UNIT COMMITMENT BY ARTIFICIAL INTELLIGENCE TECHNIQUES

BY

ABDEL-AAL HASSAN ISMAIL MANTAWY

A Dissertation Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

ELECTRICAL ENGINEERING

JUNE 1997

UMI Number: 9738990

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This dissertation, written by ***ABDEL-AAL HASSAN ISMAIL MANTAWY*** under the direction of his Dissertation Advisor and approved by his Dissertation Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of ***DOCTOR OF PHILOSOPHY in ELECTRICAL ENGINEERING***

Dissertation Committee

Dr. Y. L. Abdel-Magid (Chairman)

Dr. Shokri Z. Selim (Co-Chairman)

Dr. Ahmad S. Farag (Member)

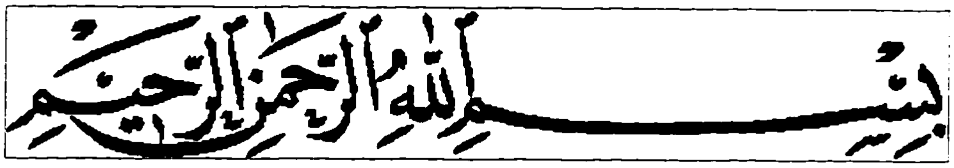
Dr. Ibrahim M. Al-Amin (Member)

Dr. Abdallah M. Al-Shehri (Member)

Dr. J. M. Bakhashwain (Member)

Department Chairman

Dean, College of Graduate Studies



In the Name of ALLAH, The Most Gracious, The Most Merciful

"Verily! in the creation of the heavens and the earth,

and in the alteration of night and day, there

are indeed signs for those who think"

The Holy Qur'an

Dedicated To

my father, my mother
my wife, and
my kids

ACKNOWLEDGMENT

First of all, praise be to **ALLAH** whose help made it possible to complete this work. Acknowledgment is due to King Fahd University of Petroleum and Minerals for the support of this research.

I wish to express my deepest thanks to my dissertation advisor, Dr.**Youssef L. Abdel-Magid.** His expert guidance, interest, criticism, enthusiasm, friendship, and above all the inordinate amount of time he spent, enabled me to achieve this work. I am deeply indebted to Dr. **Shokri Selim**, my dissertation co-advisor for his continued guidance, signs, help, friendship and encouragement for the duration of the research work.

I am grateful to my dissertation committee members Dr. **A. S. Farag**, Dr. **I. M. Al-Amin**, Dr. **A. M. Al-Shehri,** Dr. **J. M. Bakhashwain**, for their valuable suggestions and interest about my research.

I am particularly thankful to the EE Dept. chairman, Dr. **Samir A. Al-Baiyat** and the Dean of the college of graduate studies, Dr. **A. M. Al-Shehri** for their valuable support and numerous advice throughout the course of graduate studies.

I am also thankful to KFUPM faculty members who taught me the Ph.D. courses: Dr. **A. Farag**, Dr. **I. Al-Amin**, Dr. **A. Al-Shehri,** Dr **M. Bettayeb**, and Dr. **Abuelma'atti,** EE Dept., Dr. **Shokri Z. Selim**, SE Dept., Dr. **Rafeek Braham,** COE Dept., and Dr. **G. Opouko**, Research Institiute.

Special thanks go to my colleagues at KFUPM, for creating a pleasant atmosphere in which to work.

I am also thankful to Mr. **Ahmed Daud**, from KFUPM English Language Center, for revision of the manuscript.

At last, but not least, I would like to thank my wife, **Mervat,** and my kids**, Sherouk, Omar** and **Kareem**, for their love, patience and understanding that kept me going through this program and my career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Name: **Abdel-Aal Hassan Ismail Mantawy**

Title: **Unit Commitment By Artificial Intelligence Techniques**

Major Field: **Electrical Engineering**

Date of Degree: **June 1997**

The present work deals with thermal generation scheduling, which could be considered the major part of the overall scheduling problem of hydrothermal power systems. The scheduling problem of thermal generating units can be considered as two linked optimization problems. It comprises the solution of both the Unit Commitment Problem(UCP) and the Economic Dispatch Problem(EDP). The former is a combinatorial optimization problem with very hard constraints, while the later is a nonlinear programming problem.

The growing interest in the application of Artificial Intelligence(AI) techniques to power system engineering has introduced the potentials of using this state-of-the-art technology in the thermal generation scheduling of electric power systems. AI techniques, unlike strict mathematical methods, have the apparent ability to adapt to nonlinearities and discontinuities commonly found in power systems. The best known algorithms in this class include evolution programming, genetic algorithms, simulated annealing, tabu search, and neural networks.

In the present work, seven different AI-based algorithms have been developed to solve the UCP. Two of these algorithms namely, simulated annealing and genetic algorithms, are implemented in a novel way. The other five proposed algorithms are applied for the first time to solve the UCP. These algorithms are a Simple Tabu Search Algorithm (STSA), an Advanced Tabu Search Algorithm (ATSA), a hybrid of Simulated annealing and Tabu search algorithms(ST), a hybrid of Genetic and Tabu search algorithms (GT), and a hybrid of Genetic, Simulated annealing, and Tabu search algorithms (GST).

As a first step to solve the UCP, some modifications to the existing problem formulation have been made to render the formulation more generalized. An augmented model including all the problem constraints is presented.

A major step in the course of solving the UCP, is the solution of the EDP. In this regard, an efficient and fast nonlinear programming routine is implemented and tested. The implemented routine is based on a linear complementary algorithm for solving the quadratic programming problems as a linear program in a tableau form. Comparing the results of our proposed routine, it is found that the results obtained are more accurate than that obtained using an IMSL quadratic programming routine. The application of this routine to the EDP is original.

The corner stone in solving the combinatorial optimization problems is to come up with good rules for finding randomly feasible trial solutions from an existing feasible solution, in an efficient way. Because of the constraints in the UCP this is not a simple matter. The most difficult constraints to satisfy are the minimum up/down times. A major contribution of this work is the implementation of new rules to get randomly feasible solutions faster.

All the proposed algorithms have been tested on several practical systems reported in the literature, with different complexities. The numerical results obtained by the proposed algorithms are superior to the results reported in the literature

DOCTOR OF PHILOSOPHY

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA
JUNE 1997

خلاصة الرسالة

الاسم: عبدالعال حسن اسماعيل منطاوى

عنوان الرسالة: الجدولة المثالية لوحدات التوليد في نظم القوى الكهربية باستخدام طرق الذكاء الصناعي

التخصص: الهندسة الكهربية

تاريخ الشهادة: يونيو ١٩٩٧

تهدف هذه الأطروحة إلي دراسة مشكلة الجدولة المثالية علي المدى القصير لوحدات التوليد الحراري و هي تعتبر الجزء الأعظم في جدولة نظم التوليد في نظم القوي الكهربية.

ويمكن إعتبار مشكلة الجدولة المثالية لوحدات التوليد في نظم القوي الكهربية مكونة من إيجاد الحل الأمثل لمشكلتين مرتبطتين: الأولي هي جدولة وحدات التوليد وهى مسألة كثيرة التباديل والثانية إيجاد الحل الأمثل لمسألة توزيع الأحمال بين وحدات التوليد بطريقة إقتصادية و هي مسألة غير خطية.

ومما دفعنا لتطبيق طرق الذكاء الصناعي في حل مشكلة الجدولة المثالية لوحدات التوليد في نظم القوي الكهربية هوالاهتمام المتزايد بتطبيقات هذة الطرق في حل قضايا نظم القوي الكهربية. والطرق التى تم استخدامها هى: تمثيل التخمير(simulated annealing) ، البحث المقيد (tabu search) و الخوارزميات الجينية (genetic algorithms) بالإضافة إلي الخوارزميات المهجنة(hybrid algorithms).

في هذه الأطروحة تم عمل سبعة برامج مختلفة للكمبيوتر لحل هذه المسألة. إثنين من هذه البرامج يعتبر تنفيذ جديد لطرق تم استخدامها من قبل في حل المسألة و هاتين الطريقتين هما: تمثيل التخمير والخوارزميات الجينية. أما الخمسة برامج الأخرى فيتم تطبيقها لحل هذه المسألة للمرة الأولي والطرق هي: البحث المقيد السهل والبحث المقيد المتطور وثلاث خوارزميات مهجنة كالآتي: هجين بين تمثيل التخمير والبحث المقيد ، هجين بين البحث المقيد و الخوارزميات الجينية و هجين بين الطرق الثلاث تمثيل التخمير، البحث المقيد و الخوارزميات الجينية.

ولتحسين أداء هذه البرامج فقد تم إقتراح قواعد جديدة لإيجاد حلول للمسألة بطريقة عشوائية ذات كفاءة عالية وفي نفس الوقت تحقق كل القيود المفروضة في المسألة. وفي سبيل تحقيق نتائج أفضل فقد تم عمل برنامج جديد لإيجاد الحل الأمثل لمسألة توزيع الأحمال بين وحدات التوليد بطريقة إقتصادية.

وقد تم تطبيق جميع البرامج على أربع أنظمة مختلفة في الصعوبة ثلاثة منها مأخوذة من نشرات علمية معروفة و محلوله بطرق مختلفة والنظام الرابع مأخوذ من الشركة السعودية الموحدة للكهرباء في المنطقة الشرقية وقد جاءت النتائج التي حصلنا عليها في هذه الأطروحة أفضل من الحلول المنشورة لهذه الأنظمة في النشرات العلمية.

درجة الدكتوراه في الفلسفة

جامعة الملك فهد للبترول و المعادن
الظهران ، المملكة العربية السعودية
يونيو ١٩٩٧

CHAPTER ONE

INTRODUCTION

The efficient and optimum operation and planning of electric power generation systems have always occupied an important position in electric power industry. The economic operation problem in electric power systems involves the scheduling of both thermal and hydro generating units to minimize the cost of supplying the power requirements of the system over a certain period under specified system constraints. The optimal operation of the thermal generating units involves the minimization of fuel expenditure. Estimates have shown that a 1% reduction in production costs can result in about $1 million annual savings for each 1000 MW of installed capacity.

The present study deals only with thermal generation scheduling which could be considered as the major part of the scheduling problem of hydrothermal power systems. The scheduling problem of thermal generating units comprises the solution of both the unit commitment and economic dispatch problems.

The Unit Commitment Problem (UCP) is the problem of selecting the generating units to be in service during a scheduling period and for how long. The committed units must meet the system load and reserve requirements at minimum operating cost, subject to a variety of constraints.

The Economic Dispatch Problem (EDP) is the optimal allocation of the load demand among the running units while satisfying the power balance equations and the units operating limits [1].

The solution of the UCP is really a complex optimization problem. It comprises the solution of the EDP. The UCP can be considered as two linked optimization problems. The first is a combinatorial problem and the second is a nonlinear programming problem. The exact solution of the UCP can be obtained by a complete enumeration of all feasible combinations of generating units, which could be a massive number. Then, the economic dispatch problem is solved for each feasible combination. Basically, the high dimension of the possible solution space is the real difficulty in solving the problem.

The solution methods being used to solve the UCP can be divided into four categories [1-3,8-93]:

- Classical optimization methods such as: Dynamic Programming, Integer and Mixed Integer Programming, Lagrangian Relaxation, Linear Programming, Network Flow Programming, and Probabilistic Methods [9-56].
- Heuristic methods such as Priority List and Expert Systems [57-64].
- Artificial Intelligence methods such as: Neural Networks, Simulated Annealing, Genetic Algorithms, and Tabu Search [65-74].
- Hybrid Algorithms: hybridization of two or more of the previously mentioned methods [76-93].

In the following, a survey of the classical optimization methods which have been reported in the literature is presented.

1.1 CLASSICAL OPTIMIZATION METHODS

Classical optimization methods are well documented in the literature [9-56] as a direct means for solving this problem. Some of these methods give good results, like Lagrangian relaxation, while others face the problem of dimensionality, particularly in the case of large-scale systems, as in Dynamic Programming and Mixed-Integer Programming.

1.1.1 DYNAMIC PROGRAMMING

Dynamic Programming (DP) was originally developed in 1950 by Richard Bellman. Since this date, it has been recognized as an extremely powerful approach for solving optimization problems. DP methods [9-21] decomposes the UCP in time. Starting at the first hour of the scheduling horizon, commitment of units progresses one hour at a time, and combinations of units are stored for each hour. This is the forward path of the DP method. At the end of the execution of the forward path, for each hour-state pair (a state is defined as a combination of the ON/OFF status of all units) the following information is stored:

(a) The minimum total production cost to reach the state starting from the first hour.

(b) An optimal link-back pointer pointing at that state of the previous hour which resulted in the optimum (minimum total cost) transition to the current hour state, and

(c) An array whose elements represent the continuous up or down times of all units. This information is retained in order to be able to observe time dependent constraints such as unit minimum up and down time, time dependent start-up costs, etc.

Finally the most economical schedule is obtained by backtracking from the state with the least total cost in the final hour through the optimal link-back path to the state of the initial hour.

Numerous papers have been published on solving the UCP by DP [9-21]. In the following, a short description for some of these papers is presented.

In 1966, P. G. Lowery published a paper [9], to solve the UCP by DP. The real concern of this study was to demonstrate the applicability of the DP method to solve the UCP.

In 1971, J. D. Guy [10] presented a new method for solving the UCP. The security constraints were included and the DP was the basic method for the solution.

In 1971, A. K. Ayoub et al. [11] presented a method of scheduling thermal generating units to achieve minimum operating costs, including both running and start-up costs. Spinning reserve was also considered based on a probabilistic approach as a security function. The DP was used in unit commitment and security function calculations.

C. K. Pang et al. [12], 1976, presented a paper in which a truncated DP method for the UCP is described. The method used the priority list of the available units so as to limit the original DP search for economical and satisfactory schedules. As a result the method does not guarantee the optimal solution. The constraints taken into consideration in [12] were minimum up/down, crew, transmission losses, and spinning reserve. The start-up, shut-down as well as the running costs are also included.

In 1981, C. K. Pang et al. [13] presented a study to compare three different DP algorithms for the UCP. The algorithms are DP-Sequential Combinations, DP-Truncated Combinations, and DP-Sequential/Truncated Combinations. The algorithms were based on performing the different combinations of the units that are previously ordered according to a priority list, hence some computations saving were achieved. However, the accuracy of the solution is degraded because of the limited search space and building the search on the priority list results which is not globally optimal.

In 1985, P. P. van de Bosch et al. [14] solved the UCP using a decomposition approach that is based on the DP method. Decomposition technique is used to reduce the computer resources required by the DP, by dividing the problem into smaller subproblems, which is much easier to solve. The subproblems are then solved by the DP based on the successive approximation technique. Although the solution procedure stops in a finite number of steps, there is no guarantee that the optimum solution of the problem will be reached.

In 1985, G. L. Kusic [16] presented an approach to the problem of economic dispatch and unit commitment using the DP algorithm. The proposed method considered all valid combinations of the previous hour with the accumulated ON and OFF times for each unit in each combination of the previous hour. The method counts the production cost for each combination of the previous hour with accumulated cost, up to each combination of the previous hour. The start-up costs incurred in the transition from each previous hour combination to each present hour combination are also considered.

In 1987, Walter L. Snyder et al. [17] introduced a practical approach for solving the UCP of a real power system using the DP algorithm along with some heuristic rules to save computation time. This approach features the classification of the generating units into related groups so as to minimize the number of the unit combinations that must be tested without precluding the optimal path. To achieve the execution time saving, individual units were assigned status restrictions in any given hour, such as unavailable, fix loaded, must run or derated capacity. The reserve, ramping, and minimum up/down time constraints were also taken into consideration.

In 1988, Walter J. Hobbs et al. [18] presented an enhanced DP approach for the UCP. The proposed approach used a sequential priority list method for forming the unit combinations to be evaluated in each hour of solution. From a given priority ranking of all units, the approach forms the working list for each hour which excludes all unavailable, must run, fixed load, and peaking units. Unavailable units are not considered during the formulation of the combinations. Must-run and fixed load units are always committed for the hour, and peaking units are independently committed (based on price) during economic dispatch. Subsequent combinations are formed by decommitting one unit at a time.

In 1991, Chung-Ching Su Yuan-Yih Hsu et al. [19] presented an approach for solving the security constrained multi-area UCP. The method takes power system dynamic stability limit into consideration. In the proposed method, DP was first employed to perform unit commitment on the whole system. The eigenvalues for the resultant hourly generation schedules were examined to see if they satisfied the prespecified dynamic security criterion. If the dynamic security requirements are not met at certain hours, an iterative algorithm is employed to reduce the inter-area line flows gradually and to perform area dispatch and, if necessary, area unit commitment, in order for the resultant generation schedule to satisfy the dynamic security requirements. As a result, a generation schedule is obtained which satisfies the dynamic stability requirements at the prices of a higher operating cost.

Based on the previous discussion of the various DP approaches for solving the UCP, it can be concluded that the main problem of the DP methods is the curse of the dimensionality: Storing all possible unit combinations (2^N-1, N: number of units) at every hour is impossible even for moderate size systems. Thus, heuristic techniques are used to restrict the number of combinations to be searched and the number of strategies to be saved at every hour [13,17]. These heuristic techniques produce suboptimal solutions and in certain cases may require the relaxation of some constraints in order to produce a solution.

The application of the DP method to the UCP has, however, another major difficulty in treating time-dependent constraints such as unit minimum up/down times, time dependent start-up costs, start-up ramps, etc. This difficulty has been well recognized in the literature [13,18] and has lead to suboptimal solutions or failure to provide a solution even in the case of complete state enumeration. The reason for this difficulty is that the definition of the "state" in the DP solution of the UCP as the combination of 0-1 status of all units is incorrect. This incorrect choice of the "state" leads to the requirement to keep some information on the continuous up or down times of all units. However, this information is incomplete since it is stored for the optimal transition path only and not for every path. Therefore, valuable information for the determination of the optimal (or, in some cases, even a feasible) solution is lost.

To overcome this problem, the state of the DP should be defined as $(\tau_1, \tau_2, \ldots\ldots, \tau_n)$, where τ_i is the continuous up time (if positive) or down time (if negative) of unit i. However, with this definition of the state, the complexity of the DP solution becomes prohibitive even for very small size systems.

1.1.2 LAGRANGIAN RELAXATION

The Lagrangian Relaxation (LR) methodology [22-37] uses Lagrange multipliers for the system constraints (power balance and reserve) and adds the associated penalty terms in the objective function to form the Lagrangian function. The problem then is decomposed into N subproblems one for each unit.

Based on the duality theory, the LR method subsequently tries to find those values of the Lagrange multipliers that maximize the dual objective function. This is a very hard problem to solve. Even if the solution to the dual problem was found, due to the non-convexity of the primal (original problem) objective function, feasibility of the primal problem is not guaranteed and the optimal values of the primal and dual problem objective functions would not be equal (called duality gap). The efforts of the LR method are thus focused on finding some values for the Lagrange multipliers that satisfy the systems (coupling) constraints and meanwhile reduce as much as possible the duality gap. The later is known by duality theory to be a lower bound to the optimal value of the objective function.

In 1973, Marshall L. Fisher [22] proposed the applicability of the Lagrangian method for solving the scheduling problems. The paper presented an algorithm for solving resource-constrained network scheduling problems, a general class of problems that includes the classical job-shop scheduling problem. It used Lagrange multipliers to dualize the resource constraints, forming a Lagrangian problem in which the network constraints appeared explicitly, while the resource constraints appeared only in the Lagrangian function. The algorithm was applied to examples of the job-shop scheduling problems. The results provide indications of the potential for the Lagrange multiplier algorithm.

In 1977, John A. Muckstadt et al. [23], suggested the application of the LR method for solving the UCP. The fundamental idea behind the LR was the incorporation of selected inequality constraints into the objective function by the use of Lagrange multipliers. A mixed integer programming model to the UCP was presented. A branch and bound algorithm was proposed using a Lagrangian method to decompose the problem into single generator subproblems. Each of these subproblems was solved by a simple DP recursion. A subgradient method was used to select multipliers that maximize the lower bound produced by the relaxation. The minimum up/down constraints were taken into account. The crew constraints were not considered as they do not apply to individual units. The incorporation of these constraints makes the problem difficult to decompose.

In 1983, D. P. Bertskas et al. [24] presented an algorithm based on the LR and nondifferentiable optimization. The approach was based on a duality transformation of the original problem and optimal solution of the associated nondifferentiable dual problem. The algorithm has two advantages over that presented in [23]. First, computational requirements typically grow only linearly with the number of units. Second, the duality gap decreases in relative terms as the number of units increases, and as a result the algorithm tends to actually perform better for solution of large size problems.

In 1983, A. Merlin et al. [25] proposed a new implementation for solving the UCP by LR. The contribution of this work was the detailed description of the duality gap difficulty and the implementation of a new algorithm for updating the Lagrange multipliers. The paper also presented the UCP formulation in a more visualized way. The algorithm was adopted to include: pumping units, and probabilistic determination of the spinning reserve.

R. Nieva et al. [26],1987, presented an algorithm, based on the LR method, to solve very large and complex UCP in a relatively short CPU time. The approach used the DP in the successive approximations. The proposed approach gives an estimate of suboptimality that indicates how close the solution is to the optimum. In contrast to the technique of the previous LR implementations, this approach makes no attempt of maximizing the dual function. Hence, the suboptimality estimates are rather conservative, but sufficiently precise as to render the search-range reduction technique effective.

K. Aoki et al. [27,28], in 1987, presented an efficient approach for solving a more practical UCP. The algorithm includes three types of units; usual thermal units, fuel constrained units, and pumped-storage hydro units. For maximizing the Lagrangian function, a 'variable metric method' is tried instead of the subgradient method. Moreover, a new device for efficiently obtaining a feasible solution is proposed.

In 1988, F. Zhung et al. [28] proposed a LR algorithm for a large scale UCP. A system of 100 units has been solved efficiently in a low execution time. The algorithm is divided into three phases. In the first phase, the Lagrangian dual of the unit commitment is maximized with standard subgradient techniques. The second phase finds a reserve-feasible dual solution, followed in phase three by solving the Lagrangian dual of the economic dispatch problem using a nonlinear programming routine.

J. F. Bard [29], 1988, presented an expanded formulation of the UCP in which hundreds of thermal units are scheduled on an hourly basis. The model incorporates all the units and system constraints including ramping and operational status and takes the form of a mixed integer nonlinear control problem. LR is used to disaggregate the model into separate subproblems in the generating unit, which are then solved with a nested dynamic program. The strength of the methodology lies partially in its ability to construct good feasible solutions from information provided by solving the dual problem. Thus, the need for branch and bound is eliminated. Computational experience of the algorithm showed that problems containing up to 100 units and 48 time periods could be solved in a reasonable time with duality gaps less than 1%.

K. Aoki et al. [30] proposed a method for solving a long-term unit commitment problem in a large scale power system. Three types of units were considered: usual thermal units, fuel constrained thermal and pumped storage hydro. The problem was formulated as a nondifferentiable and nonconvex mixed-integer programming. In this method, the dual problem is solved while some of the constraints are relaxed. A feasible solution of the primal problem is then found using the least square method.

In 1989, Sudhir Virmani et al. [31], presented a paper in which they provide an understanding of the practical aspects of the LR methodology for solving the UCP. The implementation details of applying the LR method to a realistic problem are introduced.

In 1991, S. Ruzic et al. [32] presented an algorithm for solving an extended UCP using the LR. The algorithm includes the transmission capacity limits, regulation reserve requirements of prespecified group of units, transmission losses and fuel constraints, in addition to the standard unit commitment constraints. The algorithm is tested for 100 units for a time horizon of 48 hours with promising performance.

C. Wang et al. [33], 1995, proposed an optimal generation scheduling approach with ramping costs. The unit ramping process is related to the cost of fatigue effect in the generation scheduling of thermal systems. The system operating cost includes the fuel cost for the units, start-up and shut-down costs as well as the rotor depreciation during ramping processes. The LR is proposed for unit commitment and economic dispatch, in which the original problem is decomposed into several subproblems corresponding to the optimization process of individual units. The network model is employed to represent the dynamic process of searching for the optimal commitment and generation schedules of a unit over the entire study time span.

In 1995, Ross Baldick [34] presented a generalized formulation of the UCP. The model includes: minimum up/down times constraints, power flow constraints, line flow limits, voltage limits, reserve constraints, ramp limits, and total fuel and energy limits on hydro and thermal units. The algorithm is tested to solve a system containing 10 units for a time horizon of 24 hours.

William L. Peterson et al. [35], 1995, presented a LR algorithm which uses the maximum capacity constraint method. The algorithm is extended to incorporate unit minimum capacity constraints and unit ramp rate constraints. The algorithm incorporates other practical features such as boiler fire-up characteristics and non-linear ramp up sequences.

S. J. Wang et al. [36], 1995, proposed a new approach based on augmented LR for a short term generation scheduling problem with transmission and environmental constraints. In this method, the system constraints are relaxed by using Lagrangian multipliers, and quadratic penalty terms associated with system load demand balance are added to the Lagrangian objective function. Then the decomposition and coordination technique is used, and non-separable quadratic penalty terms are replaced by linearization around the solution obtained from the previous iteration. The exact convex quadratic terms of decision variables are added to the objective function as strongly convex, differentiable and separable auxiliary functions in order to improve the convergence property.

John J. Shaw [37], 1995, proposed a direct method for security constrained unit commitment. It accounts for the security constraints in both the dual optimization and the primal solution stages (direct). This is contrasting with other methods that omit the security constraints from the optimization stage, and consider them only to construct a feasible unit commitment schedule (indirect). The algorithm develops a feasible solution in two stages. The algorithm first solves a dual programming problem in the dual optimization stage. The solution locates scheduling options that are likely to lead to an optimal feasible solution. The algorithm then selects a number of these options and evaluates the cost of each in the primal solution stage. The algorithm assesses the hour-by-hour cost of each option by solving a series of security constrained economic dispatch problems. Upon completion, the algorithm selects the option providing the lowest cost encountered.

Unlike DP methods, LR methods do not have problems in meeting time dependent constraints. The method is based on decomposing the problem into simple subproblems of the individual generating units. Since the optimization of the operation of each unit during the time horizon is performed separately, there are no dimensionality problems involved. DP or shortest path algorithms are then used to solve the subproblems of the individual units. LR methods, however, have problems in modeling plant crew constraints since they introduce coupling to the units subproblems. Also, due to the duality gap, discussed before, there is no guarantee of the optimality of the solutions produced by LR methods.

1.1.3 INTEGER AND MIXED-INTEGER PROGRAMMING

The solution using Integer Programming (IP) and Mixed Integer Programming (MIP) is based on the Benders approach of partitioning the problem into a nonlinear economic dispatch problem and a pure-integer nonlinear UCP. The MIP approach [38-45] solves the UCP by using the Branch and Bound method to reduce the solution search space systematically through discarding the infeasible subsets. Dual programming is also suggested for the solution of the thermal UCP. The general solution concept is based on solving a linear program and checking for an integer solution. If the solution is not integer, an ordered sequence of continuous linear problems or subproblems are continuously solved. These problems differ from each other only in that a different number and type of integer variables are held at fixed integer values. Any such Linear program solution, if not integer, constitutes the lower bound on integer solutions in that subset. The strategy of determining which variables to hold constant is called branching. MIP methods suffer from the problem of taking large memory and computation time.

In 1978, T. S. Dillon et al. [40] proposed an extended and modified version of applying branch and bound technique for IP as applied to the UCP. The main disadvantages of previous implementations of the approach to the UCP are the very long computation time required and the inadequate treatment of the problem of availability of units. The main features of the method include its computational practicability for realistic systems and proper representation of reserves relating to power interchange.

Andre Turgeon [41], 1978, formulated the UCP as a mixed-integer nonlinear programming problem. The solution method is based on the Bender's approach in which the problem is partitioned into a nonlinear and a pure-integer nonlinear programming problem. The first problem, which represents the economic dispatch problem is not solved in this algorithm. The second problem, the UCP, is solved by a variational method and a branch and bound algorithm. The method is considered practical for two reasons. The first reason is that all the UCP constraints are taken into account. The second reason is that the method was tested on a network of ten generating units, and the optimal solution was obtained in a relatively small CPU time.

In 1982, G. S. Lauer [43] proposed an algorithm for solving large scale UCP. The solution methodology is based on the branch and bound technique. Lower bounds are obtained by solving an associated nondifferentiable dual optimization problem using a subgradient method. In this regard, a different nondifferentiable optimization approach has been used. This approach yields information which turns out to be very valuable in generating a near optimal feasible solution and an associated upper bound to the optimal value. This is done by producing dynamic priority lists which are used to generate a feasible solution. As a result, it is necessary to examine only a few nodes of the branch and bound tree in order to obtain agreement of the generated upper and lower bounds to within a practically acceptable tolerance.

In 1983, A. I. Cohen et al. [44] presented a new approach for solving the UCP based on branch and bound techniques [44]. The method incorporates time dependent start-up cost, demand and reserve constraints and minimum up/down constraints. It does not require a priority ordering of the units. The method can be extended to allow for a probabilistic reserve constraints.

In 1990, E. Handschin et al. [45] described a method for solving the UCP considering energy constraints obtained from a long term optimization. The method solves the problem in two steps. The first step concerns a long term constrained energy optimization to calculate optimal daily energies for each unit. In the second step the energy constrained unit commitment is solved considering the short-term constraints and the optimal daily energies from the first step. The first step is formulated as a MIP problem using branch and bound algorithm. The second step is solved by the LR. The feasibility of the presented method is demonstrated by solving a power system of 22 units.

1.1.4 LINEAR PROGRAMMING

Linear Programming (LP) is a tool that has yet to reach its full potential in power system engineering [46]. IP is considered as an extension to LP. LP is used through the

LR method and Benders decomposition technique in solving the scheduling problem. Prior to the 1960's, the application of LP at first was limited to solving the economic dispatch problem. In the early 1960's, analytical models were implemented for unit commitment in order to replace the heuristic priority list method then in use. A mixed integer model was used through the branch and bound algorithm, while the economic dispatch problem was incorporated as a subroutine.

In 1985, M. Piekutowski et al. [47] described a method to solve the UCP using a linear programming method. Linear programming techniques were used to dispatch units at each time interval to minimize production cost over a load peak, or daily or weekly load cycle. Generator configurations, flow constraints, functional reserve and time-limited reserve were modeled, with the capability for individual restrictions on any unit. The major feature of the approach was a specialized linear programming algorithm for solution of the economic dispatch. The linear programming was designed to accomplish shifts between different proposed schedules while retaining optimality of successive dispatch solutions. The linear programming was based on the revised simplex method because of the ease of programming and the ability to include modifications readily in this developmental application.

1.1.5 NETWORK FLOW PROGRAMMING

Network flow programming models are described by parameters such as arc cost, arc capacity, a lower bound and arc gain. These parameters are used to solve the network programming problem. This approach models the UCP as a network's shortest path problem and solves the problem using a network optimization algorithm [48,49]. The method is simple, efficient and fast because of exploiting network structures.

In 1988, H. Brannlund et al. [48] applied the nonlinear network flow programming technique to the scheduling of hydro plants with security constraints. In this work a nonlinear model for the hydro plant is used. The security constraints are included as a set of additional nonlinear network type constraints. The resulting large scale mathematical programming problem is solved using a special reduced gradient algorithm.

In 1992, R. Zhu et al. [49] proposed a new approach for solving the thermal UCP using the network programming technique. In the proposed formulation network, nodes represent unit combinations and are connected by arcs. The arc cost includes start-up cost and unit production cost. Two nodes are designed as the source and the sink. When the shortest path connecting the source to the sink is found, the optimal unit commitment schedule observed with various operating constraints can be easily obtained. To reduce the execution time and required memory storage for nodes, a truncated window and priority list are used. The EDP is solved using the Lagrangian multipliers method. The algorithm is applied to solve an example of 61 units in a relatively small time.

1.1.6 PROBABILISTIC METHODS

The assessment of a spinning reserve is an integral part of the UCP [50-56]. Most of the methods previously explained ignore the probabilistic aspects of system components. There has been relatively little published material on spinning reserve requirements in interconnected systems that recognizes the random nature of the system components. The basic objective in using a probabilistic technique is to maintain the unit commitment risk equal to or less than a specified value throughout the day.

In 1990, N. Chowdhury et al. [50] presented a probabilistic technique which can be used to develop unit commitment schedules for continually changing loads in an interconnected system configuration for a specified period. This technique that was designated as the 'two risk concept' involves the determination of probabilistic risk at two different levels. An interconnected system is required to satisfy a Single System Risk (SSR) in which possible assistance from its neighbors is not taken into account. In addition, the interconnected system is required to satisfy its Interconnected System Risk (ISR) in which assistance from its neighbors is considered. The unit commitment should be such that the unit commitment risk is less than or equal to the specified level. To achieve that, units have been committed according to a predetermined loading order such that a specified SSR level is satisfied at the isolated system level and a specified ISR level is also satisfied at the interconnected level.

In 1991, F. N. Lee et al. [51] presented a new concept of unit commitment risk analysis which explicitly models the stochastic sequence of events associated with rescheduling decision. Considering the concept of rescheduling, a stochastic model is proposed for risk analysis, and this model is illustrated via simulation results.

In 1993, N. Chowdhury [52] presented an energy based technique to assess spinning reserve requirements in an interconnected system. The proposed technique responds to unit size and lead time in a direct way. An assistance equivalent unit is expected to share the load of an assisted system in the event of a capacity outage until additional units can be brought into the troubled system. The expected energy assistance of one system to another will vary with a variation in the assistance equivalent unit and/or in the lead time of additional generation in the assisted system.

In 1993, M. E. Khan et al. [53] presented an algorithm, based on risk analysis, for unit commitment in composite generation and transmission systems. A new risk index designated as the composite system operating state risk is defined, using the probabilities of these operating states. This risk index can be utilized for system expansion planning and unit commitment in a composite system.

F. N Lee et al. [54], 1994, presented a multi area unit commitment method based on the sequential commitment procedures that resembles 'bidding'. Instead of the linear flow network representation usually used in multi area production simulations, the proposed method employs a dc power flow model to represent the inter area transmission network. This improved network model adds significant complexity to the multi area UCP.

1.2 HEURISTIC METHODS

Faced with such a complex problem, most utilities use non-rigorous computer aided empirical methods known as 'heuristic methods' [57-64], which avoid implementing mathematical programming models and make the unit commitment decisions according to a pre-calculated priority list, known as the 'merit order table' and incorporate all the operational constraints heuristically.

Heuristic methods have the following advantages: All the operational constraints are considered and feasible solutions (if there are any) are usually obtained, the solutions are economically reasonable, and the computational requirements, in terms of memory and running time are modest.

The main shortcoming of the heuristic methods is that they cannot guarantee the optimal solutions, or even provide an estimate of the extent of their sub-optimality. Therefore, in the heuristic methods a suboptimal solution is obtained due to the incomplete search of the solution space besides the lack of mathematical proof for reaching the optimal solution. Some of the popular heuristic methods are: Priority-List, and Expert Systems based methods.

1.2.1 PRIORITY-LIST

The Priority-List (PL) method [57-59] is one of the basic methods that has been used to solve UCP. The PL methods mimic the scheduling practices followed by system operators. The units are sorted in ascending order according to their average full load cost. The units are then committed according to this priority list, so that the most economic base load units are committed first and the peaking units last in order to meet the load demand. PL methods are very fast but they are highly heuristic and give schedules with relatively high production costs.

1.2.2 EXPERT SYSTEMS

Recently, Expert Systems (ES) have been used in solving the UCP as a mean of utilizing the expertise of the human operators [60-64]. The ES methods treat the global optimization problem as a sequence of simpler optimization steps, to the extent that such treatments can approximate the actual optimization process.

The basic strategy of using ES to solve the UCP is to use the previous knowledge of the system to get an optimal or suboptimal schedule for a given load pattern. The system knowledge could be a data base of different load patterns and the corresponding schedules solved by an efficient analytical method. The solution steps in the ES methods can be divided into three phases as follows:

- A load pattern and schedule database is built which contain a number of load patterns and their associated unit commitment schedules. A starting point schedule is then obtained for a given load pattern by searching in the data base for a load pattern that is nearest to that one.
- The selected schedule is examined for time intervals in which the spinning reserves are insufficient or surplus. Then, using a PL based optimization strategy, additional units are committed or decommitted in those time intervals. Both commitment and recommitment are performed with the minimum up/down time constraints being preserved.
- A global optimization is performed using additional rules. Then, a search procedure is performed at different time intervals of the entire time horizon in order to find out whether some replacement could lead to a more economical schedule.

1.3 ARTIFICIAL INTELLIGENCE METHODS

The growing interest for the application of Artificial Intelligence (AI) techniques to power engineering has introduced the potentials of using the state-of-the art in many problems in power systems. To offer an alternative to the existing methods, AI techniques have been applied to solve the UCP [65-74].

AI methods seem to be promising and are still evolving. Currently, four methods that are perceived as affiliated in some measure with the AI field have gained prominence as frameworks for solving different problems [65-74]:

- Neural Networks (NN), [65-66],
- Simulated Annealing (SA), [67],
- Genetic Algorithm (GA), [69-74], and
- Tabu Search (TS), [None].

GA, NN and TS are inspired by principles derived from biological processes, and SA is derived from material sciences. These methods need not be viewed competitively, and they comprise the emergence of promise for conquering the combinatorial explosion in a variety of decision-making arenas. NN have claimed intriguing successes in pattern-recognition applications, but have generally performed less than impressively in optimization settings. SA and GA have the attractive feature of assured convergence under appropriate assumptions.

In the following section a brief description of the applicability of these methods for the UCP is presented:

1.3.1 NEURAL NETWORKS

In recent years, NN computing has become an important branch of the AI. NN represent a new class of computing systems formed by thousands of simulated neurons, connected to each other in the same way as the brain neurons are interconnected.

Two main approaches have been used to solve the UCP by NN [65-66].

In the first approach, a NN functions as a pattern recognizer and this is achieved by training the NN using a set of pairs of load pattern and the corresponding optimal schedule. An optimal or suboptimal schedule is then obtained in the retrieving phase of the NN operation. This approach is usually used with other methods in the form of hybrid algorithm to refine the solution output from a NN.

In the second approach, NN are used to solve the optimization problem itself by formulating the energy function of the NN to include both the objective function and the constraints of the UCP.

Hiroshi Sasaki et al. [65], 1992, introduced the feasibility of applying the Hopfield-type NN to solve the UCP. In this approach, the NN gives the on/off states of the units at each period and then the output power of each unit is adjusted to meet the total demand. Another feature of the approach is that an ad hoc NN is installed to satisfy inequality constraints which take into account the standby reserve constraints and the minimum up/down time constraints. The approach also approximates the output of a running unit and its fuel usage by a constant, and it is necessary to use existing methods to balance the supply and demand. The proposed approach has been applied to solve an example of 30 units and 24 hours time periods; results obtained were close to those obtained using the LR method.

In 1993, M. H. Sendaula et al. [66] proposed a NN approach for solving the UCP. The approach combines the Hopfield-Tank and Chua-Lin type NN to simultaneously solve the unit commitment and economic dispatch problems. The approach is based on imbedding the various economic and electrical constraints of the UCP and EDP in a generalized energy function, and then defining the network dynamic in such a way that the generalized energy function is a Lyupunov function of the artificial NN. The novel feature of the proposed approach is that the nonlinear programming problem for EDP and the combinatorial optimization problem for the UCP are solved by one network. The model includes the spinning reserve and transmission losses but the minimum up/down constraints are not included. The effectiveness of the model is proven through an example of six buses and three generators.

Unfortunately up till now, the results of using the NN to solve the UCP are not superior to the previously used method due to two main drawbacks. The first one is the problem of entrapment in the local minimum in both two approaches. The second is the problem size that degrades the training efficiency in the first approach and also slows down the convergence in the second approach.

1.3.2 SIMULATED ANNEALING

A single paper has been published on solving the UCP using the SA method by Zhuang and Galiana in 1990 [67]. The method exploits the resemblance between the annealing of a metal and a minimization process.

In applying the SA, to solve the UCP, the basic idea is to choose a feasible solution at random and then get a neighbor to this solution. A move to this neighbor is performed if either it has a better (lower) objective value or, in case the neighbor has a higher objective function value, if $\exp(-\Delta E / Cp) \geq U(0,1)$, where ΔE is the increase in objective value if we move to the neighbor, and Cp is a control parameter (or temperature) decreased through the search according to a specified schedule. The effect of decreasing

Cp is that the probability of accepting an increase in the objective function value is decreased during the search.

In Zhuang's work [67], the constraints were divided into two categories easy constraints and difficult ones. The easy constraints were satisfied heuristically during the process of finding randomly feasible solutions, while the difficult constraints were penalized in the objective function.

The cooling schedule, used in [67], is the popular one. In this schedule, the control parameter is decremented by a constant factor, i.e., $\beta = \beta_0 \rho^m$ and $0<\rho<1$, where m is the iteration counter. The initial value of Cp is assumed empirically in the range of 100-500, while the final value of Cp is calculated as function of the problem size and the initial value of the objective function. The presented results showed that the method can yield highly near-optimal solutions and can accommodate high size problems and complicated constraints such as crew and maintenance constraints.

1.3.3 GENETIC ALGORITHMS

Genetic Algorithms have become increasingly popular in recent years in science and engineering disciplines. In the UCP several papers have been published [69-74]. The solution coding is the most important point in applying the GA to solve any optimization problem. Coding could be in a real or a binary form. Coded strings of solutions are called "chromosomes." A group of these solutions (chromosomes) are called population. Moving from one population of chromosomes to a new population is set by selection, together with a set of genetic operators of cross-over, mutation and inversion. Since the UCP lends itself to the binary coding in which zero denotes the OFF state and a one represents the ON state, all published works used the binary coding. A candidate solution is a string whose length is the product of the number of generating units and the scheduling periods.

Fitness function is the second issue in solving the UCP using GA. In the literature the fitness function is constructed as the summation of the operating costs and penalty terms for constraints violations.

A basic advantage of the GA solution is that it can be easily converted to work on parallel computers. A disadvantage of the GA is that, since they are stochastic optimization algorithms, the optimality of the solution they provide cannot be guaranteed. However, the results reported indicate good performance of the method.

In 1994, D. Dasgupta et al. [69] presented a paper which discusses the application of GA to solve the short term UCP. In this work, the problem is considered as a multi-period process and a simple GA is used for commitment scheduling. Each chromosome is encoded in the form of a position-dependent gene (bit string) representing the status of units available in the system, (on/off), at a specific time period. The fitness function is formulated by using a weighted sum of the objective function, and values of the penalty function based on the number of constraints violated and the extent of these violations. A scaled fitness function was used to determine the probability of selection of members in the population for breeding. To make the algorithm robust in finding near-optimal solutions, a number of feasible commitments with smaller costs were saved at each time period.

In 1995, X. Ma et al. [70] presented a new approach based on the GA to solve the UCP. The coding scheme used was the binary coding. A forced mutation operator was adopted to correct the solutions (or chromosomes) that do not satisfy the load demand and reserve constraints. The fitness function was constructed from the objective function and penalty terms for constraints violation. Two-point crossover was used. The algorithm was tested on a ten units example.

S. A. Kazarlis et al. [71], 1996, presented a GA solution to the UCP. The coding was implemented in a binary form. Fitness function was constructed from the objective function and penalty terms of constraints violation. A nonlinear transformation was used for fitness function scaling. Normal GA operators; crossover, mutation, and elitism were applied. Additionally another set of operators was implemented to apply hill-climbing-like techniques to the best chromosome of every generation. These set of operators are swap-mutation, and swap-window hill-climb. Another step taken was the implementation of a smooth and gradual application of the fitness function penalties that are most responsible for the complication of the search space. This modification concerns the replacement of the constant penalty multipliers with functions increasing with the generation index. With the technique of the varying quality function, the GA finally manages to locate the exact global optimum. The algorithm is applied successfully on one hundred units example.

In 1996, P. C. Yang et al. [72] presented a practical approach for using the GA to solve the UCP. The implemented algorithm deals with the constraints in a different manner. The minimum up/down time constraints are embedded in the binary representation inherently needed in the GA. The other constraints are dealt with by integrating penalty factors into the cost function to consider the constraint violations. The constraint violation penalty is in an exponential form of the violated percentage to avoid missing the schedule with a low cost which just fails to satisfy the constraint because of numerical round-off error in the computer. To emphasize the strings with better fitness and to speed up the convergence of the search, the raw fitness function obtained is normalized to between 0 and 1 inclusively to obtain the final fitness. The proposed algorithm, along with SA and LR is applied to solve a Taiwan power system consisting of 38-unit over a 24-hour period. With a reasonable computation time, the cost of the solution obtained by the GA approach was found to be the lowest among the three methods (GA, SA and LR).

S. O. Orero [73], 1996, proposed an enhanced GA approach for the UCP. The major difference between this approach and the previous ones is that it incorporates what was called 'a sequential decomposition logic', to provide a faster search mechanism. The coding is binary as usual. The fitness function includes penalty functions that are carefully graded to differentiate between feasible and non feasible solutions, but penalize the solutions that violate the linking constraints according to the magnitude of violation. The EDP is solved via a dual revised simplex linear programming algorithm. In this approach, the selection, mutation, and crossover operators are restricted to a single time interval. The time intervals are then considered in sequence starting from the first interval. As the sequence progresses all other variables such as minimum up/down times, ramp rates and spinning reserve requirements are checked for constraints violation and penalized accordingly. This method does not strictly partition the problem into single time spans, but involves a cumulative time span partitioning, in that the linking constraint parameters such as minimum up/down times and ramp rate limits are continuously updated and re-evaluated as the time steps increase. An advantage of this approach is that any constraints which are already satisfied cannot be violated later in the sequence. The method is applied to a 26-unit example with better results than a hybrid DP/NN approach.

G. B. Sheble' et al. [74], 1996, presented a paper to discuss the applicability of the GA approach to the UCP and the EDP. The first half of the paper presented the problems that the author has faced, when applying the GA to the UCP. The two main problems encountered when using a GA with penalty methods are the crossover operator can introduce new constraints violations that were not in either parent, and the problem of selecting penalty values for satisfying the five considered constraints is hopeless. These two problems resulted in each generation of population members having a similar fitness or similar unit commitment schedule cost as the preceding generation. In the second half of the paper an algorithm of solving the EDP using GA is implemented and successfully tested and compared with the lambda iteration method.

1.3.4 TABU SEARCH

In general terms, Tabu Search (TS) is an iterative improvement procedure in that it starts from some initial feasible solution and attempts to determine a better solution in the manner of a greatest-descent algorithm. However, TS is characterized by an ability to escape local optima (which usually cause simple descent algorithms to terminate) by using a short-term memory of recent solutions.

No work has been reported so far for solving the UCP by TS. In chapter 4 we will propose two new algorithms using TS to solve the UCP.

1.4 HYBRID ALGORITHMS

Hybrid algorithms are also well known techniques for solving engineering problems. Hybrid algorithms try to make use of the merits of different methods. Hence, the aim is to improve the performance of algorithms that are based on a single method. The main objective of proposing an algorithm as a hybrid of two or more methods is to speed up the convergence and/or to get better quality of solutions than that obtained when applying the individual methods.

Different hybrid algorithms, used to solve the UCP, are available in the literature [76-91]. These algorithms consist of two or more of the following methods: Classical Optimization (e.g. DP, LP, and LR), Heuristics, and Artificial Intelligence, (e.g. NN, and GA). Various types of hybrid algorithms could be grouped, according to the methods involved in the implementation of those algorithms, as follows:

- Classical methods, e.g. LP and DP [76-78],
- Classical and heuristic methods , e.g.
    - LP and heuristics methods [79],
    - DP and heuristics methods [80],
- Classical and AI methods, e.g.
    - DP and fuzzy logic methods [81],
    - DP and NN methods [82,83]
    - LR and NN methods [84]
- AI and heuristics methods [85-91], e.g.
    - NN and ES methods [85],

- •NN and heuristic methods [86-89],
- •GA and ES methods [90],
- •GA and PL methods [91]

•AI methods, [None].

In 1981, James G. Waight et al. [76] presented a paper that combines dynamic and linear programming techniques such that the operational constraints of reserve margins and ramp rates are optimally met by the resulting generation schedules. A major advantage of this method is that it can be used to determine the minimum cost of providing a particular level of reserve or operating with a particular set of ramp rate capabilities. Such a costing method makes it feasible to make economic decisions on whether to install new ramming capability. The scheduling problem is formulated as a two step optimization problem. The first step considers optimal scheduling over a period of time. In this step the scheduling problem is formulated as a DP problem. The operational period is divided into stages, and at each stage a number of combinations of generating units are examined. The second step is performed for each of these combinations. In this step, the problem is viewed as the EDP of a given set of generating units subject to reserve margins and other constraints. The EDP is solved using LP with Dantiz Wolfe decomposition.

In 1984, Hans P. Van Meeteren [77] presented a paper to solve the UCP using DP and LP approaches. This paper addresses itself to the UCP subject to constraints in the fuel supply system. In this work an iterative decoupled approach, based on both a unit commitment module and a fuel allocation module, has been proposed. In the unit commitment module a DP approach is used. In addition to that, more heuristics are introduced by defining a search window to which the actual dynamic search is restricted. The fuel allocation problem is formulated by using linear or piecewise linear models for both the fuel supply system and the generating units. LP is used to solve the fuel allocation problem.

In 1986, G. B. Sheble' et al. [78] proposed a method to solve the UCP, which is based on the LP and DP as a decision analysis problem. The main advantage of the proposed technique is the elimination of solution dependency on a prespecified PL for unit start-ups or unit shut-downs. A successive approximation in solution space algorithm is implemented. An initial feasible solution is obtained using any 'crashing' method. The results are used to define a coarse grid over the solution space. Next the load dispatch problem is solved for each possible transition from a previous solution point. The combination for the best transition is found and saved for the next stage. After all stages have been evaluated, the optimal transition is traced from the final stage to the first one. This process is repeated around the new optimal solution path until the change in objective function is within tolerance or the grid size is within tolerance. This approach is very similar to gradient optimization techniques since the neighborhood is first identified and then small changes are tried to determine if the minimum can be reduced while maintaining feasibility.

E. Khodaverdian et al. [79], 1986, presented a new method to solve large scale UCP. The method uses a hybrid algorithm of the discrete decision linear programming and heuristic techniques. The algorithm is capable of incorporating all the operational constraints of the system and fully feasible schedules. The algorithm is tested to solve the

UCP in the South-Western Region subsystem of the CEGB, UK with 74 units. In this approach, the unit commitment period is divided into a number of smaller equal or unequal time-intervals, the ends of which are defined as the time-steps. Some heuristic rules are designed to obtain a single-time-step unit commitment schedule. Then a discrete decision LP with bounded variables algorithm, based on the revised-simplex method, is used to solve the single-time-step UCP .

Various heuristics have been introduced in DP in order to reduce its execution time. The truncated window methodology is one of the schemes that provides a better compromise between the speed and the closeness of the solution to the optimal value. In 1991, Z. Ouyang et al. proposed a heuristic improvement of the truncated window DP for the UCP [80]. The proposed algorithm adjusts the window size of the DP according to the incremental load demands in adjacent hours, and controls the program execution to fine tune the optimization interactively.

In 1991, Chung-Ching Su proposed [81] a new approach for the UCP under an expected error in the forecasted load demand. The approach uses a fuzzy DP model in which the hourly loads, the cost, and system security are all expressed in fuzzy set notations. The constraints imposed on the system are divided into two groups. The power generation-load balance and spinning reserve requirements are treated as fuzzy constraints since they are related to the imprecise (fuzzy) hourly loads. The other constraints (minimum up/down time, and crew constraints) are still considered to be crisp. The possible states at each hour are crisp variables. Thus, the fuzzy DP model includes crisp state variables with some crisp constraints imposed on the states. The problem is modeled as a fuzzy objective function subject to two fuzzy constraints and two crisp constraints. The effectiveness of the proposed algorithm is demonstrated by solving the unit commitment of the Taiwan power system which contains 6 nuclear units and 48 thermal units.

Z. Ouyang et al. [82], 1992, presented a hybrid DP-artificial NN algorithm for the UCP. The proposed algorithm performed the unit commitment schedule in a two stage process. In the first stage, a preschedule with a degree of uncertainty is obtained as an output from a trained NN. Each unit in the schedule will have a certain degree of uncertainty presented by its probability in the output of NN. A high probability indicates that the unit is more likely to be committed at the current hour. The NN was trained with input/output data for most of the typical commitment schedules. A total of 35 training patterns were used in this study, which are generated by the LR method. In the second stage, the obtained schedule is refined using a modified DP procedure. The DP implementation suggested a new method for selecting window units in the truncated window DP, and new combinations were attempted at stages where uncertain units exit. The selection of a window is a major step in this approach. The uncertain units are arranged according to their probability values computed by the NN. Units with the same probability will be listed according to their priority. Also, in the DP algorithm, to speed up the solution process and save the memory space, a heuristic policy is used which limits the number of strategies saved at every stage to be less than or equal to a predefined value.

In 1995, Ruey-Hsun Liang et al. [83] presented a hybrid artificial NN-differential DP method for the scheduling of short term hydro generation. In the proposed method, the DP procedures are performed off-line on historical load data. The results are compiled and valuable information is obtained by using NN algorithms. The DP algorithm is then performed on-line according to the obtained information to give the hydro generation schedule for the forecasted load. Two types of NN algorithms, supervised and unsupervised learning is employed and compared in this paper. The effectiveness of the algorithm is demonstrated by the short term hydro scheduling of the Taiwan power system which consists of ten hydro plants.

In 1995, Adly A. Girgis et al. presented a paper of two parts [84]. The first part shows the application of artificial NN to load forecasting using new input-output models. The second part utilizes the results from the first part in the unit commitment. Two types of load forecasting, hourly and daily load forecast, are addressed. The results provide both the unit commitment schedule and the power generation for each unit using economic dispatch. The unit commitment problem was solved based on the principle of the Lagrangiane method. The unit commitment is based solely on economic dispatch and generator capacities.

Z. Ouyang et al. [85], 1992, proposed a short term unit commitment which employs a multi-stage NN-ES approach to achieve real time processing results. The operating constraints are presented as heuristic rules in the system where a feasible solution is obtained through inference. The NN are used as a pre-processor and post-processor stages. At the preprocessor stage, a load pattern matching scheme is performed to retrieve an existing optimal schedule from the database which would represent the closest solution to the given load profile. At the post-processor stage, a trained NN performs considerable adjustments to the optimal solution. The proposed approach performs the unit commitment in three steps. In the first step, an initial schedule is obtained by applying the forecasted load pattern to the trained NN. In the second step, an ES approach examines the selected schedule for the time intervals for which spinning reserves are insufficient. Then, using the PL additional units are committed to satisfy the spinning reserve constraints. In the third step, the resulted schedule of the last step is refined by applying it to the NN as an input and possible unit replacements are listed in the output which may further reduce the overall operating cost.

In 1993, C. Wang, et al. [86] proposed an algorithm to consider the ramp characteristics in starting up and shutting down the generating units as well as increasing and decreasing power generation. The proposed algorithm employs NN and some heuristic rules. These steps are used to complete the task of generation scheduling. First, the ramping constraints in the unit commitment are relaxed. A NN is used to generate a possible unit commitment schedule and a heuristic procedure is employed to modify the unit commitment to achieve a feasible and near optimal solution. Then, a dynamic adjusting process is adopted for the resulting schedule in order to incorporate the ramping constraints. Finally, a dynamic dispatch is performed to obtain a suitable unit generation schedule. A system with 26 thermal units is used to demonstrate that the method is very fast and can generate satisfactory results.

Wong et al. [87], 1990, has developed an artificial intelligence based algorithm for scheduling thermal generators for run-up-to-peak period. Later, in 1991, they extended the algorithm to cover a 24-hour scheduling horizon [88]. The algorithm is based on the heuristic-guided depth-first search framework. In the algorithm, the scheduling problem is represented as a search in a problem space that spans the set of all possible actions. To overcome the immense size of the problem, search heuristics are provided to the system to identify and select promising actions that would lead to more economical schedules. The accompanied application study demonstrates that the use of such simple but appropriate heuristics has been effective in reducing the problem to a manageable size, and directing searches towards more economical solutions.

K. Doan and K. P. Wong [87,88], 1995, extended the above work by developing a new scheduling system named search heuristic acquisition program by explanation simplification (SHAPES). SHAPES is a machine learning system capable of acquiring heuristics automatically, alleviating the need for heuristics to be manually provided, as in the previous system. This learning ability also allows the new system to adapt from one situation to another, by learning different sets of heuristics when dealing with load demands of different characteristics. The paper also describes methods of training and evaluating the effectiveness of the learning sub-system in improving search efficiency, as well as determining the performance of SHAPES in relation to the original scheduling algorithm.

In 1994, Gerald B. Sheble' et al. [90] presented a genetic-based UCP algorithm. The algorithm uses the ES s to satisfy some of the UCP constraints. The advantage of the algorithm is that the EDP routine is only used with the initialization and mutation subroutines. Since the mutation is a technique that changes a small percentage of the on/off status of the generating unit schedule, the only times ED is needed is for the hours where a mutation has occurred. An adaptive mutation operator is used. In early generations the members of the population are very distinctive, and do not need mutation. In later generations when the GA is locating good solutions, a method is needed to keep finding better solutions in these areas.

In 1995, S. O. Orero et al. [91] proposed an algorithm to incorporate a PL scheme in a hybrid GA to solve the UCP. In the GA coding process, the solution string length is the product of the scheduling period T, and the number of generating units N. Accordingly, the search space of the GA is then equal to $2^{T\times N}$ which is a very large number. Due to this problem, a premature convergence of the GA search has occurred. To counteract this problem, a method of decomposition was proposed to limit the GA search space to 2^{N}. This method sequentially solves the scheduling problem by limiting the GA search to one time interval as the search progresses with the unit minimum up/down constraints observed, while preserving the solutions already obtained earlier on in the search. The method uses the previous interval solution as a member of the starting population in a current time step. The PL algorithm is used to generate a schedule for each hour, which is then copied to the initial population of the GA for that hour. The algorithm is applied for two systems of 10 and 110 units. The obtained results are found to be superior to that obtained using the PL and the GA algorithms as individuals.

1.5 THESIS MOTIVATIONS AND CONTRIBUTIONS

1.5.1 THESIS MOTIVATIONS

Considering the previous discussions, several optimization techniques have been applied to the solution of the UCP. They range from heuristic approaches to the more sophisticated ones. Because of its nonconvex and combinatorial nature, the UCP is difficult to solve by conventional programming methods. Artificial Intelligence (AI) techniques, unlike strict mathematical methods, have the apparent ability to adapt to nonlinearities and discontinuities commonly found in power systems. In recent years AI techniques have captured interest in many fields of electric power engineering and this trend is likely to continue. The emergence of massively parallel computer made these algorithms of practical interest. The best known algorithms in this class include evolution programming, GA, simulated annealing, tabu search, and NN.

The motivation of this work was the success of the AI to solve many complex power systems problems and the aim of extending this to the optimal scheduling of power generating systems. Consequently, the objective of this work is to propose new AI-based algorithms to solve the UCP. The proposed methods for solution are as follows:

1- Simulated Annealing,

2- Tabu Search,

3- Genetic Algorithms, and

4- Hybrid Algorithms of the aforementioned methods.

Additional motivations of this work, as previously demonstrated, was the lack of papers that have been published in the application of AI techniques to solve the UCP [65-74]:

- Only one paper has been published about the solution of the UCP by Simulated Annealing [67].
- Some works about the solution of the UCP by Genetic Algorithms have been published [69-74].
- No work has been done in the application of Tabu Search to the UCP.

- No work has been done in the hybrid algorithms based on the AI methods solely as applied to the UCP

1.5.2 THESIS CONTRIBUTIONS

In brief, to achieve our objectives in this thesis of getting superior solutions than those obtained by other methods in the literature, the following contributions are achieved:

(i) Some new modifications to the existing problem formulation has been made. In this regard the relation between the binary variables is clearly stated in an augmented form. The new formulation is generalized and amenable to solution by both the classical and AI methods.

(ii) An efficient nonlinear programming routine to solve the EDP has been implemented. Solving the EDP is very crucial for achieving good results in the UCP. An original linear complementary algorithm to solve the EDP is introduced in this thesis. The proposed algorithm is proved to be fast and also more accurate than an IMSL quadratic routine that has been attempted in the first stages of this work.

(iii) New rules for generating randomly feasible solutions are proposed. A first step in solving combinatorial optimization problems is to have good rules for finding feasible trial solutions from an existing feasible solution. The trial solutions should be randomly generated, feasible, and span as much as possible the entire problem solution space. A major contribution of this work is the implementation of new rules to generate faster randomly feasible solutions.

(iv) Four new algorithms for solving the UCP are proposed and are listed as follows:

- A Simulated Annealing Algorithm [68,126],
- A Simple Tabu Search Algorithm,
- An Advanced Tabu Search Algorithm, and
- A Genetic Algorithm with local search [75].

(v) Three new hybrid algorithms for solving the UCP are also proposed and are listed as follows:

- A hybrid of Simulated Annealing and Tabu Search algorithms [93].
- A hybrid of Genetic and Tabu Search Algorithms.
- A hybrid of Genetic, Simulated Annealing and Tabu Search Algorithms [92].

1.6 THESIS ORGANIZATION

This thesis consists of nine chapters and three appendices organized as follows:

In chapter 2, the problem formulation is presented. Efficient new rules for randomly generating trial feasible solutions are also proposed. The implementation of the proposed linear complementary algorithm to solve the economic dispatch problem is detailed.

In chapter 3, a new implementation of a Simulated Annealing Algorithm (SAA) to solve the UCP is proposed. The combinatorial optimization subproblem of the UCP is solved using the proposed SAA while the EDP is solved via a quadratic programming routine. Two different cooling schedules are implemented and compared. Three examples are solved to test the developed computer model.

In chapter 4, we propose two new algorithms, based on the TS method, for the UCP. The first algorithm uses the short term memory procedure, while the second algorithm is based on advanced TS procedures. Different criteria for constructing the tabu list restrictions for the UCP are implemented and compared. Results of both algorithms along with a comparison of the results reported in the literature for the three solved examples are presented.

In chapter 5, an overview of the GA method is presented. The new proposed implementations of the GA as applied to solve the UCP along with the description of different GA components are presented. The detailed description of a local search algorithm that has been used to improve the performance of the GA is introduced. The computational results along with a comparison with previously published work are presented.

In chapter 6, we propose three different new hybrid algorithms for the UCP. The proposed hybrid algorithms integrate the use of the three previously introduced algorithms, SAA, TSA, and GA. The bases of hybridization of these algorithms are completely new ideas and are applied to the UCP for the first time. These algorithms are a hybrid algorithm of both SA and TS methods, a hybrid algorithm of GA and TSA and a hybrid algorithm integrating the three methods; SA, TS, and GA.

Chapter 7 is intended for the comparison between results of the seven proposed algorithms as well as the available results of other methods (LR, and IP) in the literature.

In chapter 8, for the aim of testing and emphasizing the effectiveness of the proposed algorithms presented in the last chapters, two of the proposed algorithms are applied to solve a real power system. The selected sample of data is extracted from the Saudi Consolidated Electric Company in the Eastern Province (SCECO-East). Considering the performance of the proposed algorithms discussed in the last chapters, the SA (as one of the individual methods) and the GT (as one of the hybrid techniques) algorithms have been selected. Modified versions of the two algorithms (ST and GT) are implemented to suit the selected practical system data.

Conclusions and recommendation for future research work in the UCP are presented in chapter 9.

Appendix A presents the details of the proposed nonlinear programming routine for solving the EDP.

Appendix B details the proofs of the SA method equations.

The data of the three systems extracted from the literature and used to demonstrate the effectiveness of the suggested techniques are shown in Appendix C.

1.7 SUMMARY

This chapter has introduced the definition of the unit commitment and the economic dispatch problems.

A survey of the methods which have been used for solving the UCP is presented. These methods have been divided into four major categories: classical optimization methods, heuristic methods, artificial intelligence methods, and hybrid algorithms. A summary of the contributions that have been achieved by the different methods along with a short description of these methods are presented.

The thesis motivation and the thesis organization have been presented.

The next chapter presents the problem formulation, proposed new rules of generating trial solutions, and a new algorithm for solving the economic dispatch problem.

CHAPTER TWO

PROBLEM STATEMENT,

NEW RULES FOR GENERATING TRIAL

SOLUTIONS AND

NEW ECONOMIC DISPATCH ALGORITHM

2.1 INTRODUCTION

The Unit Commitment Problem (UCP) is the problem of selecting the generating units to be in service during a scheduling period and for how long. The committed units must meet the system load and reserve requirements at minimum operating cost, subject to a variety of constraints. The Economic Dispatch Problem (EDP) deals with the optimal allocation of the load demand among the running units while satisfying the power balance equations and units operating limits [1].

The solution of the UCP using artificial intelligence techniques requires three major steps:

1. A problem statement or, system modeling,
2. Rules for generating trial solutions, and
3. An efficient algorithm for solving the EDP.

Problem Statement: Modeling of power system components affecting the economic operation of the system is the most important step when solving the UCP. The degree of details in components modeling varies with the desired accuracy and the nature of the problem under study. The basic components of a power system include generating power stations, transformer, transmission network, and system load.

This work is concerned with thermal generating units scheduling. Hence it is assumed that the network is capable of transmitting the power generated to the load centers without neither losses nor network failures. This means that the network is assumed to be perfectly reliable. Consequently, the following basic engineering assumptions are made[1-3]:

- The network interchange between the system under study and other systems is fixed.
- The load demand is not affected by adding or removing generating units.
- The operating cost of a generating unit is assumed to be composed of three components; start-up cost, spinning (no load) cost, and production (loading) cost.

In the UCP under consideration, one is interested in a solution which minimizes the total operating cost during the scheduling time horizon while several constraints are satisfied [20]. The objective function and the constraints of the UCP are described in Sections 2.2 and 2.3.

Rules for generating trial solutions: The corner stone in solving combinatorial optimization problems is to have good rules for generating feasible trial solutions starting from an existing feasible solution. The trial solutions (neighbors) should be randomly generated, feasible, and span as much as possible the entire problem solution space. Because of the constraints in the UCP, this is not a simple matter.

A major contribution of this work is the implementation of new rules to generate randomly feasible solutions faster [68]. In Section 2.4 we present these new rules with the help of illustrative examples.

The Economic Dispatch Problem: The economic dispatch problem is an essential problem when solving the UCP. Once a trial solution is generated, the corresponding operating cost of this solution is calculated by solving the EDP. Consequently, using an efficient and fast algorithm for solving the EDP improves the quality of the UCP solution, and therefore, the performance of the overall UCP algorithm.

In Section 2.6 an efficient algorithm for solving the EDP is presented. The method is based on Kuhn-Tucker conditions and is called the *linear complementary algorithm.*

The application of the linear complementary algorithm to solve the EDP is original. Our investigation showed that the results obtained by this algorithm are more accurate than those obtained using an IMSL quadratic programming routine.

2.2 THE OBJECTIVE FUNCTION

2.2.1 THE PRODUCTION COST

The major component of the operating cost, for thermal and nuclear units, is the power production cost of the committed units. The production cost is mainly the cost of fuel input per hour, while maintenance and labor contribute only to a small extent. Conventionally the unit production cost is expressed as a quadratic function of the unit output power as follows:

$$F_{it}(P_{it}) = A_i P^2_{it} + B_i P_{it} + C_i \qquad \text{\$/HR} \qquad (2.1)$$

2.2.2 THE START-UP COST

The second component of the operating cost is the start-up cost. The start-up cost is attributed to the amount of energy consumed to bring the unit "ON" line. The start-up cost depends upon the down time of the unit. This can vary from maximum value, when the unit is started from cold state, to a much smaller value, where the unit was recently turned off.

Calculation of the start-up cost depends also on the treatment method for the thermal unit during down time periods. There are two methods for unit treatment during the OFF hours; the cooling method and the banking method.

The former method allows the boiler of the unit to cool down and then reheat back up to the operating temperature when recommitted on line.

In the latter method, the boiler operating temperature is maintained during the OFF time using an additional amount of energy.

The cooling method is used in the present work, due to its practicability when applied to real power systems. In this work, the start-up cost, for a unit i at time t, based on the cooling method, is taken in a more general form as follows [41]:

$$ST_{it} = So_i[1 - D_i \exp(-Toff_i / Tdown_i)] + E_i \quad \$ \qquad (2.2)$$

Accordingly, the overall operating cost of the generating units in the scheduling time horizon (i.e. **objective function of the UCP**) is

$$F_T = \sum_{t=1}^{T} \sum_{i=1}^{N} (U_{it}F_{it}(P_{it}) + V_{it}ST_{it} + W_{it}SH_{it}) \quad \$ \qquad (2.3)$$

2.3 THE CONSTRAINTS

The unit commitment problem is subject to many constraints depending on the nature of the power system under study. The constraints which are taken into consideration in this work, may be classified into two main groups: system constraints and unit constraints.

2.3.1 SYSTEM CONSTRAINTS

The system constraints, sometimes called coupling constraints, include also two categories: the load demand and the spinning reserve constraints.

1- Load Demand Constraints

The load demand constraint is the most important constraint in the UCP. It basically means that the generated power from all committed units must meet the system load demand. This is formulated in the so called balance equation as follows:

$$\sum_{i=1}^{N} U_{it} P_{it} = PD_t \quad ; 1 \le t \le T \tag{2.4}$$

2- Spinning Reserve Constraint

The spinning (operating) reserve is the total amount of generation capacity available from all units synchronized (spinning) on the system minus the present load demand. It is important to determine the suitable allocation of spinning reserve from two points of view: the reliability requirements and the economical aspects.

There are various methods for determining the spinning reserve [1,20,34,52]:

- The reserve is computed as a percentage of the forecasted load demand, or
- It is determined such that the system can make up for a loss of the highest rating unit in a given period of time, or
- Determination of the reserve requirements as a function of the system reliability which is evaluated on a probabilistic basis.

In this work, the reserve is computed as a given prespecified amount which is a percentage of the forecasted load demand, i.e.

$$\sum_{i=1}^{N} U_{it} Pmax_i \ge (PD_t + R_t); \qquad 1 \le t \le T \tag{2.5}$$

2.3.2 UNIT CONSTRAINTS

The constraints on the generating units (sometimes called local constraints) are described as follow:

1- GENERATION LIMITS

The generation limits represent the minimum loading limit below which it is not economical to load the unit, and the maximum loading limit above which the unit should not be loaded.

$$U_{it}Pmin_i \le P_{it} \le Pmax_iU_{it} \quad ; 1 \le t \le T,\ 1 \le i \le N \tag{2.6}$$

2- MINIMUM UP/DOWN TIME

If the unit is running, it can not be turned OFF before a certain minimum time elapses. If the unit is also down, it can not be recommitted before a certain time elapse.

$$\begin{matrix} T_{off_i} \ge T_{down_i} \\ T_{on_i} \ge T_{up_i} \end{matrix} \quad ; \qquad 1 \le i \le N \tag{2.7}$$

These constraints could be formulated in a mathematical form as follows:

$$\sum_{l=0}^{T_{up_i}-1} U_{i,t+l} \ge V_{it}T_{up_i} \ ; \qquad 1 \le t \le T,\ 1 \le i \le N \tag{2.8}$$

$$\sum_{l=0}^{T_{down_i}-1} (1-U_{i,t+l}) \ge W_{it}T_{down_i} \ ; \qquad 1 \le t \le T,\ 1 \le i \le N \tag{2.9}$$

$$V_{it} \ge U_{it} - U_{i,t-1} \ ; \qquad 2 \le t \le T, 1 \le i \le N \tag{2.10}$$

$$W_{it} \ge U_{i,t-1} - U_{it} \ ; \qquad 2 \le t \le T,\ 1 \le i \le N \tag{2.11}$$

$$V_{i1} = U_{i1} \ ; \qquad 1 \le i \le N \tag{2.12}$$

$$W_{i1} = 1 - U_{i1} \ ; \qquad 1 \le i \le N \tag{2.13}$$

3- UNITS INITIAL STATUS CONSTRAINT

The status of unit (e.g. hours of being ON or OFF) before the first hour in the proposed schedule is an important factor to determine whether its new status violates the minimum up/down constraints. Also, the initial status of the unit affects the start-up cost calculations.

4- CREW CONSTRAINTS

If the plant consists of two or more units, they can not be turned ON at the same time due to some technical conditions or man power availability.

5- UNIT AVAILABILITY CONSTRAINT

Due to some abnormal conditions, e.g. forced outage or maintenance of a unit, the unit may become *unavailable.* The unit may also be forced in service to increase reliability or stability of the system, hence the unit becomes *must run* or *fixed at a certain output.* Otherwise the unit is *available.* The availability constraint specifies the unit to be in one of the following different situations; *unavailable, must run, available, or fixed output (MW).*

6- UNITS DERATING CONSTRAINT

During the life time of a unit its performance could be changed due to many conditions, e.g. aging factor, the environment, etc. These conditions may cause derating of the generating unit. Consequently, the unit maximum and minimum limits are changed.

2.4 PROPOSED NEW RULES FOR GENERATING TRIAL SOLUTIONS

One of the most important issues in solving combinatorial optimization problems is generating a trial solution as a neighbor to an existing solution. The neighbors should be randomly generated, feasible, and span as much as possible the entire problem solution space. In the course of generating feasible solutions, the most difficult constraints to satisfy are the minimum up/down times.

The proposed rules [68] applied to get a trial solution as a neighbor of an existing feasible solution are described, with the help of an example, in the following steps. The following values are assumed: T=12, Tup_i=2 or 4 and $Tdown_i$=1 or 4.

Step (1): Generate randomly a unit i, i ~ UD(1,N), and an hour t, t~UD(1,T). Fig.(2.1) shows the status of some unit i over a period of 12 hours. The unit is ON between the periods 5 and 8 inclusive.

Step (2): If unit i at hour t is ON, (e.g. 5,6,7 or 8 in Fig.(2.1)), then go to Step (3) to consider switching it ON around time t.

Otherwise, if unit i at hour t is OFF, (e.g. 1,2,3,4,9,10,11 or 12), then go to Step (4) to consider switching it OFF around time t.

Step (3): Switching the unit i from ON to OFF

a- Move from the hour t backward and forward in time, to find the length of the ON period.

In this example if t=6, then Ton_i=8-5+1=4, and the unit is ON during hours 5,6,7,8.

b- If $Ton_i = Tup_i$, then turn the unit OFF in all hours comprising Ton_i.

In the example if Tup_i=4, then switch the unit OFF at t=5,6,7,8 (Fig.2.2-a).

c- If $Ton_i > Tup_i$, then generate L ~ UD(1, $Ton_i - Tup_i$).

d- Turn the unit OFF for the hours t1,t1+1,....t1+L-1, where t1 is the first hour at which the unit is ON.

In the example if Tup_i=2, then L~UD(1,2). Hence, the following two solutions are possible:

if L=1, then the unit is turned OFF at t=5 (Fig.(2.2-b)), and

if L=2, the unit is turned OFF at t=5,6, (Fig.(2-.2-c)).

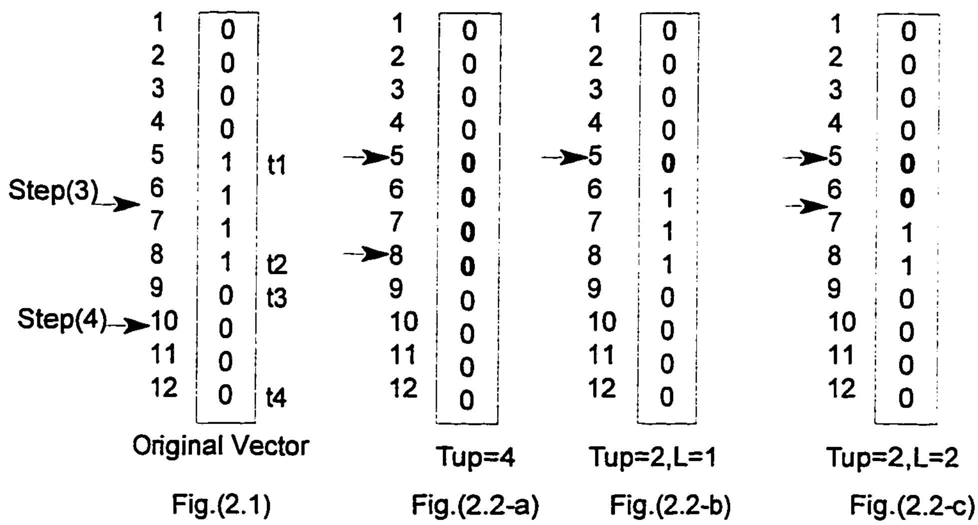


Fig. (2.1) and (2.2): Illustrative example for the rules of generating trial solutions

Step (4): Switching the unit i from OFF to ON

a- Move from the hour t backward and forward in time, to find the length of the OFF period.

In the example if t=10, then $Toff_i$=12-9+1=4, the unit is OFF during hours 9,10,11,12.

b- If $Toff_i = Tdown_i$, then turn the unit ON in all hours of $Toff_i$.

In the example if $Tdown_i$=4, then switch the unit ON at t=9,10,11,12 (Fig.2.3-a).

c- If $Toff_i > Tdown_i$, then generate L ~ UD(1, $Toff_i - Tdown_i$).

d- Turn the unit ON for the hours t3,t3+1,t3+2,....t3+L-1, where t3 is the first hour at which the unit is OFF.

In the example if $Tdown_i$=1, then L~UD(1,3). Hence, the following three solutions are possible:

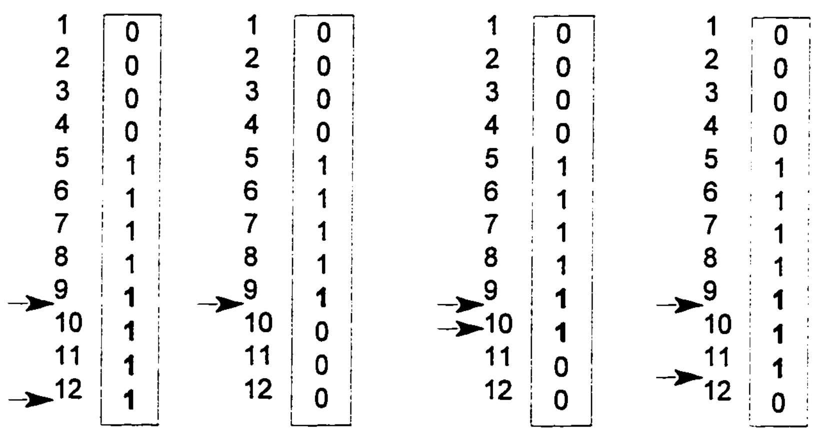
if L=1, then the unit is turned ON at t=9 (Fig.(2.3-b)),

if L=2, the unit is turned ON at t=9,10 (Fig.(2.3-c)), and

if L=3, the unit is turned ON at t=9,10,11 (Fig.(2.3-d).

Step (5): Check for reserve constraints

Check the reserve constraints satisfaction for the changed time periods in Steps (3) and (4). If it is satisfied, then the obtained trial solution is feasible, otherwise go to Step (1).



Tdown=4 Fig.(2.3-a) | Tdown=1, L=1 Fig.(2.3-b) | Tdown=1, L=2 Fig.(2.3-c) | Tdown=1, L=3 Fig.(2.3-d)

Fig. (2.3): Illustrative example for the rules of generating trial solutions

2.5 GENERATING AN INITIAL SOLUTION

Solving the UCP using any combinatorial optimization algorithm requires a starting feasible schedule. The generated starting solution must be randomly generated and feasible. The following algorithm is used for finding this starting solution [68]:

Step (1): Set U=V=P=0, t=1.

Step (2): Do the following substeps :

a- Generate randomly a unit i, i ~ UD(1,N).

b- If the unit i at hour t is OFF (U_{it}=0), then go to Step (3). Otherwise go to Step (2-a) to choose another unit.

Step (3): Follow the procedure in Step (4), in Section 2.4, to consider switching this unit ON starting from hour t.

Step (4): If t=T, go to Step (5), otherwise set t=t+1 and go to Step (2).

Step (5): Check the reserve constraints for all hours. Repeat Steps (2) and (3) for the hours at which the constraints are not satisfied.

2.6 A NEW ALGORITHM FOR THE ECONOMIC DISPATCH PROBLEM

The EDP is the heart of any algorithm used to solve the UCP. To get the objective function of a given trial solution we have to solve the EDP. The accuracy and speed of convergence for the selected routine to solve the EDP is crucial in the efficiency of the overall UCP algorithm.

Since the production cost of the UCP, formulated in Section 2.2, is a quadratic function, the EDP is solved using a quadratic programming routine. In this section we present an efficient algorithm for solving the EDP. The method is based on Kuhn-Tucker conditions and is called the *linear complementary algorithm* [6].

The application of the linear complementary algorithm to solve the EDP is new. It is an efficient and fast algorithm for solving the quadratic programming problems. In this algorithm the Kuhn-Tucker conditions are solved as a linear program problem in a tableau form. In the early stages of this research, some experiments were performed to test the proposed linear complementary algorithm. We found that the results obtained by this algorithm are more accurate than those obtained using an IMSL quadratic programming routine.

In brief, the EDP for a one hour in the scheduling time horizon could be formulated as the minimization of the summation of production costs of the committed units in this hour subjected to the load demand and unit limits constraints as follows:

$$\text{Minimize } \sum_{i=1}^{N} F_{it}(P_{it}) = A_i P^2_{it} + B_i P_{it} + C_i \qquad \$/HR \qquad (2.14)$$

subject to:

$$\sum_{i=1}^{N} P_{it} = PD_t \quad ; 1 \leq t \leq T \qquad (2.15)$$

and

$$Pmin_i \leq P_{it} \leq Pmax_i \quad ; 1 \leq t \leq T,\ 1 \leq i \leq N \qquad (2.16)$$

The theoretical basis of the linear complementary algorithm is presented in Appendix A. The detailed reformulation of the EDP as a linear complementary problem is described in the following section.

2.6.1 THE ECONOMIC DISPATCH PROBLEM IN A LINEAR COMPLEMENTARY FORM

The economic dispatch problem is generally a nonlinear programming problem. If the production cost functions of the generating units are taken in a quadratic form, then the problem can be formulated as a quadratic programming problem. Accordingly the linear complementary algorithm could be used to solve the Khun-Tucker conditions of this problems as proved in Appendix A.

The original formulation of the EDP, as described in equations (2.14), (2.15) and (2.16), could be written, for a single time period, in a simple form as follows:

Minimize $\sum_{i=1}^{N} F_i(p_i) = A_i p_i^2 + B_i p_i + C_i$ $/HR (2-17)

subject to:

$$\sum_{i=1}^{N} p_i = PD \qquad (2\text{-}18)$$

and

$$m_i \le p_i \le x_i \quad ; 1 \le i \le N \qquad (2\text{-}19)$$

where m_i and x_i are the lower and upper limits of unit i respectively.

For a system of N generating units, the number of constraints is 2N inequality constraints and one equality constraint. These constraints can be reduced to only N+1 constraints, hence the tableau size and the computation effort will also be reduced. The reduction is done by defining new variables to cancel one of the sides (upper or lower) of the inequality constraints as follows:

Let $p'_i = p_i - m_i$ (2-20)

then $p_i = p'_i + m_i$, (2-21)

substituting form (2-21) in (2-17), (2-18) and (2-19) the **EDP problem is reformulated as follows:**

Minimize $\sum_{i=1}^{N} F_i(p'_i) = A'_i p'^2_i + B'_i p'_i + C'_i$ $/HR (2-22)

subject to:

$$\sum_{i=1}^{N} p'_i = PD' \qquad (2\text{-}23)$$

$$p'_i \le x'_i \quad ; 1 \le i \le N \qquad (2\text{-}24)$$

$$p'_i \geq 0 \ ; \ 1 \leq i \leq N \tag{2-25}$$

where

$$x'_i = x_i - m_i \tag{2-26}$$

$$PD' = PD - \sum_{i=1}^{N} m_i \tag{2-27}$$

$$A'_i = A_i \tag{2-28}$$

$$B'_i = 2m_iA_i + B_i \tag{2-29}$$

$$C'_i = A_i m'^2_i + B_i m + C_i \tag{2-30}$$

Accordingly, the number of constraints are one equality constraint (equation (2-23)) and N inequality constraints (equation (2-24)), in addition to the nonnegativity constraints of the new N variables p'_i's.

Now an analogy between the reduced formulation of the EDP and the quadratic programming formulation, described in Section A-2, could be stated as follows:

$$A' \Leftrightarrow H \tag{2-31}$$

$$B' \Leftrightarrow c \tag{2-32}$$

$$x' \Leftrightarrow b \tag{2-33}$$

$$\text{Unit matrix} \Leftrightarrow A \tag{2-34}$$

Using the problem formulation (2-22) to (2-25) and the analogy equations (2-31) to (2-34), the EDP is solved using the linear complementary algorithm described in Section A.1.4.

2.6.2 TABLEAU SIZE FOR THE ECONOMIC DISPATCH PROBLEM

Considering the modified formulation of the EDP, the number of constraints is reduced by N, where N is the number of variables or the number of committed generating units. Consequently, the tableau size of the EDP in the linear complementary formulation is as follows:

Let N is the number of variables (generating units outputs).

Since we have N constraints as the upper limits on the generating units, and one constraint of the load demand, then $L = N + N + 1 = 2N + 1$

The tableau size is then $L \times (2L+2) = (2N + 1) \times (4N + 4)$.

Accordingly, the tableau Size for our solved examples are as follows:

For the 10-units example, N=10, the tableau size is 21 x 44.

For the 26-units example, N=26, the tableau size is 53 x 108.

For the 24-units example, N=24, the tableau size is 49 x 100.

2.7 SUMMARY

This chapter presented three different subjects: the problem statement, rules for generating trial solutions, and a new algorithm for solving the economic dispatch problem.

In the problem statement the objective function and the constraints of the UCP are formulated in a more generalized form.

New rules for generating trial feasible solutions are proposed.

An original application of the linear complementary algorithm for solving the EDP is also discussed.

The next Chapter presents the first proposed algorithm to solve the UCP. This algorithm uses the simulating annealing method.

CHAPTER THREE

A SIMULATED ANNEALING ALGORITHM FOR UNIT COMMITMENT

3.1 INTRODUCTION

Annealing is the physical process of heating up a solid until it melts, followed by cooling it down until it crystallizes into a state with a perfect lattice. During this process, the free energy of the solid is minimized. Practice shows that the cooling must be done carefully in order not to get trapped in a locally optimal lattice structure with crystal imperfections. In combinatorial optimization, we can define a similar process. This process can be formulated as the problem of finding-among a potentially very large number of solutions- a solution with minimal cost. Now, by establishing a correspondence between the cost function and the free energy, and between the solutions and physical states, we can introduce a solution method in the field of combinatorial optimization based on a simulation of the physical annealing process. The resulting method is called Simulated Annealing (SA). The salient features of the SA method could be summarized as follows [94-104]:

- It could find a high quality solution that does not strongly depend on the choice of the initial solution.
- It does not need a complicated mathematical model of the problem under study.
- It can start with any given solution and try to improve it. This feature could be utilized to improve a solution obtained from other suboptimal or heuristic methods.
- It has been theoretically proved to converge to the optimum solution [94].
- It does not need large computer memory.

In this chapter we propose a new implementation of a Simulated Annealing Algorithm (SAA) to solve the UCP. The combinatorial optimization subproblem of the UCP is solved using the proposed SAA while the EDP is solved via a quadratic programming routine. Two different cooling schedules are implemented and compared. Three examples are solved to test the developed computer model.

In the next section, a general description of the SA method is presented, followed in Section 3.3, by the description of the cooling schedules. Section 3.4 presents the detailed description of the proposed SAA to solve the UCP. Section 3.5 is intended for the comparison between the obtained results and others as reported in the literature [67]. In Section 3.6 the computational results along with a comparison with previously published work are presented.

3.2 SIMULATED ANNEALING METHOD

3.2.1 PHYSICAL CONCEPTS OF SIMULATED ANNEALING

Simulated Annealing, was independently introduced by Kirkpatrick, Gela and Vecchi in 1982 and 1983 [96] and Cerny in 1985 [97]. Annealing , physically refers to the process of heating up a solid to a high temperature followed by slow cooling achieved by decreasing the temperature of the environment in steps. At each step the temperature is maintained constant for a period of time sufficient for the solid to reach thermal equilibrium. At equilibrium, the solid could have many configurations, each corresponding to different spins of the electrons and to a specific energy level.

At equilibrium the probability of a given configuration, P_{confg}, is given by Boltzman distribution; $P_{confg} = K.\exp(-E_{confg} / Cp)$, where E_{confg} is the energy of the given configuration and K is a constant [98].

Metropolis et al. [99], proposed a Monte Carlo method to simulate the process of reaching thermal equilibrium at a fixed temperature Cp. In this method, a randomly generated perturbation of the current configuration of the solid is applied so that a trial configuration is obtained. Let E_c and E_t denote the energy level of the current and trial configurations, respectively. If $E_c > E_t$, then a lower energy level has been reached, and the trial configuration is accepted and becomes the current configuration. On the other hand, if $E_c \leq E_t$ then the trial configuration is accepted as the current configuration with probability $\exp[(E_c - E_t) / Cp]$.

The process continues where a transition to a configuration of higher energy level is not necessarily rejected. Eventually thermal equilibrium is achieved after a large number of perturbations, where the probability of a configuration approaches Boltzman distribution. By gradually decreasing Cp and repeating Metropolis simulation, new lower energy levels become achievable. As Cp approaches zero, the least energy configurations will have a positive probability of occurring.

3.2.2 APPLICATION OF SA METHOD TO COMBINATORIAL OPTIMIZATION PROBLEMS

By making an analogy between the annealing process and the optimization problem, a great class of combinatorial optimization problems can be solved following the same procedure of transition from an equilibrium state to another, reaching the minimum energy level of the system. This analogy can be set as follows [98]:

- Solutions in the combinatorial optimization problem are equivalent to states (configurations) of the physical system.
- The cost of a solution is equivalent to the energy of a state .
- A control parameter, Cp, is introduced to play the role of the temperature in the annealing process.

In applying the SAA to solve the combinatorial optimization problem, the basic idea is to choose a feasible solution at random and then get a neighbor to this solution. A move to this neighbor is performed if either it has a lower objective function value or, in case of higher objective function value, if $\exp(-\Delta E / Cp) \geq U(0,1)$, where ΔE is the increase in the objective function value if we moved to the neighbor. The effect of decreasing Cp is that the probability of accepting an increase in the objective function value is decreased during the search.

The most important part in using the SAA is to have good rules for finding a diversified and intensified neighborhood so that a large amount of the solution space is explored. Another important issue is how to select the initial value of Cp and how it should be decreased during the search.

3.2.3 A GENERAL SIMULATED ANNEALING ALGORITHM

A general SAA can be described as follows [94-104]:

Step (0): Initialize the iteration count k=0 and select the temperature, $Cp = Cp_0$, to be sufficiently high such that the probability of accepting any solution is close to 1.

Step (1): Set an initial feasible solution = current solution, X_i, with corresponding objective function value E_i.

Step (2): If the equilibrium condition is satisfied go to Step (5), else execute Steps (3) and (4)

Step (3): Generate a trial solution X_j, as a neighbor to X_i. Let E_j be the corresponding objective function value

Step (4): Acceptance test: If $E_j \leq E_i$: accept the trial solution, set $X_i = X_j$ and go to Step (2). Otherwise: if $\exp[(E_i - E_j)/Cp] \geq U(0,1)$ set $X_i = X_j$ and go to Step (2). Else go to Step (2)

Step (5): If the stopping criterion is satisfied then stop, else decrease the temperature Cp^k and go to Step (2).

3.3 COOLING SCHEDULES

A finite-time implementation of the SAA can be realized by generating homogenous Markov chains of a finite length for a finite sequence of descending values of the control parameter Cp. To achieve this, one must specify a set of parameters that governs the convergence of the algorithm. These parameters form a cooling schedule. The parameters of the cooling schedules are as follows:

- an initial value of the control parameter
- a decrement function for decreasing the control parameter
- a final value of the control parameter specified by the stopping criterion
- a finite length of each homogenous Markov chain.

The search for adequate cooling schedules has been the subject of study in many papers [94,101-104].

In this work, two cooling schedules are implemented, namely, the Polynomial-Time and Kirk's cooling schedules. The description of these cooling schedules is presented in the following sections.

3.3.1 THE POLYNOMIAL-TIME COOLING SCHEDULE

This cooling schedule leads to a Polynomial-Time execution of the SAA, but it does not guarantee the convergence of the final cost, as obtained by the algorithm, to the optimal value. The different parameters of the cooling schedule are determined based on the statistics calculated during the search. In the following we describe these parameters [94,101,102].

3.3.1.1 Initial Value Of The Control Parameter

The initial value of Cp, is obtained from the requirement that virtually all proposed trial solutions should be accepted. Assume that a sequence of m trials is generated at a certain value of Cp. Let m_1 denote the number of trials for which the objective function value does not exceed the respective current solution. Thus, $m_2 = m - m_1$ is the number of trials that result in an increasing cost.

It can be shown that the acceptance ratio, X can be approximated by [94]:

$$X \approx (m_1 + m_2.\exp(-\overset{(+)}{\Delta f} / Cp)) / (m_1 + m_2) \qquad (3.1)$$

where, $\overset{(+)}{\Delta f}$ is the average difference in cost over the m_2 cost-increasing trials.

From which the new temperature Cp is

$$Cp = \overset{(+)}{\Delta f} / \ln(m_2 / (m_2.X - m_1(1 - X)) \qquad (3.2)$$

3.3.1.2 Decrement of the Control Parameter

The next value of the control parameter, Cp^{k+1}, is related to the current value, Cp^k by the following function [94]:

$$Cp^{k+1} = Cp^k / (1 + (Cp^k . \ln(1 + \delta) / 3\sigma Cp^k \qquad (3.3)$$

where σ is calculated during the search according to the equations given in Appendix B. Small values of δ lead to small decrements in Cp. Typical values of δ are between 0.1 and 0.5.

3.3.1.3 THE FINAL VALUE OF THE CONTROL PARAMETER

Termination in the Polynomial-Time cooling schedule is based on an extrapolation of the expected average cost at the final value of the control parameter. Hence, the algorithm is terminated if for some value of k we have [94,101,102]:

$$\frac{Cp^k}{\langle f\rangle_\infty} \cdot \left.\frac{\partial \langle f\rangle_{Cp}}{\partial Cp}\right|_{Cp=Cp_k} < \varepsilon \tag{3.4}$$

where: $\langle f\rangle_\infty \approx \langle f\rangle_{Cpo}$ is the average cost at initial value of control parameter Cp_0.

$\langle f\rangle_{Cp}$ is the average cost at kth Markov chain.

$\left.\frac{\partial\langle f\rangle_{Cp}}{\partial Cp}\right|_{Cp=Cp_k}$ is the rate of change in the average cost at Cp^k.

ε is some small positive number. In our implementation ε =0.00001.

3.3.1.4 THE LENGTH OF MARKOV CHAINS

In [94], it is concluded that the decrement function of the control parameter, as given in (3.3), requires only a 'small' number of trial solution to rapidly approach the stationary distribution for a given next value of the control parameter. The word 'small' can be specified as the number of transitions for which the algorithm has a sufficiently large probability of visiting at least a major part of the neighborhood of a given solution. In general, a chain length of more than 100 transitions is reasonable [94]. In our implementation good results have been reached at a chain length of 150.

Proofs and more details of the polynomial-time cooling schedule equations are presented in Appendix B.

3.3.2 KIRK'S COOLING SCHEDULE

This cooling schedule was originally proposed by Kirkpatrick, Gelatt and Vecchi [96] in (1982 & 1983). This schedule has been used in many applications of the SAA and is based on a number of conceptually simple empirical rules. The parameters of this cooling schedule are described in the following subsections [94,96]:

3.3.2.1 INITIAL VALUE OF THE CONTROL PARAMETER

It is recommended to start with an arbitrary control parameter Cp [94]. If the percentage of the accepted trials solutions is close to 1, then this temperature is a satisfactory starting Cp. On the other hand, if this acceptance ratio is not close to 1, then Cp has to be increased iteratively until the required acceptance ratio is reached.

This can be achieved by starting off at a small positive value of Cp and multiplying it with a constant factor, larger than 1, until the corresponding value of the acceptance ratio, calculated from the generated transitions, is close to 1. In the physical system analogy, this corresponds to heating up the solid until all particles are randomly arranged in the liquid phase.

In our implementation, this procedure is accelerated by multiplying Cp by the reciprocal of the acceptance ratio.

3.3.2.2 DECREMENT OF THE CONTROL PARAMETER

It is important to make "small" decrement in the values of the control parameter, to allow for a very slow cooling and consequently reach an equilibrium at each value of control parameter, Cp. A frequently used decrement function is given by

$$Cp^{k+1} = \alpha . Cp^{k}, \qquad k=1,2, \tag{3.5}$$

Where α is a constant smaller than but close to 1. Typical values lie among 0.8 and 0.99.

3.3.2.3 FINAL VALUE OF THE CONTROL PARAMETER

Execution of the algorithm is terminated if the value of the cost function of the solution obtained in the last trial of the Markov chain remains unchanged for a number of consecutive chains (Lm). In our implementation, Lm is taken as 500 chains.

3.3.2.4 LENGTH OF THE MARKOV CHAIN

The length of Markov chains, L^k, is based on the requirement that equilibrium is to be restored at each value of Cp. This is achieved after the acceptance of at least some fixed number of transitions. However, since the transitions are accepted with decreasing probability, one would obtain $L^k \rightarrow \infty$ as $Cp^k \rightarrow 0$. Consequently, L^k is bounded by some constant L_{max} to avoid extremely long Markov chains for small values of Cp^k. In this work, the chain length is guided by the changes of the best solution that has been obtained thus far. The chain length is assumed equal to 150 unless the best solution changes. If so, the chain length is extended by another 150 iterations.

In the following section, we describe the details of the proposed SAA as applied to the UCP.

3.4 THE PROPOSED SIMULATED ANNEALING ALGORITHM

In solving the UCP, two types of variables need to be determined. The binary unit status variables U and V and the continuous units output power variables, P. The problem can then be decomposed into two subproblems: a combinatorial optimization problem in U and V and a nonlinear optimization problem in P. The SAA is used to solve the combinatorial optimization problem and a quadratic programming routine is used to solve the nonlinear optimization problem [68].

The main steps of the proposed SAA are presented as follows:

Step (0): Initialize all variables (U,V,P) and set iteration counter k=0.

Step (1): Find randomly an initial feasible solution (U_c^K, V_c^K), (see Section 2.5).

Step (2): Calculate the total operating cost, F_c^k, as the sum of F_{it} and S_{it} in two steps:

- Solve the economic dispatch problem (see Section 2.6).
- Calculate the start-up cost.

Step (3): Determine the initial temperature Cp^k, that results in a high probability of accepting any solution.

Step (4): If equilibrium is achieved go to Step (7). Otherwise; Repeat Steps (5) and (6) for the same temperature Cp^k, until the equilibrium criterion is satisfied.

Step (5): Find a trial solution (U_t^K, V_t^K), a neighbor to (U_c^K, V_c^K), with objective function value, F_t^K (see Section 2.4).

Step (6): Perform the acceptance test then go to Step (7). If $F_t^k \le F_c^k$ or $\exp[(F_c^k - F_t^k)/Cp] \ge U(0,1)$ then accept the trial solution and let (U_c^K, V_c^K)= (U_t^K, V_t^K), otherwise reject the trial solution.

Step (7): If the stopping criterion is satisfied then stop. Else decrease the temperature to Cp^{k+1} according to the cooling schedule, set k=k+1, and go to Step (4).

In the following section, we present a comparison with other SAA reported in the literature.

3.5 COMPARISON WITH OTHER SAA IN THE LITERATURE

There is only one application of the SA to the UCP available in the literature [67]. This will be referred to as SAA-67. There are four major differences between SAA-67 and the proposed algorithm [68]. These differences are as follows:

- In SAA-67, the starting solution is obtained using a priority list method which could be considered as a suboptimal solution, while that in the proposed algorithm we start with a randomly generated solution which may be far from the optimal one.
- There is no rule for selecting the initial temperature in SAA-67. In the proposed algorithm the initial value of the temperature is determined by applying the heating process until a prespecified value of acceptance ratio (typical values used 0.8-0.95) is reached.
- In SAA-67, the trial solutions may not be feasible and a penalty term is used for constraints violation. In the proposed algorithm, all trial solutions are feasible which results in considerable saving in the CPU time.
- Kirk's cooling schedule is used in SAA-67. The temperature is decreased by multiplying the initial temperature by a constant between 0 and 1 raised to the iteration number. In the proposed algorithm, the polynomial-time cooling schedule is used which is based on the statistics calculation during the search.

A comparison of the results obtained by the two algorithms and other methods is presented in the next section.

3.6 NUMERICAL RESULTS OF THE SAA

Based on the SAA and the proposed rules for generating random trial solution, a computer model has been implemented [68]. The model offers different choices for finding trial solutions, e.g. completely random, semi random, and a mix of both with certain probability. The two schedules of Kirk and the Polynomial-Time are implemented and compared.

To compare our results with SAA-67, we implemented the algorithm described in reference [67] and used the parameter settings (initial, decrement and final temperature) as described in the reference.

Three examples from the literature are solved. The first two examples include 10 generating units each while the third contains 26 units. The scheduling time horizon for all cases is 24 hours. The full data of the three examples are presented in Appendix C. Example 1 [29], was solved by Lagrangian Relaxation, LR. Example 2 [41], was solved by Integer Programming, IP. Example 3 [62,63], was solved by Expert Systems, ES.

The polynomial-time cooling schedule has been used in all of the solved examples. A number of tests on the performance of the proposed SAA, have been carried out on the three examples to find the most suitable cooling schedule parameters settings. The following parameters for the polynomial-time cooling schedule have been chosen after running a number of simulations: chain length=150, ε=0.00001, δ=0.3, initial acceptance ratio, x=0.95, and the maximum number of iterations = 5000.

Table (3.1) presents the comparison between the results of Examples 1 and 2, solved by LR, IP, and our SAA. The results show the improvement achieved by our algorithm over both IP and LR results.

Table (3.2) shows the comparison between the results obtained for Examples 1,2, and 3 solved by SAA-67 and our SAA. It is obvious that our SAA achieves a considerable reduction in the operating costs for the three examples.

Detailed results for Example 1 are given in Tables (3.3) and (3.4). Table (3.3) showing the load sharing among the committed units in the 24 hours. Table (3.4) gives the hourly load demand, and the corresponding economic dispatch costs, start-up costs, and total operating cost. Similar detailed results for Examples 2 and 3 are also shown in Tables (3.5)-(3.9).

Table (3.1) Comparison between our proposed SAA, the LR and the IP

	Example	LR [29]	IP [41]	Our SAA
Total Cost ($)	1	540895	-	536622
”	2	-	60667	59512
% Saving	1	0	-	0.79
”	2	-	0	1.9

Table (3.2) Comparison between our proposed SAA and the SAA-67

	Example	SAA-67	Our SAA
Total Cost ($)	1	538803	536622
”	2	59512	59512
”	3	663833	662664
% Cost Saving	1	0.38	0.79
”	2	1.9	1.9
”	3	0	0.17

Table (3.3) Power sharing (MW) of Example 1.

HR	Unit Number*							
	2	3	4	6	7	8	9	10
1	400.0	0	0	185.0	0	350.3	0	89.7
2	395.4	0	0	181.1	0	338.4	0	85.2
3	355.4	0	0	168.7	0	301.0	0	75.0
4	333.1	0	0	161.8	0	280.1	0	75.0
5	400.0	0	0	185.0	0	350.3	0	89.7
6	400.0	0	0	191.9	0	371.1	339.4	97.6
7	400.0	0	343.0	200.0	0	375.0	507.0	145.0
8	400.0	0	420.0	200.0	311.5	375.0	693.5	0.0
9	400.0	0	420.0	200.0	420.6	375.0	805.0	229.4
10	400.0	444.6	420.0	200.0	358.1	375.0	741.1	211.3
11	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
12	400.0	514.1	420.0	200.0	436.1	375.0	820.9	233.9
13	400.0	479.4	420.0	200.0	397.1	375.0	781.0	222.6
14	400.0	389.0	420.0	200.0	295.6	375.0	677.2	193.2
15	400.0	310.1	410.8	200.0	250.0	375.0	586.6	167.5
16	400.0	266.6	368.3	200.0	250.0	375.0	536.7	153.4
17	400.0	317.3	417.9	200.0	250.0	375.0	594.9	169.9
18	400.0	458.5	420.0	200.0	373.7	375.0	757.0	215.8
19	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
20	400.0	375.1	420.0	200.0	280.0	375.0	661.2	188.7
21	400.0	0	305.0	200.0	250.0	375.0	462.6	132.4
22	383.5	0	150.0	177.4	250.0	327.2	280.9	81.0
23	241.9	0	130.0	133.4	250.0	194.7	275.0	75.0
24	175.1	0	130.0	112.7	250.0	132.3	275.0	75.0

* Units (1) and (5) are OFF all hours.

Table (3.4) Load demand and hourly costs of Example 1.

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.03E+03	9.67E+03	0.00E+00	9.67E+03
2	1.00E+03	9.45E+03	0.00E+00	9.45E+03
3	9.00E+02	8.56E+03	0.00E+00	8.56E+03
4	8.50E+02	8.12E+03	0.00E+00	8.12E+03
5	1.03E+03	9.67E+03	0.00E+00	9.67E+03
6	1.40E+03	1.36E+04	9.50E+02	1.46E+04
7	1.97E+03	1.92E+04	6.50E+02	1.99E+04
8	2.40E+03	2.39E+04	9.50E+02	2.48E+04
9	2.85E+03	2.84E+04	6.25E+02	2.90E+04
10	3.15E+03	3.17E+04	9.50E+02	3.27E+04
11	3.30E+03	3.32E+04	0.00E+00	3.32E+04
12	3.40E+03	3.42E+04	0.00E+00	3.42E+04
13	3.28E+03	3.30E+04	0.00E+00	3.30E+04
14	2.95E+03	2.97E+04	0.00E+00	2.97E+04
15	2.70E+03	2.73E+04	0.00E+00	2.73E+04
16	2.55E+03	2.58E+04	0.00E+00	2.58E+04
17	2.73E+03	2.75E+04	0.00E+00	2.75E+04
18	3.20E+03	3.22E+04	0.00E+00	3.22E+04
19	3.30E+03	3.32E+04	0.00E+00	3.32E+04
20	2.90E+03	2.92E+04	0.00E+00	2.92E+04
21	2.13E+03	2.13E+04	0.00E+00	2.13E+04
22	1.65E+03	1.70E+04	0.00E+00	1.70E+04
23	1.30E+03	1.39E+04	0.00E+00	1.39E+04
24	1.15E+03	1.27E+04	0.00E+00	1.27E+04

Total operating cost = $536622

Table (3.5) Power sharing (MW) of Example 2.

H	Unit Number									
	1	2	3	4	5	6	7	8	9	10
1	60.00	80.00	100.00	120.00	150.00	189.99	372.59	0.00	186.42	200.00
2	60.00	80.00	100.00	109.98	150.00	170.54	332.62	0.00	168.85	200.00
3	60.00	80.00	100.00	99.72	146.67	155.53	301.77	0.00	155.30	200.00
4	60.00	80.00	100.00	98.10	143.74	153.15	296.87	0.00	153.14	200.00
5	60.00	80.00	100.00	96.47	140.80	150.77	291.97	0.00	150.99	200.00
6	60.00	80.00	100.00	101.47	149.82	158.09	307.02	0.00	157.60	200.00
7	60.00	80.00	100.00	109.98	150.00	170.54	332.62	0.00	168.85	200.00
8	60.00	80.00	100.00	101.47	149.82	158.09	307.02	0.00	157.60	200.00
9	60.00	80.00	100.00	96.47	140.80	150.77	291.97	0.00	150.99	200.00
10	60.00	80.00	100.00	117.59	150.00	0.00	355.50	0.00	178.91	200.00
11	60.00	80.00	100.00	109.14	150.00	0.00	330.11	0.00	167.75	200.00
12	60.00	80.00	100.00	106.33	150.00	0.00	321.64	0.00	164.03	200.00
13	60.00	80.00	100.00	101.20	149.33	0.00	306.22	0.00	157.25	200.00
14	60.00	80.00	100.00	98.96	145.29	0.00	299.47	0.00	154.28	200.00
15	60.00	80.00	100.00	97.00	141.75	0.00	293.56	0.00	151.69	200.00
16	60.00	77.69	97.37	93.62	135.68	0.00	283.42	0.00	147.23	200.00
17	60.00	74.60	94.26	90.42	129.91	0.00	273.80	0.00	143.00	200.00
18	60.00	71.52	91.15	87.23	124.15	0.00	264.18	0.00	138.78	200.00
19	60.00	66.77	86.37	82.31	115.28	0.00	250.00	0.00	132.27	200.00
20	59.82	64.41	83.99	79.86	110.88	0.00	250.00	0.00	129.04	200.00
21	58.17	62.29	81.85	77.66	106.91	0.00	250.00	0.00	126.13	200.00
22	60.00	69.92	89.54	85.57	121.17	0.00	259.20	0.00	136.59	200.00
23	60.00	76.20	95.86	92.08	132.89	0.00	278.77	0.00	145.19	200.00
23	60.00	80.00	100.00	120.00	150.00	189.99	372.59	9.00	0.00	186.42

Table (3.6) Load demand and hourly costs of Example 2.

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.46E+03	3.06E+03	0.00E+00	3.06E+03
2	1.37E+03	2.86E+03	0.00E+00	2.86E+03
3	1.30E+03	2.70E+03	0.00E+00	2.70E+03
4	1.29E+03	2.67E+03	0.00E+00	2.67E+03
5	1.27E+03	2.64E+03	0.00E+00	2.64E+03
6	1.31E+03	2.73E+03	0.00E+00	2.73E+03
7	1.37E+03	2.86E+03	0.00E+00	2.86E+03
8	1.31E+03	2.73E+03	0.00E+00	2.73E+03
9	1.27E+03	2.64E+03	0.00E+00	2.64E+03
10	1.24E+03	2.57E+03	0.00E+00	2.57E+03
11	1.20E+03	2.47E+03	0.00E+00	2.47E+03
12	1.18E+03	2.44E+03	0.00E+00	2.44E+03
13	1.15E+03	2.38E+03	0.00E+00	2.38E+03
14	1.14E+03	2.34E+03	0.00E+00	2.34E+03
15	1.12E+03	2.31E+03	0.00E+00	2.31E+03
16	1.10E+03	2.25E+03	0.00E+00	2.25E+03
17	1.07E+03	2.19E+03	0.00E+00	2.19E+03
18	1.04E+03	2.13E+03	0.00E+00	2.13E+03
19	9.93E+02	2.04E+03	0.00E+00	2.04E+03
20	9.78E+02	2.01E+03	0.00E+00	2.01E+03
21	9.63E+02	1.98E+03	0.00E+00	1.98E+03
22	1.02E+03	2.10E+03	0.00E+00	2.10E+03
23	1.08E+03	2.22E+03	0.00E+00	2.22E+03
24	1.46E+03	3.06E+03	1.65E+02	3.22E+03

Total operating cost = $59512

Table (3.7) Power sharing (MW) of Example 3 (for units 1-13).

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	36.40	15.20	15.20
2	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	16.40	15.20	15.20
3	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20
4	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20
5	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	42.40	15.20	0.00	15.20
6	2.40	2.40	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	40.60	15.20
7	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	55.65
8	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
9	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
10	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
11	0.00	0.00	0.00	2.40	2.40	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
12	0.00	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	2.40	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	2.40	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	2.40	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	2.40	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
24	2.40	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	76.00	76.00	33.00	15.20

Table (3.8) Power sharing (MW) of Example 3 (for units 14-26).

HR	Unit Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	347.2	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	327.2	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	25.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	25.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	68.95	0.00	350.0	350.0	350.0
8	100.0	100.0	85.70	155.0	155.0	155.0	155.0	0.00	68.95	68.95	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	185.7	68.95	68.95	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	185.7	68.95	68.95	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	75.25	68.95	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	165.7	68.95	68.95	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	178.1	68.95	68.95	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	158.1	68.95	68.95	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	88.10	68.95	68.95	350.0	350.0	350.0
16	100.0	44.15	25.00	155.0	155.0	155.0	155.0	68.95	68.95	68.95	350.0	350.0	350.0
17	100.0	84.15	25.00	155.0	155.0	155.0	155.0	68.95	68.95	68.95	350.0	350.0	350.0
18	100.0	100.0	99.15	155.0	155.0	155.0	155.0	68.95	68.95	68.95	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	168.1	68.95	68.95	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	75.25	68.95	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	183.3	68.95	68.95	350.0	350.0	350.0
22	100.0	100.0	94.35	155.0	155.0	155.0	155.0	68.95	68.95	68.95	350.0	350.0	350.0
23	0.00	77.25	25.00	155.0	155.0	155.0	155.0	68.95	0.00	0.00	350.0	350.0	350.0
24	0.00	25.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0

Table (3.9) Load demand and hourly costs of Example 3.

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.79E+04	0.00E+00	1.79E+04
2	1.80E+03	1.77E+04	0.00E+00	1.77E+04
3	1.72E+03	1.65E+04	0.00E+00	1.65E+04
4	1.70E+03	1.63E+04	0.00E+00	1.63E+04
5	1.75E+03	1.69E+04	8.00E+01	1.70E+04
6	1.91E+03	1.95E+04	1.80E+02	1.97E+04
7	2.05E+03	2.22E+04	3.00E+02	2.25E+04
8	2.40E+03	2.95E+04	5.00E+02	3.00E+04
9	2.60E+03	3.43E+04	3.00E+02	3.46E+04
10	2.60E+03	3.43E+04	0.00E+00	3.43E+04
11	2.62E+03	3.48E+04	0.00E+00	3.48E+04
12	2.58E+03	3.38E+04	0.00E+00	3.38E+04
13	2.59E+03	3.40E+04	0.00E+00	3.40E+04
14	2.57E+03	3.35E+04	0.00E+00	3.35E+04
15	2.50E+03	3.19E+04	0.00E+00	3.19E+04
16	2.35E+03	2.91E+04	0.00E+00	2.91E+04
17	2.39E+03	2.98E+04	0.00E+00	2.98E+04
18	2.48E+03	3.15E+04	0.00E+00	3.15E+04
19	2.58E+03	3.38E+04	0.00E+00	3.38E+04
20	2.62E+03	3.48E+04	0.00E+00	3.48E+04
21	2.60E+03	3.43E+04	0.00E+00	3.43E+04
22	2.48E+03	3.16E+04	0.00E+00	3.16E+04
23	2.15E+03	2.42E+04	0.00E+00	2.42E+04
24	1.90E+03	1.93E+04	0.00E+00	1.93E+04

Total operating cost = $662664

3.7 SUMMARY

A new implementation of a SAA to solve the UCP has been presented in this Chapter. Two different cooling schedules for the SAA are implemented and compared.

The detailed description of the proposed SAA is given. The comparison between our SAA and other SAA reported in the literature is also presented. The computational results along with a comparison with the previously published classical optimization methods showed the effectiveness of the proposed SAA.

In the next Chapter, two new algorithms based on the tabu search methods will be described.

CHAPTER FOUR

NEW TABU SEARCH ALGORITHMS FOR UNIT COMMITMENT

4.1 INTRODUCTION

Tabu Search (TS) is a powerful optimization procedure that has been successfully applied to a number of combinatorial optimization problems [100-116]. It has the ability to avoid entrapment in local minima. TS employs a flexible memory system (in contrast to 'memoryless' systems, such as SA and GAs, and rigid memory systems as in branch-and-bound). Specific attention is given to the Short Term Memory (STM) component of TS, which has provided solutions superior to the best obtained by other methods for a variety of problems [108]. Advanced TS procedures are also used for sophisticated problems. These procedures include, in addition to the STM, Intermediate Term Memory (ITM), Long Term Memory (LTM), and Strategic Oscillations (SO).

In this chapter, we propose two new algorithms for the UCP based on the TS method. The first algorithm uses the STM procedure, while the second algorithm is based on advanced TS procedures. Different criteria for constructing the Tabu List (TL) restrictions for the UCP are implemented and compared. Several examples are solved to test the proposed algorithms.

In the next section, a general Tabu Search Algorithm (TSA) is presented, followed in Section 4.3, by a Simple Tabu Search Algorithm (STSA), based on the STM procedures as applied to the UCP. Section 4.4 presents the detailed description of the different TL approaches that have been used in the developed algorithms. In Section 4.5 the computational results of the STSA along with a comparison with a previously published work are presented. In Section 4.6 a theoretical overview of the advanced TS procedures is presented, followed in Section 4.7, by the detailed description of an ATSA for the UCP. Section 4.8 details the computational results of the ATSA along with a comparison with the results of other methods reported in the literature.

4.2 TABU SEARCH METHOD

4.2.1 OVERVIEW

In general terms, TS is an iterative improvement procedure which starts from some initial feasible solution and attempts to determine a better solution in the manner of a greatest-descent algorithm. However, TS is characterized by an ability to escape local optima (which usually cause simple descent algorithms to terminate) by using a short term memory of recent solutions. Moreover, TS permits backtracking to previous solutions, which may ultimately lead , via a different direction, to better solutions [109].

The main two components of a TSA are the TL restrictions and the Aspiration Level (AV) of the solution associated with these restrictions. Discussion of these terms are presented in the following sections.

4.2.2 TABU LIST RESTRICTIONS

TS may be viewed as a 'meta-heuristic' superimposed on another heuristic. The approach undertake to surpass local optimality by a strategy of forbidding (or, more broadly, penalizing) certain moves. The purpose of classifying certain moves as forbidden - i.e. "tabu" - is basically to prevent cycling. Moves that hold tabu status are generally a small fraction of those available, and a move loses its tabu status to become once again accessible after a relatively short time.

The choice of appropriate types of tabu restrictions "list" depends on the problem under study. The elements of the TL are determined by a function that utilizes historical information from the search process, extending up to "Z" iterations in the past, where Z (TL size) can be fixed or variable depending on the application or the stage of the search.

The TL restrictions could be stated directly as a given change of variables (moves) or indirectly as a set of logical relationships or linear inequalities. Usage of these two approaches depends on the size of the TL for the problem under study.

A TL is managed by recording moves in the order in which they are made. Each time a new element is added to the "bottom" of a list, the oldest element on the list is dropped from the "top". The TL is designed to insure the elimination of cycles of length equal to the TL size. Empirically [108], TL sizes that provide good results often grow with the size of the problem and stronger restrictions are generally coupled with smaller lists.

The way to identify a good TL size for a given problem class and choice of tabu restrictions is simply to watch for the occurrence of cycling when the size is too small and the deterioration in solution quality when the size is too large (caused by forbidding too many moves). The best sizes lie in an intermediate range between these extremes. In some applications a simple choice of Z in a range centered around 7 seems to be quite effective [106].

4.2.3 ASPIRATION CRITERIA

Another key issue of TS arises when the move under consideration has been found to be tabu. Associated with each entry in the TL there is a certain value for the evaluation function called Aspiration Level (AV). If appropriate aspiration criteria are satisfied, the move will still be considered admissible in spite of the tabu classification. Roughly speaking, AV criteria are designed to override tabu status if a move is "good enough" with the compatibility of the goal of preventing the solution process from cycling [106]. Different forms of aspiration criteria are available. The one we use in this study is to override the tabu status if the tabued moves yield a solution which has a better evaluation function than the one obtained earlier for the same move.

4.2.4 STOPPING CRITERIA

There may be several possible stopping conditions for the search. In our implementation we stop the search if any of the following two conditions is satisfied:

- The number of iterations performed since the best solution last changed is greater than a prespecified maximum number of iterations, or
- The maximum allowable number of iterations is reached.

4.2.5 GENERAL TABU SEARCH ALGORITHM

In applying the TSA, to solve a combinatorial optimization problem, the basic idea is to choose a feasible solution at random and then get a neighbor to this solution. A move to this neighbor is performed if either it does not belong to the TL or, in case of being in the TL it passes the AV test. During these search procedures the best solution is always updated and stored aside until the stopping criteria is satisfied.

A general TSA, based on the STM, for combinatorial optimization problems can be described as follows:

The following notation is used in the algorithm:

X: The set of feasible solutions for a given problem.

x: Current solution, $x \in X$.

x'': Best solution reached.

x': Best solution among a sample of trial solutions.

E (x): Evaluation function of solution x.

N (x): Set of neighborhood of $x \in X$ (trial solutions).

S (x): Sample of neighborhood, of x , $S(x) \in N(x)$.

SS (x): Sorted sample in ascending order according to their evaluation functions, E (x).

Step (0): Set the TL as empty and the AV to be zero.

Step (1): Set iteration counter K=0. Select an initial solution $x \in X$, and set x''=x.

Step (2): Generate randomly a set of trial solutions $S(x) \in N(x)$ (neighbor to the current solution x) and sort them in an ascending order, to obtain SS (x). Let x' be the best trial solution in the sorted set SS (x) (the first in the sorted set).

Step (3): If E (x')>E (x'') , go to Step (4), else set the best solution x''=x' and go to Step (4).

Step (4): Perform the tabu test. If x' is NOT in the TL, then accept it as a current solution, set x=x', and update the TL and AV and go to Step (6), else go to Step (5).

Step (5): Perform the AV test. If satisfied, then override the tabu state ,Set x=x', update the AV and go to Step (7), else go to Step (6).

Step (6): If the end of the SS (x) is reached, go to Step (7), otherwise, let x' be the next solution in the SS (x) and got Step (3)

Step (7): Perform termination test. If the stopping criterion is satisfied then stop, else Set K=K+1 and go to Step (2).

The main steps of the TSA are also shown in the flow chart of Fig. (4.1).

In the following section we describe the details of the general TSA as applied to the UCP.

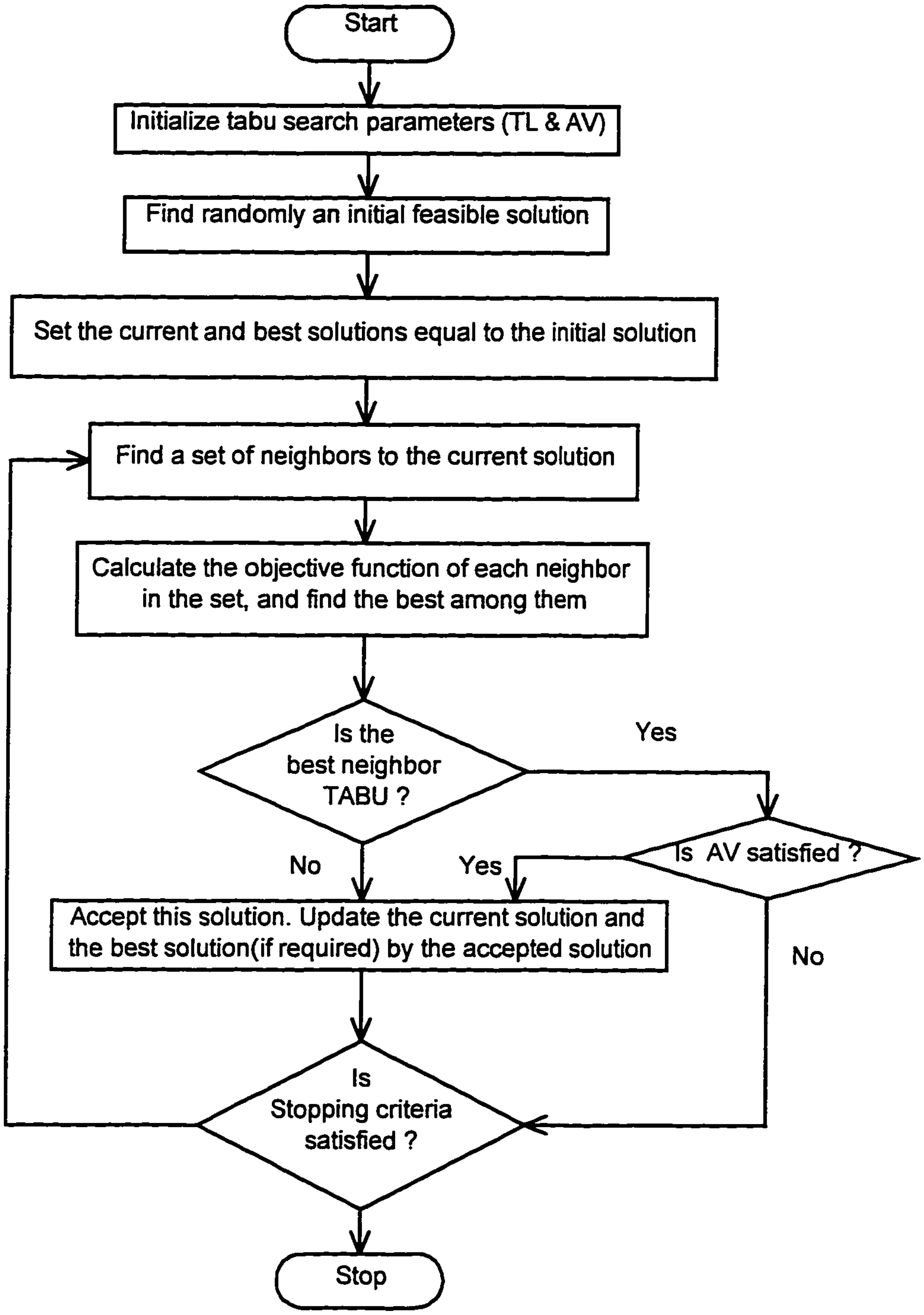


Fig. (4.1) Flow chart of a general Tabu Search Algorithm

4.3 THE PROPOSED TABU SEARCH ALGORITHM FOR UNIT COMMITMENT

As previously explained, the UCP can be decomposed into two subproblems, a combinatorial optimization problem in U and V and a nonlinear optimization problem in P. TS is used to solve the combinatorial optimization and the nonlinear optimization is solved via a quadratic programming routine. In this section a STSA, based on the STM approach, is introduced. The proposed algorithm contains three major steps:

- First, the generation of randomly feasible trial solutions.
- Second, the calculation of the objective function of the given solution by solving the EDP.
- Third, the application of the TS procedures to accept or reject the solution at hand.

The details of the STSA as applied to the UCP are given in the following steps:

Step (0): Initialize all variables (U, V, P) to be zeros and set iteration counter K=0.

Step (1): Generate, randomly, an initial current feasible solution (U_i^0, V_i^0), (see Section 2.5).

Step (2): Calculate the total operating cost, F_i^0 , for this solution in two steps:

-Solve the EDP to get the output power (P_i^0) and the corresponding production cost (see Section 2.6).

-Calculate the start-up cost for this schedule.

Step (3): Set the global best solution equal to the current solution, $(U_B, V_B) = (U_i^0, V_i^0)$, $F_B = F_i^0$.

Step (4): Find a set of trial solutions S (U_i^K, V_i^K), that are neighbors to the current solution (U_i^K, V_i^K), see Section 2.4 , with objective function values, $F^k(S)$.

Step (5): Sort the set of solutions in an ascending order. Let $SF^k(S)$ be the sorted values.

Let (U_b^K, V_b^K) be the best trial solution in the set, with an objective function F_b .

Step (6): If $F_b \geq F_B$ go to Step (7), else update the global best solution, set $(u_B, v_B) = (u_b^K, v_b^K)$ and go to Step (7).

Step (7): If the trial solution (u_b^K, v_b^K) is NOT in the TL, then update the TL, the AV and the current solution; Set $(u_i^K, v_i^K) = (u_b^K, v_b^K)$, and $F_i^k = F_b$ and go to Step (9), else go to Step (8).

Step (8): If the AV test is NOT satisfied go to Step (9), else override the tabu state, set $(u_i^K, v_i^K) = (u_b^K, v_b^K)$, update the AV and go to Step (10).

Step (9): If the end of the $SF^k(S)$ is reached, go to Step (10), otherwise let (u_b^K, v_b^K) be the next solution in the $SF^k(S)$ and go to Step (6).

Step (10): Stop if the termination criterion is satisfied, else set K=K+1 and go to Step (4).

In the following section, we describe some methods to create TL for the UCP.

4.4 PROPOSED TABU LIST TYPES FOR UCP

The TL embodies one of the primary STM functions of the TS procedure, which it executes by recording only the "Z" most recent moves, where Z is the TL size. A move in the UCP is defined as switching a unit from On to OFF or the opposite at some hours in the scheduling horizon. Since the solution matrices in the UCP (U and V) have large sizes (TxN), it is worth proposing and testing different approaches to create the TL moves attributes rather than recording a full solution matrix.

4.4.1 THE PROPOSED TABU LISTS APPROACHES FOR UCP

In this section, we propose original concepts for creating the TL for the UCP. During the early stages of implementing the TSA to solve the UCP, five approaches for creating the TL restrictions were tested with the aim of selecting the best among them. In our implementation we create a separate TL for each generating unit. The Generating Unit Tabu List (GUTL) has a dimension of ZxL, where L is the recorded move attributes length. In the following, the five proposed approaches of TL types for the UCP are described:

To illustrate the implementation of the proposed different approaches of TL a numerical example is shown in Figs. (4.2) and (4.3). Fig. (4.2) shows an initial trial solution of a unit and four different moves generated at random. Move 1, for example, is generated from the initial schedule by randomly selecting hour three (t=3) as the instant of changing the schedule. Move 2 is generated from move 1 and the change starts at instant t=7.

4.4.1.1 APPROACH (1)

In this approach each GUTL is a one dimensional array of length Z. Each entry records a time that has been previously selected randomly to generate a trial solution for this unit, irrespective of the unit status at that time. In Fig. (4.3) the TL implementation for the example of Fig. (4.2) using this approach is shown. As mentioned, only hours at which the schedule starts to be changed are recorded, i.e. hours 3,7,1 and 4 at the moves 1,2,3 and 4 respectively.

4.4.1.2 APPROACH (2)

The TL created in this approach will be of dimension Zx2. Each entry records the time that has been previously selected randomly to generate a trial solution for this unit, in addition to the unit status at this time. The implementation of this approach is shown in Fig. (4.3). For example, at the hour 3 (move 1), the unit status is 0 while at the hour 4 (move 3), the unit status was 1.

4.4.1.3 APPROACH (3)

In this case each GUTL contains one dimensional array of Z entries. Each entry records the number of ON periods for the respective unit (the number of ones in the column of that unit in the matrix U). In Fig. (4.3) the first entry (corresponds to move 1) of the TL using this approach shows 10, which is equivalent to the number of ON hours at that move.

	Initial Trial (t=3)	Move 1 (t=7)	Move 2 (t=1)	Move 3 (t=4)	Move 4
1	1	1	1	0	0
2	1	1	1	0	0
3	0	1	1	0	0
4	0	1	1	1	0
5	0	1	1	1	0
6	0	1	1	1	0
7	1	1	0	0	0
8	1	1	0	0	0
9	1	1	1	1	1
10	1	1	1	1	1
11	0	0	0	0	0
12	0	0	0	0	0

Fig.(4.2) Example of trial solutions for one unit (scheduling time horizon=12)

Approach 1	Approach 2	Approach 3	Approach 4	Approach 5
3	3, 0	10	1, 11	1023
7	7, 1	8	1, 9, 7,11	1598
1	1, 1	5	4, 9, 7,11	1592
4	4, 1	2	9, 11	1536

Fig. (4.3) Different approaches of tabu lists implementation

4.4.1.4 APPROACH (4)

In this approach we record the instances at which a unit is turned ON and OFF during the scheduling horizon. These instances come in pairs. If these ON-OFF pairs occur m times during the scheduling time horizon then GUTL will have the size Zx (2m).

Fig. (4.3) shows the TL of the unit when applying this approach to the solutions generated in Fig. (4.2). The TL entries show the starting and the shut down hours for each trial solution respectively. For example, at move 2, the unit started at hour 1 and shut down at hour 7, then started again at hour 9 and shut down at hour 11. Hence, the entry of this move in the TL is recorded as 1 and 9 as starting hours and the shut down hours are recorded as 7 and 11.

4.4.1.5 APPROACH (5)

In this approach the solution vector for each generating unit (which has 0 or 1 values) is recorded as its equivalent decimal number. Hence, the Generating Unit Tabu List (GUTL) is a one dimensional array of length Z. Each entry records the equivalent decimal number of a specific trial solution for that unit. By using this approach we record all information of the trial solution using minimum memory requirements. The TL implementation using this approach is shown in Fig. (4.3). As shown, the entries of the TL represent the equivalent decimal number for each binary vector of a trial solution. It is clear that this approach insures the uniqueness of the representation of a specific trial solution.

In the following section, the comparison between the results of the aforementioned five approaches is presented.

4.4.1.6 THE COMPARISON BETWEEN THE DIFFERENT TABU LIST APPROACHES

To find the most efficient approach among the five described approaches for creating the TL for the UCP, several tests were conducted. Example 1 was solved with different initial solutions and different random seeds using the five different approaches of the TL. To summarize the results, Table (4.1) shows the daily operating costs for the proposed five approaches of TL types as applied to example 1 [29]. In all cases we started with the same initial feasible solution and the same random number seed. It is obvious that the results of approaches 4 and 5 are the best. The reason is that the attributes of the moves are fully recorded, hence the search becomes more precise which prevents cycling during the age of TL. However approach 5 requires less memory space.

Table (4.1) Comparison of the Five Proposed Approaches of TL Types Using TL Size of 7 (Example 1)

Approach No.	1	2	3	4	5
TL Dimension	Z x 1	Z x 2	Z x 1	Z x 2m	Z x 1
Cost ($)	540986	540409	540174	538390	538390

4.4.2 TABU LIST SIZE FOR UCP

The size of the TL determines the most suitable number of moves to be recorded. To find a suitable size of TL, values of Z between 5 and 30 have been tested. Table (4.2) shows that the best value of Z was related to the TL restrictions type, where a more restricted TL corresponds to small size and visa versa. In our implementation, the results are based on a TL of size 7, which was found to be the best TL size for attempted examples.

Different experiments were carried out on different examples to find the most suitable TL size. Table (4.2) shows the daily operating costs obtained for example 1, using approach 4, with different TL sizes starting with the same initial solution. The best results for this example are obtained at a tabu size value of 7 as shown in the table. This is in agreement with the literature [106].

The rest of the results of this section are obtained with the TL implemented according to approach 4 and of size 7, while approach 5 will be used in Section 4.8.

Table (4.2) Comparison of Different TL Sizes Using Approach 4 (Example 1)

TL Size	5	7	10	20	30
Cost ($)	539496	538390	539374	539422	540215

4.5 NUMERICAL RESULTS OF THE STSA

Based on the STSA, a computer program has been implemented. The program offers different choices for finding a trial solution, e.g. completely random, semi random, and a mix of both with certain probability. The previously described approaches for TL have been implemented and compared.

In order to test the model the same examples, described in Chapter 3, are considered.

To illustrate the convergence trend in the TS method, a plot of the current and best solutions with the iteration number for Example 1 [29] is given in Fig. (4.4). As shown, the current solution has no trend, since the TS criteria is to accept any non-tabu solution regardless of its objective function value. Basically this is the main idea behind the ability of the TS method to escape local minima. On the other hand, the best solution is improving very fast at the beginning of the search while the improvement becomes slower at the end of the search.

Table (4.3) presents the comparison of results for Examples 1 and 2, solved by LR [29], IP [41], and the TSA. The results show the improvement achieved by the proposed STSA algorithm over both IP and LR results.

Tables (4.4), (4.5) and (4.6) show detailed results for Example 1 [29]. Table (4.4) presents the initial starting solution for the given results, which is randomly generated. It is obvious that this initial schedule is very far from the optimal one. The cost of this initial schedule is $615648.87, whereas that of the obtained final solution is $538390.

Table (4.5) shows the load sharing among the committed units in 24 hours. Table (4.6) gives the hourly load demand and the corresponding economic dispatch costs, start-up costs, and total operating cost.

In Example 2 [41], the optimal unit commitment is shown in Table (4.7) along with the load demand and the hourly operating costs. Table (4.8) gives the same results for Example 3 [62.63]. Since it is not clear to us the amount of reserve taken for this example in [62,63], we assumed a spinning reserve of 10%, and the corresponding total operating cost obtained is $662583.

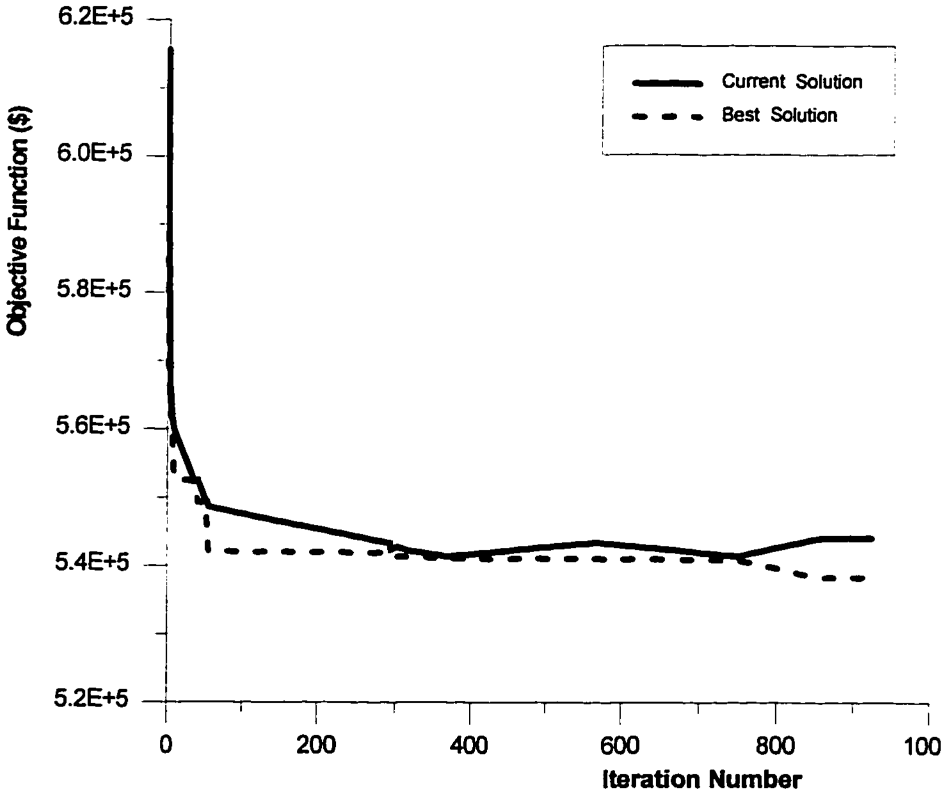


Fig. (4.4) The best and the current solutions versus iteration number for Example 1

Table (4.3) Comparison between the proposed STSA, LR and IP

	Example	LR [29]	IP [41]	Our STSA
Total Cost ($)	1	540895	-	538390
„	2	-	60667	59512
% Saving	1	0	-	0.46
„	2	-	0	1.9
No. Of Iterations	1	-	-	1924
„	2	-	-	616

Table (4.4) Power sharing (MW) of the initial schedule (Example 1)

HR	Unit Number									
	1	2		3	4		5	6	7	
1	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
2	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
3	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
4	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
5	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
6	300.00	130.00	165.00	130.00	225.00	50.00	250.00	110.00	275.00	75.00
7	300.00	232.97	165.00	130.00	225.00	130.63	250.00	186.40	275.00	75.00
8	300.00	395.67	165.00	163.06	225.00	181.18	250.00	338.65	296.14	85.30
9	300.00	400.00	201.49	304.43	225.00	200.00	250.00	375.00	461.86	132.22
10	300.00	400.00	288.35	389.55	225.00	200.00	250.00	375.00	561.62	160.47
11	311.82	400.00	331.91	420.00	225.00	200.00	250.00	375.00	611.64	174.64
12	343.28	400.00	356.09	420.00	225.00	200.00	258.72	375.00	639.41	182.50
13	303.20	400.00	325.28	420.00	225.00	200.00	250.00	375.00	604.03	172.48
14	409.01	400.00	406.62	420.00	225.00	200.00	315.43	375.00	0.00	198.94
15	317.58	400.00	336.34	420.00	225.00	200.00	250.00	375.00	0.00	176.08
16	300.00	400.00	271.70	373.24	225.00	200.00	250.00	375.00	0.00	155.06
17	329.97	400.00	345.86	420.00	225.00	200.00	250.00	375.00	0.00	179.18
18	495.78	400.00	473.32	420.00	225.00	200.00	390.27	375.00	0.00	220.63
19	530.48	400.00	500.00	420.00	225.00	200.00	420.21	375.00	0.00	229.31
20	391.66	400.00	393.28	420.00	225.00	200.00	300.46	375.00	0.00	194.60
21	300.00	400.00	261.62	363.38	225.00	200.00	0.00	375.00	0.00	0.00
22	300.00	357.61	165.00	130.00	225.00	169.36	0.00	303.03	0.00	0.00
23	300.00	201.80	165.00	130.00	225.00	120.95	0.00	157.25	0.00	0.00
24	300.00	130.00	165.00	130.00	225.00	90.00	0.00	110.00	0.00	0.00

Table (4.5) Power sharing of the final schedule (MW) of Example 1

HR	Unit Number *							
	2	3	4	6	7	8	9	10
1	400.0	0.00	0.00	185.0	0.00	350.2	0.00	89.70
2	395.3	0.00	0.00	181.0	0.00	338.3	0.00	85.19
3	355.3	0.00	0.00	168.6	0.00	300.9	0.00	75.00
4	333.1	0.00	0.00	161.7	0.00	280.1	0.00	75.00
5	400.0	0.00	0.00	185.0	0.00	350.2	0.00	89.70
6	400.0	270.3	0.00	200.0	0.00	375.0	0.00	154.6
7	400.0	383.5	420.0	200.0	0.00	375.0	0.00	191.4
8	400.0	295.5	396.6	200.0	0.00	375.0	569.9	162.8
9	400.0	468.0	420.0	200.0	0.00	375.0	768.0	218.9
10	400.0	444.6	420.0	200.0	358.0	375.0	741.0	211.2
11	400.0	486.3	420.0	200.0	404.8	375.0	788.9	224.8
12	400.0	514.1	420.0	200.0	436.0	375.0	820.8	233.9
13	400.0	479.3	420.0	200.0	397.0	375.0	780.9	222.6
14	400.0	388.9	420.0	200.0	295.6	375.0	677.1	193.2
15	400.0	310.0	410.8	200.0	250.0	375.0	586.5	167.5
16	400.0	266.6	368.2	200.0	250.0	375.0	536.6	153.4
17	400.0	317.3	417.9	200.0	250.0	375.0	594.8	169.8
18	400.0	458.5	420.0	200.0	373.6	375.0	757.0	215.8
19	400.0	486.3	420.0	200.0	404.8	375.0	788.9	224.8
20	400.0	375.0	420.0	200.0	280.0	375.0	661.2	188.6
21	400.0	0.00	404.8	200.0	0.00	375.0	579.5	165.5
22	400.0	0.00	0.00	200.0	0.00	375.0	524.9	150.0
23	396.4	0.00	0.00	181.4	0.00	339.3	297.1	85.58
24	377.6	0.00	0.00	175.5	0.00	321.7	275.0	0.00

*Units 1 and 5 are OFF all hours.

Table (4.6) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1,025.00	9,670.0	-	9,670.0
2	1,000.00	9,446.6	-	9,446.6
3	900.00	8,560.9	-	8,560.9
4	850.00	8,123.1	-	8,123.1
5	1,025.00	10,058.4	-	11,643.9
6	1,400.00	13,434.1	1,585.54	13,434.1
7	1,970.00	19,217.7	2,659.11	21,876.8
8	2,400.00	23,902.0	2,850.32	26,752.4
9	2,850.00	28,386.4	-	28,386.4
10	3,150.00	31,701.7	2,828.66	34,530.4
11	3,300.00	33,219.8	-	33,219.8
12	3,400.00	34,242.1	-	34,242.1
13	3,275.00	32,965.5	-	32,965.5
14	2,950.00	29,706.3	-	29,706.3
15	2,700.00	27,259.7	-	27,259.7
16	2,550.00	25,819.8	-	25,819.8
17	2,725.00	27,501.6	-	27,501.6
18	3,200.00	32,205.7	-	32,205.7
19	3,300.00	33,219.8	-	33,219.8
20	2,900.00	29,212.5	-	29,212.5
21	2,125.00	20,698.4	-	20,698.4
22	1,650.00	15,947.5	-	15,947.5
23	1,300.00	12,735.4	-	12,735.4
24	1,150.00	11,232.0	-	11,232.0

Total operating cost = $538390

Table (4.7) Load demand and UCT of Example 2

Hour	Load	Unit Number
1	1459.0	1 1 1 1 1 1 1 0 1 1
2	1372.0	1 1 1 1 1 1 1 0 1 1
3	1299.0	1 1 1 1 1 1 1 0 1 1
4	1285.0	1 1 1 1 1 1 1 0 1 1
5	1271.0	1 1 1 1 1 1 1 0 1 1
6	1314.0	1 1 1 1 1 1 1 0 1 1
7	1372.0	1 1 1 1 1 1 1 0 1 1
8	1314.0	1 1 1 1 1 1 1 0 1 1
9	1271.0	1 1 1 1 1 1 1 0 1 1
10	1242.0	1 1 1 1 1 0 1 0 1 1
11	1197.0	1 1 1 1 1 0 1 0 1 1
12	1182.0	1 1 1 1 1 0 1 0 1 1
13	1154.0	1 1 1 1 1 0 1 0 1 1
14	1138.0	1 1 1 1 1 0 1 0 1 1
15	1124.0	1 1 1 1 1 0 1 0 1 1
16	1095.0	1 1 1 1 1 0 1 0 1 1
17	1066.0	1 1 1 1 1 0 1 0 1 1
18	1037.0	1 1 1 1 1 0 1 0 1 1
19	993.0	1 1 1 1 1 0 1 0 1 1
20	978.0	1 1 1 1 1 0 1 0 1 1
21	963.0	1 1 1 1 1 0 1 0 1 1
22	1022.0	1 1 1 1 1 0 1 0 1 1
23	1081.0	1 1 1 1 1 0 1 0 1 1
24	1459.0	1 1 1 1 1 1 1 0 1 1

Table (4.8) Load demand and UCT of Example 3

		Unit Number **															
Hour	Load	1	2	3	4	5	10	14	15	16	17	18	19	20	21	22	23
1	1820.0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0
2	1800.0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0
3	1720.0	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0
4	1700.0	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0
5	1750.0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0
6	1910.0	1	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0
7	2050.0	0	0	0	1	0	1	1	0	0	1	1	1	1	0	1	0
8	2400.0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1
9	2600.0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
10	2600.0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
11	2620.0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
12	2580.0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
13	2590.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
14	2570.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
15	2500.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
16	2350.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
17	2390.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
18	2480.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
19	2580.0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
20	2620.0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
21	2600.0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
22	2480.0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
23	2150.0	1	0	0	1	0	1	0	1	1	1	1	1	1	1	0	0
24	1900.0	1	0	0	1	0	1	0	1	0	1	1	1	1	0	0	0

****Units 6,7,8,9 are OFF all hours. Units 11,12,13,24,25 and 26 are ON all hours.**

4.6 ADVANCED TABU SEARCH TECHNIQUES

Advanced TS procedures are recommended for sophisticated problems [105-116]. These procedures include in addition to the STM, the ITM, the LTM and the SO.

4.6.1 INTERMEDIATE TERM MEMORY

Intermediate Term Memory (ITM) function is employed within TS to achieve intensification in a specified region in the solution space at some periods of the search [105-107]. ITM operates by recording and comparing features of a selected number of best trial solutions generated during a particular period of the search. Features that are common to all or a compelling majority of these solutions are taken to be a regional attribute of good solutions. The method then seeks new solutions that exhibit these features by correspondingly restricting or penalizing available moves during a subsequent period of search. Different variants of implementation for the ITM are presented in Section 4.7.1.

4.6.2 LONG TERM MEMORY

Long Term Memory (LTM) function is used to perform global diversification of the search [105-107]. LTM function employs principles that are roughly the reverse of those for ITM. It guides the search to regions far from the best solutions examined earlier. Different implementations of LTM are introduced in Section 4.7.2.

4.6.3 STRATEGIC OSCILLATION

Strategic Oscillation (SO) is a major aspect of the proposed ATSA [105-107,111]. It allows the search to cross the feasible region in both directions to move the search into new regions and also to intensify the search in the neighbor of the bounds.

In the following section we describe the details of the proposed ATSA as applied to the UCP.

4.7 ADVANCED TABU SEARCH ALGORITHM FOR UNIT COMMITMENT (ATSA)

The proposed ATSA contains four major steps:

- First, applying STM procedures.
- Second, applying ITM procedures.
- Third, applying LTM procedures.
- Fourth, applying SO procedures.

Fig. (4.5) shows a flow chart for the proposed ATSA.

In the following subsections a description of the different components of the algorithm as applied to the UCP is presented. These components include the TL construction, the ITM, the LTM, and the SO implementations.

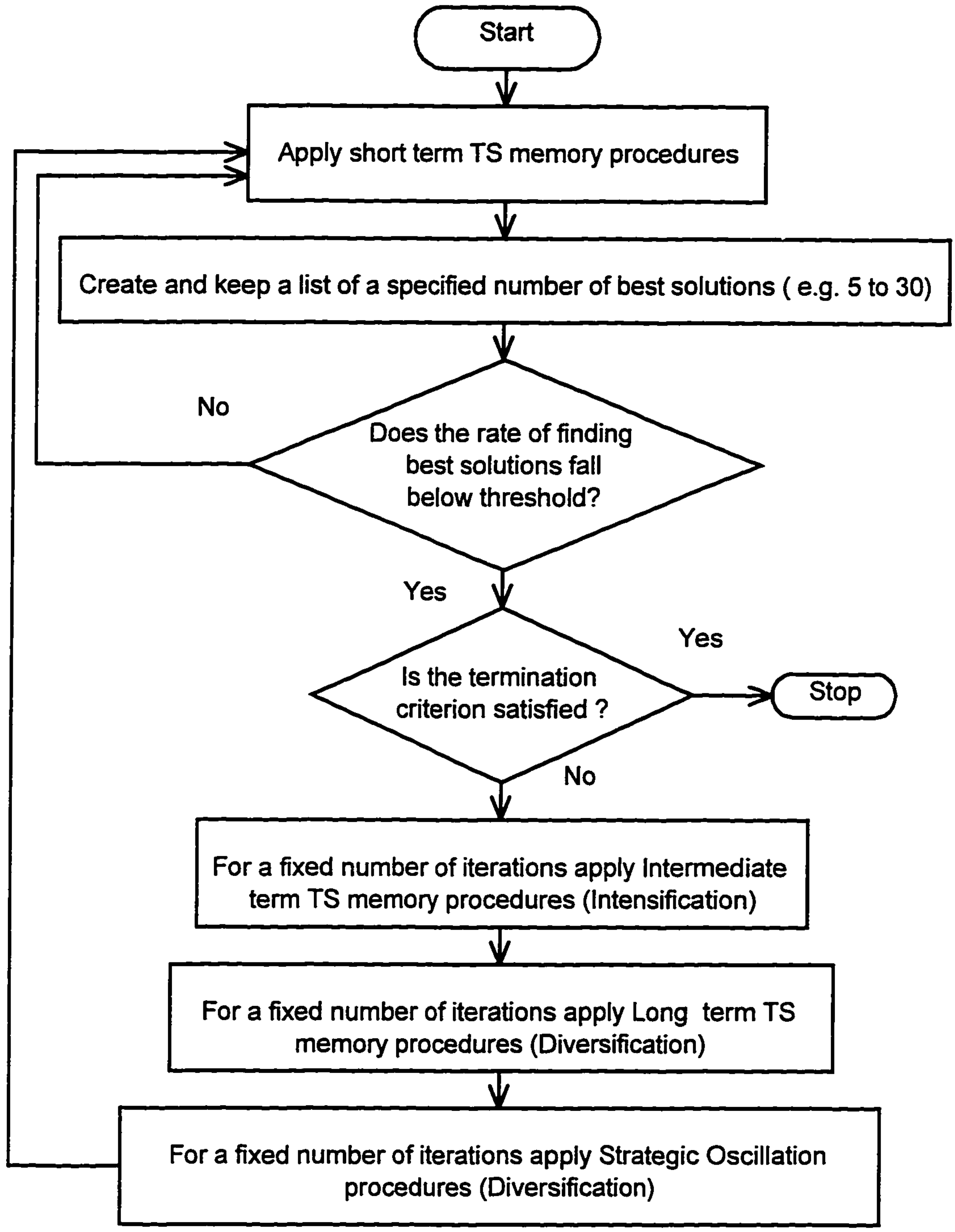


Fig.(4.5) Flow Chart of the Proposed ATSA for the UCP

4.7.1 INTERMEDIATE TERM MEMORY IMPLEMENTATION

The ITM is used to intensify the search in a specified region. Two different approaches are used to achieve this function

4.7.1.1 APPROACH (1)

In this approach the best K-solutions (K= 5 to 20) are recorded during the STM search. These best solutions are then compared to find the units that have the same schedule in a prespecified percentage of these K-solutions (in our implementation, it is taken to be 70%).These units are then included in an intermediate memory TL. At a particular period, according to the algorithm, the TL of the ITM is activated and the search is restricted to be in the neighborhood of the best solutions.

4.7.1.2 APPROACH (2)

In this approach the K-best solutions are recorded during the STM search. Then, the ITM procedure starts always with one of these best solutions at a time and performs a fixed number of iterations and repeats this for the K-solutions. The best solutions list is also updated during the search.

4.7.2 LONG TERM MEMORY IMPLEMENTATION

The LTM procedure is designed to drive the search into new regions. The LTM function is activated when a local minimum is reached during the STM search. In this procedure, the search is directed to points that are far from that of the recorded best solutions. This is achieved by activating the TL of the ITM and restricting the generated trial solutions to be far from the units that were tabu in the ITM procedures.

4.7.3 STRATEGIC OSCILLATION IMPLEMENTATION

In this procedure a specified number of moves are performed beyond the feasible boundary in a given direction before permitting a return to the feasible region. These number of moves could be changed each time this procedure is started.

4.8 NUMERICAL RESULTS OF THE ATSA

Considering the proposed ATSA, a computer program has been implemented. It has been concluded (in Section 4.4.1.6) that Approach 5 of constructing TL is the most efficient one since it requires less memory space with better solution quality. Accordingly, this Approach has been utilized in the implementation of the TL as part of the ATSA.

The three previously described examples from the literature [29,41,62,63], are solved.

Table (4.9) shows a comparison of daily operating costs for the proposed ATSA, and the STSA implemented in Section 4.3 for Examples 1,2, and 3.

Table (4.10) presents the comparison of results obtained in the literature for Examples 1 and 2, the STSA and the proposed ATSA.

From the last two tables, it is obvious that using the ATSA procedures improves the solution quality of the three examples. Both the required number of iterations and the solution cost are improved when using the advanced TS procedure. This emphasizes the effectiveness of using this approach, in addition to the STM, to solve difficult problems such as the UCP.

Tables (4.11) and (4.12) show detailed results for Example 1 [29]. Table (4.11) shows the load sharing among the committed units in the 24 hours. Table (4.12) gives the hourly load demand, and the corresponding economic dispatch costs, start-up costs, and total operating cost. For Example 3, Tables (4.13), (4.14) and (4.15) show detailed results and the corresponding total operating cost obtained is $660864.8750

Table (4.9) Comparison between the proposed ATSA and the STSA

	Example	STSA	ATSA	% Saving
Total Cost ($)	1	538390	537686	0.13
,,	2	59512	59385	0.21
,,	3	662583	660864	0.26
No. of iterations	1	1924	1235	35.8
,,	2	616	138	77.5
,,	3	3900	2547	34.6

Table (4.10) Comparison between the ATSA, LR, IP and the STSA

	Example	LR [29]	IP [41]	STSA	ATSA
Total Cost ($)	1	540895	-	538390	537686
,,	2	-	60667	59512	59385
% Saving	1	0	-	0.46	0.59
,,	2	-	0	1.9	2.1

Table (4.11) Power sharing (MW) of Example 1

Hour	Unit Number**							
	2	3	4	6	7	8	9	10
1	400.00	0.00	0.00	185.04	0.00	350.26	0.00	89.70
2	395.36	0.00	0.00	181.09	0.00	338.36	0.00	85.19
3	355.38	0.00	0.00	168.67	0.00	300.95	0.00	75.00
4	333.13	0.00	0.00	161.75	0.00	280.12	0.00	75.00
5	400.00	0.00	0.00	185.04	0.00	350.26	0.00	89.70
6	400.00	0.00	295.68	200.00	0.00	375.00	0.00	129.32
7	400.00	0.00	342.97	200.00	0.00	375.00	507.02	145.01
8	400.00	295.59	396.65	200.00	0.00	375.00	569.93	162.83
9	400.00	468.07	420.00	200.00	0.00	375.00	768.01	218.92
10	400.00	444.60	420.00	200.00	358.05	375.00	741.06	211.29
11	400.00	486.30	420.00	200.00	404.89	375.00	788.95	224.86
12	400.00	514.11	420.00	200.00	436.09	375.00	820.89	233.91
13	400.00	479.35	420.00	200.00	397.09	375.00	780.96	222.60
14	400.00	388.98	420.00	200.00	295.63	375.00	677.18	193.20
15	400.00	310.07	410.84	200.00	250.00	375.00	586.56	167.54
16	400.00	266.64	368.27	200.00	250.00	375.00	536.68	153.41
17	400.00	317.31	417.93	200.00	250.00	375.00	594.87	169.89
18	400.00	458.51	420.00	200.00	373.65	375.00	757.03	215.81
19	400.00	486.30	420.00	200.00	404.89	375.00	788.95	224.86
20	400.00	375.08	420.00	200.00	280.03	375.00	661.21	188.68
21	400.00	0.00	404.87	200.00	0.00	375.00	579.57	165.56
22	400.00	0.00	0.00	200.00	0.00	375.00	675.00	0.00
23	400.00	0.00	0.00	191.64	0.00	370.14	338.22	0.00
24	377.64	0.00	0.00	175.58	0.00	321.78	275.00	0.00

** Units 1 and 5 are OFF at all hours.

Table (4.12) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1,025	9,670.0	-	9,670.0
2	1,000	9,446.6	-	9,446.6
3	900	8,560.9	-	8,560.9
4	850	8,123.1	-	8,123.1
5	1,025	9,670.0	-	9,670.0
6	1,400	13,434.1	1,706.0	15,140.0
7	1,970	19,217.7	2,659.1	21,876.8
8	2,400	23,815.5	2,685.1	26,500.6
9	2,850	28,253.9	-	28,253.9
10	3,150	31,701.7	3,007.6	34,709.3
11	3,300	33,219.8	-	33,219.8
12	3,400	34,242.1	-	34,242.1
13	3,275	32,965.5	-	32,965.5
14	2,950	29,706.3	-	29,706.3
15	2,700	27,259.7	-	27,259.7
16	2,550	25,819.8	-	25,819.8
17	2,725	27,501.6	-	27,501.6
18	3,200	32,205.7	-	32,205.7
19	3,300	33,219.8	-	33,219.8
20	2,900	29,212.5	-	29,212.5
21	2,125	20,698.4	-	20,698.4
22	1,650	15,878.2	-	15,878.2
23	1,300	12,572.8	-	12,572.8
24	1,150	11,232.0	-	11,232.0

Total operating cost = $537686

Table (4.13) Power sharing (MW) of Example 3 (units 1-13)

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	76.00	54.80	15.20
2	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	76.00	50.00
3	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	28.40	15.20
4	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	15.20	15.20
5	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	58.40	15.20
6	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	66.40	15.20
7	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
8	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
9	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
10	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
11	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
12	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
13	0.00	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
14	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	64.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20	15.20

Table (4.14) Power sharing (MW) of Example 3 (units 14-26)

HR	Unit Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	343.2	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	0.00	0.00	69.60	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
8	0.00	100.0	100.0	155.0	155.0	155.0	155.0	150.6	68.95	0.00	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	122.6	0.00	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	122.6	0.00	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	142.6	0.00	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	102.6	0.00	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	112.6	0.00	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	96.60	0.00	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	157.0	68.95	0.00	350.0	350.0	350.0
16	100.0	100.0	38.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
17	100.0	100.0	78.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
18	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	129.0	0.00	350.0	350.0	350.0
22	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
23	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
24	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	331.3	350.0	350.0

Table (4.15) Load demand and hourly costs ($) of Example 3

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.79E+04	0.00E+00	1.79E+04
2	1.80E+03	1.76E+04	0.00E+00	1.76E+04
3	1.72E+03	1.66E+04	0.00E+00	1.66E+04
4	1.70E+03	1.63E+04	0.00E+00	1.63E+04
5	1.75E+03	1.70E+04	0.00E+00	1.70E+04
6	1.91E+03	1.93E+04	1.60E+02	1.94E+04
7	2.05E+03	2.17E+04	1.00E+02	2.18E+04
8	2.40E+03	2.98E+04	7.00E+02	3.05E+04
9	2.60E+03	3.42E+04	1.00E+02	3.43E+04
10	2.60E+03	3.42E+04	0.00E+00	3.42E+04
11	2.62E+03	3.46E+04	0.00E+00	3.46E+04
12	2.58E+03	3.37E+04	0.00E+00	3.37E+04
13	2.59E+03	3.39E+04	0.00E+00	3.39E+04
14	2.57E+03	3.33E+04	0.00E+00	3.33E+04
15	2.50E+03	3.17E+04	0.00E+00	3.17E+04
16	2.35E+03	2.85E+04	0.00E+00	2.85E+04
17	2.39E+03	2.92E+04	0.00E+00	2.92E+04
18	2.48E+03	3.12E+04	0.00E+00	3.12E+04
19	2.58E+03	3.35E+04	0.00E+00	3.35E+04
20	2.62E+03	3.44E+04	0.00E+00	3.44E+04
21	2.60E+03	3.40E+04	0.00E+00	3.40E+04
22	2.48E+03	3.12E+04	0.00E+00	3.12E+04
23	2.15E+03	2.47E+04	0.00E+00	2.47E+04
24	1.90E+03	2.14E+04	0.00E+00	2.14E+04

Total operating cost = 660864.8750

4.9 SUMMARY

In this chapter, the application of the TS method for the UCP is introduced for the first time. Two new algorithms for the UCP are proposed and tested. The first algorithm uses the STM procedure of the TS method, while the second algorithm is based on advanced TS procedures. Different criteria for constructing the TL restrictions for the UCP are implemented and compared. Several examples are solved to test the proposed algorithms.

The computational results of the two algorithms along with a comparison with previously published works are presented. The results showed that the algorithm based on the STM outperforms the results reported in the literature. On the other hand, both the required number of iterations and the solution cost are improved when using the advanced TS procedure in the second algorithm. This emphasizes the effectiveness of using this approach, along with the STM, to solve difficult problems such as the UCP.

In the next chapter, a genetic-based algorithm for the UCP will be introduced.

CHAPTER FIVE

A NEW GENETIC ALGORITHM APPROACH FOR UNIT COMMITMENT

5.1 INTRODUCTION

Genetic algorithms(GAs) have been developed by John Holland, his colleagues, and his students at the University of Michigan in the early 1970's [117]. GAs have become increasingly popular in recent years in science and engineering disciplines [117-125]. GAs have been quite successfully applied to optimization problems like wire routing, scheduling, adaptive control, game playing, cognitive modeling, transportation problems, traveling salesman problems, optimal control problems, etc.

In this chapter a new implementation of a Genetic Algorithm (GA) to the UCP is proposed. Several examples are solved to test the proposed algorithm.

In the next section, an overview of the GA method is presented, followed in Section 5.3, by the new proposed implementation of the GA as applied to solve the UCP along with the description of different GA components. Section 5.4 presents the detailed description of a local search algorithm that has been used with the GA. In Section 5.5

the computational results along with a comparison with previously published work are presented.

5.2 THE GENETIC ALGORITHM APPROACH

5.2.1 OVERVIEW

GAs are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) in the form of encoded information individuals that evolve in time. A GA for a particular problem must have the following five components [119]:

- A genetic representation for potential solution to the problem,
- A way to create an initial population of potential solutions,
- An evaluation function that plays the role of the environment, rating solutions in terms of their "fitness",
- Genetic operators that alter the composition of children,
- Values for various parameters that the GA uses (population size, probabilities of applying genetic operators, etc.).

A genetic search starts with a randomly generated initial population within which each individual is evaluated by means of a fitness function. Individuals in this and subsequent generations are duplicated or eliminated according to their fitness values. Further generations are created by applying GA operators. This eventually leads to a generation of high performing individuals [123].

5.2.2 SOLUTION CODING

GAs require the natural parameters set of the optimization problem to be coded as a finite-length string over some finite alphabet. Coding is the most important point in applying the GA to solve any optimization problem. Coding could be in a real or a binary form. Coded strings of solutions are called “chromosomes”. A group of these solutions (chromosomes) are called population. Our proposed new method of coding is presented in Section 5.3.2.

5.2.3 FITNESS FUNCTION

The fitness function is the second important issue in solving optimization problems using GAs. It is often necessary to map the underlying natural objective function to a fitness function through one or more mappings. The first mapping is done to transform the objective function into a maximization problem rather than minimization to suit the GA concepts of selecting the fittest chromosome which has the highest objective function.

A second important mapping is the scaling of the fitness function values. Scaling is an important step during the search procedures of the GA. This is done to keep appropriate levels of competition throughout a simulation. Without scaling, early on there is a tendency for a few superindividuals to dominate the selection process. Later on, when the population has largely converged, competition among population members is less strong and simulation tends to wander. Thus, Scaling is a useful process to prevent both the premature convergence of the algorithm and the random improvement that may occur in the late iterations of the algorithm. There are many methods for scaling such as linear, sigma truncation, and power law scaling [117]. Linear scaling is the most commonly used and will be discussed in details in Section 5.3.3. In the sigma truncation method, population variance information to preprocess raw fitness values prior to scaling is used. It is called sigma (σ) truncation because of the use of population standard deviation information, a constant is subtracted from raw fitness values as follows:

$$f' = f - (f' - c.\sigma) \qquad (5.1)$$

In equation (5.1) the constant c is chosen as a reasonable multiple of the population standard deviation and negative results ($f' < 0$) are arbitrarily set to 0. Following sigma truncation, fitness scaling can proceed as described without the danger of negative results.

5.2.4 GENETIC ALGORITHMS OPERATORS

There are usually three operators in a typical GA [123]. The first is the production operator which makes one or more copies of any individual that posses a high fitness value; otherwise, the individual is eliminated from the solution pool.

The second operator is the recombination (also known as the "crossover") operator. This operator selects two individuals within the generation and a crossover site and performs a swapping operation of the string bits to the right hand side of the crossover site of both individuals. The crossover operator serves two complementary search functions. First, it provides new points for further testing within the hyperplanes already represented in the population. Second, crossover introduces representatives of new hyperpalnes into the population, which is not represented by either parent structure. Thus, the probability of a better performing offspring is greatly enhanced.

The third operator is the "mutation" operator. This operator acts as a background operator and is used to explore some of the unvisited points in the search space by randomly flipping a "bit" in a population of strings. Since frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its activation.

5.2.5 CONSTRAINTS HANDLING (REPAIR MECHANISM)

Constraints handling techniques for the GAs can be grouped into a few categories [119]. One way is to generate a solution without considering the constraints but to include them with penalty factors in the fitness function. This method has been used previously [68-73].

Another category is based on the application of a special repair algorithm to correct any infeasible solution so generated.

The third approach concentrates on the use of special representation mappings (decoders) which guarantee (or at least increase the probability of) the generation of a feasible solution or the use of problem-specific operators which preserve feasibility of the solutions.

In our implementation, we are generating always solutions that are satisfying the minimum up/down constraints. However, due to applying the crossover and mutation operations the load demand and/or the reserve constraints might be violated. A mechanism to restore the feasibility is applied by committing randomly more units at the violated time periods and keeping the feasibility of the minimum up/down time constraints.

5.2.6 A GENERAL GENETIC ALGORITHM

In applying the GAs to optimization problems, certain steps for simulating the evolution must be performed. These are described as follows [118]:

Step (1): Initialize a population of chromosomes.

Step (2): Evaluate each chromosomes in the population.

Step (3): Create new chromosomes by mating current chromosomes; apply mutation and recombination as the parent chromosomes mate.

Step (4): Delete members of the population to make room for the new chromosomes.

Step (5): Evaluate the new chromosomes and insert them into the population.

Step (6): If the termination criterion is satisfied, stop and return the best chromosomes; otherwise, go to Step (3).

5.3 THE PROPOSED NEW IMPLEMENTATION OF A GENETIC ALGORITHM FOR THE UCP

5.3.1 OVERVIEW

The proposed new GA implementation for the UCP differs from other GA implementations in three respects[75]. First, the UCP solution is coded using a mix between binary and decimal representations, thus saving computer memory as well as computation time of the GA search procedure. Second, the fitness function is based only on the total operating cost and no penalties are included. Third, to improve the fine local tuning capabilities of the proposed GA, a special mutation operator based on a local search procedure, is designed.

The proposed algorithm involves four major Steps[75]:

- Creating an initial population by randomly generating a set of feasible solutions (chromosomes), using the rules presented in Section 2.4.
- Evaluating each chromosome by solving the economic dispatch problem, using the algorithm described in Section 2.6.
- Determining the fitness function for each chromosome in the population.
- Applying GA operators to generate new populations as follows:
    - Copy the best solution from the current to the new population
    - Generate new members (typically 1-10% of the population size), as neighbors to solutions in the current population, and add them to the new population.
    - Apply the crossover operator to complete the members of the new population.
    - Apply the mutation operator to the new population.

The flow chart of Fig.(5.1) shows the main steps of the proposed algorithm.

In the following sections, the implementations of the different components of the proposed algorithm are presented.

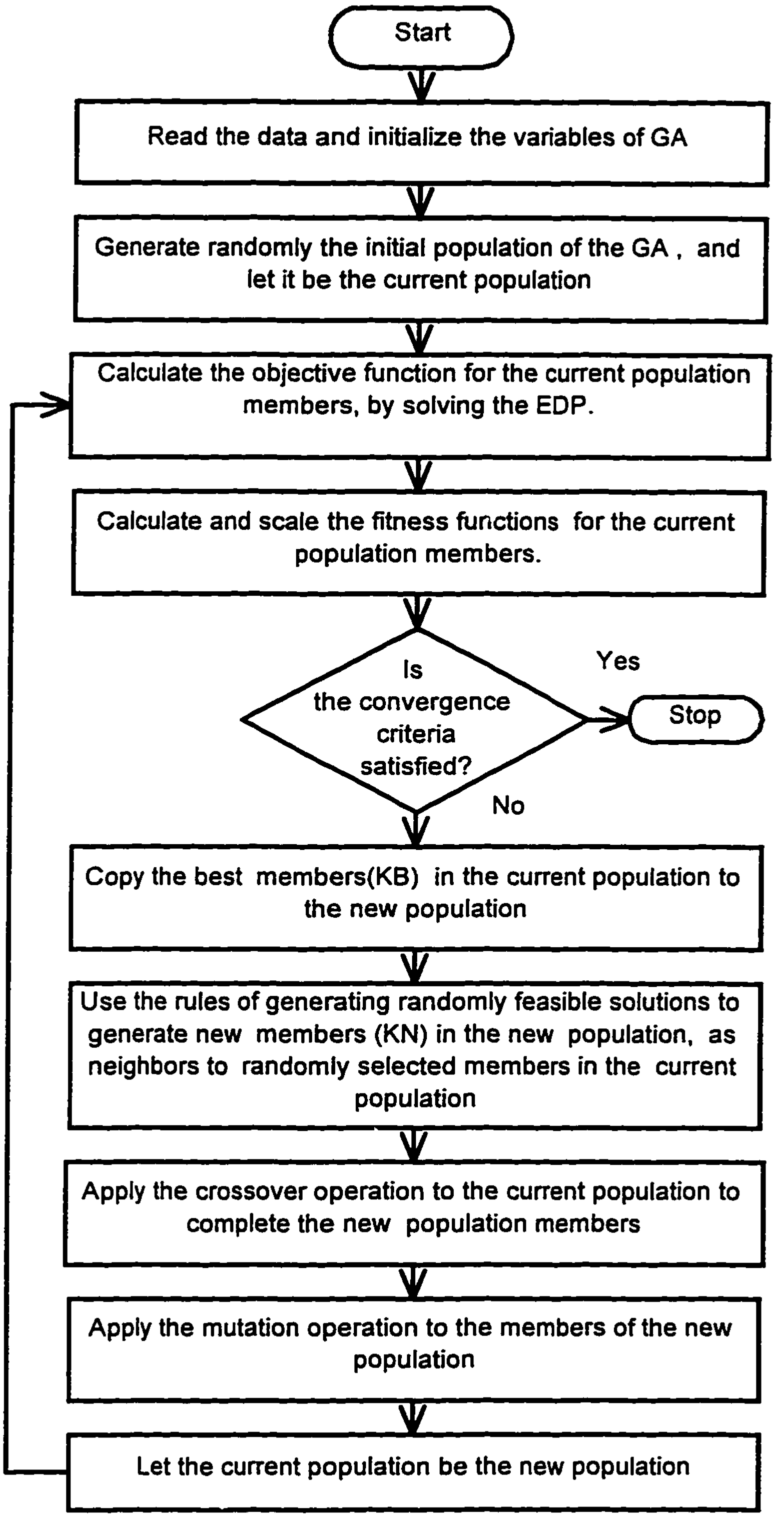


Fig.(5.1) Flow Chart of the proposed GA for UCP

5.3.2 SOLUTION CODING

Since the UCP lends itself to the binary coding in which a zero denotes the OFF state and a one represents the ON state, all published works used the binary coding [68-73]. The UCP solution is represented by a binary matrix (U) of dimension TxN (Fig.(5.2-a)). A candidate solution in the GA could then be represented by a string whose length is the product of the scheduling periods and the number of generating units TxN. In the GA a number of these solutions, equal to the population size (NPOP), is stored. The required storage size is then equal to NPOPxTxN which is a large value even for a moderate size system.

The new proposed method for coding is based on a mix between a binary number and its equivalent decimal number. Each column vector of length T in the solution matrix (which represents the operation schedule of one unit) is converted to its equivalent decimal number. The solution matrix is then converted into one row vector (chromosome) of N decimal numbers, (U1,U2,....UN); each represents the schedule of one unit as shown in Fig.(5.2-b). Typically the numbers U1,U2, ..,UN are integers ranging between 0 and $2^N - 1$. Accordingly, a population of size NPOP can be stored in a matrix of dimension NPOPxN as arbitrarily shown in Fig.(5.2-c). Hence, the proposed method requires only 1/T of the storage required if a normal binary coding is used.

HR	Unit Number							
	1	2	3	4	.	.	.	N
1	1	1	0	0	.	.	.	1
2	1	1	0	0	.	.	.	1
3	1	0	1	0	.	.	.	0
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
T	0	1	0	1	.	.	.	0

Fig.(5.2-a) The binary solution matrix U

U1	U2	U3	U4	.	.	.	UN

Fig.(5.2-b) The equivalent decimal vector (1xN)
(one chromosome)

23	14	45	56	.	.	.	62
34	52	72	18	.	.	.	91
.	.	.	.	.	.	.	.
51	36	46	87	.	.	.	21

Fig.(5.2-c) Population of size NPOPxN
(NPOP chromosomes)

5.3.3 FITNESS FUNCTION

Unlike the previous solutions of the UCP using GA [68-73], the fitness function is taken as the reciprocal of the total operating cost in (2.1), since we are always generating feasible solutions.

The fitness function is then scaled to prevent the premature convergence. Linear scaling is used. This requires a linear relationship between the original fitness function (f) and the scaled one (f_S) as follows [117]:

$$f_s = af + b \tag{5.2}$$

$$a = (c-1)f_{av} / (f_{max} - f_{min}) \tag{5.3}$$

$$b = (1-a)f_{av} \tag{5.4}$$

where: c: is a parameter between 1.2 and 2,

f_{max}, f_{min}, f_{av}: are maximum, minimum and average values of the original fitness functions respectively.

5.3.4 SELECTION

The selection of chromosomes for applying various GA operators is based on their scaled fitness function in accordance to the roulette wheel selection rule. The roulette wheel slots are sized according to the accumulated probabilities of reproducing each chromosome.

5.3.5 CROSSOVER

To speed up the calculations, the crossover operation is performed between two chromosomes in their decimal form. A two points crossover operation is used. The following steps are applied to perform the crossover operation:

- Select two parents according to the roulette wheel rule.
- Select randomly two positions in the two chromosomes.
- Exchange the bits between the two selected positions in the two parents to produce two children (Fig.(5.3)).
- Decode the two children into their binary equivalent and check for reserve constraints violation.

- If the reserve constraints are not satisfied apply the repair mechanism (described in Section 5.2.5) to restore feasibility of the produced children.

12	34	**45**	**62**	**93**	72	82	32
52	81	**69**	**55**	**26**	38	57	76

Two Parents

12	34	**69**	**55**	**26**	72	82	32
52	81	**45**	**62**	**93**	38	57	76

Two children

Fig.(5.3) Two points cross over example

5.3.6 MUTATION

The crossover operation explained in the last section is not enough for creating a completely new solution. The reason is that it exchanges the schedule of units as black boxes among different chromosomes without applying any changes in the schedules of the units themselves.

Two new types of mutation operators are introduced to create changes in the units schedules. The mutation operation is applied after reproducing all the new population members. It is done by applying the probability test to the members of the new population one by one. The mutation operation is then applied to the selected chromosome. The details of the two mutation operators are described in the following sections.

5.3.6.1 MUTATION OPERATOR (1)

The first mutation operator is implemented as follows:

1. Select a chromosome as explained before and decode it into its binary equivalent.
2. Pick randomly a unit number and a time period.
3. Apply the rules in Section 2.4 to reverse the status of this unit keeping the feasibility of the unit constraints related to its minimum up/down times.
4. For the changed time periods, check the reserve constraints.
5. If the reserve constraints are violated, apply the proposed correction mechanism and go to the next step, otherwise go to the next step.
6. Decode the modified solution matrix from binary to decimal form and update the new population.

5.3.6.2 MUTATION OPERATOR (2)

The second mutation operator is based on a local search algorithm to perform fine tuning on some of the chromosomes in the new generated population. The selection of chromosomes for applying this type of mutation could be random or based on the roulette wheel method.

The local search algorithm steps are described in details as follows:

1. Decode the selected chromosome into its binary form.
2. Sort the time periods in a descending order according to the difference between the committed units capacity and the load demand.

3. Identify the time periods at which the committed units capacity is greater than 10% above the load plus the desired reserve. These time periods have a surplus of committed power capacity.
4. At the time periods of surplus capacity, sort the committed units in an ascending order according to their percentage loading.
5. Identify the units that have a percentage loading less than 20% above their minimum output limits. These units are the costlier units among the committed units in the respective time periods, since they are lightly loaded.
6. Take the time periods, according to their order found in (2) and consider switching off the underloaded units one at a time, according to their order.
7. Check the feasibility of the solution obtained . If it is feasible, go to Step (8), otherwise go to Step (6).
8. Calculate the objective function of the solution obtained by solving the economic dispatch problem for the changed time periods.
9. Decode the new solution obtained to its decimal equivalent and replace the old one in the new population.

5.3.7 ADAPTIVE GA OPERATORS

The search for the optimal GA parameters setting is a very complex task. To achieve good performance of the GA, an adaptive scheme to control the probability rate of performing the crossover and mutation operators is designed.

The crossover rate controls the frequency with which the crossover operator is applied. The higher the crossover rate, the more quickly new structures are introduced into the population. If the crossover rate is too high, high-performance structures are discarded faster than selection can produce improvements. If the crossover rate is too low, the search may stagnate due to the lower exploration rate. In our implementation, the crossover rate is initialized with a high value (typically between 0.6 and 0.8) and is then decreased during the search according to the convergence rate of the algorithm (decrement value is 0.01).

Mutation is a secondary search operator which increases the variability of the population. A low level of mutation serves to prevent any given bit position from remaining forever converged to a single value in the entire population, and consequently increases the probability of entrapment at local minima. A high level of mutation yields an essentially random search, which may lead to very slow convergence. To guide the search, the mutation rate starts at a low value (between 0.2 and 0.5) then it is incremented by 0.01 as the algorithm likely converged to a local minimum.

5.4 NUMERICAL EXAMPLES

In order to test the proposed algorithm, three systems are considered.

Preliminary experiments have been performed on the three systems to find the most suitable GA parameters settings. The following control parameters have been chosen after running a number of simulations: population size=50, initial value of crossover rate=0.8, decrement value of crossover=0.01, initial value of mutation rate=0.2, increment value of mutation=0.01, local search mutation rate=0.1, elite copies=2, and the maximum number of generations=1000.

Different experiments were carried out to investigate the effect of the local search mutation on the results. It was found that the proposed algorithm with local search performs better than the simple GA without local search , in terms of both solution quality and number of iterations.

Table(5.1) presents the comparison of results obtained in the literature (LR and IP) for Examples 1 and 2.

Fig.(5.4) shows progress in the best objective function versus the generation number. The algorithm converges after about 400 generations, which is relatively fast.

Tables (5.2), (5.3) and (5.4) show detailed results for Example 1 [29]. Table (5.2) shows the load sharing among the committed units in the 24 hours. Table (5.3) gives the hourly load demand and the corresponding economic dispatch costs, start-up costs, and total operating cost. Table (5.4) presents the final schedule of the 24 hours, given in Table (5.2), in the form of its equivalent decimal numbers.

Tables (5.5), (5.6), (5.7) and (5.8) also present the detailed results for Example 3 with a total operating cost of $661439.8. Comparison of the results for this example with other methods is presented in Chapter (7).

Table (5.1) Comparison between LR, IP, and the proposed GA

	Example	LR [29]	IP [41]	GA
Total Cost ($)	1	540895	-	537372
,,	2	-	60667	59491
% Saving	1	-		0.65
,,	2	-		1.93
Generations No	1	-		411
,,	2	-		393

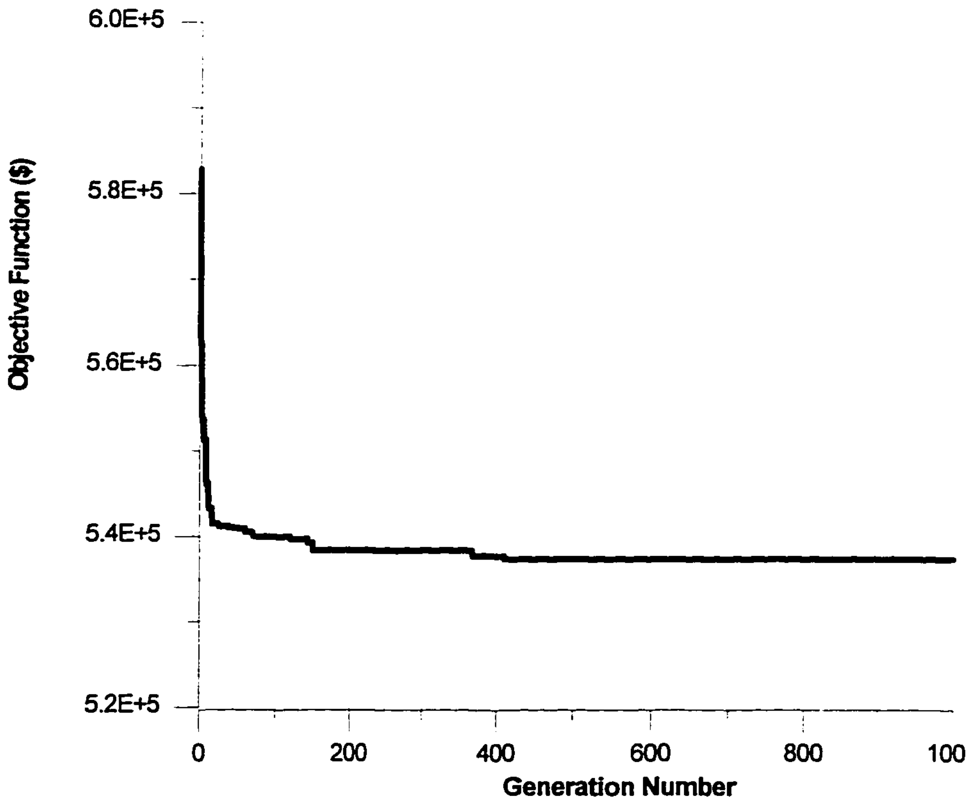


Fig.(5.4) Convergence of the proposed GA

Table (5.2) Power sharing (MW) of Example 1

HR	Unit Numbr**							
	2	3	4	6	7	8	9	10
1	400.0	0.0	0.0	185.0	0.0	350.3	0.0	89.7
2	395.4	0.0	0.0	181.1	0.0	338.4	0.0	85.2
3	355.4	0.0	0.0	168.7	0.0	301.0	0.0	75.0
4	333.1	0.0	0.0	161.8	0.0	280.1	0.0	75.0
5	400.0	0.0	0.0	185.0	0.0	350.3	0.0	89.7
6	400.0	0.0	295.7	200.0	0.0	375.0	0.0	129.3
7	400.0	0.0	343.0	200.0	0.0	375.0	507.0	145.0
8	400.0	295.6	396.7	200.0	0.0	375.0	569.9	162.8
9	400.0	468.1	420.0	200.0	0.0	375.0	768.0	218.9
10	400.0	444.6	420.0	200.0	358.1	375.0	741.1	211.3
11	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
12	400.0	514.1	420.0	200.0	436.1	375.0	820.9	233.9
13	400.0	479.4	420.0	200.0	397.1	375.0	781.0	222.6
14	400.0	389.0	420.0	200.0	295.6	375.0	677.2	193.2
15	400.0	310.1	410.8	200.0	250.0	375.0	586.6	167.5
16	400.0	266.6	368.3	200.0	250.0	375.0	536.7	153.4
17	400.0	317.3	417.9	200.0	250.0	375.0	594.9	169.9
18	400.0	458.5	420.0	200.0	373.7	375.0	757.0	215.8
19	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
20	400.0	0.0	420.0	200.0	442.2	375.0	827.2	235.7
21	400.0	0.0	404.9	200.0	0.0	375.0	579.6	165.6
22	400.0	0.0	0.0	200.0	0.0	375.0	675.0	0.0
23	400.0	0.0	0.0	191.6	0.0	370.1	338.2	0.0
24	377.6	0.0	0.0	175.6	0.0	321.8	275.0	0.0

**Units 1,5 are OFF all hours.

Table (5.3) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1025	9670.04	0.00	9670.04
2	1000	9446.62	0.00	9446.62
3	900	8560.91	0.00	8560.91
4	850	8123.13	0.00	8123.13
5	1025	9670.04	0.00	9670.04
6	1400	13434.10	1705.97	15140.00
7	1970	19217.70	2659.11	21876.80
8	2400	23815.50	2685.07	26500.60
9	2850	28253.90	0.00	28253.90
10	3150	31701.70	3007.58	34709.30
11	3300	33219.80	0.00	33219.80
12	3400	34242.10	0.00	34242.10
13	3275	32965.50	0.00	32965.50
14	2950	29706.30	0.00	29706.30
15	2700	27259.70	0.00	27259.70
16	2550	25819.80	0.00	25819.80
17	2725	27501.60	0.00	27501.60
18	3200	32205.70	0.00	32205.70
19	3300	33219.80	0.00	33219.80
20	2900	28899.00	0.00	28899.00
21	2125	20698.40	0.00	20698.40
22	1650	15878.20	0.00	15878.20
23	1300	12572.80	0.00	12572.80
24	1150	11232.00	0.00	11232.00

Total operating cost = $537371.94

Table (5.4) The UCT of Example 1 in its equivalent decimal form (best chromosome)

Unit Number				
1,6	2,7	3,8	4,9	5,10
0	16777215	524160	2097120	0
16777215	1048064	16777215	16777152	2097151

Table (5.5) Power sharing (MW) of Example 3 (units 1-13)

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	76.00	54.80	15.20
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	50.00
3	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	28.40	15.20
4	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	15.20	15.20
5	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	58.40	15.20
6	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	66.40	15.20
7	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
8	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
9	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
10	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
11	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
12	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
13	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
14	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	64.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20	15.20

Table(5.6) Power sharing (MW) of Example 3 (units 14-26)

HR	Unit Number											
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	343.2	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	0.00	0.00	69.60	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
8	0.00	100.0	100.0	155.0	155.0	155.0	155.6	150.6	68.95	0.00	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	122.6	0.00	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	122.6	0.00	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	142.6	0.00	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	102.6	0.00	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	112.6	0.00	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	92.60	0.00	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	153.0	68.95	0.00	350.0	350.0	350.0
16	100.0	100.0	34.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
17	100.0	100.0	78.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
18	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	129.0	0.00	350.0	350.0	350.0
22	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
23	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
24	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	331.3	350.0	350.0

Table (5.7) Load demand and hourly costs ($) of Example 3

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.79E+04	0.00E+00	1.79E+04
2	1.80E+03	1.76E+04	0.00E+00	1.76E+04
3	1.72E+03	1.66E+04	0.00E+00	1.66E+04
4	1.70E+03	1.63E+04	0.00E+00	1.63E+04
5	1.75E+03	1.70E+04	0.00E+00	1.70E+04
6	1.91E+03	1.93E+04	1.60E+02	1.94E+04
7	2.05E+03	2.17E+04	1.00E+02	2.18E+04
8	2.40E+03	2.98E+04	7.00E+02	3.05E+04
9	2.60E+03	3.42E+04	1.00E+02	3.43E+04
10	2.60E+03	3.42E+04	0.00E+00	3.42E+04
11	2.62E+03	3.46E+04	0.00E+00	3.46E+04
12	2.58E+03	3.37E+04	0.00E+00	3.37E+04
13	2.59E+03	3.39E+04	0.00E+00	3.39E+04
14	2.57E+03	3.35E+04	0.00E+00	3.35E+04
15	2.50E+03	3.18E+04	0.00E+00	3.18E+04
16	2.35E+03	2.87E+04	0.00E+00	2.87E+04
17	2.39E+03	2.92E+04	0.00E+00	2.92E+04
18	2.48E+03	3.12E+04	0.00E+00	3.12E+04
19	2.58E+03	3.35E+04	0.00E+00	3.35E+04
20	2.62E+03	3.44E+04	0.00E+00	3.44E+04
21	2.60E+03	3.40E+04	0.00E+00	3.40E+04
22	2.48E+03	3.12E+04	0.00E+00	3.12E+04
23	2.15E+03	2.47E+04	0.00E+00	2.47E+04
24	1.90E+03	2.14E+04	0.00E+00	2.14E+04

Total operating cost = 661439.8125

Table (5.8) The UCT of Example 3 in its equivalent decimal form(best chromosome)

Unit Number						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26		
60	0	16320	0	0	0	0
0	65535	16777184	16777185	16777215	16777215	4194048
16777088	16777152	16777215	16777215	16777215	16777215	16777088
16777088	0	16777215	16777215	16777215		

5.5 SUMMARY

In this chapter, a new implementation of a GA to solve the UCP is proposed.

The proposed new GA implementation [75] for the UCP differs from other GA implementations in three respects. First, the UCP solution is coded using a mix between binary and decimal representations. Second, the fitness function is based only on the total operating cost and no penalties are included. Third, to improve the fine local tuning capabilities of the proposed GA, a special mutation operator based on a local search procedure, is designed. The detailed description of local search algorithm that has been used with the GA are also presented.

The computational results along with a comparison with previously published work showed the effectiveness of the proposed new approach in saving cost and computer memory.

In the next Chapter, we introduce three new hybrids algorithms for the UCP.

CHAPTER SIX

NEW HYBRID ALGORITHMS FOR UNIT COMMITMENT

6.1 INTRODUCTION

In the last three chapters, we have proposed three different algorithms, based on SA, TS, and GA methods, to solve the UCP. The effectiveness of the three methods to solve the UCP has been proved. The main features of these methods were also investigated. These methods, of course, have their own merits and drawbacks.

SA is a Markov chain Monte Carlo method, and therefore, it is memoryless. Hence, the main problem is that SA will continue to jump up and down without noticing that the movement is confined.

TS is characterized as a memory-based method, and therefore learning is achieved during the search process.

GA is a global stochastic search method. The fine tuning capability when approaching a local minimum is weak.

The competitive performance of the combinatorial optimization algorithms is still an open issue. Recently, hybrid methods have come to the picture capturing the merits of different methods and exploring them in a form of hybrid scheme. It is often proved that a hybrid scheme of some methods outperforms the performance of these methods as individuals.

In this chapter we propose three different new hybrid algorithms for the UCP. The proposed hybrid algorithms integrate the use of the previously introduced algorithms, SA, TS, and GA. The bases of the hybridization of these algorithms are completely new ideas and are applied to the UCP for the first time.

The next section introduces a hybrid algorithm of SA and TS which is called ST algorithm. Section 6.3 presents the hybrid algorithm of GA and TS which is called GT algorithm. In Section 6.4 the third hybrid algorithm integrating the three methods SA, TS, and GA which is called GST algorithm is introduced.

6.2 HYBRID OF SIMULATED ANNEALING AND TABU SEARCH

6.2.1 THE PROPOSED ST ALGORITHM

Since the main feature of the TS method is to prevent cycling of solutions, we could explore this point in refining the SA algorithm. The main idea in the proposed ST algorithm is to use the TS algorithm to prevent the repeated solutions from being accepted by the SA. This will save time and improve the quality of the solution obtained.

The proposed ST algorithm [93] may be described as an SA algorithm with the TS algorithm used as a filter to reject the repeated trial solutions from being tested by the SAA. The TS method is implemented as a preprocessor step in the SAA to test a set of neighbors to the current solution. The trial solution which satisfies the tabu test is accepted. This accepted trial solution is then accepted or rejected according to the SA test. The main steps of the ST algorithm are described in the flow chart of Fig.(6.1).

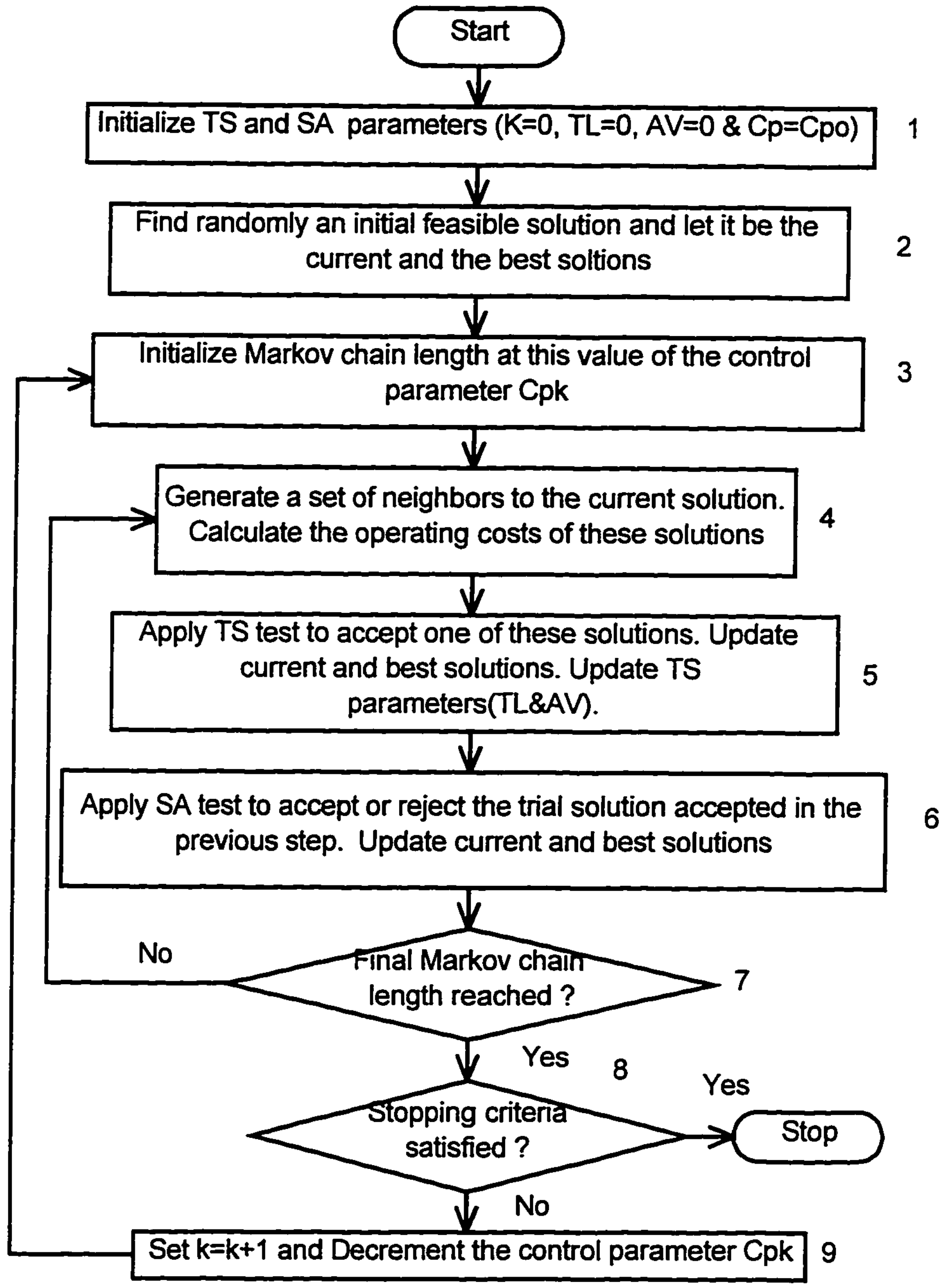


Fig.(6.1) Flow chart of the proposed ST algorithm for the UCP

6.2.2 TABU SEARCH PART IN THE ST ALGORITHM

In the TS part of the ST algorithm, the STM procedures are implemented. In this implementation, the TL is created using approach 5 which is explained in Section 4.4.

The tabu test (block 5) can be described as follows:

- Sort the set of trial solutions (neighbors) in an ascending order according to their objective functions.
- Apply the tabu acceptance test in order until one of these solutions is accepted.
- Tabu acceptance test: If the trial solution (U_j) is NOT tabu or tabu but satisfy the AV, then accept the trial solution as the current solution, set $U_C = U_j$ and $F_C = F_j$, and go to the SA test. Otherwise apply the test to the next solution.

6.2.3 SIMULATED ANNEALING PART IN THE ST ALGORITHM

In the SA part of ST algorithm we used the polynomial-time cooling schedule (described in Section 3.3.1) to decrement the control parameter during the search (block 9). The SA test implemented in the ST algorithm (block 6) is described in the following steps:

Let U_C, F_C be the SA current solution and the corresponding operating cost respectively.

Let U_j, F_j be the trial solution accepted by the previous tabu test and the corresponding operating cost respectively.

SA acceptance test:

- If $F_j \le F_c$, or $\exp[(F_c - F_j)/C_p] \ge U(0,1)$, then accept the trial solution, and update the current solution; set $U_c = U_j$ and $F_c = F_j$ then go to block (7).
- Otherwise reject the trial solution and set $U_j = U_c$ and $F_j = F_c$ then go to block (7).

6.2.4 NUMERICAL RESULTS OF THE ST ALGORITHM

Considering the proposed ST algorithm, a computer program has been implemented. The three examples, previously solved in the last chapters, are solved again using the ST algorithm [29,41,62].

The following control parameters have been chosen after running a number of simulations: maximum number of iterations=3000, tabu size=7, initial control parameter (temperature)=5000, chain length=150, ε=0.00001, and δ=0.3.

Table (6.1) shows a comparison of the results obtained for the three examples 1,2, and 3 as solved by the SA, the TS and the ST algorithms. It is obvious that the ST algorithm achieves reduction in the operating costs for the three examples. Also, the number of iterations is less for Example 1, while for Examples 2 and 3 the ATSA converges faster.

Table (6.2) also shows a comparison of the ST algorithm results with the results of the LR and IP for Examples 1 and 2. It is obvious that significant cost savings are achieved.

Detailed results for Example 1 are given in Tables (6.3) and (6.4). Table (6.3) shows the load sharing among the committed units in the 24 hours. Table (6.4) gives the hourly load demand, and the corresponding economic dispatch costs, start-up costs, and total operating cost.

Detailed results for Example 3 are given in Tables (6.5), (6.6) and (6.7).

Table (6.1) Comparison between SAA, STSA, ATSA and the ST algorithm

	Example	SAA	STSA	ATSA	ST
Total Cost ($)	1	536622	538390	537686	536386
”	2	59512	59512	59385	59385
”	3	662664	662583	660864	660596
Iterations No.	1	384	1924	1235	625
”	2	652	616	138	538
”	3	2361	3900	2547	2829

Table (6.2) Comparison between the LR, the IP and the ST algorithm

	Example	LR [29]	IP [41]	ST
Total Cost ($)	1	540895	-	536386
”	2	-	60667	59380
% Saving	1	0	-	0.83
”	2	-	0	2.11

Table (6.3) Power sharing (MW) of Example 1.

HR	Unit Number							
	2	3	4	6	7	8	9	10
1	400.00	0.00	0.00	185.04	0.00	350.26	0.00	89.70
2	395.36	0.00	0.00	181.09	0.00	338.36	0.00	85.19
3	355.38	0.00	0.00	168.67	0.00	300.95	0.00	75.00
4	333.13	0.00	0.00	161.75	0.00	280.12	0.00	75.00
5	400.00	0.00	0.00	185.04	0.00	350.26	0.00	89.70
6	400.00	0.00	295.68	200.00	0.00	375.00	0.00	129.32
7	400.00	383.56	420.00	200.00	0.00	375.00	0.00	191.44
8	400.00	295.59	396.65	200.00	0.00	375.00	569.93	162.83
9	400.00	468.07	420.00	200.00	0.00	375.00	768.01	218.92
10	400.00	444.60	420.00	200.00	358.05	375.00	741.06	211.29
11	400.00	486.30	420.00	200.00	404.89	375.00	788.95	224.86
12	400.00	514.11	420.00	200.00	436.09	375.00	820.89	233.91
13	400.00	479.35	420.00	200.00	397.09	375.00	780.96	222.60
14	400.00	388.98	420.00	200.00	295.64	375.00	677.18	193.20
15	400.00	310.07	410.84	200.00	250.00	375.00	586.56	167.54
16	400.00	266.64	368.27	200.00	250.00	375.00	536.68	153.41
17	400.00	317.31	417.93	200.00	250.00	375.00	594.87	169.89
18	400.00	458.51	420.00	200.00	373.65	375.00	757.03	215.81
19	400.00	486.30	420.00	200.00	404.89	375.00	788.95	224.86
20	400.00	375.08	420.00	200.00	280.03	375.00	661.21	188.68
21	400.00	215.96	318.62	200.00	0.00	375.00	478.49	136.93
22	400.00	217.46	320.12	200.00	0.00	375.00	0.00	137.42
23	400.00	165.00	246.88	0.00	0.00	375.00	0.00	113.12
24	396.36	165.00	163.80	0.00	0.00	339.29	0.00	85.55

Table (6.4) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.03E+03	9.67E+03	0.00E+00	9.67E+03
2	1.00E+03	9.45E+03	0.00E+00	9.45E+03
3	9.00E+02	8.56E+03	0.00E+00	8.56E+03
4	8.50E+02	8.12E+03	0.00E+00	8.12E+03
5	1.03E+03	9.67E+03	0.00E+00	9.67E+03
6	1.40E+03	1.34E+04	1.06E+03	1.45E+04
7	1.97E+03	1.94E+04	1.63E+03	2.10E+04
8	2.40E+03	2.38E+04	1.82E+03	2.56E+04
9	2.85E+03	2.83E+04	0.00E+00	2.83E+04
10	3.15E+03	3.17E+04	2.06E+03	3.38E+04
11	3.30E+03	3.32E+04	0.00E+00	3.32E+04
12	3.40E+03	3.42E+04	0.00E+00	3.42E+04
13	3.28E+03	3.30E+04	0.00E+00	3.30E+04
14	2.95E+03	2.97E+04	0.00E+00	2.97E+04
15	2.70E+03	2.73E+04	0.00E+00	2.73E+04
16	2.55E+03	2.58E+04	0.00E+00	2.58E+04
17	2.73E+03	2.75E+04	0.00E+00	2.75E+04
18	3.20E+03	3.22E+04	0.00E+00	3.22E+04
19	3.30E+03	3.32E+04	0.00E+00	3.32E+04
20	2.90E+03	2.92E+04	0.00E+00	2.92E+04
21	2.13E+03	2.12E+04	0.00E+00	2.12E+04
22	1.65E+03	1.63E+04	0.00E+00	1.63E+04
23	1.30E+03	1.31E+04	0.00E+00	1.31E+04
24	1.15E+03	1.18E+04	0.00E+00	1.18E+04

Total operating cost = $536386.

Table (6.5) Power sharing (MW) of Example 3 (for units 1-13).

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	45.85	15.20	15.20
2	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	41.05	15.20
3	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	30.00	15.20
4	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20
5	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	60.00	15.20
6	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	68.00	15.20
7	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
8	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
9	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
10	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	64.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20	15.20

Table (6.6) Power sharing (MW) of Example 3 (for units 14-26).

HR	Unit Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	68.95	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	68.95	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	344.8	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	0.00	0.00	71.20	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
8	0.00	100.0	100.0	155.0	155.0	155.0	155.0	152.2	68.95	0.00	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	124.2	0.00	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	124.2	0.00	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	119.0	0.00	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	99.00	0.00	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	157.0	68.95	0.00	350.0	350.0	350.0
16	100.0	100.0	38.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
17	100.0	100.0	78.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
18	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	129.0	0.00	350.0	350.0	350.0
22	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
23	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
24	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	331.3	350.0	350.0

Table (6.7) Load demand and hourly costs ($) of Example 3

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.87E+04	0.00E+00	1.87E+04
2	1.80E+03	1.84E+04	0.00E+00	1.84E+04
3	1.72E+03	1.64E+04	0.00E+00	1.64E+04
4	1.70E+03	1.61E+04	0.00E+00	1.61E+04
5	1.75E+03	1.68E+04	0.00E+00	1.68E+04
6	1.91E+03	1.91E+04	1.60E+02	1.93E+04
7	2.05E+03	2.15E+04	1.00E+02	2.16E+04
8	2.40E+03	2.97E+04	7.00E+02	3.04E+04
9	2.60E+03	3.40E+04	1.00E+02	3.41E+04
10	2.60E+03	3.40E+04	0.00E+00	3.40E+04
11	2.62E+03	3.44E+04	0.00E+00	3.44E+04
12	2.58E+03	3.35E+04	0.00E+00	3.35E+04
13	2.59E+03	3.37E+04	0.00E+00	3.37E+04
14	2.57E+03	3.33E+04	0.00E+00	3.33E+04
15	2.50E+03	3.17E+04	0.00E+00	3.17E+04
16	2.35E+03	2.85E+04	0.00E+00	2.85E+04
17	2.39E+03	2.92E+04	0.00E+00	2.92E+04
18	2.48E+03	3.12E+04	0.00E+00	3.12E+04
19	2.58E+03	3.35E+04	0.00E+00	3.35E+04
20	2.62E+03	3.44E+04	0.00E+00	3.44E+04
21	2.60E+03	3.40E+04	0.00E+00	3.40E+04
22	2.48E+03	3.12E+04	0.00E+00	3.12E+04
23	2.15E+03	2.47E+04	0.00E+00	2.47E+04
24	1.90E+03	2.14E+04	0.00E+00	2.14E+04

Total operating cost = $660596.75

6.3 HYBRID OF GENETIC ALGORITHMS AND TABU SEARCH

6.3.1 THE PROPOSED GT ALGORITHM

In this section we propose a new algorithm (GT) based on integrating the use of GA and TS methods to solve the UCP. The proposed algorithm is mainly based on the GA approach. TS is incorporated in the reproduction phase of the GA to generate a number of new solutions (chromosomes). These new solutions are generated as neighbors to randomly selected solutions (chromosomes) in the current population and are added to the new population of the GA. The idea behind using TS is to ensure generating new solutions and hence to prevent the search from being entrapped in a local minimum.

The major steps of the proposed GT algorithm are summarized as follows:

- Create an initial population by randomly generating a set of feasible solutions (chromosomes) using rules described in Section 2.6.
- Evaluate the population and check the convergence.
- Generate new population from the current population by applying the GA operators.
- Use TS to create new solutions (chromosomes) and add them to the new GA population.

The details of the GT algorithm are also shown in the flow chart of Fig.(6.2).

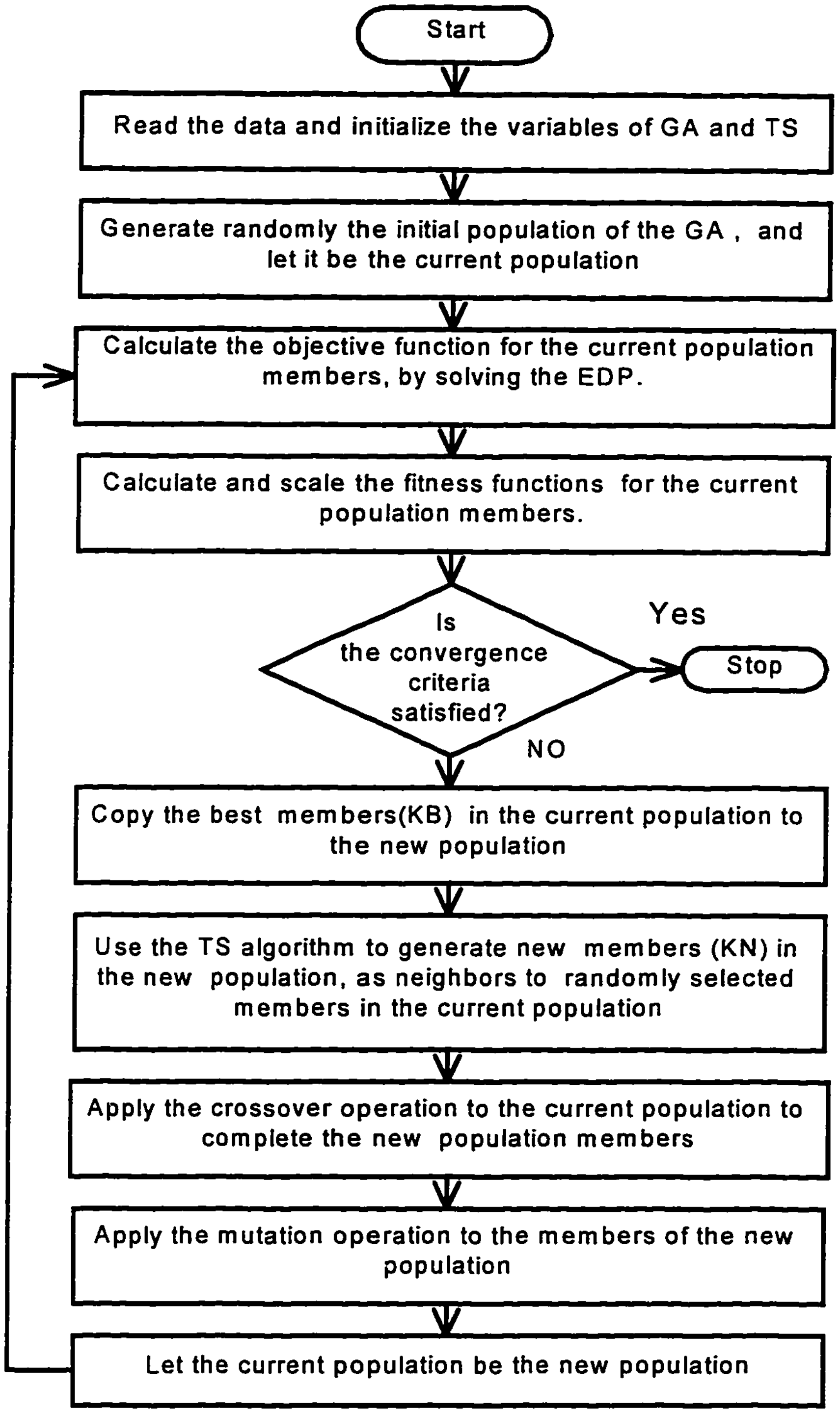


Fig.(6.2) Flow Chart of the proposed GT Algorithm

The following two sections summarize the implementations of different components of the proposed GT algorithm.

6.3.2 GENETIC ALGORITHM PART OF THE GT ALGORITHM

GA is the basic part of the proposed GT algorithm. GA implementation is similar to that described in Section 5.3. The implementation of the GA can be summarized as follows:

- Creating an initial population by randomly generating a set of feasible solutions (chromosomes) using rules described in Section 2.6.
- Solution is coded as a mix between binary and decimal number (see Section 5.3.2).
- Fitness function is constructed from the objective function only without penalty terms (see Section 5.3.3).
- Reproduction operators of the GA, crossover and mutation described in Sections 5.3.5 and 5.3.6 are used.

6.3.3 TABU SEARCH PART OF THE GT ALGORITHM

In the implementation of the GT algorithm, the TS is incorporated in the reproduction phase of the GA as a tool for escaping the local minimum and the premature convergence of the GA. TS is used to generate a prespecified number of new solutions that have not been generated before (typically 5-10% of the population size). The tabu list in the initial population is initially empty. It is then updated to accept or reject the new solutions in each generation of the GA. The TS is implemented as a short term memory algorithm. The flow chart of Fig. (6.3) shows the main steps of the TS algorithm implementation that have been used as a part of the proposed GT algorithm.

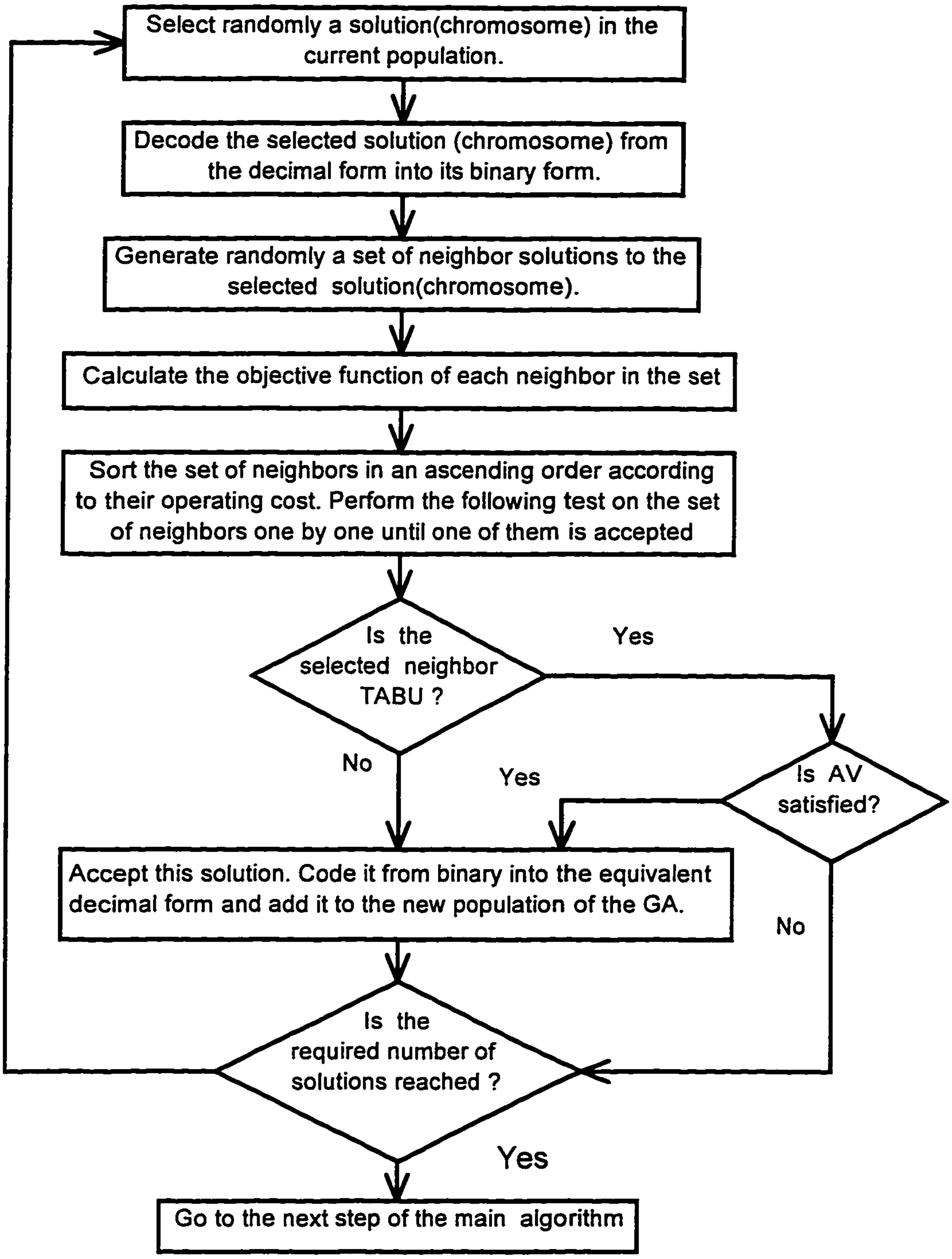


Fig. (6.3) Tabu Search Part of the GT Algorithm

6.3.4 NUMERICAL RESULTS OF THE GT ALGORITHM

In order to test the proposed GT algorithm, the same three examples are considered.

A number of tests on the performance of the proposed algorithm, have been carried out to determine the most suitable GA and TS parameters settings. The following control parameters have been chosen after running a number of simulations: population size=50, crossover rate=0.8, mutation rate=0.3, best solutions copies=2, and the maximum number of generations=1000, tabu list size=7.

Different experiments with different random number seeds were carried out to investigate the performance of the proposed algorithm. It was found that the proposed algorithm performs better than the TS algorithm and the simple GA, in terms of both solution quality and number of iterations. Fig.(6.4) shows the convergence process of the proposed algorithm when applied to solve Example 1.

Table (6.8) shows the comparison with the results of the TS and the GA for the three examples. It is obvious that a substantial reduction in the objective function, compared to the simple GA, has been achieved, while the GT algorithm converges slower. This improvement in the objective function is due to the role of the TS in generating new and good members in each new population of the GA.

Table (6.9) presents the comparison with the results obtained in the literature (LR and IP) for Examples 1 and 2. In addition to the high percentage saving in the cost over these classical methods, the proposed algorithm has so many other advantages. For instance, it may produce various solutions with the same objective function. This gives the operator the flexibility to select any of them. Also, any additional operating constraints could be easily handled without reformulating the problem.

Tables (6.10), (6.11) and (6.12) show the detailed results for Example 1 [29]. Table (6.10) shows the load sharing among the committed units in 24 hours. Table (6.11) presents the final schedule of the 24 hours, given in Table (6.10), in the form of its equivalent decimal numbers. Table (6.12) gives the hourly load demand and the corresponding economic dispatch costs, start-up costs, and total operating cost of the final schedule

Tables (6.13), (6.14) and (6.15) show the detailed results for Example 3 [62]

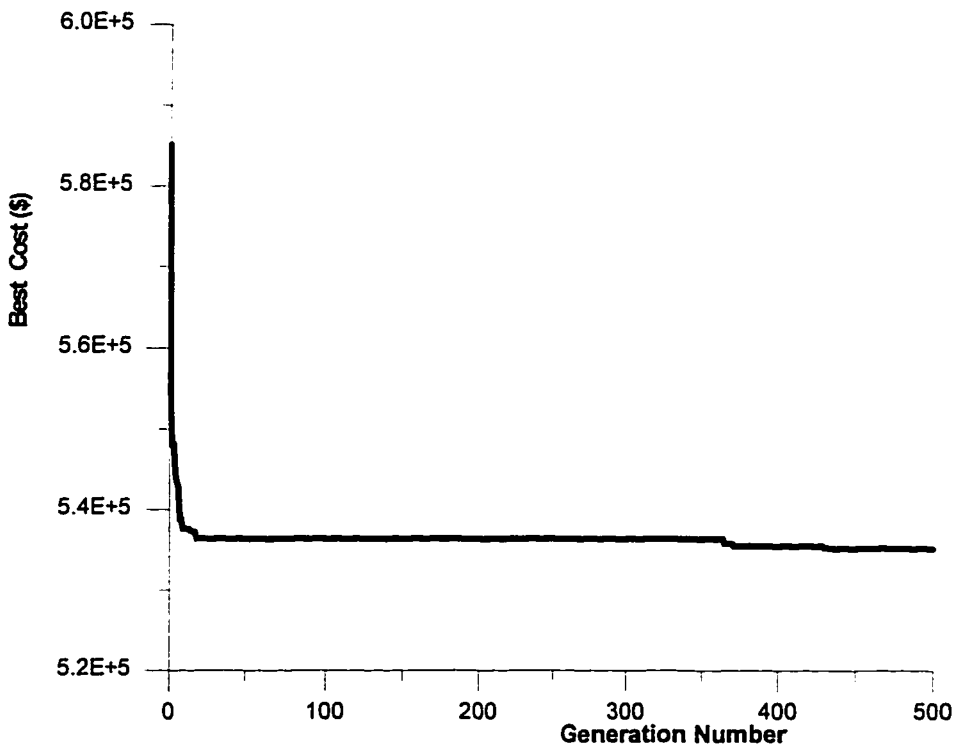


Fig.(6.4) Convergence of the proposed GT Algorithm (Example 1)

Table (6.8) Comparison between the STSA, the GA and the GT algorithm,

	Example	STSA	GA	GT
Total Cost ($)	1	538390	537372	535234
„	2	59512	59491	59380
„	3	662583	661439	660412
Generations/Iterations No.	1	1924	411	434
„	2	616	393	513
„	3	3900	985	623

Table (6.9) Comparison between LR, IP and the GT algorithm

	Example	LR [29]	IP [41]	GT
Total Cost ($)	1	540895	-	535234
„	2	-	60667	59380
% Saving	1	0	-	1.05
„	2	-	0	2.12

Table (6.10) Power sharing (MW) of Example 1

HR	Unit Number						
	2	3	4	6	7	8	9	10
1	400.0	0.0	0.0	185.0	0.0	350.3	0.0	89.7
2	395.4	0.0	0.0	181.1	0.0	338.4	0.0	85.2
3	355.4	0.0	0.0	168.7	0.0	301.0	0.0	75.0
4	333.1	0.0	0.0	161.8	0.0	280.1	0.0	75.0
5	400.0	0.0	0.0	185.0	0.0	350.3	0.0	89.7
6	400.0	0.0	295.7	200.0	0.0	375.0	0.0	129.3
7	400.0	383.6	420.0	200.0	0.0	375.0	0.0	191.4
8	400.0	295.6	396.7	200.0	0.0	375.0	569.9	162.8
9	400.0	468.1	420.0	200.0	0.0	375.0	768.0	218.9
10	400.0	444.6	420.0	200.0	358.1	375.0	741.1	211.3
11	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
12	400.0	514.1	420.0	200.0	436.1	375.0	820.9	233.9
13	400.0	479.4	420.0	200.0	397.1	375.0	781.0	222.6
14	400.0	389.0	420.0	200.0	295.6	375.0	677.2	193.2
15	400.0	310.1	410.8	200.0	250.0	375.0	586.6	167.5
16	400.0	266.6	368.3	200.0	250.0	375.0	536.7	153.4
17	400.0	317.3	417.9	200.0	250.0	375.0	594.9	169.9
18	400.0	458.5	420.0	200.0	373.7	375.0	757.0	215.8
19	400.0	486.3	420.0	200.0	404.9	375.0	789.0	224.9
20	400.0	0.0	420.0	200.0	442.2	375.0	827.2	235.7
21	400.0	0.0	404.9	200.0	0.0	375.0	579.6	165.6
22	400.0	0.0	216.5	196.7	0.0	375.0	358.8	103.0
23	0.0	0.0	235.2	200.0	0.0	375.0	380.6	109.2
24	0.0	0.0	186.4	188.0	0.0	359.1	323.5	93.0

**Units 1,5 are OFF all hours.

Table (6.11) GA population of the best solution for Example 1

Unit Number				
1,6	2,7	3,8	4,9	5,10
0	4194303	524224	16777184	0
16777215	1048064	16777215	16777088	16777215

Table (6.12) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1025	9670.04	0.00	9670.04
2	1000	9446.62	0.00	9446.62
3	900	8560.91	0.00	8560.91
4	850	8123.13	0.00	8123.13
5	1025	9670.04	0.00	9670.04
6	1400	13434.10	1055.97	14490.00
7	1970	19385.10	1631.43	21016.50
8	2400	23815.50	1817.70	25633.20
9	2850	28253.90	0.00	28253.90
10	3150	31701.70	2057.58	33759.30
11	3300	33219.80	0.00	33219.80
12	3400	34242.00	0.00	34242.00
13	3275	32965.50	0.00	32965.50
14	2950	29706.30	0.00	29706.30
15	2700	27259.70	0.00	27259.70
16	2550	25819.80	0.00	25819.80
17	2725	27501.60	0.00	27501.60
18	3200	32205.70	0.00	32205.70
19	3300	33219.80	0.00	33219.80
20	2900	28899.00	0.00	28899.00
21	2125	20698.40	0.00	20698.40
22	1650	16251.80	0.00	16251.80
23	1300	12989.60	0.00	12989.60
24	1150	11631.00	0.00	11631.00

Total operating Cost = $535234

Table (6.13) Power sharing (MW) of Example 3 (for units 1-13).

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	2.40	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	76.00	52.40	15.20
2	2.40	2.40	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	76.00	45.20
3	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	28.40	15.20
4	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	15.20	15.20
5	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	58.40	15.20
6	2.40	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	66.40	15.20
7	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
8	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
9	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
10	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
11	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	76.00	76.00	76.00	76.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	64.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20	15.20

Table (6.14) Power sharing (MW) of Example 3 (for units 14-26).

HR	Unit Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	343.2	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	0.00	0.00	72.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
8	0.00	100.0	100.0	155.0	155.0	155.0	155.0	153.0	68.95	0.00	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	125.0	0.00	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	125.0	0.00	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	145.0	0.00	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	119.0	0.00	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	99.00	0.00	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	157.0	68.95	0.00	350.0	350.0	350.0
16	100.0	100.0	38.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
17	100.0	100.0	78.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
18	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	129.0	0.00	350.0	350.0	350.0
22	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
23	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
24	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	331.3	350.0	350.0

Table (6.15) Load demand and hourly costs ($) of Example 3.

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.80E+04	0.00E+00	1.80E+04
2	1.80E+03	1.77E+04	0.00E+00	1.77E+04
3	1.72E+03	1.66E+04	0.00E+00	1.66E+04
4	1.70E+03	1.63E+04	0.00E+00	1.63E+04
5	1.75E+03	1.70E+04	0.00E+00	1.70E+04
6	1.91E+03	1.93E+04	1.60E+02	1.94E+04
7	2.05E+03	2.16E+04	1.00E+02	2.17E+04
8	2.40E+03	2.98E+04	7.00E+02	3.05E+04
9	2.60E+03	3.41E+04	1.00E+02	3.42E+04
10	2.60E+03	3.41E+04	0.00E+00	3.41E+04
11	2.62E+03	3.46E+04	0.00E+00	3.46E+04
12	2.58E+03	3.35E+04	0.00E+00	3.35E+04
13	2.59E+03	3.37E+04	0.00E+00	3.37E+04
14	2.57E+03	3.33E+04	0.00E+00	3.33E+04
15	2.50E+03	3.17E+04	0.00E+00	3.17E+04
16	2.35E+03	2.85E+04	0.00E+00	2.85E+04
17	2.39E+03	2.92E+04	0.00E+00	2.92E+04
18	2.48E+03	3.12E+04	0.00E+00	3.12E+04
19	2.58E+03	3.35E+04	0.00E+00	3.35E+04
20	2.62E+03	3.44E+04	0.00E+00	3.44E+04
21	2.60E+03	3.40E+04	0.00E+00	3.40E+04
22	2.48E+03	3.12E+04	0.00E+00	3.12E+04
23	2.15E+03	2.47E+04	0.00E+00	2.47E+04
24	1.90E+03	2.14E+04	0.00E+00	2.14E+04

Total operating cost = $660412.4375

6.4 HYBRID OF GENETIC ALGORITHMS , SIMULATED ANNEALING AND TABU SEARCH

6.4.1 THE PROPOSED GST ALGORITHM

This section presents a new algorithm (GST) based on integrating the GA, the TS and the SA methods to solve the UCP. The proposed GST algorithm could be considered as a further improvement to the GT algorithm implemented in Section 6.3.

The core of the proposed algorithm is based on GA. The TS is used to generate new population members in the reproduction phase of the GA. Moreover, the SA method is adopted to improve the convergence of the GA by testing the population members of the GA after each generation. The SA test allows the acceptance of any solution at the beginning of the search, while only good solutions will have higher probability of acceptance as the generation number increases. The effect of using the SA is to accelerate the convergence of the GA and also increase the fine tuning capability of the GA when approaching a local minimum.

The major steps of the algorithm are summarized as follows[92]:

- Create an initial population by randomly generating a set of feasible solutions (Section 2.4), and initialize the current solution of the SA algorithm.
- Apply GA operators to generate new population members.
- Use the TS algorithm to generate some members in the new population (typically 5-10% of the population size), as neighbors to the randomly selected solutions in the current population.

- Apply the SA algorithm to test all the members of the new population

Fig. (6.5) shows the flow chart of the proposed GTS algorithm.

The following sections summarize the implementations of different components of the proposed GTS algorithm.

6.4.2 GENETIC ALGORITHM PART OF THE GST ALGORITHM

The implementations of the GA in the GST algorithm are exactly the same as that described in 6.3.3.

6.4.3 TABU SEARCH PART OF THE GST ALGORITHM

The implementations of the TS part in the GST is also the same as that described in Section 6.3.4.

6.4.4 SIMULATED ANNEALING PART OF THE GST ALGORITHM

After creating a new population of the GA, the SA test is applied to the members of this population, one by one. The steps of the SA algorithm as applied at the kth generation of the proposed GST algorithm are described as follows:

Let U_c, F_c be the SA current solution and the corresponding operating cost respectively.

Let U_j, F_j be the jth solution (chromosome) in a given population and the corresponding operating cost respectively.

Step (1): Calculate the new temperature $cp^k = cp^o(\beta)^k$, where $0 < \beta < 1$.

Step (2): At the same calculated temperature, c_p^k, apply the following acceptance test for the population members of the GA one by one.

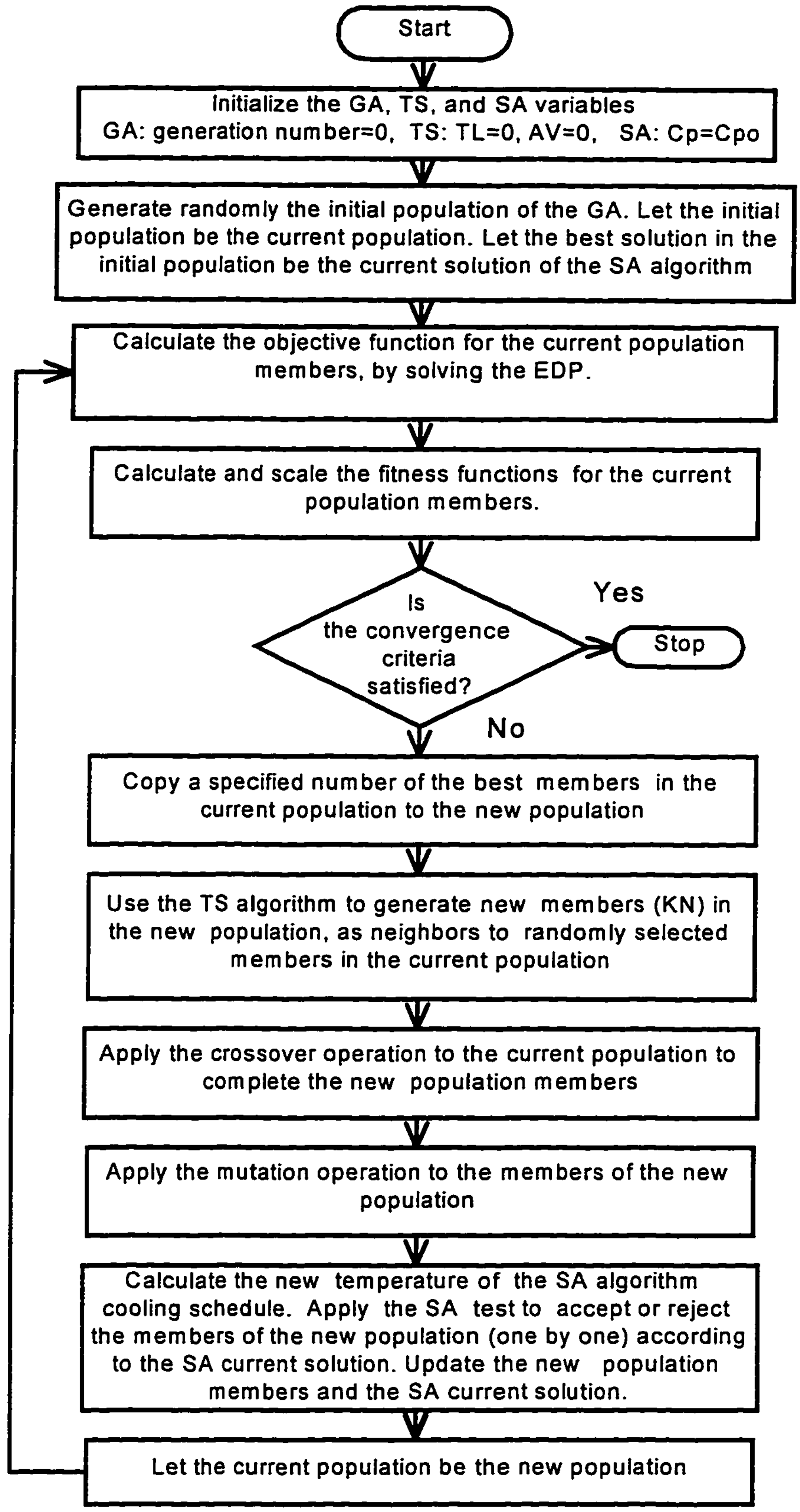


Fig.(6.5) Flow Chart of the proposed GST Algorithm for the UCP

Step (3): Acceptance test: If $F_j \le F_c$, or $\exp[(F_c - F_j) / Cp] \ge U(0,1)$, then accept the population member, and update the current solution; set $U_c = U_j$ and $F_c = F_j$ then go to Step (4). Otherwise, reject the population member and set $U_j = U_c$ and $F_j = F_c$ then go to Step (4).

Step (4): If all the population members are tested go to the next step in the main algorithm, otherwise go to Step (2).

6.4.5 NUMERICAL RESULTS OF THE GST ALGORITHM

For the purpose of testing the proposed hybrid GST algorithm, the same three examples, from the literature [29,41,62], are considered.

The following control parameters have been chosen after running a number of simulations: population size=50, crossover rate=0.8, mutation rate=0.3, elite copies=2, and the maximum number of generations=1000, tabu size=7, initial control parameter (temperature)=5000, β =0.9.

Fig.(6.6) shows the convergence speed of the GST algorithm, when Example 1 is solved.

Different experiments were carried out to evaluate the results obtained by the proposed GST algorithm and those obtained from the individual algorithms in [Chapters 3,4 and 5]. Table (6.16) shows the results of this comparison for the three examples. The superiority of the GST algorithm is obvious. It is clear that the GTS algorithm performs better than each of the individual algorithms, in terms of both solution quality and number of generations.

Table (6.17) presents the comparison of results obtained in the literature (LR and IP) for Examples 1 and 2, and the proposed GST algorithm.

Tables (6.18), (6.19) and (6.20) show detailed results for Example 1 [29]. Table (6.18) shows the load sharing among the committed units in the 24 hours. Table (6.18) presents the final schedule of the 24 hour period, given in Table (6.19), in the form of its equivalent decimal numbers. Table (6.20) gives the hourly load demand and the corresponding economic dispatch costs, start-up costs, and total operating cost.

Tables (6.21), (6.22) and (6.23) show detailed results for Example 3, [62].

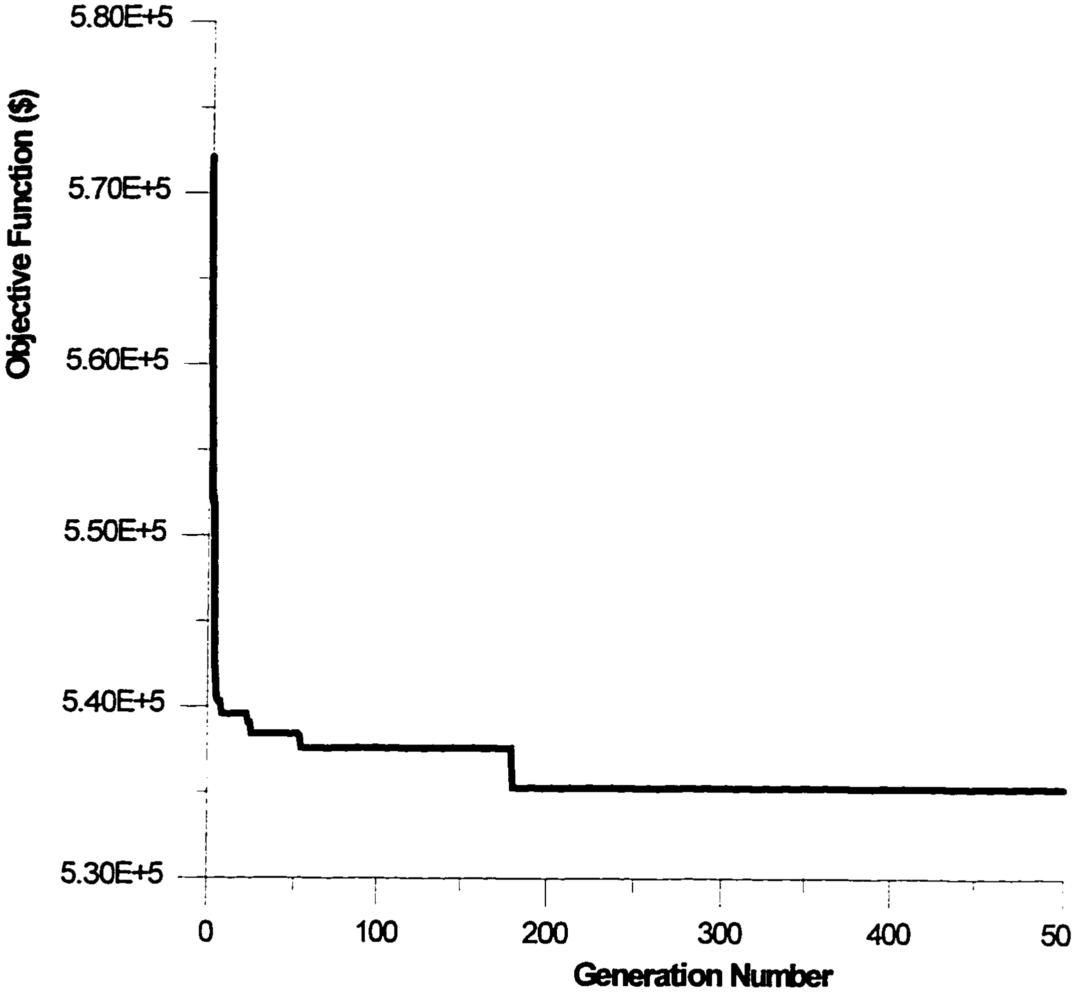


Fig.(6.6) Convergence of the proposed GST Algorithm (Example 1)

Table (6.16) Comparison between SAA, STSA, GA and the GST algorithm

	Example	SA	STSA	GA	GST
Total Cost ($)	1	536622	538390	537372	535271
,,	2	59385	59512	59491	59380
,,	3	662664	662583	661439	660151
Generations/Iterations No.	1	384	1924	411	181
,,	2	652	616	393	180
,,	3	2361	3900	985	762

Table (6.17) Comparison between LR and IP and the GST algorithm

	Example	LR	IP	GST
Total Cost ($)	1	540895	-	535271
,,	2	-	60667	59380
% Saving	1	0	-	1.04
,,	2	-	0	2.12

Table (6.18) Power sharing (MW) of Example 1

HR	Unit Number							
	2	3	4	6	7	8	9	10
1	400	0	0	185.0	0	350.2	0	89.7
2	395.3	0	0	181.0	0	338.3	0	85.19
3	355.3	0	0	168.6	0	300.9	0	75
4	333.1	0	0	161.7	0	280.1	0	75
5	400	0	0	185.0	0	350.2	0	89.7
6	400	0	295.6	200	0	375	0	129.3
7	400	383.5	420	200	0	375	0	191.4
8	400	295.5	396.6	200	0	375	569.9	162.8
9	400	468.0	420	200	0	375	768.0	218.9
10	400	444.6	420	200	358.0	375	741.0	211.2
11	400	486.3	420	200	404.8	375	788.9	224.8
12	400	514.1	420	200	436.0	375	820.8	233.9
13	400	479.3	420	200	397.0	375	780.9	222.6
14	400	388.9	420	200	295.6	375	677.1	193.2
15	400	310.0	410.8	200	250	375	586.5	167.5
16	400	266.6	368.2	200	250	375	536.6	153.4
17	400	317.3	417.9	200	250	375	594.8	169.8
18	400	458.5	420	200	373.6	375	757.0	215.8
19	400	486.3	420	200	404.8	375	788.9	224.8
20	400	491.8	0	200	411.1	375	795.3	226.6
21	400	344.7	0	200	0	375	626.4	178.8
22	400	459.0	0	200	0	375	0	215.9
23	400	194.9	0	200	0	375	0	130.0
24	389.5	165	0	179.3	0	332.9	0	83.15

**Units 1,5 are OFF all hours.

Table (6.19) The best population in the GA for Example 1

Unit Number				
1,6	2,7	3,8	4,9	5,10
0	16777215	16777152	16777184	0
4194303	1048064	16777215	2097024	16777215

Table (6.20) Load demand and hourly costs ($) of Example 1

HR	LOAD	ED-COST	ST-COST	T-COST
1	1,025	9,670.0	-	9,670.0
2	1,000	9,446.6	-	9,446.6
3	900	8,560.9	-	8,560.9
4	850	8,123.1	-	8,123.1
5	1,025	9,670.0	-	9,670.0
6	1,400	13,434.1	1,056.0	14,490.0
7	1,970	19,385.1	1,631.4	21,016.5
8	2,400	23,815.5	1,817.7	25,633.2
9	2,850	28,253.9	-	28,253.9
10	3,150	31,701.7	2,057.6	33,759.3
11	3,300	33,219.8	-	33,219.8
12	3,400	34,242.1	-	34,242.1
13	3,275	32,965.5	-	32,965.5
14	2,950	29,706.3	-	29,706.3
15	2,700	27,259.7	-	27,259.7
16	2,550	25,819.8	-	25,819.8
17	2,725	27,501.6	-	27,501.6
18	3,200	32,205.7	-	32,205.7
19	3,300	33,219.8	-	33,219.8
20	2,900	29,198.0	-	29,198.0
21	2,125	20,994.5	-	20,994.5
22	1,650	16,158.6	-	16,158.6
23	1,300	12,758.9	-	12,758.9
24	1,150	11,397.1	-	11,397.1

Total operating Cost = $535270.94

Table (6.21) Power sharing (MW) of Example 3 (for units 1-13).

HR	Unit Number												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.40	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	76.00	50.00	15.20
2	2.40	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	76.00	45.20
3	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	28.40	15.20
4	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20
5	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	58.40	15.20
6	0.00	2.40	0.00	0.00	0.00	4.00	0.00	0.00	0.00	76.00	76.00	66.40	15.20
7	0.00	2.40	2.40	0.00	0.00	4.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
8	0.00	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
9	0.00	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
10	0.00	2.40	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
11	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
12	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
13	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
14	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	76.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	76.00	76.00	76.00	64.10
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.20	15.20	15.20	15.20

Table (6.22) Power sharing (MW) of Example 3 (for units 14-26).

HR	Unit Number											
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
2	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
3	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
4	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	343.2	350.0	350.0
5	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
6	0.00	0.00	0.00	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
7	0.00	0.00	67.20	155.0	155.0	155.0	155.0	0.00	0.00	0.00	350.0	350.0	350.0
8	0.00	100.0	100.0	155.0	155.0	155.0	155.0	152.2	68.95	0.00	350.0	350.0	350.0
9	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	124.2	0.00	350.0	350.0	350.0
10	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	124.2	0.00	350.0	350.0	350.0
11	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	146.6	0.00	350.0	350.0	350.0
12	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	106.6	0.00	350.0	350.0	350.0
13	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	116.6	0.00	350.0	350.0	350.0
14	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	96.60	0.00	350.0	350.0	350.0
15	100.0	100.0	100.0	155.0	155.0	155.0	155.0	157.0	68.95	0.00	350.0	350.0	350.0
16	100.0	100.0	38.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
17	100.0	100.0	78.10	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
18	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
19	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	109.0	0.00	350.0	350.0	350.0
20	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	149.0	0.00	350.0	350.0	350.0
21	100.0	100.0	100.0	155.0	155.0	155.0	155.0	197.0	129.0	0.00	350.0	350.0	350.0
22	100.0	100.0	100.0	155.0	155.0	155.0	155.0	137.0	68.95	0.00	350.0	350.0	350.0
23	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	350.0	350.0	350.0
24	0.00	25.00	25.00	155.0	155.0	155.0	155.0	68.95	68.95	0.00	331.3	350.0	350.0

Table (6.23) Load demand and hourly costs ($) of Example 3

HR	LOAD	ED-COST	ST-COST	T-COST
1	1.82E+03	1.80E+04	0.00E+00	1.80E+04
2	1.80E+03	1.77E+04	0.00E+00	1.77E+04
3	1.72E+03	1.66E+04	0.00E+00	1.66E+04
4	1.70E+03	1.63E+04	0.00E+00	1.63E+04
5	1.75E+03	1.70E+04	0.00E+00	1.70E+04
6	1.91E+03	1.93E+04	1.60E+02	1.94E+04
7	2.05E+03	2.17E+04	1.00E+02	2.18E+04
8	2.40E+03	2.97E+04	7.00E+02	3.04E+04
9	2.60E+03	3.40E+04	1.00E+02	3.41E+04
10	2.60E+03	3.40E+04	0.00E+00	3.40E+04
11	2.62E+03	3.45E+04	0.00E+00	3.45E+04
12	2.58E+03	3.35E+04	0.00E+00	3.35E+04
13	2.59E+03	3.38E+04	0.00E+00	3.38E+04
14	2.57E+03	3.33E+04	0.00E+00	3.33E+04
15	2.50E+03	3.17E+04	0.00E+00	3.17E+04
16	2.35E+03	2.85E+04	0.00E+00	2.85E+04
17	2.39E+03	2.92E+04	0.00E+00	2.92E+04
18	2.48E+03	3.12E+04	0.00E+00	3.12E+04
19	2.58E+03	3.35E+04	0.00E+00	3.35E+04
20	2.62E+03	3.44E+04	0.00E+00	3.44E+04
21	2.60E+03	3.40E+04	0.00E+00	3.40E+04
22	2.48E+03	3.12E+04	0.00E+00	3.12E+04
23	2.15E+03	2.47E+04	0.00E+00	2.47E+04
24	1.90E+03	2.14E+04	0.00E+00	2.14E+04

Total operating cost = $660151.25

6.5 SUMMARY

In this chapter, three AI-based novel hybrid algorithms for the UCP are proposed. The hybrid algorithms integrate the use of the previously introduced algorithms: SA, TS, and GA. The ideas of the hybridization of these algorithms are original and are applied to the UCP for the first time. Various details of implementation have also been discussed.

In the first algorithm[93], the main features of the SA and the TS methods are integrated. The TS test is embedded in the SA algorithm to create a memory which prevents cycling of the solutions accepted by the SA. A significant cost saving has been achieved over the individual of both the TS and the SA methods.

The second hybrid algorithm is based on integrating the use of GA and TS methods to solve the UCP. The proposed algorithm is mainly based on the GA approach. The TS is used to induce new population members in the reproduction phase of the GA.

A third new hybrid algorithm which integrates the main features of the three algorithms: GA, TS, and SA is also proposed. The algorithm is mainly structured around the GA, while the TS is used to generate new members in the GA population. The SA algorithm is used to test all the GA members after each reproduction of a new population.

Among the three hybrid algorithms, it is found that, the overall performance of the GT algorithm is superior. The GT algorithm converges faster and gives a better quality of solutions.

In the next chapter, a comparison between the seven proposed algorithms and the other methods reported in the literature is detailed.

CHAPTER SEVEN

COMPARISONS OF THE PROPOSED ALGORITHMS FOR THE UNIT COMMITMENT PROBLEM

In the last four chapters, seven AI-based algorithms were proposed for solving the UCP.

The two proposed algorithms, presented in chapters 3 and 5, are considered new implementations of these techniques. These algorithms are

- A Simulated Annealing Algorithm (SAA), and
- A Genetic Algorithm (GA).

The other five proposed algorithms, presented in chapters 4 and 6, are original and are applied for the first time to solve the UCP. These algorithms are

- A Simple Tabu Search Algorithm (STSA),
- An Advanced Tabu Search Algorithm (ATSA),
- A hybrid of the Simulated annealing and Tabu search algorithms (ST),
- A hybrid of Genetic and Tabu search algorithms (GT), and
- A hybrid of Genetic, Simulated annealing, and Tabu search algorithms (GST).

This chapter is intended for the comparison between the results obtained using these algorithms and those obtained using other methods (LR, and IP) available in the literature [29,41,62].

7.1 RESULTS OF EXAMPLE 1

Table (7.1) and Figs.(7.1) and (7.2) show the daily operating costs and the number of iterations (or generations for the GA based algorithms), of the seven proposed algorithms and the LR and the SAA-67 methods as well, for Example 1.

Generally speaking, all the proposed algorithms outperform the results reported in the literature using the LR and the SAA-67 methods for the same example. The daily cost saving amount range between 2505 (using STSA) and 5661 (using GT) which is equivalent to a percentage saving of 0.46 to 1.05 respectively.

Comparing the results of the proposed algorithms, for Example 1, it is clear that the GT algorithm is the best as far as quality of solution and convergence speed are concerned.

Considering the algorithms based on a single technique, e.g., SA, STSA, ATSA, and GA, we conclude that the SA performance is the best among these algorithms as far as the objective function value and the number of iterations required for convergence are concerned.

It can also be concluded that the results of the hybrid algorithms, e.g. ST, GT, and GST are better than the results of the algorithms based on a single method.

Among the proposed hybrid algorithms, it is also found that the GT algorithm provides a better quality of solution and also a faster speed of convergence.

Table (7.1) Comparison of different algorithms for Example 1

Algorithm	Daily Operating Cost ($)	Amount of Daily Saving	% saving	No. of Iterations or Generations
LR [29]	540895	0.	0	-
SAA [67]	538803	2092	0.38	821
SAA	536622	4273	0.79	384
STSA	538390	2505	0.46	1924
ATSA	537686	3209	0.59	1235
GA	537372	3523	0.65	411
ST	536386	4509	0.83	625
GT	535234	5661	1.05	434
GST	535271	5624	1.04	181

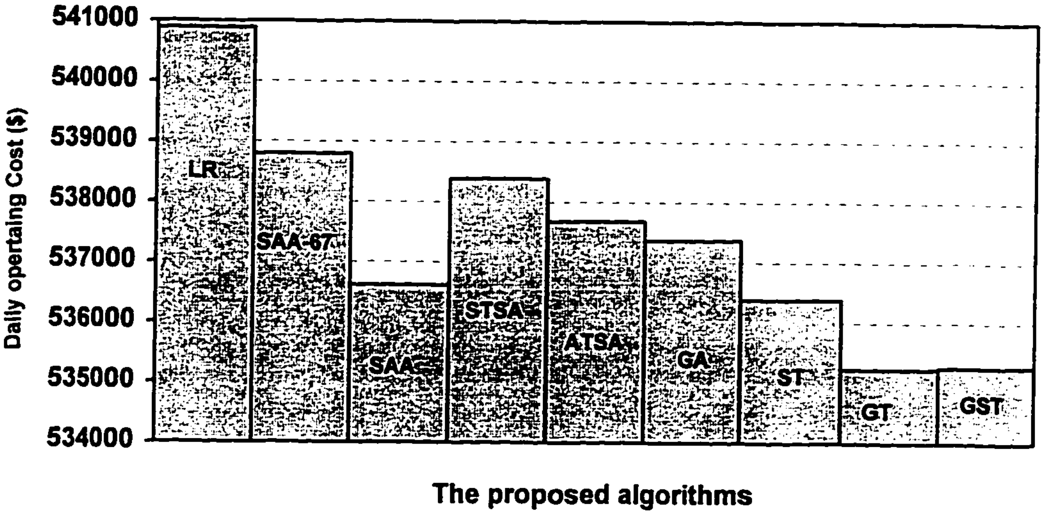


Fig.(7.1) Operating costs of different proposed algorithms for Example 1

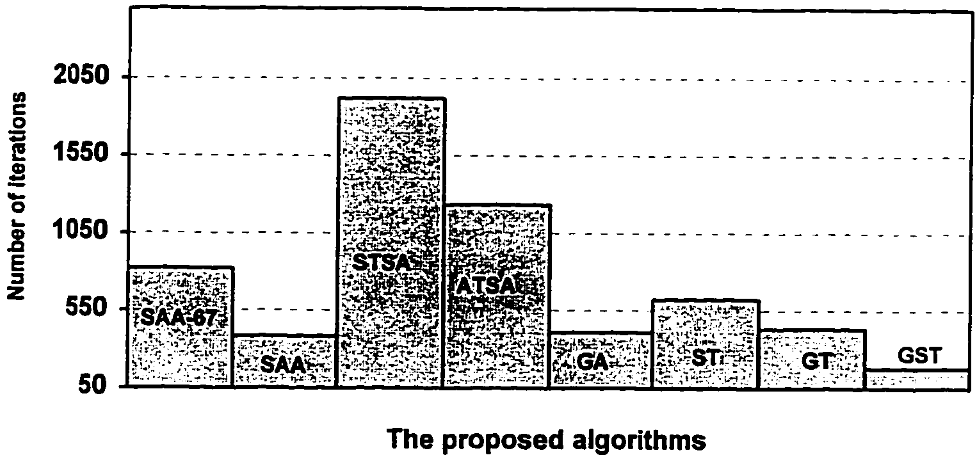


Fig.(7.2) Operating costs of different proposed algorithms for Example 2

7.2 RESULTS OF EXAMPLE 2

Table (7.2) and Figs.(7.3) and (7.4) show the daily operating costs and the number of iterations (or generations of the GA based algorithms), of the seven proposed algorithms and the IP and SAA-67 methods as well, for Example 2.

As shown, the results of our proposed algorithms are better than the results reported in the literature using the IP method for this example. Compared with the IP results, the daily cost saving amount ranges between 1155 (by SA and STSA) and 1287 (by GT and GST) which is equivalent to a percentage saving of 1.9 to 2.12 respectively.

Among the results of all the proposed algorithms, for Example 2, it is clear that the GT and GST algorithms are the best.

Considering the algorithms based on a single method , e.g. SA, STSA, ATSA, and GA, although there is not much difference between them, the ATSA gives slightly better results than the others as far as the objective function value and the number of iterations required for convergence are concerned.

It can also be concluded that the results of the hybrid algorithms, e.g. ST, GT, and GST are better than the results of the algorithms based on a single method.

Among the proposed hybrid algorithms, it is also found that the GST algorithm is the best in terms of the quality of solution and the speed of convergence.

Table (7.2) Comparison of different algorithms for Example 2

Algorithm	Daily Operating Cost ($)	Amount of Daily Saving	% saving	No. of Iterations or Generations
IP [41]	60667	0	0	-
SAA [67]	59512	1155	1.9	945
SAA	59512	1155	1.9	652
STSA	59512	1155	1.9	616
ATSA	59385	1282	2.1	138
GA	59491	1176	1.93	393
ST	59385	1282	2.11	538
GT	59380	1287	2.12	513
GST	59380	1287	2.12	180

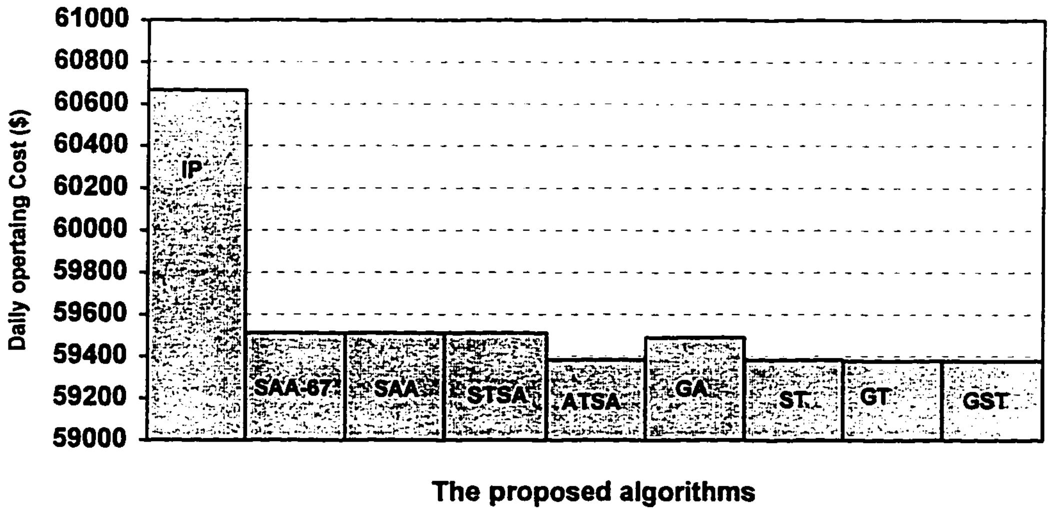


Fig.(7.3) Operating costs of different proposed algorithms for Example 2

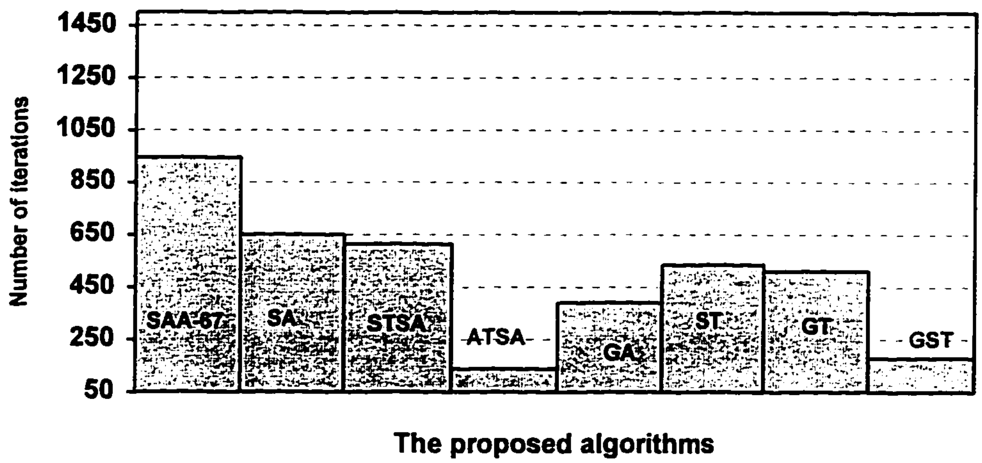


Fig.(7.4) Number of iterations of different proposed algorithms for Example 2

7.3 RESULTS OF EXAMPLE 3

Table (7.3) and Figs.(7.5) and (7.6) present the daily operating costs and the number of iterations (or generations of the GA based algorithms), of the seven proposed algorithms and of the SAA-67 method, for Example 3.

In this example, the proposed algorithms also give better results than those of the SAA-67. The daily cost saving amount, referred to the SAA-67 results, ranges between 1169 (by SA) and 3682 (by GST) which is equivalent to a percentage saving of 0.17 and 0.55 respectively.

Among all the proposed algorithms, for Example 3, it is obvious that the GST algorithm provides the best quality of solution, while the GT algorithm converges faster.

Considering the algorithms based on a single method, e.g. SA, STSA, ATSA, and GA, based on the results of this example, it is noted that the ATSA performance is the best among these algorithms as far as the objective function value is concerned, while the GA converges faster.

It can also be concluded that the results of the hybrid algorithms, e.g. ST, GT, and GST are better than the results of the algorithms based on a single method.

As far as the proposed hybrid algorithms are concerned, it is also found that the GST algorithm is the best in the quality of solution and the GT convergence is better.

Table (7.3) Comparison of different algorithms for Example 3

Algorithm	Daily Operating Cost ($)	Amount of Daily Saving	% Saving	No. of Iterations or Generations
SAA [67]	663833	0	0	2864
SAA	662664	1169	0.17	2361
STSA	662583	1250	0.19	3900
ATSA	660864	2969	0.45	2547
GA	661439	2394	0.36	985
ST	660596	3237	0.48	2829
GT	660412	3421	0.51	623
GST	660151	3682	0.55	762

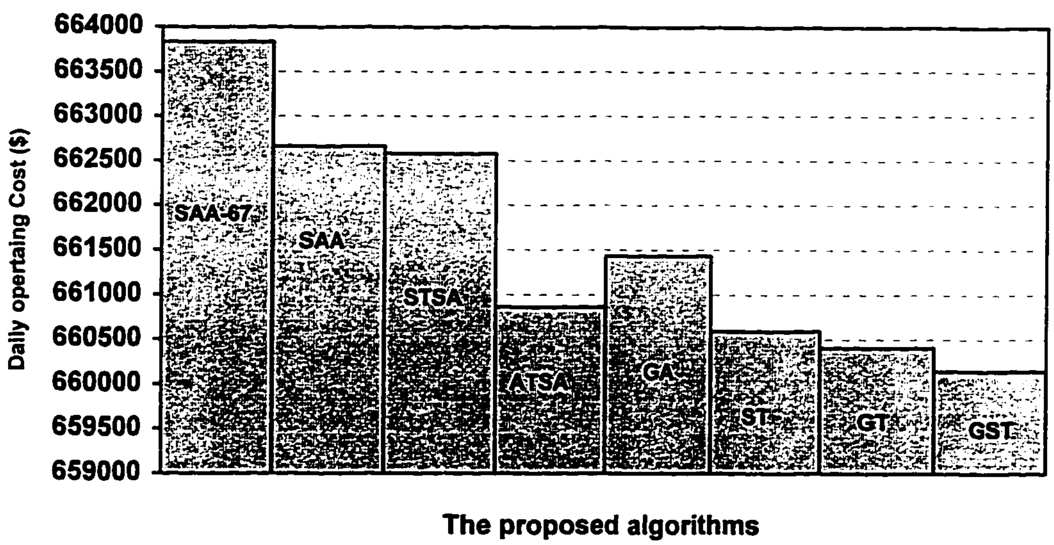


Fig.(7.5) Operating costs of different proposed algorithms for Example 3

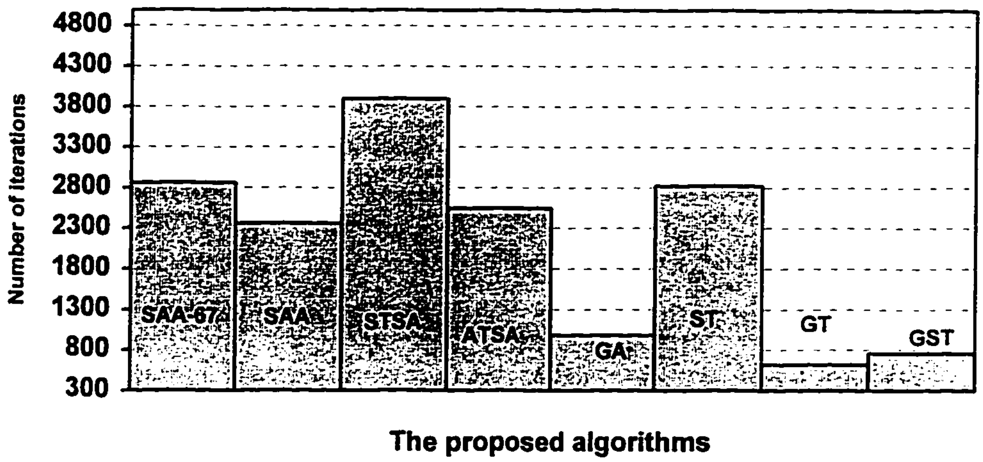


Fig.(7.6) Number of iterations of different proposed algorithms for Example 3

7.4 SUMMARY

This chapter is intended for the comparison between the results of our proposed algorithms and the results of other methods reported in the literature. Three systems extracted from the literature are considered. The effectiveness of our proposed algorithms is demonstrated.

It can be concluded that, all the proposed algorithms outperform the results reported in the literature using the LR , the IP and the SA method.

Considering the algorithms based on a single technique, e.g., SAA, STSA, ATSA, and GA, we conclude that the performance of both the SAA and the GA are the best.

It can also be concluded that the results of the hybrid algorithms, e.g. ST, GT, and GST are superior to the results of the algorithms based on a single method.

Among the proposed hybrid algorithms, it is also found that the GST algorithm provides a better quality of solution and also a faster speed of convergence.

In the following chapter the applicability of the proposed algorithms is examined through the solution of the SCECO-East power system.

CHAPTER EIGHT

APPLICATION TO THE SCECO EAST SYSTEM

8.1 OVERVIEW

In the course of testing and demonstrating the effectiveness of the proposed algorithms, presented in the last four chapters for solving the UCP, we have applied two of these algorithms to a real power system data. The selected sample of data is extracted from the Saudi Consolidated Electric Company in the Eastern Province (SCECO-East).

The Saudi Consolidated Electric Company, which is a local electric utility in the Eastern Province, (SCECO-East), is in charge of generating, transmitting and distributing the electric power to the consumers in the Eastern Province of Saudi Arabia.

Based on the performance of the proposed algorithms discussed in the last chapters, the SA (as one of the algorithms based on individual method) and the GT (as representative of the hybrids) algorithms have been selected to solve the UCP of this real system.

Modified versions of the two algorithms [126], ST and GT, are implemented to suit the selected real system data. These modifications are summarized as follows:

- One of the important constraints, in real power systems, is the unit derating due to atmospheric temperature changes. To include such a practical constraint, a routine is added to modify a unit capacity according to the given temperatures.

- The system data related to the cost function of the generating units are available in the form of heat rate curves, so we have used a curve fitting routine to calculate the cost function parameters of the generating units, A, B and C given in equation. 2.1. as a quadratic function.
- Due to the shortage of data needed to accurately calculate the start-up costs of the generating units, they were not taken into consideration.

8.2 NUMERICAL RESULTS

The modified versions of the two proposed SA and GT algorithms are applied to a sample of data extracted from the SCECO-East system. The sample of data includes 24 units with different capacities. A typical daily load curve from the winter season is chosen and modified to suit the selected sample of units. The amount of spinning reserve is taken as 400 MW in all hours. The derating constraints are also taken into account and considered as a function of the weather temperature. The full data of the sample of SCECO-East system is given in Appendix C.

In the Systems Operation Department of the SCECO-East company, a computer program has been implemented to solve the UCP. The program is based on the Dynamic Programming approach with some heuristic rules [17]. In this program, the generating units are clustered into related groups so as to minimize the number of unit combinations which must be tested without precluding the optimal path. The program also solves the EDP using piecewise linear cost curves. The implemented DP computer program has been applied to the selected sample of data. The total operating cost of the resulted schedule was found to be $1,375,865.

8.2.1 RESULTS OF THE SA ALGORITHM

For the SA algorithm[126], a number of tests on the performance of the algorithm have been carried out on the SCECO-East Example to find the most suitable SA parameters settings for the cooling schedule. Preliminary trials have been performed to set the SA algorithm parameters. The following parameter settings have been chosen: initial value of the control parameter Cp_o=7000, δ=0.3, chain length=150 and the maximum number of chains=1000, and ε=1e-6,.

Tables (8.1), (8.2) and (8.3) present the detailed results of the SA algorithm. Table(8.1) and (8.2) show the load sharing in MW among the committed units in the 24 hours. Table (8.3) gives the hourly load demand and the corresponding economic dispatch costs, start-up costs, and total operating cost. The start up costs are zero, since there is no available data for calculation.

8.2.2 RESULTS OF THE GT ALGORITHM

Different runs were carried out for the GT algorithm to find the most suitable parameters setting for the system. The following control parameters have been chosen: population size=50, crossover rate=0.8, mutation rate=0.3, number of elite copies=2, number of new solutions generated by tabu search=5, and the maximum number of generations=500, and tabu list size=7.

The detailed results of applying the GT algorithm for the SCECO-East Example are shown in Tables (8.4),(8.5) and (8.6).

8.3 COMPARISON OF THE RESULTS

Table (8.7) and Fig.(8.1) show the comparison of results obtained by the Dynamic Programming-based algorithm, implemented in SCECO-East, and our proposed SA and GT algorithms.

Referring to the results presented in Section(8.2), a considerable improvement in the daily operating cost was achieved using both SA and GT algorithms. The saving in cost for the SA algorithm is 4.1% which amounts to approximately $56,474 daily. For the GT algorithm the daily cost saving is $75,761 which is equivalent to 5.5%. These results basically show the effectiveness of the proposed algorithms and their applicability to real power systems.

Table(8.1) Power sharing (MW) of the SA algorithm for the SCECO-East Example(units 1-12)

HR	Unit Number 1	2	3	4	5	6	7	8	9	10	11	12
1	525.9	525.9	316.8	316.5	316.5	316.5	0.00	0.00	0.00	50.79	48.76	0.00
2	519.1	519.1	312.3	312.3	312.3	312.3	0.00	0.00	0.00	50.79	48.76	0.00
3	525.8	525.8	316.7	316.5	316.5	316.5	0.00	0.00	0.00	0.00	0.00	0.00
4	512.6	512.6	308.1	308.2	308.2	308.2	0.00	0.00	0.00	0.00	0.00	0.00
5	513.2	513.2	308.5	308.7	308.7	308.7	0.00	0.00	0.00	0.00	0.00	0.00
6	473.5	473.5	282.6	283.8	283.8	283.8	0.00	0.00	0.00	0.00	0.00	0.00
7	495.5	495.5	297.0	297.6	297.6	297.6	0.00	0.00	0.00	0.00	0.00	0.00
8	507.0	507.0	304.5	304.8	304.8	304.8	0.00	0.00	0.00	0.00	48.76	0.00
9	523.3	523.3	315.1	315.0	315.0	315.0	0.00	0.00	0.00	0.00	48.76	0.00
10	523.2	523.2	315.0	314.9	314.9	314.9	0.00	0.00	0.00	50.79	48.76	0.00
11	530.3	530.3	319.7	319.3	319.3	319.3	0.00	0.00	0.00	50.79	48.76	0.00
12	517.1	517.1	311.0	311.0	311.0	311.0	0.00	0.00	0.00	50.79	48.76	0.00
13	519.2	519.2	312.4	312.4	312.4	312.4	0.00	0.00	0.00	50.79	48.76	0.00
14	528.1	528.1	318.2	317.9	317.9	317.9	0.00	0.00	0.00	50.79	48.76	0.00
15	530.3	530.3	319.6	319.3	319.3	319.3	0.00	0.00	0.00	50.79	48.76	0.00
16	521.2	521.2	313.7	313.6	313.6	313.6	0.00	0.00	0.00	50.79	48.76	0.00
17	507.9	507.9	305.1	305.4	305.4	305.4	0.00	0.00	0.00	50.79	48.76	0.00
18	526.9	526.9	317.4	317.2	317.2	317.2	0.00	0.00	0.00	50.79	48.76	47.71
19	513.6	513.6	308.8	308.9	308.9	308.9	0.00	0.00	0.00	50.79	48.76	47.71
20	515.8	515.8	310.2	310.3	310.3	310.3	0.00	0.00	0.00	50.79	48.76	47.71
21	522.5	522.5	314.6	314.4	314.4	314.4	0.00	0.00	0.00	50.79	48.76	47.71
22	520.3	520.3	313.1	313.1	313.1	313.1	0.00	0.00	0.00	50.79	48.76	47.71
23	504.8	504.8	303.0	303.4	303.4	303.4	0.00	0.00	0.00	50.79	48.76	47.71
24	487.1	487.1	291.5	292.4	292.4	292.4	0.00	0.00	0.00	50.79	48.76	47.71

Table(8.2) Power sharing (MW) of the SA algorithm for the SCECO-East Example(units 13-24)

HR	Unit Number											
	13	14	15	16	17	18	19	20	21	22	23	24
1	0.00	49.19	0.00	0.00	0.00	0.00	43.07	0.00	0.00	31.07	0.00	38.73
2	0.00	0.00	0.00	0.00	0.00	0.00	43.07	0.00	0.00	31.07	0.00	38.73
3	0.00	0.00	0.00	0.00	0.00	0.00	43.07	0.00	0.00	0.00	0.00	38.73
4	0.00	0.00	0.00	0.00	0.00	0.00	43.07	0.00	0.00	0.00	0.00	38.73
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38.73
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38.73
7	0.00	0.00	50.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38.73
8	49.15	0.00	50.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38.73
9	49.15	0.00	50.24	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	0.00
10	49.15	0.00	50.24	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	0.00
11	49.15	49.19	0.00	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	38.73
12	49.15	49.19	0.00	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	38.73
13	49.15	49.19	50.24	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	38.73
14	49.15	49.19	50.24	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	38.73
15	49.15	49.19	50.24	0.00	0.00	0.00	43.07	43.03	0.00	0.00	38.73	38.73
16	49.15	49.19	50.24	0.00	0.00	0.00	43.07	43.03	31.04	0.00	38.73	38.73
17	49.15	49.19	50.24	0.00	0.00	0.00	43.07	43.03	31.04	0.00	38.73	38.73
18	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
19	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
20	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
21	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
22	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
23	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73
24	49.15	49.19	50.24	55.41	0.00	0.00	43.07	43.03	31.04	31.07	38.73	38.73

Table (8.3) Load demand and hourly costs of the SA algorithm for the SCECO-East Example

HR	LOAD	ED-COST	ST-COST	T-COST
1	2.58E+03	5.38E+04	0.00E+00	5.38E+04
2	2.50E+03	5.19E+04	0.00E+00	5.19E+04
3	2.40E+03	4.94E+04	0.00E+00	4.94E+04
4	2.34E+03	4.81E+04	0.00E+00	4.81E+04
5	2.30E+03	4.71E+04	0.00E+00	4.71E+04
6	2.12E+03	4.33E+04	0.00E+00	4.33E+04
7	2.27E+03	4.66E+04	0.00E+00	4.66E+04
8	2.42E+03	5.01E+04	0.00E+00	5.01E+04
9	2.58E+03	5.37E+04	0.00E+00	5.37E+04
10	2.63E+03	5.49E+04	0.00E+00	5.49E+04
11	2.70E+03	5.65E+04	0.00E+00	5.65E+04
12	2.64E+03	5.52E+04	0.00E+00	5.52E+04
13	2.70E+03	5.67E+04	0.00E+00	5.67E+04
14	2.74E+03	5.75E+04	0.00E+00	5.75E+04
15	2.75E+03	5.77E+04	0.00E+00	5.77E+04
16	2.74E+03	5.76E+04	0.00E+00	5.76E+04
17	2.68E+03	5.63E+04	0.00E+00	5.63E+04
18	2.90E+03	6.17E+04	0.00E+00	6.17E+04
19	2.84E+03	6.04E+04	0.00E+00	6.04E+04
20	2.85E+03	6.07E+04	0.00E+00	6.07E+04
21	2.88E+03	6.13E+04	0.00E+00	6.13E+04
22	2.87E+03	6.11E+04	0.00E+00	6.11E+04
23	2.80E+03	5.96E+04	0.00E+00	5.96E+04
24	2.72E+03	5.79E+04	0.00E+00	5.79E+04

Daily operating cost = $1,319,391

Table(8.4) Power sharing (MW) of the GT algorithm for the SCECO-East Example(units 1-12)

HR	Unit Number											
	1	2	3	4	5	6	7	8	9	10	11	12
1	527.5	527.5	317.8	317.5	317.5	317.5	0.00	0.00	0.00	0.00	0.00	0.00
2	523.2	523.2	315.0	314.9	314.9	314.9	0.00	0.00	0.00	0.00	0.00	0.00
3	514.6	514.6	309.4	309.5	309.5	309.5	0.00	0.00	0.00	0.00	0.00	0.00
4	513.2	513.2	308.5	308.7	308.7	308.7	0.00	0.00	0.00	0.00	0.00	0.00
5	504.4	504.4	302.7	303.1	303.1	303.1	0.00	0.00	0.00	0.00	0.00	0.00
6	465.0	465.0	277.0	278.5	278.5	278.5	0.00	0.00	0.00	0.00	0.00	0.00
7	497.8	497.8	298.4	299.0	299.0	299.0	0.00	0.00	0.00	0.00	0.00	0.00
8	513.5	513.5	308.7	308.8	308.8	308.8	0.00	0.00	0.00	0.00	0.00	0.00
9	523.5	523.5	315.2	315.1	315.1	315.1	0.00	0.00	0.00	0.00	0.00	0.00
10	519.6	519.6	312.6	312.6	312.6	312.6	0.00	0.00	0.00	0.00	0.00	0.00
11	521.6	521.6	314.0	313.9	313.9	313.9	0.00	0.00	0.00	0.00	0.00	0.00
12	508.3	508.3	305.3	305.6	305.6	305.6	0.00	0.00	0.00	0.00	0.00	0.00
13	521.6	521.6	314.0	313.9	313.9	313.9	0.00	0.00	0.00	0.00	0.00	0.00
14	530.4	530.4	319.7	319.4	319.4	319.4	0.00	0.00	0.00	0.00	0.00	0.00
15	517.4	517.4	311.2	311.3	311.3	311.3	0.00	0.00	0.00	0.00	0.00	0.00
16	515.2	515.2	309.8	309.9	309.9	309.9	0.00	0.00	0.00	0.00	0.00	0.00
17	501.9	501.9	301.2	301.6	301.6	301.6	0.00	0.00	0.00	0.00	0.00	0.00
18	523.7	523.7	315.3	315.2	315.2	315.2	0.00	0.00	0.00	0.00	0.00	0.00
19	510.4	510.4	306.7	306.9	306.9	306.9	0.00	0.00	0.00	0.00	0.00	0.00
20	512.6	512.6	308.1	308.3	308.3	308.3	0.00	0.00	0.00	0.00	0.00	0.00
21	519.2	519.2	312.4	312.4	312.4	312.4	0.00	0.00	0.00	0.00	0.00	0.00
22	517.0	517.0	311.0	311.0	311.0	311.0	0.00	0.00	0.00	0.00	0.00	0.00
23	530.2	530.2	319.6	319.3	319.3	319.3	0.00	0.00	0.00	0.00	0.00	0.00
24	512.6	512.6	308.1	308.2	308.2	308.2	0.00	0.00	0.00	0.00	0.00	0.00

Table(8.5) Power sharing (MW) of the GT algorithm for the SCECO-East Example(units 13-24)

HR	Unit Number											
	13	14	15	16	17	18	19	20	21	22	23	24
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	53.89	60.85	60.85
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	53.89	0.00	60.85
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	53.89	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	77.20	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78.80	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	78.84	78.80	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	0.00
10	0.00	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	0.00
11	0.00	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
12	0.00	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
13	0.00	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
14	0.00	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
15	0.00	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
16	0.00	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
17	0.00	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	0.00	60.85	60.85
18	67.77	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	53.89	60.85	60.85
19	67.77	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	53.89	60.85	60.85
20	67.77	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	53.89	60.85	60.85
21	67.77	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	53.89	60.85	60.85
22	67.77	67.81	68.86	0.00	0.00	0.00	78.84	78.80	53.86	53.89	60.85	60.85
23	67.77	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	53.89	0.00	60.85
24	67.77	67.81	0.00	0.00	0.00	0.00	78.84	78.80	53.86	53.89	0.00	60.85

Table (8.6) Load demand and hourly costs of the GT algorithm for the SCECO-East Example

HR	LOAD	ED-COST	ST-COST	T-COST
1	2.58E+03	5.31E+04	0.00E+00	5.31E+04
2	2.50E+03	5.14E+04	0.00E+00	5.14E+04
3	2.40E+03	4.92E+04	0.00E+00	4.92E+04
4	2.34E+03	4.79E+04	0.00E+00	4.79E+04
5	2.30E+03	4.70E+04	0.00E+00	4.70E+04
6	2.12E+03	4.32E+04	0.00E+00	4.32E+04
7	2.27E+03	4.64E+04	0.00E+00	4.64E+04
8	2.42E+03	4.96E+04	0.00E+00	4.96E+04
9	2.58E+03	5.31E+04	0.00E+00	5.31E+04
10	2.63E+03	5.43E+04	0.00E+00	5.43E+04
11	2.70E+03	5.58E+04	0.00E+00	5.58E+04
12	2.64E+03	5.45E+04	0.00E+00	5.45E+04
13	2.70E+03	5.58E+04	0.00E+00	5.58E+04
14	2.74E+03	5.67E+04	0.00E+00	5.67E+04
15	2.75E+03	5.70E+04	0.00E+00	5.70E+04
16	2.74E+03	5.68E+04	0.00E+00	5.68E+04
17	2.68E+03	5.55E+04	0.00E+00	5.55E+04
18	2.90E+03	6.03E+04	0.00E+00	6.03E+04
19	2.84E+03	5.91E+04	0.00E+00	5.91E+04
20	2.85E+03	5.93E+04	0.00E+00	5.93E+04
21	2.88E+03	5.99E+04	0.00E+00	5.99E+04
22	2.87E+03	5.97E+04	0.00E+00	5.97E+04
23	2.80E+03	5.81E+04	0.00E+00	5.81E+04
24	2.72E+03	5.64E+04	0.00E+00	5.64E+04

Daily operating cost = $1300104

Table(8.7) Comparison between DP, SA, and the GT Algorithm for the SCECO-East Example

	Example	DP	SAA	GT
Total Cost ($)	SCECO-East	1375865	1319391	1300104
% Cost Saving	,,	0	4.1	5.5
No. of Iterations/Generations	,,	-	903	500

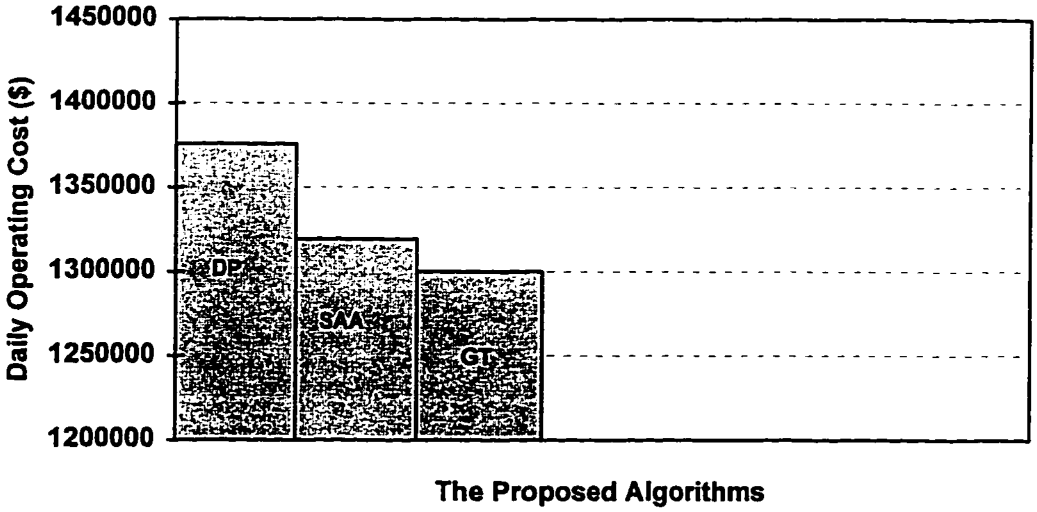


Fig.(8.1) Comparison of the operating costs of different algorithms for the SCECO-East Example

8.4 SUMMARY

To further demonstrate the applicability of the proposed algorithms, a modified version of two selected algorithms, ST and GT, are implemented to solve a sample of data extracted from the SCECO-East system[126].

A comparison of the results obtained using the SA and the GT algorithms and a dynamic programming-based algorithm, used by SCECO-East is presented.

A considerable improvement in the daily operating cost is achieved using both SA and GT algorithms. These results basically show the effectiveness of the proposed algorithms and their adaptability to real power systems.

The next chapter introduces the conclusions and the recommendations for future work in this area.

CHAPTER NINE

CONCLUSIONS AND RECOMMENDATIONS

The present study deals with thermal generation scheduling, which could be considered as the major part of the whole scheduling problem of hydrothermal power systems. The scheduling problem of thermal generating units is really a complex optimization problem. It can be considered as two linked optimization problems as it comprises the solution of both the unit commitment and economic dispatch problems.

The unit commitment problem is a combinatorial optimization problem with very difficult constraints. The economic dispatch problem is a nonlinear optimization problem.

Research has been focused on UCP techniques with various degrees of optimality, efficiency, and ability to handle difficult constraints. Exhaustive enumeration is the only technique that can find the optimal solution, because it looks at every possible combination of the generating units that fulfill the load demand and satisfy the system constraints at all hours of the scheduling horizon. Basically, the high dimension of the possible solution space, when a complete enumeration method is used, makes it impracticable.

The growing interest for the application of artificial intelligence techniques to power systems engineering has introduced the potentials of using this state-of-the art technology in the thermal generation scheduling of electric power systems.

AI techniques, unlike strict mathematical methods, have the apparent ability to adapt to nonlinearities and discontinuities commonly found in power systems. The best known algorithms in this class include evolution programming, genetic algorithms, simulated annealing, tabu search, and neural networks.

9.1 CONCLUSIONS

Considering the implementations and the results of the proposed AI-based algorithms for solving the UCP, the following conclusions could be extracted:

1) In the present study, seven different AI-based algorithms have been developed to solve the UCP. Two of these algorithms, presented in chapters 3 and 5, are considered as new implementations of existing techniques to solve the UCP. These algorithms are:

* A simulated annealing algorithm , and
* A genetic algorithm approach.

The other five proposed algorithms, presented in chapters 4 and 6, are original and are applied for the first time to solve the UCP. These algorithms are:

* A simple tabu search algorithm,
* An advanced tabu search algorithm,
* A hybrid of the simulated annealing and tabu search algorithms,
* A hybrid of genetic and tabu search algorithms, and
* A hybrid of genetic, simulated annealing, and tabu search algorithms.

2) As a first step to solve the UCP, some modifications to the existing problem formulation have been made to be more generalized. An augmented model including all the problem constraints is presented in chapter (2).

3) A major step in the course of solving the UCP is the solution of the EDP. In this regard, an efficient and fast nonlinear programming routine is implemented and tested. The routine is based on a linear complementary algorithm for solving the quadratic programming problems as a linear program in a tableau form. Comparing the results of the proposed routine, it is found that the results obtained are more accurate than that obtained using an IMSL quadratic programming routine. The application of this routine to the EDP is a new contribution.

4) The corner stone in solving the combinatorial optimization problems is to have good rules for finding randomly feasible trial solutions from an existing feasible solution, in an efficient way. Because of the constraints in the UCP, this is not a simple matter. The most difficult constraints to satisfy are the minimum up/down times. A major contribution of this work is the implementation of new rules to get randomly feasible solutions faster.

5) A new implementation of a Simulated Annealing Algorithm for solving the UCP has been implemented in Chapter (3). Two cooling schedules are implemented and compared, namely, the Polynomial-Time and Kirk's cooling schedules. The proposed SAA implementation has some advantages over those presented in the literature. The starting solution is randomly generated, while the SAA-67 starts with a solution obtained using a priority-list method

that could be considered as a suboptimal solution. Our SAA provides a methodology for determining the initial control parameter at which virtually all trial solutions are accepted. Moreover, temperature decrement is governed using statistics generated during the search (Polynomial-Time cooling schedule). Regarding the comparison of results, our SAA gives better solutions quality and takes about half the execution time that is taken by SAA-67.

6) A novel application of tabu search method for the UCP is introduced in Chapter (4). A Simple Tabu Search Algorithm for the UCP is proposed. The STSA is based on the short-term memory procedure of the TS method. TS is characterized by its ability to escape local optima by using a short-term memory of recent moves. Different approaches for constructing the tabu list for the UCP are presented and tested. Furthermore, different tabu list sizes are tried to find the most reasonable one for the examples under study. The solved examples showed that operating costs obtained by the proposed algorithm are better than the operating costs reported in the literature. The successful implementation presented in this work highlights the importance of the TS approach as a powerful tool for solving such difficult combinatorial optimization problems.

7) As an improvement to the proposed STSA, an Advanced Tabu Search Algorithm (ATSA) is implemented for the UCP. The ATSA is based on more advanced tabu search procedures, in addition to the basic short-term memory function. The advanced procedures include intermediate memory and long

term memory procedures and strategic oscillation. The operating costs obtained by the proposed algorithm are better than those obtained using short term memory and are also superior to the results reported in the literature.

8) A new Genetic Algorithm approach for the UCP is proposed in Chapter (5). The proposed algorithm differs from other GA implementations in three respects. First, the UCP solution is coded using a mix between binary and decimal representations, thus saving computer memory as well as computation time. Second, the fitness function is constructed only from the total operating cost without including penalty terms. Third, to improve the speed of calculations, the reproduction operators are implemented in a novel way. The crossover operation is applied to the populations in its decimal form. The mutation operator is used to induce new solutions in the population. Fourth, to improve the fine local tuning capabilities of the proposed GA a special mutation operator is designed based on a local search procedure. As a result, a basic advantage of the proposed GA is the high speed of convergence, the good quality of solutions obtained related to other methods, and the reduced memory space required.

9) A novel hybrid algorithm, based on the integration of the main features of both the SA and TS methods, is proposed. The main idea in the proposed ST algorithm is to use the TS algorithm to prevent the repeated solutions from being accepted by the SA. This saves time and improves the quality of solution obtained. In the SA part, the polynomial-time cooling schedule is used. In the TS part, the short-term memory procedures are implemented. The

results show an improvement in the quality of solutions obtained compared with results obtained by either the SA or TS algorithms.

10) A novel hybrid algorithm based on integrating genetic algorithm and tabu search methods to solve the UCP is proposed. In the GT algorithm, the implementation of the GA part is similar to that described in (8). Moreover, tabu search is implemented within the GA to induce new members during the reproduction phase of the GA, thus escaping from local minimum and avoiding premature convergence. The effectiveness of the proposed algorithm in solving the UCP has been demonstrated through the numerical examples. It was found that the overall performance of the GT algorithm is superior to the performance of both the individual GA and TS algorithms. The GT algorithm converges faster and gives a better quality of solution.

11) A novel hybrid algorithm for solving the UCP is proposed. The algorithm integrates the main features of the most commonly used artificial intelligence methods for solving combinatorial optimization problems: genetic algorithm, tabu search, and simulated annealing. The algorithm is mainly structured around the GA, while the TS method is used to generate new members in the GA population. The SA algorithm is used to accelerate the convergence of the GA via testing all the GA members after each reproduction of a new population. The implementation of the GA in the proposed algorithm is similar to that described in (8). The TS implementation is based on the short term memory procedures. In the SA part, a simple cooling schedule is used to simplify and speed up the calculations. The results obtained are superior to those reported in the literature using Lagrangian Relaxation and Integer Programming methods. Moreover, the obtained results (using the proposed algorithm) are better than those obtained using the individual SA, TS or GA in Chapters [3,4&5].

12) To emphasize the effectiveness and the applicability of our proposed algorithms to real power systems, two of the proposed algorithms, SA and GT, have been applied to a sample system extracted from the SCECO-East system. A considerable improvement in the daily operating cost was achieved using both the SA and the GT algorithms. Compared to the results of a dynamic programming-based heuristic approach, the daily savings in cost for the SA algorithm is 4.1% which amounts to approximately $56,474. For the GT algorithm the daily cost savings is $75,761 which is equivalent to 5.5%.

13) It can be concluded that the seven proposed AI-based algorithms have several common advantages, these are as follows:

* They do not need a complicated mathematical model of the problem under study.
* The ease of implementation.
* They could find a high quality solution that does not strongly depend on the choice of the initial solution.
* The ease of including any type of constraints that could be difficult to state mathematically.

* The algorithm could start with any given solution and try to improve it.

9.2 RECOMMENDATIONS FOR FUTURE WORK

1) As a general suggestion for a future work in the area of AI-based methods, parallel processing of any of the proposed algorithms could lead to the following results:

    * Reduce the computation time,
    * Increase the possibility of solving a large scale power system in reasonable time,
    * Explore wider solution space, and
    * A better quality of solution could be obtained.

2) More elaboration is required in the direction of creating a fast and efficient mechanism for randomly generating feasible trial solutions.

3) Our proposed GA approach is promising. More refinement could be done to reduce computation time, which may increase the efficiency of the algorithm.

4) Hybridization of GA with other artificial intelligence techniques, e.g., neural networks, fuzzy logic and expert systems, could improve the solution quality and accelerate the calculations.

5) Some other constraints could also be taken into consideration such as: transmission losses, transmission lines capacities and emission constraints.

6) The hydrothermal scheduling problem can be solved by including the hydro generation system into the proposed algorithms.

Some of these topics are already under investigations by the author.

APPENDIX A

SOLVING THE ECONOMIC DISPATCH PROBLEM

A.1 THE LINEAR COMPLEMENTARY PROBLEM [6]

A.1.1 DEFINITION 1

Let M be a given p x p matrix, and q be a given p vector. The *linear complementary problem* is to find vectors w and z such that:

$$w - Mz = q \qquad \text{(A-1)}$$

$$w, z \geq 0 \qquad \text{(A-2)}$$

$$w^t z = 0 \qquad \text{(A-3)}$$

Here, (w_i, z_i) is a pair of *complementary variables.* A solution (w,z) to the above system is called a *complementary basic feasible solution* if (w,z) is a basic feasible solution to (A-1), and (A-2) and if one variable of the pair (w_i, z_i) is basic for i = 1,....p.

A.1.2 SOLVING THE LINEAR COMPLEMENTARY PROBLEM

If q is nonnegative then we immediately have a solution that satisfies (A-1), (A-2) and (A-3) by letting w=q and z=0. If q<0, however, a new column 1 and an artificial variable are introduced, leading to the following system, where 1 is a vector of ones.

$$w - Mz - 1z_0 = q \qquad \text{(A-4)}$$

$$w, z \geq 0, \; z_o \geq 0 \tag{A-5}$$

$$w^t z = 0 \tag{A-6}$$

Letting $z_o = \text{maximum } \{-q_i : 1 \leq i \leq p\}$, $z=0$, and $w = q + 1z_o$, we obtain a starting solution to the above system. Through a sequence of pivots, to be specified later that satisfies (A-4)-(A-6), we attempt to derive the artificial variable z_o to level zero, thus obtaining a solution to the linear complementary problem.

A.1.3 DEFINITION 2

Consider the system defined by (A-4)-(A-6). A feasible solution (w, z, z_o) to this system is called an *almost complementary basic feasible* solution if:

1. (w, z, z_o) is a basic feasible solution to (A-4) and (A-5).
2. Neither w_s nor z_s are basic, for some $s \in \{1, \ldots\ldots, p\}$.
3. z_o is basic and exactly one variable from the complementary pair (w_j, z_j) is basic, for $j=1, \ldots\ldots, p$ and $j \neq s$.

Given an almost complementary basic feasible solution (w, z, z_o), where w, z, and z_o) are nonbasic, an *adjacent almost complementary basic feasible solution* (w^*, z^*, z_o^*) is obtained by introducing either w_s or z_s in the basic if pivoting derives a variable other than z_o from the basis.

From the above definition it is clear that each almost complementary basic feasible solution has, at most, two adjacent almost complementary basic feasible solutions. If increasing w_s or z_s derives z_o out of the basis or produces a ray of the set in (A-4) and

(A-5), then we have less than two adjacent almost complementary basic feasible solutions.

A.1.4 SUMMARY OF LEMKE'S COMPLEMENTARY PIVOTING ALGORITHM

We summarize below a complementary pivoting algorithm credited to Lemke (1968) for solving the linear complementary problem. Introducing the artificial variable z_o, the algorithm moves among adjacent almost complementary basic feasible solutions until either a complementary basic feasible solution is obtained or a direction indicating unboundness of the region defined by (A-4) through (A-6) is found. As will be shown later, under certain assumptions on the matrix M, the algorithm converges in a finite number of steps with a complementary basic feasible solution.

A.1.4.1 Initialization Step:

If $q \geq 0$, stop; $(w,z)=(q,0)$ is a complementary basic feasible solution. Otherwise, display the system defined by (A-4) and (A-5) in a tableau format.

Let $-q_s = \text{maximum}\{-q_i : 1 \leq i \leq p\}$, and update the tableau by pivoting at row s and z_o column. Thus the basic variables z_o and w_i for j=1,....,p and $j \neq s$ are nonnegative. Let $y_s = z_s$, and go to the main step.

A.1.4.2 Main Step

1. Let d_s be the updated column in the current tableau under the variable y_s. I$d_s \leq 0$, go to step 4. Otherwise, determine the index r by the following

minimum ratio test, where $\bar{q}$ is the updated right-hand-side column denoting the values of the basic variables.

$$\frac{\bar{q}_r}{d_{rs}} = \underset{1 \le i \le p}{\text{minimum}} \left\{ \frac{\bar{q}_i}{d_{is}} : d_{is} > 0 \right\} \quad \text{(A-7)}$$

If the basic variable at row r is z_o, go to step 3. Otherwise, go to step 2.

2. The basic variable at row r is either w_l or z_l, for some $l \neq s$. The variable y_s enters the basic, then the tableau is updated by pivoting at row r and the y_s column. If the variable that just left the basis is z_l, then let $y_s = w_l$. Go to step 1.
3. Here, y_s enters the basis, and z_o leaves the basis. Pivot at the y_s column and the z_o row, producing a complementary basic feasible solution. Stop.
4. Stop with ray termination. A ray $R=\{(w, z, z_o) + \lambda d : \lambda \ge 0\}$ is found such that every point in R satisfies (A-4)-(A-6). Here (w, z, z_o) is the almost complementary basic feasible solution associated with the last tableau, and d is an extreme direction of the set defined by (A-4) and (A-5) and has a 1 at the row corresponding to y_s, - d, at the rows of the current basic variables and zero everywhere else.

A.1.5 FINITE CONVERGENCE OF THE COMPLEMENTARY PIVOTING ALGORITHM

The following lemma shows that the algorithm must stop in a finite number of iterations, either with a complementary basic feasible solution or with ray termination. Under certain conditions of the matrix M, the algorithm stops with a complementary basic feasible solution.

A.1.5.1 LEMMA

Suppose that each almost basic feasible solution of the system (A-4)-(A-6) is nondegenerate; that is, each basic variable is positive. Then none of the points generated by the complementary pivoting algorithm is repeated, and furthermore, the algorithm must stop in a finite number of steps.

A.1.6 TABLEAU FORM

Based on the discussions in the previous sections, we can size the tableau for any problem in the linear complementary formulation as follows:

Let:

N: is the number of variables,

K: is the number of constraints rather than the nonnegativity, and

L= number of variables + number of constraints = N + K

The tableau size is then L x (2L+2). The following augmented matrix shows a tableau for N=2, and K=1, then L=3

$$
\begin{array}{c}
|\text{------ w ------}|\text{------ z ------}|\text{-- } z_0 \text{ --}|\text{R.H.S}| \\
\left[\begin{array}{ccc:c:c:c}
1 & 0 & 0 & & -1 & b \\
0 & 1 & 0 & -M & -1 & c1 \\
0 & 0 & 1 & & -1 & c2
\end{array}\right] \\
|\text{------ L ------}|\text{------ L ------}|\text{------ 2 ------}|
\end{array}
\quad \text{(A-8)}
$$

and

$$M = \begin{bmatrix} 0 & -A \\ A^t & H \end{bmatrix} \quad \text{(A-9)}$$

where:

M is a L x L matrix, e.g. 3 x 3,

A is N x K matrix, e.g. 2 x 1,

H is N x N matrix, e.g. 2 x 2,

b is the right-hand sides of the constraints, and

c1, c2 are the coeficient the variables in the the linear part of the objective function.

A.2 QUADRATIC PROGRAMMING

Quadratic programming represents a special class of nonlinear programming in which the objective function is quadratic and the constraints are linear. In this section, we show that the Kuhn-Tucker conditions of a quadratic programming problem reduce to a linear complementary problem. Thus, the complementary pivoting algorithm described in Section A.1 can be used for solving a quadratic programming problem.

A.2.1 THE KUHN-TUCKER SYSTEM

Consider the following quadratic programming problem:

Minimize $\quad c^t x + \frac{1}{2} x^t H x \qquad$ (A-10)

Subject to $\quad Ax \le b \qquad$ (A-11)

$$x \le 0 \qquad \text{(A-12)}$$

where c is an n vector, b is an m vector, A is an m x n matrix, and H is an n x n symmetric matrix. Denoting the Lagrangian multiplier vectors of the two groups of constraints (A-11) and (A-12) by u and v respectively, and denoting the vector of slack variables by y, the **Kuhn-Tucker conditions could be written as:**

$$Ax + y = b \qquad \text{(A-13)}$$

$$-Hx - A^t u + v = c \qquad \text{(A-14)}$$

$$x^t v = 0 \qquad \text{(A-15)}$$

$$u^t y = 0 \qquad \text{(A-16)}$$

$$x, y, u, v \ge 0 \qquad \text{(A-17)}$$

Now letting

$$M = \begin{bmatrix} 0 & -A \\ A^t & H \end{bmatrix} \qquad \text{(A-18)}$$

$$q = \begin{bmatrix} b \\ c \end{bmatrix} \qquad \text{(A-19)}$$

$$w = \begin{bmatrix} y \\ v \end{bmatrix} \qquad \text{(A-20)}$$

$$z = \begin{bmatrix} u \\ x \end{bmatrix} \tag{A-21}$$

The Kuhn-Tucker conditions could be rewritten as the linear complementary problem :

$$w - Mz = q, \tag{A-22}$$

$$w^t z = 0, \tag{A-23}$$

$$w, z \geq 0. \tag{A-24}$$

In a matrix form:

$$\begin{bmatrix} y \\ v \end{bmatrix} - \begin{bmatrix} 0 & -A \\ A^t & H \end{bmatrix} \begin{bmatrix} u \\ x \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} \tag{A-25}$$

$$[y \quad v] \begin{bmatrix} u \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{A-26}$$

Thus, the complementary pivoting algorithm discussed in Section A.1 can be used to find the Kuhn-Tucker point of the quadratic programming problem. The theorem gives proof that the linear complementary algorithm converges in a finite number of iterations with a Kuhn-Tucker point.

A.2.2 THEOREM

Consider the problem of minimizing $c^t x + \frac{1}{2} x^t H x$, *subject to* $Ax \leq b$, $x \leq 0$. *Suppose that the feasible region is not empty. Further, suppose that the complementary pivoting algorithm described in Section A.1 is used in an attempt to find a solution to the Kuhn-Tucker system in (A-22) to (A-24). In the absence of degeneracy, under any of the* *following conditions, the algorithm stops in a finite number of iterations with a Kuhn-Tucker point.*

1. H is positive semidefinite and c=0

2. H is positive definite.

3. H has nonnegative elements with positive diagonal elements.

Furthermore, if H is positive semidefinite, then ray termination implies that the optimal solution is unbounded.

APPENDIX B

THEORY OF THE SIMULATED ANNEALING METHOD

B.1 METROPLIS ALGORITHM

As far back as 1953, Metropolis, Rosenbluth, Rosenbluth, Teller and Teller (1953) introduced a simple algorithm evolution of a solid in a heat bath to thermal equilibrium [98]. The algorithm introduced by these authors is based on Monte carlo techniques and generates a sequence of the solid in the following way. Given a current state i of the solid with energy E_i , then a subsequent state j is generated by applying a perturbation mechanism which transforms the current state into a next state by a small distortion, for instance, by displacement of a particle. The energy of the next state is E_j . If the energy difference, $E_j - E_i$, is less than or equal to 0, the state j is accepted as the current state. If the energy difference is greater than 0, the state j is accepted with a certain probability which is given by [94];

$$\exp\left(\frac{E_i - E_j}{K_B T}\right) \tag{B.1}$$

Where T denotes the *temperature* of the heat bath and K_B a physical constant known as the *Boltzman constant.* The acceptance rule described above is known as the *Metropolis criterion* and the algorithm that goes with it is known as the *Metropolis algorithm.* If the lowering of temperature is done sufficiently slowly, the solid can reach thermal equilibrium at each temperature. In the Metropolis algorithm this is achieved by generating a large number of transitions at a given temperature value. Thermal equilibrium is characterized by the Boltzman distribution [99]. This distribution gives the probability of the solid being in state i with energy E_i at temperature T, and is given by

$$P_T\{X=i\}=\frac{1}{Z(T)}\exp\left(\frac{-E_i}{K_B T}\right) \tag{B.2}$$

where X is a stochastic variable denoting the current state of the solid. Z(T) is the partition function, which is defined as;

$$Z(T)=\sum_j \exp\left(\frac{-E_j}{K_B T}\right) \tag{B.3}$$

B.2 THE SIMULATED ANNEALING ALGORITHM

Let *(s,f)* denote an instance of a combinatorial optimization problem and i and j two solutions with cost f(i) and f(j) respectively. Then the acceptance criterion determines when j is accepted from i by applying the following acceptance probability:

$$P_c\{\text{accept } j\}=\begin{cases}1 & \text{if } f(j)\leq f(i)\\ \exp\left(\dfrac{f(i)-f(j)}{c_p}\right) & \text{if } f(j)>f(i)\end{cases} \tag{B.4}$$

where c_p denotes the control parameter.

B.2.1 CONJECTURE

Given an instance (s,f) of a combinatorial optimization problem and a suitable neighborhood structure then, after a sufficiently large number of transitions at a fixed *value* c_p, *applying the acceptance probability of* (B.4), *the simulated annealing algorithm will find a solution* $i \in s$ *with a probability equal to*

$$P_{c_p}\{X=i\} \stackrel{\Delta}{=} q_i(c_p) = \frac{1}{N_0(c_p)}\exp\left(\frac{-f(i)}{c_p}\right) \qquad \text{(B.5)}$$

where X *is a stochastic variable denoting the current solution obtained by the simulated annealing algorithm and;*

$$N_0(c_p) = \sum_{j \in s}\exp\left(\frac{-f(i)}{c_p}\right) \qquad \text{(B.6)}$$

denotes a normalization constant.

The probability distribution of (B.5) is called the *stationary or equilibrium distribution* and is the equivalent of the Boltzman distribution of (B.2). The normalization constant of (B.6) is the equivalent of the partition function of (B.3).

By using the stationary distribution of (B.5), a set of useful quantities can be defined for optimization problems in a way similar to that for physical systems. Here we define the following quantities [94]:

(i) The expected cost E_{cp} (f) at equilibrium is defined as:

$$\begin{aligned} E_{c_p} &\stackrel{\Delta}{=} \langle f \rangle_{c_p} \\ &= \sum_{i \in s} f(i) P_{c_p}\{X=i \\ &= \sum_{i \in s} f(i)\, q_i(c_p) \end{aligned} \qquad \text{(B.7)}$$

(ii) The expected square cost $E_{C_p}(f^2)$ at equilibrium is defined as

$$E_{C_p} \stackrel{\Delta}{=} \langle f^2 \rangle_{C_p} = \sum_{i \in S} f^2(i) P_{C_p}\{X = i\} = \sum_{i \in S} f^2(i)\, q_i(c_p) \tag{B.8}$$

(iii) The variance $Var_{C_p}(f)$ of the cost at equilibrium is defined as:

$$Var_{C_p}(f) \stackrel{\Delta}{=} \sigma^2{}_{C_p} = \sum_{i \in S} (f(i) - E_{C_p}(f))^2 P_{C_p}\{X = i\} = \sum_{i \in S} (f(i) - \langle f \rangle_{C_p})^2\, q_i(c_p) = \langle f^2 \rangle_{C_p} - \langle f \rangle^2{}_{C_p} \tag{B.9}$$

(iv) The entropy at equilibrium is defined as

$$S_{C_p} = -\sum_{i \in S} q_i(c_p) \ln q_i(c_p) \tag{B.10}$$

B.2.2 COROLLARY

Let the stationary distribution be given by (B.5), *then the following relations hold;*

$$\frac{\partial}{\partial c_p} \langle f \rangle_{C_p} = \frac{\sigma^2_{C_p}}{c_p^2} \tag{B.11}$$

$$\frac{\partial}{\partial c_p} S_{C_p} = \frac{\sigma^2_{C_p}}{c_p^3} \tag{B.12}$$

B.3 COOLING SCHEDULE

A finite-time implementation of a simulated annealing algorithm can be realized by generating homogeneous Markov chains of finite length for a finite sequence of descending values of the control parameter. To achieve this, one must specify a set of parameters that governs the convergence of the algorithm. These parameters are combined in a so-called cooling schedule [94].

The parameters of a cooling schedule are

- an initial value of the control parameter
- a decrement function for decreasing the control parameter
- a final value of the control parameter specified by the stopping criterion, and
- a finite length of each homogenous Markov chain

The search for adequate cooling schedules has been the subject of study in many papers [94].

B.3.1 QUASI EQUILIBRIUM

Let L_k denote the length of the K^{th} Markov chain and ${}^{c}p_k$ cp the corresponding value of the control parameter. The quasi equilibrium is achieved if a (L_k , ${}^{c}p_k$) , i.e., the probability distribution of the solution after L_k trials of the K^{th} Markov chain, is " sufficiently close" to q(${}^{c}p_k$), the stationary distribution of ${}^{c}p_k$, defined by (B.5), (B.6), i.e.,

$$\|a(L_k, c_{p_k}) - q(c_{p_k})\| < \varepsilon \qquad \text{(B.13)}$$

for some specified positive value of ε.

Requiring (B.13) to hold for arbitrarily small values of ε implies that a number of transitions is needed which is quadratic in the size of the solution space. This leads for most combinatorial optimization problems to an exponential-time execution of the simulated annealing algorithm. Thus, for a practical application of the algorithm, we need a less rigid quantification of quasi equilibrium than that of condition (B.13). In the literature, this has led to different interpretations of the concept of quasi equilibrium resulting in a rich variety of cooling schedules.

B.3.2 A POLYNOMIAL-TIME COOLING SCHEDULE

In this section, a cooling schedule presented by Aarts and Van Laarhoven [100,101,102] is discussed. This cooling schedule leads to a Polynomial-Time execution of the SAA, but it can not give any guarantee for the deviation in cost between the final solution obtained by the algorithm and the optimal cost. The different parameters of the cooling schedule are determined based on the statistics calculated during the search. In the following we describe these parameters [94].

B.3.2.1 Initial Value of the Control Parameter c_p

The initial value of c_p, is obtained from the requirement that initially, virtually all proposed trial solutions should be accepted. Assume that a sequence of m trials is generated at a certain value of c_p. Let m_1 denote the number of trials for which the objective function value does not exceed the respective current solution. Let $m_2 = m - m_1$.

It can be shown that the acceptance ratio, X can be approximated by [94]:

$$X \approx (m_1 + m_2.\exp(-\overset{(+)}{\Delta f} / c_p)) / (m_1 + m_2) \qquad \text{(B.14)}$$

Where, $\overset{(+)}{\Delta f}$ is the average difference in cost over the m_2 cost-increasing trials.

From which the new temperature c_p is

$$c_p = \overset{(+)}{\Delta f} / \ln(m_2 / (m_2.X - m_1(1 - X)) \qquad \text{(B.15)}$$

B.3.2.2 Decrement of the Control Parameter

Assume that the condition for quasi equilibrium given in (B.13) may be replaced by

$$\left\| q(c_{p_k}) - q(c_{p_{k+1}}) \right\| < \varepsilon \quad \forall k \geq 0 \qquad \text{(B.16)}$$

Thus for two successive values of the control parameter we want the stationary distribution to be "close". This can be quantified by requiring that

$$\forall i \in s: \frac{1}{1+\delta} < \frac{q_i(c_{p_k})}{q_i(c_{p_{K+1}})} < 1+\delta, k = 0,1,\ldots \qquad \text{(B.17)}$$

for some small positive number δ that can be related to ε of (B.16).

The following theorem provides sufficient conditions to satisfy (B.17)

Theorem

Let $q(c_{p_k})$ *be the stationary distribution for the homogeneous Markov chain associated with the simulated annealing algorithm with components given by* (B.5) and (B.6) *and let* cp_k *and* cp_{k+1} *be two successive values of the control parameter with* $cp_{k+1} < cp_k$, *then the inequalities of (B.15) are satisfied if the following condition holds:*

$$\forall i \in s: \frac{\exp(\frac{-\delta_i}{c_{p_K}})}{\exp(\frac{-\delta_i}{c_{p_{K+1}}})} < 1+\delta, K = 0,1,\ldots \qquad \text{(B.18)}$$

where: $\delta i = f(i) - f_{opt}$

which can be rewritten to give the following condition on the two subsequent values of the control parameter:

$$\forall i \in s:\ c_{p_{k+1}} > \frac{c_{p_K}}{1 + \dfrac{c_{p_K} \, Ln(1+\delta)}{f(i) - f_{opt}}}, K = 0,1,... \qquad (B.19)$$

By introducing some simplifications to the condition (B.18), condition (B.19) could be rewritten as follows [94]:

$$c_{p_{k+1}} > \frac{c_{p_K}}{1 + \dfrac{c_{p_K} \, Ln(1+\delta)}{<f>_{c_{p_k}} - f_{opt} + 3\sigma_{c_{p_k}}}}, K = 0,1,... \qquad (B.20)$$

For many instances of combinatorial optimization problems the value of f_{opt} is not known. However, the average value and the spreading of the cost function typically exhibit a similar behaviour as a function of the control parameter. Hence, we argue that $<f>_{c_{p_k}} - f_{opt} + 3\sigma_{c_{p_k}}$ can be replaced by $3\sigma_{c_{p_k}}$ and that the omission of the term $<f>_{c_{p_k}} - f_{opt}$ is counterbalnaced by choosing samaller values of δ. Thus, (B.20) can be replaced by the following expression:

$$c_{p_{k+1}} = \frac{c_{p_K}}{1 + \dfrac{c_{p_K} \, Ln(1+\delta)}{3\sigma_{c_{p_k}}}}, K = 0,1,... \qquad (B.21)$$

The amount by which the value of Cp is decreased by the decrement function of (B.21) is determined by the value of δ, hereafter called the *distance parameter*. Small values of δ lead to small decrements; large values of δ lead to large decrements in Cp. Typical values of δ are between 0.1 and 0.5.

B.3.2.3 The Final value of the control parameter

Termination in the Polynomial-Time cooling schedule is based on an extrapolation of the expected average cost, $<f>_{Cp_k}$, for Cp_k approaches zero value. Let

$$\Delta <f>_{Cp} = <f>_{Cp} - f_{opt} \qquad \text{(B.22)}$$

Then, execution of the algorithm is terminated if $\Delta <f>_{Cp_o}$, the expected cost at Cp_0. For sufficiently large values of Cp_0 we have $<f>_{Cp_o} \approx <f>_\infty$. Hence, we may approximate $\Delta <f>_{Cp}$ for $Cp << 1$ by:

$$\Delta <f>_{Cp} \approx Cp . \frac{\partial <f>_{Cp}}{\partial Cp} \qquad \text{(B.23)}$$

Hence, the algorithm may be terminated if, for some value of k, we have [94] :

$$\left. \frac{Cp^k}{\langle f \rangle_\infty} . \frac{\partial \langle f \rangle_{Cp}}{\partial Cp} \right|_{Cp = Cp_k} < \varepsilon \qquad \text{(B.24)}$$

where, ε is some small positive number, referred to as the *stop parameter*. In our implementation $\varepsilon = 0.00001$.

(d) The Length of Markov Chains

In [94], it is concluded that the decrement function of the control parameter, (B.21), requires only a 'small' number of trial solutions to rapidly approach the stationary distribution for a given next value of the control parameter. The word 'small' can be specified as the number of transitions for which the algorithm has a sufficiently large probability of visiting at least a major part of the neighborhood of a given solution. In general, a chain length of more than 100 transitions is reasonable [94]. In our implementation good results have been reached at a chain length of 150.

APPENDIX C

DATA OF THE SOLVED EXAMPLES

C.1 DATA OF EXAMPLE 1

This example is taken from reference [29].

C1.1 DAILY LOAD DEMAND

HR	LOAD (MW)	SPINNING RESERVE (MW)
1	1025	85
2	1000	85
3	900	65
4	850	55
5	1025	85
6	1400	110
7	1970	165
8	2400	190
9	2850	210
10	3150	230
11	3300	250
12	3400	275
13	3275	240
14	2950	210
15	2700	200
16	2550	195
17	2725	200
18	3200	220
19	3300	250
20	2900	210
21	2125	170
22	1650	130
23	1300	100
24	1150	90

C.1.2 PRODUCTION COST FUNCTION PARAMETERS

Unit	A ($/ MW².HR)	B ($ / MW.HR)	C ($ / HR)
1	0.00113	9.023	820
2	0.0016	7.654	400
3	0.00147	8.752	600
4	0.0015	8.431	420
5	0.00234	9.223	540
6	0.00515	7.054	175
7	0.00131	9.121	600
8	0.00171	7.762	400
9	0.00128	8.162	725
10	0.00452	8.149	200

C.1.3 MINIMUM AND MAXIXMUM OUTPUT LIMITS OF UNITS

Unit	P_{min} (MW)	P_{max} (MW)
1	300	1000
2	130	400
3	165	600
4	130	420
5	225	700
6	50	200
7	250	750
8	110	375
9	275	850
10	75	250

C.1.4 MINIMUM UP/DOWN TIMES AND START-UP COST PARAMETRS

Unit	T_{up}	T_{down}	ST_o	B_1	B_2	B_o
1	5	4	2050	1	0.25	825
2	3	2	1460	1	0.333	650
3	2	4	2100	1	0.25	950
4	1	3	1480	1	0.25	650
5	4	5	2100	1	0.333	900
6	2	2	1360	1	0.5	750
7	3	4	2300	1	0.25	950
8	1	3	1370	1	0.333	550
9	4	3	2200	1	0.25	950
10	2	1	1180	1	0.5	625

C.2 DATA OF EXAMPLE 2

This example is taken from reference [41].

C2.1 DAILY LOAD DEMAND

HR	LOAD (MW)
1	1459
2	1372
3	1299
4	1285
5	1271
6	1314
7	1372
8	1314
9	1271
10	1242
11	1197
12	1182
13	1154
14	1138
15	1124
16	1095
17	1066
18	1037
19	993
20	978
21	963
22	1022
23	1081
24	1459

* spinning reserve = 10%

C.2.2 PRODUCTION COST FUNCTION PARAMETERS

Unit	A ($ / MW².HR)	B ($ / MW.HR)	C ($ / HR)
1	0.0051	1.4	15
2	0.00396	1.5	25
3	0.00393	1.35	40
4	0.00382	1.4	32
5	0.00212	1.54	29
6	0.00261	1.35	72
7	0.00127	1.3954	105
8	0.00135	1.3285	100
9	0.00289	1.2643	49
10	0.00148	1.2136	82

C.2.3 MINIMUM AND MAXIXMUM OUTPUT LIMITS OF UNITS

Unit	P_{min} (MW)	P_{max} (MW)
1	15	60
2	20	80
3	30	100
4	25	120
5	50	150
6	75	280
7	250	520
8	50	150
9	120	320
10	75	200

C.2.4 MINIMUM UP/DOWN TIMES AND START-UP COST PARAMETRS

Unit	T_{up}	T_{down}	ST_o	B_1	B_2	B_o
1	2	5	85	0.588	0.2	0
2	2	5	101	0.594	0.2	0
3	2	5	114	0.57	0.2	0
4	2	5	94	0.65	0.18	0
5	2	5	113	0.639	0.18	0
6	2	5	176	0.568	0.15	0
7	2	5	267	0.749	0.09	0
8	2	5	282	0.749	0.09	0
9	2	5	187	0.617	0.13	0
10	2	5	227	0.641	0.11	0

C.3 DATA OF EXAMPLE 3

This example is taken from reference [62,63].

C3.1 DAILY LOAD DEMAND

HR	LOAD (MW)
1	1820
2	1800
3	1720
4	1700
5	1750
6	1910
7	2050
8	2400
9	2600
10	2600
11	2620
12	2580
13	2590
14	2570
15	2500
16	2350
17	2390
18	2480
19	2580
20	2620
21	2600
22	2480
23	2150
24	1900

C.3.2 PRODUCTION COST FUNCTION PARAMETERS

Unit	A ($ / MW^2.HR)	B ($ / MW.HR)	C ($ / HR)
1	0	25.547	24.389
2	0	25.675	24.411
3	0	25.803	24.638
4	0	25.932	24.76
5	0	26.061	24.888
6	0	37.551	117.755
7	0	37.664	118.108
8	0	37.777	118.458
9	0	37.89	118.821
10	0	13.327	81.136
11	0	13.354	81.298
12	0	13.38	81.464
13	0	13.407	81.626
14	0	18	217.895
15	0	18.1	218.335
16	0	18.2	218.775
17	0	10.695	142.735
18	0	10.715	143.029
19	0	10.737	143.318
20	0	10.758	143.597
21	0	23	259.171
22	0	23.1	259.649
23	0	23.2	260.176
24	0	10.862	177.057
25	0	5.492	202.5
26	0	5.503	202.91

C.3.3 MINIMUM AND MAXIXMUM OUTPUT LIMITS OF UNITS

Unit	P_{min} (MW)	P_{max} (MW)
1	2.4	12
2	2.4	12
3	2.4	12
4	2.4	12
5	2.4	12
6	4	20
7	4	20
8	4	20
9	4	20
10	15.2	76
11	15.2	76
12	15.2	76
13	15.2	76
14	25	100
15	25	100
16	25	100
17	54.25	155
18	54.25	155
19	54.25	155
20	54.25	155
21	68.95	197
22	68.95	197
23	68.95	197
24	140	350
25	140	350
26	140	350

C.3.4 MINIMUM UP/DOWN TIMES AND START-UP COST PARAMETRS

Unit	T_{up}	T_{down}	ST_o	B_1	B_2	B_o
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	30	0	0	0
7	0	0	30	0	0	0
8	0	0	30	0	0	0
9	0	0	30	0	0	0
10	3	2	80	0	0	0
11	3	2	80	0	0	0
12	3	2	80	0	0	0
13	3	2	80	0	0	0
14	4	2	100	0	0	0
15	4	2	100	0	0	0
16	4	2	100	0	0	0
17	5	3	200	0	0	0
18	5	3	200	0	0	0
19	5	3	200	0	0	0
20	5	3	200	0	0	0
21	5	4	300	0	0	0
22	5	4	300	0	0	0
23	5	4	300	0	0	0
24	8	5	500	0	0	0
25	8	5	800	0	0	0
26	8	5	800	0	0	0

C.4 DATA OF THE SCECO-EAST EXAMPLE

C4.1 DAILY LOAD DEMAND

HR	LOAD (MW)
1	2580
2	2500
3	2400
4	2340
5	2300
6	2120
7	2270
8	2420
9	2580
10	2630
11	2700
12	2640
13	2700
14	2740
15	2750
16	2740
17	2680
18	2900
19	2840
20	2850
21	2880
22	2870
23	2800
24	2720

* Spinning reserve = 400 MW

C4.2 UNITS DATA

Unit No.	A ($/MW^2.HR	B ($/MW.HR)	C ($ / HR)	P_{min} (MW)	P_{max} (MW)	T_{up} (HR)	T_{down} (HR)
1	7.62E-03	13.728	605.779	250	625	8	8
2	7.62E-03	13.728	605.779	250	625	8	8
3	1.17E-02	14.346	1186.087	180	400	8	8
4	1.22E-02	14.020	1235.064	190	400	8	8
5	1.22E-02	14.020	1235.064	190	400	8	8
6	1.22E-02	14.020	1235.064	190	400	8	8
7	3.29E-02	15.240	671.691	33	75	4	2
8	3.26E-02	15.266	671.186	33	77	4	2
9	3.24E-02	15.278	670.939	33	78	4	2
10	4.95E-02	14.087	383.963	33	69	4	2
11	4.90E-02	14.126	383.200	33	67	4	2
12	4.55E-02	14.410	377.669	20	66	4	2
13	4.46E-02	14.483	376.284	15	68	4	2
14	4.46E-02	14.483	376.291	15	68	4	2
15	4.49E-02	14.468	376.461	15	69	4	2
16	1.07E-02	16.793	633.426	20	81	4	2
17	1.08E-02	16.782	633.584	20	82	4	2
18	1.06E-02	16.795	633.408	20	81	4	2
19	4.60E-02	13.710	317.410	15	79	4	2
20	4.60E-02	13.710	317.410	15	79	4	2
21	4.60E-02	13.710	317.410	10	54	4	2
22	4.60E-02	13.710	317.417	15	54	4	2
23	4.60E-02	13.709	317.426	15	61	4	2
24	4.60E-02	13.709	317.426	15	61	4	2

NOMENCLATURE

The following notation is used throughout the thesis:

A_i, B_i, C_i	Cost function parameters of unit i (\$/MW^2.HR, \$/MW.HR, \$/HR)
Cp	Control parameter (temperature) of the cooling schedule.
Cp^k	Control parameter value at iteration k.
Cp_o	Initial value of the control parameter.
D_i, E_i	Start-up cost coefficients for unit (\$).
$F_{it}(P_{it})$	Production cost of unit i at time t (\$/HR).
F_T	Total operating cost over the scheduling horizon (\$)
F_i^k	The total operating cost for a current solution i at iteration k
N	Number of available generating units.
P_{it}	Output power from unit i at time t (MW).
P_i^k	Output power from all units for a current solution i iteration k.
P_{min_i}	Unit i minimum generation limit (MW).
P_{max_i}	Unit i maximum generation limit (MW).
PD_t	System peak demand at hour t (MW).
R_t	System reserve at hour t (MW).
ST_{it}	Start-up cost of unit i at hour t.
SH_{it}	Shut-down cost of unit i at hour t.
So_i	Unit i cold start-up cost .

T	Scheduling time horizon, (24 HRs).
T_{up_i}	Unit i minimum up time.
T_{down_i}	Unit i minimum down time.
T_{on_i}	Duration during which unit i is continuously ON.
T_{off_i}	Duration during which unit i is continuously OFF.
T_{shut_i}	Instant of shut down of a unit i.
T_{start_i}	Instant of start-up of a unit i.
$U(0,1)$	The uniform distribution with parameters 0, and 1
$UD(a,b$	The discrete uniform distribution with parameters a and b.
U_{it}	Unit i status at hour t.
	= 1 if the unit is ON and 0 if OFF at hour t.
U_i^k	Unit status matrix for a current solution i at iteration k.
V_{it}	Unit i start-up status at hour t.
	= 1 if the unit is started at hour t and 0 otherwise.
V_i^k	Unit start-up/shut-down matrix for a current solution i at iteration k
W_{it}	Unit i shut-down status at hour t.
	= 1 if the unit is turned off at hour t and 0 otherwise
σ	is the standard deviation of the cost values generated at the $k\underline{th}$ Markov chain, corresponding to c_p^k in the simulated annealing algorithm.
δ	is a constant called distance parameter, used in in the simulated annealing algorithm.

LIST OF ABBREVIATIONS

SA:	simulated annealing
TS:	tabu search
GAs:	genetic algorithms
GA:	genetic algorithm
SAA	simulated annealing algorithm
SAA-67	simulated annealing algorithm in the reference number 67
STSA	simple tabu search algorithm
ATSA	advanced tabu search algorithm
ST	hybrid algorithm of simulated annealing and tabu search
GT	hybrid algorithm of genetic and tabu search
GTS	hybrid algorithm of genetic, tabu search and simulated annealing
UCP	unit commitment problem
EDP	economic dispatch problem
UCT	unit commitment table
DP	dynamic programming
LR	Lagrangian relaxation
IP	integer programming
MIP	mixed integer programming
PL	priority list
ES	expert systems

NN	neural networks
STM	short term memory
ITM	intermediate term memory
LTM	long term memory
SO	strategic oscillation
TL	tabu list
AV	aspiration level
Z	tabu list size
GUTL	generating unit tabu list
NPOP	number of population in the genetic algorithm
ED-COST	economic dispatch cost per hour
ST-COST	start-up cost per hour
T-COST	total cost per hour
HR	hour
MW	mega watt
SCECO-East	The Saudi Consolidated Electric Company in the Eastern Province.

REFERENCES

[1] Allen J. Wood and B. F. Wollenberg, " Power Generation, Operation, and Control", *John Wiley & Sons Ltd.* 1984.

[2] M. E. El-Hawary and G. S. Christensen, " Optimal Economic Operation of Electric Power Systems", *Academic Press, Inc.* 1979.

[3] I. J. Nagrath and D. P. Koathari, "Modern Power System Analysis", *TATA McGraw Hill Publishing Co.*, 1987.

[4] Roy Billintion, "Power System Reliability Evaluation", *Gordon and Breach Science Publishers, New York, London, Paris,* 1980.

[5] Billy E. Gillets, "Introduction to Operations Research", *McGraw Hill, Inc.* 1976.

[6] Mokhtar S. Bazaraa, and C. M. Shetty, "Nonlinear Programming, Theory and Algorithms", *John Willey & Sons, Inc.* 1979.

[7] Robert E. Larson, "State Incremental Dynamic Programming", *California,* June, 1968.

[8] G. B. Sheble', and G. N. Fahd, "Unit Commitment Literature Synopsis", *IEEE Trans. on Power Systems, Vol. 9, No. 1, February* 1994, pp. 128-135.

[9] P. G. Lowery, " Generating Unit Commitment by Dynamic Programming", *IEEE Trans. on PAS-85, No. 5*, pp. 422-426, May 1966.

[10] J. D. Guy, " Security Constrained Unit Commitment", *IEEE Trans. on PAS-90*, No. 3, pp. 1385-1389, May/June 1971.

[11] A. K. Ayoub, and A. D. Patton, " Optimal Thermal Generating Unit Commitment ", *IEEE Trans. on PAS, Vol. PAS-90, No. 4*, July/August 1971, pp. 1752-1756.

[12] C. K. Pang, and H. C. Chen, " Optimal Short-Term Thermal Unit Commitment ", *IEEE Trans. on PAS , Vol. PAS-102, No. 1*, July/August, 1976, pp. 1336-1346.

[13] C. K. Pang, G. B. Sheble', and F. Albuyeh, " Evaluation of Dynamic Programming Based Methods and Multiple Area Representation for Thermal

Unit Commitments ", *IEEE Trans. on PAS*, *Vol. PAS-100, No. 3*, March 1981, pp. 1212-1218.

[14] P. P. Van De Bosch, and G. Honderd," A Solution of the Unit Commitment Problem Via Decomposition and Dynamic Programming ", *IEEE Trans. on Power Apparatus and Systems, Vol. PAS-194,* No. 7, July 1985, pp. 1684-1690.

[15] S. D. Bond, and B. Fox, "Optimal Thermal Unit Scheduling Using Improved Dynamic Programming Algorithms", *IEE Proc. C,* 1986, 133,(1), pp. 1-5.

[16] G. L. Kusic, and H. A. Putnam," Dispatch and Unit Commitment Including Commonly Owned Units ", *IEEE Trans. on Power Apparatus and Systems, Vol. PAS-104, No. 9,* September 1985, pp. 2408-2412.

[17] W. L. Snyder, H. D. Powell, and J. C. Rayburn, "Dynamic Programming Approach to Unit Commitment" , *IEEE Trans. on Power Systems , Vol. PWRS-2, No. 2*, May 1987, pp. 339-350.

[18] W. G. Hobbs, G. H. Warner, ,and G. B. Sheble', "An Enhanced Dynamic Programming Approach for Unit Commitment", *IEEE Trans. on Power Systems* , Vol. 3, No. 3, August 1988, pp. 1201-1205.

[19] Y-Y Hsu, C-C Su, C-C Liang, C-J Lin, and C-T Huang," Dynamic Security Constrained Multi-Area Unit Commitment ", *IEEE Trans. on Power Systems, Vol. 6, No. 3,* August 1991, pp. 1049-1055.

[20] A. H. Mantawy, " Optimal Scheduling of Thermal Generation in Electric Power Systems", A Master Thesis, *Ain Shams University, Cairo, Egypt,* 1988.

[21] Y. Al-kalaani, F. E. Villaseca, and F. Renovich, "Storage and Delivery Constrained Unit Commitment", *IEEE Trans. on Power Systems, Vol. 11, No. 2,* May 1996, pp. 1059-1066.

[22] Marshall L. Fisher "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I", *Operations Research, Vol. 21,* 1973, pp. 1114-1127

[23] J. A. Muckstadt, and Sherri A. Koenig, "An Application of Lagrangian Relaxation to Scheduling in Power-Generation Systems" , *Operation Research* , *Vol. 25, No. 3*, May-June 1977, pp. 387-403

[24] D. P. Bertsekas, G. S. Lauer, N. R. Sandell, and T. A. Posbergh, " Optimal Short-Term Scheduling of Large-Scale Power Systems ", *IEEE Trans. on Automatic Control, Vol. Ac-28, No. 1*, January 1983, pp. 1-11.

[25] A. Merlin, and P. Sandrin," A New Method for Unit Commitment At Electricite De France", *IEEE Trans. on PAS*, Vol. PAS-102, No. 5, May 1983, pp. 1218-1225.

[26] R. Nieva, A. Inda, and I. Guillen," Lagrangian Reduction of Search-Range for Large-Scale Unit Commitment ", *IEEE Trans. on Power Systems, Vol. PWRS-2,* May 1987, pp. 465-473.

[27] K. Aoki, T. Satoh, and M. Itoh, " Unit Commitment in A Large-Scale Power System Including Fuel Constrained Thermal and Pumped-Storage Hydro", *IEEE Trans. on Power Systems, Vol. PWRS-2,* November 1987, pp. 1077-1084.

[28] F. Zhuang and F. D. Galiana, " Towards A More Rigorous and Practical Unit Commitment by Lagrangian Relaxation", *IEEE Trans. on Power Systems , Vol. 3,* No. 2, May 1988, pp. 763-773.

[29] J. F. Bard, " Short-Term Scheduling of Thermal-Electric Generators Using Lagrangian Relaxation", *Operation Research , Vol. 36, No. 5*, September-October 1988, pp. 756-766.

[30] K. Aoki, M. Itoh, T. Satoh, K. Nara, and M. Kanezashi," Optimal Long-Term Commitment in Large-Scale Systems Including Fuel Constraints Thermal and Pumped-Storage Hydro ", *IEEE Trans. on Power Systems, Vol. 4, No. 3*, August 1989, pp. 1065-1073.

[31] S. Virmani, E. C. Adrian, K. Imhof, and S. Mukherjee, " Implementation of A Lagrangian Relaxation Based Unit Commitment Problem", *IEEE Trans. on Power Systems, Vol. 4, No. 4*, November 1989, pp. 1373-1379

[32] Slobodan Ruzic, and Nikola Rajakovic, "A New Approach for Solving Extended Unit Commitment Problem", *IEEE Trans. on Power Systems, Vol. 6, No. 1*, February 1991, pp. 269-275.

[33] C. Wang, and S. M. Shahidehpour, " Effects of Ramp-Rate Limits on Unit Commitment and Economic Dispatch ", *IEEE Trans. on Power Systems Vol. 8, No. 3,* August 1993, pp. 1341-1350.

[34] Ross Baldick, " The Generalized Unit Commitment Problem", *IEEE Trans. on Power Systems Vol. 10, No. 1,* February 1995, pp. 465-475.

[35] William L. Peterson, Steven R. Brammer, " A Capacity Based Lagrangian Relaxation Unit Commitment With Ramp Rate Constraints", *IEEE Trans. on Power Systems Vol. 10, No. 2,* May 1995, pp. 1077-1084.

[36] S. J. Wang, S. M. Shahidehpour, D. S. Kirschen, S. Mokhtari, and G. D. Irisarri, "Short-Term Generation Scheduling With Transmission and Environmental Constraints Using An Augmented Lagrangian Relaxation", *IEEE Trans. on Power Systems Vol. 10, No. 3,* August 1995, pp. 1294-1301.

[37] John J. Shaw, "A Direct Method for Security-Constrained Unit Commitment", *IEEE Trans. on Power Systems Vol. 10, No. 3,* August 1995, pp. 1329-11339.

[38] L. L. Garver, " Power Generation Scheduling by Integer Programming - Development of Theory", *A IEE Trans. No. 2,* February 1963, pp. 730-735.

[39] J. A. Muckstadt, and R. C. Wilson, " An Application of Mixed-Integer Programming Duality to Scheduling Thermal Generating Systems", *IEEE Trans. on PAS*-87, *Vol. 12,* December 1968, pp. 1968-1977.

[40] T. S. Dillon, K. W. Edwin, H. D. Kochs, and R. J Taud, "Integer Programming Approach to the Problem of Optimal Unit Commitment With Probabilistic Reserve", *IEEE Trans. on PAS, Vol. PAS-97, No. 6,* November/December 1978, pp. 2154-2166.

[41] A. Turgeon, " Optimal Scheduling of Thermal Generating Units ", *IEEE Trans. on Automatic Control, Vol. Ac-23, No. 6,* December 1978. pp. 1000-1005

[42] K. Hamam, Y. M. Hamam, K. S. Hindi, and A. Brameller, "Unit Commitment of Thermal Generation", *IEE Proc. C., 1980, 127, (1)*, pp. 3-8.

[43] G. S. Lauer, D. P. Bertsekas, N. R. Sandell, and T. A. Posbergh, "Solution of Large-Scale Optimal Unit Commitment Problems" , *IEEE Trans. on PAS, Vol. PAS-202, No. 1,* January 1982, pp. 79-86.

[44] A. I. Cohen, and M. Yoshimura, "A Branch-and-Bound Algorithm for Unit Commitment", *IEEE Trans. on PAS, Vol. PAS-102, No. 2*, February 1983, pp. 444-451.

[45] E. Handschin, and H. Slomski, "Unit Commitment in Thermal Power Systems With Long-Term Energy Constraints", *IEEE Trans. on Power Systems Vol. 5, No. 4*, November 1990, pp. 1470-1477.

[46] Jerome K. Delson, and S. M. Shaidehpour, "Linear Programming Applications to Power System Economics, Planning and Operations", *IEEE Trans. on Power Systems Vol. 7, No. 3,* August 1992, pp. 1155-1163.

[47] M. Piekutowski, and I. A. Rose, "A Linear Programming Method for Unit Commitment Incorporating Generator Configurations, Reserve and Flow Constraints", *IEEE Trans. on Power Apparatus and Systems Vol. PAS-104, No. 12*, December 1985, pp. 3510-3516.

[48] H. Brannlund, D. Sjelvgren, and Janis A. Bubenko, "Short Term Generation Scheduling With Security Constraints", *IEEE Trans. on Power Systems Vol. 3, No. 1*, February 1988, pp. 310-316.

[49] R. Zhu, C. Fu, and S. Rahamn, "Network Programming Technique for Unit Commitment", *Electrical Power & Energy Systems, Vol. 17, No. 2*, pp. 123-127, 1995.

[50] N. Chowdhury, and R. Billinton, "Unit Commitment in Interconnected Generating Systems Using A Probabilistic Technique", *IEEE Trans. on Power Systems Vol. 5, No. 5*, November 1990, pp. 1231-1238.

[51] Fred L. Lee, and Quan Chen, "Unit Commitment Risk With Sequential Rescheduling", *IEEE Trans. on Power Systems Vol. 6, No. 3*, August 1991, pp. 1017-1023.

[52] N. Chowdhury, "Energy Method of Spinning Reserve Assessment in Interconnected Generation Systems", *IEEE Trans. on Power Systems Vol. 8, No. 3*, August 1993, pp. 865-872.

[53] M. E. Khan, and R. Billinton, "Generating Unit Commitment in Composite Generation and Transmission Systems", *IEE Proceedings-C, Vol. 140, No. 5*, September 1993, pp. 404-410.

[54] Fred L. Lee, Janice Huang, and Rambabu Adapa, "Multi-Area Unit Commitment Via Sequential Method and A Dc Power Flow Network Model", *IEEE Trans. on Power Systems Vol. 9, No. 1*, February 1994, pp. 279-287.

[55] P. Carpentiet, G. Cohen, J.-C. Culioli, and A. Renaud, "Stochastic Optimization of Unit Commitment: A New Decomposition Framework", *IEEE Trans. on Power Systems ,Vol. 11, No. 2*, May 1996, pp. 1067-1073.

[56] S. Takriti, J. R. Birge, and E. Long, "A Stochastic Model for Unit Commitment Problem", A New Decomposition Framework", *IEEE Trans. on Power Systems ,Vol. 11, No. 3*, August 1996, pp. 1497-1508.

[57] Raymond R. Shoults, Show K. Chang, Steve Helmick, and W. Mack Grady, "A Practical Approach to Unit Commitment, Economic Dispatch and Savings Allocation for Multiple-Area Pool Operation With Import/Export

Constraints", *IEEE Trans. on Power Apparatus and Systems Vol. PAS-99, No. 2*, March/April 1980, pp. 625-635.

[58] F. N. Lee, "Short-Term Thermal Unit Commitment -- A New Method" , *IEEE Trans. on Power Systems* ,Vol. 3, No. 2, May 1988, pp. 421-428

[59] F. N. Lee. " The Application of Commitment Utilization Factor (CUF) to Thermal Unit Commitment", *IEEE Trans. on Power Systems ,Vol. 6, No. 2,* May 1991, pp. 691-698.

[60] G. B. Sheble'," Solution of the Unit Commitment Problem by the Method of Unit Periods", *IEEE Trans. on Power Systems, Vol. 5, No. 1*, February 1990, pp. 257-260.

[61] S. Mokhtari, J. Singh, and B. Wollenberg, " A Unit Commitment Expert System", *IEEE Trans. on Power Systems, Vol. 3, No. 1*, February 1988, pp. 272-277.

[62] Z. Ouyang, and S. M. Shahidepour, " Short-Term Unit Commitment Expert System ", *Electric Power System Research*, 20 (1990) 1-13.

[63] S. K. Tong, S. M. Shahidepour, and Z. Ouyang, " A Heuristic Short-Term Unit Commitment ", *IEEE Trans. on Power Systems, Vol. 6, No. 3*, August 1991, pp. 1210-1216.

[64] S. Li, S. M. Shahidehpour, and C. Wang, "Promoting the Application of Expert Systems in Short-Term Unit Commitment", *IEEE Trans. on Power Systems, Vol. 3, No. 1,* March 1993, pp. 286-292.

[65] M. H. Sendaula, S. K. Biswas, A. Eltom, C. Parten, and W. Kazibwe, " Simultaneous Solution of Unit Commitment and Dispatch Problems Using Artificial Neural Networks", *Electrical Power & Energy Systems, Vol. 15, No. 3*, 1993, pp. 193-199.

[66] Fugii H. Sasaki, M. Watanabe, J. Kubokkawa, and N. Yorino, "A Solution Method Using Neural Networks for the Generator Commitment Problem", *Electrical Engineering in Japan, Vol. 112, No. 7*, 1992, pp. 55-62.

[67] F. Zhuang , and F. D. Galiana, " Unit Commitment by Simulated Annealing ", *IEEE Trans. on Power Systems, Vol. 5, No. 1*, February 1990, pp. 311-318.

[68] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, " A Simulated Annealing Algorithm for Unit Commitment ", Accepted for publication in the *IEEE Trans. on Power Systems.*

[69] D. Dasgupta, and D. R. Mcgregor, " Thermal Unit Commitment Using Genetic Algorithms", *IEE Proc.-Gener. Transm. Distrib., Vol. 141, No. 5*, September 1994, pp. 459-465.

[70] X. Ma, A. A. El-Keib, R. E. Smith, and H. Ma, " A Genetic Algorithm Based Approach to Thermal Unit Commitment of Electric Power Systems", *Electric Power Systems Research* 34 (1995) 29-36.

[71] S. A. Kazarilis, A. G. Bakirtzis, and V. Petridis, " A Genetic Algorithm Solution to the Unit Commitment Problem", *IEEE Trans. on Power Systems, Vol. 11, No. 1,* February 1996, pp. 83-91.

[72] Pai-Chuan Yang, Hong-Tzer Yang, and Ching-Lien Huang, "Solving the Unit Commitment Problem With A Genetic Algorithm Through A Constraint Satisfaction Technique", *Electric Power Systems Research 37* (1996), 55-65.

[73] S. O. Orero and M. R. Irving, "A Genetic Algorithm for Generators Scheduling in Power Systems", *Electric Power & Energy Systems, Vol. 18, No. 1*, 1996, pp. 19-26.

[74] G. B. Sheble', T. T. Maifeld, K. Birttig, G. Fahd, and S. Fukurozaki-Coppinger, "Unit Commitment by Genetic Algorithm With Penalty Methods and A Comparison of Lagrangian Search and Genetic Algorithm-Economic Dispatch Example", *Electric Power & Energy Systems, Vol. 18, No. 6*, 1996, pp. 339-346.

[75] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim, "A Genetic Algorithm With Local Search For Unit Commitment", Accepted for presentation in the *International Conference on Intelligent Systems Applications to Power Systems, ISAP'97, Korea*

[76] James G. Waight, Farrokh Albuyeh, and Anjan Bose, "Scheduling of Generation and Reserve Margin Using Dynamic and Linear Programming", *IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 5*, May 1981, pp. 2226-2230.

[77] Hans P. Van Meeteren, "Scheduling of Generation and Allocation of Fuel, Using Dynamic and Linear Programming", *IEEE Trans. on Power Apparatus and Systems, Vol. PAS-103, No. 7*, July 1984, pp. 1562-1568.

[78] G. B. Sheble', and Leonard Grigsby, "Decision Analysis Solution of the Unit Commitment Problem", *Electric Power Systems Research 11* (1986), 85-93.

[79] E. Khodaverdian, A. Brameller, and R. M. Dunnett, "Semi-Rigorous Thermal Unit Commitment for Large Scale Electrical Power Systems", *IEE Proceedings, Vol. 133, Part-C, No. 4,* May 1986, pp. 157-164.

[80] Z. Ouyang, and S. M. Shahidehpour, "An Intelligent Dynamic Programming for Unit Commitment Application", *IEEE Trans. on Power Systems, Vol. 6, No. 3*, August 1991, pp. 1203-1209.

[81] Chung-Ching Su and Yuan-Yih Hsu, "Fuzzy Dynamic Programming: An Application to Unit Commitment", *IEEE Trans. on Power Systems, Vol. 6, No. 3*, August 1991, pp. 1231-1237.

[82] Z. Ouyang, and S. M. Shahidehpour, "A Hybrid Artificial Neural Network-Dynamic Programming Approach to Unit Commitment", *IEEE Trans. on Power Systems, Vol. 7, No. 1*, February 1992, pp. 236-242.

[83] Ruey-Husn, and Yuan-Yih Hsu, "A Hybrid Artificial Neural Network-Differential Dynamic Programming Approach for Short-Term Hydro Scheduling", *Electric Power Systems Research 33* (1995), 77-86.

[84] Adly A. Girgis, and Srinivas Varadan, "Unit Commitment Using Load forecasting Based on Artificial Neural Networks", *Electric Power Systems Research 32* (1995), 213-217.

[85] Z. Ouyang, and S. M. Shahidehpour, " A Multi-Stage Intelligent System for Unit Commitment", *IEEE Trans. on Power Systems, Vol. 7, No. 2*, May 1992, pp. 639-646.

[86] C. Wang, and S. M. Shahidehpour, "Effects of Ramp-Rate Limits on Unit Commitment and Economic Dispatch", *IEEE Trans. on Power Systems Vol. 8, No. 3,* , August 1993, pp. 1341-1350

[87] K. P. Wong and H. N Cheung, "Thermal Generator Scheduling Algorithm Based on Heuristic-Guided Depth-First Search", *IEE Proc. C,* 1990, 137,(1), pp. 33-43.

[88] K. P. Wong and K. P. Doan, "Artificial Intelligence Algorithm for Daily Scheduling of Thermal Generators", *IEE Proc. C,* 1991, 138,(6), pp. 518-534.

[89] Khanh Doan and Kit Po Wong, "Artificial Intelligence-Based Machine-Learning System for Thermal Generator Scheduling", *IEE Proc. -Gener. Transm. Distrib., Vol. 142, No. 2,* March 1995, pp. 195-201.

[90] G. B. Sheble', and T. T. Maifeld, " Unit Commitment by Genetic Algorithm and Expert System", *Electric Power Systems Research* 30 (1994) 115-121.

[91] S. O. Orero, and M. R. Irving, "Scheduling of Generators With A Hybrid Genetic Algorithm", *Genetic Algorithms in Engineering Systems: Innovation and Applications, IEE Conference*, September 1995, pp. 200-206.

[92] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A New Hybrid Algorithm for Unit Commitment", Accepted for presentation in *the American Power Conference, APC'97, USA.*

[93] A. H. Mantawy, Youssef L. Abdel-Magid, Shokri Z. Selim, "A New Simulated Annealing-Based Tabu Search Algorithm For Unit Commitment", Accepted for presentation in the *IEEE International Conference on Systems Man and Cybernetics, SMC'97, USA.*

[94] E. Aarts and Jan Korst, " Simulated Annealing and Boltzman Machines", A Stochastic Approach to Combinatorial Optimization and Neural Computing, *John Wiley & Sons Ltd.* 1989.

[95] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing", Science 220, 671-680, 1983.

[96] V. Cerny, " Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, *Vol. 45, No. 1*, January 1985.

[97] S. Z. Selim, K. Alsultan, " A Simulated Annealing Algorithm for the Clustering Problem ", *Pattern Recognition*, *Vol. 24, No. 10*, 1991, pp. 1003-1008.

[98] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, " Equations of State Calculations by Fast Computing Machines", *J. of Chem. Phys.* 21, 1087-1092, (1953).

[99] M. Tado, R. Kubo, and N. Saito, "Statistical Physics", *Springer-Verlag, Berlin,* 1983

[100] E. H. L. Aarts, and P. J. M. van Laarhoven, "Statistical Cooling: A General Approach to Combinatorial Optimization Problems", *Philips Journ. of Research 40, 193-226*, 1985.

[101] E. H. L. Aarts, And P. J. M. van Laarhoven, "A New Polynomial Time Cooling Schedule, Proc. *IEEE Int. Conf. on Computer-Aided Design, Santa Clara*, 206-208, 1985.

[102] E. H. L. Aarts, And P. J. M. van Laarhoven, "Simulated Annealing: A Pedestrian Review of The Theory and Some Applications, In: *P. A. Devijver*

And J. Kittler(Eds), Pattern Recognition Theory and Applications, NASI Series on Computer and Systems Sciences 30, Springer-Verlag, Berlin, 179-192, 1987.

[103] F, Glover, and H. J. Greenberg, " New Approach for Heuristic Search: A Bilateral Linkage With Artificial Intelligence", *European Journal of Operational Research* 39 (1989) 119-130.

[104] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence", *Comp. & Ops. Res.* Vol. 13, No. 5, pp. 533-549, 1986.

[105] F. Glover, "Tabu Search-Part I", *Orsa Journal on Computing,* Vol. 1., No. 3, pp. 190-206, Summer 1989.

[106] F. Glover, "Artificial Intelligence, Heuristic Frameworks and Tabu Search", *Managerial and Decision Economics,* Vol. 11, 365-375 (1990).

[107] F. Glover, "Tabu Search-Part II", *Orsa Journal on Computing,* Vol. 2., No. 1, pp. 4-32, Winter 1990.

[108] J. A. Bland and G. P. Dawson, "Tabu Search and Design Optimization", Vol. 23, No. 3, pp. 195-201, April 1991.

[109] F. Glover, " A User's Guide to Tabu Search ", *Annals of Oper. Res.* 41(1993)3-28

[110] Manuel Laguna, and Fred Glover, "Integrating Target Analysis and Tabu Search for Improved Scheduling Systems", *Expert Systems With Applications, Vol. 6,* 1993, pp. 287-297.

[111] James P. Kelly, Bruce L. Olden and Arjang A. Assad, "Large-Scale Controlled Rounding Using Tabu Search With Strategic Oscillation", *Annals of Oper. Res. 41(1993), 69-84.*

[112] J. Wesley Barnes, and Mauel Laguna, "A Tabu Search Experience In Production Scheduling", *Annals of Oper. Res. 41(1993), 141-156.*

[113] Michele Charest and Jacques A. Ferland, "Preventive Maintenance Scheduling of Power Generating Units", *Annals of Oper. Res. 41(1993), 185-206.*

[114] Richard L. Daniels and Joseph B. Mazzola, "A Tabu Search Heuristic for the Flexible-Resource Flow Shop Scheduling Problem", *Annals of Oper. Res. 41(1993), 207-230.*

[115] Mauro Dell Amico and Marco Trubian, "Applying Tabu Search to the Jop-Shop Scheduling Problem", *Annals of Oper. Res. 41(1993), 231-252.*

[116] Edward L. Mooney, and Roland L. Rardin, "Tabu Search for A Class of Scheduling Problems", *Annals of Oper. Res. 41(1993), 253-278.*

[117] D. E. Goldberg, " Genetic Algorithms in Search, Optimization and Machine Learning, Reading, Mass., *Addison Wesely*, 1989.

[118] L. Davis (Ed), " Handbook of Genetic Algorithms", *Van Nostrand*, N. York, 1991.

[119] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", *Springer-Verlag Berlin Heidelberg New York*, 1992.

[120] John J. Grefenstette, " Optimization of Control Parameters for Genetic Algorithms", *IEEE Trans. on Systems, Man, and Cybernetics*, 16, No. 1, 1986, pp. 122-128.

[121] John J. Grefenstette, and James E. Baker," How Genetic Algorithm Work: A Critical Look At Implicit Parallelism", *The Proceedings of The Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California*, 1989.

[122] Bill P. Buckles, Frederick E. Petry, and Rebecca L. Kuester, " Schema Survival Rates and Heuristic Search in Genetic Algorithms", *Proceedings of Tools for AI*, 1990, pp. 322-327.

[123] B. Awadh, N. Sepehri, and O. Hawaleshka, " A Computer-Aided Process Planning Model Based on Genetic Algorithms", *Computer Ops. Res., Vol. 22, No. 8*, pp. 841-856, 1995.

[124] David E. Goldberg, K. Deb and James H. Clark, "Genetic Algorithms, Noise, and the Sizing of Populations", *Complex systems 6 (1992) 333-362.*

[125] Abdollah Homaifar, S. Guan and G. E. Liepins, "Schema Analysis of the Traveling Salesman Problem Using Genetic Algorithms", *Complex Systems 6 (1992) 533-552.*

[126] A. H. Mantawy, Youssef L. Abdel-Magid, Shokri Z. Selim, and Mazen A. Salah, " An Improved Simulated Annealing Algorithm for Unit Commitment-Application to Sceco-East", *3rd International Conference on Intelligent Applications in Communications and Power Systems, IACPS'97, pp. 133-139, April, 1997, UAE.*

CURRICULUM VITA

Name: Abdel-aal Hassan Ismail Mantawy

Date of Birth: 1959

Place of Birth: Cairo, Egypt

Nationality: Egyptian

Education:

M.Sc. in Electrical Engineering **Date: Sept. 1988**
Ain Shams University, Cairo, Egypt.
Thesis Title: "Optimal Scheduling of Thermal Generation in Electric Power Systems".

B.Sc. in Electrical Engineering **Date: June 1982**
Ain Shams University, Cairo, Egypt.

Experience:

Sept. 92 - Present **Teaching Assistant and Ph.D. student**
Department of Electrical Engineering
King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia.

Sept. 88 - Sept. 92 **Assistant Lecturer**
Electrical Power and Machines Department
Ain Shams University, Cairo, Egypt.

June 82 - Sept. 88 **Demonstrator**
Electrical Power and Machines Department
Ain Shams University, Cairo, Egypt.

June 82 - Sept. 92 **Part-time Design Engineer**
Design Engineering Office
Consultant Engineer Samir Al-Guindy, Cairo, Egypt.

July 85 - July 86 **Part-time Analyst Engineer**
Analyst Engineer in " Standardization of 11 kv Circuit Breakers in Cairo Zone".

Research Areas:

- Power systems operation and planning.
- Application of optimization methods to electric power systems.
- Application of artificial intelligence techniques to electric power systems.

Awards:

- Recipient of the Government of Egypt Distinction Award (1978-1982).
- A Scholarship of the Egyptian Government to continue the Ph.D. degree on the channel system (1992).
- A Scholarship from King Fahd University of Petroleum & Minerals, Saudi Arabia, to get the Ph.D. degree (1992).

Publications out of the Dissertation:

(i) Journal Papers

[1] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, " A Simulated Annealing Algorithm for Unit Commitment ", **Accepted for publication in the *IEEE Trans. on Power Systems.***

[2] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, " Unit Commitment by Tabu Search", revised and resubmitted to the *IEE Proceedings - Generation, Transmission and Distribution, December 1996.*

[3] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim, " A New Genetic-Based Tabu Search Algorithm For Unit Commitment Problem", submitted to the *Journal of Electric Power Systems Research, January 1997.*

[4] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim," Integrating Genetic Algorithms Tabu Search and Simulated Annealing For Unit Commitment Problem", submitted to the *Journal of Electrical Power and Energy Systems, January 1997.*

[5] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim," Integrating Simulated Annealing and Tabu Search Algorithms For Unit Commitment Problem", submitted to the *Journal of Electric Machines and Power Systems, February 1997.*

[6] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "An Improved Tabu Search Algorithm for Unit Commitment", submitted to the *IEEE Trans. on Power Systems, 1996.*

(ii) Refereed International Conference Papers

[7] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A New Hybrid Algorithm for Unit Commitment", ***Proceedings of the American Power Conference, APC'97, April 1st, 1997, USA.***

[8] A. H. Mantawy, Youssef L. Abdel-Magid, Shokri Z. Selim, and Mazen A. Salah, " An Improved Simulated Annealing Algorithm for Unit Commitment-Application to Sceco-East", ***Proceedings of the 3rd International Conference on Intelligent Applications in Communications and Power Systems, IACPS'97, pp. 133-139, April, 1997, UAE.***

[9] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim, "A Genetic Algorithm With Local Search For Unit Commitment", ***Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, ISAP'97, July 1997, Korea***

[10] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim, "A New Simulated Annealing-Based Tabu Search Algorithm For Unit Commitment", ***Proceedings of the IEEE International Conference on Systems Man and Cybernetics, SMC'97, October 1997, USA..***

[11] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim, "A New Genetic Algorithm Implementation For Unit Commitment", ***Proceedings of the 4th Annual IEEE Technical Exchange Meeting, KFUPM, Saudi Arabia,*** **May 1997, pp. 182-187.**

[12] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim," A New Genetic Algorithm Approach For Unit Commitment", ***Proceedings of the IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA'97, September 1997, UK.***

[13] A. H. Mantawy, Youssef L. Abdel-Magid and Shokri Z. Selim," Integrating Genetic Algorithms and Tabu Search For Unit Commitment Problem", submitted to the *IEEE Industry Applications Society 32nd Annual Meeting, IAS'97, USA.*