SOLUTION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM USING GENETIC ALGORITHM

UNIVERSITY OF KWAZULU-NATAL ™

INYUVESI YAKWAZULU-NATALI

BY

DEDACUS NNADOZIE OHAEGBUCHI

IN FULFILLMENT OF THE MASTERS OF SCIENCE IN ELECTRICAL ENGINEERING, COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE, UNIVERSITY OF KWAZULU-NATAL.

MAY, 2022

SUPERVISOR: PROF AKSHAY KUMAR SAHA

DECLARATION 1 - PLAGIARISM

I …….Dedacus Nnadozie Ohaegbuchi……………., declare that

1. The research reported in this dissertation, except where otherwise indicated, is my original research.

2. This dissertation has not been submitted for any degree or examination at any other university.

3. This dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

4. This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a. their words have been re-written but the general information attributed to them has been referenced,

b. where their exact words have been used, then their writings have been placed in italics and inside quotation marks and referenced.

5. This dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source detailed in the dissertation and in the References sections.

Signed **Date**

………………………………… …………………………………

As the candidate's Supervisor I agree/do not agree to the submission of this dissertation.

Signed **Date**

……………………………… …………………………………

DECLARATION 2 – PUBLICATIONS

Publications emanating from the masters research study are:

DEDICATION

This research work is wholly dedicated to the memory of my late Grandmother, Mrs. Margaret Uzunma Ohaegbuchi.

ACKNOWLEDGEMENTS

First, I would like to give thanks to God Almighty, the all Sovereign without whom this academic endeavor would have amounted to nothing. His guidance and comprehensive support sustained this research work.

To my supervisor, Prof Akshay, Kumar Saha, I convey my honest and deepest gratitude. I would not exchange his provision of brilliant guidance, mentorship, knowledge and expertise shared with me for anything. In addition, the confidence he inspired in the key stages of my masters research was a revelation

Many thanks to all faculty members, School of Engineering, and the various paper reviewers for their constructive comments which aided my thought processes in the completion of this master's Thesis.

Special thanks to my adored family for their unconditional affection, prayers, words of encouragement and support in my entire life. I would like to apologize to my missus (Dr. Mrs. Uchemaku Anidebe-Ohaegbuchi) for my absence during the time I was away conducting this research. Their support and patience during these two years were immense.

ABSTRACT

The combination of heat and power constitutes a system that provides electricity and thermal energy concurrently. Its high efficiency and significant emission reduction makes it an outstanding prospect in the future of energy production and transmission. The broad application of combined heat and power units requires the joint dispatch of power and heating systems, in which the modelling of combined heat and power units plays a vital role. This research paper employed genetic algorithm, artificial bee colony, differential evolution, particle swarm optimization and direct solution algorithms to evaluate the cost function as well as output decision variables of heat and power in a system that has simple cycle cogeneration unit with quadratic cost function. The system was first modeled to determine the various parameters of combined heat and power units in order to solve the economic dispatch problem with direct solution algorithm. In order for modelling to be done, a general structure of combined heat and power must be defined. The system considered in this research consists of four test units, i.e. two conventional power units, one combined heat and power unit and one heat-only unit. These algorithms were applied to on the research data set to determine the required decision variables while taking into account the power and heat units, operation bound of power and heat-only units as well as feasible operation region of the cogeneration unit. Power and heat output decision variables plus cost functions from Genetic Algorithm, differential evolution, Particle Swarm Optimization and artificial bee colony were determined using codes. Also, the decision variables and cost function value were obtained by calculations using direct solution algorithm. The findings of the research paper show that there are different ways in which combined heat and power economic dispatch variables can be determined, which include genetic algorithm, differential evolution, artificial bee colony, particle swarm optimization and direct solution algorithms. However, each solution method allows for different combined heat and power output decision variables to be found, with some of the methods (particle swarm optimization and artificial bee colony) having setbacks such as: large objective function values, slower convergence and large number solution. The analysis revealed that the differential evolution algorithm is a viable alternative to solving combined heat and power problems. This is due in most part to its faster convergence, minimum cost function value, and high quality solution which are diverse and widespread, more as a result of its effective search capability than genetic algorithm, particle swarm optimization, direct solution and artificial bee colony algorithms. The methods investigated in this research paper can be used and expanded on to create useful and accurate technique of solving combined heat and power problems.

TABLE OF CONTENTS

CHAPTER 5– IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM USING DIFFERENTIAL EVOLUTION ALGORITHM……….........84

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Ant colony search algorithm	ACSA
Artificial Bee Colony	ABC
Combined heat and power	CHP
Combined heat and power economic dispatch	CHPED
Economic Dispatch	ED
Feasible Operation Region	FOR
Genetic Algorithm	GA
Greenhouse Gas	GHG
Improved genetic algorithm	IGA
Multiplier updating	MU
Particle Swarm Optimization	PSO
Real-Time Operating System	RTOS

CHAPTER 1-INTRODUCTION

As an autonomous technology, combined heat and power production are both full-grown and orthodox. The efficiency of thermal power stations can be vastly improved by integrating cogeneration or combined heat and power plants to the existing power system. This can be useful in saving the substantial energy being wasted in the form of heat when fossil fuel is burnt to produce electricity in thermal power stations. Alternatively, combined heat and power will not only produce power using a variety of fuels, it has the potential to also recover and reuse the heat, which ordinarily would have been wasted during conventional power generation. Combined heat and power are associated with high energy efficiency with less greenhouse gas emission compared with other forms of energy supply. The fundamental difference between combined heat and power system, and conventional condensing plant is in the type of power obtained and the net efficiency of each system. In conventional condensing plant for instance, fuel energy is employed to generate only electrical power, while in combined heat and power system, fuel energy is employed to generate both thermal and electrical power hence increasing the system's efficiency.

The efficiency of a conventional condensing plant is in the range of 33-55 per cent. By applying efficient fuel condensation, the net efficiency of a combined heat and power unit is in the region of 80-90 per cent. In a combined heat and power plant, power generation depends on heat production and vice versa. This mechanism introduces complexity due to the non-separable heat associated with the combined heat and power unit as well as the nature of electrical power. The economic dispatch of combined heat and power plants is a more complicated problem in comparison with power unit dispatch due to the two-dimensional nature of the problem [1].

This research is therefore aimed at determining the combined heat and power economic dispatch decision variables, using proposed algorithm and (differential evolution algorithm, particle swarm optimization, direct solution algorithm, artificial bee colony algorithm). It thereafter investigates the optimization technique with minimum objective function value. Specifically, this chapter discusses the background and motivation behind the research topic, the aims and objective of the research topic, the significance and scope of the research study, statement of problem and the key questions to be addressed in the study and method of research.

1.1 Background of Study

Economic dispatch is basically concerned with the problem of determining the output of the generating units in service. Its objective is to meet up with total load demand, while keeping the fuel cost at its barest minimum. Economic dispatch problem is basically preoccupied with the computation of the optimal schedule of online generating units. The aim is to satisfy power demands at a minimal operating cost under the system's operating constraints like ramp rate limits [2] and forbidden operating zones. Prohibited operating zones represent nonlinear characteristics of a machine due to distortion of magnetic fields controlled by the power angle, the armature and excited currents respectively [3].In this research, each generating unit's cost function for the fuel is represented by a quadratic function. Economic dispatch problems represent an important industrial class of optimization problems, which are considered particularly difficult for conventional optimization method.

The main problem in economic dispatch is how to distribute generator load to produce required electricity. Examples of economic dispatch problems include economic load dispatch in the operation of power system, dynamic or static dispatch, hydrothermal scheduling problem and others [4]. An economic load dispatch problem is usually characterized by a significant number of variables and non-linearity, including non-linear constraints due to the characteristics of modern units. Improvements in solving this class of optimization problems may lead to significant cost savings. Positive results have come from the development of modern computational intelligence algorithms, including genetic algorithms. They have allowed complex optimization problems to be solved based on principles of genetics and natural selection, while not considering any additional knowledge about the problems at hand. An evolutionary algorithm is a subset of evolutionary computation, a generic population-based, meta-heuristic optimization algorithm and they include: genetic algorithms, evolutionary programming and differential evolution [5] etc.

Another popular class of algorithms is swarm intelligence algorithms which can deliver an assurance of finding the global optimum. However, when the potency of genetic algorithm was demonstrated in this research using genetic algorithm codes, we found out that genetic algorithm can find optimal solutions in terms of the objective function value, convergence speed and the number of solutions compared with other evolutionary and swarm algorithms. The above reasons have directed a remarkable amount of attention on genetic algorithm when compared with other optimization algorithms. Essentially, the traditional form of energy system is restricted to a single electric/thermal energy source and the interaction and reciprocal advantages between varied energy sources cannot be fully employed [6]. A single form of energy can no longer guarantee green and systematic energy demand. Hence, it is important to map out a secure and inexpensive integration of heat and power system. The combined heat and power system and electricity gained from a single energy source have a high-level efficiency, compared with a single power generation system since heat from power generation can be reused. In today's energy system, with the concerns of carbon emissions believed to contribute significantly to global climate change, combined heat and power systems are preferable. Combined heat and power systems can also provide an economic advantage: it will reduce fuel use and greenhouse gas emissions; ultimately lead to certain tax exemptions and in many places, receive incentives from governments etc [7].

Several research articles have appeared in literatures in recent times to not only prove the benefits of the system, but have equally helped in gaining a robust understanding of the optimization of the system's operations. In Comparison to the laboratory scale research, case studies of real life combined heat and power systems provide more accurate results and in-sight into the system's characteristics and their optimization options. Case studies from past research at different locations of the worldDos [7], [8], [9], [10] suggest that, it is important to run the prime mover, usually an engine at its maximum efficiency in order to obtain the benefits of cogeneration. It is also suggested that sizing the engine correctly according to the demand is very important. A properly constructed combined heat and power system can certainly help in minimizing cost, thereby guarantying a return on investment. Emissions from combined heat and power systems have also been investigated in comparison with other systems such as coal-fired power station or natural gas powered boilers. The results suggested a considerable reduction of all the emissions, regardless of the original system.

A key objective of this research therefore, is to compute the combined heat and power economic dispatch decision/control variables using different optimization techniques. Doing this will entail comparing result from proposed algorithm (genetic algorithm) with differential evolution, particle swarm optimization, artificial bee colony and direct solution algorithm. Finally, the research will determine the technique with minimum objective function. Although this method has been briefly researched in other related literatures, a comprehensive research and analysis is yet to be presented. First, a comprehensive modelling of combined heat and power economic dispatch problem was formulated in chapter 3 to help solve optimization problems using direct solution algorithm. As a result, this research aims to comprehensively scrutinize and evaluate combined heat and power economic dispatch variables, using genetic algorithm in comparison with differential evolution algorithm, particle swarm optimization, artificial bee colony algorithm and direct solution algorithm. It seeks to determine which of the five optimization algorithms has the least objective function.

1.2 Statement of the problem.

Combined heat and power economic dispatch problem has the foremost objective of obtaining the optimal heat generation and power generation schedules from a list of available power generating units, combined heat and power unit, and heat only unit. It premises a situation where the optimal schedule obtained reduces the total production cost while also satisfying the heat and power demands of the system, and several operational and physical constraints. Thus, the goal of this research is to determine the combined heat and power economic dispatch decision variables (heat and power optimal values) plus the cost function value using various optimization techniques (direct solution algorithm, Genetic algorithm, Particle swarm optimization algorithm, differential evolution algorithm and artificial bee colony algorithm).The combined heat and power economic dispatch decision variables (P_1, P_2, P_3, Q_3 and Q_4) including the cost function values of different optimization algorithms employed in the research were ultimately compared to determine the optimization technique with least objective function value. The research engaged the given data to compute the dispatch parameters using:

a) Direct Algorithm;

b) Genetic Algorithm;

c) Particle Swarm Optimization, artificial bee colony and differential evolution algorithms to compare the output decision variable results obtained from the five algorithms. Using the data in tables 3.1, 3.2, 3.3 and 3.4, we compute the individual decision variables below:

    i. Power outputs of units 1, 2 and 3,

    ii. Heat output of units 3 and 4,

    iii. The system Lambdas,

    iv. The cumulative cost of generation in the plant.

1.3**Aim and Objectives of the research:**

The aim of this research is to compute combined heat and power economic dispatch decision/control variables encountered in the system with simple cycle cogeneration unit. This is to be carried out using five powerful optimization algorithms (genetic algorithm, differential evolution algorithm, artificial bee colony, direct solution algorithm and particle swarm optimization algorithm). It will then compare the output decision variables from the five algorithms and finally determine the optimization technique that has minimum objective function. Genetic algorithm, differential evolution, artificial bee colony and particle swarm optimization techniques are search optimization techniques used to find optimum solution of combined heat and power problems.

The goal of combined heat and power is to minimize cost subject to inequality, equality and other physical constraints. An equality constraint consists of power and heat balances. Conversely, an inequality constraint consists of heat and power limits as well as feasible operation region between heat and power.

1.3 Motivation

Heat is generally known to be a byproduct of power generation in conventional power generation and boiler systems. When it is not fully utilized, it yields lower efficiency. Co-generation systems or combined heat and power generation systems, convert the heat from a power plant into a reusable byproduct for distribution to requested consumers. Thus, co-generation plants can produce both heat and electricity with better energy efficiency and low fuel consumption. Some other research works have adopted other optimization algorithms such as quadratic programming, partial separable programming, two layer Lagrangian relaxation, biography-based optimization algorithm, Newton's method, Tabu-search etc., in solving the combined heat and power economic dispatch problems. In this research nonetheless, the motivation is to adopt genetic, direct solution, differential evolution, artificial bee colony and particle swarm optimization algorithms in determining the possibility of achieving optimal solution of heat and power for this problem. In addition, to know if the computational (optimization) work process is less using the five optimization algorithms.

1.4 Significance of Study

Genetic, differential evolution, artificial bee colony, direct solution and particle swarm optimization algorithms would be essentially needed in research areas such as bioinformatics, computational sciences, electrical engineering, manufacturing, dynamic tracking and Maximum/Minimum problems. Those involved in real life application of linear and non-linear optimization problems would find it useful. Our emphasis on genetic, differential evolution, artificial bee colony, and particle swarm optimization and direct solution algorithms are invaluable tools for meaningful research work.

1.5 Scope of Study

We have used genetic, artificial bee colony, particle swarm optimization, differential evolution and direct solution algorithms to evaluate combined heat and power economic dispatch problem encountered in system with simple cycle cogeneration unit. We have also calculated the combined heat and power decision variables (power and heat, dispatch values) plus the cost function using the five optimization algorithms. We found that initial and final values of lambda were very close when the optimization problem was solved using direct solution algorithm. The individual output decision variables (power and heat values) of different units are within the confined limits of generator operation and the results of cost function values from five optimization algorithms are different.

1.6 Research method

a) Literature review

Additional published literatures have been reviewed. The theory with respect to combined heat and power (cogeneration), economic dispatch and genetic algorithm, its advantages, limitations and applications were re-examined.

b) Data

Data, such as power and heat units cost coefficients, coordinate of the corners of the feasible regions of the co-generation units have been sourced from industry for the research.

c) Data measurement

Data, for instance, the ones sourced in (b) have been strengthened through further measurements and calculations. These measurements included determining combined heat and power economic dispatch variables directly. The model of evaluating combined heat and power economic dispatch variables using(genetic algorithm, differential evolution algorithm, particle swarm optimization and artificial bee colony algorithm)have been generated using software such as MATLAB.

d) Testing of models

The output decision variables of various models (genetic, particle swarm optimization, direct solution, differential evolution and artificial bee colony algorithms) were tested against each other to determine the algorithm with minimum objective function value.

e) Analysis of results

f) Compilation of report

1.8 Research question/ hypothesis

To ensure that a comprehensive research on this topic is achieved, key questions to be addressed are as follows:

a) To what extent do different methods of determining the combined heat and power economic dispatch variables differ?

b) To what extent do cost function values of the five optimization algorithms used in this research have to differ?

c) To what extent do system initial and final lambda values differ when the optimization problem was solved directly?

1.9. Dissertation Layout

The dissertation is presented in different chapters. Each chapter focuses on a certain research aspect. The layout of the dissertation is as follows:

Chapter 2 reviews literatures of existing studies similar to this one. In this section, research results obtained in the cause of this study are compared with various literatures for confirmation of findings validity. Furthermore, unacknowledged contributions from existing studies that brought about this research are highlighted in the chapter.

Chapter 3 details the methodology applied in the entire research work. The chapter further discusses three methods of implementation of combined heat and power economic dispatch problem using direct solution algorithm, genetic algorithm and particle swarm optimization. Modelling of combined heat and power economic dispatch problem and discussion of results obtained from three optimization algorithms were carried out in this chapter.

Chapter 4 discusses the procedures, analysis and implementation outcomes of combined heat and power economic dispatch problem using artificial bee colony algorithm.

Chapter 5 focuses on the procedures, analysis and result of implementation of combined heat and power economic dispatch problem using differential evolution algorithm. Also, comparisons of the obtained results by Differential Evolution with other algorithms i.e. artificial bee colony, particle swarm optimization, direct solution and Genetic Algorithms were done in this chapter.

Chapter 6 highlights the conclusion of the research and associated findings. The shortfalls of the research work are discussed along with the findings presented, with recommendations for future work suggested in the end.

CHAPTER 2- LITERATURE REVIEW

2.0 Introduction

This chapter reviews literature of existing studies conducted which are similar to this dissertation. A variety of researches on combined heat and power economic dispatch exists. Nonetheless, this review focuses on studies on combined heat and power economic dispatch using genetic algorithm and; combined heat and power economic dispatch problem applying different forms of optimization algorithms such as particle swarm optimization, evolutionary programming, differential evolution etc. This study reviewed evaluations of combined heat and power economic dispatch from available literature and the conclusions made were based on authors' ideas.

Using genetic algorithm to determine combined heat and power economic dispatch decision variables helps to minimize cost of power generation. It also allows for proper power system load distribution planning among units in a plant. Furthermore, this chapter reviews and discusses the determination of combined heat and power problem using different optimization algorithms by different authors. Direct solution, genetic, artificial bee colony, differential evolution, and particle swarm optimization algorithms were used in this research to solve combined heat and power economic dispatch problems encountered in systems with simple cycle cogeneration unit. Results from each solution algorithm were compared to find out the algorithm with least objective function, optimal solution and best convergence speed. Finally, methods of focus in the conducted literature review are economic dispatch, combined heat and power (cogeneration) and genetic algorithm.

2.1 Economic Dispatch

Economic Dispatch is a constraint optimization problem whose objective is to find the economic schedule of a generating unit while maintaining operational and other physical constraints and load demand. Economic dispatch problem aims at finding the optimal schedule of generators in a bid to minimize fuel cost of power generation, subject to power balance and other operational constraints. The formulation of economic dispatch problem to economically operate the power system is vital to optimize cost of power generation. To minimize fuel cost in power plants, an objective combined heat and power problem is formulated. This mechanism provides considerable economic benefits while operating optimally. It is well known that at the optimum point, all the units (excluding those at their limit) would be operating at equal incremental costs. To achieve economic operation of generating units in a plant, economic dispatch is carried out. Economic dispatch problem is one of the vital issues in power system operation. It is commonly formulated as an optimization problem [11]. It involves active power allocation between the generators to minimize the overall operating cost and also maintain total power and heat demand constraints as well as generator capacity constraints [12].

A number of conventional methods have been established to solve the economic dispatch problem. Instances are iteration method [21], Lagrangian relaxation method [22], proposed improved particle swarm optimization [35], etc. The authors gave a concise explanation of economic dispatch, formulation of economic dispatch problems and how to solve associated problems by implementation of optimization techniques to determine optimal decision variables of heat and power.

However, these literatures fall short of discussing forecasting cogeneration amongst units in power system. They also do not offer explicit explanation of the constraint accrued from handling mechanism of cogeneration unit. These optimization algorithms are nevertheless, implemented in a centralized form, which requires a central node to collect global information of all the generators, conduct the optimization [22] and transmit command globally. Practically, collecting detailed information is usually cost effective in both communication and computation, when the power system becomes more complex as pointed in [14].Besides; such centralized algorithms are unable to meet the plug and play requirement in the new smart grid system. The authors gave benefit of data collection and communication which helps to determine how load could be distributed between units in a plant.

2.2 Combined heat and Power (cogeneration)

This work gives a general overview of combined heat and power system, and how heat is being reprocessed from the prime mover of a plant. Cogeneration is a system that provides electricity and thermal energy concurrently. The system consists of a generator, a heat recovery system and electrical interconnections. The thermal power is constructively reprocessed from the heat secured from combustion in the prime mover of the system. The application of cogeneration systems expands the effectiveness of energy production from 35% to 85%. Economic viability has propelled people to install these systems, knowing that the production of electricity and heat is on site. Recent researches on cogeneration have concerted on novel configurations of cogeneration plants using fuel as energy sources [15],[16].Interestingly, authors of these literatures have explained high energy efficiency as well as cost savings accruing from cogeneration systems. They have however, failed to discuss the environmental impacts of cogeneration system compared with traditional fossil fuel power plant. Thus, ultimately, there is a gap in literature on how the higher efficiency of cogeneration in reducing fossil fuel consumption and how it amounts to reduction in the overall emission to the atmosphere.

The increasingly severe requirements for carbon dioxide reduction led to a more universal promotion of distributed energy systems. One of the most effective methods of combating the energy saving challenges is the use of cogeneration systems. Generally, combined heat and power production or cogeneration is significantly more methodical than the distinct production of heat and power because of its reduction of overall fuel utilization, leading to lower emission [17]. The author of this literature explained the benefits of using cogeneration over centralized systems such as energy savings, elimination of transmission loss etc. He however fails to discuss the challenges firms and individuals implementing cogeneration systems such as electricity market restructuring encounter in most countries (Denmark, Finland and Netherlands) that predominantly make use of cogeneration system vulnerable of grid systems and volatility of natural gas. To obtain optimal utilization of combined heat and power units, the combined heat and power is added to economic dispatch problem to form a combined heat and power economic dispatch as an optimization problem. Cogeneration units have power and heat output which shows that the operating orbit of cogeneration plants is more convoluted than a heat only or power only unit. Recent shift shows that combined heat and power is one of the methods to decentralized energy which results to minimal loss on transmission systems. It also boosts the system's competence while providing energy directly or near to end users. For effective application of cogeneration systems, the economic dispatch of combined heat and power was established. That said, the fundamental objective of combined heat and power economic dispatch is to evaluate the most economic loading point of the combined heat and power generation units. This is to be done in a way that both heat and power demands are utilized within the bounded region in the heat versus power dimensional surface. Thus, it forms a complex inequality constraint condition to be handled for satisfactory operation of combined heat and power units within their capacity limits. The absolute thermodynamic dispatch put forward by cogeneration plants indicates that cogeneration is the approved means of gratifying the energy demands of industries; and that it can remarkably supply to the systematized use of energy. In growing nations, principally those with inadequate energy resources and also restricted financial resources for supplying the much needed energy for social and economic development, industrial cogeneration produces the chance of amplifying national energy sources, thus serving to continue expansion programs positioned at growing the quality and living standard of the population.

In spite of the significant research conducted in the area of combined heat and power economic dispatch in the past decades, much of recent research efforts have evolved in order to obtain optimum dispatch at most favorable cost [18], [24],[20]. An algorithm proposed by [21] employed two level approaches in which the lower level was presented to solve the given heat and power lambdas, and the upper level upgrades of the lambda's reactivity coefficients. This framework is undiversified until the power and heat stipulation are encountered. The authors used particle swarm optimization algorithm to solve constrained optimization problem and the obtained objective function value was low. However, it was discovered that iteration process was long leading to decrease in cost function value. But this solution got trapped at local optimum due to unsatisfactory system constraints. Again, authors in [22] researched on algorithm for combined heat and power economic dispatch problem in which the problem was fragmented into two sub-problems: power dispatch and the heat dispatch. When this literature was compared with existing methods of solving combined heat and power problem, the technique converged to optimal point and also produced diverse, widespread solutions along with better extreme solutions due to its effective search capability. Hence the overall cost function is lower.

In the present research, genetic, particle swarm optimization, differential evolution, artificial bee colony and direct solution algorithms were employed to solve combined heat and power problem encountered in a system with simple cycle cogeneration units. The solution from proposed (genetic algorithm) where compared with ones existing in literature. Also, authors in [23] elaborated a direct solution algorithm for the combined heat and power economic dispatch problem in which he produced a formula for the system lambdas analogous to heat and power demands in terms of the coefficients of the generator cost function. He further recalculated the system lambdas in order to remove the mismatch associated with heat and power. The author solved combined heat and power economic dispatch problem directly by substituting the data inside the formulated equations to obtain the dispatch value of heat and power. But he failed to explain how to handle violations efficiently in a cogeneration unit. He used few units in this literature and therefore did not give a comprehensive description on how to solve this problem when the units are well expanded. Again, authors in [24] propounded an algorithm to solve the economic dispatch problem for combined heat and power systems. In the literature, the developed algorithm takes the method of sequential quadratic programming algorithm employed to solve non-linear optimization problems and the logic of the Lagrangian relaxation method. Thus, rather than considering linear inequality constraints, it briefly eliminates them from the problem, making the problem less difficult. The authors of this literature obtained a minimum objective function in their result but failed to handle the constraint problem on the cogeneration units where output decision variables are out of feasible operation region. Therefore, the proposed (genetic) algorithm can handle this constraint problem efficiently because of its ability to handle associated constraint problems. Furthermore, authors in [13] established a much popularized, adaptable and systematic Lagrangian relaxation method for combined heat and power economic dispatch. This technique is split into two optimization levels known as upper and lower levels. The lower level takes care of optimization of independent units while the upper levels unravel the global constraints. This procedure dispenses adaptability for separable problem in which different, effective and simple methods could be applied to decipher the lower level sub-problem for most favorable solution. In this literature the author used the Lagrangian relaxation technique to solve combined heat and power problem but the result obtained did not satisfy the system constraints and different characteristics of the system. Hence, application of the proposed method is recommended to handle constraint problem effectively in order to obtain global optimal solution with minimum cost function. Authors in [25] established an evolutionary programming algorithm for the combined heat and power economic dispatch problem for cogeneration systems.

Also established are the unsystematic initialization and constraint modification for the feasible operation region of the cogeneration unit. In this procedure, the expressions for the standard deviation applied in the mutation operation have been planned such that the size of the mutation search range can be regulated. Thus, the neighborhood of the best individual in a population is searched. In this literature, it is vital to note that solutions obtained by this method is not efficient since various parameters such as mutation rate, crossover rate and population size cannot be handled using this technique. Optimal solution of this class of combined heat and power problem can be obtained with proposed algorithm because of its convergence speed and diverse population in the search space which enables global and feasible solution to be obtained. Furthermore, authors in [26] established combined heat and power economic dispatch problem of a system by using particle swarm optimization to ascertain the unit power and heat creation so that manufacturing cost could be pruned. From the considered literature, the expected objective function value was large and supposed solution point did not satisfy the load constraints.

Authors in [27] established a harmony search algorithm for solving the combined heat and power economic dispatch problem. The output decision variables of the proposed method showed that the proposed algorithm can find better solutions when compared with traditional techniques. Asides that, it is a well-organized search algorithm for combined heat and power economic problem. The author of this literature applied harmony search to solve combined heat and power. The objective function value obtained is high because of premature convergence. When compared with the proposed method, it was discovered that genetic algorithm had a good objective function value and a well convergence speed because of its diverse population in the search space. Authors in [28] proposed a new optimization algorithm for the combined heat and power economic dispatch problem by applying artificial Bee Colony optimization algorithm. It is as warm-based algorithm inspired by the food foraging behavior of honey bees. The work has a similar concept with genetic algorithm but had a slow convergence speed, large objective function value when compared with genetic algorithm result using the same data. Furthermore, authors in [29] introduced artificial immune system algorithm for solving the combined heat and power economic dispatch problem. Artificial immune system is rooted on the clonal selection assumptions which accomplish adaptive Cloning, hyper-mutation, aging operator and tournament selection. This same literature also considered the valve point effect which makes the search difficult and ineffective to obtain an optimal solution. To achieve an optimal solution that satisfies all the system constraints and obtains minimum objective function, optimization problem of this class could be sorted by incorporating integrated simulation-based algorithm in the frame of the proposed algorithm. Author [30] scrutinized the implementation of differential evolution for solving combined heat and power economic dispatch problems in power systems.

In his research, a differential evolution established algorithm was expanded for solving the combined heat and power economic dispatch problem. This was done by taking into account quadratic cost function together with valve point loading for the electrical power generating units. The author used differential evolution with penalty factor to solve combined heat and power problem. However, the result obtained was found to be satisfactory for the conventional power and heat unit; but heat and power output of congregation units were infeasible. Hence it is recommended to employ proposed algorithm to obtain optimal result because of its characteristic of handling non-linear constraint problems effectively. Authors in [31] upgraded search methodology based on administered autocatalytic process, known as the ant colony search algorithm, to solve combined heat and power economic dispatch problem. Chief attributes of the ant colony search algorithm include: positive feedbacks, distributed computation and the implementation of a constructive greedy heuristic. Su and Authors in [35] propounded an improved genetic algorithm with multiplier updating which integrates the improved genetic algorithm and the multiplier updating such that it is systematic for large-scale combined heat and power economic dispatch problem.

In this literature, the improved genetic algorithm systematically searches the most favorable solutions in the economic dispatch process, whereas the multiplier updating practically manages the power-heat feasible region constraints. The recommended algorithm integrates the improved genetic algorithm and the multiplier updating such that it has the advantage of spontaneously modifying the randomly given penalty to acceptable value and needing only a small-size population. This method has failed to produce a better result when compared with genetic algorithm because of the premature convergence and ineffective handling of constraint related to the cogeneration unit. However, this problem can be resolved by application of proposed algorithm because of its constraint problems-handling ability targeted at generating globally optimal solution.

Authors in [26] propounded the time-varying acceleration coefficients particle swarm optimization algorithm for solving combined heat and power economic dispatch problem. The idea behind this method is to ameliorate the seeking ability of the classical particle swarm optimization and to apply a parameter automation scheme for suitable balancing between the global and local search. This algorithm has a characteristic of being consistent and sensible. The authors of this literature analyzed combined heat and power problem using time varying acceleration coefficients particle swarm optimization algorithm. The literature however failed to account for the feasible operating region of the cogeneration unit, while the proposed method considered it to obtain the desired result with minimum objective function and stable convergence speed. Again, authors in [32] propounded a new algorithm rooted on the direct search technique for the solution of the combined heat and power economic dispatch problem. The technique proposes a potent tactic rooted on a successive refinement search technique that also guarantees a possibly absolute examination of the solution space, to increase the possibility of exploring the search space where the global optimal solutions subsist. The author of this literature analyzed combined heat and power problem using two search methods: the result obtained fell out of the constraints due its inability to achieve solution with high accuracy, thereby generating high cost function. Optimization problem of this class can be resolved with the application of proposed algorithm which produces diverse and feasible solution with least objective function.

Authors in [33], initially developed the stochastic model for combined heat and power dispatch, and then an improved particle swarm optimization technique was devised to deal with the economic combined heat and power dispatch by concurrently taking into account multiple conflicting objectives. The authors highlighted the solution of combined heat and power economic dispatch using improved particle swarm optimization. The result has a large objective function value due to slow convergence speed and previous work done on this. The proposed algorithm yielded optimal solution with high quality convergence speed and low objective function value. Also, authors in [33] propounded optimization algorithm based on improved differential evolution to solve the combined heat and power economic dispatch problem. According to this literature, the evolutionary mechanism of the improved differential evolution is more efficient than the genuine differential evolution. Plus, it has the merit of being easy to comprehend, along with an easy implementation process ready for exploitation for a broad optimization problem. The author used improved differential evolution to solve combined heat and power problem and the result was compared with other well-known algorithm (genetic algorithm). The result showed that proposed algorithm has global optimal solution with low total fuel cost value than both improved differential evolution and artificial Bee colony. Therefore, proposed algorithm has superiority both in computation efficiency and solution quality. Again, authors in [31] used an improved penalty function formulation with Genetic algorithm to solve the combined heat and power economic dispatch problem. The propounded method employed an improved penalty function formulation with genetic algorithm with a penalty factor that can be adaptively modified during the evolution operation, to constructively solve constrained optimization problem. The authors of this literature applied adaptive penalty to solve combined heat and power economic dispatch. The result showed that all the best individuals in the last K (severity parameter) generation were not feasible. Asides that, there was a problem in choosing the near feasibility threshold, hence this method failed to produce diversity in the population, and the solution is infeasible.

It is recommended to deploy proposed algorithm to solve this class of optimization problem towards producing a diverse population capable of attaining global optimum with better convergence speed during iteration. Authors in [34] used a hybrid of genetic algorithm with Tabu-search in a system that has four units. The propounded technique is developed in such a way that a simple genetic algorithm is improvised as a base level search directed towards the optimal region. In a similar vein, local searches synergistically combined with Tabu-search were used to fine-tune the search to reach the optimal solution. The authors used hybridized genetic algorithm and Tabu-search to solve combined heat and power problem. The main idea was to obtain a low fuel cost value and as a well as a solution with fast and feasible convergence speed. But it was discovered that these algorithms have long computation time, slow convergence speed and output decision variable are infeasible. The problem of this literature could be resolved by using proposed algorithm because of its ability to converge effectively and generate diverse and globally optimal solution with minimum cost function.

Again, authors in [34] employed a multi-objective technique by applying a fuzzy decision index and genetic algorithm to a seven-unit system. Three targets were set in this literature: the total generation cost, the expected power generation deviation and the expected heat generation deviation of the system were pruned in their work. The application of fuzzy decision index and genetic algorithm in this literature shows that convergence speed was slower; had a relatively minimal objective function, but failed to present best feasible decision variable output. Optimal result can be obtained by combining genetic algorithm with gradient method; this will lead to a high performance in minimizing the cost function, as well as produce solution that converges to a global optimum. Furthermore, authors in [35] proposed an optimization algorithm known as mesh adaptive direct search to solve the combined heat and power economic dispatch problem with bounded feasible operation region.

In this work, the *Latin hypercube sampling*, particle swarm optimization, design and analysis of computer experiments surrogate algorithms, were employed as search schemes to solve each of the combined heat and power economic dispatch problems. This literature employed mesh adaptive search to solve combined heat and power problem. The result showed that defining of constraints was expensive and inconvenient to test. The search steps failed to generate improved mesh point which led to application of Poll which consequently resulted in a premature convergence and high objective function value. However, with the application of proposed algorithm, premature convergence is eliminated in this class of optimization problem whereas solution obtained with proposed algorithm converges to global optimal with minimum objective function. Authors in [48] propounded differential Evolution for reducing possibly non-linear and non-differential continuous space function. The authors demonstrated that this algorithm converges quicker and with more reliability. However, the authors' literature can be improved by applying annealed version of proposed algorithm to solve this class of optimization problem. This produces feasible result that is potent, converges globally optimal and is capable of interchanging information inside the model.

Also, authors in [30] proposed a differential evolution based algorithm for solving combined heat and power economic dispatch problem, while accounting for quadratic function with valve point loading effect for the electrical power generating units. One of the problems of this literature is insufficient population diversity which failed to improve mutation and search abilities. When this result was compared with genetic algorithm, being a different evolutionary algorithm, it was found that genetic algorithms results provided optimal solution due to its enhanced search capability and minimum objective function value. Again, authors in [38] proposed a differential evolution-based algorithm to solve constrained optimization problems. In his method, each solution is permitted to churn out more than one offspring. This is moreover accomplished by applying a different mutation operator which amalgamates details of the best solution in the population and also facts about the current parent to find new search directions. The author's findings contradict the results obtained from genetic algorithm with Tabu-search, when solved with this hybrid algorithm. Therefore, for optimal result, genetic algorithm is recommended for this class of combined heat and power problem, for its efficient constraint handling mechanism, optimal solution with high quality convergence speed and least objective function value. Authors in [39] propounded the idea of existing strategy in evolutionary computation model for generation alternation in differential evolution. This model finds the global minimum at a higher convergence velocity. In this literature, by selecting parents for breeding and offspring for survival, differential evolutions search capability gets further accelerated, which will be particularly useful for expensive function optimization. This literature failed to find the global minima owing to problems associated with initial parameter value which converges prematurely with infeasible solution. This work can be improved by applying the proposed algorithm which generates potent initial population that converges optimally with feasible solution and low objective function value.

Again, authors in [36] proposed differential evolution as a population-based stochastic function optimizer by applying vector differences for perturbing the population. Both researchers perceived that constraint handling technique of differential evolution minimizes the real numeric value of required objective and constraint functions evaluation. Hence, more generations or solution candidates' evaluation could be executed. In this dissertation, the author failed to obtain satisfactory objective function because cogeneration units' constraint-handling mechanism was not performed accurately. This failure led to the premature convergence of candidate solutions and the infeasibility of output decision variables. The integration of differential evolution with other intelligent meta-heuristic (genetic algorithm) will improve the quality of solution in a few iteration lines. This will further produce optimally global results as well as feasible solution.

Authors in [40] proposed a differential evolution algorithm to solve emission constrained economic dispatch problem. The developed algorithm strives to minimize the production of atmospheric emissions like nitrogen oxides and Sulphur IV oxide etc. Such minimization is achieved by including emissions as a constraint in the objective of the overall dispatch problem. The algorithm in this literature was designed to optimize fuel cost alongside emission reduction. However, results obtained yielded a compromise solution and had to be run multiple times. One of the ways to overcome this setback is to use meta-heuristic algorithm (genetic algorithm) to produce trade-off solutions in a few iteration runs which converge to a global optimum and yield minimum objective function value. Authors in [41] propounded a technique based on differential evolution algorithm, competent for optimizing all integers' i.e. discrete, continuous variables proficient for handling non-linear objective functions with multiple non-trivial constraints. The author of this literature adopted penalty method to handle the constraints equations with differential evolution. The application of penalty method has entirely eliminated high quality initial solution and this result to infeasible solution. To achieve global optima in this class of optimization problem, meta-heuristics such as genetic algorithm is required to ensure high quality initial candidate solutions are generated to enable the output decision variables converge to global optimum and feasible with minimum objective function value.

Authors in [66] propounded a differential evolution algorithm for a constrained optimization problem based on multi-objective constraint-handling. The propounded algorithm can attain global search and local search successfully by employing differential evolution algorithm whose characteristic is changing constraints into a target and transforming the problem into two-objective optimization problems. In this literature, the author's objective aimed at solving constrained optimization problem with multi-objective constraint-handling mechanism. However, the optimization problem has diversity in population of genes during search which led to early or premature convergence of the population with infeasible solution and high objective function value. With proposed algorithm, the problem of premature convergence will be eliminated by generating strong initial candidate solutions that yield optimal solutions which converge globally with minimum objective function value.

Again authors in [39] propounded a differential evolution algorithm for solving combined heat and power economic dispatch problem. This literature presented a suitable application for gratifying the power balance constraint. It included other boundary constraints acceptable for applying a reflection mechanism frequently employed in constrained optimization with differential evolution. Also in this literature, load-constraint problem was solved using differential evolution. Nonetheless, the proposed technique used by these authors failed in the area of explaining the constraint-handling mechanism. Hence, extraction of information and obtaining feasible solution becomes difficult with this strategy. To solve this problem, proposed algorithm was used for its efficient mechanism of constraint handling. Finally, genetic algorithm produced globally feasible solutions which satisfy all problem constraints and generate minimum objective function.

Also, authors in [42] developed differential evolution algorithm employed for solving two problems simultaneously:

a) A multi-optimization problem (with two objective functions to be maximized) using penalty function technique and weighing factor technique; and
b) Classical Himmelblau's function. This literature adopted differential evolution to accommodate multi-objective optimization problem.

It also removed the dominant solution in last generation during iteration to lower the function evaluation thereby improving them while also increasing the number of non-dominant solutions leading to diversity. However, the algorithm failed in the area of affording gradable spread of diversity with good convergence. The convergence speed of proposed algorithm is better than that of algorithm employed in this literature. The reason being its ability to produce diverse population in the search space during iteration. Therefore, it is suggested that genetic algorithm for this class of optimization problem be adopted to obtain a diverse, feasibly global optimal solution with minimum objective function value.

Again, authors in [43] propounded a differential evolutionary algorithm for scheduling of units. It was included with some other techniques for determining the quantity of power to be manufactured by committed units. In order to gratify power balance equality constraint, the authors propounded a binary-real-coded differential evolution. Here, the binary part deals with the scheduling of units and the real part determines the quantity of power manufactured by committed units. What the literature did was to develop a way of handling unit scheduling in a plant with differential evolution algorithm. The research result showed that differential evolution outperformed particle swarm optimization in terms of optimal solution but required large computation to explore optimal solution with maximum convergence speed.

Authors in [29] developed a new proposition based on differential evolution for solving the combined heat and power economic dispatch problem. The calculation time, coherence and its proficiency in managing a wide class of optimization problems constitute the major advantages of this procedure. Hence, the literature attempted solving combined heat and power problem using differential evolution, but the output decision variable of cogeneration unit fell out of the feasible operation region. This computation can be effectively managed by incorporating differential evolution with other random search methods such as genetic algorithm; particle swarm optimization etc., to ensure proper handling of cogeneration unit in bid to obtain diversely optimal solutions which generate minimal objective function value. As has been previously noted, the implemented classical and mathematical-based optimization algorithms are not comprehensive for solving nonlinear and non-convex optimization problems. On the other hand, meta-heuristic algorithms like proposed algorithm can find better results in comparison with classical optimization techniques in non-convex optimization problems. By investigating the literature in combined heat and power economic dispatch problem solution, it can be observed that different heuristic algorithms produced different solutions. A better solution for combined heat and power economic dispatch problem has a great economic saving in system operation cost. Hence, it is required to improve the capabilities of heuristic algorithms, such that more optimal solutions (i.e. solutions, with lower costs) are attained for non-convex combined heat and power economic dispatch problems. It is worth mentioning that some exact gradient-based mathematical programming algorithms have been not implemented for combined heat and power economic dispatch in the literature. Therefore, it is not possible to judge their performance in comparison with the meta-heuristic optimization algorithms, and it can be considered in future works.

2.3 Genetic Algorithm

Along with genetic programming, differential evolution, evolution strategies, and evolutionary programming, genetic algorithm is a member of the evolutionary algorithms family. Evolutionary algorithms may be examined as a wider class of stochastic optimization algorithm. An evolutionary algorithm preserves a population of candidate solutions for a given optimization problem. The population is then progressed by the frequent implementation of a set of stochastic operators. The set of operators usually comprises of recombination, selection mutation or something with precise homogeneity. The author of this literature has described genetic algorithm as a branch of evolutionary algorithm. He has however not done enough in the areas of its history. In particular, the body of work carried out by researchers who studied computer simulations of biological evolution was not explained in details. Here, we discuss some research works carried out in relation to genetic algorithm and other evolutionary algorithms in the recent past. Authors in [46] contrasted the results which come after using different crossover and mutation operators formulated for the traveling salesman problem. It was concluded that operators that employed heuristic information or a matrix depiction of the graph produced the best results. Although this literature applied crossover and mutation operators in travelling salesman problem, failed to take into consideration the probabilities of cross-over and mutation during the computation. Genetic algorithms are evolutionary algorithms that employ crossover and mutation operators to solve optimization problems using a survival of the fittest technique.

Genetic algorithms have been applied successfully in diverse problems, including the traveling salesman and optimization problems. In the traveling salesman problem for instance, the objective is to determine a tour of all nodes in a weighted graph so that the net weight is reduced. The traveling salesman problem is NP-hard but has many real world applications so a good solution would be useful. Author [45] in his work, Evolutionary Algorithms, proposed that evolutionary Algorithms are stochastic optimization techniques established on the principles of natural evolution. A survey of these techniques is provided with the general functioning of evolutionary algorithms. It outlines the main families into which they are separated. Beyond that, it scrutinizes the distinct constituents of evolutionary algorithms, and provides some examples on how these can be constituted. In the end it finished with a glance of the numerous applications of these techniques. The different families of evolutionary algorithms are Evolutionary Programming, Evolutionary strategies, Genetic programming and Genetic Algorithms. The fundamental differences between these algorithms lie in the nature of the depiction schemes, the duplication operators and selection techniques. The work however failed to indicate that evolutionary algorithm such as differential evolution algorithm produce a single offspring by adding the weighted difference between two parents to a third parent.

Along with propounding a coordination mechanism, author [47] proposed genetic algorithm for the resolution of real world scheduling problems. Due to the frequency in change of environment, providing efficient production management and timely delivery are two difficult problems to solve. Scheduling is to assign a set of machines to execute a set of jobs within a certain timeframe and the goal of scheduling is to determine an appropriate assigned schedule which maximizes certain performance measure. For the implementation problems, the solutions are encoded by natural representation and the order of crossover operator is employed. They used the inversion mechanism as mutation operator. Finally, author [47] solved dynamic scheduling problems using a set of static scheduling strategies through genetic algorithm to demonstrate feasibility in Job-Shop scheduling problem. This literature did not however take the factors of priority among tasks into consideration; instead, optimization sequencing was merely done from the perspective of the length of time. But in reality, the priority of each task needs to be set so as to make it more targeted and practical. Again, authors in [69] proposed the genetic algorithm used for allocating task preferences and offset to bond that real-time timing constraint. Allocating timing constraint to task is not trivial problem in real-time system. They showed how timing constraints could be mapped to show attributes of periodic tasks running on standard preemptive real-time operating system. They applied genetic algorithm for its ability to produce results that persuade a subset of the timing constraints in cases where it is impossible to fulfill all constraints.

In genetic algorithm, the mechanism of natural selection gradually enhances individuals' timing constraint assignment in a population. It has been tested on so many test cases and results obtained have been improved. This work failed to show understanding of how a result produced by genetic algorithm on allocation tasks is only satisfactory and therefore requires other optimization tools to obtain an optimal result. In his research on test functions for optimization needs supplies, author [48] proposed the evaluation of literature benchmarks (test functions) commonly employed in order to test optimization procedures devoted to multi-dimensional, continuous optimization task. Deeper observations have been made on multiple-extreme functions, treated as the quality test for opposing optimization methods like (genetic algorithm etc.). Quality of optimization procedure is repeatedly investigated by using common standard literature benchmark. The test functions are of several, continuous classes, comprising unimodal, convex, multi-dimensional classed as the first; the second is multimodal, two-dimensional with a little number of local extremes; the third is multimodal, two-dimensional with large number of local extremes; and the last is multimodal, multi-dimensional with large number of local extremes. Class one contains good functions as well as malicious cases, causing poor or slow convergence to single global extreme. Class two is intervening between first and third; whereas the last class is used to test quality of standard optimization procedures in the unfriendly environment, namely that having few local extremes with single global one. Classes three and four are recommended to test quality of intelligent resistant optimization technique. This dissertation compares proposed method with well-known optimization techniques such as particle swarm optimization and artificial bee colony, direct solution and differential evolution algorithms in different scenarios. However, test functions are often listed on literature as unconstraint problem which makes it impossible to handle constraint optimization problems.

Authors in [49], proposed a method of solving job-shop scheduling using genetic algorithm. They produced an inceptive population randomly, including the result acquired by some well-known priority rules such as shortest and longest processing times respectively. From there, the population would go through the processes of duplication, crossover and mutation to create a new population for the next generation until some stopping criteria defined were met. In the work, the number of generations is used as stopping criterion. In crossover and mutation, the critical block neighborhood is employed along with the distance measured to help evaluate the schedules. Result has shown that implementation of critical block neighborhood and the distance measured can lead to the same result acquired by other approaches. In this literature, the manner of priority calculation has a non-negligible impact on solutions. An appropriately chosen priority calculation can improve results, while an inappropriately chosen priority calculation can worsen them. Hence, this algorithm performance can be further improved by proposing new techniques of priority calculation such as proposed algorithm.

Author [47] proposed in his work on genetic algorithm approach to Operating system process scheduling problem. Scheduling in operating systems has a vital role in overall system implementation and throughout. A methodical scheduling is vital for system execution. The scheduling is considered as NP-hard problem. The power of genetic algorithm is employed to furnish the systematized operation scheduling. The goal is to acquire a methodical scheduler to assigned and schedule the operation to central processing unit. The author of this literature was unable to approach optimal solution in solving this problem and therefore the performance of his work can be improved by using dominance and diploidic operators. Again, author in [61] proposed in his work on A Genetic Algorithm on Single Machine Scheduling Problem to Minimize net Weighted Completion Time. In his work, he proffers solution on a single machine family scheduling problem where there are multiple jobs. Each job is characterized by a processing time. An associated positive weight is partitioned into families and setup time is needed between these families. For this problem, he proposed a genetic algorithm using an optimized crossover operator designed by an undirected bipartite graph to find an optimal schedule which minimizes the total weighted completion time of the jobs in the presence of the sequence-independent family setup times. The literature explored optimized crossover genetic algorithm on single machine scheduling problem. It was observed that solution produced with this algorithm tend to converge prematurely.

2.4 APPLICATIONS OF GENETIC ALGORITHM, ADVANTAGES AND LIMITATIONS

Genetic algorithms are a class of numerical and combinational optimizers known to be especially handy in solving complex, non-linear, and convex optimization problems. Holland was perhaps the earliest to adopt the crossover, recombination, mutation and selection models in the study of adaptive, artificial systems. These genetic operators constitute vital parts of genetic algorithm as a problem-solving procedure. It then follows that innumerable variants of genetic algorithms have not only been expanded, but adapted to suite a vast range of optimization problems: from graph coloring to pattern recognition; and from discrete systems exemplified by the Travelling Salesman Problem (TSP) to continuous systems— the effectual model of airfoil in aerospace engineering; from financial markets to multi-objective engineering optimization problems. The advantages of genetic algorithm are numerous. The first most prominent is: the ability to deal with complex problems and parallelism. Genetic algorithms can deal with different classes of optimization problems, whether or not the objective (fitness) function is stationary, linear, continuous, or with random noise. For the fact that multiple off-spring in a population act like independent agents, the population (or any subgroup) can explore the search space in several directions concurrently. This feature makes it perfect to parallelize the algorithms for implementation. Different parameters and even different groups of encoded strings can be manipulated at the same time [49]. Below is the summary of advantages of genetic algorithm.

2.4.1 Advantages of Genetic Algorithm

A. Parallelism

Evolution is an extreme parallel process. As distributed processing computers become more easily available, there will be a similar increased potential for applying genetic algorithms to more complex optimization problems. It is often the case that individual solutions can be independently evaluated from the assigned evaluations to competing solutions. Each solution’s estimation can be analogously handled. Thus, only the selection operator requires some serial processing. Essentially, the operational time necessary for an application may be inversely proportional to the number of processors. Regardless of these future merits, current desktop computing machines provide sufficient computational speed to produce solutions to difficult problems in reasonable time. For instance, the evolution of a neural network for classifying features of breast carcinoma involving over 5 million separate functional evaluations, needs only about three hours on a 200 MHz 604e Power personal computer [50].

B. Broad Applicability

Application of genetic algorithms may become relevant in solving early any bottleneck arising as a function optimization problem. All it requires is data structure for solutions representation; a performance index for solutions evaluation; and variation operators for new solutions regeneration. While selection is also necessary, it is a lot less dependent on human preferences. Hence, possible solutions can both be disjointed and encompass infeasible regions. It equally follows that performance index can be time-varying, or even a function of competing solutions extant in the population. The human designers can select a representation that follows their instinct. In this scenario, the mechanism is representation independent, in contrast with other numerical techniques applicable for only continuous values or other constrained sets. Thus, representation should allow for parent-offspring behavioral link maintenance variation operators. Slight changes in the structure of a parent should necessarily amount to equally slight changes in the resulting offspring. Likewise, significant alterations should engender gross alterations. A continuum of possible changes should be admitted to make allowances for effective step-size of the algorithm being tuned, perhaps online in a self-adaptive manner. Such flexibilities accommodate essential applications of the same mechanism to discrete combinatorial problems, continuous-valued parameter optimization problems, mixed-integer problems, and so forth.

C. Potential to Use Knowledge and Hybridize with other Methods

Incorporating domain-specific knowledge into an algorithm when addressing particular real-world problems is the rational thing to do. Put on a restricted interest domain, specialized algorithms have the capacity to outperform unspecialized algorithms [53]. Genetic algorithms provide a framework that makes it comparatively easy to incorporate such knowledge. For instance, individual variation operators could be useful when applied to certain representations typified by 2-OPT on the traveling salesman problem. Besides the possibility of these being directly demonstrated as recombination or mutation operations, it can be adapted to translate knowledge into the performance index, as known physical or chemical properties as seen in van der Waals interactions [51]. Such information's adaptability emphasizes genetic search, with a more efficient study of the state space for possible solutions. More traditional optimization techniques as simple as a conjugate-gradient minimization applied after primary search may also align with genetic algorithm. An instance is seen in authors [51]. It may by extension, involve simultaneous application of algorithms as typified by the evolutionary search for the structure of a model, alongside gradient search for parameter values. Seeding an initial population with solutions derived from other mechanisms, for instance, in a greedy algorithm [50], may equally be beneficial. In addition, application of genetic algorithm computation can be implemented for performance optimization of neural networks as seen in fuzzy systems [54] production systems [56]; and other program structures [50]. Several cases have shown the limitations of conventional approaches. A case like the prerequisites for differentiable hidden nodes when using back propagation to train a neural network, as an instance, could be avoided [51].

D. Ability to Solve Problems that have no known Solutions

Possibly, genetic algorithms find their profoundest edge in the ability to address problems for with no known human experts. Despite the need for the application of human expertise when available, studies have revealed its gross inadequacy in the crucial roles of automating problem-solving routines. Such expert systems have well-known challenges such as experts' likelihood of being in disagreement; likelihood of lack in self-consistency; possibility of not being qualified, or simply being in error. Artificial intelligence has seen research fragmented into techniques and tricks targeted at unraveling specific constraints in restricted domains of interest. While such techniques have reportedly recorded successes in their applications to specific problems in, for instance, the chess program, "Deep Blue", most of them yet require human expertise. Despite their impressive performances in applications to difficult problems in need of limitless computational speed, they generally do not advance human understanding of intelligence. Their problem-solving abilities notwithstanding, they do not solve the problem of how to solve problems [52]. Nonetheless, genetic algorithm in attempting to provide a technique for solving the problem of how to solve problems, simply recapitulates the scientific technique [50], applicable to learning fundamental aspects of any measurable environment.

E. Outperform Classic Methods on Real Problems

Often, practical function optimization problems:

(1) Impose nonlinear constraints,

(2) Require payoff functions not concerned with least squared error,

(3) Involve non-stationary conditions,

(4) Incorporate noisy observations or random processing, or include other vagaries that do not conform well to the prerequisites of classic optimization techniques. Often multimodal, are the response surfaces posed in practical problems which require gradient-based techniques to rapidly converge on local optima (or perhaps saddle points) which may yield insufficient performance. Where the response surface is, for instance, strongly convex, for simpler problems, genetic algorithms fail to match traditional optimization methods in performance level [53]. This, notwithstanding, is to be expected, seeing as the techniques were fore mostly designed to take advantage of the convex property of such surfaces. In a series of empirical comparisons [54] has demonstrated that genetic algorithms offer a significant merit, compared to the obverse condition of applying classical methods to multi-modal functions. This, by extension, offers an almost definitely incorrect outcome in the often encountered case of applying linear programming to problems with nonlinear constraints. The reason being that the assumptions required for the technique are violated. In contrast, genetic algorithm computation can directly incorporate arbitrary constraints [55]. Moreover, the problem of defining the payoff function for optimization lies at the heart of success or failure: Inappropriate descriptions of the performance index lead to generating the right answer for the wrong problem. Within classic statistical techniques, concern is often devoted to minimizing the squared error

between forecast and actual data. But in practice, equally correct predictions are not of equal worth, and errors of identical magnitude are not equally costly. Consider the case of correctly predicting that a particular customer will ask to purchase 10 units of a particular product (e.g., an aircraft engine). This is typically worth less than correctly predicting that the customer will seek to purchase 100 units of that product, yet both predictions engender zero error and are weighted equally in classic statistics. Further, the error of predicting the customer will demand 10 units and having them actually demand 100 units is not of equal cost to the manufacturer as predicting the customer will demand 100 units and having them demand 10. One error leaves a missed opportunity cost while the other leaves 90 units in a warehouse. Yet again, under a squared error criterion, these two situations are treated identically. In contrast, within genetic algorithm, any definable payoff function can be used to judge the appropriateness of alternative behaviors. There is no restriction that the criteria be differentiable, smooth, or continuous.

2.4.2 Limitations of Genetic Algorithm

However, genetic algorithms also have some limitations. The formulation of fitness function, the use of population size, the choice of the important parameters such as the rate of mutation and crossover, and the selection criteria of the new population should be carried out rigorously. Any unsuitable choice will either make it difficult for the algorithm to converge prematurely, or it will simply produce meaningless results. Although genetic algorithm is a powerful optimization tool, it does have certain weaknesses and limitations. The randomness of genetic algorithm operation makes it difficult to predict its performance, a factor that is vital for hard-deadline, real-time application. The source of problem lies in the diversity of the chromosomes that cause online system execution to be uncertain. Some of the limitations of genetic algorithm are therefore highlighted below:

1. No assurance of convergence: There is no assurance that genetic algorithm will converge to a global optimum in a given optimization problem. There is a possibility that it gets stuck in one of the local optima. This is the reason genetic algorithms cannot be used to solve real-time problems where the accuracy and validity of the solution cannot be compromised.
2. Difficult parameter tuning: Any implementation of genetic algorithms will require the specification of various parameters, such as population size, mutation rate, and maximum run time, as well as the design of selection, recombination, and mutation procedures. Finding effective choices for these is in itself, a difficult problem with little or no theoretical support. In practice, researchers must rely on any available anecdotal reports from related problems, and lots of trial and error techniques.

2.4.3 Applications of Genetic Algorithm

Genetic algorithms have been applied for difficult problems (such as combined heat and power economic dispatch problems), for machine learning and also for evolving simple programs. Here we discuss concisely, some real life application examples of genetic algorithms. These application examples are conducted in the context of real-life problems. Some important applications are in economics, engineering design problems, encryption and code breaking etc. The following are some applications of genetic algorithms:

a) Optimizing Chemical Kinetic Analysis: Genetic algorithms are proving very useful towards optimizing designs problems in transportation, aerospace propulsion and electrical generation. By being able to predict ahead of time, the chemical kinetics of fuels and the efficiency of engines, more optimal mixtures and designs can be quickly made available to industries and the public. Some computer modelling applications in this area also simulate the effectiveness of lubricants, can pinpoint optimized operational vectors, and may lead to greatly increased efficiency all around, well before traditional fuels run out.

b) Encryption and Code Breaking: On the security front, genetic algorithms can be applied both to create encryption for sensitive data as well as to break those codes. Since the advent of computers, encrypting data, protecting copyrights and breaking rival codes have become vital, so the competition is intense. Every time someone adds more complexity to their encryption algorithms, someone else comes up with a genetic algorithm that can break the code. It is hoped that someday soon, we will have quantum computers that will be able to produce completely indecipherable codes [56].

c) Trip Traffic and Shipment Routing: New applications of a genetic algorithm known as the Traveling Salesman Problem can be applied to plan the most efficient routes and scheduling for travel planners, traffic routers and even shipping companies. The genetic algorithm gives shortest routes for traveling, timing to avoid traffic gridlocks and rush hours. It provides the most efficient use of transport for shipping, including pickup loads and deliveries along the way. The program models all this in the background and improves productivity, while the human agents do other things.

d) Engineering Design: Getting the most out of a range of materials to optimize the structural and operational designs of buildings, factories, machines, etc., is a rapidly expanding application of genetic algorithms. These are being created for such uses as optimizing the design of heat exchangers, robot gripping arms, satellite booms, building trusses, flywheels, turbines, and just about any other computer-assisted engineering design application. There is work to combine genetic algorithms optimizing particular aspects of engineering problems to work together. Some of which may not only solve design problems, but also project them forward to analyze weaknesses and possible point failures in the future, for avoidance of such problems.

e) Economic Modelling: Author [57] applied a genetic algorithm with a weighted goal programming technique to optimize a fishery bio-economic model. Bio-economic models have been developed for a number of fisheries as a means of estimating the optimal level of exploitation of the resource and for assessing the effectiveness of the different management plans available.

2.5 CONCLUSION

Reviewed literature on economic dispatch, combined heat and power (cogeneration) and genetic algorithm all had similar considerations and conclusions. There were some contradictions between the findings, such as some authors concluding that the only efficient means for constraint handling of feasible operation region was to apply a penalty factor to regulate solution. But the proposed algorithm constraint handling mechanisms got rid of this technique. This became the case as extraction of information is not possible with the strategy and its rejection of individuals who do not satisfy the constraints. Nevertheless, this strategy is not suitable for a discontinuous search space. The basic concept of genetic algorithm is designed to simulate processes in natural system necessary for evolution, specifically those that follow Charles Darwin's principles of survival of the fittest. As such, they represent an intelligent exploitation of a random search within a defined search space to solve a problem [51]. Genetic algorithm and optimization problems have been extensively studied, experimented and applied in many fields within the engineering world. This research on combined heat and power economic dispatch indicates that genetic algorithm result is satisfactory compared to direct solution algorithm, particle warm optimization algorithm, artificial bee colony algorithm; but not better than differential evolution algorithm for this class of optimization problem.

The proposed algorithm had better convergence speed, naturally diverse, had a robust number of solutions with least objective function value, compared with particle swarm optimization and artificial bee colony (which have a considerably large number of iterationsat103 and 100 respectively). The convergence speed of latter algorithms usually deteriorates with large number of test system in a power plant. There is also an inherent large cost function values associated with latter algorithms. Furthermore, it is easy to comprehend the amount of computation involved in each step of the genetic algorithm which is considerably less than that required for one iteration of both artificial bee colony and particle swarm optimization algorithms). Conversely, the effectiveness/efficacy of direct solution method when power system has large units is not yet known.

Therefore, the novelty of this research work lies in the overall search mechanism of the proposed algorithm and the evolution of the population. The result improvements compared to existing methods in other literatures are significant. It was observed that the proposed (genetic algorithm) can converge to produce diverse and widespread solutions along with extremely better solutions due to its effective searching capability. As a result, genetic algorithm can be a viable alternative for solving this class of combined heat and power economic dispatch problem. It also can considerably save fuel cost. As a future work, effective tuning of parameter will be carried out with sensitivity analysis (Sensitivity analysis is a technique for determining how the model output is affected by the uncertainties of inputs) of parameter; and its impact on solution will be analyzed.

CHAPTER 3 - MATERIALS AND METHOD

3.0 INTRODUCTION

In this section, we present the methodology for the research objectives (to determine the combined heat and power economic dispatch decision variables using genetic, particle swarm optimization and direct solution algorithms. Thereafter, we find how the algorithms arrived at their minimum cost functions with best decision variables. For the objective and associated research questions, we introduce the mathematical model to compute the combined heat and power economic dispatch decision variables using direct solution algorithm. We also explain what happens when the calculated output decision variables stretch the operational bounds of conventional power units, the co-generation unit and heat unit by using the mathematical framework only. We then solve the combined heat and power economic dispatch problem automatically, by implementing the optimization problem using genetic and particle swarm optimization algorithms.

3.1 Model development/Formulation

Combined heat and power Economic dispatch problems are constraint optimization problems, which consist of decision variables i.e. (heat, power dispatch values) and objective function. The objective function indicates how much each decision variable contributes to the value to be optimized in the problem statement and its duty is to minimize the total generation cost in a system that consists of the conventional thermal power and heat units plus the cogeneration unit with feasible operation region. The two power units, a cogeneration unit and a heat-only unit in the research have quadratic cost functions. The limit on the outputs of the co-generation unit is specified by listing the co-ordinates of the corners of the feasible operating region of the unit as shown in table 3.4.

The objective function also represents the input fuel cost while the constraints are inequality, equality and other operational constraints that match load and heat demands with power generation. In this research, the system transmission losses were neglected, leaving power and heat loads plus the machine operation bound as the only available constraints. The research first shows that it is possible to solve combined heat and power economic dispatch problem using direct method. In the cause of the research, whenever we use combined heat and power economic dispatch, we imply combined heat and power economic dispatch of simple cycle cogeneration units having quadratic cost functions. Also, artificial bee colony, genetic, particle swarm optimization and differential evolution algorithms were employed to this class of optimization problem. After this, results from the various algorithms were compared to determine the algorithm with optimal result and best operational costs.

We initially developed a formula for the system lambdas to correspond with the power and heat demands in terms of the coefficients of the generator cost functions. The formula was developed with the most common form of this problem in mind, assuming that the cost functions are quadratic. The unit output corresponding with these system lambda values, constituted the required heat and power dispatch, provided none of the unit outputs hit their limits. In order to handle situations where some of the outputs corresponding to the computed system Lagrangian multiplier λ-values happen to violate their limits, we developed a simple scheme for setting the outputs of such units at appropriate limits. Setting some units at their limits could result in mismatches between the demand and the generation. We recalculate the system λ, for the units not set at their limits in order to eliminate mismatch. To calculate combined heat and power economic dispatch decision variables directly, the problem statement was modelled in order to determine combined heat and power economic dispatch decision variables. The Combined heat and power economic dispatch problem is modeled thus:

3.2 COMBINED HEAT AND POWER DISPATCH PROBLEM

Given the quadratic fuel cost function of power-only, cogeneration and heat-only units in Naira we have:

$c_{e,i}\left(p_i\right)=\alpha_i+\beta_i p_i+\gamma_i p_i^2$ Cost function of power only unit

$c_{c,i}\left(p_i,q_i\right)=\alpha_i+\beta_i p_i+\gamma_i p_i^2+\delta_i q_i+\varepsilon_i q_i^2+\zeta_i p_i q_i$ Cost function of cogeneration unit (3.1)

$c_{h,i}\left(q_i\right)=\alpha_i+\delta_i q_i+\varepsilon_i q_i^2$ Cost function of heat only unit

Where, α_i,β_i and γ_i are the cost coefficient of i^{th} power-only unit, $\alpha_j,\beta_j,\gamma_j,\delta_j,\varepsilon_j$ and ζ_j are the cost coefficients for the j^{th} cogeneration unit, α_k,δ_k and ε_k represent the coefficient of k^{th} heat-only unit. The objective function of the combined heat and power economic dispatch problem is to minimize the cost function, subject to equality, inequality and other operational constraints.

The objective function of the combined heat and power economic dispatch problem can be stated thus:

$$\text{Min } C=\sum_{i\in e}c_{e,i}\left(p_i\right)+\sum_{i\in c}c_{c,i}\left(p_i,q_i\right)+\sum_{i\in h}c_{h,i}\left(q_i\right) \tag{3.2}$$

Q and *P* are the heat and electrical power output decision variables of the units, respectively. $c_{e,i}\left(p_i\right)$, $c_{c,j}\left(p_j,q_j\right)$ and $c_{h,k}\left(q_k\right)$ constitute the fuel cost function of i^{th} power-only unit, fuel cost function of j^{th} cogeneration unit and fuel cost function of k^{th} heat-only unit.

Subject to real power generated by power unit, plus the real power generated by cogeneration unit is equal to the real power demand of the power systems neglecting power loss. This is stated mathematically in equation (3.3) below:

$$\sum_{i\in e} p_i + \sum_{i\in c} p_i = p^{demand} \tag{3.3}$$

Comparably, the total heat created by boilers plus the active heat created by cogeneration units is equal to the heat demand, abandoning heat loss, and can be stated thus:

$$\sum_{i\in e} q_i + \sum_{i\in h} q_i = q^{demand} \tag{3.4}$$

Where p^{demand} and q^{demand} are the total heat and power demand of system, respectively. In the heat equality constraint, heat losses are postulated to be zero because no research work about heat losses during process of transmitting heat to heat loads has been carried out. For clarity, that postulation was employed in this research. Therefore, heat losses are negligible. Furthermore, if heat losses are a function of heat outputs similar to power loss function or a constant, heat balance constraint will be solved simply and successfully.

$$p_i^{\min} \le p_i \le p_i^{\max} \tag{3.5}$$

$$q_i^{\min} \le q_i \le q_i^{\max} \tag{3.6}$$

We note that equation (3.3) and (3.4) can be written in the form of two inequality constraints:

$$\sum_{i\in e} p_i \le p^{demand} - \sum_{i\in c} p_i \tag{3.7a}$$

$$-\sum_{i\in e} p_i \le -p^{demand} + \sum_{i\in c} p_i \tag{3.7b}$$

and

$$\sum_{i\in e} q_i \le q^{demand} - \sum_{i\in h} q_i \tag{3.8a}$$

$$-\sum_{i\in e} q_i \le -q^{demand} + \sum_{i\in h} q_i \tag{3.8b}$$

Thus, formally, the optimization problem to be solved is

$$\text{Min } C = \sum_{i\in e} c_{e,i}\left(p_i\right) + \sum_{i\in c} c_{c,i}\left(p_i, q_i\right) + \sum_{i\in h} c_{h,i}\left(q_i\right) \tag{3.9}$$

Subject to

$$\sum_{i\in e} p_i \le p^{demand} - \sum_{i\in c} p_i \tag{3.10a}$$

$$-\sum_{i\in e} p_i \le -p^{demand} + \sum_{i\in c} p_i \qquad (3.10b)$$

$$\sum_{i\in e} q_i \le q^{demand} - \sum_{i\in h} q_i \qquad (3.11a)$$

$$-\sum_{i\in e} q_i \le -q^{demand} + \sum_{i\in h} q_i \qquad (3.11b)$$

$$p_i^{\min} \le p_i \le p_i^{\max} \qquad (3.12)$$

$$q_i^{\min} \le q_i \le q_i^{\max} \qquad (3.13)$$

$$a_{ij} * p_i + b_{ij} * q_i \ge c_{ji} \quad j = 1, \mathrm{K}, n_i \qquad (3.14)$$

The output of the cogeneration unit is presumed to lie in a region in the P_i– Q_i plane bounded by n_i lines. These lines are illustrated in equation (3.14).

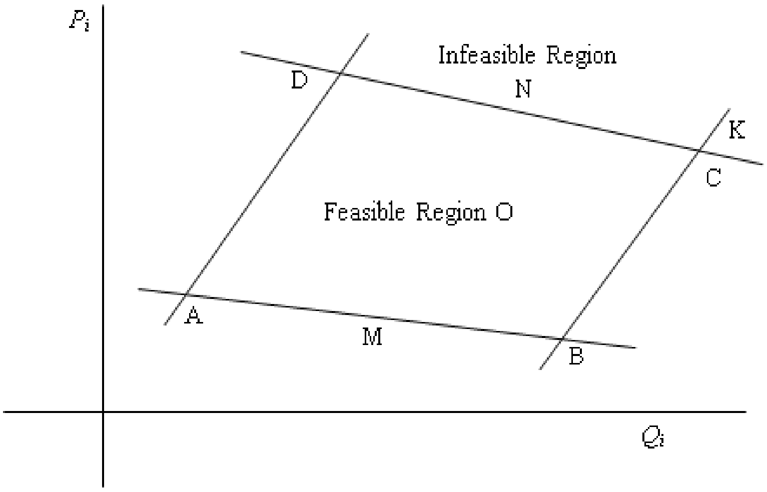


Fig3.1: Feasible operating region of a cogeneration unit

Figure 3.1 is a typical polyhedron of a feasible region (search space) of a cogeneration unit bounded by four hyper-plane lines in the *P*– *Q* plane. In such a plane, there are three kinds of operating points of a cogeneration unit. At point O (feasible region or solution space), the unit is not bounded by any constraints. At points N and M, the unit is bounded by only one constraint; and at point K, the unit is bounded by two constraints. For a point in the above figure 3.1 to be feasible, it should be above line AB, below line CD, right of AD, and left of BC. Any contrary arrangement amounts to an infeasible region. To determine if a point is feasible or not, we substitute the value of [*p*, *q*] inside equation of a line formed between two vertices connected by a line segment. For any value of [*p*, *q*], substituting the value of the function maybe zero, positive or negative. The operating point is positive when the value is in the region O, zero when the value is on the line of the quadrilateral (ABCD) and negative when it is either at positions M, N and K or infeasible region. The problems described between (3.10) to (3.14), are optimization problems of equality and inequality constraints that demand the deployment of the Karush-Kuhn-Tucker (KKT) optimality conditions.

The Karush-Kuhn-Tucker (KKT) Lagrange multiplier for the dispatch problem given is

$$L=\sum_{i\in e}c_{e,i}\left(p_i\right)+\sum_{i\in c}c_{c,i}\left(p_i,q_i\right)+\sum_{i\in h}c_{h,i}\left(q_i\right)-\lambda_p\left(\sum_{i\in e}p_i+\sum_{i\in c}p_i-p^{demand}\right)-\lambda_q\left(\sum_{i\in e}q_i+\sum_{i\in h}q_i-q^{demand}\right) \tag{3.15}$$

Where λ_p and λ_q represent the Lagrangian multipliers associated with the constraints.

The Karush-Kuhn-Tucker (KKT) necessary optimality conditions for the above problem are:

$$\frac{\partial L}{\partial p_i}=\frac{\partial L}{\partial q_i}=\frac{\partial L}{\partial \lambda_p}=\frac{\partial L}{\partial \lambda_q}=0 \tag{3.16}$$

$$\begin{aligned}\frac{\partial L}{\partial p_i}&=\sum_{i\in e}\frac{\partial}{\partial p_i}c_{e,i}\left(p_i\right)-\lambda_p\sum_{i\in e}\frac{\partial}{\partial p_i}p_i+\sum_{i\in c}\frac{\partial}{\partial p_i}c_{c,i}\left(p_i,q_i\right)-\lambda_p\sum_{i\in c}\frac{\partial}{\partial p_i}p_i\\&=\sum_{i\in e}\left(\frac{\partial}{\partial p_i}c_{e,i}\left(p_i\right)-\lambda_p\right)+\sum_{i\in c}\left(\frac{\partial}{\partial p_i}c_{c,i}\left(p_i,q_i\right)-\lambda_p\right)=0\end{aligned} \tag{3.17}$$

$$\Rightarrow\quad \frac{\partial}{\partial p_i}c_{e,i}\left(p_i\right)-\lambda_p=0\quad \forall i\in e,\quad \frac{\partial}{\partial p_i}c_{ci}\left(p_i,q_i\right)-\lambda_p=0\quad \forall i\in c \tag{3.18}$$

Similarly

$$\begin{aligned}\frac{\partial L}{\partial q_i}&=\sum_{i\in c}\frac{\partial}{\partial q_i}c_{c,i}\left(p_i,q_i\right)-\lambda_q\sum_{i\in c}\frac{\partial}{\partial q_i}q_i+\sum_{i\in h}\frac{\partial}{\partial q_i}c_{h,i}\left(q_i\right)-\lambda_q\sum_{i\in h}\frac{\partial}{\partial q_i}q_i\\&=\sum_{i\in c}\left(\frac{\partial}{\partial q_i}c_{c,i}\left(p_i,q_i\right)-\lambda_q\right)+\sum_{i\in h}\left(\frac{\partial}{\partial q_i}c_{h,i}\left(q_i\right)-\lambda_q\right)=0\end{aligned} \tag{3.19}$$

$$\Rightarrow\quad \frac{\partial}{\partial q_i}c_{c,i}\left(p_i,q_i\right)-\lambda_q=0\quad \forall i\in c,\quad \frac{\partial}{\partial q_i}c_{h,i}\left(q_i\right)-\lambda_q=0\quad \forall i\in h \tag{3.20}$$

Furthermore;

$$\frac{\partial L}{\partial \lambda_p}=\sum_{i\in e}p_i+\sum_{i\in c}p_i-p^{demand}=0\quad\Rightarrow\quad \sum_{i\in e}p_i+\sum_{i\in c}p_i=p^{demand} \tag{3.21}$$

$$\frac{\partial L}{\partial \lambda_q}=\sum_{i\in e}q_i+\sum_{i\in h}q_i-q^{demand}=0\quad\Rightarrow\quad \sum_{i\in e}q_i+\sum_{i\in h}q_i=q^{demand} \tag{3.22}$$

Hence, the equations to be solved are:

$$\frac{\partial}{\partial p_i} c_{e,i}\left(p_i\right) - \lambda_p = 0 \qquad \forall i \in e \tag{3.23}$$

$$\frac{\partial}{\partial p_i} c_{c,i}\left(p_i, q_i\right) - \lambda_p = 0 \qquad \forall i \in c \tag{3.24}$$

$$\frac{\partial}{\partial q_i} c_{c,i}\left(p_i, q_i\right) - \lambda_q = 0 \qquad \forall i \in c \tag{3.25}$$

$$\frac{\partial}{\partial q_i} c_{h,i}\left(q_i\right) - \lambda_q = 0 \qquad \forall i \in h \tag{3.26}$$

$$\sum_{i \in e} p_i + \sum_{i \in c} p_i = p^{demand} \tag{3.27}$$

$$\sum_{i \in c} q_i + \sum_{i \in h} q_i = q^{demand} \tag{3.28}$$

With $c_{e,i}\left(p_i\right) = \alpha_i + \beta_i p_i + \gamma_i p_i^2$, we have

$$\frac{\partial}{\partial p_i} c_{e,i}\left(p_i\right) - \lambda_p = 0 \quad \Rightarrow \quad \beta_i + 2\gamma_i p_i - \lambda_p = 0 \qquad \forall i \in e \tag{3.29}$$

With $c_{c,i}\left(p_i, q_i\right) = \alpha_i + \beta_i p_i + \gamma_i p_i^2 + \delta_i q_i + \varepsilon_i q_i^2 + \zeta_i p_i q_i$, we have

$$\frac{\partial}{\partial p_i} c_{c,i}\left(p_i, q_i\right) - \lambda_p = 0 \quad \Rightarrow \quad 2\gamma_i p_i + \zeta_i q_i + \beta_i - \lambda_p = 0 \quad \forall i \in c \tag{3.30}$$

$$\frac{\partial}{\partial q_i} c_{c,i}\left(p_i, q_i\right) - \lambda_q = 0 \quad \Rightarrow \quad 2\varepsilon_i q_i + \zeta_i p_i + \delta_i - \lambda_q = 0 \quad \forall i \in c \tag{3.31}$$

With $c_{h,i}\left(q_i\right) = \alpha_i + \delta_i q_i + \varepsilon_i q_i^2$, we have

$$\frac{\partial}{\partial q_i} c_{h,i}\left(q_i\right) - \lambda_q = 0 \quad \Rightarrow \quad 2\varepsilon_i q_i + \delta_i - \lambda_q = 0 \qquad \forall i \in h \tag{3.32}$$

Hence the resulting equations to be solved for p_i, q_i, λ_p and λ_q are,

$$\beta_i + 2\gamma_i p_i - \lambda_p = 0 \qquad \forall i \in e \tag{3.33}$$

$$2\gamma_i p_i + \zeta_i q_i + \beta_i - \lambda_p = 0 \quad \forall i \in c \tag{3.34}$$

$$2\varepsilon_i q_i + \zeta_i p_i + \delta_i - \lambda_q = 0 \quad \forall i \in c \tag{3.35}$$

$$2\varepsilon_i q_i + \delta_i - \lambda_q = 0 \qquad \forall i \in h \tag{3.36}$$

In equations (3.33) and (3.36) we make p_i and q_i the subject to get respectively;

$$p_i = -\beta_i / 2\gamma_i + \lambda_p / 2\gamma_i \quad \forall i \in e \qquad (3.37)$$

$$q_i = -\delta_i / 2\varepsilon_i + \lambda_q / 2\varepsilon_i \quad \forall i \in h \qquad (3.38)$$

We now solve equations (3.34) and (3.35) simultaneously. For this purpose, we recast them in matrix. First the equations are rewritten;

$$2\gamma_i p_i + \zeta_i q_i = \lambda_p - \beta_i \qquad (3.39)$$

$$\zeta_i p_i + 2\varepsilon_i q_i = \lambda_q - \delta_i \qquad (3.40)$$

In matrix form we have:

$$\begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix} \begin{bmatrix} p_i \\ q_i \end{bmatrix} = \begin{bmatrix} \lambda_p - \beta_i \\ \lambda_q - \delta_i \end{bmatrix} \qquad (3.41)$$

or

$$\begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix} \begin{bmatrix} p_i \\ q_i \end{bmatrix} = \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} - \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} \qquad (3.42)$$

Multiply equation (3.33) by $\begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1}$. It then follows that;

$$\begin{bmatrix} p_i \\ q_i \end{bmatrix} = -\begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} + \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} i \in c \qquad (3.43)$$

Substituting (3.39) for $i \in e$ and the p_i in equation (3.43) for $i \in c$ in equation (3.4).

Similarly substituting (3.40) for $i \in h$ and the q_i in equation (3.43) for $i \in c$ in equation (3.5).

For simplification let

$$\Lambda = \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \qquad (3.44)$$

Hence, equation (3.43) becomes:

$$\begin{bmatrix} p_i \\ q_i \end{bmatrix} = -\Lambda \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} + \Lambda \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} \qquad (3.45)$$

$$\Rightarrow \quad p_i = -\Lambda\beta_i + \Lambda\lambda_p, \quad q_i = -\Lambda\delta_i + \Lambda\lambda_q \; i \in c \qquad (3.46)$$

Next we consider equation (3.3) written

$$\sum_{i \in e} p_i + \sum_{i \in c} p_i = p^{demand} \qquad (3.47)$$

Remembering that equation (3.46) is:

$$p_i = -\beta_i / 2\gamma_i + \lambda_p / 2\gamma_i \quad \forall i \in e$$

Hence, the first component of (3.46) is given by:

$$\sum_{i\in e} p_i = \sum_{i\in e}\left(-\beta_i/2\gamma_i + \lambda_p/2\gamma_i\right) = -\sum_{i\in e}\beta_i/2\gamma_i + \lambda_p\sum_{i\in e}1/2\gamma_i \qquad (3.48)$$

The second component of (3.46) is:

$$\sum_{i\in c} p_i = \sum_{i\in c}\left(-\Lambda\beta_i + \Lambda\lambda_p\right) = -\sum_{i\in c}\Lambda\beta_i + \lambda_p\sum_{i\in c}\Lambda \qquad (3.49)$$

Hence equation (3.46) becomes:

$$-\sum_{i\in e}\beta_i/2\gamma_i + \lambda_p\sum_{i\in e}1/2\gamma_i - \sum_{i\in c}\Lambda\beta_i + \lambda_p\sum_{i\in c}\Lambda = p^{demand} \qquad (3.50)$$

Next we consider equation (3.4) written:

$$\sum_{i\in c} q_i + \sum_{i\in h} q_i = q^{demand} \qquad (3.51)$$

Remembering from equation (3.46), $q_i = -\Lambda\delta_i + \Lambda\lambda_q$, $i \in c$. Hence, the first component of (3.51) is given by:

$$\sum_{i\in c} q_i = \sum_{i\in c}\left(-\Lambda\delta_i + \Lambda\lambda_q\right) = -\sum_{i\in c}\Lambda\delta_i + \lambda_q\sum_{i\in c}\Lambda \qquad (3.52)$$

Also, we recall that equation (3.40) is:

$$q_i = -\delta_i/2\varepsilon_i + \lambda_q/2\varepsilon_i \qquad \forall i \in h$$

Hence, the second component of equation (3.51) becomes:

$$\sum_{i\in h} q_i = \sum_{i\in h}\left(-\delta_i/2\varepsilon_i + \lambda_q/2\varepsilon_i\right) = -\sum_{i\in h}\delta_i/2\varepsilon_i + \lambda_q\sum_{i\in h}1/2\varepsilon_i \qquad (3.53)$$

Equation (3.52) is:

$$-\sum_{i\in c}\Lambda\delta_i + \lambda_q\sum_{i\in c}\Lambda - \sum_{i\in h}\delta_i/2\varepsilon_i + \lambda_q\sum_{i\in h}1/2\varepsilon_i = q^{demand} \qquad (3.54)$$

We will now put equations (3.50) and (3.54) in matrix form and solve for λ_p and λ_q .

We thus have:

$$-\sum_{i\in e}\beta_i/2\gamma_i + \lambda_p\sum_{i\in e}1/2\gamma_i - \sum_{i\in c}\Lambda\beta_i + \lambda_p\sum_{i\in c}\Lambda = p^{demand} \qquad (3.55)$$

$$-\sum_{i\in e}\beta_i/2\gamma_i + \lambda_p\sum_{i\in e}1/2\gamma_i - \sum_{i\in c}\Lambda\beta_i + \lambda_p\sum_{i\in c}\Lambda = p^{demand} \qquad (3.56)$$

The above equations can be rearranged as:

$$\lambda_p\left(\sum_{i\in e}1/2\gamma_i + \sum_{i\in c}\Lambda\right) = p^{demand} + \sum_{i\in e}\beta_i/2\gamma_i + \sum_{i\in c}\Lambda\beta_i \qquad (3.57)$$

$$\lambda_q\left(\sum_{i\in c}\Lambda + \sum_{i\in h}1/2\varepsilon_i\right) = q^{demand} + \sum_{i\in c}\Lambda\delta_i + \sum_{i\in h}\delta_i/2\varepsilon_i \qquad (3.58)$$

While bearing in mind that:

$$\Lambda = \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1}$$

We observe that the left side of equations (3.57) and (3.58) are:

$$\lambda_p \left(\sum_{i \in e} 1/2\gamma_i + \sum_{i \in c} \Lambda \right) = \lambda_p \sum_{i \in e} 1/2\gamma_i + \lambda_p \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \tag{3.59}$$

$$\lambda_q \left(\sum_{i \in c} \Lambda + \sum_{i \in h} 1/2\varepsilon_i \right) = \lambda_q \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} + \lambda_q \sum_{i \in h} 1/2\varepsilon_i \tag{3.60}$$

On a closer observation, we see the incompatibility of the addition in (3.59) and (3.60) (*addition of scalar to a matrix*). The solution is to transform both the terms $\lambda_p \sum_{i \in e} 1/2\gamma_i$ and $\lambda_q \sum_{i \in h} 1/2\varepsilon_i$ into matrix form as follows:

$$\lambda_p \sum_{i \in e} 1/2\gamma_i = \begin{bmatrix} \sum_{i \in e} 1/2\gamma_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} = \sum_{i \in e} \begin{bmatrix} 1/2\gamma_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} \tag{3.61}$$

Similarly,

$$\lambda_q \sum_{i \in h} 1/2\varepsilon_i = \begin{bmatrix} 0 & 0 \\ 0 & \sum_{i \in h} 1/2\varepsilon_i \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} = \sum_{i \in h} \begin{bmatrix} 0 & 0 \\ 0 & 1/2\varepsilon_i \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} \tag{3.62}$$

Hence we rewrite equations (3.57) and (3.58) in matrix form as follows:

$$\begin{aligned} &\sum_{i \in e} \begin{bmatrix} 1/2\gamma_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 & 0 \\ 0 & 1/2\varepsilon_i \end{bmatrix} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} \\ &\qquad = \begin{bmatrix} p^{demand} \\ q^{demand} \end{bmatrix} + \sum_{i \in e} \begin{bmatrix} \beta_i/2\gamma_i \\ 0 \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 \\ \delta_i/2\varepsilon_i \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} \end{aligned} \tag{3.63}$$

Or,

$$\begin{aligned} &\left(\sum_{i \in e} \begin{bmatrix} 1/2\gamma_i & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 & 0 \\ 0 & 1/2\varepsilon_i \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \right) \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} \\ &\qquad = \begin{bmatrix} p^{demand} \\ q^{demand} \end{bmatrix} + \sum_{i \in e} \begin{bmatrix} \beta_i/2\gamma_i \\ 0 \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 \\ \delta_i/2\varepsilon_i \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} \end{aligned} \tag{3.64}$$

Let:

$$[A] = \sum_{i \in e} \begin{bmatrix} 1/2\gamma_i & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 & 0 \\ 0 & 1/2\varepsilon_i \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \tag{3.65}$$

Hence equation (3.64) becomes:

$$[A] \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} = \begin{bmatrix} p^{demand} \\ q^{demand} \end{bmatrix} + \sum_{i \in e} \begin{bmatrix} \beta_i/2\gamma_i \\ 0 \end{bmatrix} + \sum_{i \in h} \begin{bmatrix} 0 \\ \delta_i/2\varepsilon_i \end{bmatrix} + \sum_{i \in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix}$$

$$\Rightarrow \quad \begin{bmatrix} \lambda_p \\ \lambda_q \end{bmatrix} = [A]^{-1} \left\{ \begin{bmatrix} p^{demand} \\ q^{demand} \end{bmatrix} + \sum_{i\in e} \begin{bmatrix} \beta_i/2\gamma_i \\ 0 \end{bmatrix} + \sum_{i\in h} \begin{bmatrix} 0 \\ \delta_i/2\varepsilon_i \end{bmatrix} + \sum_{i\in c} \begin{bmatrix} 2\gamma_i & \zeta_i \\ \zeta_i & 2\varepsilon_i \end{bmatrix}^{-1} \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} \right\} \qquad (3.66)$$

3.3 IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM BY DIRECT SOLUTION ALGORITHM

3.3.1 Data Set for the New Combined Heat and Power Dispatch Problem

Table 3.1 Power units-cost coefficients

Units	PG$_{max}$	PG$_{min}$	α	β	γ
1	250	10	1000	13.5	0.0345
2	200	20	1245	13.1	0.033

Table3. 2 (Unit 3) Cogeneration unit-cost coefficients

Units	α	β	γ	δ	ε	ζ
1	2650	14.5	0.0345	4.2	0.03	0.011

Table -3.3 (Unit 4) Heat Unit Cost Coefficients

Units	α	δ	ε	Q_{max}	Q_{min}
1	1200	4.2	0.02	250	20

Table-3.4 Coordinate of the Corners of the Feasible Regions of the Co-generation Units

Corners	(p_1,q_1)	(p_2,q_2)	(p_3,q_3)	(p_4,q_4)
Unit 3	(20,0.1)	(200,0.5)	(195,120)	(15,110)

3.3.2 Combined Heat and Power Dispatch Problem Solution

From tables 3.1, 3.2, and 3.3 we have:

$$c_{e,1}(p_1) = \alpha_1 + \beta_1 p_1 + \gamma_1 p_1^2 \quad \Rightarrow \quad c_{e,1}(p_1) = 1000 + 13.5 p_1 + 0.0345 p_1^2$$

$$c_{e,2}(p_2) = \alpha_2 + \beta_2 p_2 + \gamma_2 p_2^2 \quad \Rightarrow \quad c_{e,2}(p_2) = 1245 + 13.1 p_2 + 0.033 p_2^2$$

$$c_{c,1}(p_3, q_3) = \alpha_1 + \beta_1 p_3 + \gamma_1 p_3^2 + \delta_1 q_3 + \varepsilon_1 q_3^2 + \zeta_1 p_3 q_3$$

$$\Rightarrow \quad c_{c,1}(p_3, q_3) = 2650 + 14.5 p_3 + 0.0345 p_3^2 + 4.2 q_1 + 0.03 q_3^2 + 0.011 p_3 q_3$$

$$c_{h,4}(q_4) = \alpha_1 + \delta_1 q_4 + \varepsilon_1 q_4^2 \quad \Rightarrow \quad c_{h,1}(q_4) = 1200 + 4.2 q_4 + 0.02 q_4^2$$

The optimization problem with the specified data is stated as;

$$\begin{aligned}\text{Min } C &= \sum_{i=1}^{2} c_{e,i}\left(p_i\right) + \sum_{i=1}^{1} c_{c,i}\left(p_i, q_i\right) + c_{h,1}\left(q_4\right) = c_{e,1}\left(p_1\right) + c_{e,2}\left(p_2\right) + c_{c,1}\left(p_3, q_3\right) + c_{h,1}\left(q_4\right) \\ &= \left(1000 + 13.5p_1 + 0.0345p_1^2\right) + \left(1245 + 13.1p_2 + 0.033p_2^2\right) \\ &\quad + \left(2650 + 14.5p_3 + 0.0345p_3^2 + 4.2q_3 + 0.03q_3^2 + 0.011p_3q_3\right) \\ &\quad + \left(1200 + 4.2q_4 + 0.02q_4^2\right)\end{aligned} \tag{3.67}$$

Subject

$$p_1 + p_2 + p_3 = 520 \tag{3.68}$$

$$q_3 + q_4 = 300 \tag{3.69}$$

$$\begin{aligned} p_1^{\min} \le p_1 \le p_1^{\max} \quad &\Rightarrow \quad 10 \le p_1 \le 250 \\ p_2^{\min} \le p_2 \le p_2^{\max} \quad &\Rightarrow \quad 20 \le p_2 \le 200 \end{aligned} \tag{3.70}$$

$$q_4^{\min} \le q_4 \le q_4^{\max} \quad \Rightarrow \quad 20 \le q_4 \le 250 \tag{3.71}$$

The Karush-Kuhn-Tucker (KKT) Lagrange multiplier for the dispatch problem given is

$$\begin{aligned} L &= \left(1000 + 13.5p_1 + 0.0345p_1^2\right) + \left(1245 + 13.1p_2 + 0.033p_2^2\right) \\ &\quad + \left(2650 + 14.5p_3 + 0.0345p_3^2 + 4.2q_3 + 0.03q_3^2 + 0.011p_3q_3\right) + \left(1200 + 4.2q_4 + 0.02q_4^2\right) \\ &\quad - \lambda_p\left(p_1 + p_2 + p_3 - 520\right) - \lambda_q\left(q_3 + q_4 - 410\right) \end{aligned} \tag{3.72}$$

Where λ_p and λ_q are the so called *Lagrange multipliers* associated with the constraints.

The Karush-Kuhn-Tucker (KKT) necessary optimality conditions for the above problem are:

$$\frac{\partial L}{\partial p_1} = \frac{\partial L}{\partial p_2} = \frac{\partial L}{\partial p_3} = \frac{\partial L}{\partial q_3} = \frac{\partial L}{\partial q_4} = \frac{\partial L}{\partial \lambda_p} = \frac{\partial L}{\partial \lambda_q} = 0$$

Hence, we have:

$$\frac{\partial L}{\partial p_1} = 13.5 + 0.069p_1 - \lambda_p = 0 \tag{3.73}$$

$$\frac{\partial L}{\partial p_2} = 13.1 + 0.066p_2 - \lambda_p = 0 \tag{3.74}$$

$$\frac{\partial L}{\partial p_3} = 14.5 + 0.069p_3 + 0.011q_3 - \lambda_p = 0 \tag{3.75}$$

$$\frac{\partial L}{\partial q_3} = 4.2 + 0.06q_3 + 0.011p_3 - \lambda_q = 0 \tag{3.76}$$

$$\frac{\partial L}{\partial q_4} = 4.2 + 0.04q_4 - \lambda_q = 0 \tag{3.77}$$

$$\frac{\partial L}{\partial \lambda_p} = p_1 + p_2 + p_3 - 520 = 0 \tag{3.78}$$

$$\frac{\partial L}{\partial \lambda_q} = q_3 + q_4 - 300 = 0 \tag{3.79}$$

From equations (3.73), (3.74) and (3.75) we obtain;

$$p_1 = \frac{\lambda_p - 13.5}{0.069} \tag{3.80}$$

$$p_2 = \frac{\lambda_p - 13.1}{0.066} \tag{3.81}$$

$$p_3 = \frac{\lambda_p - 0.011q_3 - 14.5}{0.069} \tag{3.82}$$

From (3.78) we have

$$p_1 + p_2 + p_3 - 520 = 0$$

$$\Rightarrow \qquad \frac{\lambda_p - 13.5}{0.069} + \frac{\lambda_p - 13.1}{0.066} + \frac{\lambda_p - 0.011q_3 - 14.5}{0.069} = 520 \tag{3.83}$$

$$\therefore \qquad 44.137\lambda_p - 0.159q_3 = 1124.282 \tag{3.84}$$

From (3.79)

$$q_3 = 300 - q_4 \tag{3.85}$$

Substituting in (3.84), we get

$$44.137\lambda_p - 0.159\left(300 - q_4\right) = 1124.282$$

$$\Rightarrow \qquad 44.137\lambda_p + 0.159q_4 = 1172 \tag{3.86}$$

From (3.77) we have

$$q_4 = \frac{\lambda_q - 4.2}{0.04}$$

Substituting in (3.86), we get;

$$44.137\lambda_p + 0.159\left(\frac{\lambda_q - 4.2}{0.04}\right) = 1172$$

$$\Rightarrow \qquad 44.137\lambda_p + 3.975\lambda_q = 1189 \tag{3.87}$$

From (3.76) we have

$$4.2 + 0.06q_3 + 0.011p_3 - \lambda_q = 0 \tag{3.88}$$

We note that (3.85) is equivalent to

$$q_3 = 410 - \left(\frac{\lambda_q - 4.2}{0.04}\right) \tag{3.89}$$

Also from (3.82)

$$p_3 = \frac{\lambda_p - 0.011q_3 - 14.5}{0.069}$$

Substituting for p_3 in (3.88) we get;

$$4.2 + 0.06q_3 + 0.011\left(\frac{\lambda_p - 0.011q_3 - 14.5}{0.069}\right) - \lambda_q = 0$$

$$\Rightarrow \qquad \lambda_q - 0.1594\lambda_p - 0.0583q_3 = 1.888 \tag{3.90}$$

We now substitute for q_3 in (3.90) to get;

$$\lambda_q - 0.1594\lambda_p - 0.0583\left[410 - \left(\frac{\lambda_q - 4.2}{0.04}\right)\right] = 1.888$$

$$2.5438\lambda_q + 0.1594\lambda_p = 36.003 \tag{3.91}$$

We will now solve equations (3.87) and (3.91) using matrix method. Thus the simultaneous equations;

$$44.137\lambda_p + 3.975\lambda_q = 1189$$

$$0.1594\lambda_p + 2.5438\lambda_q = 36.003$$

Can be recast in matrix form as follows.

$$\begin{pmatrix} 44.137 & 3.975 \\ 0.1594 & 2.5438 \end{pmatrix}\begin{pmatrix} \lambda_p \\ \lambda_q \end{pmatrix} = \begin{pmatrix} 1189 \\ 36.003 \end{pmatrix} \tag{3.92}$$

$$\begin{pmatrix} \lambda_p \\ \lambda_q \end{pmatrix} = \begin{pmatrix} 44.137 & 3.975 \\ 0.1594 & 2.5438 \end{pmatrix}^{-1}\begin{pmatrix} 1189 \\ 36.003 \end{pmatrix}$$

$$= \begin{pmatrix} 0.023 & -0.036 \\ -1.428\times10^{-3} & 0.395 \end{pmatrix}\begin{pmatrix} 1189 \\ 36.003 \end{pmatrix} = \begin{pmatrix} 25.81 \\ 12.536 \end{pmatrix}$$

$$\therefore \qquad \begin{pmatrix} \lambda_p \\ \lambda_q \end{pmatrix} = \begin{pmatrix} 25.81 \\ 12.536 \end{pmatrix}$$

Hence, $\lambda_p = 25.81$ and $\lambda_q = 12.536$. From (3.89) we have;

$$q_3 = 300 - \left(\frac{\lambda_q - 4.2}{0.04}\right) = 300 - \left(\frac{12.536 - 4.2}{0.04}\right) = 91.6 \tag{3.93}$$

$$q_4 = 300 - 91.6 = 208.4$$

From (3.77) we also have;

$$q_4 = \frac{\lambda_q - 4.2}{0.04} = \frac{14.658 - 4.2}{0.04} = 208.4 \tag{3.94}$$

From (3.13) and (3.14)

$$p_1 = \frac{\lambda_p - 13.5}{0.069} = \frac{25.81 - 13.5}{0.069} = 178.406 \quad (3.95)$$

$$p_2 = \frac{\lambda_p - 13.1}{0.066} = \frac{25.81 - 13.1}{0.066} = 192.576 \quad (3.96)$$

$$p_3 = \frac{\lambda_p - 0.011 q_3 - 14.5}{0.069} = \frac{25.81 - 0.011(91.6) - 14.5}{0.069} = 149.31 \quad (3.97)$$

Hence we have; $p_1 = 178.406,\ p_2 = 192.576,\ p_3 = 149.31,\ q_3 = 91.6,\ q_4 = 208.4$.

The optimal total cost C^* is given by;

$$\begin{aligned} C^* &= 1000 + 13.5(178.406) + 0.0345(178.406)^2 + 1245 + 13.1(192.576) + 0.033(192.576)^2 \\ &\quad + 2650 + 14.5(149.31) + 0.0345(149.31)^2 + 4.2(91.6) \\ &\quad + 0.03(91.6)^2 + 0.011(149.31)(91.6) \\ &\quad + 1200 + 4.2(208.4) + 0.02(208.4)^2 \end{aligned}$$

$$\Rightarrow \quad C^* = 18810 \quad (3.98)$$

Summary

The solution to the given problem is summarized as follows.

$$\begin{cases} \lambda_p = 25.81 \\ \lambda_q = 12.536 \\ p_1 = 178.406,\ p_2 = 192.576,\ p_3 = 149.31 \\ q_3 = 91.6,\ q_4 = 208.4 \\ C^* = 18810 \end{cases}$$

3.3.3 The Cogeneration Constraint Lines-Handling Mechanism

We have four cogeneration constraints lines so far shown in the figure.

The lines are defined by:

AB: $q = 0.056p + 109.08$ (3.99)

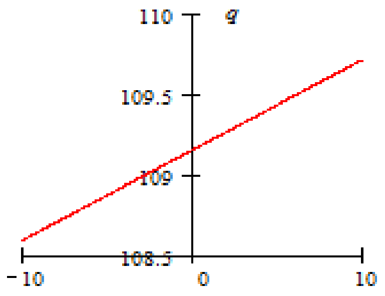
BC: $q = -23.9p + 4781$ (3.100)

CD: $q = 0.002p + 0.056$ (3.101)

AD: $q = 0.002p + 0.056$ (3.102)

The individual lines are depicted in fig. 3.2a, 3.2b, 3.2c and 3.2d below;

AB: $q = 0.056p + 109.08$ BC: $q = -23.9p + 4781$



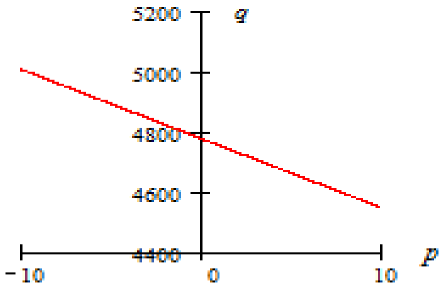
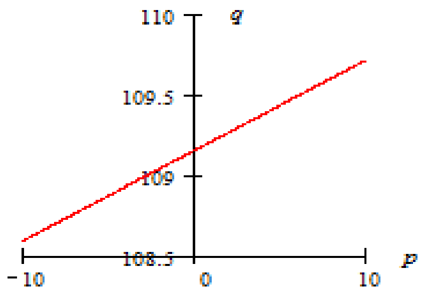


Fig. 3.2 (a)

Fig.3.2 (b)

CD: $q = 0.0022p + 0.06$ AD: $q = -21.98p + 439.7$



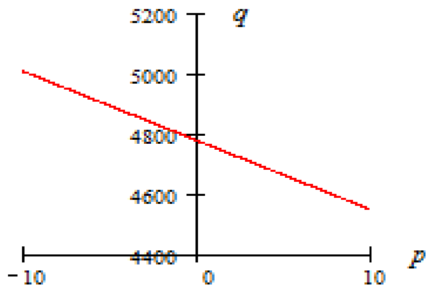


Fig.3. 2(c)

Fig.3. 2(d)

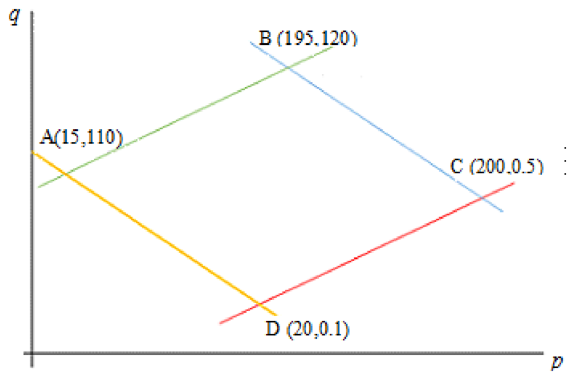


Fig. 3.3 Co-generation Unit 3 Feasible Operating Region

Fig.3.3 shows the heat-power Feasible Operation Region of a simple cycle cogeneration unit three, encountered in the problem statement. Here, the feasible operation region is enclosed by a line segment where each of the lines represents a constraint given by equation (3.14) of chapter 3. Feasible

Operation Region simply means that the cost must be minimized as well as constraints satisfied within the required limits.

We note that the feasible region is described by the inequalities expressed below:

$AB \leq 0$ above the line $q = 0.056p + 109.08$, i.e. $q = 0.056p + 109.08 \leq 0$

$BC \leq 0$ below the line $q = -23.9p + 4781$, i.e. $q = -23.9p + 4781 \leq 0$

$CD \geq 0$ above the line $q = 0.0022p + 0.06$ i.e. $q = 0.0022p + 0.06 \geq 0$

$AD \geq 0$ above the line $q = -21.98p + 439.7$ i.e. $q = -21.98p + 439.7$

We test our cogeneration units $p_3 = 149.31$ and $q_3 = 91.6$ in equations (28) to (31).

AB: 91.6 – 0.056 (149.31) – 109.08 = -25.841 < 0 (inside feasible region)

BC: 91.6 + 23.9 (149.31) – 4781 = - 1121 < 0 (inside feasible region)

CD: 91.6 – 0.0022 (149.31) – 0.05 = 91.212 > 0 (inside feasible region)

AD: 91.6 + 21.98 (149.31) – 439.7 =2934 > 0 (inside feasible region)

Hence the problem is completely solved.

But,

$$p_1 = \frac{\lambda_p - 13.5}{0.069}$$

$$p_2 = \frac{\lambda_p - 13.1}{0.066}$$

$$p_1 + p_2 = 325$$

$$\frac{\lambda_p - 13.5}{0.069} + \frac{\lambda_p - 13.1}{0.066} = 325$$

$$\Rightarrow 0.066\left(\lambda_p - 13.5\right) + 0.069\left(\lambda_p - 13.1\right) = 4.554 \times 10^{-3} \times 325$$

$$\therefore \quad 0.066\lambda_p - 0.891 + 0.069\lambda_p - 0.904 = 1.480$$

$$0.135\lambda_p = 1.480 + 1.795 = 3.275$$

$$\lambda_p = \frac{3.275}{0.135} = 24.26$$

Similarly,

$$\frac{\partial L}{\partial q_4} = 4.2 + 0.04q_4 - \lambda_q = 0 \quad \Rightarrow \quad \lambda_q = 4.2 + 0.04q_4 \cdot$$

The heat and power outputs of cogeneration unit three were infeasible. We investigated the position of these outputs and found that lines AB and BC were the violated constraints. Therefore, we select the new operating point of the cogeneration (unit3) to be on the corner i.e. [195,120]. The next stage of system lambda (Lagrangian multiplier) calculation of unit three is treated as a fixed output (both heat and power) unit. We achieved this by subtracting the value of the new operating point [195,120] from the system power, heat demand, and exclude unit three from the next lambda calculations. Thus we have:

$$p_1 + p_2 + p_3 = 520$$

$$p_1 + p_2 = 520 - p_3$$

$$p_1 + p_2 = 520 - 195 = 325\text{MW}$$

Similarly,

$$q_3 + q_4 = 300$$

$$q_4 = 300 - q_3$$

$$q_4 = 300 - 120 = 180\text{MWth}$$

Recall that,

$$\frac{\partial L}{\partial p_1} = 13.5 + 0.069 p_1 - \lambda_p = 0$$

$$\frac{\partial L}{\partial p_2} = 13.1 + 0.066 p_2 - \lambda_p = 0$$

$$4.2 + 0.04 q_4 = \lambda_q$$

$$\lambda_q = 4.2 + 0.04(180)$$

$$\lambda_q = 4.2 + 7.2$$

$$\lambda_q = 11.4$$

$$p_1 = \frac{\lambda_p - 13.5}{0.069} = \frac{24.26 - 13.5}{0.069} = 155.9 \text{ MW.}$$

$$p_2 = \frac{\lambda_p - 13.1}{0.066} = \frac{24.26 - 13.1}{0.066} = 169.1 \text{ MW.}$$

Lambda (Final):

$$\lambda_p = 24.26, \lambda_q = 11.4$$

$$p_1 = 155.9,\ p_2 = 169.1,\ p_3 = 195,\ q_3 = 120,\ q_4 = 180.$$

$$\begin{aligned}\text{Min } C &= \left(1000+13.5p_1+0.0345p_1^2\right)+\left(1245+13.1p_2+0.033p_2^2\right)\\ &\quad+\left(2650+14.5p_3+0.0345p_3^2+4.2q_3+0.03q_3^2+0.011p_3q_3\right)\\ &\qquad+\left(1200+4.2q_4+0.02q_4^2\right)\end{aligned}$$

$$C_{e_1}=1000+13.5(155.9)+0.0345(155.9)^2=1000+2104.65+838.52=3943.17$$

$$C_{e_2}=1245+13.1(169.1)+0.033(169.1)^2=1245+2215.21+943.63=4403.84$$

$$\begin{aligned}C_{eh}&=2650+14.5(195)+0.0345(195)^2+4.2(120)+0.03(120)^2+0.011(195)(120)\\ &=2650+2827.5+1311.86+504+432+257.4\\ &=7982.76\end{aligned}$$

$$C_h=1200+4.2(180)+0.02(180)^2=1200+756+648=2604$$

$$\begin{aligned}\text{Min } C &= C_{e_1}+C_{e_2}+C_{eh}+C_h\\ &=3943.17+4403.84+7982.76+2604=18933.77\end{aligned}$$

3.3.4 Numerical Result from Direct Solution (Lagrange Multiplier) Technique

The above mathematical algorithm illustrates the solution of combined heat and power economic dispatch problem by Lagrangian multiplier method or direct solution algorithm. The test system consists of two conventional power units, one cogeneration unit and a heat-only unit. The heat-power feasible operation region of the cogeneration unit is illustrated in figure 3.3 of chapter 3.As explained in chapter 3, combined heat and power economic dispatch has been formulated with the objective of minimizing fuel cost. Hence, data set for the combined heat and power dispatch problem were given in tables 3.1, 3.2, 3.3 and 3.4 along with feasible region coordinates of combined heat and power units respectively.

Table 3.5 shows result of output decision variables from the four-unit test system with power demand = 520MW and heat demand =300Mwth respectively. Combined heat and power economic dispatch decision variables were obtained by applying formula derived during modelling of the problem statement. According to table 3.5, this algorithm has objective function value (₦18933.8) and all the output decision variables (P1, P3, Q3 and Q4) were found to satisfy the given constraints of equations (3.5), (3.6) and (3.14) of chapter 3. Result shows that this technique can provide the combined heat and power dispatch solution in few steps when the load levels are such that all units can operate at the same incremental cost. However, one of the demerits of this technique is that it requires a few additional steps to identify all violating units if the load levels are such that some of the units are to be set at their limits, that is, when the unit has heat limit.

It becomes vital to mention that the intention of the above algorithm is to prove that combined heat and power economic dispatch problem can be solved directly in spite of the proposed (genetic) algorithm being automated and fast in determining the output decision variables. It attained optimal result even in large system like the one in this research compare with direct method. The Lagrangian multiplier algorithm framework is indeed a powerful paradigm and there is no reason why it should fail to provide solution for this class of optimization problems to a certain degree. Though, this technique appears compromising because of the lengthy recalculations required during overshoot. Such recalculation makes the problem difficult to converge to global minimum. Besides, its effectiveness for large units is not known, as it is convenient to obtain combined heat and power economic dispatch variables by proposed technique (Genetic Algorithm) because iterative search method gives optimal solution to this class of optimization problem as proved by the results in table 3.9.

Table 3.5. Results Obtained from Direct Solution Technique.

P_1(MW)	P_2(MW)	P_3(MW)	Q_3(MWth)	Q_4(MWth)	Cost(₦)
155.9	169.1	195.0	120.0	180.0	18933.8

3.3.5 System Lambdas (Lagrangian multipliers): Lagrange multipliers (λ) associated with the Karush-Kuhn- Tucker (KKT) necessary optimality conditions as given in equation (3.16) in section (3.2) was used to compute dispatch values of heat and power using direct solution algorithm. First, we deduced from this technique that the value of Lagrangian multiplier calculated during the initial steps is quite close to the final Lagrangian multiplier values. The final Lagrangian multiplier values gives the final dispatch heat and power values when combined heat and power economic dispatch problem was calculated directly as shown in table 3.6 below. This is because none of the computed output decision variables of the generators is outside feasible region. Neither did any of them violate their limits as specified by constraint equations (3.5), (3.6), and (3.14).If anything, the initial Lagrangian multiplier value [λp, λq] = [25.81 12.54] produced a total fuel cost (objective function) of ₦18810, but had the cogeneration unit-3output decision variables in infeasible region. Therefore, we investigated and found the new operating point of cogeneration unit-3 to be on the corner i.e. [195,120]. The next stage of system lambda calculation, unit-3 was treated as a fixed output (both heat and power) unit. This was achieved by subtracting the values of the new operating point [195,120] from the system power and heat demand and excluding unit3 from the next lambda calculation. The final Lagrangian multiplier values was calculated to be [λp λq] = [24.26 11.40] and the total fuel cost for this value was ₦ 18933.8. It is also easy to comprehend that Lagrangian multiplier algorithm outlined in section (3.2) is applicable even when there are more than one heat areas in the system. Every additional heat area requires the computation of one additional incremental cost equivalent to the heat demand of the area. This can be carried out by increasing the size of matrix *A* in (3.66) by one (for every additional heat area) and modifying (3.65) appropriately.

Table 3.6 Dispatch results of Power and heat.

Initial dispatch value	λ_p	λ_q
	25.81	12.54
Final dispatch Value	λ_p	λ_q
	24.26	11.40

3.3.6 Direct Handling Constraint Violation(S) of Combined Heat and Power Economic Dispatch Problem

It is crucial to use an effective constraint-handling mechanism to improve the quality of solutions in a combined heat and power economic dispatch problem. While either generating a new solution or modifying an existing solution, the constraint-handling mechanism must ensure the population lies within the bounds of feasible operation region of the combined heat and power units. The consideration of transmission losses makes this even more challenging. However, the constraint-handling mechanism proposed in this research did not look into transmission losses. Each line segment of the quadrilateral on the P-Q plane in figure 1 above represents a constraint of the cogeneration unit. The output value of the cogeneration unit could be at point O (feasible region), that is positive, on any of the line segment of the quadrilateral i.e. zero and negative i.e. outside (infeasible region) the polyhedron. Equations (3.65) and (3.66) together represent the main result of the research when solved directly. Given the cost functions of the units and the total power and heat demands, we can calculate λ_p and λ_q using (3.65) and (3.66). When we have the values of λ_p and λ_q , the outputs of the electrical units, heat units and co-generation units can be obtained by applying (3.45) and (3.46) of the research paper respectively. If none of the computed outputs of the generators are outside their limits specified in (3.5), (3.6), and (3.14), then they constitute the required dispatch. It can be deduced that the procedure outlined in the direct solution (Lagrangian) method is applicable even when there are more than one heat areas in the system.

Every additional heat area requires the computation of one additional ‘lambda’ corresponding to the heat demand of the area. This can be carried out conveniently by increasing the size of matrix *A* in (3.66) by one (for every additional heat area) and modifying (3.65) judiciously. However, the dispatch gained based on the lambdas computed using equation (3.66) may sometimes not be as feasible as the capacity constraints if (3.5), (3.6) and (3.14) have not been taken into account in deriving this relation. The basic idea in handling these constraints is to identify units that violate the constraints and set the violating quantities at their appropriate limit.

The system demand is modified to reflect the fact that their outputs are fixed and known. New values for the system lambdas are calculated considering only the units not at their limits using (3.66) of the research paper. The mechanism will continue until we reach a stage when no fresh limitation contraventions are experienced. The technique of adjusting the dispatch problem when the limits of power or heat units (1), (2) or (4) are breached is quite uncomplicated. The output, which is behind the designated range, is set at the maximum (that is violated) and the P^D or Q^D is lowered by this aggregate. The specific unit is shut out while calculating lambdas using equation (3.66) of the research paper in the next stage. Managing contraventions in a cogeneration unit also demands setting the output of the violating unit at one of the points on the boundary of its feasible region. The point on the boundary is so chosen that the Karush-Kuhn-Tucker (KKT) conditions for full problem optimality, as given in equation (3.16) of the research paper are gratified by the active constraints. The procedure to achieve this is as follows: For violating cogeneration units say k, identify the subset of limitations of equation (3.14) of the research paper that are contravened. Each violating constraint can be associated with a line segment forming a part of the boundary of the feasible region, that is, the Quadrilateral in figure.1.

The set of violating constraints correspond to a set of continuous boundary line segments. We refer to this part of the boundary as the violated boundary. In Figure.1 above, the closed region ABCD is the feasible region. Each line forming the boundary of this region corresponds to a constraint. If the computed output of the unit corresponds to points such as M or N, it corresponds to violation. At point M, only one constraint corresponding to line AB is violated. Hence, for this case, AB constitutes the violated boundary. If the computed output corresponds to point K, it implies the violation of the constraints corresponding to lines CD and BC. In this case, the line segments CD and BC constitute the violated boundary for a violating unit, the point representing the unit's output corresponding to the system lambdas falls outside the feasible region of the unit. It is shifted on to the violated boundary of the feasible region in one of the following way: if the unit's output is modified to correspond with this new operating point, search for points on the violated boundary where the unit's incremental cost $\lambda_{pk} = \partial c_k / \partial p_k$ or $\lambda_{qk} = \partial c_k / \partial q_k$ is equal to the corresponding component of the system lambda. This can be done easily since on any line segment in the $P – Q$ plane—the incremental cost values of the unit vary linearly and the incremental costs corresponding to any point λ_p—can be easily obtained by evaluating the gradient of the cost function at that point. If a point on the violated boundary is found where $\partial c_k / \partial p_k$ the system equals λ_p, then the power output of the unit is set at the value (say $\hat{p}_k$) corresponding to this point.

In the next stage while recalculating the system λ, the unit is treated as variable heat output unit at a fixed power output of $\hat{p}_k$. This is carried out thus: the total system power demand is decremented by a value $\hat{p}_k$; the known value of $\hat{p}_k$ is substituted into the cost function of the k^{th} cogeneration unit to get a modified cost function corresponding to that of a heat unit. In the modified cost function, the coefficient ε_k remains unchanged whereas the coefficient δ_k gets modified as $\left(\delta_k + p_k\zeta_k\right)$.Conversely, if a point on the violating boundary can be found where $\partial c_k / \partial q_k$ equals the system λ_q for the next iteration, the unit is treated as a variable power unit with its *q*-output kept at a fixed value q_k corresponding to the point on the violated boundary. The value q_k is subtracted from the system heat demand. The coefficients of the cost function of the power unit are obtained by substituting the value $\hat{q}_k$ for q_k in the cost function of the cogeneration unit. In both cases, the above points can be found on the violated boundary, the point which results in miniature change in the output of the unit from the infeasible value is chosen as the operating point for the unit. The unit is served as a variable power unit or a heat unit depending on whether we choose to fix q_k or p_k as constant, as described earlier in this research paper. Where identifying even one of these two points on the contravened borderline becomes infeasible, then the operating point of the unit is chosen as the corner point on the contravened borderline nearest to the evaluated infeasible operating point. This scenario is typical of this research. Here, the new operating point was at the corner [195,120] .Therefore, in the next stage of the system lambda computation, this unit is demonstrated as a fixed output to both heat and power unit. This is done by deducting $\hat{p}_k$ and $\hat{q}_k$ i.e. the output analogous to the chosen corner point on the contravened borderline from the system demand, and excluding this cogeneration unit from the successive lambda calculations.

5.3 Outline of Solution Scheme to Handle Limit Violations

a. Compute the system λ_p, λ_q using equations (3.65) and (3.66) of the research paper considering all the units that are not at their limits in the first iteration where all units are considered.

b. Compute the output of all the units corresponding to the computed values of system λ_p and λ_q. Check for limit violations. Set the output of violating units at appropriate limits. If there are any fresh limit violations, then the total output computed will be different from the original P^{demand}, Q^{demand}.

c. Determine the coefficients of the cost functions of violating co-generation units to be treated as power or heat units. Subtract from the given system P^{demand}, Q^{demand} the known outputs of units set at their limits [68].

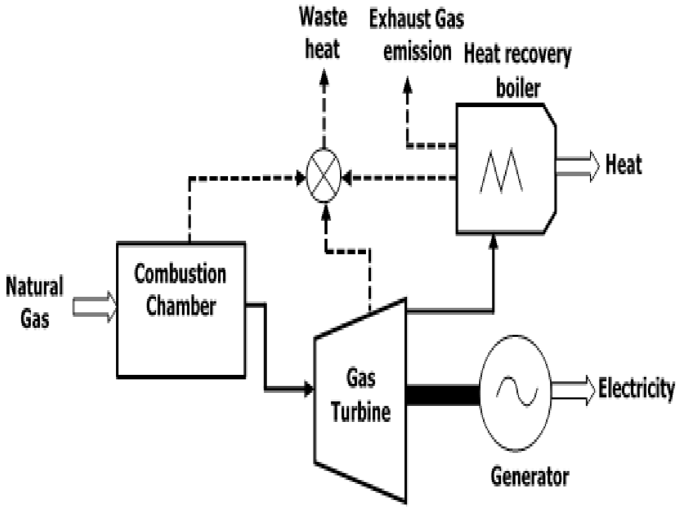


Fig. 3.4. Basic diagram of a typical combined heat and power system.

3.4 IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM BY GENETIC ALGORITHM METHOD.

Various techniques have recently been proposed for solving multimodal optimization problems. These techniques can be divided into two main categories: deterministic and stochastic (meta-heuristic) methods. Deterministic methods, for example, gradient descent method or quasi Newton method, when they solve complex multimodal optimization problems, may easily get trapped in some local optimum resulting from deficiency in exploiting local information. They depend mainly on a-priori information about an objective function capable of leading to fewer reliable results. Stochastic algorithms on the other hand, combine randomness as well as rules mimicking several phenomena. These phenomena include physical processes, for instance, the simulated annealing proposed by author [58], evolutionary processes (e.g. evolutionary algorithm) put forward by authors [59], [60], [50]. Genetic algorithms suggested by [44], and immunological systems e.g. Artificial immune systems put forward by author[62] electromagnetism-like put forward by author[63] and gravitational search algorithm put forward by author[64] all fall under this category.

Genetic algorithms got its idea from the Darwinian theory of biological evolution as an optimization technique. Its principal objective was obtained from natural evolution. Biological operators like crossover, mutation and selection play significant roles in Genetic Algorithms. Genetic algorithm has three randomly created phases: original population of chromosomes, crossover operator, and mutation operator. Each chromosome constitutes a unique solution to the problem with its quality being best determined by the value of fitness function. Genetic algorithm commences by creating some random solutions denoted as initial population. In the next phase, random crossovers give rise to new successors and in step three, random values of mutation from a few genes in the chromosome are adjusted or replaced. The new generation of solutions is then used in the next iteration of the algorithm. Traditional genetic algorithms usually work well for unique optimum problems but unsuccessful if they have to find multiple solutions. However, genetic algorithms coincide often with a local optimum after a certain number of generations. This is due to a low variety in the population or the incapacity of the mutation process to avoid local optima.

3.4.1 Algorithm1 Pseudo-code of the standard genetic algorithm

```
Input: $N$: Population size; $P_c$ : Crossover rate; $P_m$ : Mutation rate.
Output: Best Chromosome.
$t \leftarrow 0$
Initialize arbitrarily the initial population $P(t)$.
while (not termination condition) do
    Evaluate $P(t)$ using a fitness function
    Select $P(t)$ from $P(t-1)$
    Recombine $P(t)$
    Mutate $P(t)$
    Evaluate $P(t)$
    Replace $P(t-1)$ by $P(t)$
    $t \leftarrow t+1$
end
```

3.4.2 Five Components for the Sequential Execution of Genetic Algorithm

•A worthy genetic representation for individual (chromosomes).

•A technique to create the initial population.

•A fitness function to calculate the quality of each potential solution.

•Genetic operators that adjust the genetic configuration of parents to produce a new offspring.

•The choice of the values of the various genetic algorithm parameters (population size, cross over rate, mutation rate, stopping criteria... etc.)

3.4.2.1 Genetic operators initial population:

As genetic algorithm begins its search process for the optimal solution by acting on the initial population which is a set of potential candidates, the initialization method is a very important step since it alters the efficiency of the genetic algorithm. Hence, the choice of an efficient initial population method enhances the genetic algorithms search effectiveness. The initial population is usually created randomly in the standard genetic algorithm. However, the use of a random process causes invalid solutions which increase the algorithm’s convergence time. Thus, coupled with proposing new constructive methods that permit only valid solutions in the initial population, researchers have also used a combination of random and constructive methods to construct the initial population of genetic algorithm. Population size is a generally fixed parameter during genetic algorithm execution. But there are modified versions of genetic algorithm where the size is dynamic. The choice of the value of this parameter is an influential factor for determining the quality of genetic algorithm convergence. In this research, a random approach was used for creation of initial population with a fixed population size throughout the algorithm execution. This is because the generated solution satisfied the underlying constraints (power and heat) of the combined heat and power problem. It explored the operating bounds and then generated several feasible solutions capable of constituting the initial population of genetic algorithm.

3.4.2.2 Fitness function: Once the initial population is created, genetic algorithm must determine the performance of each individual by using an adaptive function which assigns to each possible solution, a fitness value that reflects its quality. Fitness function must consider several criteria, such as distance, safety, smoothness etc. The definition of a suitable fitness function is a crucial task since genetic algorithm uses the information generated by this function to choose the individuals for reproduction, mutation, and at the end of the process, it selects the best solution in the final population according to its fitness value.

3.4.2.3 Selection operator: Selection is a genetic operator used to choose parents likely to survive to produce the next generation. Parents with the best fitness values are more likely to be selected for mating. There are different selection methods that can be used: Elitism, Tournament, Roulette Wheel, Stochastic Universal Sampling, Linear Rank, Exponential Rank, and Truncation Selections. The main objective of the selection operator is promoting individuals with high adaptability to be selected for the next generation. The selection pressure is an important criterion which strongly influences the performance of genetic algorithm. Where selection pressure is high, genetic algorithm converges quickly without exploring every available search space. On the other hand, a low selection pressure produces random solutions. In our approach, Elitist and Truncation Selection methods are used to control the pressure selection. Elitist which has high pressure selection is used to keep the fittest solutions throughout generations, and Truncation Selection is used to create an avenue for weak chromosomes to be selected from the last generation for reproduction in the current one, and to avoid the dominance of the best individual [12].

3.4.2.4 Crossover operator: After selecting individuals using the selection operator, the crossover is applied. Crossover is a genetic operator that blends the genetic information (genes) of two selected chromosomes (parents) to yield new chromosomes (offspring/child) for population heterogeneity, and to boost the fitness value of the candidate solutions. The main idea behind crossover is that new chromosomes inherit the best characteristic of their parents. Thus, the result is having a better child that performs better than its parents. The crossover rate is the probability of performing crossover. Different crossover operators have been introduced: the Partially-Mapped, the Order Crossover, the Cycle Crossover, and Same Point crossover. Same point crossover seems to be the most used mechanism. Again, author [16], in his work has used the standard crossover mechanism, same point crossover which holds two crossover strategies: the one-point and the two-point crossovers are applied if there are at least two identical genes between the parents. Same point crossover was applied because it provided a better solution than the rest.

3.4.2.5Mutation operator: Mutation is a genetic operator applied to improve diversity and prevent premature convergence of algorithms. Generally, this operator randomly selects a position (gene) and replaces it with a new, non-existing gene on the path. Yet, as mentioned in [17], random mutations could generate invalid paths. Even if a solution is valid before the application of the mutation operator, the new gene altered can contain an obstacle and as well create an inappropriate path. In this study, we adopt random mutation. Mutation is performed by randomly choosing a cell from an individual and trying to replace same with one of its neighboring cells on the grid map.

3.4.3 CONSTRAINED NONLINEAR OPTIMIZATION PROBLEM CONVERTED TO GENETIC ALGORITHM FORMAT (Modelling of combined heat and power economic dispatch problem using genetic algorithm)

AMATLAB sub-routine was created to determine the combined heat and power economic dispatch decision variables (output Heat and output power) plus objective function value of four units' test system. The program, including sub-routines, was created with the inputs (tables 3.1, 3.2, 3.3 and 3.4 respectively) being the data and the output being the decision variables (power and heat from respective units) plus objective function value.

The optimization problem with the specified constraint conditions is given as:

$$\begin{aligned}\text{Min }\; C &= \sum_{i=1}^{2} c_{e,i}\left(p_i\right) + \sum_{i=1}^{1} c_{c,i}\left(p_i, q_i\right) + c_{h,1}\left(q_4\right) = c_{e,1}\left(p_1\right) + c_{e,2}\left(p_2\right) + c_{c,1}\left(p_3, q_3\right) + c_{h,1}\left(q_4\right) \\ &= \left(1000 + 13.5p_1 + 0.0345p_1^2\right) + \left(1245 + 13.1p_2 + 0.033p_2^2\right) \\ &\quad + \left(2650 + 14.5p_3 + 0.0345p_3^2 + 4.2q_3 + 0.03q_3^2 + 0.011p_3q_3\right) \\ &\quad + \left(1200 + 4.2q_4 + 0.02q_4^2\right)\end{aligned} \tag{3.103}$$

$$p_1 + p_2 + p_3 = 520 \tag{3.104}$$

$$q_3 + q_4 = 300 \tag{3.105}$$

$$p_1^{\min} \le p_1 \le p_1^{\max} \quad \Rightarrow \quad 10 \le p_1 \le 250 \tag{3.106}$$

$$p_2^{\min} \le p_2 \le p_2^{\max} \quad \Rightarrow \quad 20 \le p_2 \le 200 \tag{3.107}$$

$$q_4^{\min} \le q_4 \le q_4^{\max} \quad \Rightarrow \quad 20 \le q_4 \le 250 \tag{3.108}$$

For the purpose of implementing the Genetic Algorithm solution to the above problem in MATLAB 7.0 on an H.P Pavilion Laptop configured in1:80*GHZ*, Intel *i*7 processor, 16*GB* RAM with WINDOWS 10 operating system, we make the following change of variables:

$$\begin{aligned}p_1 &:= x_1 \\ p_2 &:= x_2 \\ p_3 &:= x_3 \\ q_3 &:= x_4 \\ q_4 &:= x_5\end{aligned}$$

Hence we enter the following in MATLAB

$$\begin{aligned}\text{Min }\; \mathrm{z} &= \left(1000 + 13.5x_1 + 0.0345x_1^2\right) + \left(1245 + 13.1x_2 + 0.033x_2^2\right) \\ &\quad + \left(2650 + 14.5x_3 + 0.0345x_3^2 + 4.2x_4 + 0.03x_4^2 + 0.011x_3x_4\right) \\ &\quad + \left(1200 + 4.2x_5 + 0.02x_5^2\right)\end{aligned}$$

Subject to the linear constraint:

$$x_1 + x_2 + x_3 = 520$$

$$x_4 + x_5 = 300$$

Or in matrix form we can write

$$\begin{cases} x_1 + x_2 + x_3 + 0\cdot x_4 + 0\cdot x_5 = 520 \\ 0\cdot x_1 + 0\cdot x_2 + 0\cdot x_3 + x_4 + x_5 = 300 \end{cases} \quad \Rightarrow \quad \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 520 \\ 300 \end{bmatrix}$$

In MATLAB the above linear constraint is written in the form $Ax = b$, i.e.

$$\begin{bmatrix}1 & 1 & 1 & 0 & 0;0 & 0 & 0 & 1 & 1\end{bmatrix}\boldsymbol{x}=\begin{bmatrix}520;300\end{bmatrix}$$

where, $A=\begin{bmatrix}1 & 1 & 1 & 0 & 0;0 & 0 & 0 & 1 & 1\end{bmatrix}$, $b=\begin{bmatrix}520;300\end{bmatrix}$.

Similarly the given lower and upper bounds for the *p*'s can be converted to the *x*'s as follows.

$$p_1^{\min} \le p_1 \le p_1^{\max} \ \Rightarrow\ 10 \le p_1 \le 250 \ \Rightarrow\ 10 \le x_1 \le 250$$

$$p_2^{\min} \le p_2 \le p_2^{\max} \ \Rightarrow\ 20 \le p_2 \le 200 \ \Rightarrow\ 20 \le x_2 \le 200$$

$$q_4^{\min} \le q_4 \le q_4^{\max} \ \Rightarrow\ 20 \le q_4 \le 250 \ \Rightarrow\ 20 \le x_5 \le 250$$

For the purpose of convergence of the GA in MATLAB the other variables such as x_3 and x_4 are also required to have a lower and upper bound, otherwise putting their values equal to zero will not give a solution. We thus peg their respective lower and upper bounds to be $10 \le x_3 \le 100$ and $10 \le x_4 \le 100$. Hence the complete set is;

$10 \le x_1 \le 250$

$20 \le x_2 \le 200$

$10 \le x_3 \le 100$

$10 \le x_4 \le 100$

$20 \le x_5 \le 250$

In MATLAB we input these as;
Lower bound vector: [10 20 10 10 20]
Upper bound vector: [250 200 100 100 250]

Once again, the sub-routines created in MATLAB showed the respective tables (3.1, 3.2, 3.3 and 3.4) as input variables to the program with the output decision variable remaining the power, heat from respective units plus objective function value.

3.4.3.1 Analysis and discussion of Genetic Algorithm Results: Proposed algorithm (genetic algorithm) has been applied for 4 generating units of combined heat and power economic dispatch problems. Cost function parameters along with feasible region coordinate of combined heat and power units are taken from tables (3.1), (3.2), (3.3) and (3.4) respectively. The system consists of two conventional power units one cogeneration unit; and a heat-only unit. The heat-power feasible operation region of the cogeneration unit is illustrated in figure 3.3. As explained in chapter 3, combined heat and power economic dispatch has been formulated with the objective of minimizing fuel cost. Table 3.7 shows power generation and heat generation output results of four-unit test system with power demand (P^d)= 520MW and heat demand H^D=300Mwth. The result in Table 3.7shows that when the above heat and power loads were used, only unit two reached its capacity limit with proposed algorithm being employed to solve the combined heat and power economic dispatch problem. It also follows that the minimum objective function value was obtained after 51 iterations, with cost function value (₦21120.0).

In this research, obtained output decision variables (P_1, P_3, Q_3 and Q_4) were satisfactory for all the available constraints (equality and inequality constraints) associated with the combined heat and power economic dispatch problem, while output decision variable of unit 2 remained infeasible. The global optimal solution obtained in this 4-unit test system confirms the applicability of the proposed (genetic algorithm) for dealing with optimization problems of this class. The result improvements compared to existing techniques are significant as demonstrated with particle swarm optimization and artificial bee colony algorithm. It is therefore observed that due to its effective searching capability, proposed genetic algorithm can converge to produce intensely diverse and widespread solutions along with better, extreme solutions. We therefore conclude that proposed algorithm does not only provide a reasonable assessment of global solutions, but better convergence speed. Genetic algorithm therefore, being a probabilistic search technique, is known to be computationally more efficient for problems that permit probability solutions similar to the one proposed in this research.

Genetic Algorithm-Output

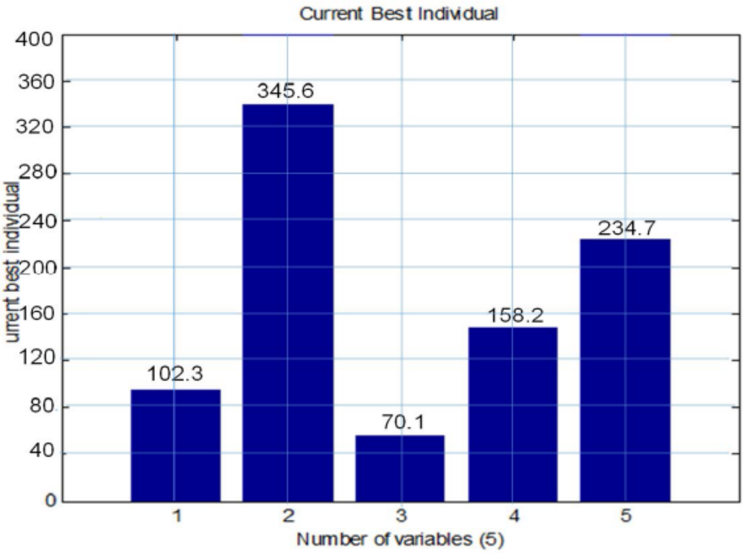


Fig 3.5: GA-Output with respective values of the independent variables.
$p_1 = 102.3, p_2 = 345.6, p_3 = 70.1, q_3 = 158.2, q_4 = 234.7$

The above figure 3.5 depicts the combined heat and power output decision variables computed using genetic algorithm in the form of bar chart. Each bar represents an output decision variable (power or heat). Power and Heat Outputs decision variables of units1, 3 and 4 operate within the required (feasible) bound, whereas unit 2output decision variables are infeasible. The result implies that simulated output decision variables—heat and power— at respective 520MW and 300MW loads have global minima on units 1, 3 and 4. They satisfy all available constraints, unlike unit-2 that could not find the optima in the specified maximum number of cycles. The proposed algorithm produced results quite close to the global optima with minimum objective function value.

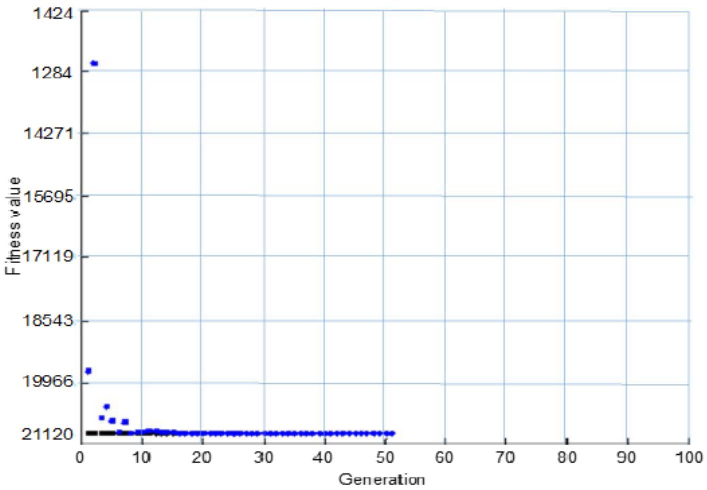


Fig3.6: After 51 iterations we get the minimum value of the fitness function; *z*= 21120.

Figure 3.6 indicates fitness function values against a number of generation simulation results obtained from combined heat and power economic dispatch problems, using data sets from tables (3.1), (3.2), (3.3) and (3.4) at power and heat loads of 520MW and 300Mwth. From the above graph it could be seen that the objective function value converged at 51 iterations. Convergence of genetic algorithm is generally difficult to obtain due to the fact that evolutionary computations incorporate complex nonlinear stochastic processes. In a negative case where more random individuals are generated with no enhanced optimized minimum, no improved state (better optimized minimum) is achieved. Nevertheless, we can state with a determined probability that any minimum with lower value than the optimized minimum does not exist. This probability can be made arbitrarily close to 1 if we generate sufficiently large number of random individuals. The result in figure 3.6 shows that proposed algorithm converges quite close to the global minimum of cost function. This is because random individuals explore the search space of the optimization problem and not only the neighborhood of the optimized minimum in each generation. Also, the selection process chooses the best individual from parents and offspring (elitism); and every state is reachable from any other state.

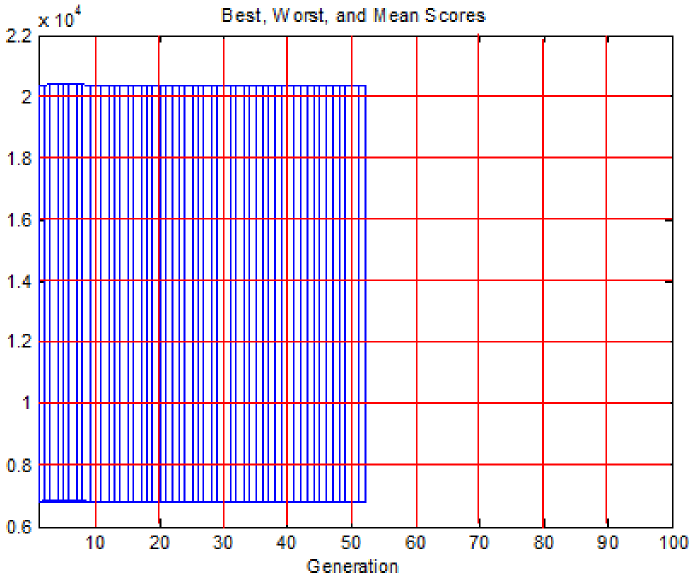


Fig 3.7: Best, Worst and Mean Scores (Number of iterations =51)

The result of the experiment for the genetic algorithm is given in table 3.7. Comparative results of the best mean and worst solutions of the proposed algorithm using the 4-unit test system are presented based on table 3.7.The best solution obtained by genetic algorithm for 4 test function, as seen in table 3.7, genetic algorithm has found the global minimum (feasible solution) of the three of four units (units 1, 3 and 4) through 51 iterations. On unit 2, genetic algorithm could not find the optima in the above number of iteration lines. Table 3.9 indicates that genetic algorithm is better than particle swarm optimization on two units (1 and 2) while proposed algorithm shows better performance over artificial bee colony on units (1, 2 and 4).With respect to the mean solution result, from the graph using the 4-unit test system over 51 iteration runs, proposed algorithm had optimal solutions from units (1, 3 and 4) but could not find optima in the specified number of iteration lines. The worst solution obtained by proposed algorithm was on unit 2. Here genetic algorithm could not find result quite close global optima after 51 iteration lines.

Table 3.7. Results Obtained from Genetic algorithm (GA)

P_1(MW)	P_2(MW)	P_3(MW)	Q_3(MWth)	Q_4(MWth)	Cost(₦)
102.3	345.6	70.1	158.2	234.7	21120.0

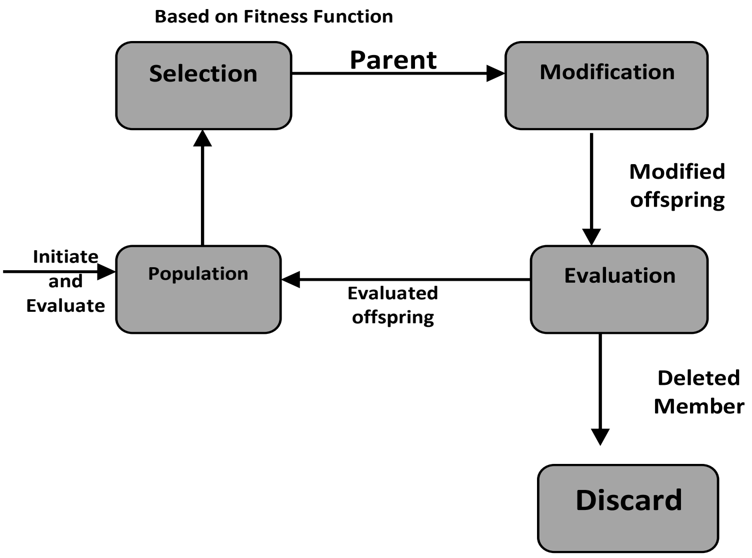


Fig 3.8: Genetic algorithm Evolutionary Cycle

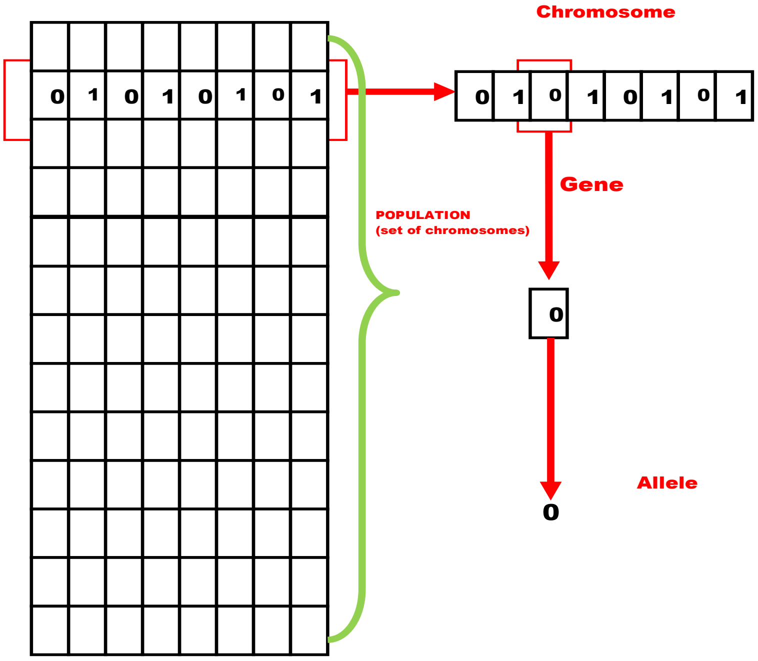


Fig3. 9: Genetic algorithm Population (set of chromosomes)

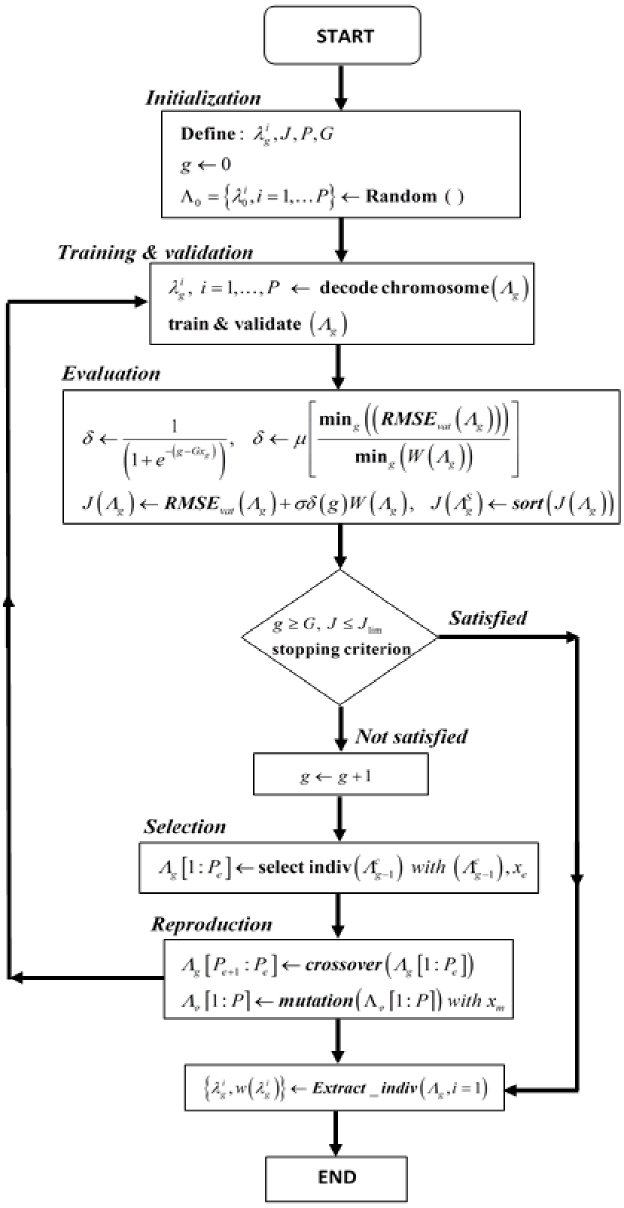

START
Initialization
Define: λ_g^i, J, P, G
$g \leftarrow 0$
$\Lambda_0 = \{\lambda_0^i, i = 1, \dots P\} \leftarrow$ Random ( )
Training & validation
$\lambda_g^i,\ i = 1, \dots, P \leftarrow$ decode chromosome(Λ_g)
train & validate (Λ_g)
Evaluation
$\delta \leftarrow \frac{1}{\left(1+e^{-(g-Gx_g)}\right)}, \quad \delta \leftarrow \mu\left[\frac{\min_g\left(\left(RMSE_{vat}(\Lambda_g)\right)\right)}{\min_g\left(W(\Lambda_g)\right)}\right]$
$J(\Lambda_g) \leftarrow RMSE_{vat}(\Lambda_g) + \sigma\delta(g)W(\Lambda_g), \quad J(\Lambda_g^S) \leftarrow sort(J(\Lambda_g))$
$g \geq G,\ J \leq J_{\lim}$
stopping criterion
Satisfied
Not satisfied
$g \leftarrow g+1$
Selection
$\Lambda_g[1:P_e] \leftarrow$ select indiv(Λ_{g-1}^S) with $(\Lambda_{g-1}^S), x_e$
Reproduction
$\Lambda_g[P_{e+1}:P_e] \leftarrow crossover(\Lambda_g[1:P_e])$
$\Lambda_e[1:P] \leftarrow mutation(\Lambda_e[1:P])$ with x_m
$\{\lambda_g^i, w(\lambda_g^i)\} \leftarrow Extract\_indiv(\Lambda_g, i=1)$
END

Fig 3.10: Flow Chart of the genetic algorithm Based Combined Heat and Power Optimization Problem.

Genetic algorithms, though theoretically capable of providing global optimum from any random set of initial population of candidate solutions, routinely fail with automatic all-constraints-satisfying solutions. Optimality proof of genetic algorithms is based on infinite stages of search completely infeasible in any context. This problem can be resolved by applying a partial solution that ensures all the candidate solutions considered at all stages of search are feasible solutions (i.e. satisfying all the available constraints). In this research, we wrote a small code to generate the members of the first generation such that the real power and heat outputs of each generator satisfy their corresponding equality and inequality constraints. The same method was used throughout the search while generating new candidate solutions through genetic operations to ensure that only those candidate solutions that satisfy their loads' equality and inequality constraints are in the pool. Finally, the search was limited to feasible solutions which satisfy equality and inequality constraints that give the minimum cost function.

3.5 IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM BY PARTICLE SWARM OPTIMIZATION ALGORITHM.

The Particle Swarm Optimization Algorithm was advanced by J. Kennedy and R. Eberhart in 1995.This Algorithm was originally used for solving continuous non-linear functions. The concept of particle swarm optimization comes from a simplified social system like birds flocking or fish schooling. Assume a group of birds is searching for food in an n-dimension area (*n* equals the number of control variables). None of these birds knows where the food is. However, they know which bird is nearest to the food (assume the closest bird to the food is Bird A). What happens is that, the rest of the birds will obtain food by following bird A and searching its adjoining region. A single particle in a PSO can be viewed as a bird. The position of each particle can be expressed as $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \mathrm{K}\ , x_{in})$. The original particle in particle swarm optimization is randomly selected and then particle swarm optimization will continually search for optimal value by updating the particles in each iteration. The fitness value of the particle is related to the objective function. And the velocity of the particles $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \mathrm{K}\ , v_{in})$ is related to its pervious velocity, global best known position, and local best known position. Velocity indicates the directions of all the particles in the next iteration. The local best known position is the best solution achieved by each particle so far. The global best known position is the best solution among all the achieved solutions. The inertia velocity part, local best known position part, and global best known position part of the velocity reflect the cooperation and competition mechanism in particle swarm optimization. Similar to genetic algorithm, particle swarm optimization also starts with a group of randomly generated solutions and updates the solutions in each iteration. However, particle swarm optimization uses historical data rather than does crossover and mutation operations. The behavior of all the particles appears to be managed by a control center. However, in reality, as formulas 1 and 2 describe below, the principle of the particle swarm optimization algorithm is quite straightforward.

3.5.1 A Particle Swarm Optimization Algorithm

The basic particle swarm optimization algorithm is developed by exploiting social model simulations. The method is developed with inspiration from flocking of birds and schooling of fish. The particle swarm optimization method was first designed to simulate behavior of birds searching for food in a bounded area. A single bird would find food through social cooperation with other birds— its neighbors— in the flock. Later, the method was extended for multi-dimensional search, and neighborhood topologies are considered to determine the relationship between particles in a swarm. The particle swarm optimization algorithm with dynamic neighborhood topology for every particle $1, 2, \ldots, N$ can be described as;

$$v_{t+1}^{i} = \Theta\left[v_{t}^{i} + \varphi_{1t}^{i}\left(p_{t}^{i} - x_{t}^{i}\right) + \varphi_{2t}^{i}\left(g_{t}^{i} - x_{t}^{i}\right)\right] \tag{3.109}$$

$$x_{t+1}^{i} = x_{t}^{i} + v_{t+1}^{i} \tag{3.110}$$

where $x_{t}^{i} \in \mathbb{R}^{n}$ is the position of i^{th} particle at time *t*, $p_{t}^{i} \in \mathbb{R}^{n}$ is the best position achieved by the i^{th} particle until time *t*, $g_{t}^{i} \in \mathbb{R}^{n}$ is the best position achieved by i^{th} particle and its neighbors until time *t*, $v_{t}^{i} \in \mathbb{R}^{n}$ is the rate of position change (velocity) of the i^{th} particle at time *t*, and *N* is the number of particles in the swarm. The coefficients $\varphi_{1t}^{i} \in \left[0, \bar{\varphi}_{1}\right]^{n}$ and $\varphi_{2t}^{i} \in \left[0, \bar{\varphi}_{1}\right]^{n}$ are *n*-dimensional uniform vectors with random distribution referred to as social and cognitive learning coefficients, respectively. They determine the relative significance of social and cognitive components. Equation in (3.109) shows how particles update their velocities dynamically during search. Equation (3.110) shows how particles adjust their positions according to their updated velocities. Equation in (3.109) has three components. The first component is the momentum component, which shows an adjustment to updated velocity to prevent a rapid change while updating same. The second component is the cognitive component, where particles have memory and are able to use their previous experiences while determining their velocity in search space. The last component, referred also to as the social component, shows social cooperation of particles in swarm ability, that is, particles' abilities to exploit their neighbors' experiences while determining their velocities in search space. The sum of the three components designated in equation (3.109) could result in large velocity values. In such cases the algorithm is said to be in *explosion* operation, where high values of the updated velocity prevent the particles from coinciding; and they disperse through the search space. $V_{\max}$ is the most remarkable variable in the basic particle swarm optimization algorithm affecting its performance, and it is the only variable that needs to be modified in order to use the basic particle swarm optimization algorithm.

A large value of $V_{\max}$ causes the particles to search in a larger area and to move far from areas with good solutions, while a small value causes the particles to search within a smaller area and to possibly get trapped in local minima. In order to prevent such cases, each particle's velocity could be limited to a range $\left[-V_{\max}, V_{\max}\right]$. Particle swarm optimization algorithms have a simple structure, are easy to implement and have a high computational efficiency. In the basic particle swarm optimization algorithm, each particle in *n*-dimensional search space is assigned randomly generated position and velocity vectors. A fitness value according to the chosen fitness function is assigned to each particle according to their initial positions in the search space. During search, each particle's fitness value is compared with the best fitness value achieved until that instant (pbest). The better value is then assigned as the best fitness value achieved until that instant, and its position is recorded as p_t^i. If all the particles are connected, it is global best, otherwise it is neighborhood best. A better value is assigned as the global best fitness value and the corresponding position is g_t^i. After determining the best and neighborhood global best position vectors using equation (3.109), each particle updates its position and velocity vectors. This situation continues iteratively until it reaches a predefined stopping criterion, that determines the desired performance aspects of the algorithm. The particle swarm optimization algorithm, like the genetic algorithms, simulated annealing; hill climbing is randomly initialized, where the members of the population interact with one another. Also, particle swarm optimization can converge to the possible solutions faster than other algorithms. But an incorrect fine-tuning of the algorithm parameters could result in a slower convergence [65].

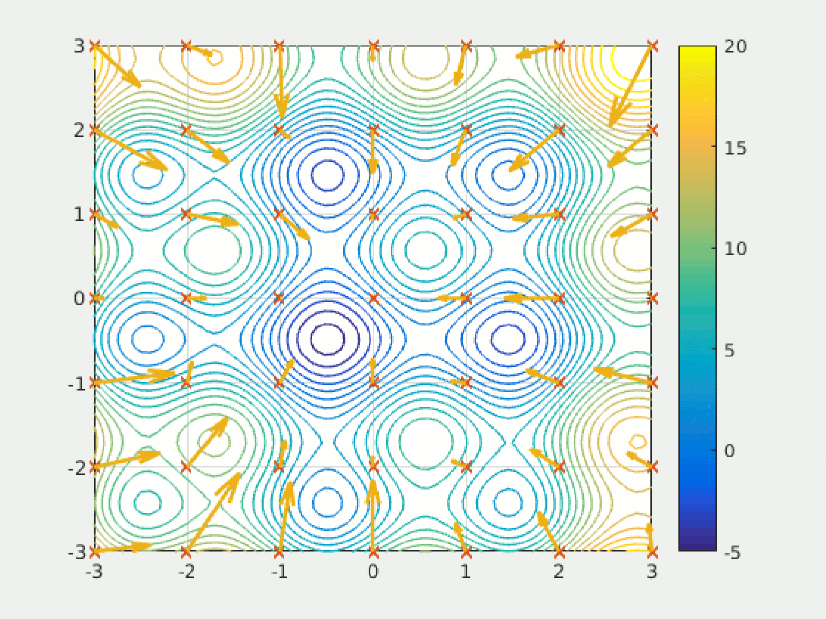
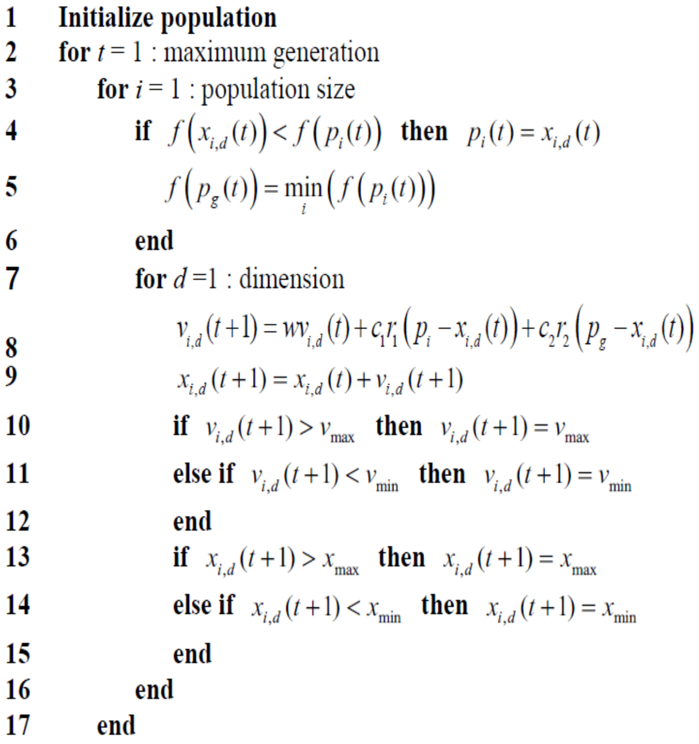


Fig 3.11: Particle swarm optimum search

3.5.2 Algorithm 2 Pseudo-code of the standard particle swarm optimization Algorithm.

The algorithm is thus presented below:



```
Initialize population
for $t = 1$ : maximum generation
    for $i = 1$ : population size
        if $f\left(x_{i,d}(t)\right) < f\left(p_i(t)\right)$ then $p_i(t) = x_{i,d}(t)$
            $f\left(p_g(t)\right) = \min_i \left(f\left(p_i(t)\right)\right)$
        end
        for $d = 1$ : dimension
            $v_{i,d}(t+1) = wv_{i,d}(t) + c_1 r_1 \left(p_i - x_{i,d}(t)\right) + c_2 r_2 \left(p_g - x_{i,d}(t)\right)$
            $x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$
            if $v_{i,d}(t+1) > v_{\max}$ then $v_{i,d}(t+1) = v_{\max}$
            else if $v_{i,d}(t+1) < v_{\min}$ then $v_{i,d}(t+1) = v_{\min}$
            end
            if $x_{i,d}(t+1) > x_{\max}$ then $x_{i,d}(t+1) = x_{\max}$
            else if $x_{i,d}(t+1) < x_{\min}$ then $x_{i,d}(t+1) = x_{\min}$
            end
        end
    end
```

3.6 MODELLING OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM USING PARTICLE SWARM OPTIMIZATION ALGORITHM

A MATLAB subroutine was created, using the algorithm process of 3.5.2, to calculate the combined heat and power economic dispatch output decision variables (heat and power) plus cost function; and displayed in Appendix B. The modelling equation used to implement combined heat and power economic dispatch problem using particle swarm optimization algorithm is illustrated below:

$$\begin{aligned}\text{Min } C &= \sum_{i=1}^{2} c_{e,i}\left(p_i\right) + \sum_{j=1}^{1} c_{c,j}\left(p_j, q_j\right) + \sum_{k=1}^{1} c_{h,k}\left(q_k\right) = c_{e,1}\left(p_1\right) + c_{e,2}\left(p_2\right) + c_{c,1}\left(p_1, q_1\right) + c_{h,1}\left(q_1\right) \\ &= \left(1000 + 13.5p_1 + 0.0345p_1^2\right) + \left(1245 + 13.1p_2 + 0.033p_2^2\right) \\ &\quad + \left(2650 + 14.5p_3 + 0.0345p_3^2 + 4.2q_3 + 0.0q_3^2 + 0.011p_3q_3\right) \\ &\quad + \left(1200 + 4.2q_4 + 0.02q_4^2\right)\end{aligned} \tag{3.111}$$

Subject to

$$p_1 + p_2 + p_3 = 520 \tag{3.112}$$

$$q_3 + q_4 = 300 \tag{3.113}$$

$$p_1^{\min} \le p_1 \le p_1^{\max} \quad \Rightarrow \quad 10 \le p_1 \le 250 \tag{3.114}$$

$$p_2^{\min} \le p_2 \le p_2^{\max} \quad \Rightarrow \quad 20 \le p_2 \le 200 \tag{3.115}$$

$$q_4^{\min} \le q_4 \le q_4^{\max} \qquad \Rightarrow \qquad 20 \le q_4 \le 250 \tag{3.116}$$

Once again, the sub-routines created in MATLAB showed the data sets from tables (3.1), (3.2), (3.3) and (3.4) as the inputs to the program and the output decision variables being the power and heat from respective units plus objective function values.

3.6.1 Analysis and Discussion of Particle Swarm Optimization Results

In the case of **c**ombined heat and power economic dispatch for 4 generating units (test system), particle swarm optimization algorithm has been applied. Cost coefficient parameters along with feasible region coordinates of combined heat and power unit is taken from tables (3.1), (3.2), (3.3) and (3.4) respectively. The test system comprises two conventional power units, one cogeneration unit and a heat-only unit. The heat-power feasible operation region of the cogeneration unit is illustrated in figure 3.3.

Combined heat and power economic dispatch has been formulated in section 3.2 of this chapter with the objective of minimizing fuel cost. Table 3.8 shows power and heat generation output decision variables of four-unit test system with power demand (P^d) = 520MW and heat demandH^D=300Mwth. It is vital to note that from power and heat loads specified above, particle swarm optimization algorithm has found the global minimum on units (1), (3) and (4). These units satisfy all the available constraints while the algorithm could not find optima on unit (2) in the specified maximum number of iterations. It must be mentioned that cost function value of (₦22917.0) was obtained after 103 iterations which indicated a slow convergence speed. The slow convergence speed of this algorithm makes computation cumbersome since extra arithmetic is required to correct this problem, unlike proposed algorithm. This class of optimization problem requires algorithm to make a perfect initial guess for optimization variables. The initial guess or starting point is vital, and can significantly affect the objective value. Also, the particles of particle swarm optimization do not utilize genetic operators, and their information-sharing mechanism is slow compared to proposed algorithm. For complicated problems, particle swarm optimization algorithm tends to get trapped in the local optimum for its high dimensional space. This can be avoided by continuously updating the parameters of particle swarm optimization algorithm. The value of cost function was higher than the expected threshold, which implies results are not globally feasible in all the units. Also, the result indicates the algorithm is incapable of attaining solutions that are both optimal and feasible.

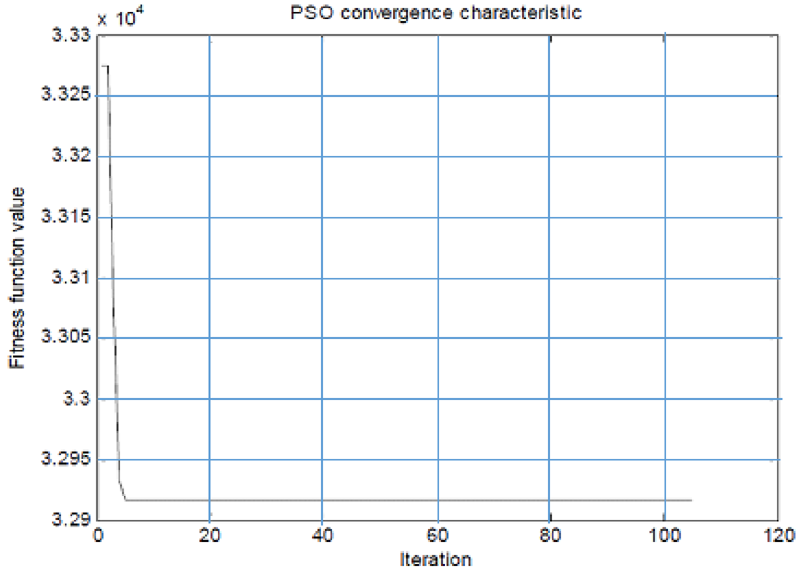


Fig 3.12: Particle swarm optimization convergence characteristic

3.6.2 Particle swarm Optimization Convergence Characteristics.

Figure 3.12 depicts particle swarm convergence characteristics of the combined heat and power economic dispatch problem of the research with power and heat loads of 520MW and 300Mwth respectively. The graph indicates fitness function values on the vertical axis and the number of iterations on the horizontal axis. Particle swarm optimization is a population-based stochastic optimization originating from artificial life and evolutionary computation. Particle swarm optimization is influenced by the social behavior of organisms, such as birds flocking, fish schooling and human social relations. Its characteristics of low constraint on the continuity of objective function and ability of adapting to the dynamic environment make particle swarm optimization one of the most important swarm intelligence algorithms.

From the convergence characteristics of particle swarm optimization above, we can easily draw the following important conclusions: Fitness function value of (₦22917.0) was obtained after 103 iteration lines which implies a slow convergence rate and low accuracy. To ensure better convergence characteristics of particle swarm optimization, the inertia weight should be selected around the center region of the convergence interval and this will eliminate the swinging process resulting in one step backward and two steps forward. Also the pliability (flexibility) of control parameters needs to be improved to strengthen the position updating randomness of particles for improved exploration ability of the algorithm and assistance with skipping local optimum. For its convenience of realization and low constraints on the environment and objective functions, particle swarm optimization has been accepted widely as a potential global optimizing algorithm. It does however not rule out the existing need for further research of the algorithm itself. Hence, of a deeper concern to particle swarm optimization researchers should be the process of studying and analyzing particle swarm optimization convergence characteristics to reflect its working mechanism. Finally, it is observed that particle swarm algorithm converges slowly in early iterations and hence, the number of maximum runs (iterations) can be increased to save the solution time.

Table 3.8 .Results Obtained from Particle Swarm Optimization (PSO)

P_1(MW)	P_2(MW)	P_3(MW)	Q_3(MWth)	Q_4(MWth)	Cost(₦)
110.3	390.6	69.1	160.0	230.7	22917.0

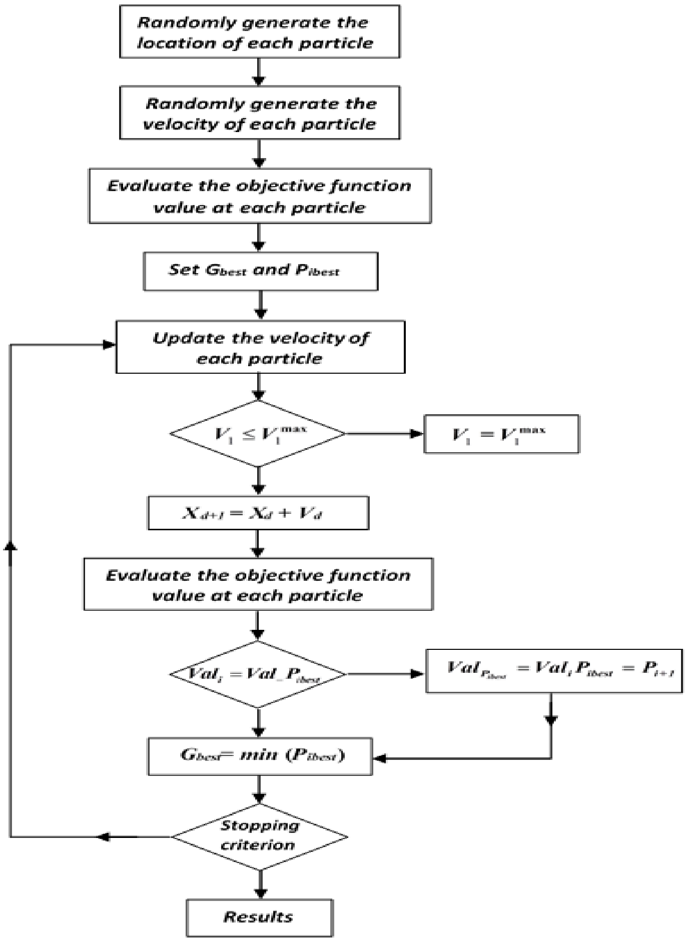

Randomly generate the location of each particle
Randomly generate the velocity of each particle
Evaluate the objective function value at each particle
Set G_{best} and P_{ibest}
Update the velocity of each particle
$V_1 \leq V_1^{max}$
$V_1 = V_1^{max}$
$X_{d+1} = X_d + V_d$
Evaluate the objective function value at each particle
$Val_i = Val\_P_{ibest}$
$Val_{P_{ibest}} = Val_i P_{ibest} = P_{i+1}$
G_{best}= min (P_{ibest})
Stopping criterion
Results

Fig3.13: Flow Chart of the particle swarm optimization based Combined Heat and Power Optimization Problem.

3.7 Conclusion.

Modelling and simulation of combined heat and power economic dispatch system comprising two conventional power units, one cogeneration unit and a heat-only unit was presented. Its performances under various optimization techniques (Genetic, artificial bee colony particle swarm optimization and direct solution algorithms were used in this chapter. The studies revealed that proposed(genetic) algorithm can realize better solutions in terms of the objective function value, convergence speed and the number solution that are near global minimum compared with Particle swarm optimization. The slow convergence speed associated with particle swarm optimization makes computation cumbersome because extra arithmetic is required to correct this problem at each generation. Furthermore, particle swarm optimization easily falls into local optimum in high dimensional space as seen in this class of optimization problem. This undeniable drawback associated with particle swarm optimization, makes it comparatively less practical and less attractive than proposed algorithm. On the other hand, direct solution (Lagrangian multiplier) algorithm has less objective function, but requires length recalculations when any of the units violates the constraints as indicated in unit (3) where the output decision variables violated the cogeneration unit constraints and its effectiveness for large units is not known. The performance of the proposed algorithm has been critically assessed in comparison with other known algorithms like artificial bee colony used in chapter four of this research to highlight its merits. The development of genetic algorithm to optimize the cost of power generation is efficient, robust and likely to continue. As a result, further development of simpler approaches to cost of power generation modelling is of paramount importance.

It is thus shown in table 3.9 that the genetic algorithm method is superior to particle swarm optimization technique for this class of constraint optimization problem. Therefore, we compare the two optimization algorithms rather than the direct solution approach, for its lengthy calculation rules, difficult constraint-handling mechanism rules associated with cogeneration units when any of its units violates the constraint and the effectiveness of this method for large units is uncertain. A perusal of the results provided in table 3.9 shows that particle swarm optimization algorithm requires a considerably large number (over 103 in some cases) of iteration lines to converge, compared with the number of iteration lines steps required by the genetic algorithm method.

As pointed out earlier, particle swarm optimization algorithm has slow convergence speed; this slow convergence characteristic is inevitable, because of the sequential nature of the algorithm. Particle swarm optimization characteristics also indicate that the convergence speed could deteriorate with increase in number of units in the system. Plus, initial design variables are difficult to define. However, it is easy to see that the amount of computation involved in each step of the genetic algorithm is considerably less than that required for one generation of the particle swarm optimization algorithm. Therefore, comparing the performance of particle swarm optimization algorithm and artificial bee colony algorithm with the proposed algorithm (genetic algorithm) may not be proper. The reason being that the solutions provided by these algorithms (particle swarm optimization and artificial bee colony algorithm) do not even correspond to acceptable solutions. It does necessarily not imply that these algorithms cannot be developed for the combined heat and power economic problems, but the possibility of such implementations outperforming genetic algorithm solution seems very remote for this class of constraint optimization problem. The reason is that the attractiveness of the particle swarm optimization algorithms lies in their ability to provide solution to unstructured, very complex problems, which cannot be readily solved by other meta-heuristic approaches. Particle swarm optimization algorithm, being probabilistic search method, is not known to be computationally more efficient for problems that permit a genetic algorithm similar to the one proposed in this research. On the other hand, it is well known that convergence speed of artificial bee colony degenerates as the number of units' increases in the system with optimal solution seeming inaccessible. Therefore, it can be concluded that the proposed algorithm (genetic algorithm) outperforms all of the two algorithms applied in chapters 3 and 4 of this research. The analysis of the results further shows that proposed algorithm (genetic algorithm) reduces the system operational cost significantly compared to the other two algorithms. It therefore appears reasonable to conclude that the proposed (genetic algorithm) has extreme promise when compared to particle swarm optimization and other known combined heat and power economic dispatch meta-heuristic algorithms.

Table 3.9. Summary of Results Obtained from Direct Solution, Genetic and Particle Swarm Optimization Algorithms.

	Direct solution(Lagrangian multiplier method)	Genetic Algorithm(GA)	Particle Swarm Optimization
P_1(MW)	155.9	102.3	110.3
P_2(MW)	169.1	345.6	390.6
P_3(MW)	195	70.1	69.1
Q_3(MWth)	120	158.2	160
Q_4(MWth)	180	234.7	230.7
Cost(₦)	18933.8	21120.0	22917.0

CHAPTER 4- IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM USING ARTIFICIAL BEE COLONY ALGORITHM.

4.0 INTRODUCTION

This chapter addresses implementation of combined heat and power economic dispatch problem encountered in a system with simple cycle cogeneration unit using a novel meta-heuristic algorithm— the artificial bee colony— inspired by the operation of honey bee feeding food. Its efficiency is more constructive than some other meta-heuristic and optimization techniques. Data obtained from chapter three will aid the demonstration of artificial bee colony algorithm operation in this chapter. The test system considered consists of four units: two conventional power units, one combined heat and power unit and one heat-only unit. Constrained and convex optimization problems are encountered in many applications such as structural optimization, engineering design, economics and allocation and location problems. These are just a few of the scientific fields in which constraint optimization problems are frequently met. The considered problem in this chapter is reformulated to take the form of optimizing two functions, the objective function and the constraint violation function. The artificial bee colony algorithm was applied on the data while taking into account the power and heat units, operation bounds of the units and feasible operation region of the cogeneration unit.

Essentially, optimization design, having been widely applied in product design, is an extremely effective way of ensuring the product has an excellent performance, sheds weight and volume, and reduces product cost. The optimization problem is to find optimal solution of multi-modal problems in the feasible region. Thus, it provides multiple choices or multi-faceted information for decision makers. Many classical numerical methods for optimizing problem can actually get good results for some problems. But these rather continuous and differentiable methods have a strong constraint for the objective function. They equally have strong dependence on optimization problem. At the same time, the algorithm results are not only related to the selection of initial values, they are easily trapped in local minimum when selection is made wrongly. When dealing with a complex multimodal optimization problem, the traditional search method for single point often only searches uncertainty at one extreme point at a time. It therefore is basically invalid for the complicated multimodal optimization problem. The developing swarm intelligence algorithms become new effective approaches for solving multi-peak function optimization problems.

In recent years, the booming evolutionary algorithm with the global optimality, parallelism and efficiency has been widely used in function optimization problems. Aided by nature evolution, the evolutionary algorithm overcoming the drawback of traditional numerical method as a global optimization method for multiple clues is based on the population and random search mechanism. It has attracted widespread attention of evolutionary computation in the field of optimization application, from where various forms of evolution algorithm emerge endlessly. On the other hand, social insects' colonies like ants and bees have an instinctive ability termed "swarm intelligence". It is an advanced behavior that enables colonies of insects to solve problems beyond the capabilities of individual members by functioning collectively and interacting primitively amongst members of the group. In a honey bee colony for example, this behavior allows honey bees to explore the environment in search of flower patches (food sources) and then indicate the food source to the other bees of the colony when they return to their hive. Such a colony is not only characterized by self-organization, but is naturally adaptive and robust [66]. Based on MATLAB software, the Artificial Bee Colony Optimization Algorithm program was developed for combined heat and power economic dispatch problem. The example shows that this algorithm had no special requirements on the characteristics of optimal designing problems that have a fairly good universal adaptability and a reliable operation of program with a strong ability of global convergence. Like other swarm intelligent algorithms, artificial bee colony has some performance impeding demerits. It is perfect at exploration but poor at exploitation. Therefore, accelerating convergence speed and preventing local optima have become two vital and appealing objectives when using artificial bee colony algorithm in solving constraint optimization problems.

4.1Artificial Bee Colony Algorithm

Division of labor and self-organization are the constituent bases of a bee colony. Through a division of labor devoid of any centralized control in a self-organizing system, each covered unit may respond to local stimuli individually, and act together to accomplish a global task. Certain functions are accomplished by skilled individuals in an actual bee colony. These skilled bees try to optimize the aggregate nectar reserved in the hive, using methodical division of labor and self-organization. This algorithm is based on particle swarm intelligence inspired by the characteristics of honey bees finding food. The artificial bee colony algorithm, suggested by author [31], for actual variable optimization, is an optimization technique that simulates the searching operation of a bee colony. The minimal model of swarm intelligent search selection in a honey bee colony which the artificial bee colony algorithm simulates comprises three categories of bees: employed, onlooker, and scout bees. Half of the colony is made up of employed bees, whereas the second half consists of sentry bees. Employed bees account for utilization of the nectar sources earlier investigated. They equally undertake the responsibility of passing information to the waiting bees (sentry bees) in the hive about the quality of the food source plots they are utilizing. Onlooker bees wait in the hive and decide on a food source to utilize based on the information shared by the employed bees. Scouts either randomly search the environment to find a new food source depending on internal stimulation or based on possible external suggestion or clue (sign). To implement artificial bee colony algorithm, the considered optimization problem is first converted to the problem of finding the best parameter vector that minimizes an objective function. Then, the artificial bees randomly discover a population of initial solution vectors before iteratively improving them by moving towards better solutions by means of a neighbor search mechanism, while abandoning poor solutions.

Searching bees' emergent intelligent behavior is further summarized below:

1) At the commencement of the searching procedure, bees go about investigating the environment randomly in order to find a food sources.

2) Upon locating a food source, bees become employed searchers and begin to utilize the discovered source. Afterwards, they go back to the hive with the nectar for unloading. After unloading in the nectar, they either return to their discovered source plot directly or elect to share information about their source plot by executing a dance on the area. If their source is spent, they become scouts and start to randomly explore a newer source.

3) Onlooker bees waiting in the hive watch the dances advertising the profitable sources. They choose a sources plot depending on the frequency of the dance proportional to the quality of the source.

Food source position of in artificial bee colony constitutes a possible solution to the optimization problem. The nectar amount of a food source corresponds to the profitability (fitness) of associated solution. Each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of food sources existing around the hive (number of solutions in the population). The employed bee whose food source has been abandoned becomes a scout. Using the analogy between emergent intelligence in foraging of bees and the artificial bee colony algorithm, the principal constituents of the basic artificial bee colony algorithm can be further outlined thus:

4.2 Producing Initial Food Source Sites

If the search spaces considered being the environment of the hive that contains the food source plots, the algorithm starts with randomly producing food sources plots that correspond to the solutions in the search space. Initial food sources are produced randomly within the range of the parameters, and assuming the abandoned source is x_i then the scout bee replace this food source with new x_i as follows;

$$x_i^j = x_{\min}^j + rand\left[0,1\right]\left(x_{\max}^j - x_{\min}^j\right) \qquad (4.1)$$

Where $i = 1, 2, K, SN$, $j = 1, 2, K, D$. SN is the number of food sources and D is the number of optimization variables. Furthermore $x_{\max}^j$ and $x_{\max}^j$ are bounds of x_i in the j^{th} direction.

In addition, counters which store the number of trials of solutions are reset to zero in this phase. After initialization, the population of the food sources (candidate solutions) is subjected to repeat cycles of the search process of the employed bees, the onlooker bees and the scout bees.

4.3 Sending employed bees to the food sources sites

As mentioned earlier, each employed bee is associated with only one food source plot. Hence the number of food source plot is equal to the number of employed bees. An employed bee produces a modification on the position of the food source (solution) in her memory depending upon local information (visual information) and finds neighboring food source, and then evaluates its quality. In artificial bee colony, finding a neighboring food source is defined by:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \qquad (4.2)$$

Where $\phi_{ij}\left(x_{ij} - x_{kj}\right)$ is called step size, $k \in \{1,2,\ldots,SN\}$, $j \in \{1,2,\ldots,D\}$ are two randomly chosen indices. k must be different from i so that the step size has some significant contribution and ϕ_{ij} is a random number in $[-1,1]$.

As it is evident from equation (4.2) as the difference between the variables of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases. Thus, as the search approaches to the optimal solution in the search space, the step length is adaptively reduced. If a variable value produced by this operation exceeds its predetermined boundaries the variable can be set to an acceptable value. If the value of the variable exceed its boundary is set to its corresponding boundaries. If $x_i > x^i_{\max}$ then $x_i = x^i_{\max}$, if $x_i < x^i_{\min}$ then $x_i = x^i_{\min}$. After producing v_i within the boundaries a fitness value for a minimization problem can be calculated to the solution v_i by ;

$$Fitness_i = \begin{cases} 1/\left(1+f_i\right) & \text{if } f_i \geq 0 \\ 1+|f_i| & \text{if } f_i < 0 \end{cases} \qquad (4.3)$$

where f_i is cost value of the solution v_i. For maximization problems, the cost function can be directly used as a fitness function. A greedy selection is applied between x_i and v_i, the better one is selected depending on fitness values representing the nectar amount of the food sources at x_i and v_i. If the source at v_i is superior to that of x_i in terms of fitness values, the employed bees memorize the new position and forget the old one. Otherwise the previous position is kept in memory. If x_i cannot be improved its counter holding the number of trials is incremented by one, otherwise the counter is reset to zero.

4.4 Computing probability values involved in probabilistic selection

After all employed bees complete their searches, they share their information related to the nectar amount and the positions of their sources within the onlooker bees on the dance area. This is the multiple interaction features of the artificial bees of artificial bee colony. Onlooker bees evaluate the nectar information taken from all employed bees and choose a food source plot with a probability related to its nectar amount. This probabilistic selection depends on the fitness value of the solutions in the population. A fitness-base selection might be roulette wheel, ranking base, stochastic universal sampling, tournament selection etc. In basic artificial bee colony, roulette wheel selection scheme in which each slice is proportional to size to the fitness value is used in equation (4.4).

$$p_i = \frac{Fitness_i}{\sum_{i=1}^{SN} Fitness_i} \tag{4.4}$$

4.5 Food source site selection by onlookers based on the information provided by employed bees:

In the basic artificial bee colony algorithm, a random real number within the range [0, 1] is produced for each food source. If the probability value (p_i in equation (4.4)) connected with that food source is substantial than this random number then the onlooker bee generates an adjustment on the position of this food source plot by using equation (4.2) as in the case of the employed bee. After the food source is assessed, greedy selection is employed and the onlooker bee either memorizes the new position by forgetting the old one or keeps the old one. If solution x_i cannot be up-graded, its counter holding trial is increased by one; if not, the counter is reset to zero. This process is duplicated until all onlookers are disseminated onto food source sites.

4.6 Abandonment criteria: limit and scout production

In a cycle, after all employed bees and onlooker bees concluded their searches the algorithm scrutinize to see if there is any exhausted food source to be abandoned. However, to decide if a food source is to be abandoned, the counters which have been modernized during search are applied. If the value of the counter is substantial than the control variables the artificial bee colony algorithm, known as the "*limit*", then the source associated with this counter is presumed to be finished and is abandoned. The food source abandoned by its bee is returned with a new food source is discovered by the scout, which constitutes the negative feedback procedure and fluctuation property in the self-organization of artificial bee colony. This is simulated by generating a plot position randomly and returning it with the abandoned one. Assume that the abandoned source is x_i , and then the scout randomly discovered a new food source to be replaced with x_i. This operation can be defined as equation (4.1). In the basic artificial bee colony, it is presumed that only one source can be finished in each cycle, and only one employed bee can be a scout. If more than one counter surpasses the "*limit*" values, one of the maximum ones might be chosen programmatically.

4.7 Artificial Bee Colony Optimization for Combined Heat and Power Economic Dispatch

In order to adapt the artificial bee colony algorithm for solving constrained optimization problems in this research, we adopted Deb's constrained handling method instead of the selection process (greedy selection) of the artificial bee colony algorithm since Deb's technique consists of very simple three heuristic rules. Deb's technique employs a tournament selection operator, where two solutions are compared at a time, and the following criteria are always enforced:

1. Any feasible solution is preferred to any infeasible solution,
2. Among two feasible solutions, the one having better objective function value is preferred,
3. Among two infeasible solutions, the one having smaller constraint violation is preferred. Because initialization with feasible solutions is very time consuming process and in some cases it is impossible to produce a feasible solution randomly, the artificial bee colony algorithm does not consider the initial population to be feasible.

The structure of the algorithm already directs the solutions to feasible region in running process due to the Deb's rules employed instead of greedy selection. Scout production process of the algorithm provides a diversity mechanism that allows new and probably infeasible individuals to be in the population. In order to produce a candidate food position from the old one in memory, the adapted artificial bee colony algorithm uses the following expression:

$$v_j = \begin{cases} x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) & \text{if} \quad R_j < MR \\ x_{ij} & \text{otherwise} \end{cases} \tag{4.5}$$

where $k \in \{1, 2, K, SN\}$ is randomly chosen index. Although k is determined randomly, it has to be different from i . R_j is randomly chosen real number in the range [0, 1] and MR a modification rate, is a control parameter that controls whether the parameter x_{ij} will be modified or not. In the version of the artificial bee colony algorithm proposed for constrained optimization problems, artificial scouts are produced at a predetermined period of cycles for discovering new food sources randomly. This period is another control parameter called scout production period of the algorithm.

Step 1: Initialization of the control variables.

Variables of the basic artificial bee colony algorithm include the following sizes of colony: (NP), the number of food sources (SN=NP/2), the limit for scout, L= SN*D.D is the dimension of the problem (number of optimization variable) and a termination criterion or maximum cycle number.

Step 2: Producing initial food source sites.

The initial food sources vector

$X_i = \left[P_1, P_2, \ldots, P_\alpha, P_{\alpha+1}, \ldots, P_\beta, H_{\alpha+1}, H_\beta, H_{\beta+1}, H_{\beta+2}, \ldots, H_n\right]^T$ $i = 1, 2, \ldots, NP$ is determined by equation (4.7) and setting $P: U\left(P^{\min}, P^{\max}\right)$ and $H: U\left(H^{\min}, H^{\max}\right)$. $U\left(a, b\right)$ denotes a uniform random variable range over [a, b] and evaluates the fitness value using equation (4.7);it then chooses SN, the best food source on the basis of highest fitness value as initial food sources, and set the cycle = 1; the trial number of each solution X_i, $trial_i$ is equal to zero.

Here $X = \left[X_1, X_2, \ldots, X_{NP}\right]$, $X_i = \left[X_{i1}, X_{i1}, \ldots, X_{iD}\right]^T$ and $P_d = P_D - \sum_{i=1}^{\beta} P_i$, $H_d = H_D - \sum_{\substack{i=\alpha+1 \\ \neq d}}^{n} H_i$.

$$\left.\begin{aligned} P_{ij} &= P_j^{\min} + rand\left(0,1\right)\left(P_j^{\max} - P_j^{\min}\right) \\ H_{ij} &= H_j^{\min} + rand\left(0,1\right)\left(H_j^{\max} - H_j^{\min}\right) \end{aligned}\right\} \tag{4.6}$$

$$Fitness\left(F\right) = \frac{1}{\left[\sum_{i=1}^{\alpha} 1 + F_{ti}\left(P_i\right) + \sum_{i=\alpha+1}^{\beta} 1 + F_{ci}\left(P_i, H_i\right) n + \sum_{i=\beta+1}^{n} 1 + F_{hi}\left(H_i\right)\right]} \tag{4.7}$$

Step 3: Sending employed bees to the food sources [SN] and assigning the nectar amount.

In this step, each employed bee generates a new solution V_i by using equation (4.2); and computes the fitness value of the new solution by using equation (4.7) to satisfy all constraints. If the fitness of the new one is higher than that of the previous one, the employed bee memorizes the new position and forgets the old one. Otherwise, the employed bee keeps the old solution.

Step 4: Sending the onlooker bees to the food sources depending on their amount of nectar

This step requires computation of the probability value P_i of the solution X_i by means of their fitness value using equation (4.4). An onlooker bee selects a means to update its solution depending on the probabilities; and also determines a neighboring solution around the chosen one. In the selection procedure for the first onlooker, a random number is produced between 0-1, and if this number is less than P1, the solution is updated using equation (4.2). Otherwise, the random number is compared with P2;and if less than that, the second solution is chosen. Otherwise, the probability of a third solution is checked. This process is repeated until all onlookers have been distributed to solutions. The distributed onlooker bees update their own solutions just as the employed bees do.

Step 5: Sending the scouts to the search area to discover new food sources.

If the solution X_i is not improved through steps 3 and 4, the $trial_i$value of solution $X_i$will be increased by 1. If the $trial_i$of the solution is more than the predetermined *"limit"*, the solution X_i is considered to be an abandoned solution. Meanwhile, the employed bee will be transformed into a scout. The scout randomly generates the new solution by equation (4.6) and then compares the fitness of new solution with the old one. If the new solution is better than the old one, it is replaced with the old one to set its own *trial*$_i$ into zero. This scout will be transformed into employed bee. Otherwise, the old one is retained in the memory.

Step 6: Record the best solution.

In this step, the best solution so far is recorded and it increases the cycle by 1.

Step 7: Check the termination criterion.

If the cycle is **equal** to the maximum cycle number, then the algorithm is finished; otherwise go to step 3.

4.8 Modeling of combined heat and power economic dispatch problem using artificial bee colony Algorithm

AMATLAB sub-routine was created in section 4.7 of chapter 4 to determine the combined heat and power economic dispatch decision variables of different units in the system. The program, including sub-routines, were created with the inputs (tables 3.1, 3.2, 3.3 and 3.4 respectively) being the variables; and the output decision variables being the power, including heat from respective units plus objective function value. Once again, the sub-routines created in MATLAB showed the variables as the inputs to the program; the output decision variables being the power, heat from individual units plus objective function value. As seen in Table 4.1, the artificial bee colony algorithm has found the global minimum of three out of four problems (units 1,3 and 4) through 100 iteration lines. On one problem in unit 2, the artificial bee colony algorithm could not find the optima in the specified maximum number of cycles, that is, 100.The result showed that artificial bee colony algorithm has a slow convergence speed of 100 which makes it fall into local optimum. The total fuel cost from artificial bee colony algorithm was (₦21124.1) which is large compare with proposed (genetic) algorithm. This characteristic of long convergence associated with artificial bee colony makes it produce solutions that are neither well-spread and diverse, nor effective for optimization problem of this class.

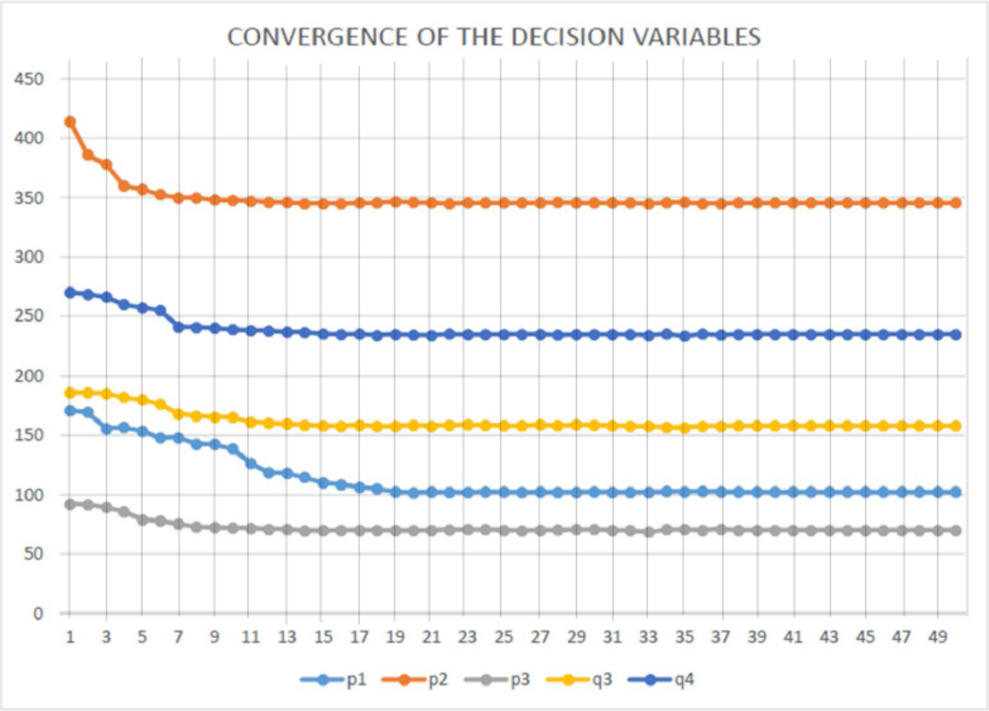


Fig. 4.1 Convergence of the decision variables

The graph above represents convergence characteristics of the output decision variable in artificial bee colony algorithm**.** In general, initially the particle state is not at equilibrium. So it is necessary to analyze whether the particle will eventually move towards equilibrium or not, that is, whether or not the optimization algorithm will converge. From the result of the graph, it can be concluded that the Eigen-values of the matrix plays a vital role in explaining the time behavior of potential solutions. The necessary and sufficient condition for equilibrium point to be stable is that the magnitude of Eigen-values of the matrix should be less than unity. In this case, the potential solutions will eventually settle at equilibrium and the algorithm will converge. It therefore can be concluded that artificial bee colony algorithm performs better when parameters are considered from the convergent region.

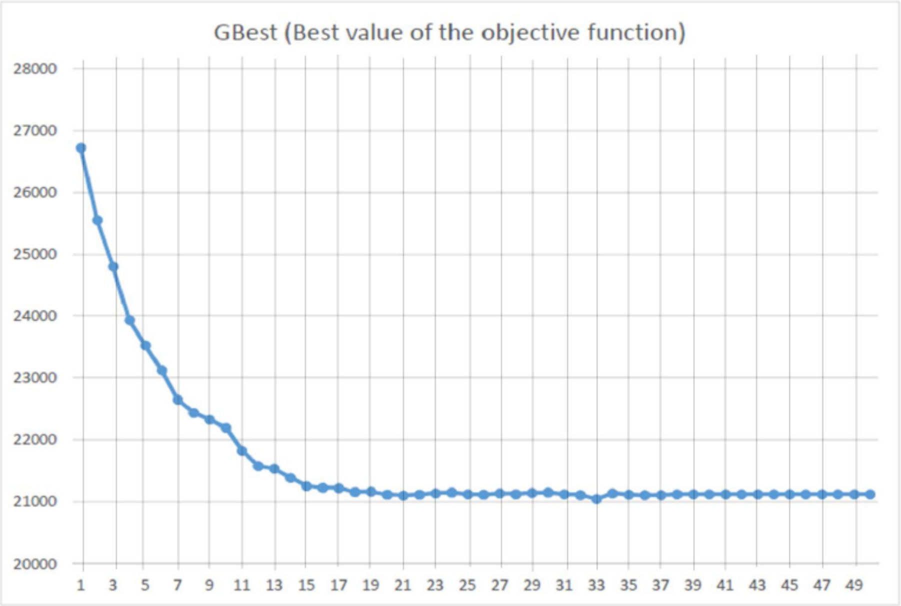


Fig. 4.2 GBest (Best value of the objective function)

Figure 4.2 illustrates the best value of objective function characteristic from artificial bee colony algorithm. The result shows that objective functions converge after 100 iteration lines; and the best objective function value of artificial bee colony algorithm was ₦21124.1. Analysis of finding the conditions, under which an algorithm converges to an equilibrium point, plays a vital role in making the algorithm methodical, authentic and correct. For nature inspired algorithms like artificial bee colony, the solution update process depends upon the guided random search, which makes the algorithm's nature a probabilistic one. This probabilistic nature makes the convergence analysis a difficult task. Convergence analysis of artificial bee colony algorithm is performed using results from the theory of dynamical systems. Also, the condition of convergence of algorithm to equilibrium point is subject to parameters. It can be concluded that the algorithm performs better when parameters are considered from the convergent region. Lastly, for the purpose of analyzing the search behavior of artificial bee colony algorithm a study of the movement of solutions in the search space is important. It is observed that the artificial bee colony algorithm converges quickly in early iterations and hence, the number of maximum runs can be decreased to save the solution time.

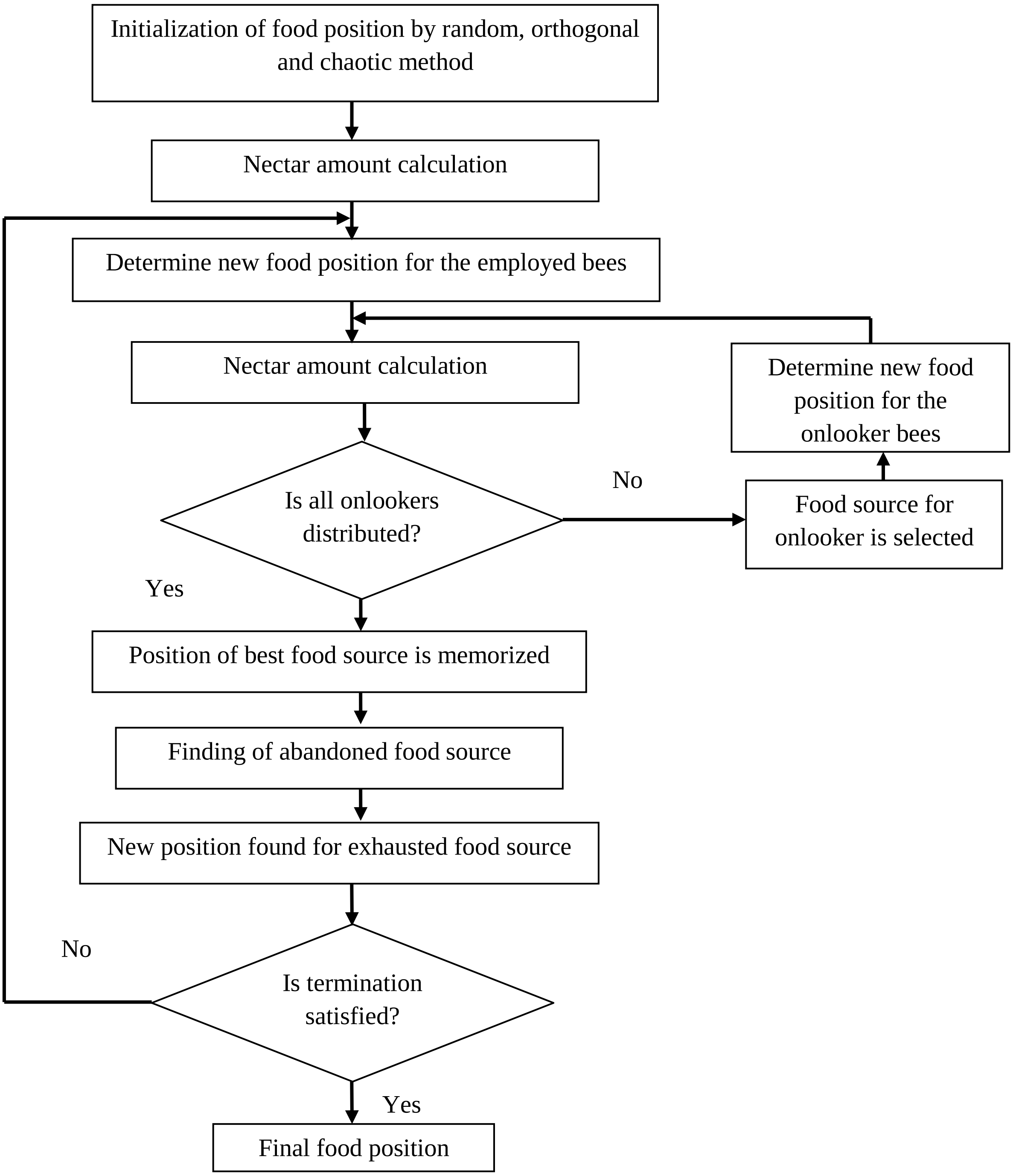

Initialization of food position by random, orthogonal and chaotic method
Nectar amount calculation
Determine new food position for the employed bees
Nectar amount calculation
Determine new food position for the onlooker bees
Is all onlookers distributed?
No
Food source for onlooker is selected
Yes
Position of best food source is memorized
Finding of abandoned food source
New position found for exhausted food source
No
Is termination satisfied?
Yes
Final food position

Fig.4.3 Typical flowchart for Artificial Bee Colony Optimization algorithm

4.9 ANALYSIS AND DISCUSSION OF ARTIFICIAL BEE COLONY RESULT

Artificial bee colony algorithm has been applied for combined heat and power economic dispatch for a 4-unit test system. Cost function parameters along with feasible region coordinate of combined heat and power unit are taken from tables (3.1), (3.2), (3.3) and (3.4) respectively. The test system comprises two conventional power units, one cogeneration unit and a heat-only unit. The heat-power feasible operation region of the cogeneration unit is illustrated in figure 3.3. As explained in chapter 3, combined heat and power economic dispatch has been formulated with the objective of minimizing fuel cost. Table 4.1 shows power and heat output decision variables of four-unit systems with power demand = 520MW and heat demand =300Mwth using artificial bee colony algorithm. The result confirms that artificial bee colony algorithm has that tendency of falling into a local optimum that slows down the convergence speed. As seen from Table 4.1, artificial bee colony algorithm has found the global minimum of the three of four problems (P1, P3, Q3 and Q4) through 100 cycles. On one problem, P2, the artificial bee colony algorithm could not find the optima in the specified maximum number of cycles. Furthermore, unlike proposed algorithm, artificial bee colony algorithm converges in relatively large number of iteration lines (100) with cost function value (₦21124.1). This implies that due to its ineffective searching capability, artificial bee colony algorithm has a slow convergence speed which makes it impossible to produce diverse, widespread and extreme solutions.

Table 4.1. Results Obtained from Artificial bee Colony algorithm

P_1(MW)	P_2(MW)	P_3(MW)	Q_3(MWth)	Q_4(MWth)	Cost(₦)
102.4	345.7	70.1	158.2	234.8	21124.1

4.10 CONTRIBUTIONS

Several existing researches listed in chapter two concentrated on possible approaches to solving the combined heat and power economic dispatch problem. However, it was discovered that given that it is a non-convex and nonlinear problem of optimization no mathematical or meta-heuristic algorithm that can guarantee an optimal global solution for it. For the high economic saving potentials of better algorithms nonetheless, this chapter focused on solution of combined heat and power economic dispatch problem using artificial bee colony algorithm. Artificial bee colony algorithm is a heuristic optimization algorithm, based on the intelligent search behavior of honey bee swarm. It provides a population-based search procedure in which individuals (foods positions) are modified by artificial bees, whose collective aim is the discovery of newer food sources with the highest nectar amount. Thus, the main contributions of this work can be summarized as follows:

1. It compared proposed algorithm with other swarm optimization algorithms like particle swarm optimization to investigate the method with better results using the provided test system.

2. It adopted artificial bee colony algorithm for dealing with convex optimization problems.

4.11 CONCLUSION

For an efficient solution of combined heat and power economic dispatch problem encountered in a simple cycle cogeneration unit, a new perspective based on artificial bee colony algorithm was proposed in chapter four. Combined heat and power economic dispatch is cumbersome to model. The dependence of heat and power constraints on each other as well as the feasible operating region of the cogeneration unit(s). Different attributes such as constraints, feasible operation region of combined heat and power unit, and capacity limits of units are taken into account in the formulation. A subroutine was developed in MATLAB to calculate combined heat and power economic dispatch variables using artificial bee colony algorithm. The sub-routine was created using data from tables (3.1), (3.2), (3.3) and (3.4) as inputs to the program; while the output remained the heat, power and cost function values. Heat and power demands are 300Mwth and 520MW respectively. Figure 4.1 shows convergence characteristic result of the algorithm which demonstrates how Eigen-values of the matrix play a vital role in explaining the time behavior of the potential solutions. The necessary and sufficient condition for equilibrium point to be stable is that the magnitude of Eigen-values of the matrix should be less than unity. Hence the potential solutions will eventually settle at equilibrium and the artificial bee colony algorithm will converge. The result in fig 4.1 further indicates that for nature-inspired algorithms like the artificial bee colony, for instance, the solution update process depends upon the guided random search. This makes the algorithm nature probabilistic. The probabilistic nature of this algorithm makes the convergence analysis a difficult task. Convergence analysis of artificial bee colony algorithm is performed via results from the theory of dynamical systems. Lastly, the convergence condition of this algorithm to equilibrium point depends upon the parameters. Thus, it can be interpreted that artificial bee colony algorithm performs well when parameters are considered from the convergence region. Therefore, it has been established that artificial bee colony algorithm can be efficiently applied for solving constrained optimization problem. The performance of the artificial bee colony algorithm can be also tested for real engineering problems existing in the literatures and compared with that of other algorithms such as particle swarm optimization, differential evolution etc. Also, the effect of constraint handling mechanisms on the performance of the artificial bee colony algorithm can be scrutinized in future works.

CHAPTER 5- IMPLEMENTATION OF COMBINED HEAT AND POWER ECONOMIC DISPATCH PROBLEM USING DIFFERENTIAL EVOLUTION ALGORITHM

5.0 INTRODUCTION

In this section, we present a flexible algorithm to solve the combined heat and power economic dispatch problem. This involves the Differential Evolution (DE) procedure described below. Differential evolution is a population-based stochastic algorithm that optimizes a problem by trying to improve candidate solutions iteratively with regard to a given measure of quality. This algorithm employs a greedy and less stochastic method when solving a problem better than evolutionary algorithms like genetic algorithms, evolutionary programming, genetic programming etc. To evolve from a randomly generated starting population to a final solution, Differential Evolution joins arithmetic operators with classical operators like mutation, recombination and selection. The algorithm is different from basic genetic algorithm in its application of perturbing vectors— the actual difference between two randomly selected parameter vectors. This is an idea taken from the operators of simplex optimization method. Differential evolution algorithm was proposed by Storn and Price and was efficiently employed in the optimization problem of some familiar non-convex, on-linear and non-differentiable functions under a number of constraints.

Furthermore, differential evolution is a current heuristic algorithm outlined to optimize problems that have continuous domains. In differential evolution, the decision variables, that is, heat and power are represented in the chromosomes by a real number. The initial population of differential evolution algorithm is randomly created, thereafter evaluated and afterwards, selection process takes place. At the time of selection, three parents are taken to create a single offspring which competes with a parent to determine who passes to the next generation. The principal idea of differential evolution is to modify the search during the evolutionary process. At the beginning of evolution process, the perturbation is huge, because the parent populations are far away from each other. Unlike genetic algorithm that creates two offspring, Differential Evolution creates one instead by addition of weighed difference vector between two parents to a third parent. In respect of single objective optimization encountered in this research, if the resulting vector results in a lower objective function value than a prearranged population member, then the newly created vector replaces the vector with respect to which it was contrasted. Also, the best parameter vector is assessed for every generation to maintain the progress achieved during minimization process. The fittest of offspring participates one to one with that of analogous or alike parent in differential evolution which is different from other class of evolutionary algorithm. This one to one rivalry leads to fast convergence rate.

5.1 The Differential Evolution (DE) Algorithm

Differential evolution algorithm based on the theory of natural selection and evolution has Natural Phenomena in its design [65]. It works on real values and is also an evolutionary algorithm [66] which is used for minimization. Differential evolution program implements the natural selection theory by designing three basic operators, namely:

- Selection
- Crossover
- Mutation

The importance of understanding the disparity between Differential Evolution **(**DE) and Genetic Algorithm (GA), led to the observation that both employ the same operators. However, their application ordering is different. In DE, the order is Mutation, Crossover and then Selection. The program repeats the following process for fixed number of times:

Parent population Offspring population Mutation, crossover and selection ***n****umber of times*

5.2 WORKING EXAMPLE OF DIFFERENTIAL EVOLUTION ALGORITHM

Owing to the rather complex procedure involved in the DE algorithm, it is important to illustrate the working procedure with a concrete example. The procedure can be explained in four parts, namely:

- Problem definition
- Differential evolution parameters
- Initialization
- Differential evolution position update

We consider a simple optimization problem (sphere function) with two decision variables.

There are many variants of differential evolution known equally as differential evolution strategies. In this example we employ the only the differential evolution /1/rand/bin strategy for explanation purpose.

5.2.1 Problem definition

Consider the optimization problem:

$$\text{Min } f\left(x\right)=x_1^2+x_2^2 \qquad -5\leq x_1,x_2\leq 5$$

5.2.1.1 Differential Evolution and Problem Parameters

Population size $N=5$

Dimension of the problem $= 2$

Stopping Criteria $=$ Number of iterations $= 2$

Scaling factor $F = 0.5$

Crossover probability $P_{cr} = 0.7$

5.2.1.2 Initialization

Position	Function value
$x_1 = (1.7667, -4.1337)$	20.2089
$x_2 = (4.8071, 0.6642)$	23.5502
$x_3 = (-2.9232, -4.0439)$	24.8983
$x_4 = (-4.3747, -4.7421)$	41.6270
$x_5 = (-1.6587, 0.5680)$	3.0741

5.2.1.3 Differential evolution Position Update

Strategy: DE/rand/1/bin

In this notation ‘*DE*’ stands for differential evolution, ‘*rand*’ makes the random selection of the target vector, ‘1’ is for the differentials while ‘bin’ implies the type of crossover which is binary crossover in this case.

Generation-1

In differential evolution, position update is carried out by two operators, mutation and crossover. It is different from the evolutionary algorithms where the solution is mutated after the crossover. But here mutation is first operator and then crossover is applied.

Mutation

The mutation operator is employed in differential evolution, to create a *Trial Vector u* for each solution (*Parent Vector*). It is done by mutating a *Target Vector* (selection of *Target Vector* depends on different strategies of DE).

Trial Vector = ***Target Vector***+ ***Scale Factor*** × (*Randomly selected solution*$_1$ – *Randomly selected solution*$_2$)

Here, it should be noted that *Randomly Selected Solution*$_2$, *Target Vector* and *Parent Vector* should be different from one another. *Scale Factor* is a user defined parameter $\in (0, \infty)$. The recommended value for this parameter is 0.5.

Now, to begin with mutation, select the first as the *Parent Vector*.

Suppose $x_1 = (1.7667, -4.1337)$ is the Parent Vector.

Corresponding function value is $f(x_1) = 20.2089$.

For mutation, let *Target Vector* (selected randomly from current population) is $x_4 = (-4.3747, -4.7421)$

Randomly selected solution$_1 = x_5$

Randomly selected solution$_2 = x_3$

Let *Scale Factor* $= 0.5$

Now the *Trial Vector* u_1 is computed as:

$$\begin{aligned} u_{11} &= x_{41} + 0.5 \times (x_{51} - x_{31}) \\ &= -4.3747 + 0.5 \times (-1.6587 - (-2.9232)) \\ &= -3.7425 \end{aligned}$$

$$\begin{aligned} u_{12} &= x_{42} + 0.5 \times (x_{52} - x_{32}) \\ &= -4.7421 + 0.5 \times (0.5680 - (-4.0439)) \\ &= -2.4362 \end{aligned}$$

Trial Vector $u_1 = (-3.7425, -2.4362)$

At this point, we will check if the *Trial Vector* u_1 is within the search space or not. Clearly

$-5 \le x'_{11}, x'_{12} \le 5$, we accept it. This completes the mutation process.

Crossover

In the differential evolution crossover, an *Offspring* is generated using the discrete recombination of *Parent Vector* and *Trial Vector*. Consider the crossover probability $P_{cr} = 0.7$ since we are going to calculate *Offspring* corresponding to the *Parent Vector* x_1, let us denote the *Offspring* by x'_1.

$$\textit{Offspring}\ x'_{1j} = \begin{cases} u_{1j}, & \text{if } j \in I \\ x_{1j}, & \text{otherwise} \end{cases}$$

Here, the index set *I* consists of the crossover points which depend on the choice of crossover type and the crossover probability. Binomial crossover is quite popular in DE. In Binomial crossover, the crossover points are selected from the set $\{1, 2, K, \text{problem dimension}\}$ with probability P_{cr}. This may lead to a situation where no point is selected for *I* (more probable for the small dimension problems. For example, the current problem is only of dimension 2). If this happens, i.e. $I = \varnothing$ then there will not be any change in the *Offspring* and it will be the same as a parent. In order to avoid this situation,

I is always considered a non-empty set by including a random point (say j_0) from the set $\{1, 2, K, \text{problem dimension}\}$ initially.

Thus we set $I = \{2\}$.

Now other crossover points are selected using the following algorithm:

```
$for\ j \in \{1, 2, K, problem\ dimension\}$
$if\ \text{rand}(0,1) < P_{cr}\ and\ j \neq j_0$
$I = I \cup \{j\};$
$end$
```

For this example, let $\text{rand}(0,1) = 0.67$. Then $I = \{1, 2\}$.

Now, the *Offspring* x_1' will be formed from the *Parent Vector* $x_1 = (1.7667, -4.1337)$ and *Trial Vector* $u_1 = (-3.7425, -2.4362)$. Since the crossover points have both the dimensions, both the variables of the *Offspring* x_1' are selected from the *Trial Vector* $x_1 = (1.7667, -4.1337)$.

Thus the *Offspring* $x_1' = (-3.7425, -2.4362)$.

At this point, we will decide whether in the population *Parent Vector* x_1 or *Offspring* x_1' will survive. For that we will compare the function values of x_1 and x_1'.

Since $f(x_1') = 19.9417$ is better than $f(x_1) = 20.2089$, therefore for the next generation in the population, *Offspring* x_1' will survive and *Parent Vector* x_1 will die out, hence x_1' will replace x_1. Thus new $x_1 = (-3.7425, -2.4362)$.

The same procedure will be applied to the next solution where $x_2 = (4.8071, 0.6642)$ with corresponding function value of 23.5503.

Mutation

Target Vector $= x_5$

After applying the same procedure, the *Trial Vector* $u_2 = (-1.2491, -0.2358)$ which will be accepted as both the variables, are within the search space.

Crossover

Let $I = \{1, 2\}$ using the Binomial crossover.

Hence *Offspring* $x_2' = (-1.2491, -0.2358)$, and $f(x_2') = 1.6158$.

Clearly $f(x_2') < f(x_2)$, so we accept the new $x_2 = (-1.2491, -0.2358)$

For the solution x_3

Mutation

Target Vector $= x_4$

Randomly selected solution$_1 = x_1$
Randomly selected solution$_2 = x_5$

After applying the same procedure, the *Trial Vector* $u_3 = (-5.4167, -6.2443)$. Now since u_{31} and u_{32}, both are beyond the search space. Therefore, we will pull the solution to the corresponding boundary of the search space, i.e.

Trial Vector $u_3 = (-5, -5)$

Crossover

Let $I = \{1, 2\}$ using Binomial crossover.

Hence *Offspring* $x_3' = (-5, -5)$, and $f(x_3') = 50$.

Clearly $f(x_3') > f(x_3)$, thus the new $x_3 = (-2.9232, -4.0439)$ which is the *Parent Vector*.

Applying the same procedure to the remaining two solutions, we get:

new $x_4 = (-4.3748, -4.3402)$ with $f(x_4) = 37.9765$

new $x_5 = (-1.6587, 0.5681)$ with $f(x_5) = 3.0741$

After first generation, the updated population is

Position	**Function value**
$x_1 = (-3.7425, -2.4362)$	19.9417
$x_2 = (-1.2491, -0.2358)$	1.6158
$x_3 = (-2.9232, -4.0439)$	24.8983
$x_4 = (-4.3748, -4.3402)$	37.9765
$x_5 = (-1.6587, 0.5680)$	3.0741

We observe that solutions x_1, x_4 and x_5 are modified. Furthermore, after the *second generation,* the best solution is $x_4 = (0.5233, -0.0876)$ with $f(x_4) = 0.2815$ which is better than that of first generation.

5.3 Combined Heat and Power Economic Dispatch (CHPED)

The combined heat and power economic dispatch problem of interest to us is presented as follows. Given the quadratic fuel cost function of power-only, cogeneration and heat-only units in Naira we have:

$$\begin{aligned}
c_{e,i}(p_i) &= \alpha_i + \beta_i p_i + \gamma_i p_i^2 && \textit{cost function of power only unit}\\
c_{c,i}(p_i, q_i) &= \alpha_i + \beta_i p_i + \gamma_i p_i^2 + \delta_i q_i + \varepsilon_i q_i^2 + \zeta_i p_i q_i && \textit{cost function of cogeneration unit}\\
c_{h,i}(q_i) &= \alpha_i + \delta_i q_i + \varepsilon_i q_i^2 && \textit{cost function of heat only unit}
\end{aligned} \tag{5.1}$$

Here, α_i, β_i and γ_i are the cost coefficients of i^{th} power-only unit, $\alpha_j, \beta_j, \gamma_j, \delta_j, \varepsilon_j$ and ζ_j are the cost coefficients for the j^{th} cogeneration unit, α_k, δ_k and ε_k represent the coefficient of k^{th} heat-only unit. The objective function of the CHPED problem is a cost function to be minimized, subject upon the equality and inequality constraints.

The objective function is written as:

$$\text{Min } C = \sum_{i\in e} c_{e,i}(p_i) + \sum_{i\in c} c_{c,i}(p_i, q_i) + \sum_{i\in h} c_{h,i}(q_i) \tag{5.2}$$

where the pair (p_i, q_i) represent respectively the electrical power and heat output of the i^{th} unit, $c_{e,i}(p_i)$, $c_{c,j}(p_j, q_j)$ and $c_{h,k}(q_k)$ constitute the fuel cost function of i^{th} power-only unit, fuel cost function of j^{th} cogeneration unit and fuel cost function of k^{th} heat-only unit, respectively.

Subject to: there all power created by power unit, plus the real power created by cogeneration unit being equal to the real power demand of power systems abandoning power loss. This is stated mathematically in equation 3 below:

$$\sum_{i\in e} p_i + \sum_{i\in c} p_i = p^{demand} \tag{5.3}$$

Comparably, the total heat created by boilers plus the active heat created by cogeneration unit are equal to the heat demand, abandoning heat loss, and can be stated thus:

$$\sum_{i\in e} q_i + \sum_{i\in h} q_i = q^{demand} \tag{5.4}$$

Where p^{demand} and q^{demand} are the total heat and power demands of the system, respectively, in the heat equality constraint, heat losses are postulated to be zero. The reason being that no research work about heat losses during heat transmission process to heat loads has been carried out. For clarity, that postulation was employed in this research. Therefore, heat losses are negligible. Furthermore, if heat losses are a function of heat outputs similar to power loss function or a constant, heat balance constraint will be solved simply and successfully.

$$p_i^{\min} \le p_i \le p_i^{\max} \tag{5.5}$$

$$q_i^{\min} \le q_i \le q_i^{\max} \tag{5.6}$$

Thus the above problem is summarized as:

$$\text{Min } C = \sum_{i \in e} c_{e,i}(p_i) + \sum_{i \in c} c_{c,i}(p_i, q_i) + \sum_{i \in h} c_{h,i}(q_i)$$

$$\sum_{i \in e} p_i + \sum_{i \in c} p_i = p^{demand}$$

$$\sum_{i \in e} q_i + \sum_{i \in h} q_i = q^{demand}$$

$$p_i^{\min} \le p_i \le p_i^{\max}, \qquad q_i^{\min} \le q_i \le q_i^{\max}.$$

5.4 Implementation of Combined Heat and Power Economic Dispatch Problem Using the Differential Evolution:

We begin with the Pseudo-code for the differential evolution Algorithm

```
1.  Input: Fitness function, Lb, Ub, $N_p$, T, F
2.  Evaluate fitness (f) of P
    fort = 1 to T
    for i = 1 to $N_p$
            Generate the donor vector ($V_i$) using mutation
            Perform crossover to generate offspring ($U_i$)
    end
    for i = 1 to $N_p$
    Bound$U_i$
    Evaluate fitness ($f_{Ui}$) of $U_i$
              Perform greedy selection using $f_{Ui}$and $f_i$ to update P
    end
    end
```

$$V = X_{r_1} + F\left(X_{r_2} - X_{r_1}\right)$$

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ or } j = \delta \\ x^j & \text{if } r > p_c \text{ or } j \ne \delta \end{cases}$$

$$\left.\begin{aligned} X_i &= U_i \\ f_i &= f_{U_i} \end{aligned}\right\} \text{ if } f_{U_i} < f_i$$

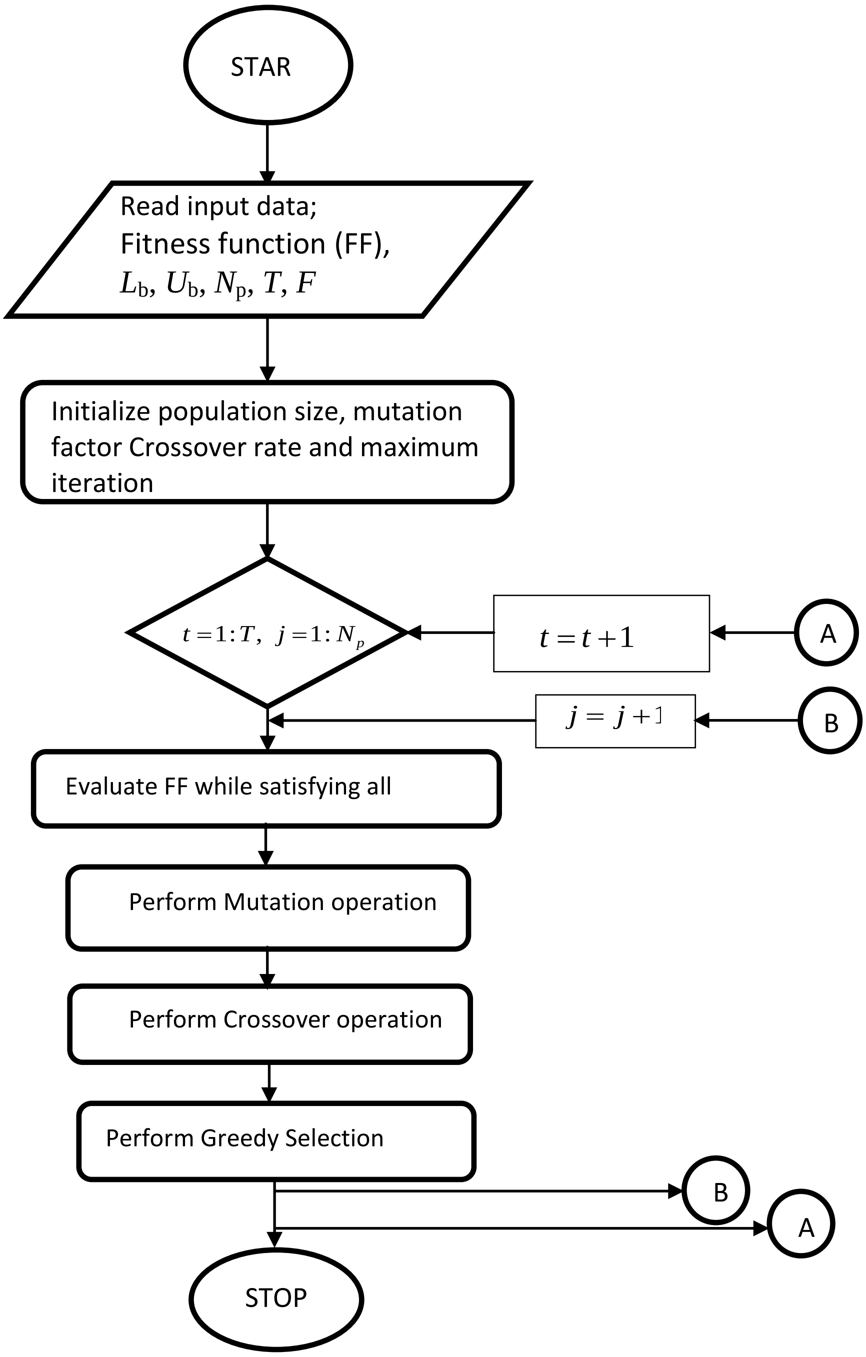

STAR
Read input data;
Fitness function (FF),
L_b, U_b, N_p, T, F
Initialize population size, mutation factor Crossover rate and maximum iteration
$t=1:T,\ j=1:N_p$
$t=t+1$
A
$j=j+1$
B
Evaluate FF while satisfying all
Perform Mutation operation
Perform Crossover operation
Perform Greedy Selection
B
A
STOP

Fig. 5.1Flowchart of Differential Evolution Procedure

5.5 RESULTS

The results of the MATLAB differential evolution code are presented below.

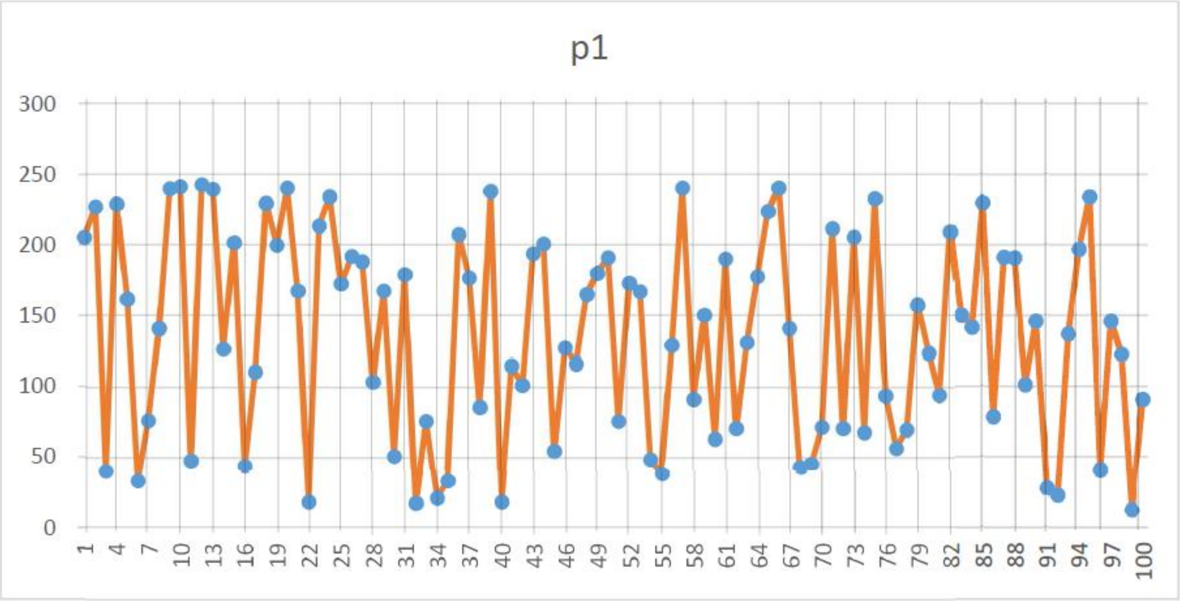


Fig. 5.2 Profile of the decision variable p1

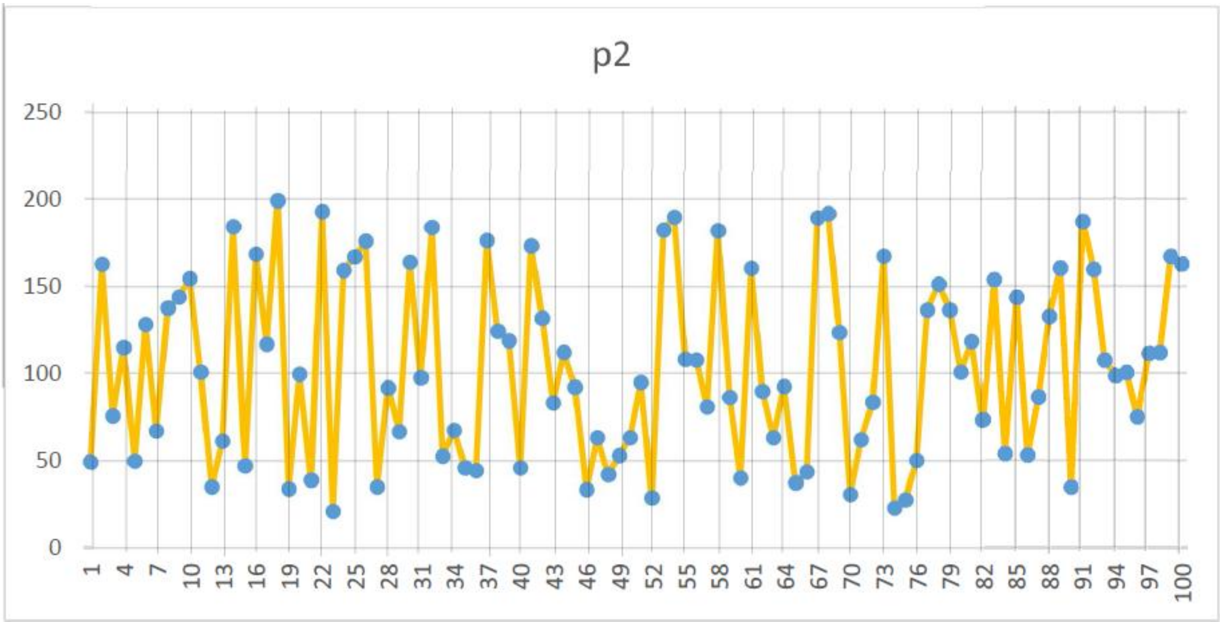


Fig. 5.3 Profile of the decision variable p2

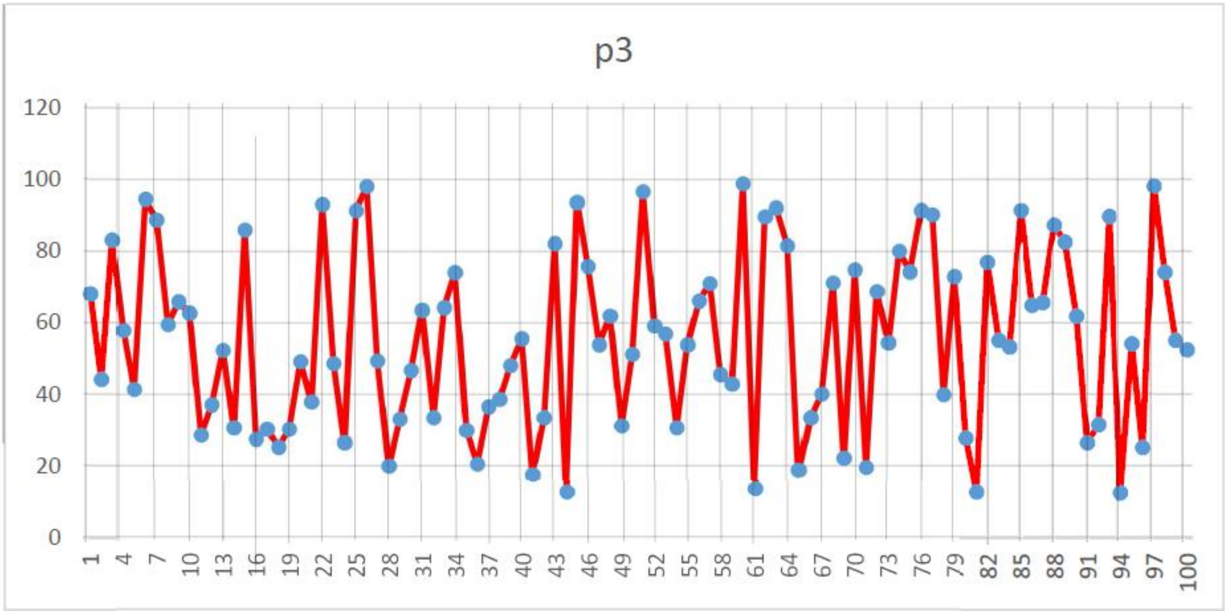


Fig. 5.4 Profile of the decision variable p3

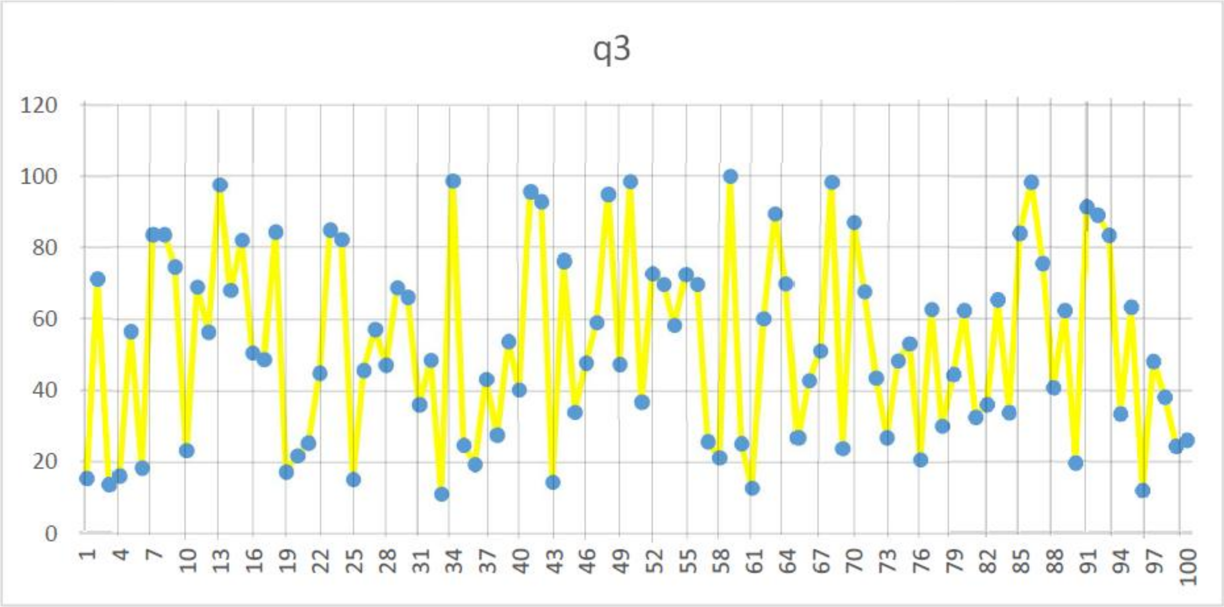


Fig. 5.5 Profile of the decision variable q3

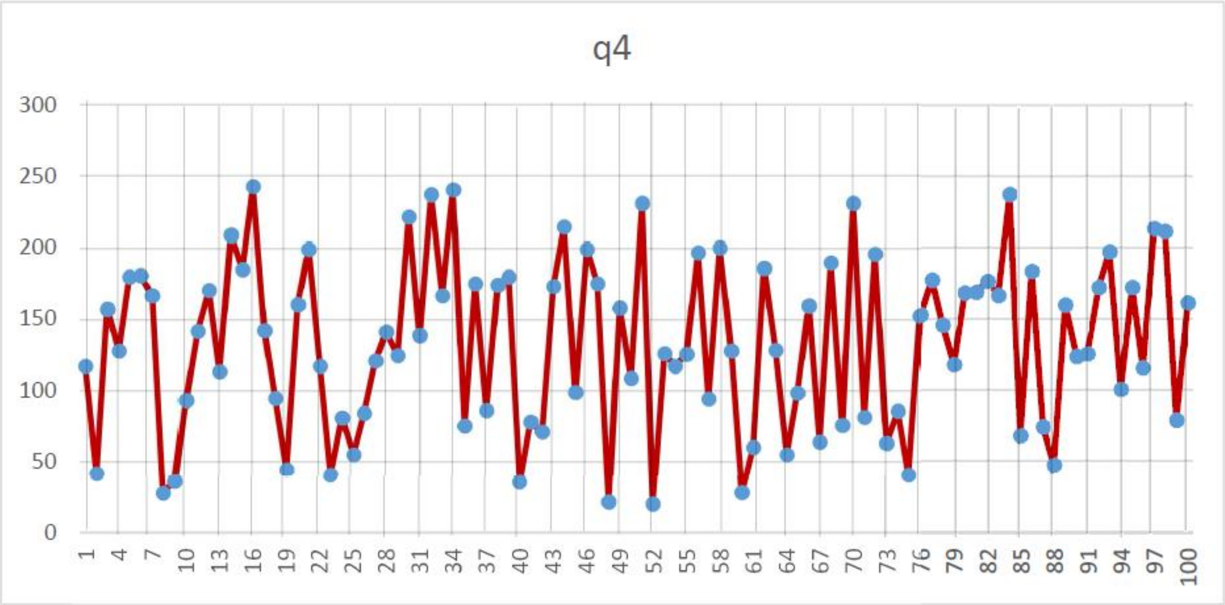


Fig. 5.6 Profile of the decision variable q4

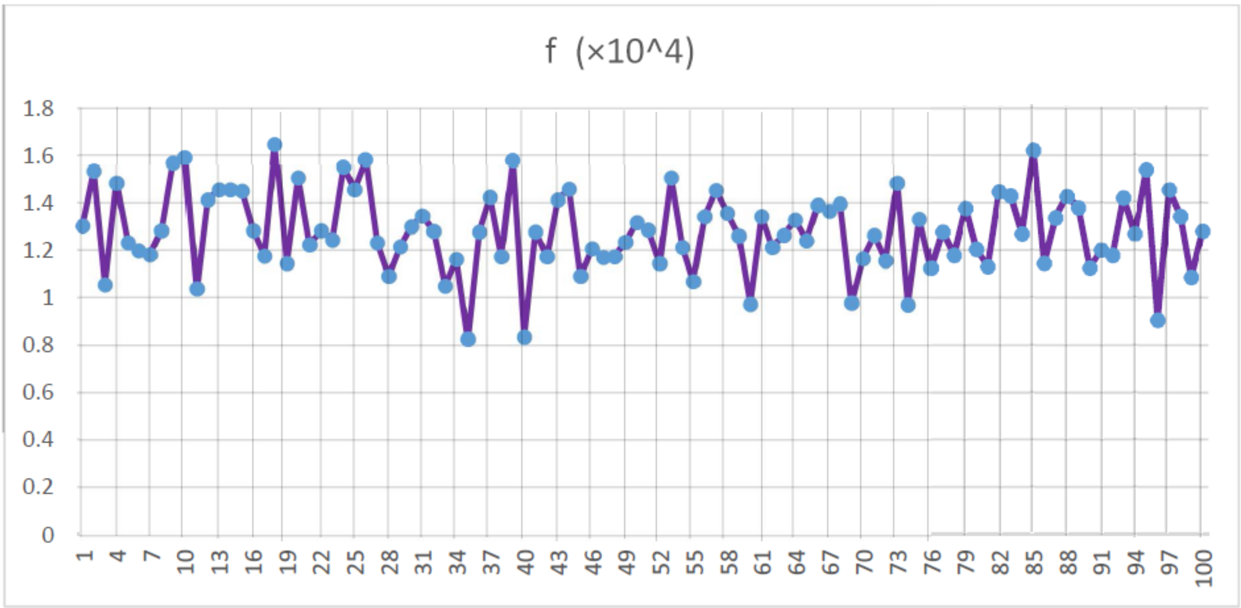


Fig. 5.7 Profile of the fitness (cost) function

Figure 5.7 depicts convergence characteristics of the differential evolution algorithm for 4-unit test system. The total electricity demand is 520MW and the total heat demand is 300MWth. Table 5.1

summarizes the optimal heat and power dispatches using differential evolution algorithm. The algorithm has lower cost when compared with total cost from artificial bee colony in chapter 4, and with no violation of the technical constraints. Also, the feasibility analysis of differential evolution algorithm certifies the solutions obtained by this technique. It is observed from iteration lines that in 100 runs, the obtained solutions are feasible with minimum operations cost. This is indicative of good diversity of the obtained solutions by this algorithm. The convergence characteristic of differential evolution algorithm for this system is depicted in Figure 5.7. As can be observed from the figure, differential evolution algorithm converged to the optimal solution in 100 iterations, a confirmation of the algorithm's capability in solving this class of combined heat and power economic dispatch problems.

Table 5.1 Comparison of the obtained results by Differential Evolution with other algorithms i.e. artificial bee colony, particle swarm optimization, direct solution and Genetic Algorithms.

	Differential evolution (DE)	Direct solution(Lambda method)	Genetic Algorithm(GA)	Particle Swarm Optimization (PSO)	Artificial bee colony (ABC)
P_1(MW)	33.3116	155.9	102.3	110.3	102.4
P_2(MW)	46.197	169.1	345.6	390.6	345.7
P_3(MW)	29.9572	195	70.1	69.1	70.1
Q_3(MWth)	25.0452	120	158.2	160	158.2
Q_4(MWth)	75.3626	180	234.7	230.7	234.8
Cost(₦)	8286.3	18933.8	21120.0	22917.0	21124.1

To investigate the performance of differential evolution over other algorithms applied in this research, we compare our results. The test system consists of two power-only units (units 1–2), one combined heat and power unit (unit 3) and a heat-only unit (unit 4). The cost coefficient parameters of this case along with the feasible region coordinates of combined heat and power unit is available in tables (3.1 –3.4). The optimal dispatches of the units in this case are provided in Table 5.1. The obtained results using differential evolution algorithm are compared with the results of direct solution algorithm (Lambda Method), Genetic algorithm (GA), Particle swarm optimization (PSO) and artificial bee colony algorithm (ABC). The convergence characteristics of differential evolution algorithm for this case are depicted in Figures (5.2-5.7). We observed that differential evolution algorithms converge quickly in 100 iteration lines. Hence, the number of maximum runs can be decreased to save the solution time, an evidence of good diversity of the obtained solutions by the algorithm in context. It was found that differential evolution algorithm outperformed all of the algorithms in the research in less time. Also, Genetic algorithm, particle swarm optimization and artificial bee colony solutions are not entirely feasible. For instance, for the direct solution method, the combined heat and power unit 3 violates its feasible region. In addition, the total power and heat mismatch were recalculated to obtain feasible dispatch values.

The lambda method results in chapter three shows that combined heat and power dispatch solution can be achieved in few calculation steps if the load levels are such that all units operate at the same incremental cost. A few additional steps are required to identify all violating units if the load levels are such that some of the units are to be set at their limits. We found initially that output of cogeneration unit 3 was an infeasible region while units 1, 2 and 4 decision variable outputs were optimally global. Furthermore, we investigated and found the new operating point of cogeneration unit 3 to be on the corner, that is, [195,120]. We treated cogeneration unit 3 to be fixed decision output variable to enable us recalculate system lambdas which gave us the final output of units 1, 2 and 4 respectively. As seen from table 3.5, direct solution method found the global minimum in the 4 test units. The obtained results for this method are found satisfactory for all the available constraints. The optimum λ-s obtained using the formula, gives the final dispatch if none of the units violates their limits. On the other hand, in case of violations, the affected unit(s) is/are identified and set at appropriate limits. The final dispatch is obtained by recalculating the λ-s considering only the non-violating units when all the violating units are identified and set at their limits. The effectiveness of direct solution algorithm has been demonstrated by considering four-test units in this research. Also, the formulation of the combined heat and power dispatch problem considered here conforms to the prevailing practice of using quadratic cost functions for the units. Conversely, the demerits of this algorithm compared to differential evolution are:

a. it requires extensive investigation of violated units;
b. lengthy recalculation is needed if any of the units violates optimization constraint problem and;
c. Finally, its suitability for large test system is not known.

The class of combined heat and power problem considered in this research can be solved with particle swarm optimization algorithm. Results in table 5.1 show that differential evolution method is superior to particle swarm optimization algorithm for this class of combined heat and power problem. Are views of the results provided in table 5.1 shows that particle swarm optimization requires a considerable number of iterations 103 lines as compared with the number of iterations required by differential evolution which converges to optimal solution in earlier iteration lines i.e. 100 as shown in figures (5.2-5.7).The slow convergence speed of particle swarm optimization makes computation cumbersome since extra arithmetic is required to correct this problem unlike differential evolution algorithm. Similarly, with respect to table 3.8, particle swarm optimization algorithm was optimally global for units 1, 3 and 4 but decision variable output of unit 2 was not feasible. Also, results of the two algorithms show that differential evolution algorithm outputs are global optimal as well as feasible, unlike particle swarm optimization that got trapped locally. Finally, the cost function value of particle swarm optimization was higher compared with differential evolution because output decision variables of this algorithm are neither optimally global nor feasible.

Comparing the performance of differential evolution algorithm with artificial bee colony algorithm may not be proper. The reason being that the solution provided by this algorithm does not even correspond to acceptable solutions. This does not imply that such algorithm cannot be developed for the combined heat and power problem considered in this research. Rather, the possibility of such algorithm outperforming differential evolution algorithm seems quite remote. As seen in table 4.1, artificial bee colony algorithm found the global minimum of the three of four-unit test system (1, 3 and 4) through 100 iteration lines. On unit 2, artificial bee colony algorithm could not find global minimum in that number of iteration lines; hence, decision variable output of this unit is infeasible. Although the two algorithms have the same number of iteration lines, artificial bee colony algorithms are not globally feasible in unit two, unlike differential evolution algorithm that is feasible in the entire units. This comparison shows that differential evolution algorithm reduces the system’s operations cost considerably as shown in table 5.1.The effectiveness of differential evolution algorithm is demonstrated through its comparison with artificial bee colony and other algorithms used in this research. It was found that the differential evolution algorithm can find better solutions in terms of the objective function value, convergence speed and the number of solutions with lower objective function, compared to artificial bee colony algorithm. Also, the better solution results, especially in large test systems, confirm the applicability of the differential evolution algorithm for dealing with real world applications.

Results in table 3.7 show that the proposed algorithm (GA) can provide combined heat and power dispatch solution in 51 iteration lines. As seen in table 3.7, genetic algorithm has found the global minimum in three of the four-test units (1, 3 and 4). On one problem, unit 2, the proposed algorithm could not find the optima in the specified maximum number of iteration lines. Convergence characteristics of the proposed algorithm for this system are depicted in Figure 3.6. As can be observed in this figure, the proposed algorithm is converged to produce results quite close to the global optima after 51 iteration lines. Despite the proposed algorithm being theoretically capable of providing global optimum starting from any random set of initial population, it tends to normally fail with providing automatic, all-constraints-satisfying solutions. Optimality proof of genetic algorithm is based on infinite stages of search. Infinite searches are not at all feasible in any context. Similarly, with respect to table 5.1, differential evolution algorithm found better solution in terms of objective function value, as well as alternative solutions than genetic algorithm. Also, results obtained using differential evolution algorithm when compared with genetic algorithm found to converge to a feasible solution with the lower total cost in a reasonable time.

The effectiveness of differential evolution algorithm verified using the test system, shows that the algorithm can find better solutions in terms of the objective function value, convergence speed and the number of solutions with lower objective functions compared with other algorithms (Genetic algorithm, direct solution method, particle swarm optimization and artificial bee colony algorithm). Some vital results from differential evolution algorithm uphold that:

1. The obtained results are also feasible— an indication that differential evolution algorithm has the capability of attaining doubly optimal and feasible solutions;

2. The algorithm converges in relatively small number of iteration lines— an implication that the algorithm has a good convergence speed which enables it for effective use in this class of optimization problem.

3. In this four-unit test system, the obtained value for the objective function is less with differential evolution algorithm compared with other algorithms employed in this research. It becomes important to note that our objective of the above observations is to clarify that our solution (though different from those of direct solution, genetic algorithm, particle swarm optimization and artificial bee colony algorithm) is the most suitable solution. Essentially, our aim is not to advance an argument invalidating other algorithms used in this research as rather unreliable for this class of optimization problems. The objective however, it to show that Genetic algorithm, particle swarm optimization, direct solution algorithm and artificial bee colony algorithm framework are indeed powerful models. Thus, there is no reason why they should fail to provide solution for this class of fairly straight forward optimization problems. Hence, it appears logical to infer that differential evolution algorithm is extremely promising in comparison with the known combined heat and power algorithms (Genetic algorithm, particle swarm optimization, Lambda or direct solution algorithm and artificial bee colony algorithm).

5.7 CONCLUSION

The combined heat and power economic dispatch problem has been solved using the differential evolution algorithm. This was implemented in MATLAB software, and reasonable results were obtained. Evidently, from the graphical profiles of the decision variables as well as the fitness or cost function, the differential evolution procedure is stochastic in nature. This is not surprising since the MATLAB-code contains special functions such as randperm, and rand which basically perform random permutations of their arguments. However, the algorithm is sophisticated enough to pick out the required minimum cost function for the combined heat and power economic dispatch problem presented. Also, the feasible operation region of cogeneration unit 3 was employed to correct the minimum or maximum heat/power limit violation. Differential evolution algorithm has been applied to solve combined heat and power economic dispatch problems for a four generators system of two conventional power units, one cogeneration unit and one heat only unit. Experimental results confirm the performance of the differential evolution over genetic algorithm, direct solution algorithm, artificial bee colony, particle swarm optimization etc. as a powerful technology for optimization problems. The obtained results are found to be adequate for all the available constraints. The solutions obtained in this case are displayed in table 5.2.

P_1(MW)	P_2(MW)	P_3(MW)	Q_3(MWth)	Q_4(MWth)	Cost(₦)
33.3116	46.197	29.9572	25.0452	75.3626	8286.3

CHAPTER 6- CONCLUSION AND RECOMMENDATIONS

6.0 CONCLUSION

A new perspective based on Genetic algorithm is proposed in this research work for coherent solution of combined heat and power economic dispatch problem. Different attributes and constraints such as heat and power demands, feasible operation region of combined heat and power units, capacity limits of units and other operational constraints are taken into consideration in the formulation of combined heat and power economic dispatch problem. The feasible operation region of a cogeneration unit is employed to remedy minimum or maximum heat/power limit violations. The efficacy of the Genetic algorithm was established using genetic algorithm codes. Genetic algorithm, it was realized, can proffer better solutions in terms of the objective function value, convergence speed and actual number solutions, compared with Particle swarm optimization, artificial bee colony, and direct solution algorithms. It however could not outperform results from differential evolution Algorithm.

Notably, direct solution algorithms have less objective functions but require length recalculations when any of the units violates the constraint(s) and its effectiveness for larger units unknown. Formulation of the combined heat and power dispatch problem considered here conforms with the prevailing practice of using quadratic cost functions for the units. The possibility of extending the genetic algorithm solution for cases where the cost functions are not quadratic but linear is currently being explored. Hence, it appears reasonable to conclude that in comparison with direct solution, artificial bee colony and particle swarm optimization algorithms, Genetic algorithm holds extreme promise. Finally, this research explained in details how genetic algorithm can be used to model combined heat and power economic dispatch problem encountered in simple cycle cogeneration systems. This work may equally be adapted to develop additional models or extend the current ones discussed. The development of genetic algorithm to optimize the cost of power generation is efficient, robust, likely has long term relevance. The implication here is that further development of easier power generation cost modeling is of paramount importance.

6.1 RESEARCH CONTRIBUTIONS

In addition to results obtained and the algorithms developed to determine the combined heat and power economic dispatch decision variables, this dissertation presents a more detailed approach of identifying and handling units that violate the constraints. It also sets the violating quantities at their appropriate limit, as earlier explained on conventional heat and power units as well as the cogeneration unit respectively. The study also recognizes that most related literatures did not dive into handling of constraints in cogeneration unit and how to solve associated problems during violations. This research can therefore be useful when solving problems on combined heat and power economic dispatch as a more accurate approach to determining the combined heat and power economic dispatch decision variables similar to what has been explored in the current study. Ultimately, the research explored alternative means to computing the combined heat and power economic dispatch decision variables directly by substituting the research data inside the developed model equations. Through this technique (direct solution algorithm), the researcher was able to compute dispatch values of heat and power directly. The results obtained have addressed the key questions of the research and the objective, both of which are key determiners of the combined heat and power economic dispatch decision variables. It has also minimized the cost function using genetic algorithm; however, there are improvements that can be done for a more optimum system.

6.2 RECOMMENDATION

The following recommendations have been suggested for the improvement of combined heat and power plants and the maximization of its benefits on the energy sector. For its capacity to eliminate rigorous calculations, slower convergence and infeasible result(s) involved in direct solution algorithm, artificial bee colony and particle swarm optimization algorithms respectively, genetic algorithm is a powerful optimization tool for finding solutions of combined heat and power economic dispatch decision variables. It gives the exact solution of combined heat and power problems which converge as fast as possible as exemplified in this research study. Although, particle swarm optimization, differential evolution, artificial bee colony and direct solution algorithms were also applied to solve the combined heat and power problem, the effectiveness of direct solution algorithm for larger units is not known yet. The reason is that the calculations involved are difficult to manipulate efficiently despite it having the least objective function value. Conversely, particle swarm optimization usage is limited to a few problems which make it rather impermissible to use in this kind of research since it falls into local optimum in high-dimensional space, besides also having a low convergence rate characteristic in the iterative process. Results obtained by artificial bee colony are often associated with large diversity and in most situations, convergence to either optimal or near optimal solution is rather difficult. Differential evolution algorithm found the global minimum in all the four test systems.

The research therefore recommends that not only should this proposed method (genetic algorithm) be included as in the curriculum for higher programs, but applied when solving combined heat and power economic dispatch problems. Doing this will assist research students in accomplishing desired result(s), eliminate rigorous calculation processes and obtain optimally converging solutions. The financial evaluation should be enhanced by including certain factors such as potential loss of heat during generation, reduced maintenance on the operating device and the deferred replacement of machine components. A computer program should be developed to convert the developed MATLAB programs into a user friendly input/output unit. Also the database for the combined heat and power economic dispatch variables employed in the program may be expanded.

Finally, Genetic algorithm can be hybridized with other random search techniques such as artificial Bee colony, artificial immune system algorithm, and bacterial foraging optimization techniques for optimal result on combined heat and power economic dispatch problems. Also, combined heat and power units can be integrated with other source of energy such as gas turbine unit, hydrothermal plant etc.

REFERENCES

[1]Abdolmohammadi, H.A and Kazemi, A., (2013);“A benders decomposition approach for a combined heat and power economic dispatch’’,Energy conversion and management,Vol.71, pp.21-31.

[2] Shahidehpour, S. M., Wang, C., (1989) “Effects of Ramp Rate Limits on Unit Commitment and Economic Load Dispatch” IEEE Transaction on Power Systems, Vol. 8, 1,pp 128.

[3]Lee F. N., Breipohi A. M., (1993); “Reserve Constrained Economic Dispatch with Prohibited Zones” IEEE Transactions on Power Systems, Vol. 8, No. 1, pp. 246.

[4]Kalyan S., Kadali L., Rajaji V., Moorthy J. V., (2007);Economic generation schedule on thermal power system considering emission using grey wolves optimization. Elsevier Energy Procedia,Vol. 117, pp. 509-518.

[5] Swarup K. S., Yamashiro S. (2002); “Unit commitment solution methodologies using genetic algorithm”, IEEE Trans. Power Syst., vol. 17, pp. 87–91.

[6]Coelho H., Mariani V., (2006); “Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect”, Power Systems, IEEE Transactions on, vol. 21, no. 2, pp. 989 – 996.

[7]Dos Santos Coelho L., de Andrade B. D. L., Mariani V. C., (2013); A chaotic firefly algorithm applied to reliability-redundancy optimization. Evolutionary Computation (CEC), 2011 IEEE Congress on, Vol. 18, IEEE, pp. 89–98.

[8] McQueen D. H. O., Hyland P. R., Watson S. J., (2004); Monte Carlo simulation of residential electricity demand for forecasting maximum demand on distribution networks. IEEE Transactions on Power Systems, vol. 19, no. 3, pp. 1685–1689.

[9] Sasson A.M., (2013). Decomposition techniques applied to the nonlinear programming load-flow method, IEEE Trans. Power Appar. Syst. 1 78–82.

[10]Azmy A. M., Mohamed M. R., Erlich I., (2005); “Decision tree-based approach for online management of fuel cells supplying residential loads,” in Proceedings the IEEE PowerTech Conference, St. Petersburg, Russia.

[11]Cardell J. B., (2007); “Distributed resource participation in local balancing energy markets,” in Proceedings of IEEE PowerTech, Lausanne, Switzerland.

[12]Lothar M. S., (2001); Theory of genetic algorithms. Theoretical Computer Science, Elsevier
vol. 259, Issues 1–2, 28, 1-61.

[13] Tan K. C., Lee T. H., Khor E. F., (2001) “Evolutionary algorithms with dynamic population size and local exploration for multi-objective-optimization, ”IEEE Transactions on Evolutionary Computation, vol. 5, no. 6, pp. 565–588.

[14]Deb K., Pratap A., Agarwal S., Meyarivan T., (2002); "A fast and elitist multi-objective genetic Algorithm:NSGA-II", IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182-197.

[15]Back T., (2004); “Selective pressure in evolutionary algorithms: A characterization of selection mechanisms,” in Evolutionary Computation, 1994.IEEE World Congress on Computation Intelligence, Proceedings of the First IEEE Conference on. IEEE,pp. 57–62.

[16] Munetomo M., Takai Y., Sato Y., (1998); “A migration scheme for the genetic adaptive routing algorithm,” in Systems, Man, and Cybernetics. 1998 IEEE International Conference on, vol. 3. IEEE, pp. 2774–2779.

[17]Alajlan M.,Koubaa A.,Chaari I.,Bennaceur H., Ammar A., (2013); “Global path planning for mobile robots in large-scale grid environments using genetic algorithms,” in Individual and Collective Behaviors in Robotics (ICBR), International Conference on. IEEE.

[18] Ratnaweera A., Halgamuge S., Watson H. C., (2004); Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,‖ Evolutionary Computation, IEEE Transactions on, vol. 8, no. 3, pp. 240–255.

[19] Zhang J.,Chung H., Lo W.L.,(2006); Pseudo code evolutionary genetic algorithms for power electronic circuits optimization,‖ Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 36, no. 4, pp. 590– 598.

[20]Kennedy R., Eberhart J., (1995); Particle swarm optimization,‖ Neural Networks. 1995 Proceedings., IEEE International Conference on Neural Networks, Perth, WA,Australia, vol. 4, pp. 1942– 1948.

[21]Chaymaa L., Said B., Ali E., (2018); Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. Elsevier Science Direct Procedia, Computer Science 127,pp. 180–189.

[22]Gou T., Henwood, M.I., Ooijen, M.V., (1996); “An algorithm for combined heat and power economic dispatch”, IEEE Trans.on power systems, vol 11,pp. 1778-1784.

[23] Rao P.S.N. (2006); ”Combined heat and power economic dispatch: a direct solution”, Electric power components and systems,vol.34,pp.1043-1056.

[24]Chapa, M.A.G, Galza,J.R.V., (2004); An economic dispatch algorithm for cogeneration system”,IEEE General meeting on power engineering, pp. 989-994.

[25] Sashirekha, A., Pasupuleti,J.,Moin,N.H., Tan, C.S., (2013); “Combined heat and power economic dispatch solved using Lagrangian relaxation with surrogate sub gradient multiplier updates”,Int.J. of Electrical power and energy systems, Vol.44,pp. 421-430.

[26] Tyagi, G. Pandit, M.,(2012); “combined heat and power dispatch using particle swarm optimization”, IEEE students’ conference on electrical, electronics and computer Science (SCEECS),pp. 1-4.

[27] Vasebi, A., Fesanghary, M., (2007); Bathaee, S.M.T.(2007),”Combined heat and power Economic dispatch by harmony search algorithm”, Int.J of Electrical power and energy system,Vol.29,pp. 713-719.

[28]Basu, M., (2011); “Combined heat and power economic dispatch by using differential evolution”, Electric power components and systems, Vol.38,pp. 996-1004.

[29]Basu, M., (2012); “Artificial immune system for combined heat and power economic dispatch”, Int.J.of electrical power and Energy system, Vol.43, 1-5.

[30] Sinha,N., Saikia, L.C. and Malakar, T., (2010); “Optimal solution for non-convex combined heat and power dispatch problems using differential evolution”, IEEE international conference on ICCIC, pp. 1-5.

[31] Song, Y.H., Chou, C.S. and Stonham, T.J, (1999); “Combined heat and power economic dispatch by improved at colony search algorithm”, Electric power system research, Vol.52,pp. 115-121.

[32]Chen,C. L., Lee, T.Y., Jan, R.M and Lu C.L., (2012); “A novel direct search approach for combined heat and power dispatch”, Int.J of electrical power and Energy system, Vol.43,pp.766-773.

[33] Wang, L., Singh, C., (2008).” Stochastic Combined heat and power dispatch based on multi-objective particle swarm optimization”, Int.J. of electrical power and energy systems, vol.30, pp.226-234.

[34] Sudhakaran, M., Slochanal, S.M. R., (2003); ”Integrating genetic algorithms and tabu search for combined heat and power economic dispatch”, IEEE conference on convergent Technologies for the Asia-Pacific Region, Vol. 1, pp. 67-71.

[35]Chang, C.S., Fu,W., (1998); “Stochastic multi-objective generation dispatch of combined heat and power systems”, IET proceedings-Generation,Transmission and Distribution, Vol.145, pp. 583-591.

[36]Kukkonen, S., (2006); “Constrained real-parameter optimization with generalized differential evolution”, IEEE congress on evolutionary computation, pp. 207-214.

[37]Clerc M., Kennedy J., (2002); The particle swarm-explosion, stability, and convergence in a Multi-dimensional complex space, Evolutionary Computation, IEEE Transactions on, vol. 6, no. 1, pp. 58–73.

[38] Mezura-Montes, E., Valazquez-Reyes, J., Coello Coello, C. A., (2006); “Modified differential evolution for constrained optimization”, IEEE Congress on Evolutionary computation, pp.25-32.

[39] Noman N., Iba, H., (2006);”A new generation alternation model for differential evolution”, Proceedings of the 8th Annual conference on Genetic and Evolutionary computation,pp.1265-1272.

[40]El Ela, A.A.A., Abido, M.A and Spea, S.R., (2010); “Differential Evolution algorithm for emission constrained economic power dispatch problem”, Electric power system Research,Vol.80, pp.1286-1292.

[41]Lampinen, J., Zelinka, I., (1999); “Mixed integer-discrete- continuous optimization by differential evolution”, Proceedings of the 5th International conference on soft computing, pp. 71-76.

[42]Babu, B.V., Jehan, M.M.L., (2003); " Differential evolution for multi-objective optimization"; Congress on evolutionary computation, Vol4,pp.2696-2703, 2003.

[43]Datta, D., Dutta, S., (2012); "A binary real coded differential evolution for unit commitment problem", Int.J. of Electrical power and Energy system, vol.42, pp 517-524.

[44] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Boston: Addison-Wesley.

[45] Davis, L., (1991); Handbook of Genetic Algorithm. Von Nostrand Reinhold, New York.

[46] Bryant K., (2000); Genetic Algorithms and the Traveling Salesman Problem, In Proceedings of 1st GNT Regional Conference on Mathematics, Statistics and Applications.

[47] Madureira, A., Ramos, C., Silva, S. C., (2002); A Coordination Mechanism for Real World Scheduling Problems using Genetic Algorithms. Evolutionary Computation, in Proceedings of the 2002 CEC, vol. 1, pp. 175 –180.

[48]Hosseini, S.S.S., Jafarnejad, A., Behrooz, A. H., Gandomi, A. H., (2011); "Combined heat and power economic dispatch by mesh adaptive direct search algorithm", Expert systems with Applications, Vol.38, pp.6556-6564.

[49] Omar, M., Baharum, A., Hasan, Y. A., (2006); A Job-Shop Scheduling Problem (JSSP) Using Genetic Algorithm in Proceedings of 2nd IMT-GT Regional Conference on Mathematics, Statistics.

[50]Fogel, D.B., Ghozeil, A., (1996); Using fitness distributions to design more efficient evolutionary computations. Proc. of 1996 IEEE Conf. on Evol. Comp.,Keynote Lecture,IEEE Press, NY. 11-19.

[51]Gehlhaar, D.K., Verkhivker, G.M., Rejto, P.A., Sherman, C.J., Fogel, D.B., Fogel, L.J., Freer, S.T., (1995); Molecular recognition of the inhibitor AG-1343 by HIV-1 protease: conformationally flexible docking by evolutionary programming.Chem. & Biol., 2**,**317-324.

[52]Fogel, D.B., (1995):Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, IEEE Press, NY.

[53]Bäck, T. (1996); Evolutionary Algorithms in Theory and Practice, Oxford, NY.

[54] Schwefel, H. P., (1995; Evolution and Optimum Seeking, John Wiley, NY.

[55] Michalewicz, Z. (2003): Genetic Algorithms, Data Structures and Evolution Programs, 3rd edition., Springer.

[56] Inazawa, H., Kitakaze, K. (2006); Locus-Shift Operator for Function Optimization in Genetic Algorithms. Complex Systems Publications, Inc.

[57] Mardle, S., Pascoe, S., Tamiz, M., (2000); An investigation of genetic algorithms for the optimization of multi-objective fisheries bio-economic models. International Transactions of Operations Research, 7(1), 33–49.

[58]Dehuri, S., (2008); Application of elitist multi-objective genetic algorithm for classification rule generation, Applied Soft Computing, pp. 477–487.

[59] Sivanandam, S. N., Deepa, S. N., (2008); Introduction to Genetic Algorithms. Springer.

[60]Birch, J. B., Wan, W. (2009); An Improved Genetic Algorithm Usinga Directional Search. Tech report presented at Virginia Polytechnic Institute and State University,Blacksburg.

[61] Nazif H., (2009); A Genetic Algorithm on single Machine Scheduling Problem to Minimize Total Weighted Completion Time. European Journal of Scientific Research, Vol.35 No.3, pp.444-452.

[62]Haibin Duan, Xiufen Yu, (2007); "Progresses and Challenges of Ant Colony Optimization-Based Evolvable Hardware", Evolvable and Adaptive Hardware 2007. WEAH 2007.IEEE Workshop on, pp. 67-71.

[63] Riffat R. M.,(1997); “Practical considerations in applying economic dispatch models to combined cycle cogeneration plants,” Proceedings of WESCANEX’97, Winnipeg, pp. 59–63.

[64]Binwu, D.,(1997); Optimization design on automobile gearbox. Journal of Fuzhou University, 25 (5), 59-61.

[65]Chao, W., Youxin, L., Zheming, H. (2007); Optimum design of automobile gearbox based on Lingo10.0 software. Machinery Design and Manufacture, 12, 75-77.

[66] Wei L.Y., Zhao, M. (2005). A niche hybrid genetic algorithm for global optimization of continuous multimodal function. Applied Mathematics and Computation, 160 (3), 649-661.

[67] Molga, M., Smutnicki, C. (2005); Test functions for optimization needs, in Proceedings of 4th Conference on Genetic Algorithms.

[68]Ali M.M., Torn A. (2002); Topographical differential evolution using pre-calculated differentials. In G. Dzemyda et al., editor, Stochastic Methods in Global Optimization, pages 1–17. Kluwer Academic.

[69] Storm, R., Price, K., (1997):”Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces”, Journal of Global optimization, vol 11,pp. 341-359.