Journal of ELECTRICAL ENGINEERING, VOL. 60, NO. 2, 2009, 69–78

GENETIC ALGORITHM BASED TABU SEARCH METHOD FOR SOLVING UNIT COMMITMENT PROBLEM WITH COOLING — BANKING CONSTRAINTS

Christober C. Asir Rajan *

This paper presents a new approach to solving the short-term unit commitment problem using the genetic algorithm based tabu search method with cooling and banking constraints. The objective of this paper is to find the generation scheduling such that the total operating cost can be minimized, when subjected to a variety of constraints. This also means that it is desirable to find the optimal generating unit commitment in the power system for the next H hours. A power station with seven generating units in India demonstrates the effectiveness of the proposed approach; extensive studies have also been performed for different power systems consisting of IEEE 10, 26, 34 generating units. Numerical results are shown comparing the cost solutions and computation time obtained by the genetic algorithm, tabu search method and other conventional methods.

K e y w o r d s: unit commitment, genetic algorithm, tabu search, cooling-banking

1 INTRODUCTION

Power systems have the problem of deciding how best to meet the varying demand for electricity, which has a daily and weekly cycle. The short-term optimization problem is how to schedule generation to minimize the total fuel cost or to maximize the total profit over a study period of typically a day, subject to a large number of constraints that must be satisfied. The daily load pattern for a given system may exhibit large differences between minimum and maximum demands. Therefore enough reliable power generation to meet the peak load demand must therefore be synchronized prior to the actual occurrence of the load. Thus it is clear that it is not proper and economical to run all the units available all the time. Since the load varies continuously with time, the optimum condition of units may alter during any period. Therefore the problem of determining the units of a plant that should operate for a given load is the problem of unit commitment. For total number of units of higher order, the problems associated with unit commitment have generally been difficult to solve because of uncertainty of particular aspects of the problem. For instance the availability of fuel in precise, load forecast variable costs affected by the loading of generator units and the losses caused by reactive flows are some of the unpredictable issues. In order to reach a feasible solution for Unit Commitment Problem (UCP), different considerations must be considered.

Research endeavors, therefore, have been focused on; efficient, near-optimal UC algorithms, which can be applied to large-scale, power systems and have reasonable storage and computation time requirements. A survey of existing literature [1-33] on the problem reveals that various numerical optimization techniques have been employed to approach the complicated unit commitment problem. More specifically, these are the Dynamic Programming method (DP), the Mixed Integer Programming method (MIP), the Lagrangian relaxation method (LR), the Enhanced Lagrangian relaxation method (ELR), the Branch and Bound method (BB), the Expert system (ES), the Fuzzy Theorem method (FT), the Hop Field method (H), the Artificial Neural Network (ANN), the Artificial Neural Network with Short Term Load Forecasting (ANN STLF), the Simulated Annealing method (SA), the Tabu Search (TS), the Genetic Algorithm (GA), the integration of genetic algorithm, tabu search, simulated annealing (GTS), the TS and decomposition method (TSD), the extended neighbourhood search algorithm (ENSA) and so on. The major limitations of the numerical techniques are the problem dimensions, large computational time and complexity in programming.

The DP method [1, 2] is flexible but the disadvantage is the "curse of dimensionality", whose results may lead to more mathematical complexity and increase in computation time if the constraints are taken into consideration. The MIP methods [3, 4] for solving the unit commitment problems fail, when the number of units increases because they require a large memory and suffer from a great computational delay and the MIP based approach requires less computation time and lesser the memory requirement for large system [5]. The LR approach [6–9] to solve the short-term UC Problems was found to provide faster solution but it will fail to obtain solution feasibility and solution quality problems and becomes complex if the number of units increased. The ELR method was less expensive than the conventional methods and requires less computation time [10]. The BB method [11]

* Department of EEE Pondicherry Engineering College Puducherry – 605 014 India, asir_70@hotmail.com

ISSN 1335-3632

employs a linear function to represent the fuel cost and start-up cost and obtains a lower and upper bounds. The difficulty of this method is the exponential growth in the execution time for systems of a practical size. An ES algorithm [12] rectifies the complexity in calculations and saving in computation time. But it will face the problem if the new schedule is differing from the schedule in the database. In the FT method [13] using fuzzy set solves the forecasted load schedules error but it will also suffer from complexity. The H neural network technique [14] considers more constraints but it may suffer from numerical convergence due to its training process. The ANN has the advantages of giving good solution quality and rapid convergence. And this method can accommodate more complicated unit-wise constraints and are claimed for numerical convergence and solution quality problems. The solution processing in each method is very unique. The ANN STLF method [15] improved the level of accuracy of forecasting performance and the quality was improved of UC scheduling and results in a large amount of cost savings per year. SA [16–19] is a powerful, general-purpose stochastic optimization technique, which can theoretically converge asymptotically to a global optimum solution with probability one. But it will take much time to reach the near-global minimum. TS [20, 21] is an iterative improvement procedure that starts from some initial feasible solution and attempts to determine a better solution in the manner of a greatest-decent algorithm. However, TS is characterized by an ability to escape local optima by using a short-term memory of recent solutions. GA [22, 23] is a general-purpose stochastic and parallel search method based on the mechanics of natural selection and natural genetics. It is a search method to have potential of obtaining near-global minimum. And it has the capability to obtain the accurate results within short time and the constraints are included easily. The GA has the advantages of good convergent property and a significant speedup over traditional methods and can obtain high quality solutions. The "curse of dimensionality" is surmounted, and the computational burden is almost linear with the problem scale. The execution time was reduced compared to binary coded GA implementations and did not require heuristic changes to the population size or the number of generations in order to converge, regardless of the system size [24]. The clusters with the intelligent mutation and grey-zone modification methods and found that the effective relaxed pruned ELD calculations with GA operators to remove the drawbacks of GA [25].

GTS [26] shows a reasonable combination of local and global search. It adopts the acceptance probability of SA to improve the convergence of the simple GA, and the tabu search is introduced to find more accurate solutions. The TSD [27] has considered the time varying start-up costs as well as the non-linearity in the hydrothermal systems. It can be used as a post processor for existing generation scheduling methods or in cases where rescheduling of units is required due to change in the system status. The application of the modified Benders decomposition method is to solve with constraints that are difficult to formulate. In order to obtain the better results, the experience of the operators in applying some system specific conditions has been included in the tabu search method. The proposed approach by this paper can be used in conjunction with the other optimization method to pursue a more comprehensive feasible solution if the initial solutions obtained by other optimization methods fail to satisfy some specific constraints.

In ENSA [28] the constrained models for fuel limits, emission limits and generation capacity limits are discussed and used for typical models. The method can make use of an algorithm that satisfies the objective of the sub problem. Most suitably, and starts from an initial solution even though the solution may be feasible. The higher integral economic effect is pursued, and the feasibility of the algorithm is maintained. The proposed method may be used for rescheduling purposes where the experience of human experts will be combined with the analytical method of optimal scheduling. The algorithm can also be used in other complicated mixed integer programming problems, such as integrated resource planning.

From the literature review, it has been observed that there exists a need for evolving simple and effective methods, for obtaining an optimal solution for the UCP. Hence, in this paper, an attempt has been made to couple GA with TS for meeting these requirements of the UCP, which eliminates the above-mentioned drawbacks. TS is a general-purpose stochastic search technique to solve hard-constrained optimization problems. Though it takes long time, it has many strong features such as, it is easy to implement, requires little expert knowledge and is not memory intensive. In case of TS, the demand is taken as control parameter. Hence the quality of solution is improved. Different criteria for constructing the tabu list restrictions for the UCP are implemented. Several examples are solved to test the developed computer model. Hence, the quality of solution is improved. GA is robust and relatively domain-independent global optimization methods, although convergence to a global optimization solution is guaranteed only in a weak probabilistic sense. GA's are well suited for combinational optimization problems, and have been successfully applied to a variety of problems. GA is capable of determining the global or near global solution. It is based on the basic genetic operation of human chromosomes. It operates with the stochastic mechanics, which combine offspring creation based on the performance of current trail solutions and selection based on the successive generations, form a considerably robust scheme for large-scale real-valued combinational optimization. In this proposed work, the parents are obtained from a pre-defined set of solution's *ie* each and every solution is adjusted to meet the requirements.

The GA combines good solution quality for TS with rapid convergence for GA. The TS embedded GA (GATS) is used to implement and locate optimal or near-optimal solutions to typical optimization problems such as UCP. By doing so, it can help to find the optimum solution

rapidly and efficiently. However they take long time to finally reach the global solution. Many advanced operators are proposed to reduce the search time to acceptable values. But these sophisticated operators are problem specific and sensitive to the problem parameters. Hence, local search and hybrid combinations of different methods have been proposed to obtain a robust optimization method. A power station with seven generating units in India demonstrates the effectiveness of the proposed approach; extensive studies have also been performed for different power systems consist of IEEE 10, 26, 34 generating units.

2 PROBLEM FORMULATION

The objective is to find the generation scheduling such that the total operating cost can be minimized, when subjected to a variety of constraints [10]. In the UCP under consideration, an interesting solution would be minimizing the total operating cost of the generating units with several constraints being satisfied. The major component of the operating cost, for thermal and nuclear units, is the power production cost of the committed units and this is given in a quadratic form

$$F_{it}(P_{it}) = A_i P_{it}^2 + B_i P_{it} + C_i \text{Rs/h}\,, \tag{1}$$

where

A_i, B_i, C_i – the cost function parameters of unit i (Rs./MW^{2}h, Rs./MWh, Rs/h)
$F_{it}(P_{it})$ – production cost of unit i at a time t (Rs/h)
P_{it} – output power from unit i at time t (MW).

The start up cost depends upon the down time of the unit, which can vary from a maximum value, when the unit i is started from cold state, to a much smaller value, if the unit i has been turned off recently. The start up cost calculation depends upon the treatment method for the thermal unit during down time periods. The start-up cost S_{it}, is a function of the down time of unit i as given in by

$$S_{it} = So_i[1 - D_i \exp(-Toff_i/Tdown_i)] + E_i \quad \text{Rs}\,, \tag{2}$$

where

So_i – unit i cold start-up cost (Rs)
D_i, E_i – start-up cost coefficients for unit i.

The overall objective function of the UCP is

$$F_T = \sum_{t=1}^{T}\sum_{i=1}^{N}\big(F_{it}(P_{it})U_{it} + S_{it}V_{it}\big) \quad \text{Rs/h}\,, \tag{3}$$

where

U_{it} – unit i status at hour $t = 1$ (if unit is ON) $= 0$ (if unit is OFF)
V_{it} – unit i start up/shut down status at hour $t = 1$ if the unit is started at hour t and 0 otherwise
F_T – total operating cost over the schedule horizon (Rs/h)
S_{it} – start up cost of unit i at hour t (Rs).

2.1 Constraints

Depending on the nature of the power system under study, the UCP is subject to many constraints, the main being the load balance constraints and the spinning reserve constraints. The other constraints include the thermal constraints, fuel constraints, security constraints *etc* [10].

1) Load Balance Constraints

The real power generated must be sufficient enough to meet the load demand and must satisfy the following factors given in as

$$\sum_{i=1}^{N} P_{it}U_{it} = PD_t\,, \tag{4}$$

where

PD_t – system peak demand at hour t (MW)
N – number of available generating units
$U(0,1)$ – uniform distribution with parameters 0 and 1
$UD(a,b)$ – discrete uniform distribution with parameters a and b.

2) Spinning Reserve Constraints

The spinning reserve is the total amount of real power generation available from all synchronized units minus the present load plus the losses. The reserve is considered to be a pre specified amount or a given percentage of the forecasted peak demand. It must be sufficient enough to meet the loss of the most heavily loaded unit in the system. This has to satisfy the equation

$$\sum_{i=1}^{N} P\max_i U_{it} \geq (PD_t + R_t) \quad 1 \leq t \leq T\,, \tag{5}$$

where

$P\max_i$ – Maximum generation limit of unit i
R_t – spinning reserve at time t (MW)
T – scheduled time horizon (24 h).

3) Thermal Constraints

The temperature and pressure of the thermal units vary very gradually and the units must be synchronized before they are brought online. A time period of even 1 hour is considered as the minimum down time of the units. There are certain factors, which govern the thermal constraints, like minimum up time, minimum down time and crew constraints.

a) Minimum up time:

If the units have already been shut down, there will be a minimum time before they can be restarted and the constraint is

$$Ton_i \geq Tup_i\,, \tag{6}$$

where

Ton_i – duration for which unit i is continuously ON (h)
Tup_i – unit i minimum up time (h).

b) Minimum down time:

If all the units are running already, they cannot be shut down simultaneously and the constraint is

$$Toff_i \geq Tdown_i \,, \tag{7}$$

where

$Tdown_i$ –unit i minimum down time (hours)
$Toff_i$ –duration for which unit i is continuously OFF (hours)

4) Must Run Units

Generally in a power system, some of the units are given a must run status in order to provide voltage support for the network.

vskip4mm

3 GENETIC ALGORITHM

3.1 Introduction

The genetic algorithm is essentially a search algorithm based on the mechanics of natural selection and natural genetics. It combines solution evaluation with randomized, structured exchanges of information between solutions to obtain optimality. GA is a robust approach because no restrictions on the solution space are made during the search process. The power of this algorithm comes from its ability to exploit historical information structures from previous solution guesses in an attempt to increase performance of future solution structures. By simulating "the survival of the fittest" criterion of Darwinian evaluation among chromosome structures, the optimal solution is searched by randomized information exchange. While randomized, GA is not a simple random walk. It effectively exploits historical information to speculate on new search points with expected improved performance. In every generation, a new set of artificial chromosomes is created using bits and pieces of the fittest of the old ones. The three prime operators associated with the GA are reproduction, crossover and mutation [30, 31].

3.2 Genetic Components

1) Representation

In genetic algorithms the design variable or features that characterize an individual are represented in order list called string. Each design variable corresponds to gene and the string of design variables.

2) Initialization

GA's operate with a set of strings instead of a single string. This set or population of string goes through the process of evolution to produce new individual strings. The initial population is chosen at random. The initial population should contain a wide variety of structures.

3) Evaluation Function

Fitness is the value of the only objective function to be optimized. Evaluation is a procedure to determine the fitness of each string in the population and is very much application oriented. The performance of the algorithm is highly sensitive to the fitness values because GA proceeds in the direction of evolving better fit strings and the fitness value is the only information suitable to the GA.

3.3 Genetic Operators

Genetic operators are the stochastic transition rules employed by GA. These operators are applied on each string during each generation to generate a new improved population from old one. Casual produces a population consisting of gene. A gene is a key to the problem and consists of a number of bits. In the next pace each gene is evaluated and given a positive fitness rate. The fitness rate represents the probability and the optimality of a given gene. The gene and the equivalent fitness rate are referred to an individual. Depending on their fitness rate, a certain proportion of the population is chosen and deleted. The existing individuals are recombined and mutated. After the population has been evaluated, the selection process starts again.

3.3.1 Selection

After the estimation of the initial arbitrarily generated population the GA begins the formation of the new generation of solutions. The selection constraint, survival rate determines the number of individuals that are selected for reproduction. Elitism armed forces the best individual of the present population to be member of the next generation. The elitism function may put too to a great extent pressure on the elite and guide to premature convergence to a non-optimal point; therefore, an elite approach that mutates the elite structures is adopted. The genotypes are chosen using the Roulette wheel parent selection algorithm that selects a genotype with a probability comparative to the genotype's relative fitness with the population. Then a new off spring genotype is created by means of the following operators.

3.3.2 Reproduction

Reproduction is a procedure in which individual string are copied according to their objective function value. This operator emphasizes the survival of the fittest in GA's. It selects individual strings in the population according to their fitness. Population size affects the efficiency of the algorithm. If we have a smaller paper it would cover more space, resulting in poor performance. A larger poly would cover more space and prevents premature convergence to local solutions local or global. At the same time a large population needs more evaluations per on and may slow down the convergence rate.

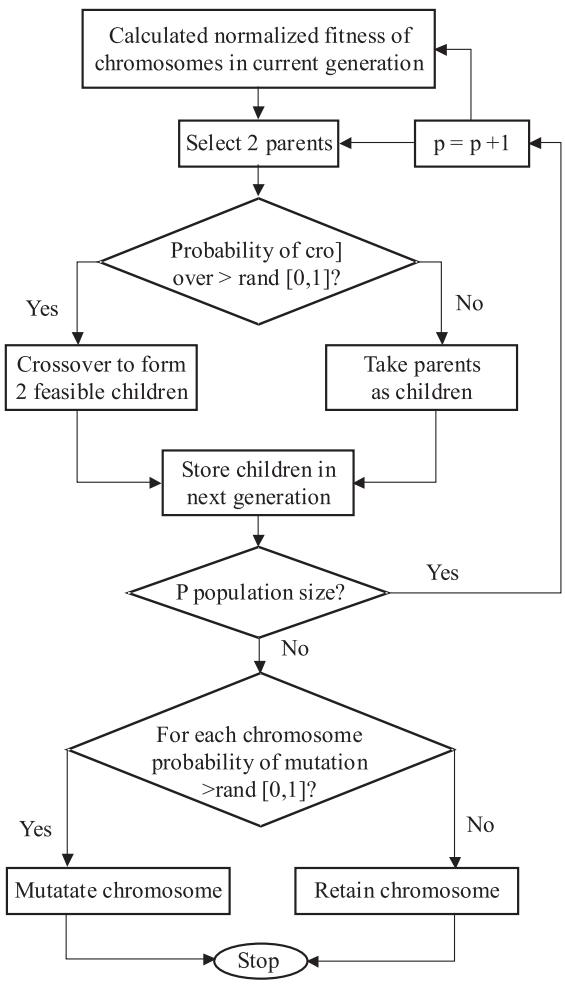

Fig. 1. Crossover and Mutation Operations in an iteration of basic GA

3.3.3 Cross over

Cross over is a recombination operator. After strings for mating are selected, a cross site is selected at random and bits are swapped between the strings following the cross site. Crossover is performed at a fairly high probability (*ie* 0.8 to 1). The cross over rate is the parameter that affects the rate at which the cross over operator is applied. A higher crossover rate introduces new strings more quickly into the population. If the cross over rate is too high performance strings are eliminated faster that selection can produce improvements. A low cross over rate may cause stagnation due to the lower exploration rate.

3.3.4 Mutation

After cross over, strings are subjected to mutation. Mutation involves switching a single bit in a random position in a string and thus introducing fresh genetic material. This is performed at a low probability (*ie* 0–0.005). The mutation of a bit does not affect the probability of mutation or others. Mutation rate is the probability with which each bit portion of any string in the new population undergoes a random change. It increases the diversity in the population low value of mutation rate helps to prevent any bit portion from getting strict to a single value.

3.4 GA Cycle

The basic genetic algorithm [30, 31] steps are

Step(1):Construct an initial population (P) of chromosomes by random process.

Step(2):Evaluate fitness of each chromosome.

Step(3):Genetic mating pool based on fitness function values.

Step(4):Select mating pair of chromosomes called parent chromosomes from mating pool.

Step(5):Create two child chromosomes from the parent chromosomes by applying genetic operators.

Step(6):Repeat Steps (4, 5), till the child population of size P is generated.

Step(7):Store the chromosome having the maximum fitness and also the corresponding objective function.

Step(8):Repeat Steps (2–7), until the specified numbers of genetic iterations are completed.

Step(9):Return the chromosome with highest fitness function as the solution.

3.5 Elitism

Probabilistic nature of the selection process gives a chance of reproduction even to the weakest number of the population. Likewise there is a chance that the best-performing member might not be present in the next generation due to structural changes following cross over and mutation. Hence it is derivable to copy the clite structures into the next gene. Best-fit string of ach gun is copied to next gun without undergoing cross over and mutation

3.6 Modifications to the Basic GA

Several modifications can be applied to the basic GA to improve the performance on practical problems [30, 31]. One of them is Elitism; Probabilistic nature of the selection process gives a chance of reproduction even to the weakest number of the population. Likewise, there is a chance that the best-performing member might not be present in the next generation due to structural changes following cross over and mutation. Hence it is desirable to copy the clite structures into the next gene. The best-fit string of each gun is copied to the next gun without undergoing cross over and mutation. In order to reduce the memory space requirement for storing the genes, the simple GA is altered in such a way that chosen genes replace the new child genes in the current generation. The dotted line path indicates this change in the flowchart given. The cross over and mutation operations in an iteration of the basic GA algorithm is shown in the flowchart of "Fig. 1". GA's have also been used cooperatively with other methods such as neural networks, heuristic methods, and the tabu search method. Many algorithms have been proposed on these lines, and one such representative method is given in [30, 31].

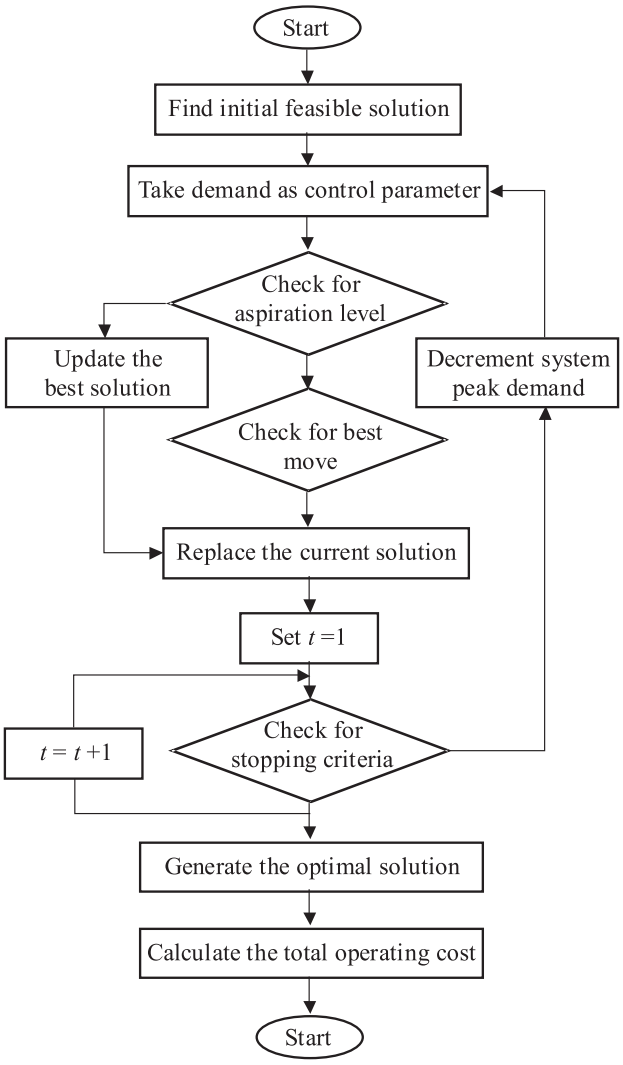

Fig. 2. Flowchart of TS algorithm

3.7 Unit Commitment Using GA

1. Initial binary coded solutions are produced at random to form the initial population.
2 Calculate the total power production cost of the committed units as given in (1) by solving the EDP.
3 Calculate and scale the fitness function and assigned to the initial population as given in (2).
4 Check the convergence criterion is satisfied. If yes go to stop. Otherwise, go to next step.
5 Copy the best solution of every generation.
6 Create the offspring with the application of rules for generating randomly feasible solutions by means of the crossover and mutation operations.

4 TABU SEARCH

4.1 Overview

To solve the UCP, two types of variables require being determined. The unit's status variables U and V are integer variables and the units, output power variables P are continuous variables. The problem can then be decomposed into two sub problems, a combinatorial problem in U and V and a non-linear optimization problem in P. TS are used to solve the combinatorial optimization and the non-linear optimization is solved via a quadratic programming routine [20]. The flowchart for TS is shown in Fig. 2. The proposed algorithm contains three major steps:

- First, generating randomly feasible trail solutions
- Second, calculating the objective function of the given solution by solving the EDP.
- Third, applying the TS procedures to accept or reject the solution in hand.

4.2 Tabu Search General Algorithm

Step(0): Assume that the fuel costs to be fixed for each hour and all the generators share the loads equally.

Step(1): By optimum allocation find the initial feasible solution (U_i, V_i).

Step(2): Demand is taken as the control parameter.

Step(3): Generate the trial solution.

Step(4): Calculate the total operating cost, F_t, as the summation of running cost and Start up — shut down cost.

Step(5): Tabulate the fuel cost for each unit for every hour.

4.3 Generating Trial Solution

The neighbours should be randomly generated, feasible, and span as much as possible the problem solution space. 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. The implementation of new rules to obtain randomly feasible solutions faster are done by the rules is described in [20].

4.4 Generating an Initial Solution

The TS algorithm requires a starting feasible schedule, which satisfies all the system and units constraints. This schedule is randomly generated. The algorithm given in [20] is used for finding this starting solution.

4.5 Operating Cost Calculation

Once a trail solution is obtained, the corresponding total operating cost is determined. Since the production cost is a quadratic function, the EDP is solved using a quadratic programming routine. The start-up cost is then calculated for the given schedule.

4.6 Stopping Criteria

There may be several stopping criteria for the search. For this implementation, the search is stopped if the following conditions are satisfied:

- The load balance constraints are satisfied.
- The spinning reserve constraints are satisfied.

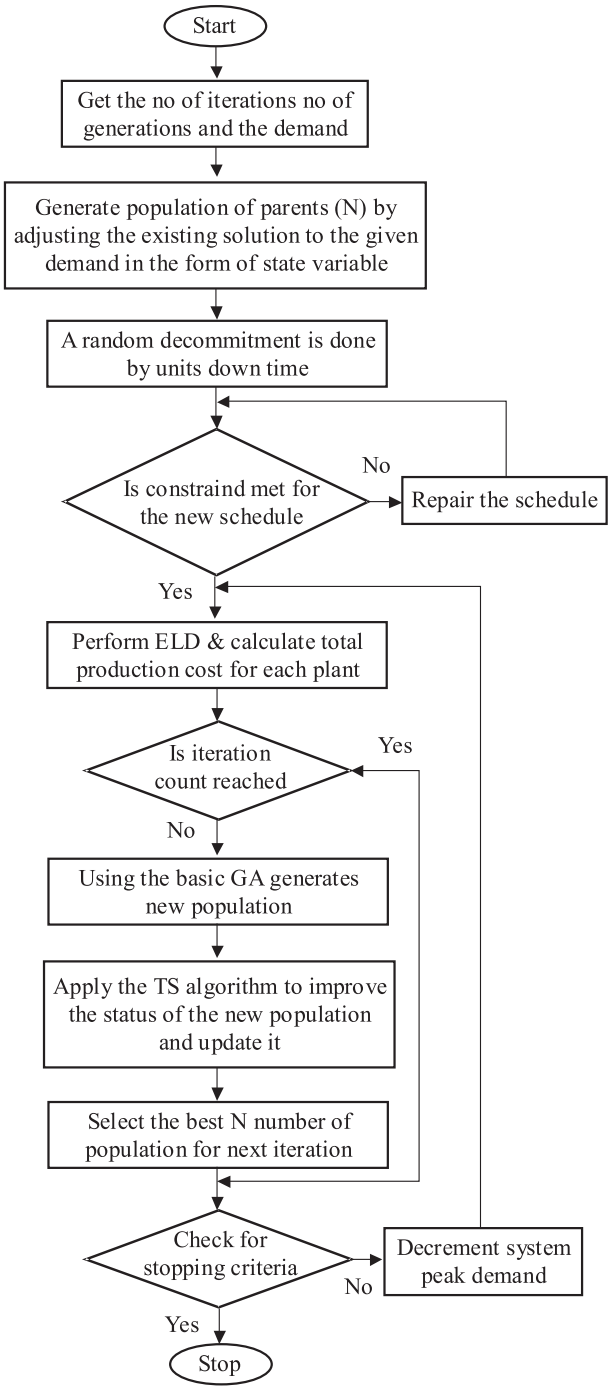

Fig. 3. Flowchart for TS embedded GA algorithm

4.7 Tabu List

TL is controlled by the trial solutions 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". Empirically, TL sizes, which provide good results, often grow with the size of the problem and stronger restrictions are generally coupled with smaller sizes [20]. Best sizes of TL lie in an intermediate range between these extremes. In some applications a simple choice of TL size in a range cantered on seven seems to be quite effective.

4.8 Aspiration Criteria

This is another important criteria of TS arises when the move under consideration has been found to be associated with each entry in the tabu list there is a certain value for the evaluation function called "Aspiration Level". Normally, the aspiration level criteria are designed to override tabu status if a move is "good enough" [20].

4.9 Implementation

To solve for the UCP using TS, software in Turbo C package is developed. The software provides interactive approach in dealing with the various data input required for solving the UCP from the user.

5 GA BASED TS ALGORITHM FOR UCP

In solving the UCP, two types of variables need to be determined. The unit's status variables U and V are integer variables and the units, output power variables P are continuous variables. The problem can then be decomposed into two sub problems, a combinatorial problem in U and V and a non linear optimization problem in P. TS is used to solve the combinatorial optimization and the nonlinear optimization is solved via a quadratic programming routine.

In the TS technique for solving UCP, Initial Operating Schedule status in terms of maximum real power generation of each unit is given as input. As we know that TS is used to improve any given status by avoiding entrapment in local minima, the offspring obtained from the GA algorithm is given as input to TS and the refined status is obtained. And in this method, no advanced operators are required. The complete GATS algorithm is as follows:

Step (1): Get the demand for 24 hours and number of iterations to be carried out.

Step (2): Generate population of parents (N) by adjusting the existing solution to the given demand to the form of state variables.

Step (3): Unit down time makes a random recommitment.

Step (4): Check for constraint in the new schedule by TS algorithm.

Step (5): Perform ELD and calculate the total production cost for each parent.

Step (6): Has final iteration count reached? If yes go to Step (10) else go to step (7).

Step (7): Using the basic GA generates a new population.

Step (8): Improve the status of the new population and update it by TS algorithm.

Step (9): Select the best N number of population for next iteration.

Step (10): Stop if stopping criterion is satisfied. Otherwise go to Step (5).

Step (11): For the units, which are in the off states, calculate the cost for both cooling and banking.

Step (12): Compare the cooling and banking costs, if banking cost is lesser than cooling, bank the unit.

Step (13): Print the optimum schedule.

The complete algorithm is shown as flowchart in Fig. 3.

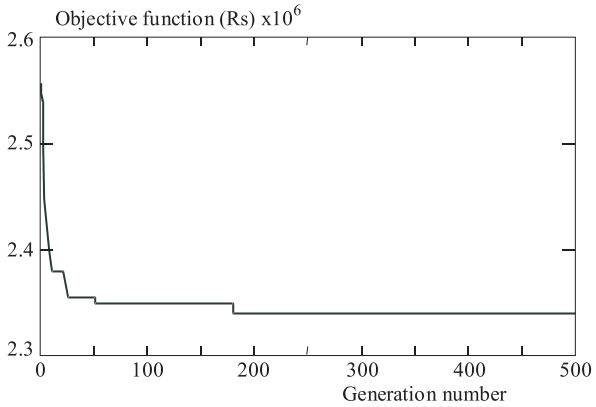

Fig. 4. Convergence of the GATS Algorithm.

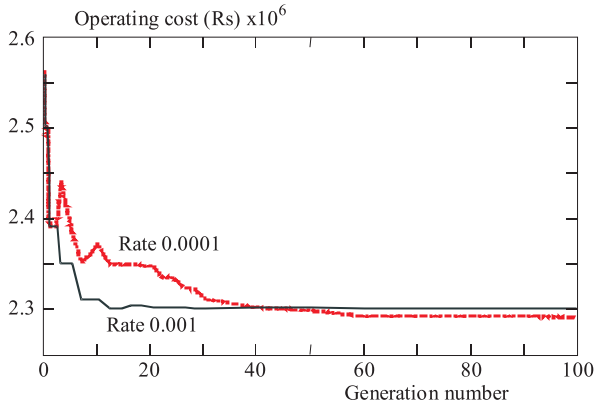

Fig. 5. Optimal Solution for Various Mutations Rate

Table 1. Daily Generation of Seven Units in MW

Hour	P max	I	II	III	IV	V	VI	VII
1	840	60	80	100	101	149	150	200
2	757	60	0	100	100	147	150	200
3	775	60	0	100	115	150	150	200
4	773	60	0	100	113	150	150	200
5	770	60	0	100	110	150	150	200
6	778	60	0	100	118	150	150	200
7	757	60	0	100	100	147	150	200
8	778	60	0	100	118	150	150	200
9	770	60	0	100	110	150	150	200
10	764	60	0	100	104	150	150	200
11	598	60	0	99	97	142	0	200
12	595	60	0	100	96	139	0	200
13	545	0	0	100	99	146	0	200
14	538	0	0	99	97	142	0	200
15	535	0	0	100	95	139	0	200
16	466	0	0	0	116	150	0	200
17	449	0	0	0	101	148	0	200
18	439	0	0	0	97	142	0	200
19	466	0	0	0	116	150	0	200
20	463	0	0	0	113	150	0	200
21	460	0	0	0	110	150	0	200
22	434	0	0	0	95	139	0	200
23	530	60	0	0	120	150	0	200
24	840	60	80	100	101	149	150	200

5.1 Repair Mechanism

A repair mechanism to restore the feasibility of the constraints is applied and described [20, 26] as follows:

- Pick at random one of the OFF units at one of the violated hours.
- Apply the rules in section 4.3 to switch the selected unit from OFF to ON keeping the feasibility of the down time constraints.
- Check for the reserve constraints at this hour. Otherwise repeat the process at the same hour for another unit.

5.2 Making offspring feasible

While solving the constrained optimization problem, there are various techniques to repair an infeasible solution [20]. In this paper we have chosen the technique, which evolve only the feasible solutions. That is the schedule, which satisfies the set of constraints as mentioned earlier. This is achieved by the minimum up/down constraints satisfaction, which are the most important constraints in the UCP, while the reserve constraints are checked and corrected, if necessary, using a repair mechanism in section 5.1. Before the best solution is copied, the trail is made to correct the unwanted mutations.

5.3 Implementation

Software programs were developed using MATLAB software package and the test problem was simulated for 10 independent trials using Genetic Algorithm. The training and identification part as implemented in the tabu search technique is employed here and considered as a process involving random recommitment, constraint verification and offspring creation.

6 NUMERICAL RESULTS

A power station in India with seven generating units, each with a capacity of 210 MW has been considered as a case study. A time period of 24 hours is considered; the unit commitment problem is solved for these seven units and also compared. The required inputs for solving the UCP are briefed here. The total number of generating units, the maximum real power generation of each unit and the cost function parameters of each unit are tabulated for a day, respectively, as shown in Tab. 1 and Tab. 2 for the power station. The status of unit i at time t and the start-up/shut-down status obtained are the necessary solution for TS, GA, GATS, DP, LR methods for the power station. The comparison of the total costs and Central Processing unit (CPU) time is shown in Tab. 3. Figure 2 shows the convergence characteristics of GATS. Figure 3 shows the optimal solution for various mutations rate. From these results, the GATS method with cooling-banking constraints had lesser total cost and took lesser CPU time in all the power systems considered including the power station considered.

Table 2. Generation System Operation Data

Unit	P min (MW)	P max (MW)	Running cost C_i (Rs)	B_i (Rs/MWh)	A_i (Rs/MWh^2)	Start-up cost So_i (Rs)	D_i (Rs)	E_i (Rs)
1	15	60	750	70	0.255	4250	29.5	10
2	20	80	1250	75	0.198	5050	29.5	10
3	30	100	2000	70	0.198	5700	28.5	10
4	25	120	1600	70	0.191	4700	32.5	9
5	50	150	1450	75	0.106	5650	32	9
6	50	150	4950	65	0.0675	14100	37.5	4.5
7	75	200	4100	60	0.074	11350	32	5.5

Table 3. Comparisons of total cost and CPU time

System	Methods	Total Cost (pu)	CPU Time (Sec)
7-Unit Utility System	DP	1.00000	130
	LR	0.97843	115
	TS	0.94780	84
	GA	0.94005	77
	GATS (Without Cooling-Banking)	0.93953	71
	GATS (With Cooling-Banking)	0.93479	69
IEEE 10-Unit System	DP	1.00000	325
	LR	0.94123	279
	TS	0.93780	267
	GA	0.90952	221
	GATS (Without Cooling-Banking)	0.90752	215
	GATS (With Cooling-Banking)	0.90351	213
IEEE 26-Unit System	DP	1.00000	409
	LR	0.95968	395
	TS	0.94980	370
	GA	0.91743	342
	GATS (Without Cooling-Banking)	0.91532	338
	GATS (With Cooling-Banking)	0.91098	335
IEEE 34-Unit System	DP	1.00000	555
	LR	0.99910	545
	TS	0.97752	519
	GA	0.94127	487
	GATS (Without Cooling-Banking)	0.94113	481
	GATS (With Cooling-Banking)	0.93710	478

In our proposed method, the GA method was used in conjunction with the tabu search method. As we indicated in the paper, the GA algorithm has also proved to be an efficient tool for solving the important economic dispatch problem for units with "non-smooth" fuel cost functions as referred in [20]. Such functions may be included in the proposed GA search for practical problem solving. There is no obvious limitation on the size of the problem that must be addressed, for its data structure is such that the search space is reduced to a minimum; no "relaxation of constraints" is required; instead, populations of feasible solutions are produced at each generation and through out the evolution process. The main advantages of the proposed algorithm are speed. The proposed GATS approach was compared to the related methods in the references indented to serve this purpose, such as the DP with a zoom feature, the SA, and the GA approaches. By means of stochastically searching multiple points at one time and considering trail solutions of successive generations, the GATS approach avoids entrapping in local optimum solutions. Also, disadvantages of huge memory size required by the SA method are eliminated. Moreover, intellectual schemes of encoding and decoding entailed by the GA approach are not needed in the proposed GATS approach. The problem of power unbalance previously existing in the solution of GA is circumvented as well in this paper. In comparison with the results produced by the referenced techniques, the GATS method obviously displays a satisfactory performance with respect to the quality of its evolved solutions and to its computational requirements.

7 CONCLUSION

This paper presents a GATS method with cooling-banking constraints to the unit commitment problem. In this method to the UCP, the essential process simulated in the procedure is mutation, competition, and selection. The mutation rate is computed as a function of the ratio of the total cost by the schedule of interest to the cost of the best schedule in the current population. The best solutions to form the basis of the subsequent generation are selected from among the parents and the offspring in the current population. In this proposed work, the parents are obtained from a pre-defined set of solution's i.e. each and every solution is obtained from the TS method. Then, a random recommitment is carried out with respect to the unit's minimum down times. In comparison with the results produced by the referenced techniques (GA, DP, LR, TS), the GATS method obviously displays a satisfactory performance. There is no obvious limitation on the size of the problem that must be addressed, for its data structure is such that the search space is reduced to a minimum. No relaxation of constraints is required; instead, populations of feasible solutions are produced at each generation and throughout the process. And GA is

inherently parallel; the proposed method is suitable for parallel and distributed implementation.

References

[1] SNYDER, W. L.—POWELL,, H. D.—RAYBURN, C.: Dynamic Programming Approach to Unit Commitment, IEEE Trans. Power Systems **3** No. 2 (1987), 339–350.

[2] HSU, Y. Y.—SU, C. C.—LIANG, C. C.—LIN, C. J.—HUANG, C. T.: Dynamic Security Constrained Multi-Area Unit Commitment, IEEE Trans. Power Systems **6** (1991), 1049–1055.

[3] TAKRITI, S.—BIRGE, J. R.: Using Integer Programming to Refine Lagrangian – Based Unit Commitment Solutions, IEEE Trans. Power Systems **15** No. 1 (2000), 151-156.

[4] MOHAN, M. R.—KUPPUSAMYK.—KHAN, M. A.: Optimal Short-term Hydro-Thermal Scheduling using Decomposition Approach and Linear Programming Method, International Journal of Electrical Power and Energy Systems **14** No. 1 (1992), 39–44.

[5] TAO LI—SHAHIDEHPOUR, M.: Price Based Unit Commitment: A Case of Lagrangian Relaxation versus Mixed Integer Programming, IEEE Trans. Power Systems **20** No. 4 (2005), 2015–2025.

[6] NGUNDAM, J. M.—KENFACK, F.—TATIETSE, T. T.: Optimal Scheduling of Large-Scale Hydrothermal Power Systems Using the LR Technique, International Journal of Electrical Power & Energy Systems **22** No. 1 (2002.), 237–245.

[7] BAKIRTZIS, A. G.—ZOUMAS, C. E.: Lambda of Lagrangian Relaxation Solution to Unit Commitment Problem, IEE Proc. Generation, Transmission and Distribution **147** No. 2, (2000), 131–136.

[8] CHANG, C-P.—LIU, C. W.—LIU, C. C.: Unit Commitment by LR and GA, IEEE Trans. Power Systems **15** No. 2 (2000), 707–714.

[9] NIEVA, R.—INDA, A.—GUILLEN, I.: Lagrangian Reduction of Search Range for Large Scale Unit Commitment, IEEE Trans. Power Systems **2** (1987), 465–473.

[10] ONGSAKUL, W.—PETCHARAKS, N.: Unit Commitment by Enhanced Adaptive Lagrangian Relaxation, IEEETrans. Power Systems **19** No. 1 (2004), 620-628.

[11] COHEN, A. I.—YOSHIMURA, M.: A Branch and Bound Algorithm for Unit Commitment, IEEE Trans. Power and Apparatus **102** No. 2 (1983), 444–451.

[12] OUYANG, Z.—SHAHIDERPOUR, S. M.: Short term Unit Commitment Expert System, International Journal of Electrical Power System Research, **20** (1990), 1–13.

[13] SU, C. C.—HSU, Y. Y.: Fuzzy Dynamic Programming: an Application to Unit Commitment, IEEE Trans. Power Systems **6** No. 3 (1991), 1231–1237.

[14] SASAKI, H.—WATANABE, M.—KUBOKAWA, J.—YORINO, N.: A Solution Method of Unit Commitment by Artificial Neural Networks, IEEE Trans. Power Systems **7** No. 3 (1992.), 974–981.

[15] SAKSORNCHAI, T.—LEE, W.-J.—METHAPRAYOON, K.—LIAO, J. R.—ROSS, R. J.: Improve the Unit Commitment Scheduling by Using the Neural-Network-Based Short-Term Load Forecasting, IEEE Trans. Industry Applications **40** No. 1 (2005), 169–179.

[16] PADHY, N. P.: Unit Commitment using Hybrid Models: A Comparative Study for Dynamic Programming, Expert Systems, Fuzzy System and GA, International Journal of Electrical Power & Energy Systems **23** No. 1 (2000), 827–836.

[17] ZHUANG, F.—GALIANA, F. D.: Unit Commitment by Simulated Annealing, IEEE Trans. On Power Systems **5** No. 1 (1990), 311–318.

[18] KIRKPATRICK, S.—GELATT, C. D. JR.—VECEHI, M. P.: Optimization by Simulated Annealing Science,.

[19] MANTAWY, A. H.—ABDEL-MAGID, J. L.—SELIM, S. Z.: A Simulated Annealing Algorithm for Unit Commitment, IEEE Trans. On Power Systems, **13** No. 1 (1998).

[20] MANTAWY, A. H.—ABDEL-MAGID, J. L.—SELIM, S. Z.: A Unit Commitment By Tabu Search, IEE Proc. Generation, Transmission and Distribution **145** No. 1 (1998), 56–64.

[21] LIN, W.-M.—CHENG, F.-S.—TSAY, M.-T.: An Improved Tabu Search for Economic Dispatch With Multiple Minima, IEEE Trans. Power Systems **17** No. 1 (2002), 108–112.

[22] WU, Y.-G.—HO, C.-Y.—WANG, D.-Y.: A Diploid Genetic Approach to Short-Term Scheduling of Hydro-Thermal System, IEEE Trans. Power Systems **15** (2000), 1268–1274.

[23] HONG, Y.-Y.—LI, C.-Y.: Genetic Algorithm Based Economic Dispatch for Cogeneration Units Considering Multiplant Multibuyer Wheeling, IEEE Trans. Power Systems **17** No. 1 (2002), 134–140.

[24] DAMOUSIS, I. G.—BAKIRTZIS, A. G.—DOKOPOULOUS, P. S.: A Solution to the Unit Commitment Problem Using Integer-Coded Genetic Algorithm, IEEE Trans. Power Systems **19** No. 2 (2004), 1165–1172.

[25] SENJYU, T.—SABER, A. Y.—MIYAGI, T.—SHIMABUKURO, K.—URASAKI, N.—FUNABASHI, T.: Fast Technique for Unit Commitment by Genetic Algorithm Based on Unit Clustering, IEEE Trans. Power Systems **152** No. 5 (2005), 705–713.

[26] MANTAWY, A. H.—ABDEL-MAGID, Y. L.—SELIM, S. Z.: Integrating Genetic Algorithm, Tabu Search and Simulated Annealing for the Unit Commitment Problem, IEEE Trans. Power Systems **14** No. 3 (1999), 829–836.

[27] BAI, X.—SHAHIDEHPOUR, M.: Hydro-Thermal Scheduling by Tabu Search and Decomposition Method, IEEE Trans. Power Systems **11** No. 2 (1996), 968–975.

[28] BAI, X.—SHAHIDEHPOUR, M.: Extended Neighbourhood Search Algorithm for Constrained UC, Elect. Power and Energy Systems **19** No. 5 (1997), 349–356.

[29] WOOD, A. J.—WOOLENBERG, B. F.: Power Generation and Control, 2nd Edn., John Wiley and Sons, New York, 1996.

[30] GOLDBERG, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning, Eastern Press, Addison-Wesley, 1999.

[31] MITCHELL, M.: An Introduction To Genetic Algorithms, MIT Press, U.S.A., 2002.

Received 14 September 2008

Christober C. Asir Rajan, born in 1970 and received his BE (Distn) degree (Electrical and Electronics) and ME (Distn) degree (Power System) from the Madurai Kamaraj University (1991, 1996), Madurai, India. And he received his postgraduate degree in DIS (Distn) from the Annamalai University, Chidambaram. He did PhD in Power System at College of Engineering, Guindy, Anna University (2001-2004), Chennai, India. He published technical papers in International and National Journals and Conferences. He is currently working as Associate Professor in Electrical Engineering Department at Pondicherry Engineering College, Pondicherry, India. His area of interest is power system optimization, operational planning and control. He acquired Member in ISTE and MIE in India.