We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Genetic Algorithms Application to Electric Power Systems

Abdel-aal H. Mantawy
Ain Shams University, Faculty of Engineering,
Electrical Power and Machines Department,
Egypt

1. Introduction

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.

With the opening of the power industry to competition, the power system structure is changing. According to these changes, power system operation, planning, and control need modifications. In the past, utilities had to produce power to satisfy their customers with objective to minimize costs and all demand/reserve were met. However, it is not necessary in a restructured system. Under new structure, generation companies schedule their generators with objective to maximize their own profit without regard for system social benefit. Power and reserve prices become important factors in decision process.

The optimal scheduling of thermal generating units in the electrical power system is called thermal UC. 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. To solve the UCP, another crucial problem must simultaneously be solved; the economic dispatch problem.

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 unit's operating limits [1-3].

The UCP is obtained by considering many factors, including:

- Unit operating costs/constraints
- Generation and reserve constraints; and
- Plant start-up and shut down constraints.

The objective of this report is to introduce the general UCP and discuss various considerations involving the chapter solution of this problem. It also aims to summarize the methods that have been used to solve this problem. Moreover an interest will be given to the Genetic Algorithms [4, 5] technique as a powerful tool to solve the UCP as one of the complex optimization problems.

This chapter includes seven sections organized as follows: In section 2, a literature survey for the UCP solution methods is presented. Section 3 introduces the problem formulation. In section 4, genetic algorithm background is presented. Section 5, outlines the genetic algorithm solution to the UCP. In section 6, general conclusions of the chapter report are presented. Section 7 includes the chapter report references.

2. Literature survey

The solution of the UCP is really a complex optimization problem. It comprises the solution of the EDP as well. The UCP can be considered as two linked optimization problems. The first is the UCP, which is considered as a combinatorial problem, and the second is the EDP, which 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 EDP 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 [1-74]:

- Classical optimization methods such as: Dynamic Programming, Integer and Mixed Integer Programming, Lagrangian Relaxation, Linear Programming, Network Flow Programming, Probabilistic Methods and other methods [6-18].
- Heuristic methods such as Priority List and Expert Systems [19-21].
- Artificial Intelligence methods such as: Neural Networks, Simulated Annealing, Tabu Search and Genetic Algorithms [22-57].
- Hybrid Algorithms: hybridization of two or more of the previously mentioned methods [58-74].

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

2.1 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.

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. AI techniques have been applied successfully to solve the UCP [99-163].

- Neural Networks (NN), [22-26],
- Simulated Annealing (SA), [27-28],
- Tabu Search (TS), [29-30], and
- Genetic Algorithm (GA), [31-57].

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, TS and GA have the attractive feature of assured convergence under appropriate assumptions.

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

2.2 Genetic algorithms application to the UCP

Genetic Algorithms (GAs) have become increasingly popular in recent years in science and engineering disciplines [31-34]. The GA, as a powerful tool to achieve global optima, has been successfully used for the solution of this complex optimization problem. Several papers have been published in solving the UCP using the GA [35-57].

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 crossover, 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.

The following is a summary of a literature survey on solving the UCP using the GA [35-57]:

In 1994, D. Dasgupta et al. [35] 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.

Hong-Tzer Yang, Pai-Chuan Yang and Ching-Lien Huang, 1995, [36], proposed an innovative GA (GA) approach to solve the thermal UCP in power generation industry through a constraint satisfaction technique. Due to a large variety of constraints to be satisfied, the solution space of the UCP is highly nonconvex, and therefore the UCP cannot be solved efficiently by the standard GA.

Hong-Tzer Yang, Pai-Chuan Yang and Ching-Lien Huang [37] modified their previous algorithm [36]. Ramp rate constraints on the units being started up or shut down are tackled in the economic dispatch subprogram by limiting the associated maximum available

capacities for generating. The proposed approach is shown to be well amenable to parallel implementation.

In 1995, X. Ma et al. [38] 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.

S. A. Kazarlis et al. [39], 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. With the technique of the varying quality function, the GA finally manages to locate the exact global optimum.

In 1996, P. C. Yang et al. [40] presented a practical approach for using the GA to solve the UCP. The implemented algorithm deals with the constraints in a different manner. 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 [41], 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. In this approach, the selection, mutation, and crossover operators are restricted to a single time interval. An advantage of this approach is that any constraints, which are already satisfied, cannot be violated later in the sequence.

G. B. Sheble' et al. [42], 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 UC 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.

Maifeld, T.T. and Sheble, G.B. [43], presented a new UC scheduling algorithm. The proposed algorithm consists of using a GA with domain specific mutation operators. The proposed algorithm can easily accommodate any constraint that can be true hosted. Robustness of the proposed algorithm is demonstrated by comparison to a Lagrangian relaxation UC algorithm on three different electric utilities.

Arroyo, J.M, Conejo, A. and Jimenez, N. [44], addressed the UCP with a new idea of implementing an interior point GA conducted through heuristics to get a near optimal solution to the problem. The modeling framework provided was less restrictive than the frameworks provided by other approaches such as dynamic programming or Lagrangian relaxation. The CPU time requirement to solve problems of realistic size was moderate.

In 1997, Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. [45], presented a new GA approach to solve the UCP in electric power systems. In the proposed algorithm, coding the UCP solution is based on mixing binary and decimal representations. A fitness function is constructed from the total operating cost of the generating units without penalty terms. Genetic operators are implemented to enhance the search speed and to save memory space. The UCP is divided into two subproblems a combinatorial optimization problem and a nonlinear programming problem. The former is solved using the proposed GA while the latter problem is solved via a quadratic programming routine.

Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. [46], presents an improved algorithm to the one in [143]. To improve the fine-tuning capabilities of the GA and escape from entrapment in local minimum, a special mutation operator based on a local search algorithm is designed. The new operator enhanced the speed of convergence and improved the quality of solution.

Zhao Hongwei, Yi Liangting, Wang Buyun, Cheng Gang and Yang Haiping [47], presented a revised GA for UCP. The model adjusts the parameters of GA automatically with the population evolution and different chromosomes. A new convergence rule is also given to enhance the convergence speed with global optimum reserved.

Hong-Tzer Yang, Pai-Chuan Yang and Ching-Lien Huang [48], proposed a parallel GA approach to solving the thermal UCP by using a constraint handling technique. The proposed topology of dual-direction ring is shown to be well amenable to parallel implementation of the GA for the UCP.

Zhu Mingyu, Cen Wenhui, Wang Mingyou and Zhang Peichao [49], proposed an enhanced GA to solve the UCP. The new features of the proposed algorithm include chromosome mapping, problem specific operators and local search technique. Significant improvements have been achieved with this implantation.

In 1999, Juste, K.A., Kita, H., Tanaka, E. and Hasegawa, J.[50] proposed algorithm to employ the evolutionary programming (EP) technique, in which populations of contending solutions are evolved through random changes, competition, and selection. The practical implementation of this procedure yielded satisfactory results.

F [50] proposed modification to counter the drawbacks of the GA which are their slow processing speed and their solution inconsistency. The method proposes a sequential UC implementation. It decouples a 24-hour UCP into 24 individual scheduling problems. The proposed modified GA algorithm was successfully tested with 6 generators system.

Christiansen, J.C., Dortolina, C.A. and Bermudez, J.P. [52] presented a new implementation of GA to solve the UCP. The proposed algorithm includes the basic GA operators (cross over and mutation) besides five particular operators that proved to be very useful in order to obtain faster and more accurate solutions lowering the possibility of reaching local optimums.

In January 2002, Senjyu, T., Yamashiro, H., Uezato, K. and Funabashi, T. [53] present a new approach for thermal UCP. To handle the UCP constraints, some cording methods have been proposed. However, these methods require computation time. To overcome these problems, a new genetic operator based on unit characteristic classification and intelligent techniques generating initial populations are introduced.

In February 2002, Swarup, K.S. and Yamashiro, S. [54] presented a solution methodology of UCP using GA. Problem specific operators are proposed for the satisfaction of time dependent constraints. Problem formulation, representation and the simulation results for a 10 generator-scheduling problem are presented

Haoyong Chen Xifan Wang [55] presented a new cooperative co-evolutionary algorithm (CCA) for UCP. CCA is an extension of the traditional GA (GA), which appears to have considerable potential for formulating and solving more complex problems by explicitly modeling the coevolution of cooperating species. This method combines the basic ideas of Lagrangian relaxation technique (LR) and GA to form a two-level approach.

In November 2002, Arroyo, J.M. and Conejo, A.J. [56] addressed the UCP of thermal units. This paper proposed a novel repair GA conducted through heuristics to achieve a near optimal solution to this problem. This optimization technique is directly parallelizable.

In August 2003, Mashhadi, H.R. and Shanechi, H.M. Lucas, C. [57] proposed an improved GA to solve the UCP. In order to improve the convergence of the GA, a new local optimizer for the UCP based on Lamarck theory in the evolution, has been proposed. This local optimizer, which tries to improve the fitness of one chromosome in the population, effectively uses the information generated in calculating the fitness.

2.3 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 [58-74]. 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).

The following survey is divided into two parts: hybrid algorithms without the GA and hybrid algorithms with the GA

2.3.1 Hybrid algorithms with the GA

Many AI and classical optimization techniques have been hybridized with the GA to solve the UCP. The following is a summary of these algorithms:

In 1994, Gerald B. Sheble' et al. [62] presented a genetic-based UCP algorithm. The algorithm uses the ES 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 1995, S. O. Orero et al. [65] 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^{TxN} 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.

Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. [68] presented a new algorithm based on integrating GA, TS and SA methods to solve the UCP. The core of the proposed algorithm is based on GAs. TS is used to generate new population members in the reproduction phase of the GA. A SA method is used to accelerate the convergence of the GA by applying the SA for all the population members. A new implementation of the GA is introduced. In the TS part of the proposed algorithm, a simple short-term memory procedure is used to counter the danger of entrapment at a local optimum, and the premature convergence of the GA. A simple cooling schedule has been implemented to apply the SA test in the algorithm.

Chuan-Ping Cheng, Chih-Wen Liu and Chun-Chang Liu [70-72] presented an application of a combined GA and LR methods for the UCP (LRGA). The proposed LRGA incorporates GA into LR method to update the Lagrangian multipliers and improve the performance of LR method in solving combinatorial optimization problems such as the UCP.

Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. [67] presented a new algorithm based on integrating GA and TS to solve a fuzzy UCP model. The core of the proposed algorithm is based on the GA while TS is used to generate new population members in the reproduction phase of the GA. The uncertainties in the load demand and the spinning reserve constraints are formulated in a fuzzy logic (FL) frame. A fitness function is constructed from the total operating cost of the generating units with penalty term related fuzzy spinning reserve and load demand membership functions. In the TS part of the proposed algorithm, a simple short-term memory procedure is used to counter the danger of entrapment at a local optimum, and the premature convergence of the GA.

Mantawy, A.H [71] presented a fuzzy model for the UCP. The model takes the uncertainties in the forecasted load demand and the spinning reserve constraints in a fuzzy frame. The genetic algorithm (GA) approach is then used to solve the proposed fuzzy UCP model. In the implementation for the GA, coding of the UCP solutions is based on mixing binary and decimal representations. A fitness function is constructed from the total operating cost of the generating units plus a penalty term determined due to the fuzzy load and spinning reserve membership functions.

Gwo-Ching Liao and Ta-Peng Tsao. [73], introduced a hybrid GA/fuzzy system and tabu search method (GAFS-TS) for solving short-term thermal generating UCP. This system makes three important improvements to the GA. First, it generates a set of feasible UC schedules and then put the solution to TS. The GAFS has good global optima search capabilities, but poor local optima search capabilities. The TS method has good local optima search capabilities.

3. Unit Commitment Problem formulation

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:

- A problem statement or, system modeling,
- Rules for generating trial solutions, and
- 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.
- Adding or removing generating units does not affect the load demand.
- 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.

The objective function and the constraints of the UCP are described in Sections 3.1 and 3.2.

3.1 The objective function

The objective function of the UCP is represented by the total operating cost of thermal generating units, which contains two major parts; the production costs and the start-up costs.

3.1.1 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:

$$\mathbf{F_{it}(P_{it}) = A_i P^2{}_{it} + B_i P_{it} + C_i} \quad \$/\mathrm{HR} \tag{3.1}$$

3.1.2 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:

$$\mathbf{ST_{it} = So_i[1 - D_i exp(-Toff_i / Tdown_i)] + E_i} \quad \$ \tag{3.2}$$

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

$$\mathbf{F_T = \sum_{t=1}^{T} \sum_{i=1}^{N} (U_{it}F_{it}(P_{it}) + V_{it}ST_{it} + W_{it}SH_{it})} \quad \$ \tag{3.3}$$

3.2 The constraints

The UCP 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.

3.2.1 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:

$$\mathbf{\sum_{i=1}^{N} U_{it}P_{it} = PD_t \quad ; 1 \le t \le T} \tag{3.4}$$

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 economic 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.

The most commonly used approach is computing the reserve as a given prespecified amount, which is a percentage of the forecasted load demand, i.e.

$$\sum_{i=1}^{N} U_{it} Pmax_i \geq (PD_t + R_t) ; \qquad 1 \leq t \leq T \tag{3.5}$$

3.2.2 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.

$$U_{it} Pmin_i \leq P_{it} \leq Pmax_i U_{it} \; ; 1 \leq t \leq T , \; 1 \leq i \leq N \tag{3.6}$$

2. Minimum up/down time

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

$$\begin{aligned} T_{off_i} &\geq T_{down_i} \\ T_{on_i} &\geq T_{up_i} \end{aligned} \; ; \; 1 \leq i \leq N \tag{3.7}$$

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

$$\sum_{1=0}^{T_{up_i} - 1} U_{i,t+1} \geq V_{it} T_{up_i} ; \qquad 1 \leq t \leq T , \; 1 \leq i \leq N \tag{3.8}$$

$$\sum_{1=0}^{T_{down_i} - 1} (1 - U_{i,t+1}) \geq W_{it} T_{down_i} ; \qquad 1 \leq t \leq T , \; 1 \leq i \leq N \tag{3.9}$$

$$V_{it} \geq U_{it} - U_{i,t-1} ; \qquad 2 \leq t \leq T , 1 \leq i \leq N \tag{3.10}$$

$$W_{it} \geq U_{i,t-1} - U_{it} ; \qquad 2 \leq t \leq T , \; 1 \leq i \leq N \tag{3.11}$$

$$V_{i1} = U_{i1} \; ; \qquad 1 \le i \le N \tag{3.12}$$

$$W_{i1} = 1 - U_{i1} \; ; \qquad 1 \le i \le N \tag{3.13}$$

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 cannot be turned ON at the same time due to some technical conditions or manpower 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 lifetime 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.

3.3 The economic dispatch problem

The economic dispatch problem (EDP) 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 modeling and solving the EDP improves the quality of the UCP solution, and therefore, the performance of the overall UCP algorithm.

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:

$$\text{Minimize} \sum_{i=1}^{N} F_{it}(P_{it}) = A_i P^2_{it} + B_i P_{it} + C_i \qquad \$/HR \tag{3.14}$$

Subject to:

$$\sum_{i=1}^{N} P_{it} = PD_t \quad ; \; 1 \le t \le T \tag{3.15}$$

and

$$Pmin_i \le P_{it} \le Pmax_i \; ; \; 1 \le t \le T \, , \; 1 \le i \le N \tag{3.16}$$

Since the EDP is formulated in a quadratic function, the EDP is solved using a quadratic programming routine.

4. Genetic algorithms approach

4.1 Overview

Genetic algorithms (GAs) have been developed by John Holland, his colleagues, and his students at the University of Michigan in the early 1970's [3,5, 122-128]. GAs have become increasingly popular in recent years in science and engineering disciplines. 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.

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.

GAs are search methods based on the mechanics of natural selection and natural genetics. They combine survival of the strongest among string structures with a structured, yet random, information exchange. In every generation, a new set of artificially developed strings is produced using elements of the strongest of the old; an occasional new element is experimented with for enhancement. Although random in nature, genetic algorithms are not random search. They efficiently utilize historical information to predict new search points with expected improved performance. Furthermore, they are not fundamentally restricted by assumptions about the search space (assumptions concerning continuity, existence of derivatives uni-modality, and other matters). The genetic algorithm is an example of an optimization procedure that utilizes random choice as a mechanism to control the search through a coding of a parameter space.

GAs require the parameters of the optimization problem to be coded as a finite length string over some finite alphabet. In many optimization methods we migrate from a single point in the search space to the next using a transition rule to determine the next point. This point-to-point method is dangerous because it may lead to locating local minima (maxima) in multimodal (many peaked) search spaces. On the other hand, GAs work from a population of strings at the same time, climbing several peaks in parallel; hence, reducing the possibility of locating a false peak over methods that go point to point. The mechanics of a genetic algorithm are extremely simple, only involving copying strings and swapping partial strings.

A simple genetic algorithm that produces satisfactory results in many applications problems must have five components:

A genetic representation for potential solution to the problem; "solution coding"

A way to create an initial population of potential solutions; "Initialization"

An evaluation function that plays the role of the environment, rating solutions in terms of their "fitness function",

Genetic operators that alter the composition of children "Genetic operators",

Values for various parameters that the GA uses (e.g. population size, probabilities of applying genetic operators, etc.); "parameters settings and selection"

4.2 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.

In this step, a coding scheme is selected to code the parameter subject to optimization. This will allow access to parameter features that are not apparent using the parameter default code. This access will allow more freedom and resolution for modifying the parameter features to arrive at the optimal solution.

4.3 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 [4]. Linear scaling is the most commonly used and will be discussed in details in Section 5.3. 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:

$$\mathbf{f}' = \mathbf{f} - (\mathbf{f}' - \mathbf{c}.\sigma) \tag{5.1}$$

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

4.4 Genetic algorithms operators

There are usually three operators in a typical GA; reproduction, crossover and mutation

Reproduction is a process in which individual strings are copied according to their pay-off function values, which can be thought of as a measure of profit, utility or goodness that we want to maximize. This means that copying strings according to their fitness values will give strings with higher fitness value a higher probability of generating one or more offspring in the following generation. So the reproduction operator is simply makes one or more copies of any individual that posses a high fitness value in the current generation to the next generation; otherwise, the individual is eliminated from the solution pool.

Crossover is the second operator (also known as the "recombination"). 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.

Mutation is the third 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. Mutation is needed because, although reproduction and crossover effectively search and recombine information contained in the strings, occasionally they may fail and lose some potentially useful genetic material (1 or 0 at particular locations in the case of binary coding). In artificial genetic systems, the mutation operator protects against such an irrecoverable loss. In the simple GA, mutation is the occasional (with small probability) random alteration of the value of a string position. When used randomly with reproduction and cross over, it is a guaranty against early loss of important notions.

4.5 Parameters settings and selection

Based on the results of the fitness evaluation step, certain individuals of the population are selected for further processing. The selection is based on a predetermined threshold of fitness level. 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.

The search for the optimal GA parameters setting is a very complex task. To achieve good performance of the GA, many experiments for the system under study must be performed to get the most suitable parameters for a specific problem. 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.

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.

4.6 Constraints handling (Repair mechanism)

Constraints handling techniques for the GAs can be grouped into a few categories. One way is to generate a solution without considering the constraints but to include them with penalty factors in the fitness function.

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. However, due to applying the crossover and mutation operations the some constraints might be violated. A mechanism to restore the feasibility is applied according to the problem under study.

4.7 Merits and demerits

The GA technique has the following advantages:

- GAs work with a coding of the parameters of the function they optimize, not the parameters themselves.
- GAs search from a group (population) of points, not a single point.
- GAs use penalty (objective function) information, not auxiliary knowledge such as derivatives or other information.
- GAs use probabilistic transition rules, not deterministic rules.

The followings are some of the shortcomings of the GA technique:

- GA's do not work for large problems due to stochastic algorithm
- Its convergence depend on problem specific parameters that are not clearly defined
- It suffers from convergence and computational requirements

4.8 A general Genetic algorithm

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.

The followings are the major steps of a basic general GA for any optimization problem:

Step 1. Select a suitable coding of the parameters in the problem under study.

Step 2. Initialize a population of chromosomes.

Step 3. Evaluate each chromosome in the population.

Step 4. Create new chromosomes by mating current chromosomes; apply mutation and recombination as the parent chromosomes mate.

Step 5. Delete members of the population to make room for the new chromosomes.

Step 6. Evaluate the new chromosomes and insert them into the population.

Step 7. If the termination criterion is satisfied, stop and return the best chromosomes; otherwise, go to Step (3).

5. Unit commitment solution using Genetic algorithms approach

5.1 Overview

Many GA implementations for the UCP are proposed in the literature. They differ from each other in three respects. First, the UCP solution coding which could be binary or real or mix between binary and real representations. Second, the fitness function could include, in addition to total operating cost, penalty terms for constraints violations or without penalty terms. Third, the GA operators are also differing from one implementation to another. Some algorithms used the basic GA operators only, while others used special operators in order to enhance the solution and speed up the convergence.

Generally speaking the GA for the UCP starts by coding the variables, randomly selecting several initial values, calculating the resultant objective function by solving the EDP based on the UCP decision variables, selecting a subset of the initially selected variables based on highest savings, cross mating the coded locations and mutating the resultant code to arrive at a better solution.

The major steps of the GA as applied to the UCP could be summarized as follows (Fig (5.1)):

- Creating an initial population by randomly generating a set of feasible solutions (chromosomes).
- Evaluating each chromosome by solving the economic dispatch problem.
- 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.

In the following sections, the implementations of the different components of the GA for the UCP are presented.

5.2 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, most of the published works used the binary coding [129-140]. 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.
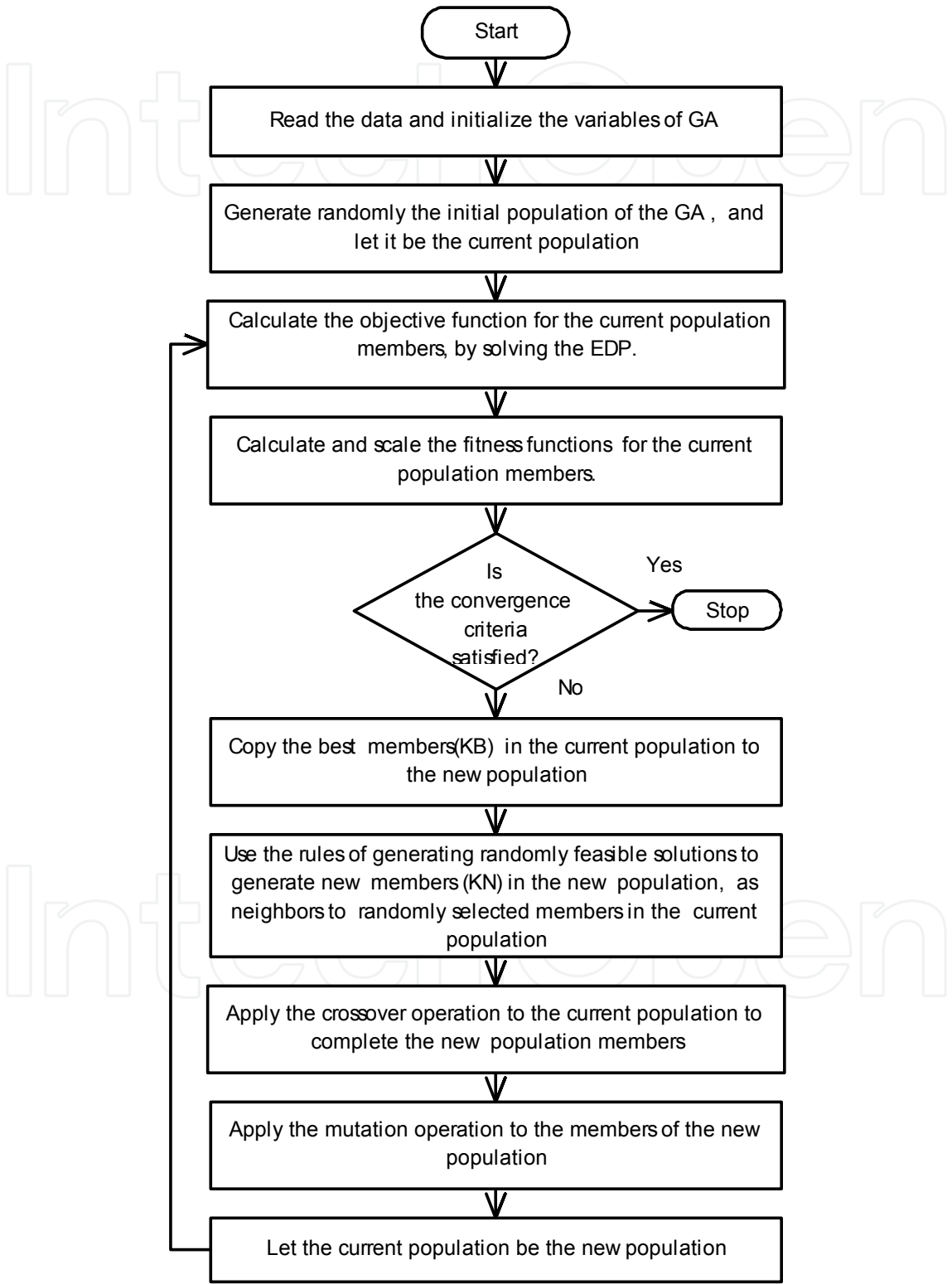
Fig. 5.1. Flow char of the GA for the UCP

Another method of coding presented is based on a mix between a binary number and its equivalent decimal number [141-142]. 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 2 3 . . T	1 1 0 0 . . . 1 1 1 0 0 . . . 1 1 0 1 0 . . . 0 . . . . 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 Fitness function

Unlike the previous solutions of the UCP using GA [129-140], the fitness function is taken as the reciprocal of the total operating cost in (3.1), where all new generated solutions are feasible.

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 [4]:

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

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

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

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.

5.4 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.

5.5 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 4.6) 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

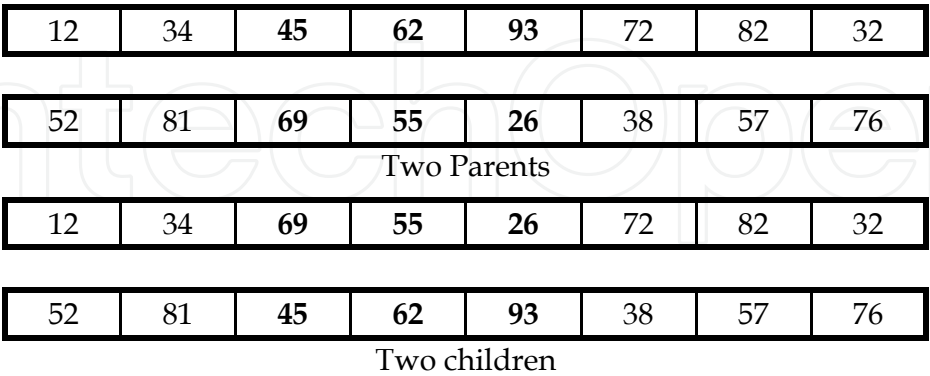| 52	81	**45**	**62**	**93**	38	57	76

Two children

Fig. 5.3. Two points cross over example

5.6 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 [141-142]. 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.

5.6.1 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.

5.6.2 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.

1. The local search algorithm steps are described in details as follows:
2. Decode the selected chromosome into its binary form.
3. Sort the time periods in a descending order according to the difference between the committed units capacity and the load demand.
4. 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.
5. At the time periods of surplus capacity, sort the committed units in an ascending order according to their percentage loading.
6. 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.
7. 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.
8. Check the feasibility of the solution obtained. If it is feasible, go to Step (8), otherwise go to Step (6).
9. Calculate the objective function of the solution obtained by solving the economic dispatch problem for the changed time periods.
10. Decode the new solution obtained to its decimal equivalent and replace the old one in the new population.

5.7 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.

6. 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(6.1) presents the comparison of results obtained in the literature (LR and IP) for Examples 1 and 2.

Fig.(6.1) shows progress in the best objective function versus the generation number. The algorithm converges after about 400 generations, which is relatively fast.

Tables (6.2), (6.3) and (6.4) show detailed results for Example 1 [4.29]. Table (6.2) shows the load sharing among the committed units in the 24 hours. Table (6.3) 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 (6.2), in the form of its equivalent decimal numbers.

Tables (6.5), to (6.8) also present the detailed results for Example 3 with a total operating cost of $661439.8

	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

Table 6.1. Comparison between LR, IP, and the proposed GA
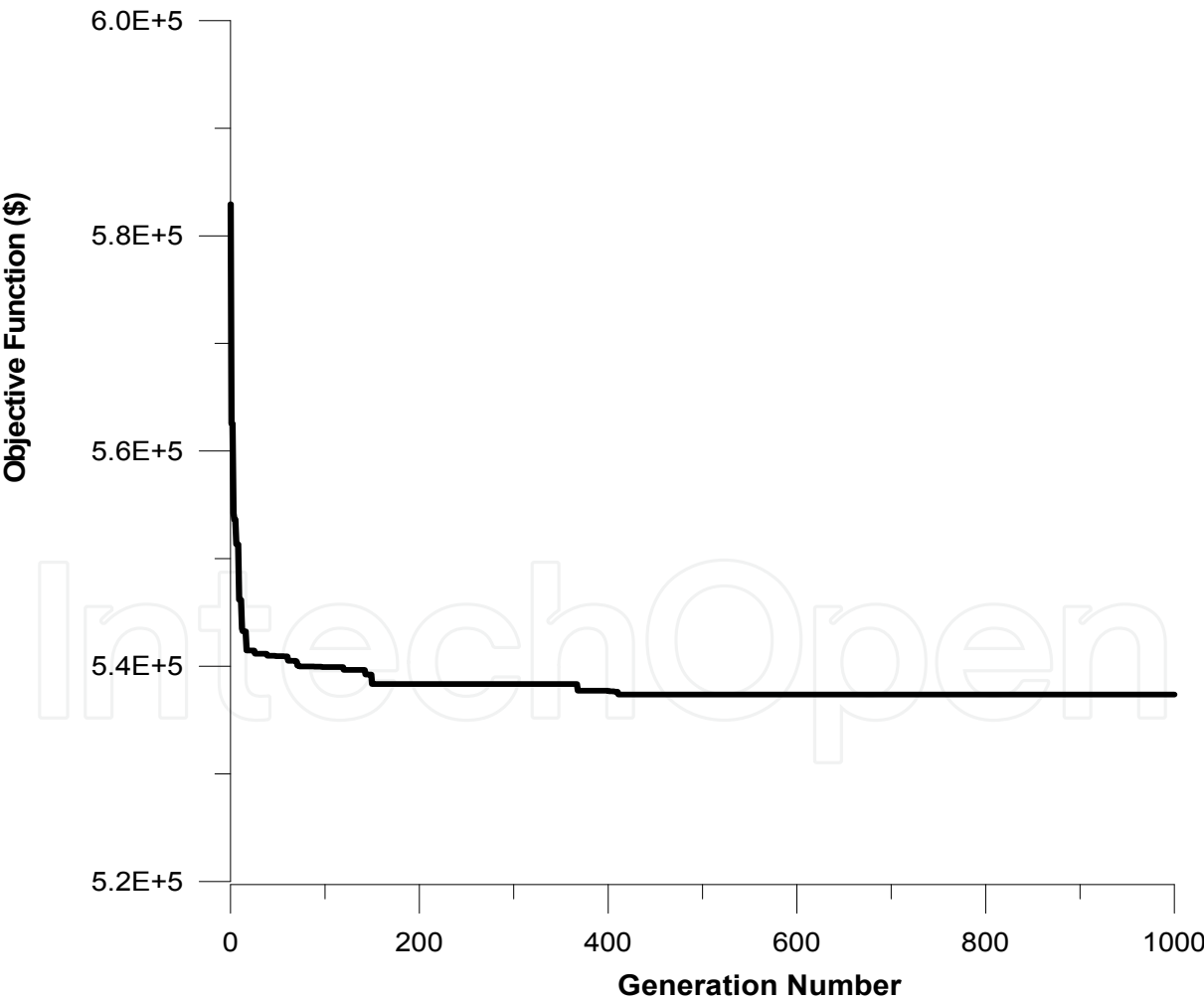
Fig. 6.1. Convergence of the proposed GA

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	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 6.2. Power sharing (MW) 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 6.3. Load demand and hourly costs ($) of Example 1

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

Table 6.4. The UCT of Example 1 in its equivalent decimal form (best chromosome)

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

HR	Unit Number												
	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
2	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
3	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
4	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	343.20	350.00	350.00
5	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
6	0.00	0.00	0.00	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
7	0.00	0.00	69.60	155.00	155.00	155.00	155.00	0.00	0.00	0.00	350.00	350.00	350.00
8	0.00	100.00	100.00	155.00	155.00	155.00	155.65	150.65	68.95	0.00	350.00	350.00	350.0
9	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	122.60	0.00	350.00	350.00	350.00
10	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	122.60	0.00	350.00	350.00	350.00
11	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	142.60	0.00	350.00	350.00	350.00
12	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	102.60	0.00	350.00	350.00	350.00
13	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	112.60	0.00	350.00	350.00	350.00
14	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	92.60	0.00	350.00	350.00	350.00
15	100.00	100.00	100.00	155.00	155.00	155.00	155.00	153.05	68.95	0.00	350.00	350.00	350.00
16	100.00	100.00	34.10	155.00	155.00	155.00	155.00	68.95	68.95	0.00	350.00	350.00	350.00
17	100.00	100.00	78.10	155.00	155.00	155.00	155.00	68.95	68.95	0.00	350.00	350.00	350.00
18	100.00	100.00	100.00	155.00	155.00	155.00	155.00	137.05	68.95	0.00	350.00	350.00	350.00
19	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	109.00	0.00	350.00	350.00	350.00
20	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	149.00	0.00	350.00	350.00	350.00
21	100.00	100.00	100.00	155.00	155.00	155.00	155.00	197.00	129.00	0.00	350.00	350.00	350.00
22	100.00	100.00	100.00	155.00	155.00	155.00	155.00	137.05	68.95	0.00	350.00	350.00	350.00
23	0.00	25.00	25.00	155.00	155.00	155.00	155.00	68.95	68.95	0.00	350.00	350.00	350.00
24	0.00	25.00	25.00	155.00	155.00	155.00	155.00	68.95	68.95	0.00	331.30	350.00	350.00

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

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

Unit Number						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
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		

Table 6.8. The UCT of Example 3 in its equivalent decimal form(best chromosome)

7. Summary

GA is one of the most powerful techniques for solving optimization problems and the UCP as well. The major components of the GA as applied to the UCP, which are different from one implementation to another, are: First, the UCP solution coding which could be binary, real or mix between binary and real representations. Second, the fitness function, which is basically, the total operating costs with or without adding penalty terms for constraints violations. Third, the GA operators: reproduction, crossover and mutation in addition to special operators that might be used to improve the solution speed and quality.

Discussions of the available GA implementations, as applied to the UCP, are summarized.

A complete GA implementation as applied to the UCP is presented. The features of the presented algorithm are: 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.

8. List of abbreviations

UC   Unit Commitment
UCP   Unit Commitment Problem
EDP   Economic Dispatch Problem
Gas: Genetic Algorithms
GA: Genetic Algorithm
NN   Neural Networks
SA: Simulated Annealing
TS: Tabu Search
DP   Dynamic Programming
LR   Lagrangian Relaxation
IP   Integer Programming
MIP   Mixed Integer Programming
PL   Priority List
ES   Expert Systems
HR   Hour
MW   Mega Watt

9. Nomenclature

The following notations are used throughout the thesis:

$\mathbf{A_i, B_i, C_i}$ Cost function parameters of unit i ($\$/ MW^2 .HR, \$/MW.HR, \$/HR$)

$\mathbf{D_i, E_i}$ Start-up cost coefficients for unit ($).

$\mathbf{F_{it}(P_{it})}$ Production cost of unit i at time t ($\$/HR$).

$\mathbf{F_T}$ Total operating cost over the scheduling horizon ($)

$\mathbf{F_i^k}$ The total operating cost for a current solution i at iteration k

$\mathbf{N}$ 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_{upi} Unit i minimum up time.

T_{downi} Unit i minimum down time.

T_{oni} Duration during which unit i is continuously ON.

T_{offi} Duration during which unit i is continuously OFF.

T_{shuti} Instant of shut down of a unit i.

T_{starti} Instant of start-up of a unit i.

$U(0,1)$ The uniform distribution with parameters 0, and 1

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

10. 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] J. Nagrath and D. P. Koathari, "Modern Power System Analysis", TATA McGraw Hill Publishing Co., 1987.
[4] D. E. Goldberg, " Genetic Algorithms in Search, Optimization and Machine Learning, Reading, Mass., Addison Wesely, 1989.
[5] L. Davis (Ed), " Handbook of Genetic Algorithms", Van Nostrand, N. York, 1991.
[6] P. G. Lowery, " Generating Unit Commitment by Dynamic Programming", IEEE Trans. on PAS-85, No. 5, Page(s): 422-426, May 1966.

[7] W. L. Snyder, H. D. Powell, and J. C. Rayburn, "Dynamic Programming Approach to Unit Commitment", IEEE Trans. on Power Systems, Volume PWRS-2, No. 2, May 1987, Page(s): 339-350.

[8] Mantawy A. H., " Optimal Scheduling of Thermal Generation in Electric Power Systems", A Master Thesis, Ain Shams University, Cairo, Egypt, 1988.

[9] G. B. Sheble', and G. N. Fahd, "Unit Commitment Literature Synopsis", IEEE Trans. on Power Systems, Volume 9, No. 1, February 1994, Page(s): 128-135.

[10] Merlin, and P. Sandrin," A New Method for Unit Commitment At Electricite De France", IEEE Trans. on PAS, Volume PAS-102, No. 5, May 1983, Page(s): 1218-1225.

[11] J. F. Bard, " Short-Term Scheduling of Thermal-Electric Generators Using Lagrangian Relaxation", Operation Research, Volume 36, No. 5, September-October 1988, Page(s): 756-766.

[12] 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 Volume 10, No. 3, August 1995, Page(s): 1294-1301.

[13] R. Zhu, C. Fu, and S. Rahamn, "Network Programming Technique for Unit Commitment", Electrical Power & Energy Systems, Volume 17, No. 2, Page(s): 123-127, 1995.

[14] S. Takriti, J. R. Birge, and E. Long, "A Stochastic Model for Unit Commitment Problem", A New Decomposition Framework", IEEE Trans. on Power Systems, Volume 11, No. 3, August 1996, Page(s): 1497-1508.

[15] Ma, H.; Shahidehpour, S.M.; "Unit commitment with transmission security and voltage constraints", IEEE Transactions on Power Systems, Volume: 14, Issue: 2, May 1999, Page(s): 757 – 764.

[16] Motto, A.L.; Galiana, F.D.; "Equilibrium of auction markets with unit commitment: the need for augmented pricing", IEEE Transactions on Power Systems, Volume: 17, Issue: 3, Aug. 2002, Page(s): 798-805.

[17] Momoh, J.A.; Jizhong Zhu; "Optimal generation scheduling based on AHP/ANP", IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume: 33, Issue: 3, June 2003, Page(s): 531-535.

[18] Jing Wang; Redondo, N.E.; Galiana, F.D.;" Demand-side reserve offers in joint energy/reserve electricity markets", IEEE Transactions on Power Systems, Volume: 18, Issue: 4, Nov. 2003, Page(s): 1300 – 1306.

[19] F. N. Lee, "Short-Term Thermal Unit Commitment -- A New Method", IEEE Trans. on Power Systems, Volume 3, No. 2, May 1988, Page(s): 421-428

[20] S. Mokhtari, J. Singh, and B. Wollenberg, " A Unit Commitment Expert System", IEEE Trans. on Power Systems, Volume 3, No. 1, February 1988, Page(s): 272-277.

[21] Z. Ouyang, and S. M. Shahidepour, " Short-Term Unit Commitment Expert System ", Electric Power System Research, 20 (1990) 1-13.

[22] Ronne-Hansen, P.; Ronne-Hansen, J. "Neural networks as a tool for unit commitment", Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, 1991, Page(s): 266 –270.

[23] Ouyang, Z.; Shahidehpour, S.M. "A hybrid artificial neural network-dynamic programming approach to unit commitment", IEEE Transactions on Power Systems, Volume: 7, No.1, Feb. 1992, Page(s): 236 –242.

[24] Sasaki, H.; Watanabe, M.; Kubokawa, J.; Yorino, N.; Yokoyama, R. "A solution method of unit commitment by artificial neural networks", IEEE Transactions on Power Systems, Volume: 7, No. 3, Aug. 1992, Page(s): 974–981.

[25] Walsh, M.P.; O'Malley, M.J., "Augmented Hopfield network for unit commitment and economic dispatch", IEEE Transactions on Power Systems, Volume: 12 4, Nov. 1997, Page(s): 1765 –1774.

[26] M. E El-Hawary, "applications of artificial neural networks in electric power systems operational plannibg", Engineering Intelligent Systems, Vol. 7, No. 1, March 1999, Page(s): 49-61.

[27] Zhuang, F.; Galiana, F.D. "Unit commitment by simulated annealing", IEEE Transactions on Power Systems, Volume: 5, No. 1, Feb. 1990, Page(s): 311-318.

[28] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z., "A simulated annealing algorithm for unit commitment", IEEE Transactions on Power Systems, Volume: 13, No.1, Feb. 1998, Page(s): 197 –204.

[29] Mori, H.; Usami, T. "Unit commitment using Tabu search with restricted neighborhood", International Conference on Intelligent Systems Applications to Power Systems, 1996. Proceedings, ISAP'96, Page(s): 422 –427.

[30] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z. "Unit commitment by tabu search ", Generation, Transmission and Distribution, IEE Proceedings- Volume: 145, No. 1, Page(s). 1998, Page(s): 56–64.

[31] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag Berlin Heidelberg New York, 1992.

[32] 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.

[33] David E. Goldberg, K. Deb and James H. Clark, "Genetic Algorithms, Noise, and the Sizing of Populations", Complex systems 6 (1992) 333-362.

[34] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic algorithms and their applications", IEEE Signal processing magazines, Issue: November 1996, Page(s): 22–37.

[35] Dasgupta, D.; McGregor, D.R.; "Short term unit-commitment using genetic algorithms", Proceedings of the Fifth International Conference on Tools with Artificial Intelligence, TAI '93, 8-11 Nov. 1993, Page(s): 240–247

[36] Hong-Tzer Yang; Pai-Chuan Yang; Ching-Lien Huang; "Applications of the genetic algorithm to the unit commitment problem in power generation industry", Proceedings of the IEEE International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium, Volume: 1, 20-24 March 1995, Page(s): 267–274.

[37] Hong-Tzer Yang; Pai-Chuan Yang; Ching-Lien Huang; "Optimization of unit commitment using parallel structures of genetic algorithm", Proceedings of the International Conference on Energy Management and Power Delivery, EMPD '95, Volume: 2, 21-23 Nov. 1995, Page(s): 577-582.

[38] 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.

[39] S. A. Kazarilis, A. G. Bakirtzis, and V. Petridis, " A Genetic Algorithm Solution to the Unit Commitment Problem", IEEE Trans. on Power Systems, Volume 11, No. 1, February 1996, Page(s): 83-92.

[40] 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.

[41] S. O. Orero and M. R. Irving, "A Genetic Algorithm for Generators Scheduling in Power Systems", Electric Power & Energy Systems, Volume 18, No. 1, 1996, Page(s): 19-26.

[42] 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, Volume 18, No. 6, 1996, Page(s): 339-346.

[43] Maifeld, T.T.; Sheble, G.B.; "Genetic-based unit commitment algorithm" IEEE Transactions on Power Systems, Volume: 11, Issue: 3, Aug. 1996, Page(s): 1359-1370.

[44] Arroyo, J.M; Conejo, A.; Jimenez, N., "A genetic algorithm approach to solve the unit commitment problem", Intelligent System Application to Power Systems (ISAP'97), July 6-10, 1996, Seoul Korea, Page(s): 182-187.

[45] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z. "A new genetic algorithm approach for unit commitment", Second International Conference On Genetic Algorithms In Engineering Systems: Innovations and Applications, GALESIA 97, (Conf. Publ. No. 446), 1997, Page(s): 215 -220.

[46] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z. "A genetic algorithm with local search for unit commitment", Intelligent System Application to Power Systems (ISAP'97), July 6-10, 1996, Seoul Korea, Page(s): 170-175.

[47] Zhao Hongwei; Yi Liangting; Wang Buyun; Cheng Gang; Yang Haiping "A new genetic algorithm for unit commitment", IEEE International Conference on Intelligent Processing Systems, 1997, ICIPS '97, Volume: 1, Page(s): 606–610.

[48] Hong-Tzer Yang; Pai-Chuan Yang; Ching-Lien Huang, "A parallel genetic algorithm approach to solving the unit commitment problem: implementation on the transputer networks", IEEE Transactions on Power Systems, Volume: 12, No. 2, May 1997, Page(s): 661 -668

[49] Zhu Mingyu; Cen Wenhui; Wang Mingyou; Zhang Peichao; "Using an enhanced genetic algorithm to solve the unit commitment problem", IEEE International Conference on Intelligent Processing Systems, ICIPS '97, Volume: 1, 28-31 Oct. 1997, Page(s): 611 – 614.

[50] Juste, K.A.; Kita, H.; Tanaka, E.; Hasegawa, J.; "An evolutionary programming solution to the unit commitment problem", IEEE Transactions on Power Systems, Volume: 14, Issue: 4, Nov. 1999, Page(s): 1452 – 1459.

[51] Richter, C.W., Jr.; Sheble, G.B.; "A profit-based unit commitment GA for the competitive environment", IEEE Transactions on Power Systems, Volume: 15, Issue: 2, May 2000, Page(s): 715-721.

[52] Christiansen, J.C.; Dortolina, C.A.; Bermudez, J.P.; "An approach to solve the unit commitment problem using genetic algorithm", IEEE Power Engineering Society Summer Meeting, 2000, Volume: 1, 16-20 July 2000, Page(s): 261-266.

[53] Senjyu, T.; Yamashiro, H.; Uezato, K.; Funabashi, T.; "A unit commitment problem by using genetic algorithm based on unit characteristic classification", IEEE Power Engineering Society Winter Meeting, Volume: 1, 27-31 Jan. 2002, Page(s): 58–63.

[54] Swarup, K.S.; Yamashiro, S.; "Unit commitment solution methodology using genetic algorithm", IEEE Transactions on Power Systems, Volume: 17, Issue: 1, Feb. 2002, Page(s): 87-91.

[55] Haoyong Chen; Xifan Wang; "Cooperative co evolutionary algorithm for unit commitment", IEEE Transactions on Power Systems, Volume: 17, Issue: 1, Feb. 2002, Page(s): 128– 33.

[56] Arroyo, J.M.; Conejo, A.J.; "A parallel repair genetic algorithm to solve the unit commitment problem", IEEE Transactions on Power Systems, Volume: 17, Issue: 4, Nov. 2002, Page(s): 1216-1224.

[57] Mashhadi, H.R.; Shanechi, H.M.; Lucas, C.; "A new genetic algorithm with Lamarckian individual learning for generation scheduling", IEEE Transactions on Power Systems, Volume: 18, Issue: 3, Aug. 2003, Page(s): 1181-1186.

[58] Hans P. Van Meeteren, "Scheduling of Generation and Allocation of Fuel, Using Dynamic and Linear Programming", IEEE Trans. on Power Apparatus and Systems, Volume PAS-103, No. 7, July 1984, Page(s): 1562-1568.

[59] K. P. Wong and K. P. Doan, "Artificial Intelligence Algorithm for Daily Scheduling of Thermal Generators", IEE Proc. C, 1991, 138,(6), Page(s): 518-534.

[60] Z. Ouyang, and S. M. Shahidehpour, "An Intelligent Dynamic Programming for Unit Commitment Application", IEEE Trans. on Power Systems, Volume 6, No. 3, August 1991, Page(s): 1203-1209.

[61] Z. Ouyang, and S. M. Shahidehpour, "A Hybrid Artificial Neural Network-Dynamic Programming Approach to Unit Commitment", IEEE Trans. on Power Systems, Volume 7, No. 1, February 1992, Page(s): 236-242.

[62] G. B. Sheble', and T. T. Maifeld, " Unit Commitment by Genetic Algorithm and Expert System", Electric Power Systems Research 30 (1994) 115-121.

[63] 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.

[64] Khanh Doan and Kit Po Wong, "Artificial Intelligence-Based Machine-Learning System for Thermal Generator Scheduling", IEE Proc. -Gener. Transm. Distrib., Volume 142, No. 2, March 1995, Page(s): 195-201.

[65] 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, Page(s): 200-206.

[66] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z. "A new simulated annealing-based tabu search algorithm for unit commitment", IEEE International Conference on Computational Cybernetics and Simulation., Systems, Man, and Cybernetics, 1997., Volume: 3, Page(s): 2432–2437.

[67] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z. "A new genetic-based tabu search algorithm for unit commitment problem", Journal of Electric Power Systems Research 49 (1999), Page(s): 71-78.

[68] Mantawy, A.H.; Abdel-Magid, Y.L.; Selim, S.Z.; "Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem", IEEE Transactions on Power Systems, Volume: 14, Issue: 3, Aug. 1999, Page(s): 829-836.

[69] Mantawy, A.H.; Abdel-Magid, Y.L.; Abido, M.A., " A simulated annealing algorithm for fuzzy unit commitment problem", IEEE Transmission and Distribution Conference, 1999, Volume: 1, Page(s): 142 -147.

[70] Chuan-Ping Cheng; Chih-Wen Liu; Chun-Chang Liu; "Unit commitment by Lagrangian relaxation and genetic algorithms", IEEE Transactions on Power Systems, Volume: 15, Issue: 2, May 2000, Page(s): 707-714

[71] Mantawy, A.H.; "A genetic-based algorithm for fuzzy unit commitment model", IEEE Power Engineering Society Summer Meeting, Volume: 1, 16-20 July 2000, Page(s): 250–254.

[72] Jimenez-Redondo, N.; Chuan-Ping Cheng; Chih-Wen Liu; Chun-Cheng Liu; "Unit commitment by Lagrangian relaxation and genetic algorithms [discussion and closure]", IEEE Transactions on Power Systems, Volume: 16, Issue: 4, Nov. 2001, Page(s): 938– 39.

[73] Gwo-Ching Liao; Ta-Peng Tsao; "The use of genetic algorithm/fuzzy system and tabu search for short-term unit commitment", Proceedings OF the International Conference on Power System Technology, PowerCon 2002, Volume: 4, 13-17 Oct. 2002, Page(s): 2302 – 2307.

[74] Rajan, C.C.A.; Mohan, M.R.; Manivannan, K.; "Neural-based tabu search method for solving unit commitment problem", Generation, Transmission and Distribution, IEE Proceedings-, Volume: 150, Issue: 4, 14 July 2003 Page(s): 469-474.
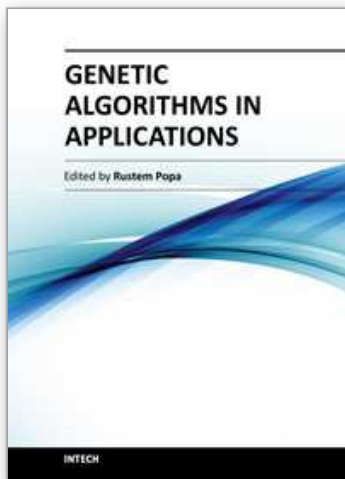
Genetic Algorithms in Applications

Edited by Dr. Rustem Popa

Genetic Algorithms (GAs) are one of several techniques in the family of Evolutionary Algorithms - algorithms that search for solutions to optimization problems by "evolving" better and better solutions. Genetic Algorithms have been applied in science, engineering, business and social sciences. This book consists of 16 chapters organized into five sections. The first section deals with some applications in automatic control, the second section contains several applications in scheduling of resources, and the third section introduces some applications in electrical and electronics engineering. The next section illustrates some examples of character recognition and multi-criteria classification, and the last one deals with trading systems. These evolutionary techniques may be useful to engineers and scientists in various fields of specialization, who need some optimization techniques in their work and who may be using Genetic Algorithms in their applications for the first time. These applications may be useful to many other people who are getting familiar with the subject of Genetic Algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds