1.15P_t - 0.0236P_t^2 + 0.0442P_t^3 - 0.0037P_t^4 \quad (6.40)$$

The aim of the optimization is to find an operating schedule for the system with minimum operation costs. The objective function is as follows:

$$\min \sum_{t=1}^T (DFC + STC + MC + CG) \quad (6.41)$$

The unit constraints include

- The minimum and maximum unit rated capacities

The operation levels of the fuel cell and battery are limited by the lower bounds  $P^{\min}$ ,  $BL^{\min}$  and upper bounds  $P^{\max}$ ,  $BL^{\max}$ .

$$P^{\min}U_t \leq P_t \leq P^{\max}U_t \quad (6.42)$$

$$BL^{\min} \leq BL(t) \leq BL^{\max} \quad (6.43)$$

- Ramp rates

$$P_tU_t - P_{t-1}U_{t-1} \leq \Delta P_U \quad (6.44)$$

$$P_{t-1}U_{t-1} - P_tU_t \leq \Delta P_D \quad (6.50)$$

$\Delta P_U$  and  $\Delta P_D$  are the upper and lower limits for the ramp rate.

- Minimum up/down time limits of the units

$$(T_{t-1}^{\text{on}} - MUT)(U_{t-1} - U_t) \geq 0 \quad (6.45)$$

$$(T_{t-1}^{\text{off}} - MDT)(U_t - U_{t-1}) \geq 0 \quad (6.46)$$



Where  $T_{t-1}^{\text{on}}$ ,  $T_{t-1}^{\text{off}}$  are the unit on and off times at interval  $(t-1)$ ,  $MUT$ ,  $MDT$  are the minimum up and down time limits.

- Switching frequency of the fuel cell

The maximum number of times the fuel cell can be switched on /off is controlled by the following constraint.

$$n_{\text{start-stop}} \leq N_{\text{max}} \quad (6.47)$$

- Initial and final charge levels for the battery

$$BL(0) = BL^0 \quad (6.48)$$

$$BL(T) = BL^T \quad (6.49)$$

The charge and discharge of the battery can be calculated from the following equations.

$$BL(t+1) = BL(t) - BD(t) * \eta_C \quad (6.50)$$

$$BL(t+1) = BL(t) - BD(t) / \eta_D \quad (6.51)$$

Where  $BL_t$  and  $BL_{t-1}$  are the battery levels at the beginning and end of the interval  $t$  respectively,  $BD(t)$  is the power delivered to/from the battery. It is positive if power is delivered from the battery and is negative if power is delivered to the battery.  $\eta_D$  and  $\eta_C$  are the battery discharging and charging efficiencies respectively.

To meet the electrical load ( $P_L$ ), the system has to optimally operate in one of the following modes.

- (a) Fuel cell only

$$P_{L,t} = P_t \quad (6.52)$$

- (b) PV only

$$P_{L,t} = P_{PV,t} \quad (6.53)$$

- (c) Battery only

$$P_{L,t} = BD(t) \quad (6.54)$$

- (d) Fuel cell and PV supplying load

$$P_{L,t} = P_t + P_{PV,t} \quad (6.55)$$

(e) Fuel cell and Battery supplying load

$$P_{L,t} = P_t + BD(t) \quad (6.56)$$

(f) Battery and PV supplying load

$$P_{L,t} = BD(t) + P_{PV,t} \quad (6.57)$$

(g) Fuel cell, battery and PV supplying load

$$P_{L,t} = P_t + P_{PV,t} + BD(t) \quad (6.58)$$

### 6.3.2.2 Stochastic Extension

The deterministic model described above assumes that the information regarding the load profile and PV generation can be either estimated or forecasted. However in real time these variables can not forecasted accurately. For instance the electrical power utilization of a single household over a 24 hour planning horizon is highly unpredictable. The inhabitants may follow regular activities that involve the use of stove, oven, coffee maker, refrigerator, dishwasher, television, computer, lighting etc. but the time when these devices will be used during the day can't be estimated with any degree of precision. This depends on the life-style and personal behavioral characteristics [148] of the inhabitants. These characteristics can not be mathematically modeled. The only way to model this uncertainty is to use the scenario tree analysis where all possible occurrences of the uncertainty are considered. The basic cost model has to be reformulated as a multistage stochastic cost model.

$$\min \sum_{t=1}^T E(DFC + STC + MC + C_G \max(D_{th}(\omega) - P_{th,t}, 0)) \quad (6.59)$$

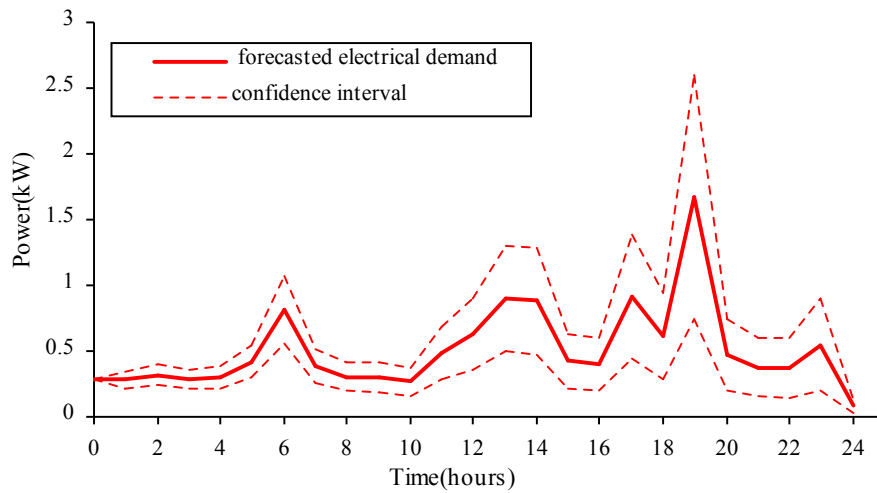
Where  $\omega$  indicates stochasticity. The stochastic load balance equation is as follows:

$$PL(\omega) = P_t + P_{PV}(\omega) + BD(t) \quad (6.60)$$

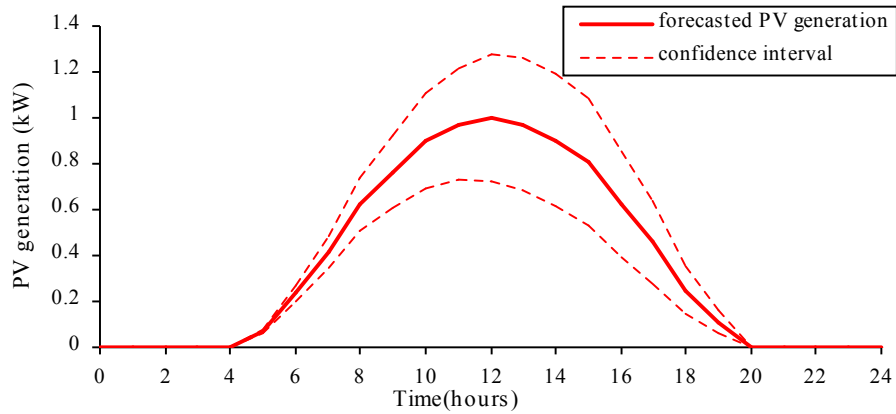
This model aims at minimizing the expectation of the operating costs of the system.

### 6.3.3 Scenarios

The significant uncertainties associated with the above cost model are the electrical, thermal demand and PV generation. They are considered as a multi-variant random process. The uncertainty is assumed to increase with time. The increase in uncertainty of electrical demand and PV generation is as shown in Fig. 6.26. The



**Fig. 6.26** The evolution of the electrical demand with time

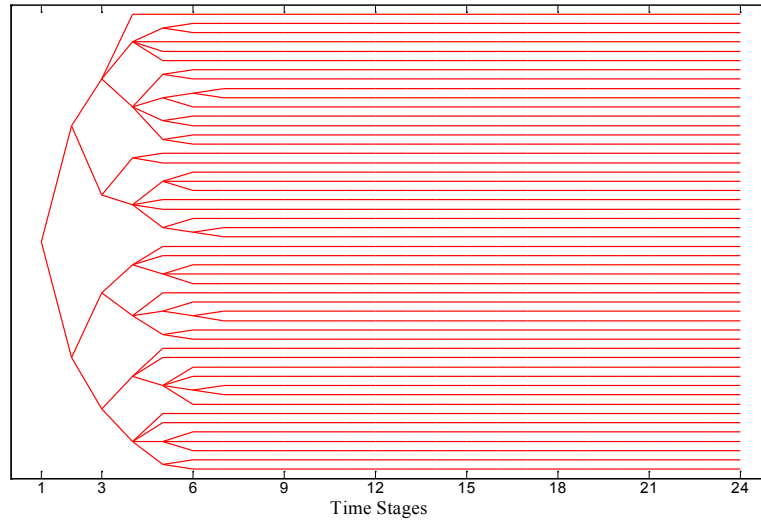


**Fig. 6.27** The evolution of the PV generation with time

deviation of the forecasted uncertain variables (electrical, thermal demand and PV generation) is 20% and increases to 45% of the forecast with time. The bold curve in Fig. 6.26 shows the forecasted electrical demand. The other two curves show the 90% confidence interval of the forecasted demand. The marginal distribution

for each of these variables is assumed to be normal distribution. For simplicity, each marginal distribution is approximated to five discrete samples at every time step. Hence the random process has 125 samples at each time stage. Each of these samples has equal probability. The information regarding the three random variables is completely defined at  $t=1$ . The evolution of the random process for the next 23 hours has to be modeled using scenario analysis. Hence there are 23 branching stages and the random process evolves into  $23^{125}$  scenarios. Each of these scenarios represents a future realization of the random process. This type of uncertainty modeling results in a multistage scenario tree with 23 branching stages and 125 samples at each stage. So each node ( $n$ ) of the scenario tree has 125 equally probable successor nodes ( $n^+$ ). To solve this stochastic model, the random process with huge set of scenarios has to be approximated to a simple random process with finite set of scenarios and should be as close as possible to the original process.

By considering all possible outcomes of the random process, the stochastic nature of the multivariate random process can be completely captured. This will help to develop better stochastic cost models which will help in planning and operation of the system. Hence there is a need to develop such algorithm which could handle extremely huge number of scenarios. The scenario reduction algorithm describes in chapter 5 is used to generate an appropriate scenario tree as shown in fig. 6.28.



**Fig. 6.28** Scenario tree for stochastic electrical, thermal demand and PV generation generated by PSO for 24 branching stages and 5 branches at each stage

### 6.3.4 Solution procedure

Using the above mentioned scenario tree modeling for the uncertainties, the basic cost model can be remodeled using stochastic programming approach as shown below.

$$\min \sum_{n \in N} \pi_n \left( C_{FC} U^n \frac{P^n + P_d}{\eta^n} + U^n (1 - U^n) (\alpha + \beta (1 - e^{\frac{-T_{off}^n}{\tau}})) \right) + K_{MC} P^n + C_G \max(D_{th}^n - P_{th}^n, 0) \quad (6.61)$$

Where,  $n$  refers to the node index and  $N$  represents the total number of nodes of the reduced scenario tree. The above model is subjected to the constraints (6.42)-(6.51). The stochastic load balance equation reads as follows:

$$PL^n = P^n + P_{PV}^n + BD^n \quad (6.62)$$

The aim of this model is to find a unit commitment schedule for the fuel cell, to find an optimal operation mode (6.52)-(6.58) for the hybrid system at each time step and to minimize the expectation of the operation cost of the model over the whole scenario tree. This model is solved by multi-stage stochastic programming approach. This solution technique requires a set of decision variables at each node of the scenario tree. Therefore the size of the optimization increases with the size of the scenario tree. APSO is used to solve the above problem.

### 6.3.5 Numerical results

#### 6.3.5.1 Deterministic Model

The test system consists of a fuel cell (1.5kW), PV (1.5kW), gas boiler (2.5kW) and a storage battery supplying a residential load (peak electrical demand=2.338kW and peak thermal demand=3.236kW). The objective of this deterministic optimization problem is to use the basic cost model to find the fuel cell commitment schedule  $U$ , the power output levels of the fuel cell  $P$  and the battery schedule  $BD$ . There is no thermal storage, so the gas boiler will operate only when the fuel cell thermal output is unable to meet the thermal demand. The optimization is carried out at each time step of one hour, over a scheduling period of 24 hours, so that the total operating costs are minimized subjected to unit and system operating constraints. The charging and discharging efficiency of the battery is assumed to be 66%. The load profiles of a single household on a working day in July at Baden-Württemberg, Germany are considered (Fig. 6.29). The insolation profile is for the month of July. The PV generation is available from 5AM to 7PM.

The hourly scheduling of PV/Fuel cell/ Battery system is shown in Fig. 6.30. In this figure, positive discharge  $BD$  corresponds to hours when battery is discharging, while negative discharge corresponds to charging hours of the battery. The fuel cell charges the battery during the off-peak period. The excess PV generation during the day also charges the battery. The battery supports the peak demand in the morning and evening. The use of battery smoothens the load

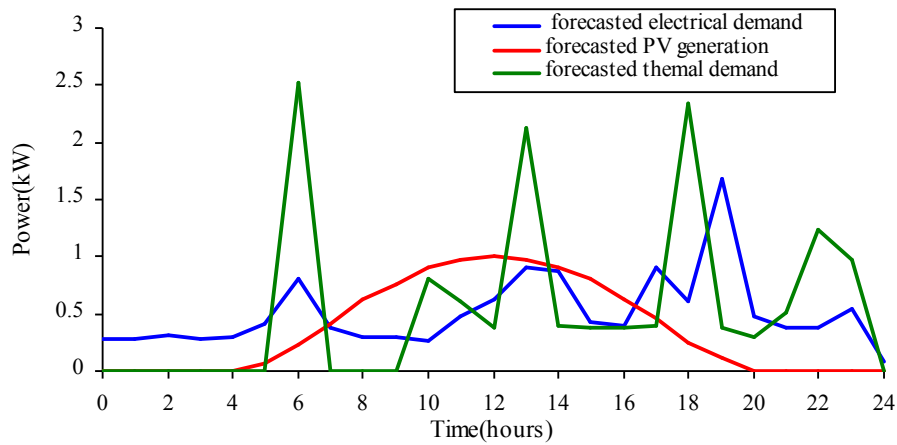


Fig. 6.29 Electrical and thermal load profile considered in the simulation

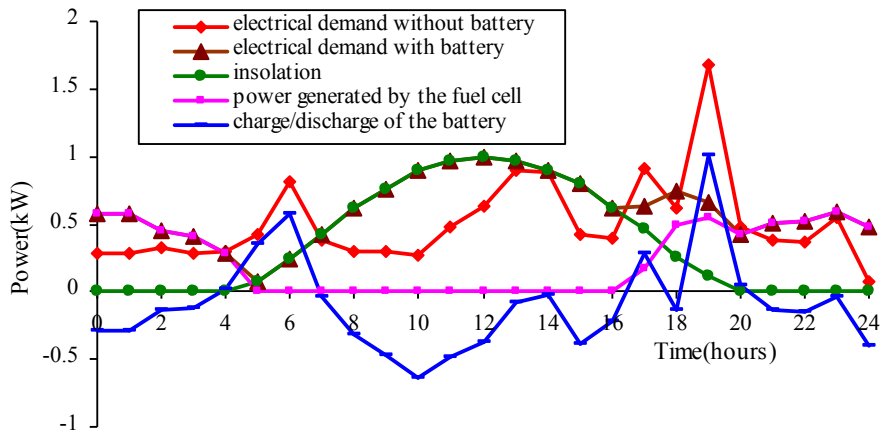


Fig. 6.30 hourly scheduling of PV/Fuel cell/Battery units based on deterministic optimization

profile. The new load profile has no peaks. This is shown by the curve with triangle bullets. The fuel cell operates in the early hours of the morning when there is no PV generation to supply the load and also to charge the battery. It is switched off when PV is in operation. In the evening when the load is high and PV generation is scarce, the fuel cell is again switched on. The total operating cost for the whole day amounts to 2.09euro or 0.16euro/kWh.

### 6.3.5.2 Stochastic Extension

Multistage stochastic programming approach is used to solve the stochastic cost model. The optimization was solved using the adaptive particle swarm optimization. The initial set of scenarios was reduced to fifty scenarios as shown in Fig. 6.28 using the PSO based scenario reduction algorithm. This reduced set of scenarios is used in the stochastic model to represent the uncertainties. The average operating cost associated with this model amounts to 2.86 euro/day. The solution set generated by the deterministic model is optimal only for the forecasted scenario. But this scenario may not certainly happen. So planning the operation of the power system based on this cost model may lead to poor utilization of the PV and battery, which results in expensive operation schedule for the fuel cell and the boiler. More over if the battery is not properly utilized, the peak loads may not be met. Stochastic models will generate robust generation plans for reliable power supply. In multi-stage models, decisions are made at the current time stage and the recourse actions are decided after the uncertainty is revealed. Solutions to these models are obtained by minimizing the costs associated with the recourse actions. Therefore the decisions made at any time step are optimal to any scenario for the uncertainty at a later time step. Decisions regarding the amount of electrical and thermal power to be generated from fuel cell are made before the demand at that hour is known. After the demand and PV generation are revealed, the generation-demand imbalances are adjusted using the battery and the boiler. The charging/discharging and generation of the boiler form the recourse actions. Optimizing these imbalances results in efficient utilization of the battery and the boiler; and reducing the overall operation cost of the power system. The stochastic solution corresponding to five different scenarios selected from the reduced fifty scenario set is shown in Fig. 6.31. In all the scenarios, the operation of the fuel cell is optimally planned so as to completely utilize the PV generation. The peak demand is always supplied irrespective of the uncertainties. Hence stochastic models provide solutions for reliable operation of the hybrid power system.

### 6.3.6 Conclusion

This section presented a solution for a day-ahead operation of a PV-Fuel system with battery storage considering the electrical & thermal demand and PV generation uncertainties. The stochastic cost model can be used to predict a unit commitment schedule for the optimal operation of the system with unpredictable uncertainties. The uncertainty modeling and stochastic programming approach presented in this section can be very useful for planning and operation of the power system subjected to the influence of many uncertainties.

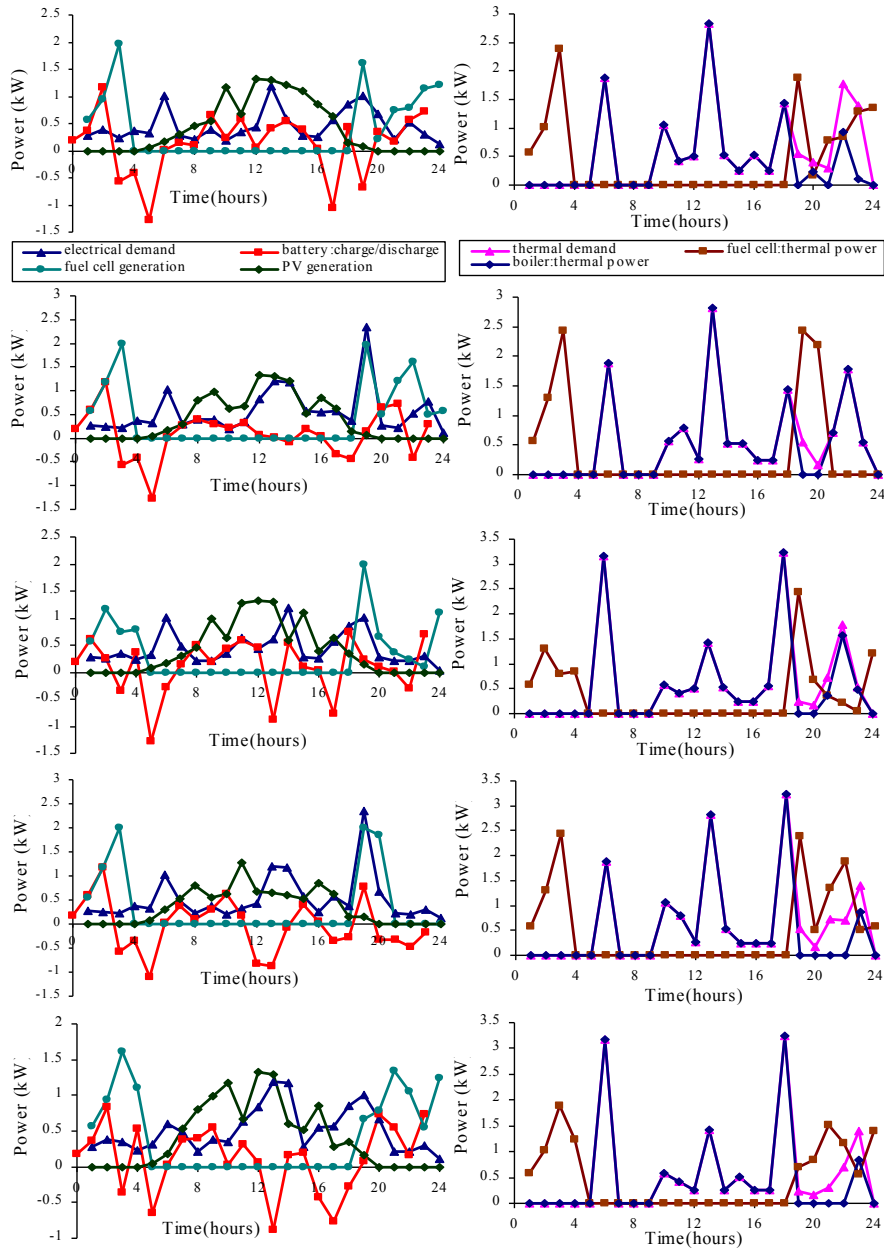


Fig. 6.31 Multi-stage stochastic solution for a set of five scenarios



## Chapter 7

### **Conclusions**

#### **7.1 Summary**

The main focus of this dissertation is on finding an effective black box optimization tool for solving complex and high dimensional problems in power systems. Conclusions on the major contributions described in the thesis will be listed in this chapter.

##### ***7.1.1 Optimization algorithm***

Unlike many evolutionary based search algorithms, Particle Swarm Optimization is very simple to understand and implement. More over it has very few problem dependent parameters. The proposed PSO algorithm reduces the burden of selecting the right set of parameters for a given problem such as swarm size, acceleration coefficients, neighbourhood size, topology etc. The algorithm is completely parameter free and independent of the nature of the problem being solved. The algorithm learns from its search progress. It inherits new information in the form of new particles and unloads unwanted information by dumping some of its worst performing particles. Thus it manages its population in order to trace optimal solutions. The results of this algorithm are either superior or identical to manually tuned PSO variants. This is evident from the statistical results presented in Table 2.1 and 2.2. This proves that the new algorithm on an average performs better than the already existing PSO variants. However the probabilistic comparison of the results in Table 2.3 indicates that the probability that the proposed algorithm results in solutions either similar or better than the other variants is only 59%. This implies the presence of outliers in the solutions over several independent runs. In other words the consistency or repeatability of high quality solutions is low. The algorithm needs further improvements in this direction.

The major challenge in constrained optimization is the selection of appropriate weights for the penalties. These weights vary on the nature of the constraint and also on the level of violation. These parameters should be manually tuned and therefore can not be used in black box optimization tools. The proposed self learning penalty function approach (AP) has the ability to adapt the magnitude of the penalties based on the performance of the swarm. The second penalty function technique addressed in this thesis is based on certain rules (RBP) and is completely parameter free. These approaches are the only means to tackle constraints in black box optimization. The statistical results in Table 3.2 indicate that RBP and AP perform better than static (SP) or dynamic (DP) penalty approaches on

most benchmark problems. Although SP and DP dominate on some functions, the compromise is on rigorous penalty coefficients tuning. The performance of the proposed approaches was consistent over several independent runs (3.3). They have better search capabilities and therefore require fewer function evaluations and smaller swarm size to obtain global solutions. They also have the ability to trace feasible solutions in any environment (Table 3.6). The directional alternative hypothesis conducted using Mann-Whitney U tests further validate the superior probabilistic performance of the new approaches. This research is a valuable contribution in developing black box constrained optimization tool.

### ***7.1.2 Power system operation***

The most significant task in power system operation is the optimal scheduling of the various generation resources to meet the demand and reserve requirements i.e. unit commitment. The duty cycle based unit commitment approach proposed in chapter 4 resulted in 80 percent reduction in the number of variables. This enormous reduction in the search space dimension had resulted in faster convergence and considerable reduction in the overall operation cost of the power system. The new modelling technique had enhanced the exploration of the swarm even without the repair operators which were mandatory in binary UC modelling. The swarm was able to observe the sudden changes in demand and therefore schedule the generators in such a way that a good trade off is maintained between operation cost and reliability. It was also proved that by using special operators such as reserve manager and demand equalizer, the quality of the solution can further be improved.

The number of duty cycles assigned to each unit is fixed. It is evident from Table 4.5 that some units (coal-fired) require less than the assigned duty cycles whereas fast switching units (diesel engines) require more duty cycles. Future research in this area should try to make the number of duty cycles assigned to each unit adaptive. It means the number of duty cycles required for each unit has to be decided during the optimization process. This leads to a new area of optimization called dimensionless optimization i.e. an optimization process where the dimension of the search space is not fixed but decided during the search process.

### ***7.1.3 Uncertainty modelling***

The scenario reduction algorithm presented in chapter 5 offers better uncertainty modelling with less computational effort. This is first of its kind with computational intelligence techniques. The proposed approach eliminates the need for one to one comparison that is required for scenario reduction in traditional approaches and thereby abolishes the restrictions imposed on uncertainty modelling. The approach provides a new challenging optimization problem wherein the flight of one

particle alters the fitness of the remaining particles. The results in Fig. 6.23 and 6.25, show that the particles have sufficient information about their neighbours to improve their fitness without degrading the fitness of their neighbours. The improved qualitative performance as compared to the well established algorithm in Fig. 6.22 shows that this is a good choice for scenario reduction process. The future advancement in this area is to address the quantitative stability of this algorithm.

### ***7.1.4 Optimization under uncertainty***

Planning the operation of a power system with renewable generation in feed based on the forecast information as shown in chapter 6 is not reliable. The day ahead wind generation forecast involves nearly 60% marginal error. If the unpredictable nature of the renewable power is not modelled as an uncertainty quantity, the power system has to operate at a lower reliability margin. This is because, the huge forecast error indicates that there is possibility for a sudden change (excess or deficit) in power generation. In case of excess generation, the economical renewable energy has to be curtailed or during dearth generation, the deficit energy has to be purchased from the spot market. More over the conventional generators should be scheduled such that they are ready to ramp up/down to follow the changes in random renewable generation. The stochastic cost models presented in chapter 6 are capable of generating robust generation schedule considering the renewable generation as a stochastic process. The UC schedules generated by these models are not the global optimal solutions to any particular scenario but are optimal over a set of scenarios. The power system is therefore planned with regard to the predefined reliability level irrespective of the actual realization of the uncertainty. The unforeseen changes in the renewable generation can be optimally absorbed by the scheduled generators. Similar approach can be extended to develop stochastic risk models.



## References

- [1] J. Kennedy and R.C Eberhart, "Particle swarm optimization," Proc.of IEEE International Conf. on Neural Networks, IV, pp. 1942–1948, Piscataway, NJ, 1995.
- [2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," Proc. of Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, October 1995.
- [3] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Academic Press, 2001.
- [4] M. Richards, D. Ventura, "Dynamic sociometry in particle swarm optimization," Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing, pp. 1557–1560, North Carolina, 2003.
- [5] Y. Shi and R.C. Eberhart, "Parameter selection in particle swarm optimization," Proceedings of the Seventh Annual Conference on Evolutionary Programming, pp. 591–600, New York, 1998.
- [6] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 69–73, Piscataway, NJ, 1998.
- [7] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1951–1957, 1999.
- [8] R.C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 84–88, San Diego, CA, 2000.
- [9] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," Proceedings of the Congress of Evolutionary Computation, pp. 1931–1938, Washington D.C., 1999.
- [10] S. A. Hamdan, "Hybrid Particle Swarm Optimiser using multi-neighborhood topologies," INFOCOMP - Journal of Computer Science, Vol. 7, No. 1, pp. 36–43, March, 2008.
- [11] M. Clerc and J. Kennedy, "The particle swarm – explosion, stability and convergence in a multidimensional complex space," IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, pp. 58–73, February 2002.
- [12] J. Kennedy and R.C. Eberhart, "A discrete binary version of the particle swarm algorithm," Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, pp. 4104–4109, Piscataway, NJ, 1997.
- [13] J. Kennedy, R.C. Eberhart, "A discrete binary version of the particle swarm algorithm," Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, pp. 4104–4109, Piscataway, NJ, 1997.
- [14] M. Clerc, "Discrete Particle Swarm Optimization," New Optimization Techniques in Engineering Springer-Verlag, 2004.
- [15] EC Laskari, KE Parsopoulos and MN Vrahatis, "Particle Swarm Optimization for Integer Programming," IEEE Congress on Evolutionary Computation, pp. 1582–1587.
- [16] Y. Shi and R.C. Eberhart, "Fuzzy adaptive particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea, 2001.
- [17] Hongbo Liu, Ajith Abraham, "Fuzzy Adaptive Turbulent Particle Swarm Optimization," Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, pp. 445 – 450, 2005.
- [18] Xiang-han Chen, Wei-Ping Lee, Chen-yi Liao, Jang-ting Dai, "Adaptive Constriction Factor for Location-related Particle Swarm," Proceedings of the 8th WSEAS

- International Conference on Evolutionary Computing, Vancouver, British Columbia, Canada, June 19-21, 2007.
- [19] X. Hu and R.C. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002), pp. 1677-1681, Honolulu, Hawaii USA.
  - [20] C.A. Coello Coello and M.S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," Technical Report EVOCINV-01-2001, Mexico, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN.
  - [21] C.A. Coello Coello, G. Toscano Pulido, and M. Salazar Lechuga, "An extension of particle swarm optimization that can handle multiple objectives," Workshop on Multiple Objective Metaheuristics, Paris, France, 2002.
  - [22] M. R. AlRashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power operations," Electric Power Components and Systems, Vol. 34, No. 12, pp. 1349-1357, December 2006.
  - [23] Yamille del Valle, GK Venayagamoorthy, S Mohagheghi, JC Hernandez, RG Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," IEEE Transactions on Evolutionary Computation, Vol. 12, No. 2, pp. 171-195, 2008.
  - [24] Maurice Clerc, "TRIBES, a Parameter Free Particle Swarm Optimizer" French version: 2003-10-02, presented at OEP'03, Paris, France.
  - [25] Z. Michalewicz, "A Survey of Constraint Handling Techniques in Evolutionary Computation Methods," Proceedings of the 4th Annual Conference on Evolutionary Programming, pp. 135-155, MIT Press, Cambridge, MA, 1995.
  - [26] A.E. Smith and D.W. Coit, "Penalty functions," Section C 5.2 of *Handbook of Evolutionary Computation*, A Joint Publication of Oxford University Press and Institute of Physics Publishing.
  - [27] T. Baeck and S Khuri, "An evolutionary heuristic for the maximum independent set problem," Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 531-535, 1994.
  - [28] A. Homaifar, C. Qi, and S. Lai, "Constrained optimization via genetic algorithms," Simulation, Vol. 62, No. 4, pp. 242-254, 1994.
  - [29] W. Siedlecki and J. Sklanski, "Constrained Genetic Optimization via Dynamic Reward Penalty Balancing and Its Use in Pattern Recognition," Proceedings of the Third International Conference on Genetic Algorithms, pp. 141-150, San Mateo, California, June 1989.
  - [30] J.A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 579-584.
  - [31] R. Farmani and J. Wright, "Self-adaptive fitness formulation for constrained optimization," IEEE Transaction on Evolutionary Computation, Vol. 7, No. 5, pp. 445-455, 2003.
  - [32] A.C.C. Lemonge and H.J.C. Barbosa, "An adaptive penalty scheme in genetic algorithms for constrained optimization problems," in Proceedings of Genetic and Evolutionary Computation Conference, pp. 287-294, New York, 2002.
  - [33] D.W. Coit A.E. Smith and D.M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," ORSA Journal on Computing.
  - [34] D.W. Coit and A. Smith, "Penalty guided genetic search for reliability design optimization," Special Issue on Genetic Algorithms and Industrial Engineering, edited by M. Gen, G.S. Wasserman, & A.E. Sriiith, International Journal of Computers and Industrial Engineering, 1996.
  - [35] Biruk Tessema and Gary G. Yen, "A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization," IEEE Congress on Evolutionary Computation, 2006.

- [36] G.T Pulido, C.A.C Coello, "A constraint-handling mechanism for particle swarm optimization," CEC2004. Congress on Evolutionary Computation, Vol. 2, pp. 1396- 1403, June 2004.
- [37] Y. del Valle, M. Digman, A. Gray, J. Perkel, G.K. Venayagamoorthy, R.G. Harley, "Enhanced particle swarm optimizer for power system applications," IEEE Swarm Intelligence Symposium, pp. 1-7, September 2008.
- [38] David J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, CRC press, Second Edition.
- [39] Frederick J. Gravetter, Larry B. Wallnau, *Statistics for the Behavioral Sciences*, Thomson/Wadsworth press, Seventh Edition, 2007.
- [40] R.C. Johnson, H.H. Happ, W.J. Wright, "Large Scale Hydro-Thermal Unit Commitment Method and Results," IEEE Transactions on Power Apparatus and Systems Volume PAS-90, Issue 3, pp.1373-1384, May 1971.
- [41] P.G. Lowery, "Generation unit commitment by dynamic programming," IEEE Trans. Power App. Syst., Vol. PAS-102, pp. 1218-1225, 1983.
- [42] F. Zhuang, and F.D. Galiana, "Toward a more rigorous and practical unit commitment by Lagrangian relaxation," IEEE Trans. Power Systems, Vol. 3, No. 2, pp. 763–770, May 1988.
- [43] A.I. Cohen, and M. Yoshimura, "A branch-and-bound algorithm for unit commitment," IEEE Trans. Power App. Systems, Vol. PAS-102, No. 2, pp. 444–451, 1983.
- [44] T.O Ting, M.V.C. Rao, C.K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," IEEE Transactions on Power Systems, Vol. 21, Issue 1, pp. 411 – 418, February 2006.
- [45] Zwe-Lee Gaing, "Discrete particle swarm optimization algorithm for unit commitment," IEEE Power Engineering Society General Meeting, Vol. 1, pp. 13-17, July 2003.
- [46] T.O. Ting, M.V.C. Rao, C.K. Loo and S.S Ngu, "Solving Unit Commitment Problem Using Hybrid. Particle Swarm Optimization," Journal of Heuristics, Vol. 9, pp. 507–520, 2003.
- [47] A.Y. Saber, T. Senjyu, A. Yona, T. Funabashi, "Unit commitment computation by fuzzy adaptive particle swarm optimization," IET Generation, Transmission & Distribution, Vol. 1, Issue 3, pp. 456 – 465, May 2007.
- [48] S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, "A genetic algorithm solution to the unit commitment problem," IEEE Transactions on Power Systems, Vol. 11, Issue 1, pp. 83 – 92, February 1996.
- [49] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," IEEE Transactions on Power Systems, Vol. 14, Issue 4, pp. 1452 – 1459, November 1999.
- [50] D.N. Simopoulos, S.D. Kavatza, C.D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm," IEEE Transactions on Power Systems, Vol. 21, Issue 1, pp. 68 – 76, February 2006.
- [51] N.S. Sisworahardjo, A.A. El-Keib, "Unit commitment using the ant colony search algorithm," Large Engineering Systems Conference on Power Engineering, LESCOPE 02, pp. 2 – 6.
- [52] A.H. Mantawy, Y.L. Abdel-Magid, S.Z. Selim, "Unit commitment by tabu search," IEE Proceedings Generation, Transmission and Distribution, Vol. 145, Issue 1, pp. 56 – 64, January 1998.
- [53] T. Senjyu, A.Y. Saber, T. Miyagi, K. Shimabukuro, N. Urasaki, T. Funabashi, "Fast technique for unit commitment by genetic algorithm based on unit clustering," IEE Proceedings Generation, Transmission and Distribution, Vol. 152, Issue 5, pp. 705-713, September 2005.

- [54] T.O. Ting, M.V.C Rao, C.K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," IEEE Transactions on Power Systems, Vol. 21, Issue 1, pp. 411 – 418, February 2006.
- [55] S. K. Tong, S. M. Shahidehpour, and Z. Ouyang, "A heuristic short term unit commitment," *IEEE Trans. Power Systems*, Vol. 6, pp. 1210–1217, August 1991.
- [56] Yuan Xiaohui, Yuan Yanbin, Wang Cheng and Zhang Xiaopan, "An Improved PSO Approach for Profit-based Unit Commitment in Electricity Market," IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005.
- [57] K.S. Swarp and S. Yamashiro, "Unit Commitment Solution Methodology Using Genetic Algorithm," IEEE Power Engineering Review, Vol. 22, Issue 1, pp. 70-71, January 2002.
- [58] C. Christober Asir Rajan and M.R. Mohan, "An evolutionary programming based simulated annealing method for solving the unit commitment problem," *Electrical Power and Energy Systems* 29, pp. 540–550, 2007.
- [59] J.R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, 1997.
- [60] Peter Kall, *Stochastic Programming*, Lecture Notes.
- [61] N. Sahinidis, "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & Chemical Engineering*, Vol. 28, No. 6-7, pp. 971-983, June 2004.
- [62] P. Kall, *Stochastic Linear Programming*, Springer, Berlin, 1976.
- [63] G.B. Dantzig, *Linear programming under uncertainty*, *Management Science* 1, pp. 197-206, 1955.
- [64] V.N.S. Samarasekera and P.K. Varshney, "A fuzzy modeling approach to decision fusion under uncertainty," *Fuzzy Sets and Systems*, Vol. 114, Issue 1, pp. 59-69, August 2000.
- [65] B. Bereanu, "The distribution problem in stochastic linear programming," *Operations Research Verfahren* 8, pp. 22-35, 1970.
- [66] H.I. Gassmann and A.M. Ireland, "On the formulation of stochastic linear programs using algebraic modelling languages," *Annals of Operations Research*, 1996.
- [67] Cheng Seong Khor, "A Hybrid of Stochastic Programming Approaches with Economic and Operational Risk Management for Petroleum Refinery Planning under Uncertainty," PhD thesis, university of Waterloo, Ontario, Canada, 2006.
- [68] András Prékopa, *Stochastic Programming*, Springer, ISBN 0792334825, 9780792334828, 1995.
- [69] Peter Kall and Janos Mayer, *Stochastic Linear Programming: Models, Theory, and Computation*, International Series in Operations Research & Management Science, Vol. 80, Springer, 2005.
- [70] K. Frauendorfer. *Stochastic two-stage programming*, Vol. 392, Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, 1992.
- [71] Dantzig George Bernard, *Linear programming under uncertainty*, *Management Science* 1, pp. 97-206, 1955.
- [72] R.T. Beckafellar and R. J.-B. Wets, "Scenario and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, Vol. 16, pp. 119-147, 1991.
- [73] Jitka Dupacova, "Applications of stochastic programming: Achievements and questions," *European Journal of Operations Research*, 2002.
- [74] W. Römis and R. Schultz, "Multistage stochastic integer programs: An introduction," in: *Online Optimization of Large Scale Systems* (M. Grötschel, S.O. Krumke, J. Rambau eds.), Springer-Verlag, Berlin, pp. 579-598, 2001.
- [75] H. Brand, E. Thorin, C. Weber, J. Hlouskova, S. Kossmeier, M. Obersteiner, A. Schnabl, "Methodology to identify the relevant uncertainties," OSCOGEN, deliverable 3.1, October 2001.
- [76] Sovan Mitra, "A white paper on scenario generation for stochastic programming," *Optirisk systems: white paper series*, ref. No. OPT004, July 2006.



- [77] J. P. DupaCov, B. G. Consigli and S. W. Wallace “Scenarios for Multistage Stochastic Programs,” *Annals of Operations Research* 100, pp. 25-53, 2000.
- [78] P. Kall and W. Stein, *Stochastic Programming*, Wiley, New York, 1994.
- [79] A. Kurawarwala and H. Matsuo, “Forecasting and inventory management of short lifecycle products,” *Operation Research*, Vol. 44, Jan.–Feb. 1996.
- [80] Jitka Dupacova, Nicole Growe-Kuska, and Werner Römischi, “Scenario reduction in stochastic programming,” *Mathematical Programming*, pp. 493–511, 2003.
- [81] W. Römischi and H. Heitschi, “Scenario reduction algorithms in stochastic programming,” *Computational Optimization and Applications*, pp. 187–206, 2003.
- [82] Michal Kaut and Stein W. Wallace, “Evaluation of scenario generation methods for stochastic programming,” *Pacific Journal of Optimization*, pp. 257-271, 2007.
- [83] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *Proc. of the Seventh Annual Conference on Evolutionary Programming*, pp. 591-600, New York, 1998.
- [84] A. Ferguson and G.B. Dantzig, “The allocation of aircraft to routes: an example of linear programming under uncertain demands,” *Management Science* 3, pp. 45-73, 1956.
- [85] S.-E. Fleten and T. K. Kristoffersen, “Short-term hydropower production planning by stochastic programming,” *Computers and Operations Research*, pp. 2656–2671, 2008.
- [86] C.C. Carøe and R. Schultz (1999), “Dual decomposition in stochastic integer programming,” *Operations Research Letters*, pp. 37–45.
- [87] F.V. Louveaux(1980), “A solution method for multistage stochastic programs with recourse with application to an energy investment problem,” *Operations Research*, pp. 889–902.
- [88] D.P. Morton, (1996), “An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling,” *Annals of Operations Research*, pp. 211–235.
- [89] M. V. F. Pereira and L. M. V. G. Pinto, “Multi-stage stochastic optimization applied to energy planning,” *Mathematical Programming*, pp. 359–375.
- [90] S. Takriti, J.R. Birge, and E. Long, “A stochastic model of the unit commitment problem,” *IEEE Transactions on Power Systems*, pp. 1497–1508.
- [91] N. Gröwe-Kuska, K. C. Kiwiel, M. P. Nowak, W. Römischi, and I. Wegner, “Power management in a hydro-thermal system under uncertainty by Lagrangian relaxation,” in *Greengard and Ruszczyński*, pp. 39–70.
- [92] N. Gröwe-Kuska and W. Römischi, “Stochastic unit commitment in hydrothermal power production planning,” in S. W. Wallace and W. T. Ziemba, editors, *Applications of Stochastic Programming*, MPS/SIAM Series on Optimization, pp. 633–653. SIAM, Philadelphia, USA, 2005.
- [93] S. W. Wallace and S.-E. Fleten. *Stochastic programming models in energy*, in *Ruszczynski and Shapiro* [36], chapter 10, pp 637–677.
- [94] S.-E. Fleten, S. W. Wallace and W. T. Ziemba, “Hedging electricity portfolios via stochastic programming,” in *Greengard and Ruszczyński*, pp. 71–93.
- [95] V.S. Pappala, S.N. Singh, and I. Erlich, “Unit Commitment under Wind Power and Demand Uncertainties,” *IEEE Power India Conference*, New Dehli, India, October 2008.
- [96] V.S. Pappala, and I. Erlich, “Stochastic Optimization of Hybrid PV/Fuel cell system with focus on Uncertainty Modeling,” *16th Power Systems Computation Conference*, Glasgow, Scotland, July 2008.
- [97] M. Víctor, Albornoz, Pablo Benario and Manuel E. Rojas, “A two-stage stochastic integer programming model for a thermal power system expansion,” *International Transactions in Operational Research*, Vol. 11 Issue 3, pp. 243–257, May 2004.
- [98] Misak Avetisyan, David Bayless, and Tigran Gnuni, “Optimal expansion of a developing power system under the conditions of market economy and environmental constraints,” *Energy Economics*, Vol. 28, Issue 4, pp. 455-466, July 2006.

- [99] Juan Álvarez López, Kumaraswamy Ponnambalam and Víctor H. Quintana, "Generation and Transmission Expansion Under Risk Using Stochastic Programming," IEEE Trans. on Power Systems, Vol. 22, No. 3, August 2007.
- [100] S.-E. Fleten and T. K. Kristoffersen, "Short-term hydropower production planning by stochastic programming," Computers and Operations Research, pp. 2656–2671, 2008.
- [101] S. Sen, L. Yu and T. Genc, "A stochastic programming approach to power portfolio optimization," Operations Research, pp. 55–72, 2006.
- [102] S. Takriti, B. Krasenbrink, and L. S.-Y. Wu, "Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem," Operations Research, pp. 268–280, 2000.
- [103] John M. Mulvey and Hafize Gaye Erkan, "Decentralized risk management for global property and casualty insurance companies," chapter 25, *Applications of Stochastic Programming*, W. Ziemba and S. Wallace (ed.), Springer-Verlag, 2003.
- [104] Bennett Golub, Martin Holmer, Raymond McKendall, Lawrence Pohlman, Stavros A. Zenios, "A stochastic programming model for money management," European Journal of Operational Research, Vol. 85, pp. 282–296, 1995.
- [105] Stavros A. Zenios, "Optimization models for structuring index funds," chapter 24, *Applications of Stochastic Programming*, W. Ziemba and S. Wallace (ed.), Springer-Verlag, 2003.
- [106] J.M. Mulvey and H. Vladimirov(1992), "Stochastic network programming for financial planning problems," Management Science, Vol. 38, pp. 1642–1664.
- [107] Pongsakdi, Arkadej, Pramoch Rangsunvigit, Kitipat Siemanond, and Miguel J. Bagajewicz, "Financial risk management in the planning of refinery operations," International Journal of Production Economics, Vol. 103, pp. 64.86, 2006.
- [108] Goel, Vikas and Grossmann, Ignacio E, "A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves," Computers & Chemical Engineering, Vol. 28 , pp.1409.1429, 2004.
- [109] Li, Wenkai, "Modeling Oil Refinery for Production Planning, Scheduling, and Economic Analysis," Hong Kong University of Science and Technology, PhD Thesis, 2004.
- [110] S. Hsieh and C. C. Chiang, "Manufacturing-to-sale planning model for fuel oil production," International Journal of Advanced Manufacturing Technology, Vol. 18, pp. 303–311, 2001.
- [111] J.M. Guldmann and F. Wang, "Optimizing the natural gas supply mix of local distribution utilities," European Journal of Operational Research, Vol. 112, pp. 598–612, 1999.
- [112] Bok, Jin-Kwang, Heeman Lee, and Sunwon Park, "Robust investment model for langrange capacity expansion of chemical processing networks under uncertain demand forecast scenarios," Computers & Chemical Engineering, Vol. 22, pp. 1037–1049, 1998.
- [113] Liu, Ming Long and Nikolaos Vasili Sahinidis, "Optimization in process planning under uncertainty," Industrial & Engineering Chemistry Research, Vol. 35, pp. 41–54, 1996.
- [114] A.E. Bopp, V. R. Kannan, S. W. Palocsay and S. P. Stevens, "An optimization model for planning natural gas purchases, transportation, storage, and deliverability," Omega: International Journal of Management Science 24, pp. 511–522, 1996.
- [115] J.R. Birge and C.H. Rosa(1996), "Incorporating investment uncertainty into greenhouse policy models," The Energy Journal, Vol. 17, pp. 79–90.
- [116] V.I. Norkin, Y.M. Ermoliev and A. Ruszczyński(1998), "On optimal allocation of indivisibles under uncertainty," Operations Research, Vol. 46, pp. 381–395.
- [117] A.K. Shukla and S.N. Gupta(1989), "A stochastic linear programming approach for crop planning," Acta Ciencia Indica Mathematics, Vol. 15, pp. 265–270.

- [118] Boychuk, Dennis; Martell, L. David, "A Multistage Stochastic Programming Model for Sustainable Forest-Level Timber Supply Under Risk of Fire," *Forest Science*, Vol. 42, No. 1, pp. 10-26, February 1996.
- [119] A. Martel and W. Price, "Stochastic Programming Applied to Human Resource Planning," *Journal of the Operational Research Society*, Vol. 32, pp. 187-196, 1981.
- [120] A.B. Philpott, "Stochastic optimization in yacht racing, Applications of Stochastic Programming," W. Ziemba and S. Wallace (ed.), Springer-Verlag, 2003.
- [121] T. Helgason, and S.W. Wallace(1991), "Approximate scenario solutions in the progressive hedging algorithm. A numerical study with an application to fisheries management," *annals of Operations Research*, Vol. 31, pp. 425-444.
- [122] M. Laguna(1998), "Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty," *Management Science*, Vol. 44, pp. S101-S110.
- [123] J. Dupacová, A. Gaivoronski, Z. Kos and T. Szantai(1991), "Stochastic programming in water management: A case study and a comparison of solution techniques," *European Journal of Operational Research*, Vol. 52, pp. 28-44.
- [124] Xian Liu, "The role of stochastic programming in communication network design," *Computers and Operations Research*, Vol. 32 , Issue 9, pp. 2329 – 2349, September 2005,.
- [125] Gaivoronski, A. Alexei, Stella, Fabio, "On-line portfolio selection using stochastic programming," *Journal of Economic Dynamics & Control*, ISSN: 0165-1889, 2003.
- [126] Alan J. King, László Somlyódy and Roger J.-B.Wets; "Stochastic Optimization for Lake Eutrophication Management," chapter 19, *Applications of Stochastic Programming*, edited by Stein W. Wallace and William T. Ziemba.
- [127] Dennis Boychuk and David L. Martell , "A Multistage Stochastic Programming Model for Sustainable Forest-Level Timber Supply Under Risk of Fire," *Forestscience*, Vol 2, 1996.
- [128] E. A. DeMeo, W Grant, MR. Milligan and M.J. Schuerger, "Wind plant integration: Cost, Status and Issues, A Multistage Stochastic Programming Model for Sustainable Forest-Level Timber Supply Under Risk of Fire," *IEEE Power & Energy Magazine*, Vol. 3, No. 6, pp. 38-46, 2005.
- [129] C. Chompoo-inwai, W.J. Lee, P Fuangfoo, M. Williams and J.R. Liao, "System Impact Study for the interconnection of Wind Generation and Utility System," *IEEE Trans on Industry Application*, Vol. 41, No. 1, pp. 163-168, 2005.
- [130] E.on Netz, Grid Code, High and extra high voltage, April 1, 2006 <http://www.eon-netz.com/>
- [131] Chung-Lung Chen, "Optimal wind-thermal generating unit commitment," *IEEE Trans. Energy Conversions*, Vol. 23, No. 1, March 2008.
- [132] G.C. Contaxis, J. Kabouris, "Short term scheduling in a wind/diesel autonomous energy system," *IEEE Trans. Power Systems*, Vol. 6, pp. 1161-1167, August 1991.
- [133] K. Methaprayoon, W.J. Lee, C. Yingvivanapong, J. Liao, "An integration of ANN wind power estimation into UC considering the forecasting uncertainty," *IEEE Trans. Industry Applications*, Vol. 43, No. 6, pp. 116-124, November 2007.
- [134] K. Rohrig, "Online-monitoring and prediction of wind power in german transmission system operation centres," in *World Wind Energy Conference*, Cape Town, South Africa, 2003.
- [135] B. Lange, K. Rohrig, B. Ernst, F. Schlögl, R. Jursa, J. Moradi, "Wind power forecasting in Germany - Recent advances and future challenges," in *Zeitschrift für Energiewirtschaft*, Vol. 30, Issue 2, pp. 115-120, 2006.
- [136] R. Jursa, B. Lange, K. Rohrig, "Advanced wind power prediction with artificial intelligence methods," in *first International ICSC Symposium on Artificial Intelligence in Energy Systems and Power*, Island of Madeira, Portugal, February 2006.

- [137] K. Rohrig, B. Lange, "Application of wind power prediction tools for power system operations," presented at the IEEE Power Engineering Society General Meeting, Montreal, Canada, 2006.
- [138] F. Bouffard, F.D. Galiana, "Stochastic security for operations planning with significant wind power generation," IEEE Trans. Power Systems, Vol. 23, No. 2, pp. 306–316, May 2008.
- [139] Samer Takriti, John R. Birge and Erik Long, "A Stochastic Model for the Unit Commitment Problem," IEEE Trans. Power Systems, Vol. 11, No. 3, pp. 1497–1506, August 1996.
- [140] N Gröwe-Kuska, M.P. Nowak, I. Wegner, "Modeling of uncertainty for the real-time management of power systems," preprint series, Institut für Mathematik, Humboldt-Universität, ISSN S863-0976, Berlin, 2001.
- [141] M. Lange: Analysis of the Uncertainty of Wind Power Predictions, Dissertation, Oldenburg, 2003.
- [142] Tsung-Ying Lee, "Optimal spinning reserve for a wind-thermal power system using EIPSO," IEEE Trans. On Power Systems, Vol.22, No.4, pp.1612–1621, November 2007.
- [143] <http://www.mathematik.hu-berlin.de/~nicole/scenred/gamsscenred.html>.
- [144] Miguel Carrión, 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.
- [145] Kall, Peter and Janos Mayer. Stochastic Linear Programming: Model, Theory, and Computation, Springer, New York, 2005.
- [146] R. Hemmecke, R. Schultz, "Decomposition methods for two-stage stochastic integer programs," in: M. Grötschel, S.O. Krumke, J. Rambau (Eds.), *Online Optimization of Large Scale Systems*, Springer-Verlag Berlin, pp. 601 – 622, 2001.
- [147] Ahmed M. Azmy, B.P. Wachholz and I.Erlich, "Management of PEM Fuel Cells for Residential Applications using Genetic Algorithms," The ninth International Middle-east Power systems conference(MEPCON), pp:16-18, Egypt, December 2003.
- [148] A. Capasso, W. Grattieri, R. Lamedica and A. Prudenzi (1994), "A Bottom-Up Approach to Residential Load Modeling," IEEE Transactions on Power Systems, Vol. 9, No. 2, pp. 957-964.

## Resume

### Personal Data:

Name	Venkata Swaroop Pappala
Date of Birth	27-07-1981
Nationality	Indian

### Studies:

2005-2009	Research Assistant Institute of Electrical Power Systems University of Duisburg-Essen, Duisburg
2003-2005	Graduate Student (best student award) Institute of Electrical Power Systems University of Duisburg-Essen, Duisburg
1998-2002	Undergraduate Student Dept. Of Electrical & Electronics Engineering Nagarjuna University, India
1996-1998	Intermediate Education Indian School Certificate Timpany School, India
1986-1996	Primary and Secondary School Indian Council of Secondary Education St. Anns School, India



## List of Publications

- [1] V.S. Pappala and I. Erlich, "A Variable-Dimension Optimization Approach to Unit Commitment Problem," accepted for publication in IEEE Transactions on Power Systems.
- [2] V.S. Pappala, W. Nakawiro, W. & I. Erlich, "Predictive Optimal Control of Wind Farm Reactive Sources," IEEE 2010 Transmission and Distribution Conference and Exposition, May 2010.
- [3] V.S. Pappala, I. Erlich, K. Rohrig and J. Dobschinski, "A Stochastic Model for the Optimal Operation of a Wind-Thermal Power System," IEEE Transactions on Power Systems, Vol. 24, no. 2, pp. 940-950.
- [4] V.S. Pappala, M. Wilch, S.N. Singh & I. Erlich, "Reactive Power Management in Offshore Wind Farms by Adaptive PSO," International Journal of Engineering Intelligent Systems for electrical Engineering & Communications, Vol. 15 No.2, June 2007 presented at 14th International Conference on Intelligent System Applications to Power Systems, Kaohsiung, Taiwan, Nov. 2007, pp. 63-70. (Best paper award).
- [5] V.S. Pappala and I. Erlich, "Power system optimization under uncertainties: A PSO approach," IEEE Swarm Intelligence symposium, St. Louis, Missouri, USA, Sep 21-23, 2008, pp.1-8.
- [6] V.S. Pappala and I. Erlich, "Unit Commitment under Wind Power and Demand Uncertainties," IEEE POWERCON and Power India Conference, New Delhi, India, Oct. 2008, pp. 1-6.
- [7] V.S. Pappala and I. Erlich, "A new approach for solving the unit commitment problem by adaptive particle swarm optimization," IEEE PES General Meeting, Pittsburgh, Pennsylvania, USA, July 2008, pp. 1-6.
- [8] V.S. Pappala and I. Erlich, "Optimal Operation of hybrid PV/Fuel cell system with focus on uncertainty modeling," 16th power systems computational conference, Glasgow, Scotland, July 2008.
- [9] V.S. Pappala and I. Erlich, "Consideration of wind power and demand uncertainties in unit commitment problem using PSO," IFAC world congress, Seoul, South Korea, July 2008.
- [10] V.S. Pappala and I. Erlich, "Management of Distributed Generation Units under Stochastic Load Demands using Particle Swarm Optimization," IEEE Power Engineering Society General Meeting, June 2007 Tampa, Florida USA, pp. 1-7.
- [11] V.S. Pappala and I. Erlich, "Uncertainty Modeling for the Management of Distributed Generation Units using PSO," IEEE PowerTech, Lausanne, Switzerland, July 2007, pp. 497-503.
- [12] M. Wilch, V.S. Pappala, S.N. Singh and I. Erlich, "Reactive Power Generation by DFIG Based Wind Farms with AC Grid Connection," IEEE PowerTech, Lausanne, Switzerland, July 2007, pp. 626-632.
- [13] V.S. Pappala and I. Erlich, "Particle Swarm Optimization," Dresdner Kreis 2006, Magdeburg, Germany, ISBN 3-929757-85-0.