# DEVELOPMENT OF A TOOL FOR PROFIT BASED UNIT COMMITMENT IN DEREGULATED ELECTRICITY MARKETS USING A HYBRID LAGRANGIAN RELAXATION – EVOLUTIONARY PARTICLE SWARM OPTIMIZATION APPROACH

**ADLINE KERUBO BIKERI**

**MASTER OF SCIENCE**

**(Electrical and Electronic Engineering)**

**JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**

**2019**

# Development of a Tool for Profit Based Unit Commitment in Deregulated Electricity Markets Using a Hybrid Lagrangian Relaxation – Evolutionary Particle Swarm Optimization Approach

**Adline Kerubo Bikeri**

**A thesis submitted in partial fulfillment for the degree of Master of Science in Electrical Engineering in the Jomo Kenyatta University of Agriculture and Technology**

**2019**

# DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signed................................................ Date..............................................

**Adline Kerubo Bikeri**

This thesis has been submitted for examination with our approval as University Supervisors.

Signed................................................ Date..............................................

**Dr. Peter K. Kihato**

**JKUAT, Kenya**

Signed................................................ Date..............................................

**Prof. Christopher M. Muriithi**

**Murang'a University of Technology, Kenya**

## ACKNOWLEDGMENT

I would first like thank my supervisors Prof. C. M. Maina and Dr P. K. Kihato for their consistent support and consideration while carrying out this research. Without their guidance and help, this research would not have materialized. They allowed this thesis to be my work and always steered me in the right direction with constructive comments and encouragement.

I am also grateful for the assistance provided by the JKUAT department of Electrical and Electronics Engineering and of the former Chair of department Prof. S. Kamau.

I would like to acknowledge with thanks my former employer Kenya Power Company and especially my former boss Eng. Ezra Ndenderu for offering support and understanding through this period.

Finally, I must express the deepest appreciation to my parents, to my husband and children for their love, unfailing support and continuous encouragement throughout my Masters study and through the process of researching and writing this thesis. This would not have been possible without them. Thank you.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF APPENDICES

## ABBREVIATIONS AND ACRONYMS

**ABC** Artificial Bee Colony

**ACO** Ant Colony Optimization

**AIS** Artificial Immune System

**B&B** Branch and Bound

**CPSO** Chaotic Particle Swarm Optimization

**DISCO** Distribution Company

**DP** Dynamic Programming

**DPSO** Dispersed Particle Swarm Optimization

**EP** Evolutionary Programming

**EPSO** Evolutionary Particle Swarm Optimization

**GA** Genetic Algorithms

**GENCO** Generation Company

**ICA** Imperialistic Competitive Algorithm

**IPP** Independent Power Producer

**ISO** Independent System Operator

**KENGEN** Kenya Generation Company

**LM** Lagrange Multiplier

**LMP** Locational Marginal Price

**LR** Lagrangian Relaxation

**MCP** Market Clearing Price

**MIP** Mixed Integer Programming

**NPSO** New Particle Swarm Optimization

**OBS** Optimal Bidding Strategy

**PBUC** Profit Based Unit Commitment

**PL** Priority Listing

**Poolco** Pool Company

**PPSO** Parallel Particle Swarm Optimization

**PSO** Particle Swarm Optimization

**PX** Power Exchange

**QPSO** Quantum-inspired Particle Swarm Optimization (QPSO)

**SFLA** Shuffled Frog Leaping Algorithm

**TRANSCO** Transmission Company

**TS** Tabu Search

**UC** Unit Commitment

## ABSTRACT

As electricity markets undergo deregulation all over the world, the approach to generation scheduling or unit commitment (UC) changes significantly. In traditional electricity markets with electricity utilities which act as system operators and also own generation units, UC is done based on a cost minimization objective. However, in deregulated markets, individual generation companies (GENCOs) have to carry out UC independently based on forecasts of energy and reserve prices for the scheduling period. The Generation Company (GENCO)'s UC strategies are developed with the aim of maximizing expected profit in what is known as Profit Based Unit Commitment (PBUC). Such profits are not only dependent on revenues from sale of energy and ancillary services such as reserve, but also on the cost characteristics of the generating units owned by the GENCO. This research develops a tool for carrying out PBUC for GENCOs in deregulated electricity markets. The tool is presented as a collection of MATLAB m-files that can be easily applied to any test system with the data stored in a specified format in an excel file. The MATLAB code is an implementation of solution algorithms that are developed and tested using simulations carried out for typical test systems. First, a solution methodology for the PBUC problem using a hybrid of the Lagrangian Relaxation (LR) and Particle Swarm Optimization (PSO) algorithms is implemented in MATLAB software. The PSO algorithm is used to update the Lagrange multipliers resulting in an optimal solution. It is found that the final solution is dependent on the values of the PSO algorithm parameters that have to be specified before running the algorithm. An analysis of the solution quality for various PSO algorithm parameters is carried out to determine the parameters that give the best solution. The algorithm is tested for a GENCO with 54 thermal units adapted from the standard IEEE 118-bus test system. To tackle the challenge of the solution quality being dependent on the algorithm parameters, the Evolutionary Particle Swarm Optimization (EPSO) algorithm is explored. EPSO is chosen based on previous research which showed that it generally results in better solutions than PSO because of a "self-tuning" characteristic of the parameters. Simulation results for a test GENCO show that the EPSO algorithm provides better solutions and has better convergence characteristics than the classic PSO algorithm. A second important consideration in the solution of the PBUC problem is the GENCO's market power i.e. it's influence on the market prices and/or demand. While a GENCO's bilateral demand is previously agreed on and therefore well known, allocations from the spot energy market depend largely on the GENCO's bidding strategy which is dependent on the GENCO's market power. A GENCO thus requires an optimal bidding strategy (OBS) which when combined with a PBUC approach would maximize its profits. A solution of the combined OBS-PBUC problem is therefore developed. Simulation results carried out for a test power system with GENCOs of differing market strengths show that the OBS depends largely on a GENCO's market power. Larger GENCOs with significant market power would typically bid higher to raise prices, while smaller GENCOs would typically bid lower to capture a larger portion of the spot market demand.

# CHAPTER ONE
# INTRODUCTION

## 1.1 Background

Over the last two-to-three decades, all over the world, the electric energy sub-sector has been undergoing significant changes that have necessitated a re-look into the various operation procedures. Probably the biggest change has been deregulation of many power systems, especially in the developed world; though aspects of deregulation are also beginning to take root in developing nations. Deregulation is the unbundling of vertically integrated power systems into GENCOs, Transmission Companies (TRANSCOs) and Distribution Companies (DISCOs) [2].

In Kenya, the electric energy sub-sector has been deregulated to a level where generation has been liberalized with a number of licensed generation companies known as Independent Power Producers (IPPs). These IPPs complement the main generation company, (Kenya Generation Company (KENGEN)), and control about 10% of the market share. Though the ideal market environment has not been set up, there is significant movement in this direction. There is also a single entity managing the transmission system and the traditional utility, Kenya Power Company manages the distribution and supply of electrical energy.

The main aim of deregulation is to create competition among GENCOs and hence provide different choices of generation options at lower prices to consumers [3, 4]. Other reasons driving deregulation include the positive experience of privatization in other industries, expected drop in electricity prices, improvement in customer focus, encouragement of innovation in the electricity sector among others. With deregulation, one of the main differences is the approach to what is traditionally known as the Unit Commitment (UC) problem.

Traditionally, solution of the UC problem can be defined as the problem of determining the turn-on and turn-off schedule of a set of electrical power generating units to give the minimum operating costs. This is done to meet the load demand while satisfying a set of operational constraints. The production costs include fuel, start-up, shut-down, and no-load costs. Constraints include capacity reserve, minimum generator up/down time, maximum power flow in the transmission lines and technical operating limits etc. The UC problem is a mixed combinatorial and continuous optimization problem, which is very complex to solve because of its enormous dimensions, a non-linear objective function, and a large number of constraints.

As electricity markets undergo deregulation all over the world, the approach to the unit commitment problem changes significantly. In the deregulated environment, generation units are not owned by a single company. There are a number of GENCOs who bid for a share of the market through an Independent System Operator (ISO). The ISO has the role of matching the load to the generation at all times. This is done simply by picking the generation with the least price successively until the allocated generation equals the load demand at a given time. The ISO relies on the bid prices to determine which units to use at what time. In such an environment, individual GENCOs carry out independent unit commitment. They determine schedules for their generators based on forecasts of energy prices at different times. Being privately owned entities, the main aim of these GENCOs is the maximization of their profits in the competitive environment. In this sense, the UC problem has been coined slightly differently in deregulated markets as the Profit Based Unit Commitment (PBUC) problem [3, 5].

Unit commitment decisions in deregulated markets are significantly different from regulated environments and it is important to determine the best approach.

Firstly, in the case of the traditional utility, UC decisions are *cost based* requiring the full cost-power characteristics of the various generation units. However, in PBUC, commitment decisions are *price based* for GENCOs. All that a GENCO needs is a forecast of the electricity price at a given time to decide whether or not to turn on a generator and how much output to schedule. Note that depending on bid decisions from competitors, a GENCO's schedule may be interfered with. Secondly, in the traditional unit commitment environment, spinning reserve is usually treated as an operational constraint. However, GENCOs in deregulated environments need not worry about reserve. This is the ISO's concern and in fact the GENCO would be compensated for providing spinning reserve. This then becomes a source of revenue. Therefore, in PBUC, spinning reserve is modeled as an income to the GENCO than as an operational constraint.

Numerous methodologies for solving the PBUC problem have been proposed in literature. These methodologies can be classified as classical methods and non-classical methods. Classical methods include Priority Listing (PL), Dynamic Programming (DP), Branch and Bound (B&B), Mixed Integer Programming (MIP), and Lagrangian Relaxation (Lagrangian Relaxation (LR)) [5, 6]. Non-classical methods include Genetic Algorithms (GA), Particle Swarm Optimization (Particle Swarm Optimization (PSO)), Artificial Bee Colony (ABC), Muller method among others [7, 8]. There have also been proposals for hybridization of some of these methods taking advantage of the strengths of two or more methods to provide a more effective solution algorithm [9–11]. A comprehensive review of these methods can be found in [12–14]

A third difference is that the size of the GENCO in the market place will significantly affect the adopted unit commitment strategy. An important determinant of a GENCO's bidding strategy in the spot market is its market power i.e. the GENCO's ability to alter the market price and allocations in the market. A GENCO whose actions cannot affect the market equilibrium is referred to a *price taker* and conversely, a GENCO whose actions significantly affect the market is referred to as a *price maker*. Electricity markets usually assume an oligopolistic structure characterized by several price takers and one or two price makers; usually large companies that are offshoots of the previous regional or national utilities prior to deregulation [4,15]. Since the GENCO's market power can influence the market price, it is a significant consideration in the determination of an optimal bidding strategy (Optimal Bidding Strategy (OBS)) and hence the solution of the PBUC problem.

## 1.2 Problem Statement

As introduced in section 1.1, UC in a deregulated environment is fundamentally different from UC in a regulated environment. Each GENCO proposes their generation schedule based on expected energy and reserve prices in a bid to maximize their profit. This is definitely a significant operational problem for GENCOs since it may mean large differences in company profits or losses. In the recent past, there has been extensive research on the topic of the PBUC problem with most papers focusing on the solution methodology as this is the main challenge for what can be described as a very complex mathematical optimization problem.

This research focuses on a tool for the solution of the PBUC problem that combines traditional optimization techniques with heuristic methods. The main problems studied in this research include:

- How to formulate the optimization problem to capture the deregulated market characteristics including reserve as an income to the GENCO and incorporating bilateral markets agreements.

- How to incorporate heuristic algorithms (specifically PSO and Evolutionary Particle Swarm Optimization (EPSO)) in developing more effective PBUC solution methodologies.
- How does UC decisions change depending on the size of the GENCO.

### 1.3 Justification

Electricity market deregulation is a relatively new concept. While it has been in place in various forms in different countries and to differing degrees, there still remains a lot of unanswered questions and lots of room for improvement. This can only be achieved through scientific research. More so, different regions and countries take different approaches based on their market needs. For these reasons, researchers have to learn from past experiences and also predict possible future trends and areas that will require improvements. Adequate tools for operation in deregulated markets should be developed and tested.

As for the GENCOs operating in such an environment, it is very important to be competitive and the business must be viable. The dynamic electrical market will significantly alter the practices of the traditional electric utilities as they no longer have a monopoly and cannot unilaterally set prices so as to cover their operational costs. For these companies to maintain profitability, they'll have to find ways to be more efficient both technically and financially. The newer IPPs also have to develop business strategies that will enable them capture significant portions of the market share so as to recover the huge capital costs usually associated with putting up new power plants and also be profitable in the long run. The determination of the best Unit Commitment strategy to maximize profits is therefore very important and a software tool for this is also imperative.

In the past decade or so, a number of academic papers have been written on the solution of the PBUC problem [7, 9, 10, 16–27] which shows the need for these tools. However, in most cases, the proposals still remain at the conceptual stage. This research advances some of these ideas by improving the mathematical representation of the problem and giving a more effective method for solving what is a complex mathematical problem.

Currently, even in the more advanced electric markets, many GENCOs still operate using strategies similar to the traditional utilities. This is because most electric markets still operate with only one or two large companies and a few IPPs who usually have some form of government rebates to encourage their participation in the electric market with the long term goal of having a truly competitive market. GENCOs therefore usually still approach the UC problem in a similar way to the traditional unit commitment with the objective of minimizing operational costs. The solution methodologies are still largely based on classical mathematical optimization techniques such as dynamic programming or Lagrangian relaxation.

## 1.4 Objectives

### 1.4.1 Main Objective

To develop a tool for the optimal solution of the Profit Based Unit Commitment (PBUC) problem in a deregulated environment by the use of a hybrid optimization approach and incorporating market characteristics including expected energy prices, reserve payments and GENCO size.

### 1.4.2 Specific Objectives

1. Mathematical formulation of the PBUC problem in a deregulated electricity market incorporating expected energy and reserve prices.

2. Developing a solution algorithm and an accompanying software tool for the solution of the PBUC problem using a hybrid of the LR and PSO methods.
3. Analysis of the effects of GENCO market power in the solution of the PBUC problem whereby *market power* refers to the GENCO's installed capacity.

## 1.5 Scope

This research covered the following:

- The development of an algorithm for the solution of the PBUC problem using a hybrid LR-PSO method. The methodology is implemented in MATLAB software. The solution quality is found to be dependent on the PSO parameters. Hence, a selection of the best PSO parameters is carried out.
- In order to overcome the challenge of parameter tuning with the PSO algorithm, the a hybrid LR-EPSO method is then developed. The LR-EPSO method is compared to the LR-PSO methods in terms of solution quality and convergence characteristics.
- Investigation of the effect of GENCO size in the solution of the PBUC problem. A methodology for determining optimal bidding strategies depending on individual GENCO market power is developed.

## 1.6 Publications

1. **Bikeri, A.K.**, Maina, C.M. and Kihato, P.K., "GENCO Optimal Bidding Strategy and Profit Based Unit Commitment using Evolutionary Particle Swarm Optimization illustrating the effect of GENCO Market Power," *International Journal of Electrical and Computer Engineering (IJECE)*, (*in press*).
2. **Bikeri, A.K.**, Maina, C.M. and Kihato, P.K., "Profit Based Unit Commitment Using Evolutionary Particle Swarm Optimization," *in proceedings*

*of the 2017 IEEE Africon Conference*, Cape Town, South Africa, 18-20 September 2017, pp. 1180–1185.

3. **Bikeri, A.K.**, Maina, C.M. and Kihato, P.K., "Profit Based Unit Commitment in Deregulated Electricity Markets Using A Hybrid Lagrangian Relaxation - Particle Swarm Optimization Approach," *in proceedings of the 2017 Sustainable Research and Innovation (SRI) Conference,* Nairobi, Kenya, 3-5 May 2017, pp. 1–6.
4. **Bikeri, A.K.**, Maina, C.M and Kihato, P.K., "A Review of Unit Commitment in Deregulated Electricity Markets," *in proceedings of the 2015 Sustainable Research and Innovation (SRI) Conference,* Nairobi, Kenya, 6-8 May 2015, pp. 9–13.

## 1.7 Thesis organization

This thesis is organized as follows:

*Chapter 1*

This chapter presents an overview of the thesis.

*Chapter 2*

This chapter reviews various literature related to the research concepts. These include:

- Deregulation in electric power systems
- Unit commitment in deregulated electricity markets
- GENCO bidding strategies
- Mathematical optimization methods including LR, DP, PSO, and EPSO.

*Chapter 3*

This chapter presents the methodologies used in the research:

- PBUC problem formulation and solution using LR-PSO and LR-EPSO methods

– Optimal bidding strategy and PBUC solution considering the GENCO market power.

*Chapter 4*

This chapter discusses the results obtained from the numerical simulations and describes the developed software tool.

*Chapter 5*

This chapter summarizes the conclusions of the research.

# CHAPTER TWO

# LITERATURE REVIEW

This chapter reviews various concepts regarding the solution of the PBUC problem in deregulated electricity markets. In section 2.1, the concept of deregulation in electric power systems is described. Here, the differences between regulated and deregulated electricity market structures are highlighted. In section 2.2, the approach to Unit Commitment in deregulated markets is explained. This includes formulation of the PBUC problem and the different solution methodologies including classical and non-classical methods. An important consideration in the solution of the PBUC problem is the GENCO bidding strategy that is dependent on GENCO market power. This is described in section 2.3 illustrated with a simple example. Finally, the main mathematical optimization concepts used in the research are described in section 2.4. These include: LR, DP, PSO, and EPSO.

## 2.1 Deregulation in Electric Power Systems

For decades, electric power systems operated under the vertically integrated model where all functions of the system - generation, transmission, and distribution - were under a single utility usually operated by a national or regional government (see Figure 2.1) [1]. However, from the mid 1980's, electricity markets all over the world have been undergoing restructuring or what is referred to as deregulation [1, 28].

Deregulation involves breaking up of the traditional monopolies into smaller entities each dealing with a section of the electric power supply chain i.e. either generation, transmission, or distribution [3]. It has also introduced competition allowing a number of GENCOs, TRANSCOs and DISCOs into the industry to compete with the traditional monopolies. Competition can be introduced at the wholesale level to give the wholesale competition model of Figure 2.2 or the retail competition level of Figure 2.3 [1].

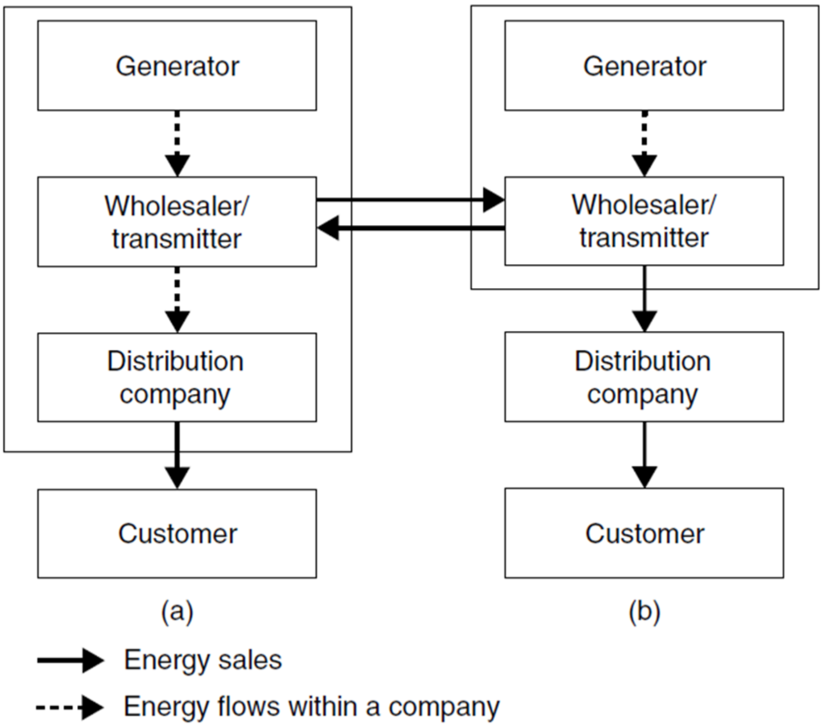



**Figure 2.1:** Vertically integrated model of electricity markets [1]

Deregulation mainly begun in the UK and US and also quickly took root in other western countries such as the Scandinavian countries, Western European countries and Australia. There are many reasons that led to deregulation of the electric power industry. One of the biggest forces was the change in generation economies of scale that occurred throughout the 1980's [4, 29]. Traditionally, electric utility systems evolved with the centralized generation concept because of significant economies of scale in power generation. However, during the 1980's/1990's, a change in the economies of scale was observed mainly because of the following reasons [29]:

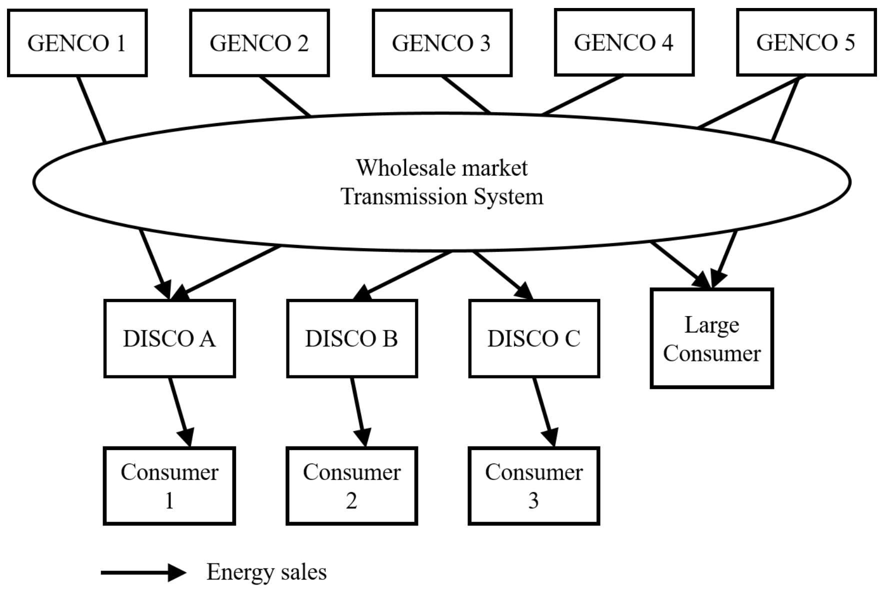



**Figure 2.2:** Wholesale competition model of electricity market [1]
.

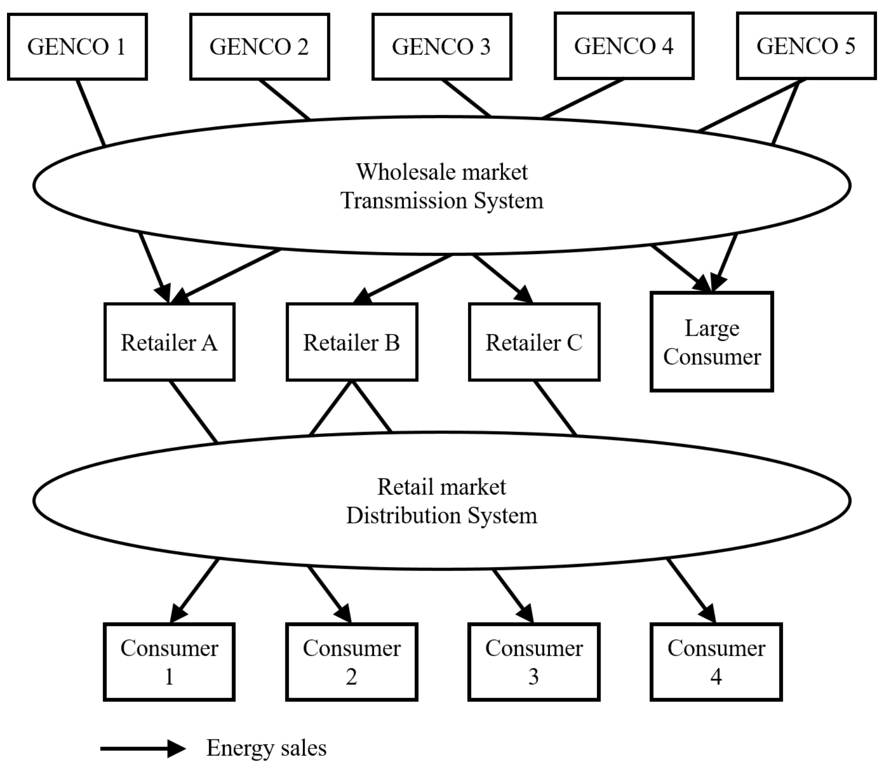



**Figure 2.3:** Retail competition model of electricity market [1]
.

1. Technological innovations improved the efficiency of small units for gas turbines, combined cycle, hydro and fuel cells over that of large ones.
2. Improvements in materials, including new high temperature metals, special lubricants, ceramics, and carbon fiber, permitted vastly stronger and less expensive small machinery to be built.
3. Computerized control systems were developed that often significantly reduced the number of required on-site personnel.
4. Data communications and off-site monitoring systems can control the units from remote operations centers, where one central operator can monitor a dozen units at various sites, as if present at each.

Thus in many instances, it was possible to build new power plants that could provide energy at a lower price than what customers were paying the existing old, giant power plants. It became possible for the industrial and commercial users of electricity to build and operate their own plants to produce power cheaper than that of utility and also sell the excess power to small customers.

Once the idea of deregulation took root, the following reasons were used to push for restructuring [4]:

1. *The need for regulation changed.* - More fundamental than any other reasons for change was the fact that the basic needs for regulation of electric industry had died away before the end of 20th century. First, the original need for regulation, which was to provide risk free finance to build the infrastructure, did not exist any more. Second, the omnipresent electric system created was already paid for. The revenues gained by the electric utilities was invested to renew their system and the level of risk in doing so was less as compared to that which existed in the initial era. Being a proven technology, the risk involved in investing money in such a technology was nullified. Electricity could be thought of an essential commodity, which can

be bought and sold in the marketplace in a competitive manner, just like other commodities.

2. *Privatization* - There is always the conviction that a private industry could do a better job of running the power industry than a government owned company. This belief came from better privatization experiences of the other industries. Deregulation does not necessarily have to be a part of privatization efforts but deregulation to free up the rules nearly always accompanies privatization.
3. *Expected drop in electricity costs* - Competition brings innovation, efficiency, and lower costs. The rate of cost decline is different in different areas and there are many reasons for this. However, the overall experience all over the world where deregulation has been implemented is that the electricity prices have declined.
4. *Improvement in customer focus* - Although monopoly utilities have an obligation to serve all customers, it does not promote the pro-active attention to customer needs. A monopoly utility listens to its customers when they explain their needs, and then responds. A competitive electric service company anticipates customer's needs and responds in advance. The technological advances that will be applied under deregulation, address customer service. One of the most important gains of competition in the electricity market is the increase in customer value.
5. *Encouraging innovation* - The regulatory process and the lack of competition gave electric utilities no incentive to improve on yesterday's performance or to take risks on new ideas that might increase customer value. If a new idea succeeded in cutting costs, the utility still made only its regulated rate of return on investment; if it didn't work, the utility would usually have to 'eat' a good deal of the failed attempt, as imprudent expenses. Fur-

thermore, why would a regulated utility want to use new ideas to lower its costs under a regulated rate of return framework? Under deregulated environment, the electric utility will always try to innovate something for the betterment of service and in turn save its costs and maximize the profit. By means of this, the utility will try to ensure that it will maintain its customer base in spite of competition.

Other forces supporting the main reasons for motivating deregulation include [4]:

- Overstaffing in the regulated electric industry
- Global economic crisis of the 1970's
- Political and ideological changes
- Managerial inefficiency in regulated companies
- Lack of public resources for the further development
- More demanding environment issues
- Pressure of financial institutions

As deregulation begun to take root, different countries took different approaches in their systems. Generally, two main models of deregulation appeared: - the *Poolco model* adopted primarily in the UK, and the *ISO model* adopted in California, US and in the Nordic pool [29]. Other countries like Australia, New Zealand and European Union countries are employing one of the two models with some changes to meet their specialized demands. Table 2.1 gives comparisons between the two market structures.

**Table 2.1:** Comparisons of the Open Access and Poolco electricity market structures

| **Open Access** | **Poolco** |
|---|---|
| • Bulk of the energy transactions are carried out as bilateral trades while there may also exist a day ahead spot market. | • All energy transactions are carried out through the pool, which may be organized through a day ahead trading mechanism. |
| • The ISO is responsible for market administration, generation scheduling or dispatch functions. | • The Poolco Operator is responsible for the market settlements, unit commitment and determination of pool price. |
| • Participation in the market by GENCOs is not mandatory. | • Participation by GENCOs is mandatory. |
| • The ISO is responsible for system security and control, procuring necessary ancillary services. | • The Poolco operator is responsible for system security and control, procuring necessary ancillary services. |

1. **Poolco model**

   In this model there is only one single buyer for all the energy generated by GENCOs [29]. The buyer here is a Pool Company (Poolco) which is a governmental or quasi-governmental agency that buys on behalf of all consumers, taking bids from all sellers and buying enough power to meet the total need, taking the lowest cost bidders. The Poolco operator also has responsibility for running the power system, and is thus a combined buyer-system operator.

   Therefore, the Poolco is responsible for inviting bids for energy and deciding the energy price for a particular period in the future markets like the day-ahead market. It is also responsible for real time operation of the system. This market works in a way quite similar to centralized unit commitment and economic dispatch. The difference is that in traditional unit commitment and economic dispatch the actual cost of the energy generations are considered but in deregulated environment, the GENCOs place price curves of each of its generators and the actual cost is hidden from general knowledge.

The Poolco being the system operator and auctioneer as well, takes care of network congestion at the auction level itself in a manner similar to the economic dispatch. Participation in the auction conducted by Poolco is a must for all GENCOs. As the Poolco is the only buyer, there are no bids from buyer's side; the auction is a single sided auction.

2. **Open access model**

   The models used in Nordic Pool and California are examples of this model [29]. The energy auction and future markets are conducted by an independent entity called a Power Exchange (PX) and the system is operated by another independent body called the ISO, who assures equal opportunities to all sellers and buyers through open access to the grid. The buyers and sellers have an option of entering in bilateral transactions or be participants in the energy auction conducted by the PX. The auction conducted by PX is double sided auction as sellers as well as buyers place the bids. The sellers and buyers are allowed to place a portfolio bid, i.e. a combined bid for many generators.

## 2.2 Unit Commitment in Deregulated Markets

The approach to Unit Commitment in the deregulated environment is significantly different from that in the regulated environment. Here, the GENCO is not the system operator. This means that, unlike the regulated market where the objective of the utility in unit commitment is the minimization of operating cost, in the deregulated environment, the objective of the GENCO is the maximization of profit. This has led to what is now referred to as PBUC in deregulated markets [3, 17, 18]. From the GENCOs point of view, an optimal solution to the PBUC problem is very important because of the potential economic consequences. Reducing the fuel cost by as little as 0.5 percent can result in savings of millions of dollars per year for large GENCOs which would translate to significant gains in profit [30]. Because of the potential economic benefits, one of the biggest needs for GENCOs in deregulated markets is an effective tool for making unit commitment decisions and developing bidding strategies [30].

### 2.2.1 PBUC Problem Formulation

The PBUC in deregulated power systems determines the generating unit schedules for maximizing the profit of GENCOs subject to operational constraints such as load demand, spinning reserve and ramp rate limits. Profit ($PF$) is defined as the difference between revenue ($RV$) obtained from sale of energy and in some cases reserve; minus the total operating cost ($TC$) of the GENCO. The objective function of the PBUC problem is then given as [6]:

$$\textit{Maximize} \qquad PF = RV - TC \tag{2.1}$$

GENCO revenue is from selling power to both the energy and reserve markets.

The objective function given in (2.1) is formulated subject to the constraints [6]:

1. *Power Balance Constraints:* In PBUC, power generation by a single GENCO may be less than the demand and reserve at a given time. This is because a number of GENCOs are available to serve the system load and a single generator may not be able to meet the load anyway. This is fundamentally different from the "generation equals demand" constraint of the traditional UC problem. The relaxed power balance constraint allows for more flexibility in the unit commitment schedule.
2. *Unit Generation Limits:* Generation units usually have operational maximum

and minimum power output limits within which the unit output must be maintained.

3. *Minimum Up/Down Time Constraints:* A thermal unit can only undergo gradual temperature changes. Hence, there is a minimum up-time once the unit is running and; for a de-committed unit, a minimum down time before it can be recommitted.
4. *Ramp Rate Limits:* The ramp rate limits confine the power output increase or decrease between adjacent hours for certain units.
5. *Crew Constraints:* To cater for limitations in the number of operational personnel, restrictions on the number of units to be turned ON at the same time may be included.

### 2.2.2 Solution Methodologies

Once the optimization problem has been formulated, a methodology for solving the highly nonlinear optimization problem is usually proposed. In fact, most researches in literature focus on the methodology for solving the PBUC problem. The UC problem is a mixed combinatorial and continuous optimization problem, which is very complex to solve because of its enormous dimension, a non-linear objective function, and a large number of constraints [31]. Solution methodologies for the traditional UC problem can be found in [30–36].

In the deregulated environment, an efficient solution methodology is key to the success of the operations of an individual GENCO. Numerous methodologies for solving the PBUC problem have been proposed in literature. These methodologies can be classified as classical methods and non-classical methods. Classical methods include Priority Listing (PL), Dynamic Programming (DP), Branch and Bound (B&B), Mixed Integer Programming (MIP), and Lagrangian Relaxation (LR) [5, 6, 37–39]. Non-classical methods include GA, Memetic Algorithm, Ant

Colony Optimization (ACO), PSO, ABC, Imperialistic Competitive Algorithm (ICA), Muller method among others [7, 16–22]. There have also been proposals for hybridization of some of these methods taking advantage of the strengths of two or more methods to provide a more effective solution algorithm [9, 10, 23–27].

#### 2.2.2.1 Classical Methods

Falling under the category of classical methods for solving the PBUC problem are PL, DP, B&B, MIP, and LR. In the PL method, plants are activated according to a pre-prepared list while schedules are adapted to respect technical restrictions such as minimum up and down times, and minimum and maximum operating points [38]. The PL method is a simple, almost rule of thumb method hence the solution is only an estimate of the actual optimal unit allocations [38]. In fact using PL almost never results in an optimal solution. Reference [20] gives an Improved Pre-prepared Power Demand (IPPD) table for solving the PBUC problem in a deregulated environment. The method, quite similar to the traditional PL method gives a solution that is closer to the optimal solution and in significantly less computation time.

DP is one of the earliest optimization based techniques to be applied to the UC problem and is still used extensively all over the world especially in regulated markets [34]. The DP technique employs a systematic searching algorithm that tries to achieve an optimal solution without having to access all possible combinations. Generating units are classified into related groups from which the optimal path is searched with a reduced number of possible combinations as a result of the classification. The method however suffers from the problem of huge computational time as the number of units being considered grows and hence for large systems with hundreds of units, DP as a solution algorithm for the PBUC problem becomes impractical. In [6], a DP approach is used to obtain a near-optimal unit commitment in a competitive power market. More significantly though, the problem formulation is incorporated into a Multi-area unit commitment with import/export and tie-line constraints. The method therefore illustrates the process of maximization of GENCO profit in a multi-area system.

Reference [37] uses the classical MILP to solve the PBUC problem. The main contribution of the paper is however not the solution methodology but rather a quantification of the sub-optimality of profit that can be expected in a PBUC when incorrect price forecasts are used. The results show how crucial an accurate price forecasting regime is for the realization of expected profits. In [5], the MIP and LR methods are compared and the authors note that though the MIP method produces more optimal results, the computation time and memory requirements would be a major obstacle when applying MIP to large UC problems.

LR is one of the most popular of the classical methods. The main advantage of the method is the speed with which the algorithm converges to a solution [5]. It has however been pointed out that the method suffers from often being stuck at local optima. This is because the quality of the solution strongly depends on the algorithm used to update the Lagrangian multipliers. For this reason, a number of the more recent papers combine LR with one or more of the non-classical methods so as to improve the quality of the solution [9, 27].

#### 2.2.2.2 Non-Classical Methods

Non-classical methods for solving the PBUC problem include GA, PSO, Muller Method, Shuffled Frog Leaping Algorithm (SFLA), ICA, ACO, ABC, Simulated Annealing (SA), Tabu Search (TS), among others. These heuristic algorithms have the advantage that they do not require derivative information to solve the optimization problem [18]. It is actually possible to encode the variables so that the optimization is done with the encoded variables with little attention to the systematic movement towards an optimal solution. The second major advantage is that these methods are better capable of searching through the entire solution space for the global optimal solution. Because of these advantages, they are more capable of dealing with the complex nonlinear constraints related to the PBUC problem and thus heuristic methods have received much more attention from researchers over the last few years.

Reference [16] uses genetic algorithms to solve the PBUC problem. The authors show the improved solution quality of the GA method compared to classical methods. The algorithm is tested with the two interconnected regions of the National Electricity Market in Australia and hence illustrates the practicality on an actual power market. An earlier example of the implementation of GA to solve the PBUC problem is given in [17].

In [18], an improved discrete binary PSO and a standard value PSO are used iteratively to solve the PBUC problem. The effectiveness of the solution methodology is illustrated for a GENCO with ten units in a competitive market. The PSO technique is also used to solve the GENCOs PBUC problem in a day ahead competitive electricity market in [7]. Apart from the traditional PSO technique, the authors also test three other PSO techniques: Chaotic PSO (CPSO), New PSO (NPSO) and Dispersed PSO (DPSO) and compare the results. Generation, spinning reserve, non-spinning reserve, and system constraints are considered in proposed formulation. To tackle the problem of long computation time that is usually associated with heuristic methods, [19] proposes a Parallel Particle Swarm Optimization (PPSO) solution to the PBUC problem. The method uses a cluster of computers performing parallel operations in a distributed environment and the results show the effectiveness of parallel computing in handling the huge dimensions of the PBUC problem. The authors report significant reductions in algorithm computation time. The time complexity and the solution quality with respect to the number of processors in the cluster are also investigated.

The Muller method is used for solving the PBUC problem in [20]. The methodology is implemented in two stages. Initially, the determination of units to be committed is obtained by a simple approach and then a non-linear programming sub problem of economic dispatch is solved by the Muller method. The biggest promise with this method is reduced computational time though the initial allocation generally results in a sub-optimal solution.

Reference [21], uses the Shuffled Frog Leaping Algorithm (SFLA) to solve the Profit Based Unit Commitment problem under deregulated environment with emission constraints. The bi-objective function optimization problem is formulated as a maximization of the Generation Companies profit and a minimization of the emission output of the thermal units.

A relatively newer heuristic method known as the Imperialistic Competitive Algorithm (ICA) is used in [22] to solve the PBUC problem in a competitive environment. The algorithm is presented as a tool to be used by GENCOs in making commitment decisions for maximum profit in the day-ahead energy market. The method is validated on a typical 10 generating unit system available in the literature.

The main challenge with heuristic methods is that the computation process is usually rather time consuming especially as the number of generating units increases. However, these algorithms can be easily implemented in high-speed parallel computing techniques with which the challenge of long computational time can be overcome [20], [25].

#### 2.2.2.3 Hybrid Methods

The classical, gradient based search algorithms tend to be faster in convergence but suffer from getting stuck in local optima. The non-classical heuristic methods on the other hand are better at searching through the solution space but are more time consuming. Because of these characteristics a number of researchers have proposed hybrid methods which combine two or more of the above techniques for better solution quality in terms of computation time and solution quality.

Reference [23] was one of the first papers on hybrid methods for the PBUC problem. The paper presented a hybrid model between LR and GA to solve the unit commitment problem with the GA being used to update the Lagrangian multipliers. Better results than those obtained from traditional unit commitment are reported. Reference [10] proposes a hybrid Artificial Immune System (AIS) based GA method to solve the PBUC problem. The authors report that the incorporation of the AIS into the GA algorithm results in increased diversity in the initial strings to ensure that the GA searches the entire problem space hence resulting in better solutions.

A hybrid model between LR and Quantum-inspired Particle Swarm Optimization (QPSO) is used to solve the PBUC problem in [26]. Constraints including load demand, spinning reserve, generation limits and minimum up and down time constraints are included and the method is tested on two different size systems. The authors report higher quality solutions compared to other methods in literature. A second example of the hybrid LR-PSO algorithm can be found in [27]. Again the authors highlight the improvement in solution quality by updating the Lagrangian multipliers using the PSO technique.

Reference [9] uses a hybrid LR - Evolutionary Programming (EP) model to solve the PBUC problem in a deregulated electricity market. Here, significantly, a consideration of the losses in the transmission system is included resulting in higher profits for the GENCO as they supply not only system load but also network losses. A hybrid LR-EP method is also used in [40] with an important consideration of both the power and reserve prices.

## 2.3 GENCO Bidding Strategies

In deregulated electricity markets, electric energy is sold either through bilateral agreements between GENCOs and consumers or through an electricity pool operated by an independent system operator i.e. the *electricity spot market* [3]. In the case of the bilateral market, the buyer and seller agree on a transaction price from which the GENCO meets all costs for transmission, distribution, and other ancillary services. The electricity pool is however operated by an independent system operator ISO who receives and aggregates hourly energy supply bids from GENCOs and hourly demand bids from consumers after which a *Market Clearing Price (MCP)* is determined [4]. The GENCOs are allocated portions of the demand based on a cheapest-bid first while ensuring system reliability and security. The MCP is defined as the cost of supplying the last MW of demand and all GENCOs who receive load allocations for the given hour are paid at this price irrespective of their bids.

Each GENCO will combine the bilateral demand with the allocation from the spot market as its *own demand* and from this data draw up a UC schedule based on a profit maximization objective. Since the spot market allocation is based largely on the GENCO's bid and those of its competitors, the GENCO bid decisions significantly affect its allocation and hence its profits. Should a GENCO have enough influence, it could affect the MCP and consequently its profits. The magnitude of this influence defines the GENCO's market power. Under perfect competition, so as to maximize its profits, a GENCO should bid at its marginal cost (cost of supplying an extra MW of electricity) [41]. However, depending on the market environment, the GENCO could increase its profits in one of two ways:

- The GENCO could lower its bid (*bid low*) thereby potentially increasing its allocation in the spot market though this could reduce the MCP. Bidding low is justified if the reduced revenue due to the lower prices is covered by the increased revenue due to a larger allocation.
- The GENCO could raise its bid (*bid high*) thereby potentially reducing its allocation in the spot market but increasing the MCP. This is justified if the increased revenue due to the higher prices cover the revenue lost due to a smaller allocation.

A minimal example to illustrate the spot market dynamics follows next.

The GENCO marginal cost curve forms the basis of its bidding strategy. The marginal cost curve is a plot of the incremental cost of power generation against the total power output for a GENCO. Mathematically, $MC_i$ – the marginal cost curve for GENCO $i$ is given by:

$$MC_i = \frac{\partial C_{Ti}}{\partial P_{Ti}}, \tag{2.2}$$

where $C_{Ti}$ is the total operating cost of GENCO $i$ when supplying a total of $P_{Ti}$ MW. Assuming a quadratic cost curve (as typically used in literature [7, 8]) for GENCO costs, $C_{Ti}$ is given by:

$$C_{Ti} = \sum_{j=1}^{N} \left( a_{ij} + b_{ij}P_{ij} + c_{ij}P_{ij}^2 \right), \tag{2.3}$$

and

$$P_{Ti} = \sum_{j=1}^{N} P_{ij}. \tag{2.4}$$

In (2.3) and (2.4) $a_{ij}$, $b_{ij}$, and $c_{ij}$ are the coefficients of the quadratic cost curves for unit $j$ operated by GENCO $i$ while $P_{ij}$ is the output of unit $j$ operated by GENCO $i$.

Consider two GENCOs each owning one generating unit with cost characteristics shown in Table 2.2. The marginal cost curves for the two GENCOs are plotted in Figure 2.4(a) showing that GENCO G1 has the cheaper generating unit of the two GENCOs. If each GENCO submits its marginal cost curve as its supply curve, the combined system supply curve will be as shown in Figure 2.4(b). Assuming a nominal system demand of 200 MW with a linear demand curve as shown in Figure 2.4(b), the market equilibrium will then be the point at which the two curves intersect. When read from Figure 2.4(b) this point is ($P_d = 200\,\text{MW}$, MCP = \$30.78/MWh). When extrapolated to the supply curves of the two GENCOs, G1 and G2 will supply 144.4 MW and 55.6 MW respectively.

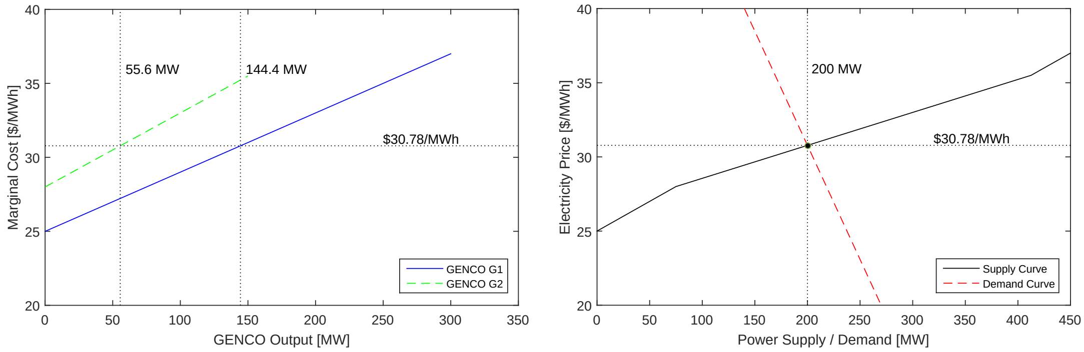
**Table 2.2:** Example GENCO cost characteristics

| GENCO | $P_{i1}^{\min}$ | $P_{i1}^{\max}$ | Cost Equation $C_{Ti}(P_{ij})$ |
|---|---|---|---|
| G1 | 0 | 300 | $25P_{11} + 0.020P_{11}^2$ |
| G2 | 0 | 150 | $28P_{21} + 0.025P_{21}^2$ |

Now, consider a case where GENCO G1 submits bids where the gradient of its marginal cost curve is multiplied by a factor $\mu_1$. Its bid curve, $BC_1$ is then given by:

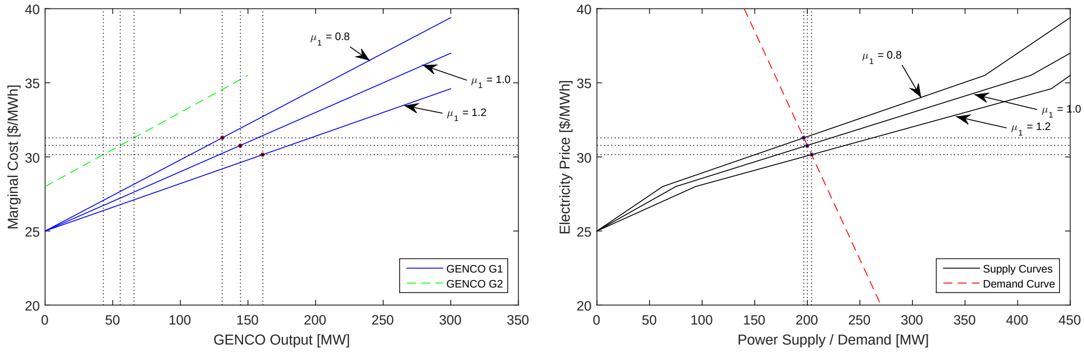
$$BC_1 = b_{11} + \mu_1 \cdot 2c_{11}P_{11} = 25 + 0.04\mu_1 P_{11} \tag{2.5}$$

A value of $\mu_1 > 1$ raises the bid curve above the nominal meaning that the GENCO *bids high* while a value of $\mu_1 < 1$ means that the GENCO *bids low.* The effect of $\mu_1$ on the MCP and the GENCO allocations is illustrated in Figure




(a) (b)

**Figure 2.4:** (a) Marginal cost curves for two GENCOs and (b) market equilibrium obtained from the intersection of the aggregated supply curve and the system demand curve.




(a) (b)

**Figure 2.5:** Illustration of the effect of GENCO G1's bid strategy on (a) the demand allocations and (b) the MCP.

2.5 for values of $\mu_1 = 0.8$, and $\mu_1 = 1.2$. The results are summarized in Table 2.3 showing that as $\mu_1$ increases, the MCP increases, GENCO G1's allocation reduces (as does its revenue and costs) but its profit increases.

A plot of the GENCO profit against the value of $\mu_i$ for the two GENCOs acting individually is illustrated in Figure 2.6 which shows that the two GENCOs achieve maximum profits at different values of $\mu_i$ ($\mu_1 = 1.9$ and $\mu_2 = 0.6$). These results show that the larger GENCO G1 should *bid high* to increase its profits while conversely, the smaller GENCO G2 should *bid low* to increase its profits.

**Table 2.3:** Effect of GENCO bidding strategy on spot market prices, allocations, revenues, costs, and profits

| $\mu_1$ | MCP [$/MWh] | $P_1$ [MW] | Revenue [$/h] | Cost [$/h] | Profit [$/h] |
|---|---|---|---|---|---|
| 0.8 | 30.15 | 161.01 | 4,854.98 | 4,543.87 | 311.11 |
| 1.0 | 30.78 | 144.44 | 4,445.68 | 4,028.40 | 417.28 |
| 1.2 | 31.29 | 130.97 | 4,097.48 | 3,617.21 | 480.27 |

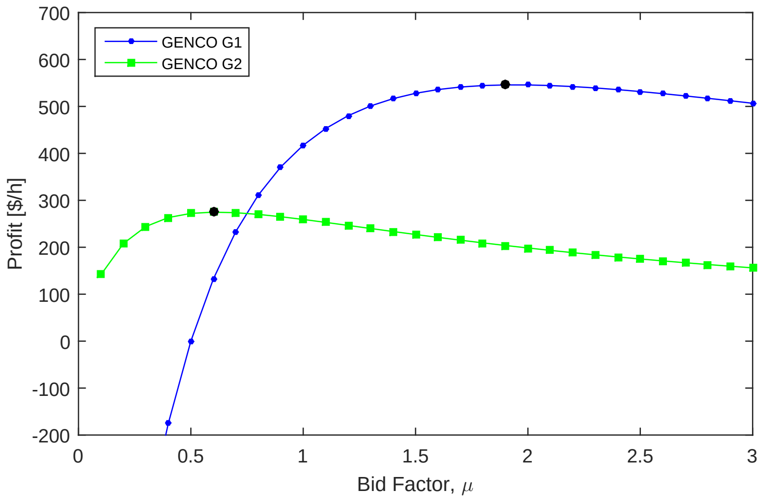



**Figure 2.6:** Illustration of the effect of the bid factor $\mu_i$ on GENCO profits.

## 2.4 Mathematical Optimization Methods

### 2.4.1 Lagrangian Relaxation

The Lagrangian relaxation (LR) method for solving an optimization problem works by incorporating some of the constraints of the problem into the objective function using a penalty term known as the *Lagrangian multiplier*. LR is a technique well suited for problems where the constraints can be divided into two sets: "good" constraints, with which the problem is solvable very easily, and "bad" constraints which make the problem harder to solve.

The main idea is to "relax" the problem by removing the "bad" constraints and putting them into the objective function, assigned with weights (the Lagrangian multiplier). Each weight represents a penalty which is added to a solution that does not satisfy the particular constraint. The relaxed problem is usually simpler to solve and through a systematic method of updating the multipliers an optimal solution to the original problem can be obtained.

Consider the following optimization problem:

$$\begin{aligned} &\textit{Minimize} && f(x) \\ &\textit{Subject to} && h_1(x) \le 0 \\ & && h_2(x) \le 0 \\ & && x \ge 0 \end{aligned} \tag{2.6}$$

If the set of constraints $h_1(x) \le 0$ represents the "bad" constraints, we can formulate a Lagrangian relaxation problem as:

$$\begin{aligned} &\textit{Minimize} && L(x, \lambda) = f(x) + \lambda h_1(x) \\ &\textit{Subject to} && h_2(x) \le 0 \\ & && x \ge 0; \quad \lambda \ge 0 \end{aligned} \tag{2.7}$$

where $\lambda$ is the Lagrangian multiplier. $L(x, \lambda)$ is referred to as the Lagrangian function. If the constraints are chosen correctly, the minimization of $L(x, \lambda)$ is much simpler than the solution of the original problem.

In equation 2.7, for a feasible solution of the original problem, $h_1(x)$ is less than or equal to zero. Hence, if $\lambda$ is positive ($\lambda > 0$), the value of $\lambda h_1(x)$ will always be less than zero and the value of the Lagrangian will be smaller than the value of the original problem ($f(x)$). It is therefore said that a solution of the relaxed problem (also known as the dual problem) forms a lower bound to the solution of the original problem (also known as the primal problem).

A determination of the value of $\lambda$ that maximizes the Lagrangian for feasible values of $x$ therefore gives a solution to the original problem. A duality gap is defined as the difference between the optimal solution to the original (primal) problem (2.7) and the optimal solution of the dual problem (2.7). The solution procedure is then to iteratively update the Lagrange multipliers until the duality gap is within a given tolerance or after a set number of iterations. The main challenge is then to determine an efficient method of tuning the Lagrange multipliers.

There are two important points to note here:

- LR is not in itself an optimization method. It is just a concept of penalizing certain constraints in the objective function to form an "easier" problem. An actual optimization technique such as dynamic programming is required to solve the relaxed optimization problem.
- The Lagrangian multiplier usually has a physical meaning of the cost of satisfying a given constraint. If at the end of the optimization algorithm, $\lambda_i = 0$, then the corresponding constraint has no effect on the optimal solution of the original problem and could as well be ignored in the problem formulation. Otherwise, the value of $\lambda$ indicates the cost of ensuring that the corresponding constraint is met and usually forms a basis for pricing such constraints. A good example is the use of the Locational Marginal Price (LMP) in deregulated power systems where a Lagrange multiplier corresponding to enforcing the power balance constraint at a given bus (location) is used to set the price of electricity at the stated bus.

### 2.4.2 Dynamic Programming

Dynamic programming (DP) is one of the optimization methods commonly used to solve the relaxed optimization problem that results after Lagrangian relaxation. DP works by transforming a complex problem into a sequence of simpler problems and finding a solution to the more complex problem by successively solving the simpler problems. The most important characteristic of DP is the multistage nature of the optimization procedure.

The basic idea behind DP is to take an optimization problem and somehow break it down into a reasonable number of subproblems in such a way that the optimal solutions to the smaller subproblems can be used to give the optimal solution to the main problem. The idea is illustrated here using a simple example.

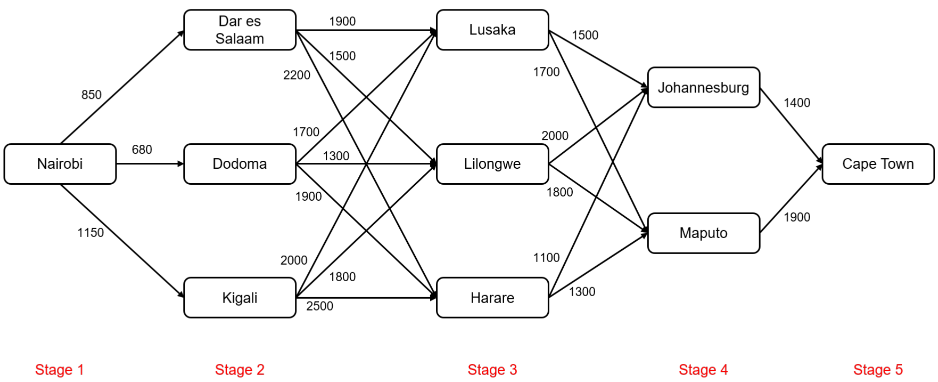



**Figure 2.7:** Multi-Stage Network Example

Figure 2.7 represents possible routes between Nairobi and Cape Town with eight possible stopover cities between the start and end. The numbers attached to the lines connecting two cities is the distance between the cities in km. The optimization problem is then to find the shortest path between the start and end cities. In this case, there are 36 possible routes $(2 \times 3 \times 3 \times 2)$ between Nairobi and Cape Town and it is possible to list all 36 routes and determine the shortest. Of course, the number of possible routes increases significantly as the number of possible stopover cities increases or as the connections become more complex.

This problem could also be solved using Dynamic Programming as follows:

1. *Determining the Shortest Paths to Stage 2*

   There is only one route to each of the three cities at stage 2 (Dar es Salaam, Dodoma, and Kigali). Hence, the shortest paths to the three are obvious as 850 km to Dar, 680 km to Dodoma, and 1150 km to Kigali.

2. *Determining the Shortest Paths to Stage 3*

   Lets consider the possible paths to Lusaka. There are three possibilities: Nairobi-Dar-Lusaka for 2750 km; Nairobi-Dodoma-Lusaka for 2380 km; and Nairobi-Kigali-Lusaka for 3150 km. Here, the shortest path to Lusaka is through Dodoma for 2380 km. It means that as long as the path to Cape Town goes through Lusaka, it is always shorter to pass through Dodoma. Hence, we can discard considering the paths Nairobi-Dar-Lusaka and Nairobi-Kigali-Lusaka in the future. Similarly, it is found that the shortest path to Lilongwe is Nairobi-Dodoma-Lilongwe for 1980 km and the shortest path to Harare is Nairobi-Dodoma-Harare for 2580 km.

3. *Determining the Shortest Paths to Stage 4*

   Next, we consider the possible paths to Johannesburg considering the shortest paths to the towns at stage 3. There are three possibilities: Nairobi-Dodoma-Lusaka-Johannesburg for 3880 km; Nairobi-Dodoma-Lilongwe-Johannesburg for 3980 km; and Nairobi-Dodoma-Harare-Johannesburg for 3680 km. Hence, the shortest path to Johannesburg is through Dodoma and Harare for 3680 km. We discard all other possible routes to Johannesburg. Similarly, it is found that the shortest path to Maputo is Nairobi-Dodoma-Lilongwe-Maputo for 3780 km.

4. *Determining the Shortest Paths to Stage 5 (Destination)*

Finally, we consider the possible paths to Cape Town considering the shortest paths to the towns at stage 4. There are two possibilities: Nairobi-Dodoma-Harare-Johannesburg-Cape Town for 5080 km or Nairobi-Dodoma-Lilongwe-Maputo-Cape Town for 5680 km. Clearly the former route is 600 km shorter and is hence the shortest path. This is stored as the solution to this optimization problem. The shortest paths to the various cities and the final optimal solution are highlighted in Figure 2.8.

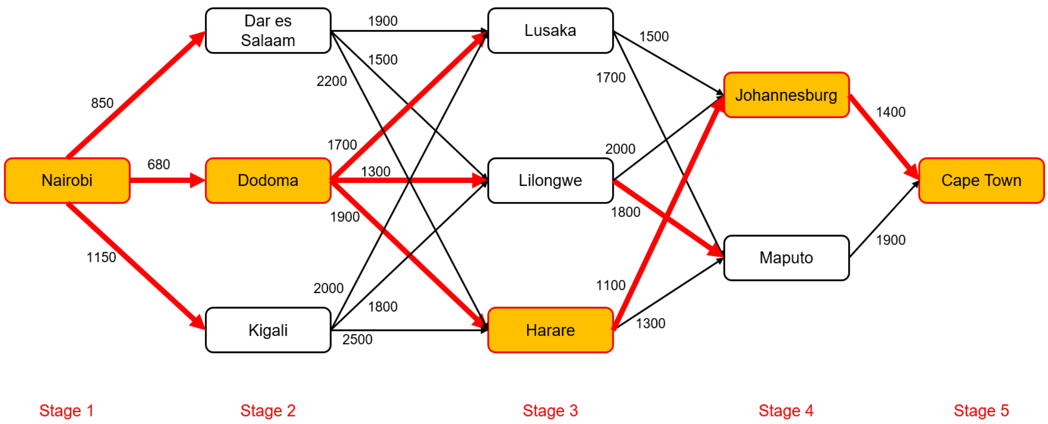



**Figure 2.8:** Multi-Stage Network Example Solution

In the above example, the shortest path problem has been solved sequentially using DP. The number of required path computations is significantly reduced when compared to enumerating all possible paths.

In DP terminology, each point where decisions are made is usually called a *stage* of the decision-making process and each stage has a number of *states* which define the decision made. The solution of the optimization problem using DP involves the following intuitive idea usually referred to as the *principle of optimality* [34].

*Any optimal policy has the property that, whatever the current state and decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from the current decision.* [34]

In other words, given the current state, the optimal decision for the remaining stages is independent of decisions made in previous states. For a problem with the *state-stage* structure – especially where the number of states at each stage are few – DP can be a very efficient method of solving an optimization problem.

The Unit Commitment problem in power systems is one such problem where the *stage* is each hour of the commitment schedule while the *state* is whether a unit is ON or OFF. The decision is therefore whether to turn a generator ON or OFF at a given hour to minimize the generation costs. DP has been applied severally in the solution of the UC problem in power systems especially when dealing with a few generating units.

### 2.4.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique for simulating the natural animal's behavior to adapt to the best of the characters among the entire populations like bird flocking and fish schooling [42]. Since its inception in the mid 90's, PSO has been widely applied by researchers in various optimization applications including the solution of the UC problem [7].

In simple terms, a population (swarm) of processing elements called particles, each of which representing a candidate solution forms the basis of computation in the PSO algorithm. A possible solution to the existing optimization problem is represented by each particle in the swarm. A population of random solutions is used to initialize the PSO algorithm and optima are searched by updating the solution in each iteration (epoch).

During a PSO iteration, every particle moves towards its own personal best solution that it achieved so far ($pBest$), as well as towards the global best ($gBest$) solution which is best among the best solutions achieved so far by all particles present in the population. This is done in a random manner ensuring that the algorithm searches the solution space as much as possible. After a certain preset number of iterations (generations), the particle with the global best solution is stored as the optimal solution to the optimization problem. A typical PSO algorithm flowchart is given in Fig. 2.9.

In the classical form of the PSO algorithm, sets of possible solutions to an optimization problem (particles) are moved around the solution space iteratively. At every iteration $k$, each particle $X_j$ moves according to a simple "move equation" written as:

$$X_j^k = X_j^{k-1} + V_j^k \tag{2.8}$$

where $V_j^k$ is the velocity of particle $j$ at the $k^{\text{th}}$ iteration defined by:

$$V_j^k = w_{j0}V_j^{k-1} + r_1 w_{j1}(pBest_j - X_j^{k-1}) + r_2 w_{j2}(gBest - X_j^{k-1}) \tag{2.9}$$

In (2.9), $w_{j0}$, $w_{j1}$, and $w_{j2}$ are weighting factors while $r_1$ and $r_2$ are uniformly distributed random numbers between 0 and 1. $pBest_j$ is the best solution (position) found by the particle in its past life while $gBest$ is the global best solution found by the swarm of particles in their past lives.

Equation (2.9) shows that the motion of each particle is dependent on three factors. The first term of (2.9) represents motion in the direction the particle is already moving in (the *inertia habit*); the second term represents motion towards the best solution the particle has found so far (the *memory habit*); and the third term represents motion towards the best solution found by all particles so far (the *information exchange or co-operation habit*).

The biggest challenge with the classical PSO algorithm lies on the choice of the weight parameters ($w_0$, $w_1$, and $w_2$) at the beginning of the solution algo-

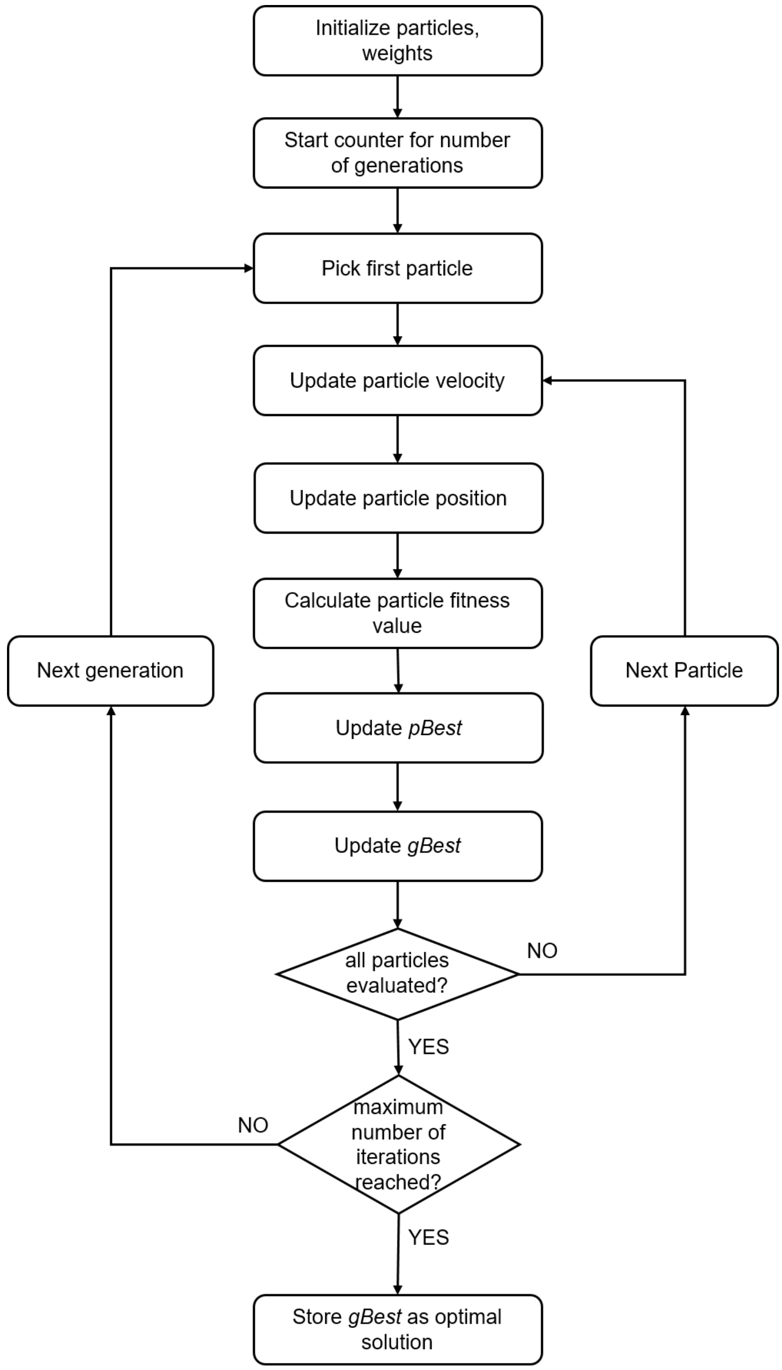



**Figure 2.9:** PSO algorithm Flowchart

rithm. It has been shown by many researchers that a parameter tuning method is imperative to obtain both a good solution and good convergence characteristics.

### 2.4.4 Evolutionary Particle Swarm Optimization

Evolutionary Particle Swarm Optimization (EPSO) is an optimization algorithm based on a combination of the evolutionary programming (EP) and Particle Swarm Optimization (PSO) concepts and works to handle the parameter tuning challenge of the classical PSO algorithm by progressively "mutating" the weight parameters with successive iterations.

The basic structure of EPSO as originally explained in [43] carries out the following processes at each iteration:

- REPLICATION - each particle is replicated $r$ times.
- MUTATION - each particle has its weights $w_{j0}$, $w_{j1}$, and $w_{j2}$ mutated.
- REPRODUCTION - each mutated particle generates an offspring according to the *particle movement rule.*
- EVALUATION - each offspring has its fitness evaluated.
- SELECTION - the best particles between the original set and the mutated set survive based on a stochastic tournament to form a new generation.

In EPSO, at the reproduction stage, new particles (offspring) are generated using a particle movement rule given by:

$$X_j^{k*} = X_j^{k-1} + V_j^{k*} \tag{2.10}$$

where $V_j^{k*}$ is a mutated version of the velocity of particle $j$ at the $k^{\text{th}}$ iteration defined by:

$$V_j^{k*} = w_{j0}^* V_j^{k-1} + w_{j1}^*(pBest_j - X_j^{k-1}) + w_{j2}^*(gBest^* - X_j^{k-1}) \tag{2.11}$$

The EPSO particle movement rule is similar to the classical PSO move equation with the following alterations:

- The model weights undergo mutation according to:

$$w_{jp}^{*} = w_{jp} + \tau_w \mathrm{N}(0,1) \qquad p = 0,1,2. \tag{2.12}$$

where $\mathrm{N}(0,1)$ is a random variable normally distributed with a mean of zero and a variance of 1.

- The global best $gBest$ is randomly disturbed as:

$$gBest^{*} = gBest + \tau_g \mathrm{N}(0,1) \tag{2.13}$$

The parameters $\tau_w$ and $\tau_g$ are learning parameters that define the width within which the weights and the global best values are defined.

By disturbing the position of the global best in (2.13), it is possible to search the area around the global best where the global optimum may lie rather than just the current global best and hence increase the chances of finding a better solution especially when the area defining the best solution is small.

Incorporation of the Darwinistic characteristics of mutation and selection allows the EPSO algorithm to take advantage of the faster convergence characteristics of Evolutionary Programming (EP) strategies. The EPSO flowchart is shown in Fig. 2.10 and can be contrasted with the classical algorithm of Fig. 2.9 to show the incorporation of the Darwinistic characteristics of mutation and selection.

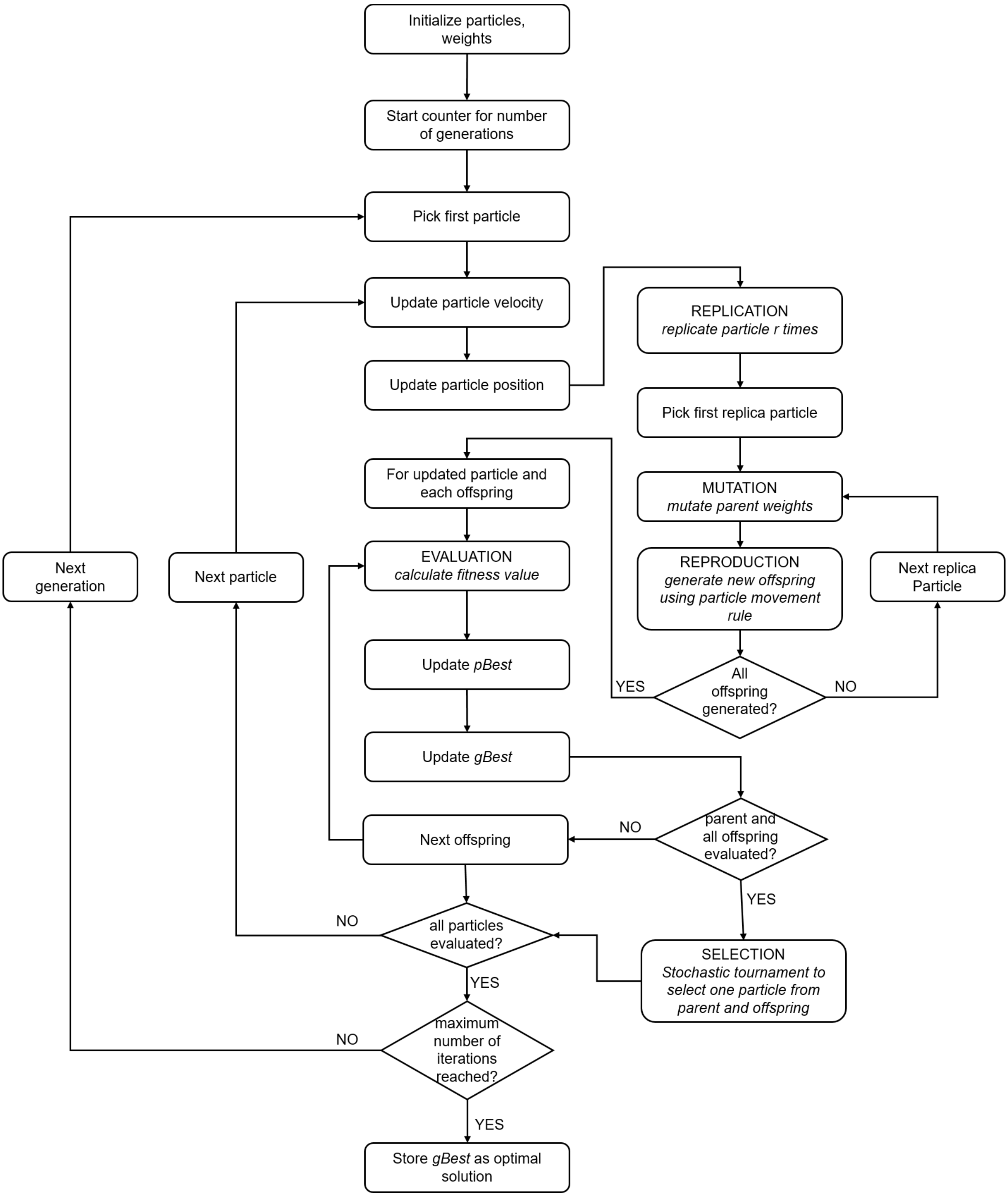



**Figure 2.10:** EPSO algorithm Flowchart

# CHAPTER THREE
# METHODOLOGY

In this section, the methodologies used in the research are explained. First, in section 3.1 the PBUC problem is formulated. The formulation in section 3.1.1 explains both the objective function of maximizing GENCO profit and the operational constraints to the mathematical problem. Then, in section 3.1.2, a solution algorithm using the hybrid LR-PSO method is explained. This is given in a step-by-step format. To overcome the difficulty of parameter selection, a hybrid LR-EPSO algorithm was also implemented as explained in section 3.1.3. As an extension to the PBUC problem formulation of section 3.1, and to study the effect of GENCO market power in the PBUC solution, an algorithm that also determines an optimal bidding strategy is implemented. The step-by-step methodology is explained in section 3.2. However, first Table 3.1 lists definitions of various variables used in the equations in this chapter.

**Table 3.1:** Nomenclature of indices, parameters, and variables

| | |
|---|---|
| *Indices* | |
| $h$ | hour index |
| $i$ | generator index |
| $j$ | PSO/EPSO particle index |
| $k$ | algorithm iteration number index |
| $r$ | EPSO replicated particle index |
| | |
| *Algorithm Parameters* | |
| $H$ | number of scheduling hours |
| $J$ | number of PSO/EPSO particles |
| $K$ | maximum number of PSO/EPSO algorithm generations |
| $N$ | total number of generating units owned by a GENCO |

| | |
|---|---|
| *Algorithm Parameters (Specific to EPSO Algorithm)* | |
| $R$ | number of replications of an EPSO particle |
| $\tau_\lambda$ | standard deviation of the random initial value of Lagrange multipliers |
| $\tau_g$ | standard deviation of the random disturbance of the value of $gBest$ |
| $\tau_w$ | standard deviation of the random mutation of a weight parameter |
| $p_{luck}$ | probability of the best offspring of a particle surviving after an iteration |
| *Test System Parameters* | |
| $a_i, b_i, c_i$ | fuel cost curve constants for generator $i$ |
| $CSC_i$ | cold start-up cost of generator $i$ |
| $HSC_i$ | hot start-up cost of generator $i$ |
| $CShr_i$ | Number of hours after which generator $i$ is considered cold |
| $\alpha_s^h$ | unit price for spot market energy sales at hour $h$ |
| $\alpha_b^h$ | unit price for bilateral contracts energy sales at hour $h$ |
| $\alpha_r^h$ | unit price for reserve capacity sales at hour $h$ |
| $\kappa$ | factor for contract of differences |
| $P_b^h$ | scheduled power generation for bilateral contracts at hour $h$ |
| $P_i^{min}, P_i^{max}$ | minimum and maximum outputs of generator $i$ respectively |
| $RU_i, RD_i$ | ramp up and ramp down limits of generator $i$ respectively |
| $MUT_i, MDT_i$ | minimum up time and minimum down time limits of generator $i$ respectively |
| *Variables (Common to both PSO and EPSO algorithms)* | |
| $PF$ | GENCO Profit |
| $RV$, $TC$ | GENCO revenue and costs respectively |
| $RVp^h$ | revenue from energy (MWh) sales at hour $h$ |
| $RVr^h$ | revenue from reserve capacity (MW) sales at hour $h$ |
| $FC_i^h$ | fuel cost of generator $i$ at hour $h$ |
| $SC_i^h$ | start up cost of generator $i$ at hour $h$ |
| $P_i^h$ | power output from generator $i$ at hour $h$ |
| $U_i^h$ | state of generator $i$ at hour $h$ |
| $pBest_{j,k}$ | personal best solution of particle $j$ at iteration $k$ |
| $gBest_k$ | global best solution for all particles at iteration $k$ |

*Variables (Specific to PSO Algorithm)*

| | |
|---|---|
| $\lambda_{j,k}^{h}$ | Lagrange Multiplier for particle $j$ at hour $h$ for iteration $k$ |
| $v_{j,k}^{h}$ | velocity of particle $j$ at hour $h$ for iteration $k$ |
| $w_1, w_2, w_3$ | weighting factors corresponding to the particle's previous velocity, |
| | personal best position and global best position respectively |
| $r_1, r_2$ | random numbers in [0  1] |

*Variables (Specific to EPSO Algorithm)*

| | |
|---|---|
| $\lambda_{j,k}^{h,r}$ | Lagrange Multiplier for the $r^{\text{th}}$ replica of particle $j$ at hour $h$ and iteration $k$ |
| $v_{j,k}^{h,r}$ | velocity of the $r^{\text{th}}$ replica of particle $j$ for hour $h$ and iteration $k$ |
| $w_{j,k}^{0,r}, w_{j,k}^{1,r}, w_{j,k}^{2,r}$ | weighting factors corresponding to the $r^{\text{th}}$ replica of particle $j$ at iteration $k$ |

## 3.1 Profit Based Unit Commitment Problem

### 3.1.1 Problem Formulation

As explained in section 2.2, the profit based unit commitment (PBUC) problem is formulated as as a maximization of a GENCO's profit. Mathematically, we seek to determine a GENCO's optimal unit commitment schedule i.e. generation units turn ON - turn OFF and power output schedule based on predicted energy and reserve prices. The GENCO's bilateral contract commitments are also considered. The objective function and the operational constraints are explained in the following subsections. There are numerous indices and variables used in the equations – all of which are summarized in Table 3.1.

#### 3.1.1.1 Objective Function

Profit ($PF$) is defined as the difference between revenue ($RV$) obtained from sale of energy and reserve[1] and the total operating cost ($TC$) of the GENCO. Mathematically, the PBUC problem objective function is given as:

$$\text{Maximize} \qquad PF = RV - TC \tag{3.1}$$

#### 3.1.1.2 GENCO Revenue

In (3.1), GENCO Revenue $RV$ is given by:

$$RV = \sum_{h=1}^{H} \left( RV_p^h + RV_r^h \right) \tag{3.2}$$

Revenue from the energy market at a given hour $RV_p^h$ is calculated as:

$$RV_p^h = \alpha_b^h P_b^h + \alpha^h \left( \sum_{i=1}^{N} P_i^h - P_b^h \right) + \kappa \left( \alpha^h - \alpha_b^h \right) P_b^h \tag{3.3}$$

The first term in (3.3) represents revenue from bilateral contracts, the second term represents revenue from the energy sold at the spot market, while the third term represents revenue from contracts of differences (cfds) [44].

Contracts of differences cfd's are usually included in bilateral contracts to compensate suppliers and consumers for differences between the bilaterally agreed prices and the prevailing market price. A cfd factor of $\kappa = 0$ would mean that the GENCO sells power in the bilateral market at the bilaterally agreed price even if the market price is higher (no compensation) while a cfd factor of $\kappa = 1$ means that the GENCO sells power in the bilateral market at the prevailing market price (full compensation). A value of $\kappa = 0.5$ is adopted in this research as this implies that both the GENCO and the consumer are compensated equally when

[1] Revenue from other ancillary services could be included in a similar manner.

there is a difference in prices.

Revenue from sale of reserve power at hour $h$ is given by:

$$RV_r^h = \alpha_r^h \sum_{i=1}^{N} \left(P_i^{max} - P_i^h\right) \tag{3.4}$$

In (3.4), it is assumed that both spinning and standing reserve are paid at the same rate. If the pricing is different, the equation could be split to have two terms accounting for each type of reserve.

#### 3.1.1.3 GENCO Costs

In (3.1), the total costs $TC$ is a sum of fuel costs[2] ($FC$) and start up costs[3] ($SC$) for all generators over the entire scheduling period. This is given as:

$$TC = \sum_{h=1}^{H} \sum_{i=1}^{N} \left(FC_i^h + SC_i^h\right) \tag{3.5}$$

where

$$FC_i^h = a_i + b_i P_i^h + c_i \left(P_i^h\right)^2 \tag{3.6}$$

$$SC_i^h = \begin{cases} CSC_i \cdot \left(1 - U_i^{h-1}\right) U_i^h & \text{if } \sum_{t=h-CShr}^{h} U_i^t \geq CShr \\ HSC_i \cdot \left(1 - U_i^{h-1}\right) U_i^h & \text{if } \sum_{t=h-CShr}^{h} U_i^t < CShr \end{cases} \tag{3.7}$$

#### 3.1.1.4 Operational Constraints

The GENCO operational constraints are given as:

[2]Equation (3.6) assumes the typical thermal generator cost equations. Costs for other types of generators could be included using appropriate models.

[3]Only the start up costs are considered in this research. If applicable, shut down costs could be included in a similar manner.

(a) Power balance for bilateral contracts

$$\sum_{i=1}^{N} P_i^h \geq P_b^h \qquad \forall h \tag{3.8}$$

(b) Generation limit constraints

$$P_i^{min} \leq P_i^h \leq P_i^{max} \qquad \forall i, \forall h \tag{3.9}$$

(c) Generator ramp up constraints

$$P_i^h - P_i^{h-1} \leq RU_i \qquad \forall i, \forall h \tag{3.10}$$

(d) Generator ramp down constraints

$$P_i^{h-1} - P_i^h \leq RD_i \qquad \forall i, \forall h \tag{3.11}$$

(e) Generator minimum up time

$$U_i^h = 1 \quad \text{if } U_i^t - U_i^{t-1} = 1, \qquad \text{for } h = t, ..., t + MUT - 1 \tag{3.12}$$

(f) Generator minimum down time

$$U_i^h = 0 \quad \text{if } U_i^{t-1} - U_i^t = 1, \qquad \text{for } h = t, ..., t + MDT - 1 \tag{3.13}$$

Constraints (3.9)−(3.13) are similar to the traditional UC formulation [12]. However, constraint (3.8) indicates that the GENCO's total generation must be greater than its bilateral contracts commitments. This is in contrast with the traditional case where generation must equal total system demand and losses. Also, unlike the traditional UC formulation, there is no spinning reserve constraint as this is not the GENCO's responsibility. The GENCO only gets payments for supplying part of the reserve. Revenue from reserve sales is therefore added to the objective function as given by (3.2).

### 3.1.2 Solution Algorithm Using LR-PSO

The basic structure of the solution algorithm for solving the PBUC problem using LR and PSO is shown in Fig. 3.1. Basically, a Lagrangian function is formed by relaxing constraint (3.8) into the objective function. This is because it is the only constraint that couples the units. Possible solutions to the relaxed problem are then randomly generated and iteratively solved using a two-step process.

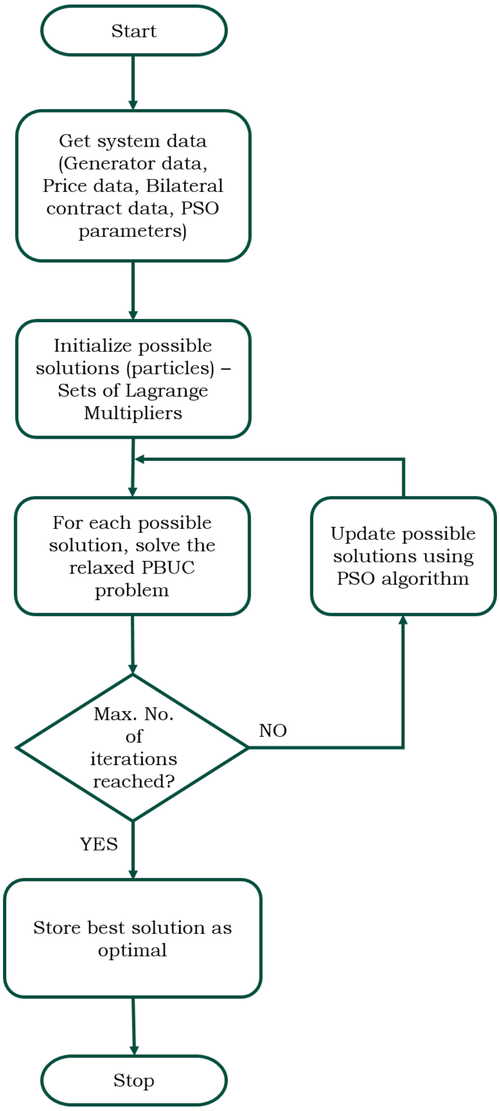



**Figure 3.1:** PBUC solution algorithm using LR-PSO

The first step involves solving the relaxed problem for each possible solution (sets of Lagrange multipliers). With the relaxation, optimal schedules of individual generation units can be easily determined by breaking down the relaxed function into subproblems for each unit. A 2-state dynamic programming code is implemented to find an optimal UC schedule for each unit given a set of Lagrange multipliers.

The second step involves updating of the possible solutions (particles) using the PSO algorithm. This is done iteratively for a number of pre-set iterations (maximum number of PSO generations). The two steps are outlined in the following subsections.

#### 3.1.2.1 Solution of the Relaxed Problem

Constraint (3.8) – the power balance for bilateral contracts – is the only constraint that couples the generating units and is therefore relaxed by being included in the objective function to form the Lagrangian function $L$ as:

$$L = RV - TC - \sum_{h=1}^{H} \lambda^h \left( P_b^h - \sum_{i=1}^{N} P_i^h \right) \tag{3.14}$$

The relaxed problem is therefore the maximization of $L$ subject to constraints (3.9) to (3.13).

To maximize $L$ with respect to $P_i^h$ in (3.20):

$$\frac{\partial L}{\partial P_i^h} = 0 \quad \forall i, h \tag{3.15}$$

i.e.

$$\frac{\partial L}{\partial P_i^h} = \left( \alpha_s^h - \alpha_r^h \right) - \left( b_i + 2c_i P_i^h \right) + \lambda^h = 0 \tag{3.16}$$

hence

$$P_i^h = \frac{\alpha_s^h - \alpha_r^h + \lambda^h - b_i}{2c_i} \tag{3.17}$$

The following procedure is thus used to solve the relaxed PBUC problem for a set of Lagrange multipliers: $\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^H\}$.

**Step 1**: Get input data (generator cost data, hourly price data, Lagrangian multipliers)

**Step 2**: Set $i = 1$

**Step 3**: Set $h = 1$

**Step 4**: calculate $P_i^h$ from (3.23)

**Step 5**: check for generator limit constraints
if $P_i^h > P_i^{max}$ set $P_i^h = P_i^{max}$
if $P_i^h < P_i^{min}$ set $P_i^h = P_i^{min}$

**Step 6**: check the ramp up and ramp down constraints and change $P_i^h$ accordingly

**Step 7**: check the minimum up time and minimum down time constraints and change $P_i^h$ accordingly

**Step 8**: determine the optimal UC schedule using 2-state dynamic programming

**Step 9**: $h = h + 1$. If $h \leq H$ go to **Step 4**. Else go to **Step 10**

**Step 10**: $i = i + 1$. If $i \leq N$ go to **Step 3**. Else go to **Step 11**

**Step 11**: Calculate total revenue, costs and profits

**Step 12**: Store the results (UC status for all generators, scheduled power, profit)

#### 3.1.2.2 Lagrange Multipliers Update via Particle Swarm Optimization

The PSO algorithm is used to update the Lagrange Multipliers to determine the set that provides the best results. A particle represents a candidate solution which is a set of Lagrange Multipliers – one for each hour of the scheduling horizon. For a scheduling period of $H$ hours, the $j^{th}$ particle after $k$ iterations $\Lambda_{j,k} = \{\lambda^1_{j,k}, \lambda^2_{j,k}, \lambda^3_{j,k}, \ldots, \lambda^H_{j,k}\}$ represents a position in the $H$-dimension solution space. The particle also has an associated velocity $V_{j,k} = \{v^1_{j,k}, v^2_{j,k}, v^3_{j,k}, \ldots, v^H_{j,k}\}$ which represents a direction in which the particle is moving in the solution space.

The PSO algorithm moves the particles around the solution space after each iteration in a search for the best possible solution. The particle position update follows two "best" positions: $pBest$ and $gBest$. $pBest_j$ is the $j^{th}$ particle's personal best solution found so far while $gBest$ is the entire population's global best solution (the best amongst the various $pBest$s).

At each iteration, the velocity of each particle is updated using:[4]

$$V_{j,k+1} = w_1 V_{j,k} + w_2 r_1 (pBest_j - \Lambda_{j,k}) + w_3 r_2 (gBest - \Lambda_{j,k}) \tag{3.18}$$

The position is then updated using the move equation:

$$\Lambda_{j,k+1} = \Lambda_{j,k} + V_{j,k+1} \tag{3.19}$$

The following procedure is used to solve the PBUC problem updating candidate solutions (sets of Lagrange Multipliers) using the PSO algorithm:

**Step 1**: Randomly initialize $J$ particles (candidate solutions)

**Step 2**: set $k = 1$

**Step 3**: set $j = 1$

**Step 4**: Solve the relaxed PBUC problem for the $j^{th}$ particle and determine the corresponding GENCO profit $PF_{j,k}$

**Step 5**: If $k = 1$, set $pBest_j = PF_{j,k}$

else if $PF_{j,k} > pBest_j$; set $pBest_j = PF_{j,k}$

**Step 6**: $j = j + 1$.

If $j < J$ go to **Step 4**. Else go to **Step 7**

[4]see variable definitions on the nomenclature list in Table 3.1.

**Step 7**: Determine $gBest$ as:

$gbest = \max\{pBest_1, pBest_2, \ldots, pBest_J\}$

**Step 8**: set $j = 1$

**Step 9**: Update the velocity of particle $j$ using (3.18)

**Step 10**: Update the position of particle $j$ using (3.19)

**Step 11**: $j = j + 1$.

If $j < J$ go to **Step 9**. Else go to **Step 12**

**Step 12**: $k = k + 1$

If $k \leq K$ go to **Step 3**. Else STOP

### 3.1.3 Solution Algorithm Using LR-EPSO

The proposed solution methodology involves the solution of a relaxed form (Lagrangian) of the PBUC problem. The Lagrangian function is formed by relaxing constraint (3.8) into the objective function. Possible solutions to the relaxed problem are then initialized and the solutions are iteratively updated using a two-step process.

The first step involves solving the relaxed problem for each possible solution (sets of Lagrange multipliers). With the relaxation, optimal schedules of individual generation units are determined by breaking down the relaxed function into subproblems for each unit. A 2-state dynamic programming code is implemented to find an optimal UC schedule for each unit given a set of Lagrange multipliers. The second step involves updating of the possible solutions (particles) using the EPSO algorithm.

The subsequent subsections outline: (A) formation of the Lagrangian function; (B) initialization of possible solutions; (C) solution of the relaxed problem and (D) updating of Lagrange multipliers using EPSO.

#### 3.1.3.1 Formation of Lagrangian Function

Constraint (3.8) – the power balance for bilateral contracts – is a unit coupling constraint meaning that a decision taken on one generator will affect decisions taken for the other generators. This makes it a difficult constraint to handle and it is therefore chosen to be relaxed using a set of Lagrange multipliers. Constraints (3.9) to (3.13) are not coupling constraints as they affect individual units independently.

A Lagrangian function $\mathcal{L}$ is formed as:

$$\mathcal{L} = RV - TC - \sum_{h=1}^{H} \lambda^h \left( P_b^h - \sum_{i=1}^{N} P_i^h \right) \tag{3.20}$$

The relaxed PBUC problem is then the maximization of $\mathcal{L}$ subject to constraints (3.9) to (3.13). For a given set of Lagrange multipliers: $\Lambda = \left\{\lambda^1, \lambda^2, \ldots, \lambda^H\right\}$, it is possible to determine a UC schedule that maximizes the Lagrangian function. The Lagrange multiplier set – and its corresponding UC schedule – that maximizes the Lagrangian function while meeting all operational constraints is then the optimal solution to the PBUC problem.

To maximize $\mathcal{L}$ with respect to $P_i^h$ in (3.20):

$$\left.\frac{\partial \mathcal{L}}{\partial P_i^h}\right|_{P_i^{h*}} = \left(\alpha_s^h - \alpha_r^h\right) - \left(b_i + 2c_i P_i^{h*}\right) + \lambda^h = 0 \tag{3.21}$$

or

$$\lambda^h = \left(b_i + 2c_i P_i^{h*}\right) - \left(\alpha_s^h - \alpha_r^h\right) \tag{3.22}$$

The term $b_i + 2c_i P_i^{h*}$ in (3.22) is the unit marginal cost when generating $P_i^{h*}$ MW. Hence, (3.22) states that, at the optimal generation level, the value of the Lagrange multiplier equals the difference between the marginal cost and the difference between the energy price and reserve price. This conclusion is used in section 3.1.3.2 to determine a suitable initial values of the Lagrange multipliers.

Making $P_i^{h*}$ the subject of the formula in (3.22):

$$P_i^{h*} = \frac{\alpha_s^h - \alpha_r^h + \lambda^h - b_i}{2c_i} \tag{3.23}$$

$P_i^{h*}$ is the optimal output of unit $i$ at hour $h$ corresponding to Lagrange multiplier $\lambda^h$ before considering the unit generation limits, minimum up time, minimum down time and ramp rate constraints. This conclusion is used in section 3.1.3.3 in the procedure for solving the relaxed PBUC problem.

#### 3.1.3.2 Initialization of Lagrange Multipliers

The solution space of the PBUC problem is large. For example, if the scheduling period is 24 hours, the solution will have 24 Lagrange multipliers hence the solution is defined in a 24-dimensional space. For such a large solution space, the chances of finding a good solution is reduced if the initial solution is not carefully chosen.

An initial "rough" solution is determined by solving the relaxed PBUC problem while disregarding the unit minimum up time, minimum down time and ramp rate constraints using the GENCO marginal cost curve as follows:

- For each hour, determine the marginal cost corresponding to the bilateral power commitment $MC(P_b^h)$ from the marginal cost curve.
- From (3.22), the initial value of the Lagrangian multiplier at hour $h$: $\lambda^{h,0}$ is given by:

$$\lambda^{h,0} = MC(P_b^h) - \left(\alpha_s^h - \alpha_r^h\right) \tag{3.24}$$

The Lagrange multipliers set: $\Lambda^0 = \left\{\lambda^{1,0}, \lambda^{2,0}, \ldots, \lambda^{H,0}\right\}$ is used as an initial solution to the PBUC problem.

The EPSO algorithm is initialized using random possible solutions (particles). Since the optimal solution when all constraints are considered will be close to $\Lambda^{h,0}$, the initial value of a Lagrange Multiplier (LM) corresponding to particle $k$ for hour $h$ is given by:

$$\lambda_k^h = \max\left\{0, \lambda^{h,0} + \tau_\lambda N(0,1)\right\} \tag{3.25}$$

where $\tau_\lambda$ is the standard deviation of initial randomly generated values of LM's and $N(0,1)$ is a normally distributed random number with a mean of zero and variance of 1.

#### 3.1.3.3 Solution of Relaxed Problem

The following procedure is used to solve the relaxed PBUC problem for a set of Lagrange multipliers: $\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^H\}$.

***Step 1***: Get the price data for both the energy and reserve markets and values of Lagrange multipliers for each scheduling hour.

***Step 2***: set $i = 1$.

***Step 3***: Get the input data for unit $i$: $P_i^{max}, P_i^{min}$ etc.

***Step 4***: Set $h = 1$.

***Step 5***: Calculate $P_i^{h*}$ from (3.23).

***Step 6***: Check and correct for generator limit constraints:

if $P_i^{h*} > P_i^{max}$, set $P_i^{h*} = P_i^{max}$

if $P_i^{h*} < P_i^{min}$, set $P_i^{h*} = P_i^{min}$

***Step 7***: Check and correct for unit ramp up and ramp down constraints:

if $P_i^{h*} > P_i^{h-1} + RU_i$, set $P_i^{h*} = P_i^{h-1} + RU_i$

if $P_i^{h*} < P_i^{h-1} - RD_i$, set $P_i^{h*} = P_i^{h-1} - RD_i$

***Step 8***: Compute the unit profits corresponding to various state transitions considering the minimum up time and minimum down time constraints and pick the optimal (more profitable) options.

***Step 9***: set $h = h + 1$. If all hours have been covered, go to ***Step 10*** else, go back to ***Step 5***.

***Step 10***: Pick the option that results in higher profits and return the corresponding UC schedule as optimal solution.

***Step 11***: set $i = i + 1$. If all generators have been covered, go to ***Step 12*** else, go back to ***Step 3***.

***Step 12***: Return the optimal UC schedule.

#### 3.1.3.4 Lagrange Multipliers Update Using Evolutionary Particle Swarm Optimization

The EPSO algorithm is used to update the Lagrange Multipliers to determine the set that provides the best results. In the solution of the PBUC problem, a particle represents a candidate solution to the problem i.e. a set of Lagrange Multipliers with one Lagrange multiplier for each hour of the scheduling horizon. Given a scheduling period of $H$ hours, the $j^{th}$ particle after $k$ iterations $\Lambda_{j,k} = \{\lambda^1_{j,k}, \lambda^2_{j,k}, \lambda^3_{j,k}, \ldots, \lambda^H_{j,k}\}$ represents a position in the $H$-dimension solution space. The particle also has an associated velocity $V_{j,k} = \{v^1_{j,k}, v^2_{j,k}, v^3_{j,k}, \ldots, v^H_{j,k}\}$ which represents a direction in which the particle is moving in the solution space. The particle also has an associated set of weights $W_{j,k} = \{w^0_{j,k}, w^1_{j,k}, w^2_{j,k}\}$ which govern the direction of particle movement. $w^0_{j,k}$ governs the particle's inertia habit, $w^1_{j,k}$ governs its memory habit, while $w^2_{j,k}$ governs its cooperation habit [43].

The following procedure is used to solve the PBUC problem while updating particles (candidate solutions) using the EPSO algorithm:

***Step 1***: *Initialization*:

Initialize $J$ particles $\Lambda_{j,0}$ $j = 1, 2, \ldots, J$. Each particle is a set of $H$ Lagrange multipliers whereby the Lagrange multiplier corresponding to the $j^{\text{th}}$ particle and hour $h$ is obtained from (3.25). Store each initialized particle as $pBest_j$; the fitness of each initialized particle as the best fitness value for the corresponding particle; and the fittest particle of all initialized particles as initial $gBest$.

***Step 2***: set $k = 1$.

***Step 3***: *Replication*

Each particle is replicated $R$ times i.e. $R$ new particles are created as:

$$\Lambda^r_{j,k} = \Lambda_{j,k} \qquad r = 1, 2, \ldots R \tag{3.26}$$

***Step 4***: *Mutation*

Each particle has its weights mutated as:

$$w^{l,r}_{j,k+1} = w^{l,0}_{j,k} + \tau_{w,l} N(0,1) \quad l = 0, 1, 2; \quad r = 1, 2, \ldots R \tag{3.27}$$

***Step 5***: *Reproduction*

Each particle and its replicas generate an offspring according to the *particle movement rule*[5].

$$\Lambda^r_{j,k+1} = \Lambda^r_{j,k} + V^r_{j,k+1} \qquad r = 0, 1, 2, \ldots R \tag{3.28}$$

where

$$V^r_{j,k+1} = w^{0,r}_{j,k} \cdot V^r_{j,k+1} + w^{1,r}_{j,k} \cdot \left(pBest_{j,k} - \Lambda^r_{j,k}\right) + w^{2,r}_{j,k} \cdot \left(gBest^*_k - \Lambda^r_{j,k}\right) \tag{3.29}$$

In (3.29), the $gBest_k$ value is disturbed to give $gBest^*_k$ using:

$$gBest^*_k = gBest_k + \tau_g N(0,1) \tag{3.30}$$

[5] $\Lambda^0_{j,k}$ refers to the original particle while $\Lambda^1_{j,k}$, $\Lambda^2_{j,k}, \ldots$ refer to the replica particles

***Step 6***: *Evaluation*

For each offspring, an optimal UC schedule is obtained by solving the relaxed PBUC problem as described by the procedure in section 3.1.3.3. The obtained UC schedule is used to calculate the offspring's fitness.

***Step 7***: *Updating pBest and gBest*

The fitness value of each offspring is used to update the $pBest_{j,k}$ and $gBest_k$ values.

***Step 8***: *Selection*

For each set of offspring, one is chosen to survive to the next generation through a stochastic tournament. The stochastic tournament is carried out as follows:

- The best particle between the offspring of each particle is determined.
- This particle survives to the next generation with a probability $p_{luck}$ while the other particles survive with a probability $(1 - p_{luck})/R$.
- If $p_{luck}$ is set to 1 then the best particle will always be chosen (pure *elitism* selection) while if $p_{luck}$ is set to $1/(R+1)$, there will pure random selection.

***Step 9***: *Convergence test*

$k = k + 1$. If $k = K$ go to ***Step 10***. Else go to ***Step 3***.

***Step 10***: Store $gBest_K$ and its corresponding UC schedule as the optimal solution and STOP.

## 3.2 OBS-PBUC Solution Methodology

Section 3.2.1 explains the procedure adopted to determine the profit corresponding to a given bidding strategy while section 3.2.2 details the step-by-step procedure implemented to select an optimal bidding strategy using the EPSO algorithm.

### 3.2.1 Profit Maximization Procedure

As illustrated in section 2.3, a GENCO can opt to *bid high* or *bid low* with respect to its marginal cost curve aiming to maximize its profits. Assume a linear reference marginal cost curve given by[6]:

$$MC^{\text{ref}} = \rho + \beta P_T^h, \tag{3.31}$$

where $\rho$ and $\beta$ are the marginal cost curve coefficients for the GENCO and $P_T^h$ is its total output at hour $h$. Then, let $\mu^h$ be the bid curve multiplying factor at hour $h$ so that the GENCO bid curve at hour $h$ is given by:

$$BC^h = \rho + \mu^h \beta P_T^h. \tag{3.32}$$

The value of $\mu^h$ then defines the GENCO's bidding strategy at hour $h$. For a scheduling period of $H$ time periods, the set of bid factors $\mathcal{U} = \{\mu^1, \mu^2, \mu^3, \ldots, \mu^H\}$ constitutes the GENCO's bidding strategy.

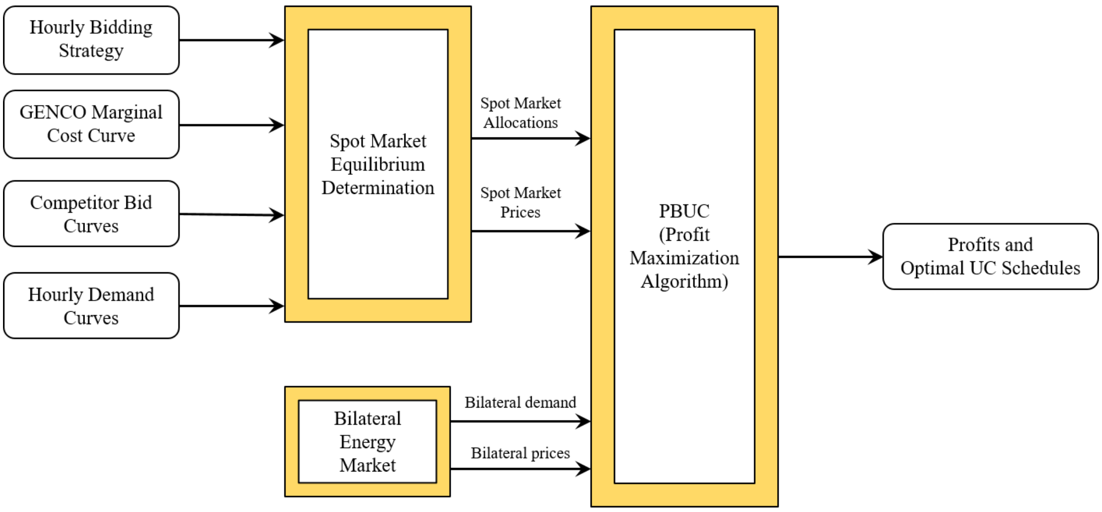



**Figure 3.2:** Profit Maximization procedure for a given bidding strategy.

For a given bidding strategy, the procedure used to determine a profit maxi-

[6]the subscript $i$ indicating the GENCO number is dropped to improve readability of the equations.

mization schedule is shown in Figure 3.2. Given a particular bidding strategy, the reference marginal cost curve, forecasted competitor bid curves, and the hourly demand curves, the GENCO can forecast the market's supply and demand curves and hence the market equilibrium point as illustrated in section 2.3. As seen from Figure 3.2, this gives the GENCO's spot market allocations and the MCPs (spot market prices). The spot market data is then combined with the bilateral market data (demand and prices) which is fed to a profit maximization algorithm to determine the optimal UC schedules and hence the profit associated with the bidding strategy $\mathcal{U}$.

### 3.2.2 EPSO Algorithm

Different bidding strategies give different spot market allocations and hence different optimal UC schedules. Thus, an algorithm that determines the optimal bidding strategy is implemented in this paper using the EPSO algorithm [43]. In the solution of the OBS-PBUC problem, a particle represents a candidate solution to the problem which in this case is a set of bid factors with one bid factor for each time period of the scheduling horizon. Given a scheduling period of $H$ hours, the $j^{th}$ particle after $k$ iterations $\mathcal{U}_{j,k} = \{\mu^1_{j,k}, \mu^2_{j,k}, \mu^3_{j,k}, \ldots, \mu^H_{j,k}\}$ represents a position in the $H$-dimension solution space. The particle also has an associated velocity $V_{j,k} = \{v^1_{j,k}, v^2_{j,k}, v^3_{j,k}, \ldots, v^H_{j,k}\}$ and an associated set of weights $W_{j,k} = \{w^0_{j,k}, w^1_{j,k}, w^2_{j,k}\}$. The velocity represents a direction in which the particle is moving in the solution space while the weights govern the direction of particle movement. $w^0_{j,k}$ governs the particle's inertia habit, $w^1_{j,k}$ governs its memory habit, while $w^2_{j,k}$ governs its cooperation habit [43].

A step by step outline of the procedure used to solve the OBS-PBUC problem using the EPSO algorithm follows.

***Step 1***: *Initialization*:

Randomly initialize $J$ particles $\mathcal{U}_{j,0}\quad j = 1, 2, \ldots, J$. Each particle is a set of $H$ bid factors defining a particular bidding strategy. For each particle, an optimal unit commitment schedule is obtained using the profit maximization procedure shown in Figure 3.2. The obtained profit $PF_{j,0}$ is the particle's initial fitness value. Each initialized particle is stored as $pBest_j$; the corresponding fitness values as the best fitness values; and the fittest particle of all initialized particles as initial $gBest$.

***Step 2***: Set the algorithm generation counter $k = 1$.

***Step 3***: Set the particles counter $j = 1$.

***Step 4***: *Replication*

Each particle is replicated $R$ times i.e. $R$ new particles are created as:

$$\mathcal{U}^r_{j,k} = \mathcal{U}_{j,k}, \qquad r = 1, 2, \ldots R. \tag{3.33}$$

***Step 5***: Set the particles replica counter $r = 0$.

***Step 6***: *Mutation*

The weights for replica $r$ of particle $j$ are mutated as:

$$w^{l,r}_{j,k+1} = w^{l,0}_{j,k} + \tau_{w^l} N(0, 1), \qquad l = 0, 1, 2; \tag{3.34}$$

where $\tau_{w^l}$ is the standard deviation of the random mutation of weight parameter $w^l$.

***Step 7***: *Reproduction*

A new offspring is generated according to the *particle movement rule*[7]:

$$\mathcal{U}^r_{j,k+1} = \mathcal{U}^r_{j,k} + V^r_{j,k+1}, \qquad r = 0, 1, 2, \ldots R, \tag{3.35}$$

[7] $\mathcal{U}^0_{j,k}$ refers to the original particle while $\mathcal{U}^1_{j,k}, \mathcal{U}^2_{j,k}, \ldots$ refer to the replica particles.

where

$$V^{r}_{j,k+1} = w^{0,r}_{j,k} \cdot V^{r}_{j,k} + w^{1,r}_{j,k} \cdot \left(pBest_{j,k} - \mathcal{U}^{r}_{j,k}\right) + w^{2,r}_{j,k} \cdot \left(gBest^{*}_{k} - \mathcal{U}^{r}_{j,k}\right). \tag{3.36}$$

In (3.36), the $gBest_k$ value is disturbed to give $gBest^*_k$ using:

$$gBest^*_k = gBest_k + \tau_g N(0,1) \tag{3.37}$$

where $\tau_g$ is the standard deviation of the random disturbance of the $gBest$ value.

***Step 8***: *Fitness Evaluation*

An optimal UC schedule is obtained using the procedure described by Figure 3.2. The profit obtained from the optimal UC schedule is is the offspring's fitness.

***Step 9***: Increase the replica counter by 1. If all replicas have been evaluated, go to ***Step 10***, else go back to ***Step 6***.

***Step 10***: *Updating pBest*

The fitness values of particle $j$'s offspring are used to update the $pBest_{j,k}$.

***Step 11***: *Selection*

One offspring is chosen to survive to the next generation through a stochastic tournament. The stochastic tournament is carried out as follows:

- The fittest between the particle's offspring is determined.
- This particle survives to the next generation with a probability $p_{luck}$ while the other particles survive with a probability $(1 - p_{luck})/R$.
- If $p_{luck}$ is set to 1 then the best particle will always be chosen (pure *elitism* selection) while if $p_{luck}$ is set to $1/(R+1)$, there will pure random selection.

***Step 12***: Increase the particle counter by 1. If all particles have been evaluated, go to ***Step 13***, else go back to ***Step 4***.

***Step 13***: *Updating gBest*

The original $gBest_{k-1}$ value and the highest profit from the $pBest_{j,k}$ values are used to update the $gBest_k$ value.

***Step 14***: Increase the algorithm generations counter by 1. If $K$ generations have been exhausted, go to ***Step 15***, else go back to ***Step 3***.

***Step 15***: Store $gBest_K$ and its corresponding UC schedule as the optimal solution and STOP.

# CHAPTER FOUR
# RESULTS AND DISCUSSION

This section presents simulation results from the various algorithms described in Chapter 3 and the final software tool developed for PBUC problem solution. Section 4.1 gives results of the PBUC solution using LR-PSO, section 4.2 gives results of PBUC solution using LR-EPSO, while section 4.3 gives results of the OBS-PBUC solution using LR-EPSO illustrating the effect of GENCO market power. Finally, section 4.4 describes the software tool developed for solution of the PBUC problem using LR-PSO or LR-EPSO algorithms. The tool is given as a MATLAB toolbox i.e. a collection of MATLAB m-files which can be run for a set of input data given in an excel file. Screenshots of inputs and outputs of the software tool are also given.

## 4.1 PBUC Solution Using LR-PSO

### 4.1.1 Test System

The algorithm described in section 3.1.2 is tested for a GENCO with 54 thermal units. The generator data is adapted from the IEEE 118-bus test system and obtained from [47]. The GENCO's own load (bilateral market commitment) is assumed to be constant at 3,500 MW with PEAK and OFF-PEAK prices as shown in Table 4.1 [47].

### 4.1.2 Selection of PSO Parameters

The quality of the solution obtained from the PSO algorithm is largely dependent on the values of the parameters used. Parameter selection is done in this research by trying various combinations of the weighting factors $w_1$, $w_2$, and $w_3$ in equation (3.18). $w_1$ was varied from 0.25 to 1.0 in steps of 0.25, while $w_2$ was varied

**Table 4.1:** Electricity Market Price Data

| Hour | Energy Price | Reserve Price | Bilateral Price | Hour | Energy Price | Reserve Price | Bilateral Price |
|---|---|---|---|---|---|---|---|
| 1 | 29.23 | 2.00 | 30.00 | 13 | 57.01 | 2.77 | 56.00 |
| 2 | 26.40 | 1.70 | 30.00 | 14 | 54.42 | 2.87 | 56.00 |
| 3 | 22.47 | 1.27 | 30.00 | 15 | 63.12 | 2.92 | 56.00 |
| 4 | 21.07 | 1.12 | 30.00 | 16 | 65.59 | 3.32 | 56.00 |
| 5 | 23.16 | 1.35 | 30.00 | 17 | 67.24 | 3.23 | 56.00 |
| 6 | 30.86 | 2.18 | 30.00 | 18 | 63.87 | 2.97 | 56.00 |
| 7 | 31.56 | 2.17 | 30.00 | 19 | 55.61 | 2.96 | 56.00 |
| 8 | 47.39 | 2.34 | 56.00 | 20 | 52.55 | 2.73 | 56.00 |
| 9 | 49.70 | 2.51 | 56.00 | 21 | 47.55 | 2.35 | 30.00 |
| 10 | 52.10 | 2.69 | 56.00 | 22 | 39.69 | 1.76 | 30.00 |
| 11 | 55.35 | 2.94 | 56.00 | 23 | 37.00 | 1.57 | 30.00 |
| 12 | 55.50 | 2.95 | 56.00 | 24 | 30.51 | 1.16 | 30.00 |

**Table 4.2:** PSO Parameter Sets

| Set No. | $w_1$ | $w_2$ | $w_3$ | Set No. | $w_1$ | $w_2$ | $w_3$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.25 | 1.00 | 3.00 | 11 | 0.75 | 1.00 | 3.00 |
| 2 | 0.25 | 1.50 | 2.50 | 12 | 0.75 | 1.50 | 2.50 |
| 3 | 0.25 | 2.00 | 2.00 | 13 | 0.75 | 2.00 | 2.00 |
| 4 | 0.25 | 2.50 | 1.50 | 14 | 0.75 | 2.50 | 1.50 |
| 5 | 0.25 | 3.00 | 1.00 | 15 | 0.75 | 3.00 | 1.00 |
| 6 | 0.50 | 1.00 | 3.00 | 16 | 1.00 | 1.00 | 3.00 |
| 7 | 0.50 | 1.50 | 2.50 | 17 | 1.00 | 1.50 | 2.50 |
| 8 | 0.50 | 2.00 | 2.00 | 18 | 1.00 | 2.00 | 2.00 |
| 9 | 0.50 | 2.50 | 1.50 | 19 | 1.00 | 2.50 | 1.50 |
| 10 | 0.50 | 3.00 | 1.00 | 20 | 1.00 | 3.00 | 1.00 |

from 1.0 to 3.0 in steps of 0.5. $w_3$ was set using the formula: $w_2 + w_3 = 4$ as suggested in literature [42]. These settings give 20 different combinations of the PSO parameters as shown in Table 4.2. In each case, the number of particles was set to $J = 20$ and the number of PSO iterations was set to $K = 500$. The Lagrange multipliers were initialized to take random values ranging from 0 to 50. The velocity was however not restricted so that the final value of the Lagrange multipliers could be any positive real number. For each combination of PSO parameters, 10 different trials of the PSO algorithm were run and the solutions analyzed.

The maximum profit, average profit, and minimum profit from each combination of PSO parameters was determined and the results are shown in Fig. 4.1. From Fig. 4.1, it is seen that the 12$^{\text{th}}$ combination of PSO parameters ($w_1 = 0.75$; $w_2 = 1.5$; and $w_3 = 2.5$) provides the best results in terms of obtained values of GENCO profit. Hence, for the subsequent simulations these values are chosen as the PSO parameters.

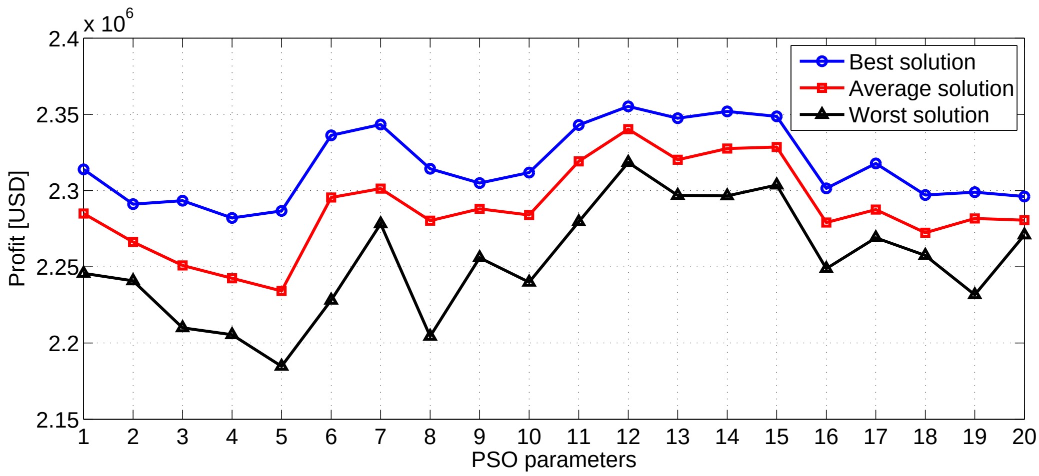



**Figure 4.1:** PSO Parameter Sets Performance

### 4.1.3 Optimal Solution

#### 4.1.3.1 Unit Commitment

The Unit Commitment schedule for the best solution amongst all the trials carried out is shown in Fig. 4.2. The horizontal axis represents the scheduling hour, while the vertical axis refers to the unit number. A single box in the grid therefore, indicates whether a unit is ON (shown in red) or OFF (shown in white). The results show that some of the units e.g. 27 and 45 are ON throughout the day, while others such as 33 and 46 are OFF throughout the day. Most of the units are ON or OFF depending on the market price at a given hour.

#### 4.1.3.2 Optimal Power Schedule

Fig. 4.3 shows the total committed generation for the 24 hours and the GENCO's own load from the UC schedule of Fig. 4.2. It also indicates the day's total profit as $2,355,259. From Fig. 4.3, the total scheduled power from the LR-PSO algorithm is always greater than the GENCO's load. There is no deficiency in meeting the bilateral contract agreements hence the value of the Energy Not Served ($ENS$) is indicated as zero. Should there be a deficiency in meeting the total committed schedule, the value of $ENS$ will be greater than zero. The value of $ENS = 0$ is ensured by penalizing a result in which $ENS > 0$ when determining the $pBest$ and $gBest$ value in the PSO algorithm.

| Gen No. | HOUR | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 14 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 19 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 21 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 22 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 23 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 24 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 25 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 26 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 27 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 28 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 29 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 36 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 37 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 39 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 40 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 43 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 44 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 45 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 47 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 54 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

**Figure 4.2:** Unit Commitment Schedule Corresponding to the Optimal Solution

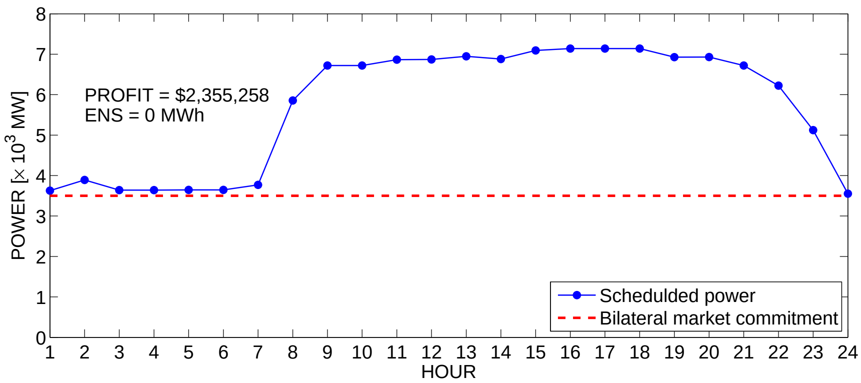



**Figure 4.3:** Optimal GENCO Total Power Generation Schedule

#### 4.1.3.3 Optimal Values of Lagrange Multipliers

Fig. 4.4 shows the resulting values of the Lagrange Multipliers corresponding to the schedule shown in Fig. 4.2. It is observed that the LMs are larger for durations of low market price (hrs 0 to 8) and when the market price is lower than the bilateral contract price (hr 14, 20, 21). In these cases, it is relatively expensive to participate in the spot market but it is necessary to generate power to meet bilateral contract commitments. During the periods of relatively high spot market price and when the spot market price is higher than the bilaterally agreed price, constraint (3.8) is met and there is no need to add a penalty factor hence the value $LM = 0$.

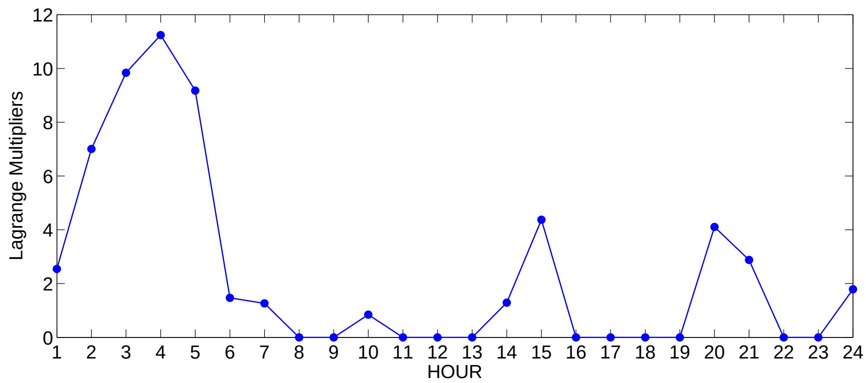



**Figure 4.4:** Lagrange Multipliers Corresponding to the Optimal Solution

#### 4.1.3.4 Solution Convergence and Computation Time Analysis

Fig. 4.5 shows the evolution of the best solution (value of $gBest$) as well as the computation time against the algorithm iteration number. It is seen that after about 300 iterations, the optimal solution does not change much hence it is sufficient to say that 500 iterations are enough for the current problem size. The solution time increases linearly with the number of iterations hence increasing the number of iterations would only increase the computation time without significantly improving the best solution.

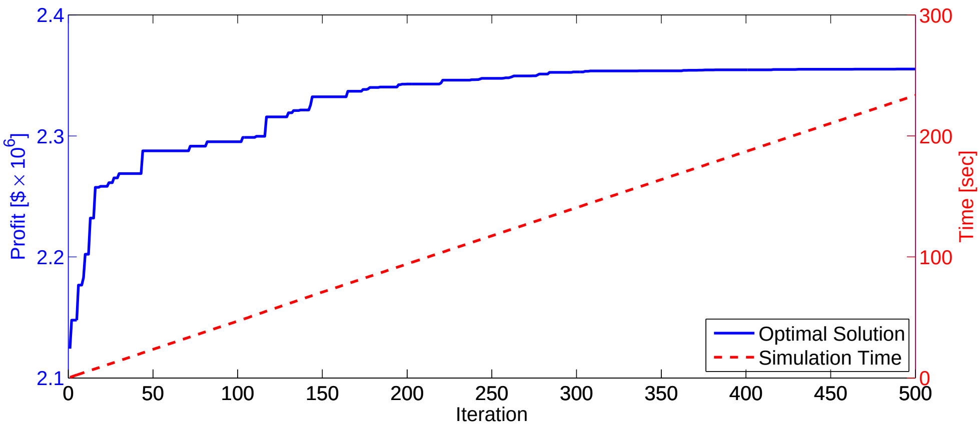



**Figure 4.5:** Analysis of solution convergence and computation time

## 4.2 PBUC Solution Using LR-EPSO

### 4.2.1 Test System

The methodology described in section 3.1.3 is implemented in MATLAB and tested for a GENCO with 10 thermal units whose data is shown in Table 4.3. The data is adapted from the IEEE 118-bus test system which has 54 thermal units. Other unit data such as ramp rate limits and minimum up and down times can be found from http://motor.ece.iit.edu/data/ PBUCData.pdf [46]. The total installed capacity for the GENCO is 830 MW, about 10% of the system installed capacity of 8270 MW. In this case, because of the size of the GENCO it can be assumed to be a price taker (negligible market power) and hence it can be assumed that there is no relationship between its power output and the electricity market price.

**Table 4.3:** Generating Units Data

| Unit No. | $P_i^{min}$ [MW] | $P_i^{max}$ [MW] | $P_i^0$ [MW] | $a$ [$] | $b$ [$/MW] | $c$ [$/MW$^2$] |
|---|---|---|---|---|---|---|
| 1 | 8 | 20 | 0 | 35.90 | 75.39 | 0.05660 |
| 2 | 8 | 20 | 0 | 35.90 | 75.39 | 0.05660 |
| 3 | 5 | 30 | 0 | 63.34 | 52.49 | 0.13932 |
| 4 | 5 | 30 | 0 | 63.34 | 52.49 | 0.13932 |
| 5 | 20 | 50 | 0 | 117.62 | 45.88 | 0.01954 |
| 6 | 25 | 50 | 0 | 117.62 | 45.88 | 0.01954 |
| 7 | 30 | 80 | 40 | 148.66 | 30.94 | 0.09184 |
| 8 | 25 | 100 | 100 | 20.30 | 35.64 | 0.02560 |
| 9 | 50 | 200 | 100 | 78.00 | 26.58 | 0.00880 |
| 10 | 50 | 250 | 250 | 56.00 | 24.66 | 0.00480 |

Fig. 4.6 shows the GENCO's marginal cost curve obtained from the unit characteristics of Table 4.3. The marginal cost curve is used to determine the initial values of the Lagrange multipliers as explained in Section 3.1.3.2. The hourly price of energy and reserve is shown in Table 4.4. It is assumed that reserve price is the same for both standing and spinning reserve. The hourly bilateral

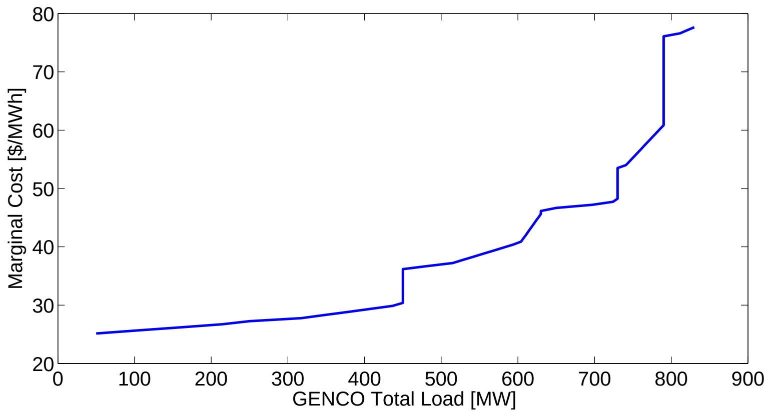



**Figure 4.6:** GENCO Marginal cost curve

market commitment and price is shown in Table 4.5. The hourly bilateral market price is assumed to be 10% higher than the marginal cost corresponding to the bilaterally committed load as can be read from Fig.4.6.

**Table 4.4:** Energy and Reserve Price Data

| Hour | Energy Price [$/MWh] | Reserve Price [$/MWh] | Hour | Energy Price [$/MWh] | Reserve Price [$/MWh] |
|---|---|---|---|---|---|
| 1 | 29.23 | 2.00 | 13 | 57.01 | 2.77 |
| 2 | 26.40 | 1.70 | 14 | 54.42 | 2.87 |
| 3 | 22.47 | 1.27 | 15 | 63.12 | 2.92 |
| 4 | 21.07 | 1.12 | 16 | 65.59 | 3.32 |
| 5 | 23.16 | 1.35 | 17 | 67.24 | 3.23 |
| 6 | 30.86 | 2.18 | 18 | 63.87 | 2.97 |
| 7 | 31.56 | 2.17 | 19 | 55.61 | 2.96 |
| 8 | 47.39 | 2.34 | 20 | 52.55 | 2.73 |
| 9 | 49.70 | 2.51 | 21 | 47.55 | 2.35 |
| 10 | 52.10 | 2.69 | 22 | 39.69 | 1.76 |
| 11 | 55.35 | 2.94 | 23 | 37.00 | 1.57 |
| 12 | 55.50 | 2.95 | 24 | 30.51 | 1.16 |

**Table 4.5:** Bilateral Market Data

| Hour | Bilateral Load [MW] | Bilateral Price [$/MWh] | Hour | Bilateral Load [MW] | Bilateral Price [$/MWh] |
|---|---|---|---|---|---|
| 1 | 397 | 32.09 | 13 | 422 | 32.57 |
| 2 | 387 | 31.89 | 14 | 412 | 32.37 |
| 3 | 371 | 31.58 | 15 | 417 | 32.47 |
| 4 | 360 | 31.37 | 16 | 422 | 32.57 |
| 5 | 347 | 31.12 | 17 | 435 | 32.82 |
| 6 | 380 | 31.75 | 18 | 445 | 33.21 |
| 7 | 397 | 32.08 | 19 | 457 | 39.91 |
| 8 | 417 | 32.47 | 20 | 467 | 40.10 |
| 9 | 427 | 32.66 | 21 | 472 | 40.19 |
| 10 | 442 | 33.08 | 22 | 447 | 33.30 |
| 11 | 445 | 33.21 | 23 | 440 | 33.00 |
| 12 | 432 | 32.76 | 24 | 427 | 32.66 |

### 4.2.2 Comparison of Solution Quality

The PBUC problem was solved using EPSO for the GENCO described in Section 4.2.1. For the purpose of solution quality and convergence characteristics comparison, an equivalent methodology based on the classic PSO algorithm [45] was also tested. Both algorithms were run thirty times. In each case, the initial weights were randomly generated from uniform distributions. $w_j^0$ was drawn from the range [0 1], while both $w_j^1$ and $w_j^2$ were drawn from the range [0 2]. In each trial the same values of the weights were used for both the EPSO and PSO algorithms. The other EPSO parameters are set as: $J = 20$, $K = 500$, $R = 1$, $p_{luck} = 0.8$, $\tau_\lambda = 1$, $\tau_w = 0.5$, and $\tau_g = 0.05$. For both algorithms, the best solution, average solution, and worst solutions were determined and are reported in Table 4.6. It is clearly seen that the proposed EPSO algorithm produces better solutions than the classic PSO algorithm in terms of the final value of the GENCO profit.

**Table 4.6:** Comparison of Solution Quality obtained by PSO and EPSO

| Algorithm | Best Solution | Average Solution | Worst Solution |
|---|---|---|---|
| PSO | $178,069 | $177,591 | $172,685 |
| EPSO | $178,911 | $178,317 | $177,408 |

The GENCO's total committed generation from the best run of the EPSO and PSO algorithms are shown in Fig. 4.7. The slight differences in the two curves result in the differences in the values of profit shown in Table 4.6.

### 4.2.3 Comparison of Convergence Characteristics

The convergence characteristics of the average value of the objective function at each iteration is shown in Fig. 4.8 It is seen that the proposed EPSO algorithm converges faster and generally to a higher value of profit than the PSO algorithm. On average, it takes about less than 6 iterations for the EPSO algorithm to reach

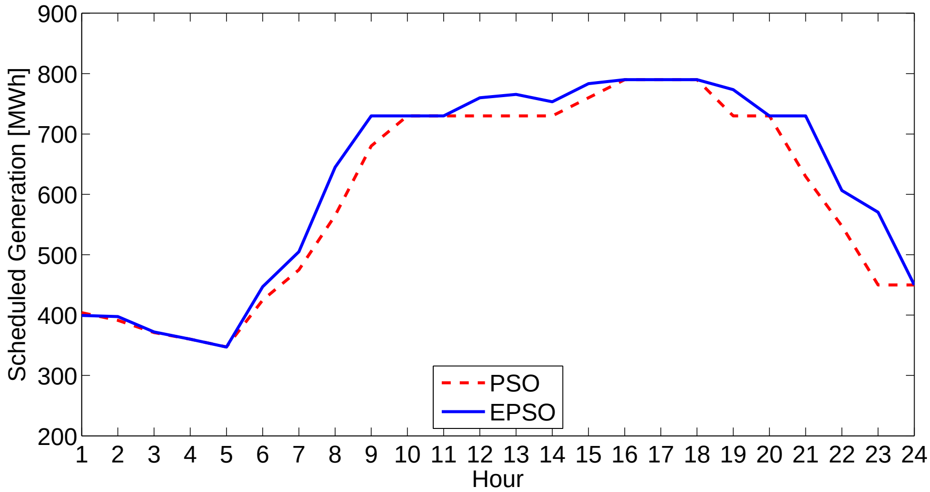



**Figure 4.7:** Comparison of best generation schedule produced by the EPSO and PSO algorithms

the average final objective function value reached by the PSO algorithm. The better performance of the EPSO algorithm confirmed by Table 4.6 and Fig. 4.8 is due to the inherent parameter tuning characteristic due to the mutation step of the algorithm.

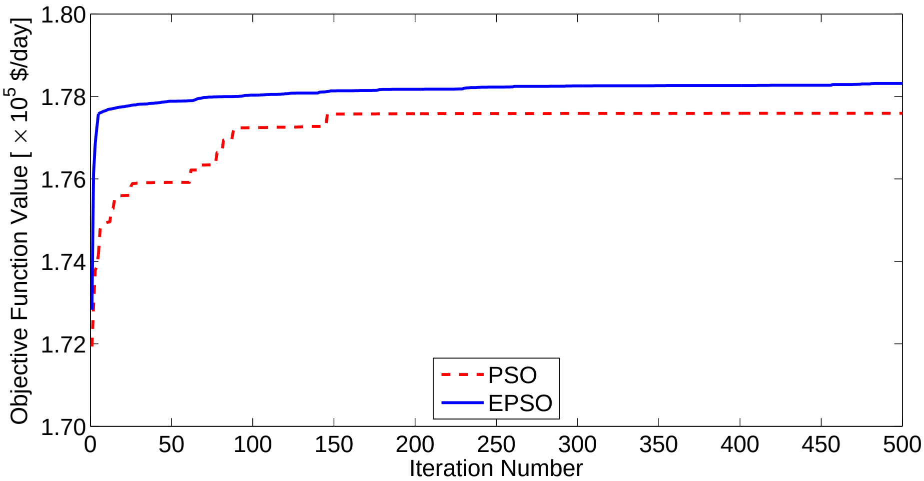



**Figure 4.8:** Comparison of PSO and EPSO convergence characteristics

## 4.3 OBS-PBUC Solution Illustrating GENCO Market Power

### 4.3.1 Test System

The IEEE 118-bus system data [46, 47] was used to simulate a deregulated electricity market environment with three GENCOs of different sizes in terms of installed capacity of generators. The three GENCOs operate several of the 54 thermal units in the IEEE 118-bus test system and the generating units data are given in Tables 4.7, 4.8, and 4.9 for GENCOs A, B, and C respectively.

The generator cost coefficients are scaled up from the values given in [47] so as to give more realistic energy prices. Based on the installed capacity, GENCO A controls about 60% (4340 MW out of 7220 MW) of the system capacity; GENCO B controls about 30% (2140 MW out of 7220 MW); while GENCO C controls about 10% (740 MW out of 7220 MW) of the system capacity. The reference linear marginal cost curves for each of the three GENCOs and the aggregated system marginal cost curve are shown in Figure 4.9. The marginal cost curves show that GENCO A operates the cheaper units while GENCO C operates the most expensive units.

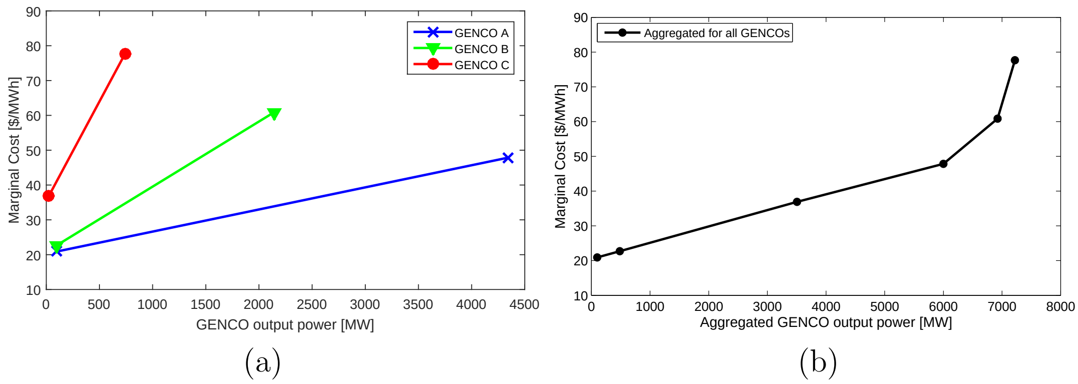



**Figure 4.9:** (a) Individual marginal cost curves for the three GENCOs and (b) aggregated system marginal cost curve.

**Table 4.7:** GENCO A's Generator Data

| Unit Code | No. of Units | $P_i^{\min}$ [MW] | $P_i^{\max}$ [MW] | Capacity [MW] | $a$ [$/h] | $b$ [$/MWh] | $c$ [$/MWh$^2$] | MUT [hrs] | MDT [hrs] | RU [MW] | RD [MW] | HSC [$/h] | CSC [$/h] | CShr [hrs] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 2 | 100 | 420 | 840 | 128.32 | 16.68 | 0.0212 | 10 | 10 | 210 | 210 | 250 | 500 | 20 |
| A2 | 8 | 100 | 300 | 2400 | 13.56 | 25.78 | 0.0218 | 8 | 8 | 150 | 150 | 110 | 220 | 16 |
| A3 | 2 | 50 | 250 | 500 | 56.00 | 24.66 | 0.0048 | 8 | 8 | 125 | 125 | 100 | 200 | 16 |
| A4 | 1 | 50 | 200 | 200 | 13.56 | 25.78 | 0.0218 | 8 | 8 | 100 | 100 | 400 | 800 | 16 |
| A5 | 3 | 25 | 100 | 300 | 20.30 | 35.64 | 0.0256 | 5 | 5 | 50 | 50 | 50 | 100 | 10 |
| A6 | 2 | 25 | 50 | 100 | 117.62 | 45.88 | 0.0195 | 2 | 2 | 25 | 25 | 45 | 90 | 4 |
| Total | 18 | | | 4340 | | | | | | | | | | |

**Table 4.8:** GENCO B's Generator Data

| Unit Code | No. of Units | $P_i^{\min}$ [MW] | $P_i^{\max}$ [MW] | Capacity [MW] | $a$ [$/h] | $b$ [$/MWh] | $c$ [$/MWh$^2$] | MUT [hrs] | MDT [hrs] | RU [MW] | RD [MW] | HSC [$/h] | CSC [$/h] | CShr [hrs] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B1 | 1 | 100 | 350 | 350 | 65.92 | 21.50 | 0.0060 | 8 | 8 | 175 | 175 | 100 | 200 | 16 |
| B2 | 1 | 100 | 300 | 300 | 65.92 | 21.50 | 0.0060 | 8 | 8 | 150 | 150 | 440 | 880 | 16 |
| B3 | 2 | 50 | 200 | 400 | 78.00 | 26.58 | 0.0088 | 8 | 8 | 100 | 100 | 100 | 200 | 16 |
| B4 | 8 | 25 | 100 | 800 | 20.30 | 35.64 | 0.0256 | 5 | 5 | 50 | 50 | 50 | 100 | 10 |
| B5 | 1 | 20 | 50 | 50 | 117.62 | 45.88 | 0.0195 | 2 | 2 | 25 | 25 | 45 | 90 | 4 |
| B6 | 8 | 5 | 30 | 240 | 63.34 | 52.49 | 0.1393 | 1 | 1 | 15 | 15 | 40 | 80 | 2 |
| Total | 21 | | | 2140 | | | | | | | | | | |

**Table 4.9:** GENCO C's Generator Data

| Unit Code | No. of Units | $P_i^{\min}$ [MW] | $P_i^{\max}$ [MW] | Capacity [MW] | $a$ [$/h] | $b$ [$/MWh] | $c$ [$/MWh$^2$] | MUT [hrs] | MDT [hrs] | RU [MW] | RD [MW] | HSC [$/h] | CSC [$/h] | CShr [hrs] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 4 | 25 | 100 | 400 | 20.30 | 35.64 | 0.0256 | 5 | 5 | 50 | 50 | 50 | 100 | 10 |
| C2 | 1 | 30 | 80 | 80 | 48.66 | 30.94 | 0.0918 | 3 | 3 | 40 | 40 | 45 | 90 | 6 |
| C3 | 6 | 5 | 30 | 180 | 63.34 | 52.49 | 0.1393 | 1 | 1 | 15 | 15 | 40 | 80 | 2 |
| C4 | 4 | 5 | 20 | 80 | 35.90 | 75.39 | 0.0566 | 1 | 1 | 10 | 10 | 30 | 60 | 2 |
| Total | 15 | | | 740 | | | | | | | | | | |

Nominal market clearing prices $\alpha_s^h$ corresponding to a spot market demand $P_D^h$ can be read off from the aggregated reference marginal cost curve of Figure 4.9(b). Additionally, linear demand curves are assumed for various load levels with a per-unit gradient of $-5$ i.e.

$$\frac{\Delta\alpha_s^h/\alpha_s^h}{\Delta P_T^h/P_T^h} = -5. \tag{4.1}$$

Equation (4.1) means that a 100% increase in the spot market price would result in a 20% reduction in the spot market demand. A 24-hour (day ahead) scheduling period is applied and the load level is shown in Figure 4.10.

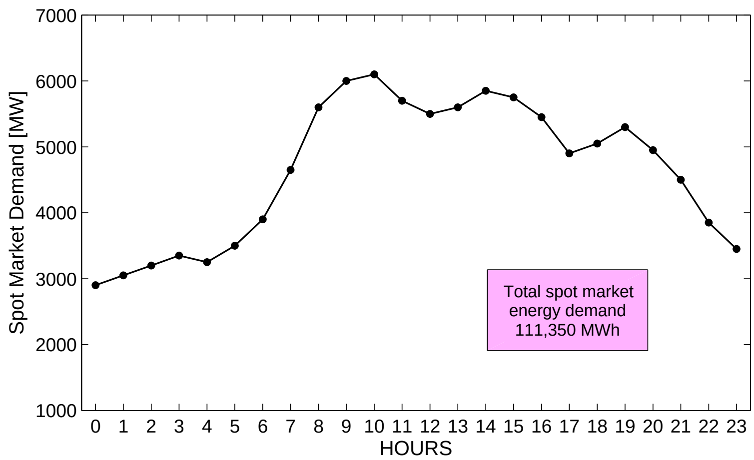



**Figure 4.10:** Spot market demand curve.

Apart from the allocations in the spot market, GENCO A is assumed to have a bilateral load demand equivalent to 10% of of the system spot market demand (Figure 4.10) at a constant price of \$45/MWh. A contract of differences factor ($\kappa$ in equation (3.3)) is set at 0.5. GENCOs B and C are assumed to have no bilateral commitments. The price of reserve power (both spinning and non-spinning) is set at a constant \$4.50/MW.

First, in section 4.3.2 a discussion of the nominal system equilibrium is presented i.e. the market prices, spot market load allocations, and expected GENCO profits (results of the PBUC) if each GENCO were to bid its nominal bid curve as shown in Figure 4.9(a). Then, the solution to the OBS-PBUC problem for each of the three GENCOs using the proposed EPSO algorithm is presented in section 4.3.3. Finally, in section 4.3.4, a comparison of the simulations results using the proposed EPSO algorithm and the classical PSO algorithm is presented.

### 4.3.2 Nominal System Equilibrium

The spot market demand curve plotted in Figure 4.10 shows the load varying from a low of 2900 MW at midnight to a maximum of 6100 MW at 10 am. If all three GENCOs bid their reference marginal cost curves, the hourly market clearing prices (MCPs) will be as shown in Figure 4.11(a). As expected, the

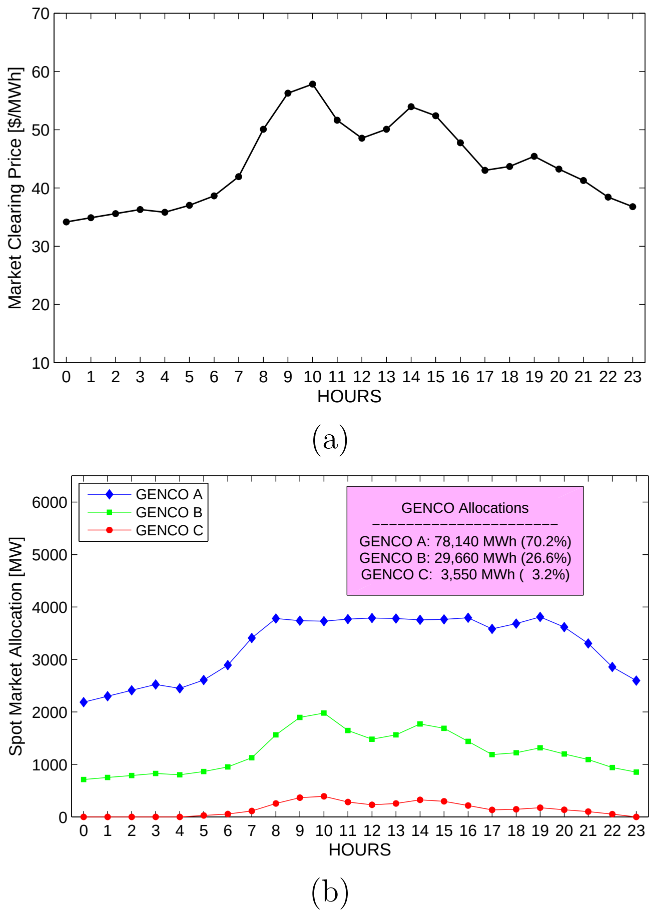



**Figure 4.11:** (a) Hourly MCPs corresponding to the nominal marginal cost curves and (b) spot market allocations for each individual GENCO.

MCP curve follows the spot market load with a minimum value of \$34.17/MWh at midnight and a peak of \$57.82/MWh 10 am. Also shown in Figure 4.11(b) is the hourly allocation of the spot market load to the three GENCOs. Figure 4.11(b) shows that GENCO A gets most of the load and is actually limited during the peak hours of 8 am to 4 pm by its capacity less the bilateral market demand. GENCO B supplies mainly the intermediate load with the allocation of 1978 MW at 10 am almost equaling its capacity of 2140 MW. GENCO C primarily acts as a peak load generator only supplying energy during the peak hours. During the low peak hours before 5 am, GENCO C receives no allocation thus only receiving payments from sale of reserve power.

Also shown in Figure 4.11(b) is the total daily energy to be supplied by the three GENCOs in the spot market. It is observed that GENCO A is allocated 70.2% of the spot market demand during the day while GENCOs B and C are allocated only 26.6% and 3.2% respectively. While GENCO B's allocation compares relatively well to its installed capacity (30% relative to the system capacity) GENCO A's allocation is significantly higher than its relative installed capacity while GENCO C's allocation is significantly lower than its relative installed capacity. Again, this is because GENCO A has cheaper units hence is usually allocated first while GENCO C has more expensive units and is allocated last. A Profit Based Unit Commitment algorithm was run for each of the three GENCOs with the base case spot market allocations shown in Figure 4.11(b) and the the expected values of daily revenue, operating costs, and profits for each GENCO are shown in Table 4.10.

**Table 4.10:** Revenues, costs, and profits for nominal bidding strategies.

| | GENCO A | GENCO B | GENCO C |
|---|---|---|---|
| REVENUES | | | |
| Spot Market | 3,523,184 | 1,369,344 | 177,640 |
| Reserve Sales | 59,434 | 86,817 | 56,838 |
| Bilateral Market | 501,075 | – | – |
| CFDs | 2,922 | – | – |
| TOTAL REVENUE | 4,086,615 | 1,456,161 | 234,479 |
| TOTAL COST | 2,620,007 | 914,826 | 148,268 |
| PROFIT | 1,466,608 | 541,335 | 86,211 |

### 4.3.3 Optimal Bidding Strategy

The OBS-PBUC problem was solved using the EPSO algorithm described in section 3.2.2 for each of the three GENCOs assuming that each is acting independently and using the competing GENCO's nominal marginal cost curves as the expected competitor bidding curves in each case. Values for the various parameters used in the EPSO algorithm are given in Table 4.11. Simulation results including the optimal values of the hourly bid factors, the expected MCPs, spot market allocations, revenues, costs, and profits are discussed next.

**Table 4.11:** Parameter values for EPSO algorithm.

| Parameter | Value |
|---|---|
| no. of particles | 20 |
| no. of iterations | 500 |
| initial value of weight $w^0$ | 0.4 |
| initial value of weight $w^1$ | 1.0 |
| initial value of weight $w^2$ | 2.0 |
| standard deviation of weights, $\tau_{w^l}$ | 0.1 |
| standard deviation of gBest, $\tau_g$ | 0.01 |
| no. of replica particles | 1 |
| probability of best particle surviving, $p_{luck}$ | 0.8 |

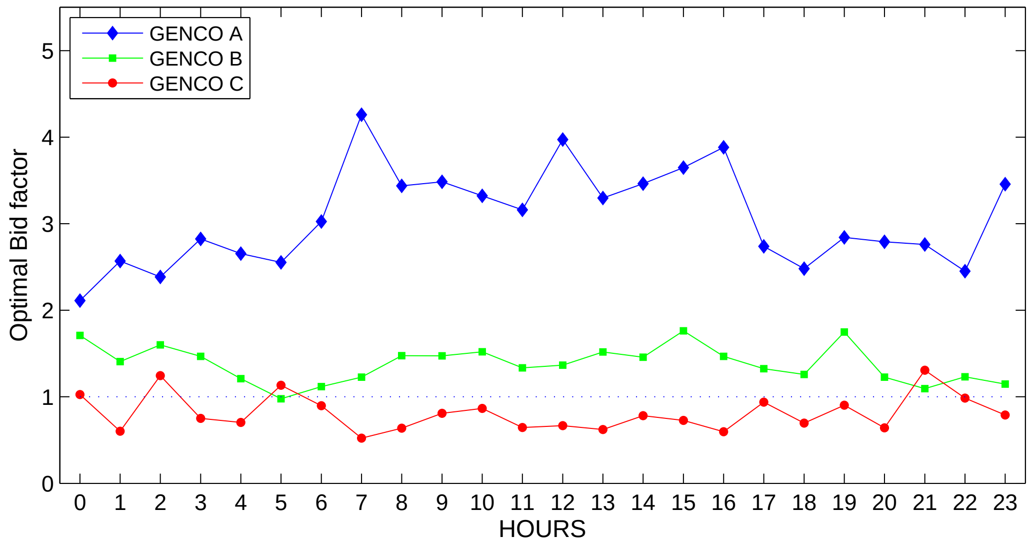



**Figure 4.12:** Optimal hourly bid factors.

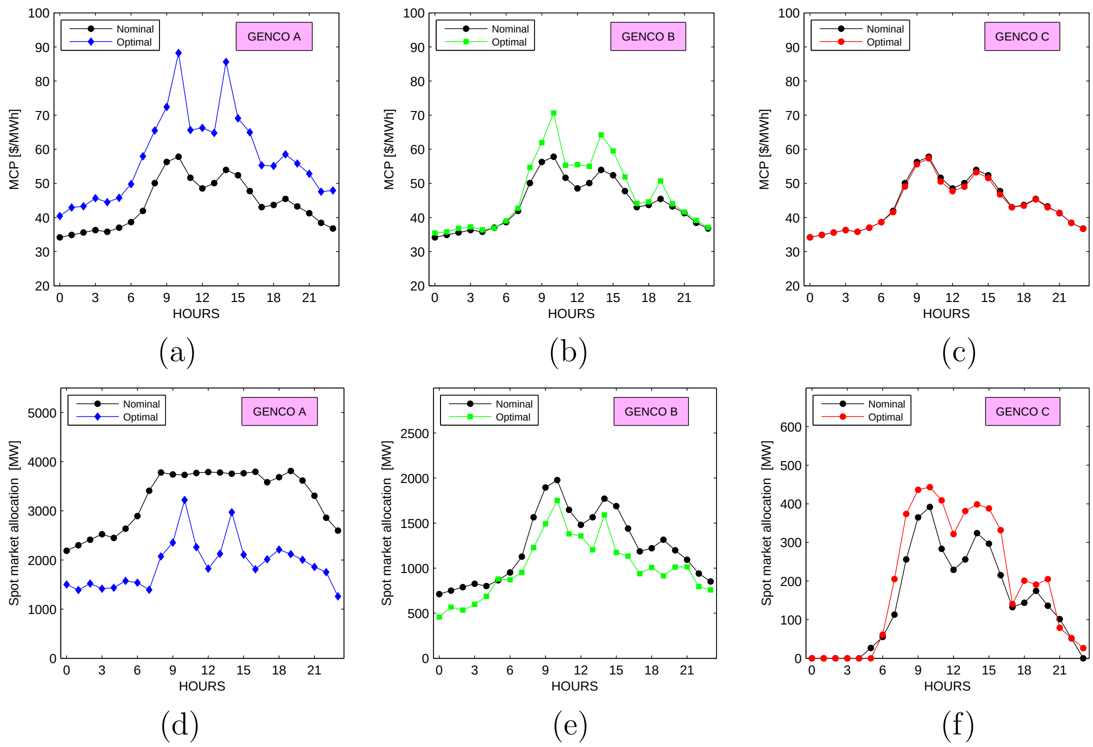



**Figure 4.13:** Effect of individual GENCO optimal bidding strategies on the MCPs and the spot market allocations.

The hourly values of the bid factors that give the maximum profits defines a GENCO's optimal bidding strategy. Figure 4.12 shows the obtained values of the optimal bid strategies for the three GENCOs. As seen from Figure 4.12, the optimal values of the hourly bidding factors are in the range of 2.0 to 4.3 for GENCO A, 0.9 to 1.8 for GENCO B, and 0.5 to 1.3 for GENCO C. The interpretation here is that GENCO A should generally bid higher than its nominal bidding curve, GENCO B should bid just slightly higher than its nominal bidding curve while GENCO C should bid lower than its nominal bidding curve in order to maximize their respective profits.

Figures 4.13 (a), (b), and (c) show how the optimal bidding strategy for each GENCO would alter the hourly spot market clearing prices. GENCO A's high bidding factors significantly raise the MCPs; GENCO B's strategy would slightly raise the MCPs while GENCO C's strategy has almost no effect on the MCPs. These results confirm that GENCO C is in fact a *price taker* while GENCO A is a *price maker* with significant market power to alter the MCPs. The effect of the optimal bidding strategies on the spot market allocations are shown in Figures 4.13 (d), (e), and (f). GENCO A's optimal bidding strategy would significantly reduce its allocation in the spot market; GENCO B's optimal bidding strategy would slightly reduce its allocation in the spot market; while GENCO C's optimal bidding strategy would significantly increase its allocation in the spot market during peak hours.

Comparisons of the revenues, costs, and profits for each of the three GENCO's are given in Tables 4.12, 4.13, and 4.14. Table 4.12 — corresponding to GENCO A — shows a reduction in the revenues from the spot market resulting from the reduced allocations due to the optimal bidding strategy. However the reduction in operating costs coupled with revenues from the increased reserve sales and CFDs

**Table 4.12:** GENCO A revenues, costs, and profits for optimal bidding strategy.

| | Nominal | Optimal | Difference |
|---|---|---|---|
| REVENUES | | | |
| Spot Market | 3,523,184 | 2,768,844 | −754,340 |
| Reserve Sales | 59,434 | 189,213 | +129,779 |
| Bilateral Market | 501,075 | 501,075 | 0 |
| CFDs | 2,922 | 85,562 | +82,640 |
| Total Revenues | 4,086,615 | 3,544,694 | −541,921 |
| Total Costs | 2,620,007 | 1,518,954 | −1,101,053 |
| Profits [$/day] | 1,466,608 | 2,025,739 | +559,131 |
| | | | (+38.1%) |

gives an overall increase in profits. These results show that it is more beneficial for GENCO A to *bid high* to increase the MCPs even though this may reduce its spot market allocations and its spot market revenues as a result.

Table 4.13 — corresponding to GENCO B — shows similar results to those of GENCO A though to a lesser extent. The reduction in operating costs coupled with revenues from the increased reserve sales outweighs reduction in spot market revenues resulting from the reduced allocations due to the optimal bidding strat-

**Table 4.13:** GENCO B revenues, costs, and profits for optimal bidding strategy.

| | Nominal | Optimal | Difference |
|---|---|---|---|
| REVENUES | | | |
| Spot Market | 1,369,344 | 1,221,779 | −147,565 |
| Reserve Sales | 86,817 | 108,219 | +21,402 |
| Bilateral Market | – | – | – |
| CFDs | – | – | – |
| Total Revenues | 1,456,161 | 1,329,998 | −126,163 |
| Total Costs | 914,826 | 716,916 | −197,910 |
| Profits [$/day] | 541,335 | 613,082 | 71,747 |
| | | | (+13.3%) |

egy. However, GENCO B achieves only a 13.3% increase in profits compared to a 38.1% increase for GENCO A. This is a consequence of the GENCO A's market power enabling it to have a greater influence on the MCPs a result that is also deduced from Figures 4.13 (a) and (b).

Table 4.14 — corresponding to GENCO C — shows that for the smallest GENCO the optimal bidding strategy is to *bid low* in order to capture slightly more of the spot market demand. As seen from Table 4.14, the net increase in revenues (increase in spot market revenues less reductions in reserve sales) is greater than the increase in operating costs for supplying more energy. However, the final increase in profits (+3.7%) is relatively small when compared to the profit increases realized by GENCOs A and B. Again this is attributed to GENCO C's relative weakness in the electricity market.

**Table 4.14:** GENCO C revenues, costs, and profits for optimal bidding strategy.

| | Nominal | Optimal | Difference |
|---|---|---|---|
| REVENUES | | | |
| Spot Market | 177,640 | 228,785 | 51,145 |
| Reserve Sales | 56,838 | 52,467 | −4,371 |
| Bilateral Market | – | – | – |
| CFDs | – | – | – |
| Total Revenues | 234,479 | 281,252 | 46,773 |
| Total Costs | 148,268 | 191,856 | 43,588 |
| Profits [$/day] | 86,211 | 89,396 | 3,185 |
| | | | (+3.7%) |

The relationship between the GENCO market power (measured by the GENCO's relative size in the market) and the effect of their optimal bidding strategies on the market dynamics are summarized in Table 4.15. Note that if the three GENCOs had equal market power, GENCO A would have less effect on market price and allocations while GENCO C would have a larger effect on the same. The effect of GENCO B would be somewhat similar to the results given in Table 4.15 since it controls about one third of the market share. The imbalance observed in the percentage change in expected profits would not be observed.

**Table 4.15:** Effect of GENCO market power on market prices, allocations, and individual profits.

| | GENCO A | GENCO B | GENCO C |
|---|---|---|---|
| % installed capacity (market power) | 60% | 30% | 10% |
| Average % change in spot energy market prices | +31.4% | +7.2% | −0.8% |
| Average % change in spot energy market allocations | −41.5% | −18.1% | +30.8% |
| % change in expected profits | +38.1% | +13.3% | +3.7% |

### 4.3.4 Comparison of EPSO and PSO algorithms

The performance of the implemented EPSO algorithm was compared to the performance of a similar implementation using the classical PSO algorithm. The problem was solved for the three GENCOs 20 times using both algorithms starting from the same initial particles in each trial. The same parameter values (i.e. number of particles, number of iterations, and weight values) given in Table 4.16 for the EPSO algorithm were used for the implemented PSO algorithm. The performance of the two algorithms was measured in terms of the objective function values from the 20 runs. Table 4.16 gives the best, average, and worst solutions obtained for both algorithms for each of the three GENCOs. Also given in Table 4.16 is the standard deviation for the solutions. The table shows that the proposed EPSO algorithm outperforms the classical PSO algorithm in all cases. The superior performance of the EPSO algorithm is attributed to the mutation characteristics embedded in the algorithm which tunes the algorithm parameters in subsequent generations hence resulting in a better search of the solution space.

**Table 4.16:** Comparison of solution quality using PSO and EPSO algorithms.

| | GENCO A | | GENCO B | | GENCO C | |
|---|---|---|---|---|---|---|
| | PSO | EPSO | PSO | EPSO | PSO | EPSO |
| Best solution [$/day] | 1,954,488 | 2,025,739 | 610,497 | 613,082 | 88,703 | 89,396 |
| Average solution [$/day] | 1,903,261 | 1,967,363 | 598,663 | 600,745 | 87,987 | 88,982 |
| Worst solution [$/day] | 1,850,798 | 1,931,389 | 584,485 | 585,105 | 87,293 | 88,482 |
| Standard dev. [$/day] | 40,644 | 31,811 | 7,348 | 6,365 | 348 | 262 |

## 4.4 PBUC Solution Tool

The algorithms presented in this thesis are put together as a package in MATLAB software. The package is a collection of MATLAB m-files that can be used to solve the Profit Based Unit Commitment problem using either the PSO or EPSO algorithms. The PSO and EPSO code can, however, be used to solve any other optimization problem as long as the data is presented in the right format. This section gives a discussion of the Toolbox organization and simulation in MATLAB.

### 4.4.1 MATLAB Toolbox Organization

The tool is a set of MATLAB m-files for solving the PBUC problem. Generally, the m-files can be grouped into four categories as shown in Figure 4.14. The application of the four sets of scripts is described in the subsequent sub-sections:

#### 4.4.1.1 PBUC m-files

The m-files under `PBUC` are used to solve the PBUC problem as described in section 3.1.2. Either the PSO or EPSO algorithms is used to update the values


**Figure 4.14:** MATLAB toolbox organization.

of Lagrange multipliers. The algorithm to be applied must be defined in the inputs to the main code. The main code is `PBUC` which is complemented by 9 sub routines. The details of the MATLAB scripts are summarized in Table 4.17.

#### 4.4.1.2 OBS-PBUC m-files

The m-files under `OBS_PBUC` are used to solve the OBS-PBUC problem as described in section 3.2. Again, either the PSO or EPSO algorithms is used to update the values of the optimal bid factors. The algorithm to be applied must be defined in the inputs to the main code which is `OBS_PBUC`. The main code is complemented by 10 sub routines – the details of which are summarized in Table 4.18.

**Table 4.17:** PBUC functions

| Function name | Description |
|---|---|
| PBUC | Runs the main PBUC code. Inputs include the name of the excel file holding the data and the solution algorithm i.e. PSO/EPSO. |
| PBUC_get_data | Fetches the system data and algorithm data from the specified excel sheet. |
| PBUC_initialize | Randomly generates initial values of possible solutions (PSO or EPSO particles). |
| PBUC_marginal_cost_calc | Calculates the marginal cost. Used to give initial approximate values of Lagrange multipliers (see section 3.1.3.2). |
| PBUC_obj_fcn_calc | Calculates the objective function value (particle fitness) for a given possible solution. |
| PBUC_relaxed_problem_soln | Solves the relaxed PBUC problem (see section 3.1.2.1). |
| PBUC_dynamic_program | Implements the hour-by-hour dynamic programming code for solving the relaxed PBUC problem. |
| PBUC_profit_calc | Calculates the profit given a UC and power generation schedule. |
| PBUC_format_results | Formats the results from the PSO/EPSO code. |
| PBUC_store_display_results | Used to store and display results after a successful run of the algorithm. |

#### 4.4.1.3 PSO m-files

The m-files under `PSO` implement the PSO algorithm described in section 2.4.3 to solve a mathematical optimization problem. Any optimization problem set up in the specified format can be solved using the `PSO` m-files. In this research, the optimization problems are the PBUC and OBS-PBUC problems. The main code is PSO and is complemented by 5 sub routines – the details of which are summarized in Table 4.19.

**Table 4.18:** OBS-PBUC functions

| Function name | Description |
|---|---|
| `OBS_PBUC` | Runs the main OBS-PBUC code. Inputs include the name of the excel file holding the data and the solution algorithm i.e. PSO/EPSO. |
| `OBS_PBUC_get_data` | Fetches the system data and algorithm data from the specified excel sheet. |
| `OBS_PBUC_initialize` | Randomly generates initial values of possible solutions (PSO or EPSO particles). |
| `OBS_PBUC_marginal_cost_calc` | Calculates the marginal cost. Used to give initial approximate values of Lagrange multipliers (see section 3.1.3.2). |
| `OBS_PBUC_obj_fcn_calc` | Calculates the objective function value (particle fitness) for a given possible solution. |
| `OBS_PBUC_relaxed_problem_soln` | Solves the relaxed OBS-PBUC problem (see section 3.1.2.1). |
| `OBS_PBUC_dynamic_program` | Implements the hour-by-hour dynamic programming code for solving the relaxed OBS-PBUC problem. |
| `OBS_PBUC_profit_calc` | Calculates the profit given a UC and power generation schedule. |
| `OBS_PBUC_format_results` | Formats the results from the PSO/EPSO code. |
| `OBS_PBUC_store_display_results` | Stores and displays results after a successful run of the algorithm. |

**Table 4.19:** PSO functions

| Function name | Description |
|---|---|
| `PSO` | Runs the PSO algorithm for a given objective function returning the results as a MATLAB struct variable. |
| `PSO_parameters` | Extracts PSO algorithm parameters. |
| `PSO_evaluation` | Evaluates objective function values for a given set of PSO particles. |
| `PSO_pBest_gBest_update` | Updates the `pBest` and `gBest` values after a PSO iteration. |
| `PSO_particle_update` | Updates the PSO particles using the PSO particle update equation. |
| `PSO_display_results` | Displays results after a successful run of the algorithm. |

#### 4.4.1.4 EPSO m-files

The m-files under `EPSO` implement the EPSO algorithm described in section 2.4.4 to solve a mathematical optimization problem. Similar to the PSO set of m-files, any mathematical optimization problem set up in the specified format can be solved using the `EPSO` m-files. In this research, the optimization problems are the PBUC and OBS-PBUC problems. The main code is EPSO and is complemented by 8 sub routines – the details of which are summarized in Table 4.20.

**Table 4.20:** EPSO functions

| Function name | Description |
|---|---|
| `EPSO` | Runs the EPSO algorithm for a given objective function returning the results as a MATLAB struct variable |
| `EPSO_parameters` | Extracts EPSO algorithm parameters. |
| `EPSO_evaluation` | Evaluates objective function values for a given set of EPSO particles |
| `EPSO_pBest_gBest_update` | Updates the `pBest` and `gBest` values after an EPSO iteration |
| `EPSO_replication` | Replicates a set of EPSO particles |
| `EPSO_mutation` | Mutates weights for the EPSO particles |
| `EPSO_reproduction` | Reproduces EPSO particles using the EPSO particle movement rule |
| `EPSO_selection` | Selects EPSO particles surviving to the next generation through a stochastic tournament |
| `EPSO_display_results` | Displays results after a successful run of the algorithm. |

### 4.4.2 Simulation Details

The developed tool is used to solve the PBUC problem with the option of including an Optimal Bidding Strategy in the simulation results as described in section 3.2. A simulation will involve the following three main steps:

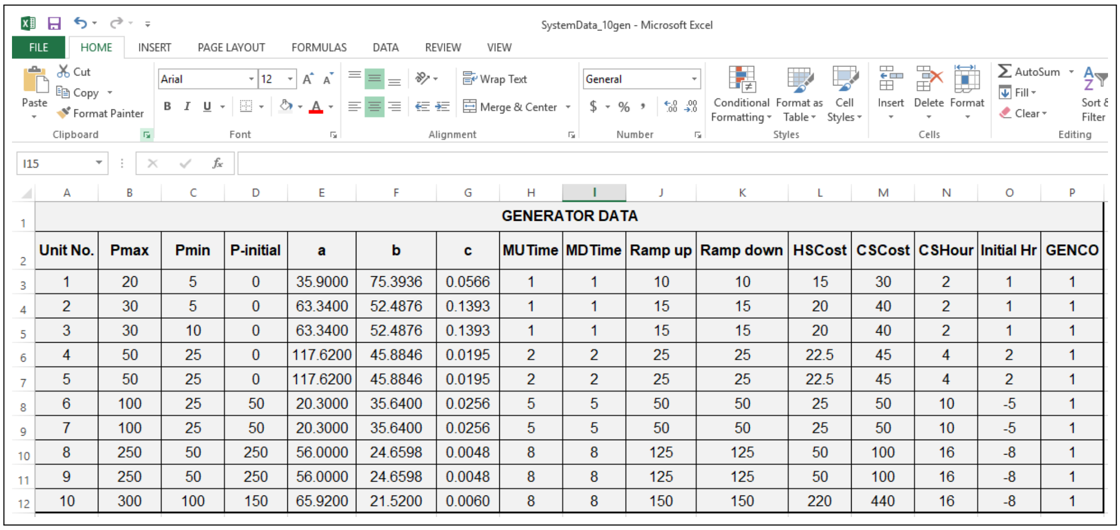
1. Preparing the required data (system data, algorithm parameters, other required options)

2. Running the simulation (using the appropriate MATLAB function)
3. Viewing / storing the results. The results can be viewed on the screen and/or saved in an excel file.

Details of the above three steps are given in the following sub sections.

#### 4.4.2.1 Preparing required data

The required data includes the generator data, price and demand data, other constants, and solution algorithm parameters. The sets of data have to be stored in an excel file (`.xls` or `.xlsx` format) as shown in Figures 4.15 and 4.16.




| GENERATOR DATA | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Unit No. | Pmax | Pmin | P-initial | a | b | c | MUTime | MDTime | Ramp up | Ramp down | HSCost | CSCost | CSHour | Initial Hr | GENCO |
| 1 | 20 | 5 | 0 | 35.9000 | 75.3936 | 0.0566 | 1 | 1 | 10 | 10 | 15 | 30 | 2 | 1 | 1 |
| 2 | 30 | 5 | 0 | 63.3400 | 52.4876 | 0.1393 | 1 | 1 | 15 | 15 | 20 | 40 | 2 | 1 | 1 |
| 3 | 30 | 10 | 0 | 63.3400 | 52.4876 | 0.1393 | 1 | 1 | 15 | 15 | 20 | 40 | 2 | 1 | 1 |
| 4 | 50 | 25 | 0 | 117.6200 | 45.8846 | 0.0195 | 2 | 2 | 25 | 25 | 22.5 | 45 | 4 | 2 | 1 |
| 5 | 50 | 25 | 0 | 117.6200 | 45.8846 | 0.0195 | 2 | 2 | 25 | 25 | 22.5 | 45 | 4 | 2 | 1 |
| 6 | 100 | 25 | 50 | 20.3000 | 35.6400 | 0.0256 | 5 | 5 | 50 | 50 | 25 | 50 | 10 | -5 | 1 |
| 7 | 100 | 25 | 50 | 20.3000 | 35.6400 | 0.0256 | 5 | 5 | 50 | 50 | 25 | 50 | 10 | -5 | 1 |
| 8 | 250 | 50 | 250 | 56.0000 | 24.6598 | 0.0048 | 8 | 8 | 125 | 125 | 50 | 100 | 16 | -8 | 1 |
| 9 | 250 | 50 | 250 | 56.0000 | 24.6598 | 0.0048 | 8 | 8 | 125 | 125 | 50 | 100 | 16 | -8 | 1 |
| 10 | 300 | 100 | 150 | 65.9200 | 21.5200 | 0.0060 | 8 | 8 | 150 | 150 | 220 | 440 | 16 | -8 | 1 |

**Figure 4.15:** Generator data stored in an excel sheet.

### Generator data

The data detailing the generator characteristics are stored in columns `A` to `P` of the excel sheet as shown in Figure 4.15. The data in each column of the excel sheet is as detailed in Table 4.21.

### Price and Demand data

The data detailing the expected system demand, expected hourly prices, bilateral demand, and bilateral prices are stored in columns `R` to `W` of the excel sheet as

**Table 4.21:** Generator data as stored in input excel sheet.

| Column | Data | Details |
|---|---|---|
| A | `Unit number` | a numerical value for identifying each generator |
| B | `Pmax` | maximum generator output in MW |
| C | `Pmin` | minimum generator output in MW |
| D | `P-initial` | initial generator output in MW |
| E | `a` | constant term of the generator cost curve equation in $/h |
| F | `b` | linear term of the generator cost curve equation in $/MWh |
| G | `c` | quadratic term of the generator cost curve equation in $\$/(\text{MWh})^2$ |
| H | `MUT` | generator minimum up time in hrs |
| I | `MDT` | generator minimum down time in hrs |
| J | `Ramp Up` | generator ramp up limit in MW/hr |
| K | `Ramp Down` | generator ramp down limit in MW/hr |
| L | `HSCost` | generator hot start cost coefficient in $/hr |
| M | `CSCost` | generator cold start cost coefficient in $/hr |
| N | `CSHour` | duration of time after which a generator is considered cold in hrs |
| O | `Initial Hr` | duration of time for which a generator has been ON/OFF at the beginning of the simulation in hrs. Important for enforcing the MUT and MDT constraints at the initial hours. |
| P | `GENCO` | a numerical value denoting to which GENCO a generator belongs to. Important in the optimal bidding strategy simulations |

**Table 4.22:** Price and demand data as stored in input excel sheet.

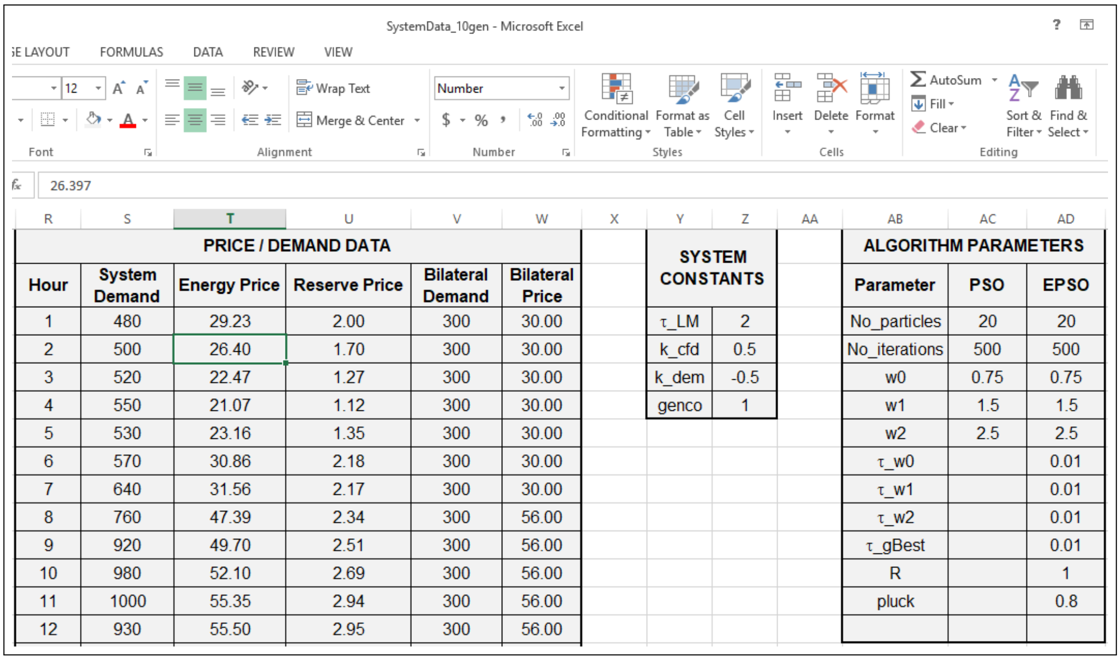
| Column | Data | Details |
|---|---|---|
| R | `Hour` | a numerical value for identifying each hour of the scheduling period |
| S | `System Demand` | expected spot market demand at each hour in MW |
| T | `Energy Price` | expected energy price at each hour in $/MWh |
| U | `Reserve Price` | expected reserve price at each hour in $/MWh |
| V | `Bilateral Demand` | a GENCO's bilateral demand commitment at each hour in MW |
| W | `Bilateral Price` | bilateral contract price at each hour in $/MWh |

**Table 4.23:** Other System Constants as stored in input excel sheet.

| Cell | Data | Details |
|---|---|---|
| Z3 | $\tau$_LM | standard deviation of initial randomly generated values of Lagrange multipliers – see equation (3.25) |
| Z4 | k_cfd | contract of differences factor – see equation (3.3) |
| Z5 | k_dem | gradient of linear demand curve – see equation (4.1) |
| Z6 | genco | a numerical value identifying the GENCO whose simulation is being run |

**Table 4.24:** Algorithm parameters as stored in input excel sheet.

| Cell | Data | Details |
|---|---|---|
| | PSO | |
| AC3 | No_particles | number of PSO particles |
| AC4 | No_iterations | number of PSO iterations |
| AC5 | w0 | PSO weighting factor corresponding to particle inertia |
| AC6 | w1 | PSO weighting factor corresponding to particle pBest |
| AC7 | w2 | PSO weighting factor corresponding to overall gBest |
| | EPSO | |
| AD3 | No_particles | number of EPSO particles |
| AD4 | No_iterations | number of EPSO iterations |
| AD5 | w0 | initial value of EPSO weighting factor corresponding to particle inertia |
| AD6 | w1 | initial value of EPSO weighting factor corresponding to particle pBest |
| AD7 | w2 | initial value of EPSO weighting factor corresponding to overall gBest |
| AD8 | $\tau$_w0 | standard deviation of mutation of weighting factor w0 |
| AD9 | $\tau$_w1 | standard deviation of mutation of weighting factor w1 |
| AD10 | $\tau$_w2 | standard deviation of mutation of weighting factor w2 |
| AD11 | $\tau$_gBest | standard deviation of random disturbance of gBest |
| AD12 | R | number of EPSO particle replications |
| AD13 | pluck | probability of best particle offspring surviving after an iteration |



| PRICE / DEMAND DATA | | | | | |
|---|---|---|---|---|---|
| Hour | System Demand | Energy Price | Reserve Price | Bilateral Demand | Bilateral Price |
| 1 | 480 | 29.23 | 2.00 | 300 | 30.00 |
| 2 | 500 | 26.40 | 1.70 | 300 | 30.00 |
| 3 | 520 | 22.47 | 1.27 | 300 | 30.00 |
| 4 | 550 | 21.07 | 1.12 | 300 | 30.00 |
| 5 | 530 | 23.16 | 1.35 | 300 | 30.00 |
| 6 | 570 | 30.86 | 2.18 | 300 | 30.00 |
| 7 | 640 | 31.56 | 2.17 | 300 | 30.00 |
| 8 | 760 | 47.39 | 2.34 | 300 | 56.00 |
| 9 | 920 | 49.70 | 2.51 | 300 | 56.00 |
| 10 | 980 | 52.10 | 2.69 | 300 | 56.00 |
| 11 | 1000 | 55.35 | 2.94 | 300 | 56.00 |
| 12 | 930 | 55.50 | 2.95 | 300 | 56.00 |

| SYSTEM CONSTANTS | |
|---|---|
| τ_LM | 2 |
| k_cfd | 0.5 |
| k_dem | -0.5 |
| genco | 1 |

| ALGORITHM PARAMETERS | | |
|---|---|---|
| Parameter | PSO | EPSO |
| No_particles | 20 | 20 |
| No_iterations | 500 | 500 |
| w0 | 0.75 | 0.75 |
| w1 | 1.5 | 1.5 |
| w2 | 2.5 | 2.5 |
| τ_w0 | | 0.01 |
| τ_w1 | | 0.01 |
| τ_w2 | | 0.01 |
| τ_gBest | | 0.01 |
| R | | 1 |
| pluck | | 0.8 |

**Figure 4.16:** Price and demand data, system constants, and algorithm parameters stored in an excel sheet.

shown in Figure 4.16. The data in each column of the excel sheet is as described in Table 4.22.

**Other Constants**

Other necessary constants are stored in rows `3` to `5` of column `Z` of the excel sheet as shown in Figure 4.16. The data in each cell is as described in Table 4.23.

**Algorithm parameters**

The data detailing the PSO algorithm parameters are stored in rows `3` to `7` of column `AC` while the data detailing the EPSO algorithm parameters are stored in rows `3` to `13` of column `AD` of the excel sheet as shown in Figure 4.16. The data in each cell is as described in Table 4.24.

#### 4.4.2.2 Running Simulations

A simulation is run by calling one of the main simulation functions i.e. `PBUC` or `OBS_PBUC` with two inputs: `filename` and `algname`. The input `filename` should hold the name of the excel file in which the simulation data is stored and the input `algname` should define the algorithm being used in the simulation i.e. PSO or EPSO. A simulation is therefore run by typing into the MATLAB command window as shown in Fig. 4.17.

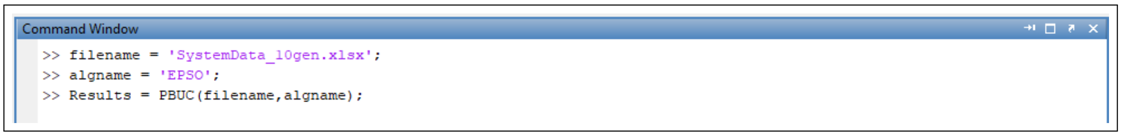



**Figure 4.17:** Running a PBUC simulation in the MATLAB command window.

In Fig. 4.17 a PBUC simulation is run for data in a file: `SystemData_10gen.xlsx` using the `EPSO` algorithm. The results are stored in a MATLAB struct `Results` from which the simulation results can be accessed.

#### 4.4.2.3 Displaying Results

After running a simulation, the results can be stored and displayed by calling the `PBUC_store_display_results` or `OBS_PBUC_store_display_results` functions as shown in Fig. 4.18. The input to the function is the result struct obtained from the simulation shown in Fig. 4.17.

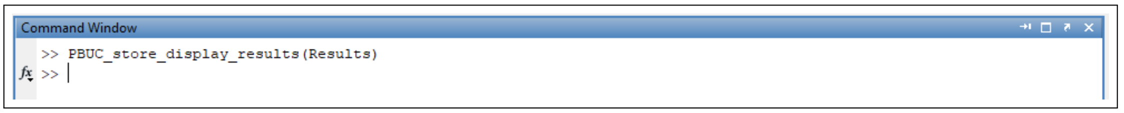



**Figure 4.18:** Storing and displaying the results of a simulation.

While the results can be displayed explicitly as shown in Fig. 4.18, the main codes `PBUC` and `OBS_PBUC` run the display results algorithm as an internal subroutine. Figures 4.19 and 4.20 show a typical output on the computer screen for a solution of the `PBUC` problem.

```
Command Window

Profit Based Unit Commitment (PBUC) problem solution
PBUC problem solution for a GENCO with 10 generating units

==================================================================

Running Evolutionary Particle Swarm Optimization (EPSO) Algorithm!

Performing Iterations:...  010  020  030  040  050  060  070  080  090  100
Performing Iterations:...  110  120  130  140  150  160  170  180  190  200
Performing Iterations:...  210  220  230  240  250  260  270  280  290  300
Performing Iterations:...  310  320  330  340  350  360  370  380  390  400
Performing Iterations:...  410  420  430  440  450  460  470  480  490  500

EPSO algorithm done!!

==================================================================

PBUC solution using EPSO algorithm
Optimal solution obtained in 2 minutes 60 sec
Objective function value ($/day) =   447,416

=========================================================

SOLUTION SUMMARY
=========================================================

REVENUE
---------------------------------
Spot Market:              739,497
Reserve Sales:             11,594
Bilateral Market:         317,400
Contract of Diffs:          3,142
---------------------------------
TOTAL REVENUE:          1,071,633

COSTS
---------------------------------
Fuel Cost:                624,082
Start Up Cost:                135
---------------------------------
TOTAL COST:               624,217

---------------------------------
PROFIT:                   447,416

=========================================================

OPTIMAL LAGRANGE MULTIPLIERS
---------------------------------
Hours 1 to 12
------------------------------------------------------------------------------------------
HOUR:     01     02     03     04     05     06     07     08     09     10     11     12
-----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----
LM:      0.0    0.0    4.1    4.5    2.3    0.0    0.1    0.0    0.0    0.0    0.0    1.3
-----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----


Hours 13 to 24
------------------------------------------------------------------------------------------
HOUR:     13     14     15     16     17     18     19     20     21     22     23     24
-----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----
LM:      0.0    1.1    2.0    0.0    0.6    1.3    0.0    1.1    0.0    2.2    0.0    0.0
-----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----   ----
```

**Figure 4.19:** PBUC Solution results printed on the screen - page 1.

OPTIMAL UNIT COMMITMENT

Hours 1 to 12

| UNIT No. | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Hours 13 to 24

| UNIT No. | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

OPTIMAL GENERATION SCHEDULE

Hours 1 to 12

| UNIT No. | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 50 | 50 | 50 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 50 | 50 | 50 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 100 | 100 | 100 | 100 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 100 | 100 | 100 | 100 |
| 8 | 250 | 125 | 66 | 50 | 50 | 175 | 250 | 250 | 250 | 250 | 250 | 250 |
| 9 | 250 | 125 | 66 | 50 | 50 | 175 | 250 | 250 | 250 | 250 | 250 | 250 |
| 10 | 300 | 265 | 300 | 243 | 219 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |

Hours 13 to 24

| UNIT No. | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 15 | 30 | 30 | 30 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 10 | 25 | 30 | 30 | 30 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 0 | 0 | 0 | 0 |
| 5 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 0 | 0 | 0 | 0 |
| 6 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 88 | 0 | 0 |
| 7 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 88 | 0 | 0 |
| 8 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 |
| 9 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 |
| 10 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |

Results stored in Results_SystemData_10gen.xlsx

>>

**Figure 4.20:** PBUC Solution results printed on the screen - page 2.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

1. **PBUC solution using LR-PSO**

   An algorithm that combines the Lagrangian relaxation technique with the heuristic particle swarm optimization techniques is used to solve the profit based unity commitment problem for GENCOs in deregulated electricity markets. The problem is formulated to include a constraint setting the minimum GENCO output at a given hour as the bilaterally committed generation for the hour. Based on an implementation for a GENCO with 54 thermal units, the research concluded that:

   - The proposed methodology is effective in solving the PBUC problem.
   - The performance of the methodology in the solution of the PBUC problem strongly depends on the PSO parameters. In particular, an assessment of various combinations of PSO parameters results in the choice of $w_1 = 0.75, w_2 = 1.5$, and $w_3 = 0.25$ as best weight parameters for the test system.

     Because the choice of PSO parameters has a significant effect on the solution quality, the EPSO algorithm was explored since research showed that EPSO has better performance.

2. **PBUC solution using LR-EPSO**

   An algorithm based on the evolutionary particle swarm optimization technique is used to solve the profit based unit commitment problem for GENCOs in deregulated markets. The algorithm was implemented for a GENCO with 10 thermal units and an analysis of the results led to the following conclusions:

- The proposed LR-EPSO methodology has a better performance than the classic PSO algorithm both in terms of solution quality and convergence characteristics.
- However, the profitability of the GENCO depends on its market power. Hence, the algorithm was extended not only to determine the optimal unit commitment schedule, but to also establish an optimal bidding strategy that depends on the GENCO market power.

**3. OBS-PBUC solution using LR-EPSO considering GENCO market power**

An algorithm based on the evolutionary particle swarm optimization technique is used to determine an optimal bidding strategy and unit commitment for a GENCO operating in a competitive electricity market. The procedure determines how a GENCO should structure its hourly bid curves so as to maximize expected profits from both the spot and bilateral energy markets. Numerical results from simulations of a typical system with three GENCOs of different sizes led to the following conclusions:

- GENCO market power has a significant effect on the optimal solution of the PBUC problem.
  - Generally, large GENCOs with significant market power could raise their bids thereby significantly raising electricity prices and hence increasing their profits, though the higher bids could reduce their allocations in the spot market.
  - On the other hand, smaller GENCOs with less market power are generally price takers and have relatively less influence on the

market equilibrium. The optimization algorithm results in only slight increases in expected profits for these GENCOs.

- The proposed EPSO algorithm gives better solutions (in terms of the final optimal values) when compared to the classic PSO algorithm.

4. **PBUC and OBS-PBUC solution Tool in MATLAB**
   - The MATLAB scripts used to solve the PBUC and OBS-PBUC problems are aggregated in a MATLAB toolbox. The toolbox has four sets of MATLAB m-files under the titles: PBUC, OBS-PBUC, PSO, and EPSO.
   - The toolbox can be used to solve the optimization problems with data presented in a specified format in an excel sheet.

## 5.2 Recommendations

Some of the feasible advancements and further research work are suggested below:

1. The only ancillary service provided by the GENCO that is considered in this study is spinning reserve. The researched could be advanced by providing a solution of the PBUC problem considering other ancillary services such as reactive power and voltage support; active power loss compensation; and load following.
2. The problem formulation could be also be extended by considering the effect of actions by the system regulator including price caps and the enforcement of renewable energy targets. This line of research could reveal the strategies that GENCOs could adopt to take advantage of the renewable energy market which should form a larger part of electricity supply in the future.
3. This study provides solution algorithms based on the PSO and EPSO algo-

rithms. Other heuristic based optimization methods such as Artificial Neural Networks and Fuzzy Optimization could be explored especially when considering factors such as fuel prices and uncertainty in renewable energy sources.

## REFERENCES


[1] D. Kirschen and G. Strbac, *Fundamentals of Power System Economics*. Chichester, England: John Wiley and Sons, 2004.

[2] A. Wood, B. Wollenberg, and G. Sheble, *Power Generation Operation and Control*. New Jersey: John Wiley and Sons, 3rd ed., 2013.

[3] M. Shahidehpour and M. Alomoush, *Restructured Electrical Power Systems, Operation, Trading, and Volatility*. New York: Marcel Decker, 1st ed., 2000.

[4] L. Philipson and H. L. Willis, *Understanding Electric Utilities and De-Regulation*. Florida, USA: Taylor & Francis Group, 2nd ed., 2006.

[5] T. Li and M. Shahidehpour, "Price-based unit commitment: a case of lagrangian relaxation versus mixed integer programming," *Power Systems, IEEE Transactions on*, vol. 20, pp. 2015–2025, Nov 2005.

[6] A. Rajan, C. Christober, P. Sundarajan, V. Jamuna, R. Madhusubash, and B. Udayakumar, "Multi-area unit commitment in deregulated electricity market using DP approach," *Inter. Journal on Recent Trends in Engineering and Technology*, vol. 3, pp. 210–213, May 2010.

[7] J. Raglend, C. Raghuveer, G. R. Avinash, N. Padhy, and D. Kothari, "Solution to profit based unit commitment problem using particle swarm optimization," *Applied Soft Computing*, vol. 10, no. 4, pp. 1247–1256, Sept. 2010.

[8] H. S. Madraswala and A. S. Deshpande, "Genetic algorithm solution to unit commitment problem," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–6, July 2016.

[9] S. Selvi, M. Moses, and C. Rajan, "LR-EP approach for solving profit based unit commitment problem with losses in deregulated markets," *Przeglad Elektrotechniczny*, vol. 11, pp. 210–213, 2013.

[10] K. Lakshmi and S. Vasantharathna, "Hybrid artificial immune system approach for profit based unit commitment problem," *J. Electrical Eng. Technology*, vol. 8, no. 1, pp. 742–751, March 2013.

[11] A. Sudhakar, C. Karri, and A. J. Laxmi, "Profit based unit commitment for GENCOs using lagrange relaxation - differential evolution," *Engineering Science and Technology, an International Journal*, pp. –, 2016.

[12] B. Saravanan, S. Das, S. Sikri, and D. P. Kothari, "A solution to the unit commitment problem — a review," *Frontiers in Energy*, vol. 7, no. 2, pp. 223–236, 2013.

[13] A. K. Bikeri, C. M. Muriithi, and P. K. Kihato, "A review of unit commitment in deregulated electricity markets," in *Scientific Research and Innovation (SRI) Conference, 2015*, pp. 1–5, May 2015.

[14] N. Padhy, "Unit commitment problem under deregulated environment - A review," in *Power Engineering Society General Meeting, 2003, IEEE*, vol. 2, pp. –1094 Vol. 2, July 2003.

[15] M. M. Belsnes, G. Warland, and O. Wolfgang, "Simulating equilibrium prices in oligopoly power markets," in *2011 IEEE Trondheim PowerTech*, pp. 1–8, June 2011.

[16] C. Baloi, J. Belward, and M. Bulmer, "Genetic Unit commitment model in a deregulated power energy environment," in *International congress on modelling and simulation, Townsville, Australia*, pp. 1078–1083, 2003.

[17] C. Richter Jr. and G. Sheble, "A profit-based unit commitment GA for the competitive environment," *Power Systems, IEEE Transactions on*, vol. 15, pp. 715–721, May 2000.

[18] Y. Xiaohui, Y. Yanbin, W. Cheng, and Z. Xiaopan, "An Improved PSO Approach for Profit-based Unit Commitment in Electricity Market," in *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, pp. 1–4, 2005.

[19] C. Columbus and S. Simon, "Profit based unit commitment for GENCOs using parallel PSO in a distributed cluster," *ACEEE Int. J. Electrical and Power Engineering*, vol. 2, no. 3, Nov. 2011.

[20] K. Chandram, N. Subrahmanyam, and M. Sydulu, "New approach with Muller method for Profit Based Unit Commitment," in *Electric Utility Deregulation and Restructuring and Power Technologies, Nanjuing, China*, pp. 952–957, 2008.

[21] T. Venkatesana and M. Sanavullahb, "SFLA approach to solve PBUC problem with emission limitation," *International Journal of Electrical Power and Energy Systems*, vol. 46, pp. 1–9, March 2013.

[22] M. Ghadi and A. Baghramian, "A new heuristic method for solving unit commitment problem in competitive environment," *International Journal of Soft Computing and Engineering*, vol. 2, no. 6, pp. 2231–2307, Jan. 2013.

[23] H. Yamin and S. Shahidehpour, "Unit commitment using a hybrid model between lagrangian relaxation and genetic algorithm in competitive electricity markets," *Electric Power Systems Research*, vol. 68, no. 2, pp. 83 – 92, 2004.

[24] D. Dimitroulas and P. Georgilakis, “A new memetic algorithm approach for the price based unit commitment problem,” *Applied Energy*, vol. 88, no. 12, pp. 4687 – 4699, 2011.

[25] R. Muthu and S. Periasamy, “Profit based self scheduling of the genco’s by using particle swarm optimization,” *UPB Scientific Bulletin, Series C: Electrical Engineering*, vol. 74, no. 2, 2012.

[26] K. Asokan and R. Ashokkumar, “Emission Controlled Profit Based Unit Commitment for GENCOs using MPPD Table with ABC Algorithm under Competitive Environment,” *WSEAS Transactions on Systems*, vol. 13, pp. 523–542, 2014.

[27] D. Nahomi and V. Yuvaraju, “A new approach for profit-based unit commitment using lagrangian relaxation combined with particle swarm optimization algorithm,” *Inter. J. Communications and Engineering*, vol. 4, no. 4, pp. 159–166, Mar. 2012.

[28] X. Wang, Y. Song, and M. Irving, *Modern Power Systems Analysis*. Springer, 2008.

[29] K. Bhattacharya, M. Bollen, and J. Daalder, *Operation of Restructured Power Systems*. New York: Springer Science+Business Media, 2001.

[30] N. Padhy, “Unit commitment problem under deregulated environment - A review,” in *Power Engineering Society General Meeting, 2003, IEEE*, vol. 2, pp. 11088–1094, July 2003.

[31] N. Padhy, “Unit commitment-a bibliographical survey,” *Power Systems, IEEE Transactions on*, vol. 19, pp. 1196–1205, May 2004.

[32] W. L. Snyder, H. D. Powell, and J. C. Rayburn, "Dynamic programming approach to unit commitment," *Power Systems, IEEE Transactions on*, vol. 2, pp. 339–348, May 1987.

[33] S. Virmani, E. Adrian, K. Imhof, and S. Mukherjee, "Implementation of a Lagrangian Relaxation based unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 4, pp. 1373–1380, Nov 1989.

[34] P. Lowery, "Generating unit commitment by dynamic programming," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-85, pp. 422–426, May 1996.

[35] C.P. Cheng and C.W. Liu and C.C. Liu, "Unit commitment by Lagrangian relaxation and genetic algorithms," *Power Systems, IEEE Transactions on*, vol. 15, pp. 707–714, May 2000.

[36] M. Carrion and J. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 21, pp. 1371–1378, Aug 2006.

[37] E. Delarue, P. V. D. Bosch, and W. D'haeseleer, "Effect of the accuracy of price forecasting on profit in a price based unit commitment," *Electric Power Systems Research*, vol. 80, no. 10, pp. 1306 – 1313, 2010.

[38] E. Delarue, D. Cattrysse, and W. D'haeseleer, "Enhanced priority list unit commitment method for power systems with a high share of renewables," *Electric Power Systems Research*, vol. 105, no. 0, pp. 115 – 123, 2013.

[39] I. Damousis, A. Bakirtzis, and P. Dokopoulos, "A solution to the unit-commitment problem using integer-coded genetic algorithm," *Power Systems, IEEE Transactions on*, vol. 19, pp. 1165–1172, May 2004.

[40] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, “A hybrid LR-EP for solving new profit-based uc problem under competitive environment,” *IEEE Trans. Power Systems*, vol. 18, no. 1, pp. 229–237, Feb. 2003.

[41] R. H. Kwon and D. Frances, *Handbook of Networks in Power Systems I*, ch. Optimization-based bidding in day-ahead electricity auction markets: A review of models for power producers, pp. 41–59. Springer Berlin Heidelberg, 2012.

[42] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[43] V. Miranda and N. Fonseca, “EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems,” in *IEEE/PES Transmission and Distribution Conference and Exhibition*, vol. 2, pp. 745–750 vol.2, Oct 2002.

[44] X. Ma, D. I. Sun, and K. W. Cheung, “Evolution toward standardized market design,” *IEEE Transactions on Power Systems*, vol. 18, pp. 460–469, May 2003.

[45] S. Gao, *Bio-Inspired Computational Algorithms and Their Applications.* InTech, 2012.

[46] University of Washington, “Power system test case archive,” October 2017.

[47] Illinois Institute of Technology, “PBUC Data,” October 2017.

# APPENDICES

## Appendix 1: PBUC Code

```
function Results = PBUC(filename,alg)
% PBUC profit based unit commitment solution.
%
% Solves the PBUC problem for a test system whose data is stored in an
%   input filename using the PSO/EPSO algorithms returning the results as
%   a MATLAB struct variable
%
% Inputs
%   filename : a MATLAB string specifying the name of the excel file in
%      the system data is stored (must have the extension .xls or .xlsx).
%   alg : a MATLAB string specifying the optimization algorithm to be used
%      in solving the problem (options are 'PSO' and 'EPSO').
%
% Output
%   Results : solution results. This is a MATLAB struct with the fields:
%       et : problem solution time in seconds
%       UC : optimal unit commitment schedule
%       LM : optimal values of Lagrange multipliers
%       P_sched : optimal power generation schedule
%       PF : total profit for scheduling period
%       RV : total revenue for scheduling period
%       TC : total costs for scheduling period
%       RV_m : total revenue from spot energy market
%       RV_b : total revenue from bilateral energy market
%       RV_r : total revenue from power reserve sales
%       RV_cfd : total revenue from contracts of differences
%       RV_hr : hourly total revenue
%       RV_m_hr : hourly revenue from spot energy market
%       RV_b_hr : hourly revenue from bilateral energy market
%       RV_r_hr : hourly revenue from power reserve sales
%       RV_cfd_hr : hourly revenue from contracts of differences
%       TC_hr : hourly total costs
%       FC : hourly fuel costs for each generating unit
%       SUC : hourly startup costs for each generating unit
%       TC_gen_hr : hourly total costs for each generating unit
%       ENS_hr : hourly energy not served
%       ENS : total energy not served
%       gBest_final : final global best solution from PSO/EPSO algorithm
%           (same as RV)
%       f_gBest_final : final values of the PSO/EPSO objective function
%           (same as LM)
%       gBest : gBest values after each PSO/EPSO iteration
%       pBest : pBest values after each PSO/EPSO iteration
%       f_gBest : values of the objective function corresponding to the
%           gBest values after each PSO/EPSO iteration
%       f_pBest : values of the objective function corresponding to the
%           pBest values after each PSO/EPSO iteration

%%
tic
fprintf('\n Profit Based Unit Commitment (PBUC) problem solution  \n')
```

```
%% Get data: system data and PSO/EPSO parameters
[syst_data,alg_data] = PBUC_getdata(filename,alg);

fprintf(' PBUC problem solution for a GENCO with %d generating units\n'...
    ,size(syst_data.gen,1))
fprintf(' \n ================================================================== \n')

%% Initialize particles (possible solutions) and velocities
[X0,V0] = PBUC_initialize(syst_data,alg_data);

%% run PSO/EPSO algorithm
obj_fcn = @(x)PBUC_obj_fcn_calc(x,syst_data);
xmin = zeros(length(X0(:,1)),1);
xmax = 10^5*ones(length(X0(:,1)),1);

alg = upper(alg);
if strcmp(alg,'PSO')
    param = struct('n_iter',alg_data(2),...
                   'w0',alg_data(3),...
                   'w1',alg_data(4),...
                   'w2',alg_data(5));

    Results_alg = PSO(obj_fcn,X0,V0,param,xmin,xmax);
    Results_alg.alg = 'PSO';
elseif strcmp(alg,'EPSO')
    param = struct('n_iter',alg_data(2),...
                   'w0',alg_data(3),...
                   'w1',alg_data(4),...
                   'w2',alg_data(5),...
                   't_w0',alg_data(6),...
                   't_w1',alg_data(7),...
                   't_w2',alg_data(8),...
                   't_g',alg_data(9),...
                   'r',alg_data(10),...
                   'pluck',alg_data(11));

    Results_alg = EPSO(obj_fcn,X0,V0,param,xmin,xmax);
    Results_alg.alg = 'EPSO';
else
    error('Please specify solution algorithm as PSO or EPSO!')
end
fprintf(' \n ================================================================== \n')

Results_alg.et = toc;
%% format and display results
Results = PBUC_format_results(Results_alg,syst_data);
PBUC_store_display_results(Results,filename)
```

# Appendix 2: OBS-PBUC Code

```
function Results = OBS_PBUC(filename,alg)
% OBS_PBUC optimal bidding strategy / profit based unit commitment solution
%
% Solves the OBS-PBUC problem for a test system whose data is stored in an
%   input filename using the PSO/EPSO algorithms returning the results as
%   a MATLAB struct variable
%
% Inputs
%   filename : a MATLAB string specifying the name of the excel file in
%      the system data is stored (must have the extension .xls or .xlsx).
%   alg : a MATLAB string specifying the optimization algorithm to be used
%      in solving the problem (options are 'PSO' and 'EPSO').
%
% Output
%   Results : solution results. This is a MATLAB struct with the fields:
%       et : problem solution time in seconds
%       UC : optimal unit commitment schedule
%       BF : optimal values of hourly bid factors
%       P_sched : optimal power generation schedule
%       PF : expected total profit for scheduling period
%       RV : expected total revenue for scheduling period
%       TC : expected total costs for scheduling period
%       RV_m : expected total revenue from spot energy market
%       RV_b : total revenue from bilateral energy market
%       RV_r : expected total revenue from power reserve sales
%       RV_cfd : expected total revenue from contracts of differences
%       RV_hr : expected hourly total revenue
%       RV_m_hr : expected hourly revenue from spot energy market
%       RV_b_hr : hourly revenue from bilateral energy market
%       RV_r_hr : expected hourly revenue from power reserve sales
%       RV_cfd_hr : expected hourly revenue from contracts of differences
%       TC_hr : expected hourly total costs
%       FC : expected hourly fuel costs for each generating unit
%       SUC : expected hourly startup costs for each generating unit
%       TC_gen_hr : expected hourly total costs for each generating unit
%       ENS_hr : expected hourly energy not served (bilateral load not met)
%       ENS : expected total energy not served (bilateral load not met)
%       gBest_final : final global best solution from PSO/EPSO algorithm
%           (same as RV)
%       f_gBest_final : final values of the PSO/EPSO objective function
%           (same as BF)
%       gBest : gBest values after each PSO/EPSO iteration
%       pBest : pBest values after each PSO/EPSO iteration
%       f_gBest : values of the objective function corresponding to the
%           gBest values after each PSO/EPSO iteration
%       f_pBest : values of the objective function corresponding to the
%           pBest values after each PSO/EPSO iteration

%%
tic
fprintf('\n Optimal Bidding Strategy - Profit Based Unit Commitment')
```

```
fprintf('(OBS-PBUC) problem solution  \n')

%% Get data: system data and PSO/EPSO parameters
[syst_data,alg_data] = OBS_PBUC_getdata(filename,alg);

fprintf(' OBS-PBUC problem solution for a GENCO with %d generating units\n'...
    ,size(syst_data.gen,1))
fprintf(' \n ==================================================================== \n')

%% Initialize particles (possible solutions) and velocities
[X0,V0] = OBS_PBUC_initialize(syst_data,alg_data);

%% run PSO/EPSO algorithm
obj_fcn = @(x)OBS_PBUC_obj_fcn_calc(x,syst_data);
xmin = zeros(length(X0(:,1)),1);
xmax = 10*ones(length(X0(:,1)),1);

alg = upper(alg);
if strcmp(alg,'PSO')
    param = struct('n_iter',alg_data(2),...
                   'w0',alg_data(3),...
                   'w1',alg_data(4),...
                   'w2',alg_data(5));

    Results_alg = PSO(obj_fcn,X0,V0,param,xmin,xmax);
    Results_alg.alg = 'PSO';
elseif strcmp(alg,'EPSO')
    param = struct('n_iter',alg_data(2),...
                   'w0',alg_data(3),...
                   'w1',alg_data(4),...
                   'w2',alg_data(5),...
                   't_w0',alg_data(6),...
                   't_w1',alg_data(7),...
                   't_w2',alg_data(8),...
                   't_g',alg_data(9),...
                   'r',alg_data(10),...
                   'pluck',alg_data(11));

    Results_alg = EPSO(obj_fcn,X0,V0,param,xmin,xmax);
    Results_alg.alg = 'EPSO';
else
    error('Please specify solution algorithm as PSO or EPSO!')
end
fprintf(' \n ==================================================================== \n')

Results_alg.et = toc;
%% format and display results
Results = OBS_PBUC_format_results(Results_alg);
OBS_PBUC_store_display_results(Results,filename,alg)
```

# Appendix 3: PSO Code

```
function Results = PSO(obj_fcn,X0,V0,param,xmin,xmax)

% PSO particle swarm optimization algorithm.
%
% Runs the PSO algorithm for a given objective function returning the
%   results as a MATLAB struct variable
%
% Inputs
%   obj_fcn : a handle to the function that evaluates the objective
%       function.
%   X0 : initial values of PSO particles (possible solutions)
%   V0 : initial values of the velocities of PSO particles
%   param : PSO parameters. This is a MATLAB struct with the fields:
%       n_iter : number of iterations;
%       w0 : weighting factor corresponding to the inertia habit;
%       w1 : weighting factor corresponding to the memory habit;
%       w2 : weighting factor corresponding to the cooperation habit;
%   xmin : minimum allowed values of the components of the PSO particles
%   xmax : maximum allowed values of the components of the PSO particles
%
% Outputs
%   Results : algorithm results. This is a MATLAB struct with the fields:
%       gBest_final : final global best solution after all iterations are
%           done. This is the identified optimal solution
%       f_gBest_final : final values of the objective function after all
%           iterations are done. This is the identified optimal solution
%           value
%       gBest : gBest values after each iteration (holds all gBest values)
%       pBest : pBest values after each iteration (holds all pBest values)
%       f_gBest : values of the objective function corresponding to the
%           gBest values after each iteration
%       f_pBest : values of the objective function corresponding to the
%           pBest values after each iteration

%%
fprintf('\n Running Particle Swarm Optimization (PSO) Algorithm! \n')

%% extract model parameters
[n_iter,w0,w1,w2,xmin,xmax] = PSO_parameters(X0,param,xmin,xmax);

%% determine initial pBest and gBest
f_X0 = PSO_evaluation(X0,obj_fcn);
[pBest,f_pBest,gBest] = PSO_pBest_gBest_update(X0,f_X0);

%%
fprintf('\n Performing Iterations:... ')

%%
X = X0; V = V0;

%% PSO iterations
```

```
for iter = 1:n_iter
    if rem(iter,10) == 0
        fprintf(' %03d ',iter)
    end
    if rem(iter,100) == 0 && iter ~= n_iter
        fprintf('\n\n Performing Iterations:... ')
    end

    %% PARTICLE UPDATE
    [X,V] = PSO_particle_update(X,V,pBest,gBest,w0,w1,w2,xmin,xmax);


    %% EVALUATION
    f_X = PSO_evaluation(X,obj_fcn);

    %% Update Pbest & Gbest
    [pBest,f_pBest,gBest,f_gBest] = PSO_pBest_gBest_update(X,f_X,pBest,f_pBest);

    %% Store results for current iteration
    Results.pBest(:,:,iter) = pBest;
    Results.gBest(:,iter)   = gBest;
    Results.f_pBest(iter,:) = f_pBest;
    Results.f_gBest(iter,:) = f_gBest;
end

%% Store final results
Results.gBest_final   = gBest;
Results.f_gBest_final = f_gBest;

%%
fprintf('\n\n PSO algorithm done!!\n\n')
```

# Appendix 4: EPSO Code

```
function Results = EPSO(obj_fcn,X0,V0,param,xmin,xmax)

% EPSO evolutionary particle swarm optimization algorithm.
%
% Runs the EPSO algorithm for a given objective function returning the
%   results as a MATLAB struct variable
%
% Inputs
%   obj_fcn : a handle to the function that evaluates the objective
%       function.
%   X0 : initial values of EPSO particles (possible solutions)
%   V0 : initial values of the velocities of EPSO particles
%   param : EPSO parameters. This is a MATLAB struct with the fields:
%       n_iter : number of iterations;
%       w0 : initial values of the weighting factors corresponding to the
%           inertia habit;
%       w1 : initial values of the weighting factors corresponding to the
%           memory habit;
%       w2 : initial values of the weighting factors corresponding to the
%           cooperation habit;
%       t_w0 : standard deviation of the random mutation of weight w0;
%       t_w1 : standard deviation of the random mutation of weight w1;
%       t_w2 : standard deviation of the random mutation of weight w2;
%       t_g : standard deviation of the random mutation of the gBest value;
%       r : number of replications of an EPSO particle;
%       pluck : probability of the best offspring of a particle surviving
%           after an iteration;
%   xmin : minimum allowed values of the components of the EPSO particles
%   xmax : maximum allowed values of the components of the EPSO particles
%
% Outputs
%   Results : algorithm results. This is a MATLAB struct with the fields:
%       gBest_final : final global best solution after all iterations are
%           done. This is the identified optimal solution
%       f_gBest_final : final values of the objective function after all
%           iterations are done. This is the identified optimal solution
%           value
%       gBest : gBest values after each iteration (holds all gBest values)
%       pBest : pBest values after each iteration (holds all pBest values)
%       f_gBest : values of the objective function corresponding to the
%           gBest values after each iteration
%       f_pBest : values of the objective function corresponding to the
%           pBest values after each iteration

%%
fprintf('\n Running Evolutionary Particle Swarm Optimization (EPSO) Algorithm! \n')

%% extract model parameters
[n_iter,w0,w1,w2,t_w0,t_w1,t_w2,t_g,r,pluck,xmin,xmax] = ...
    EPSO_parameters(X0,param,xmin,xmax);
```

```
%% determine initial pBest and gBest
f_X0 = EPSO_evaluation(X0,obj_fcn);
[pBest,f_pBest,gBest] = EPSO_pBest_gBest_update(X0,f_X0);

%%
fprintf('\n Performing Iterations:... ')

%%
X = X0; V = V0;
w0 = w0*ones(size(X(1,:)));
w1 = w1*ones(size(X(1,:)));
w2 = w2*ones(size(X(1,:)));

%% EPSO iterations
for iter = 1:n_iter
    if rem(iter,10) == 0
        fprintf(' %03d ',iter)
    end
    if rem(iter,100) == 0 && iter ~= n_iter
        fprintf('\n Performing Iterations:... ')
    end

    %% REPLICATION
    [X1,V1] = EPSO_replication(X,V,r);

    %% MUTATION
    [w0_1,w1_1,w2_1] = EPSO_mutation(w0,w1,w2,r,t_w0,t_w1,t_w2);

    %% REPRODUCTION
    [X1,V1] = ...
        EPSO_reproduction(X1,V1,pBest,gBest,t_g,w0_1,w1_1,w2_1,xmin,xmax);

    %% EVALUATION
    f_X1 = EPSO_evaluation(X1,obj_fcn);

    %% Update Pbest & Gbest
    [pBest,f_pBest,gBest,f_gBest] = ...
        EPSO_pBest_gBest_update(X1,f_X1,pBest,f_pBest);

    %% SELECTION
    [X,V,w0,w1,w2] = EPSO_selection(X1,V1,w0_1,w1_1,w2_1,f_X1,pluck);

    %% Store results for current iteration
    Results.pBest(:,:,iter) = pBest;
    Results.gBest(:,iter)   = gBest;
    Results.f_pBest(iter,:) = f_pBest;
    Results.f_gBest(iter,:) = f_gBest;
end

%% Store final results
Results.gBest_final   = gBest;
Results.f_gBest_final = f_gBest;

%%
fprintf('\n\n EPSO algorithm done!!\n')
```