INTEGRATING GENETIC ALGORITHMS, TABU SEARCH, AND SIMULATED ANNEALING FOR THE UNIT COMMITMENT PROBLEM

A. H. Mantawy, Member **Youssef L. Abdel-Magid,** Senior Member **Shokri Z. Selim**
Electrical Engineering Department Systems Engineering Department
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

***Abstract*- This paper presents a new algorithm based on integrating genetic algorithms, tabu search and simulated annealing methods to solve the unit commitment problem. The core of the proposed algorithm is based on genetic algorithms. Tabu search is used to generate new population members in the reproduction phase of the genetic algorithm. Simulated annealing method is used to accelerate the convergence of the genetic algorithm by applying the simulated annealing test for all the population members. A new implementation of the genetic algorithm is introduced. The genetic algorithm solution is coded as a mix between binary and decimal representation. The fitness function is constructed from the total operating cost of the generating units without penalty terms. In the tabu search 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 genetic algorithm. A simple cooling schedule has been implemented to apply the simulated annealing test in the algorithm. Numerical results showed the superiority of the solutions obtained compared to genetic algorithms, tabu search and simulated annealing methods, and to two exact algorithms.**

1. INTRODUCTION

The Unit Commitment Problem (UCP) is the problem of selecting the power generating units to be in service during a scheduling period and for how long. The committed units must meet the system load and reserve requirements at minimum operating cost, subject to a variety of constraints. The Economic Dispatch Problem (EDP) is to optimally allocate the load demand among the running units while satisfying the power balance equations and units operating limits [1-17].

The exact solution of the UCP can be obtained by a complete enumeration of all feasible combinations of generating units, which could be a very huge number, while the economic dispatch problem is solved for each feasible combination. However, the high dimension of the possible solution space is the real difficulty in solving the problem.

PE-098-PWRS-0-08-1998 A paper recommended and approved by the IEEE Power System Analysis, Computing and Economics Committee of the IEEE Power Engineering Society for publication in the IEEE Transactions on Power Systems. Manuscript submitted November 12, 1997; made available for printing August 14, 1998.

Artificial intelligence techniques have come to be the most widely used tool for solving many optimization problems. These methods (e.g., genetic algorithms, simulated annealing and tabu search) seem to be promising and are still evolving.

Genetic Algorithms (GA) have become increasingly popular in recent years in science and engineering disciplines. Some works have been published [11-15] covering the solution of the UCP using GA. Solution coding and fitness functions are the most important issues in solving problems using GA. In the literature, UCP solutions coding have been used in the binary form. The fitness function has been constructed as the summation of the objective function and penalty terms for constraints violations.

Tabu Search (TS) is a powerful optimization procedure that has been successfully applied to a number of combinatorial optimization problems [22-27]. It has the ability to avoid entrapment in local minima by employing a flexible memory system. In [10], a simple TS algorithm based on short-term memory has been proposed for solving the UCP. Specific attention is given here to the short-term memory component of TS, which has been considered as the simplest form of TS procedures [10].

Simulated Annealing (SA), is a powerful technique for solving combinatorial optimization problems. It has the ability of escaping local minima by incorporating a probability function in accepting or rejecting new solutions. A main advantage of the SA method is that it does not need large computer memory. A simple cooling schedule has been used [9] to simplify and speed up the computation.

In this paper we propose a new hybrid algorithm (GTS) for solving the UCP. The algorithm integrates the main features of the GA, TS and SA algorithms. A new implementation of GA as applied to the UCP is presented. TS is incorporated in the reproduction phase of the GA to generate new members in the GA population. The effect of injecting new members in each GA generation is to help escaping the local minimum and to prevent the premature convergence of the GA. Moreover, SA method is explored to improve the convergence of the GA by testing the population members of the GA after each generation. The SA test allows the acceptance of any solution at the beginning of the search, while only good solutions will have higher probability of acceptance as the generation number increases.

Several examples are solved to test the proposed algorithm. The results show an improvement in the quality of solutions compared with results of other methods in the literature [5,6] and with our results in [9-12].

In the next section, a mathematical formulation of the problem is introduced. In Section 3, the proposed GTS

0885-8950/99/$10.00 © 1998 IEEE

algorithm is described. Sections 4, 5 and 6 present the detailed description of the implemented GA, TS and SA components. In Section 7, the computational results along with a comparison to previously published work are presented. Section 8 outlines the conclusions. In the Appendix a general background of the GA, TS and the SA methods is also presented.

2. PROBLEM STATEMENT

In the UCP under consideration, one is interested in a solution that minimizes the total operating cost of the generating units during the scheduling time horizon while several constraints are satisfied [1,8-12].

2.1 The Objective function

The overall objective function of the UCP of N generating units for a scheduling time horizon T, (e.g., 24 HRs), is:

$$F_T = \sum_{t=1}^{T} \sum_{i=1}^{N} (U_{it} F_{it}(P_{it}) + V_{it} S_{it}) \ \$ \qquad (1)$$

Where

U_{it} : is status of unit i at hour t (ON=1, OFF=0).

V_{it} : is start-up/shut-down status of unit i at hour t.

P_{it} : is the output power from unit i at time t

The production cost, $F_{it}(P_{it})$, of a committed unit i, is conventionally taken in a quadratic form:

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

Where, A_i, B_i, C_i : are the cost function parameters of unit i.

The start-up cost, S_{it}, is a function of the down time of unit i [6]:

$$S_{it} = So_i [1 - D_i \exp(-Toff_i / Tdown_i)] + E_i \ \$ \qquad (3)$$

Where, So_i : is unit i cold start-up cost, and

D_i, E_i : are start-up cost coefficients for unit i.

2.2 The Constraints

The constraints that have been taken into consideration in this work, may be classified into two main groups:

(i) System Constraints:

a- Load demand constraints:

$$\sum_{i=1}^{N} U_{it} P_{it} = PD_t \ ; \forall t \qquad (4)$$

Where PD_t : is the system peak demand at hour t (MW).

b- Spinning Reserve

Spinning reserve, R_t, is the total amount of generation capacity available from all units synchronized (spinning) on the system minus the present load demand.

$$\sum_{i=1}^{N} U_{it} Pmax_i \geq (PD_t + R_t) \ ; \forall t \qquad (5)$$

(ii) Unit constraints:

The constraints on the generating units are

a- Generation limits

$$U_{it} Pmin_i \leq P_{it} \leq Pmax_i U_{it} \ \forall i, t \qquad (6)$$

Where, $Pmin_i, Pmax_i$ is minimum and maximum generation limit (MW) of unit i, respectively.

b- Minimum up/down time

$$\begin{matrix} Toff_i \geq Tdown_i \\ Ton_i \geq Tup_i \end{matrix} \ ; \forall i \qquad (7)$$

Where $Tup_i, Tdown_i$ are unit i minimum up/down time.

$Ton_i, Toff_i$ are time periods during which unit i is continuously ON/OFF.

c- Unit initial status

d- Crew constraints

e- Unit availability; e.g. , must run, unavailable, available, or fixed output (MW).

f- Unit derating

3. THE PROPOSED ALGORITHM

3.1 Overview

In solving the UCP, two types of variables need to be determined U_{it} and V_{it}, which are 0-1 (binary) variables, and the units output power variables. P_{it}, which are continuous variables. The first is a combinatorial optimization problem while the second is a nonlinear one. A hybrid of GA, TS and SA is proposed to solve the combinatorial optimization. The nonlinear optimization (EDP) is simultaneously solved via a quadratic programming routine.

Fig. (1) shows the flow chart of the proposed GTS algorithm. The major steps of the algorithm are summarized as follows:

(a) Create an initial population by randomly generating a set of feasible solutions (chromosomes) [9].

(b) Evaluate each chromosome by solving the EDP.

(c) Determine the fitness function of each chromosome in the population.

(d) Apply GA operators to generate new populations as follows:

- Copy the best solution from the current to the new population
- Use TS algorithm to generate new members in the new population (typically 1-10% of the population size), as neighbors to randomly selected solutions in the current population.
- Apply the crossover operator to complete the members of the new population.
- Apply the mutation operator to the new population.

(e) Apply SA algorithm to test the members of the new population

3.2 Stopping Criteria

There are several possible stopping conditions for the search. In our implementation, we stop the search if one of the following two conditions is satisfied in the order given:

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

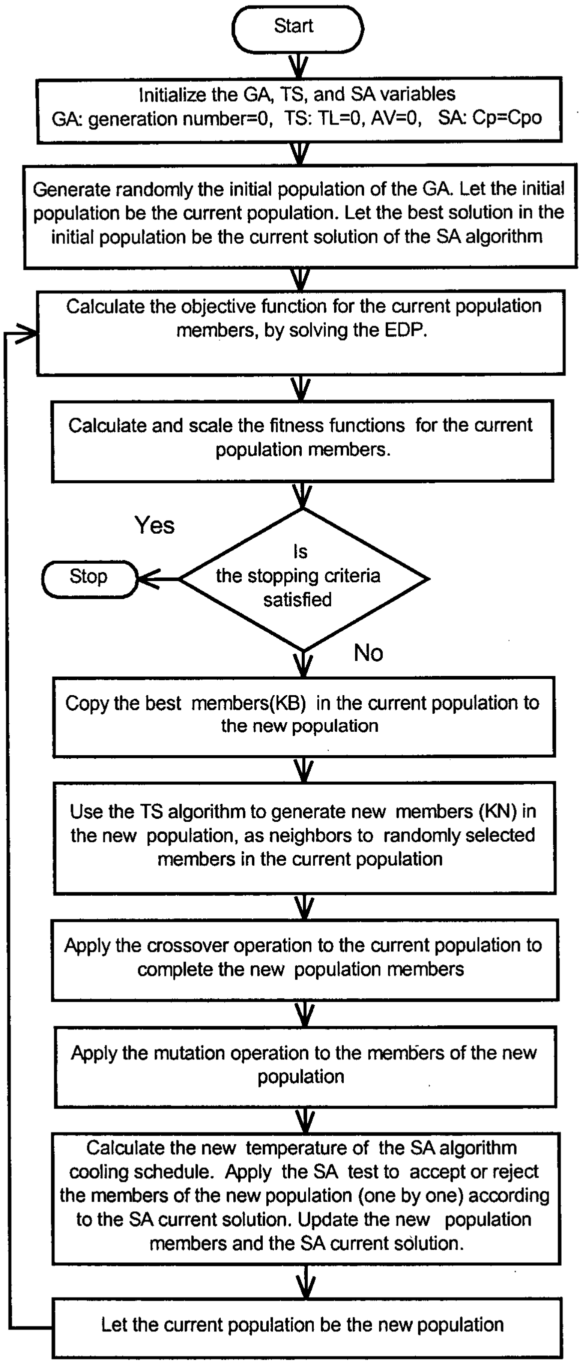


Fig. (1) Flow chart of the proposed GTS Algorithm

4. GA IMPLEMENTATION IN THE GTS ALGORITHM

The details of the GA components implementation are described in [11,12] and are summarized here as follows:

4.1 Solution Coding

The solution in the UCP is represented by a binary matrix (U) of dimension TxN (Fig.(2-a)). The proposed method for coding is a mix between binary and decimal numbers. Each column vector in the solution matrix (which represents the operation schedule of one unit) of length T 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(Fig.(2-b)). Typically the numbers U1,U2, ..,UN are integers ranging between 0 and $2^N - 1$. Accordingly, a population of size NPOP is stored in a matrix NPOPxN (Fig.(2-c)).

H	Units							
R	1	2	3	4	.	.	.	N
1	1	1	0	0	.	.	.	1
2	1	1	0	0	.	.	.	1
3	1	0	1	0	.	.	.	0
.	.	.	.	.	.	.	.	.
T	0	1	0	1	.	.	.	0

Fig.(2-a) The binary solution matrix U

U1	U2	U3	U4	.	.	.	UN

Fig.(2-b) The equivalent decimal vector(one chromosome)

U1	U2	U3	U4	.	.	.	UN
U1	U2	U3	U4	.	.	.	UN
.	.	.	.	.	.	.	.
U1	U2	U3	U4	.	.	.	UN

Fig.(2-c) Population of size NPOP chromosomes

4.2 fitness function

Unlike the previous solutions of the UCP using GA [13-15], the fitness function is taken as the reciprocal of the total operating cost in (1), since we are generating always feasible solutions [11,12]. The fitness function is then scaled to prevent premature convergence. Linear scaling is used which requires a linear relationship between the original fitness function and the scaled one [17].

4.3 Crossover

To speed up the calculations, the crossover operation is done between two chromosomes in their decimal form. Two parents are selected according to the roulette wheel rule. Two positions in the two chromosomes are selected at random. The decimal numbers are exchanged between the two parents to produce two children. The two children are then decoded into their binary equivalent and checked for constraints violation (load demand and/or reserve constraints). If the constraints are not satisfied a repair mechanism is applied to restore feasibility to the produced children.

4.4 Mutation

Mutation operation is done by randomly selecting any chromosome with a prespecified probability. The selected chromosome is then decoded into its binary equivalent. A unit number and a time period are randomly selected. Then the proposed rules in [9] are applied to reverse the status of this unit keeping the feasibility of the unit constraints related to its minimum up/down times. A check for the changed time periods, and correction if necessary, for the reserve constraints is then made.

4.5 Generating Feasible Trial Solutions

In [9], we have proposed some rules to generate randomly a feasible trial solution as a neighbor to an existing feasible solution. These rules were designed to achieve the minimum up/down constraints satisfaction, which are the most difficult constraints in the UCP, while the reserve constraints are checked and corrected, if necessary, using a repair mechanism (Sec. 4.6). The main idea of these rules could be summarized in two points. First, the difference between minimum up or down time of a unit is subtracted from the ON or OFF hours of that unit. Second, the unit status is reversed randomly at some hours ranging between 0 and this difference.

4.6 Repair Mechanism

Due to applying the crossover and mutation operations, the reserve constraints might be violated. A repair mechanism to restore the feasibility of these constraints is applied and described as follows:

- Pick at random one of the OFF units at one of the violated hours.
- Apply the rules in Sec. 4.5 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. If satisfied go to another hour. Otherwise, repeat the process at the same hour for another unit.

This procedure has proven faster than algorithms that use penalty functions [13-15].

5. TS IMPLEMENTATION IN THE GTS ALGORITHM

In the proposed algorithm TS is used to generate new neighbors to randomly selected members of the GA populations. The flow chart of Fig.(3) describes the main step of the TS algorithm. The implementation details of the TS algorithm are described in following sections [10].

5.1 Tabu-list approach for UCP

In this work, TL for the UCP is created as a matrix of dimension ZxN, where Z and N are the TL size, and number of units respectively. Each vector in the matrix represents the TL for one generating unit. For example, each entry in the vector i records the equivalent decimal number of the binary representation of a specific trial solution ($\mathbf{U}_{it}$, t=1,T) for unit i. By using this approach we record all information of the unit trial solution using minimum memory requirements. Fig. (4) shows an initial trial solution for certain unit and four trial solutions generated from each other in sequence. It also shows the implementation of the TL for these four trial solutions. Each entry of the TL represents the equivalent decimal number for one binary representation of the trial solutions. It is clear that this approach insures the uniqueness of the representation of a specific trial solution.

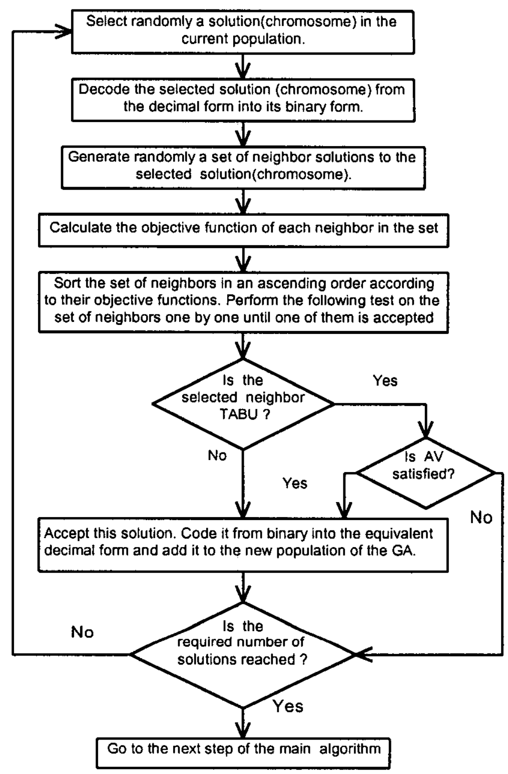


Fig. (3) Tabu search part of the GST Algorithm

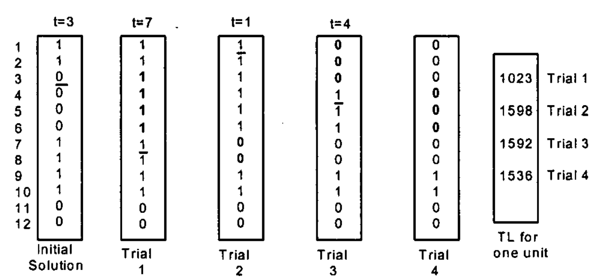


Fig. (4) Trial solutions for one unit and the corresponding TL (T=12, Z=4)

5.2 Aspiration Criteria

Different forms of aspiration criteria are used in the literature. The one we used in this work is to override the tabu status if the current solution associated with tabu status has a better objective function than the one obtained before, for the same move.

6. SA IMPLEMENTATION IN THE GTS ALGORTIHM

The steps of the SA algorithm as applied at the kth generation of the proposed algorithm are described as follows[9]:

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

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

Step(3): Acceptance test: If $E_j \le E_i$, then accept the population member, set $X_i = X_j$ and go to Step (4). Otherwise: If $\exp[(E_i - E_j)/Cp] \ge U(0,1)$ set $X_i = X_j$ then go to Step (2). Else go to Step(2). Where X_i, X_j, E_i, E_j are the SA current solution, the individual GA population members and their corresponding cost respectively.

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

7. NUMERICAL EXAMPLES

In order to test the proposed hybrid algorithm (GTS), three examples from the literature, solved by Lagrangian Relaxation (LR), Integer Programming (IP) and Expert Systems respectively [5,6,7], are considered. Examples 1&2 include 10 generating units while Example 3 contains 26 units all with a scheduling time horizon of 24 hours.

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

Different runs were carried out to evaluate the results obtained by the proposed algorithm (GTS) and those obtained from the individual algorithms in [9,10,11,12]. Table (1) shows the results of this comparison for the three examples. The superiority of the GTS is obvious. It is clear that the GTS algorithm performs better than the individual algorithms, in terms of both solution quality and number of generations.

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

Tables (3)-(5) show detailed results for Example 1, [5]. Table (3) shows the load sharing among the committed units in the 24 hours. Table (4) presents the final schedule of the 24 hour period, given in Table (3), in the form of its equivalent decimal numbers. Table (5) gives the hourly load demand, and the corresponding committed capacities, economic dispatch costs, start-up costs, and total operating cost.

Table (1) Comparison Between SA, TS, &GA and GTS

	Example	SA[9]	TS[10]	GA[11]	GTS
Total Cost ($)	1	536622	538390	537686	535271
	2	59385	59512	59385	59385
	3	662664	662583	662323	662113
Generations No.	1			1984	181
	2			1839	430
	3			2836	762

Table (2) Comparison Between LR & IP and GTS

	Example	LR[5]	IP[6]	GTS
Total Cost ($)	1	540895	-	535271
	2	-	60667	59385
% Saving	1			1.05
	2			2.15

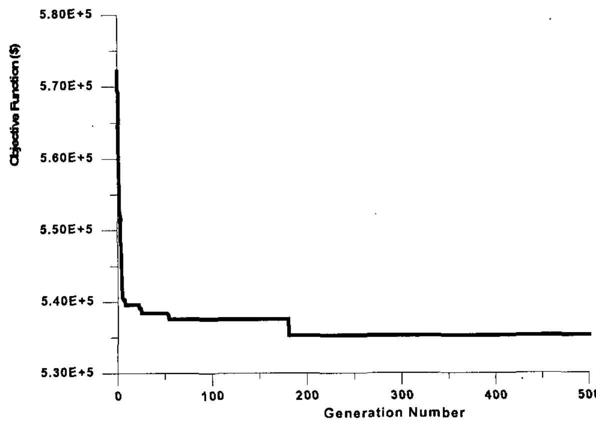


Fig. (5) Convergence of the GTS Algorithm (Example 1).

Table(3) Power Sharing (MW) of Example 1

HR	Unit Number**							
	2	3	4	6	7	8	9	10
1	400	0	0	185.04	0	350.26	0	89.7
2	395.36	0	0	181.09	0	338.36	0	85.19
3	355.38	0	0	168.67	0	300.95	0	75
4	333.13	0	0	161.75	0	280.12	0	75
5	400	0	0	185.04	0	350.26	0	89.7
6	400	0	295.68	200	0	375	0	129.32
7	400	383.56	420	200	0	375	0	191.44
8	400	295.59	396.65	200	0	375	569.93	162.83
9	400	468.07	420	200	0	375	768.01	218.92
10	400	444.6	420	200	358.0	375	741.06	211.29
11	400	486.3	420	200	404.8	375	788.95	224.86
12	400	514.11	420	200	436.0	375	820.89	233.91
13	400	479.35	420	200	397.0	375	780.96	222.6
14	400	388.98	420	200	295.6	375	677.18	193.2
15	400	310.07	410.84	200	250	375	586.56	167.54
16	400	266.64	368.27	200	250	375	536.68	153.41
17	400	317.31	417.93	200	250	375	594.87	169.89
18	400	458.51	420	200	373.6	375	757.03	215.81
19	400	486.3	420	200	404.8	375	788.95	224.86
20	400	491.88	0	200	411.1	375	795.35	226.67
21	400	344.77	0	200	0	375	626.41	178.82
22	400	459.02	0	200	0	375	0	215.98
23	400	194.91	0	200	0	375	0	130.09
24	389.59	165	0	179.3	0	332.96	0	83.15

**Units 1,5 are OFF all hours.

Table (4) The Last Population in the GA for Example 1

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

Table (5) Load, Capacities(MW), and Hourly Costs($) of Example 1

HR	Load	Cap.	ED-Cost	ST-Cost	T-Cost
1	1,025	1,225	9,670.0	-	9,670.0
2	1,000	1,225	9,446.6	-	9,446.6
3	900	1,225	8,560.9	-	8,560.9
4	850	1,225	8,123.1	-	8,123.1
5	1,025	1,225	9,670.0	-	9,670.0
6	1,400	1,645	13,434.1	1,056.0	14,490.0
7	1,970	2,245	19,385.1	1,631.4	21,016.5
8	2,400	3,095	23,815.5	1,817.7	25,633.2
9	2,850	3,095	28,253.9	-	28,253.9
10	3,150	3,845	31,701.7	2,057.6	33,759.3
11	3,300	3,845	33,219.8	-	33,219.8
12	3,400	3,845	34,242.1	-	34,242.1
13	3,275	3,845	32,965.5	-	32,965.5
14	2,950	3,845	29,706.3	-	29,706.3
15	2,700	3,845	27,259.7	-	27,259.7
16	2,550	3,845	25,819.8	-	25,819.8
17	2,725	3,845	27,501.6	-	27,501.6
18	3,200	3,845	32,205.7	-	32,205.7
19	3,300	3,845	33,219.8	-	33,219.8
20	2,900	3,425	29,198.0	-	29,198.0
21	2,125	2,675	20,994.5	-	20,994.5
22	1,650	1,825	16,158.6	-	16,158.6
23	1,300	1,825	12,758.9	-	12,758.9
24	1,150	1,825	11,397.1	-	11,397.1

Total operating cost = $535270.94

8. CONCLUSIONS

In this paper we proposed a new hybrid algorithm for the UCP. The algorithm integrates the main features of the most commonly used artificial intelligence method for solving combinatorial optimization problems, GA, TS, and SA. The algorithm is based mainly on the GA, while the TS method is used to generate new members in the GA population. The SA algorithm is used to accelerate the convergence of the GA by testing all the GA members after each reproduction of a new population.

The implementation of the GA in the proposed algorithm differs from other GA implementations in three respects [11,12]. First, the UCP solution is coded using a mix between binary and decimal representations, thus saving computer memory as well as computation time of the GA search procedure. Second, the fitness function is based only on the total operating cost and no penalties are included. Third, to improve the fine local tuning capabilities of the proposed GA a special mutation operator is designed based on a local search procedure [11]. TS implementation is based on the short term memory procedures [10]. In the SA part a simple cooling schedule is used to simplify and speed up the calculations [9].

Three examples from the literature were solved for comparison with other methods. The obtained results are superior to those reported in [5,6] using Lagrangian Relaxation and Integer Programming. Moreover the obtained results (using the proposed algorithm) are better than those obtained using the individual SA, TS or GA in [9,10,11].

A basic advantage of the proposed algorithm is the high speed of convergence besides the high quality of solutions compared to those obtained by GA, SA and TS methods. Further work in this area may be in the application of parallel processing techniques, thus reducing the computation time or exploring wider solution space.

ACKNOWLEDGMENT

The authors acknowledge the support of King Fahd University of Petroleum and Minerals.

9. REFERENCES

[1] Allen J. Wood and B. F. Wollenberg, Power Generation, Operation, and Control, John Wiley & Sons Ltd. 1984.

[2] G. B. Sheble', G. N. Fahd, " Unit Commitment Literature Synopsis ", *IEEE Trans. on Power Systems*, Vol. 9, No. 1, 1994, pp. 128-135.

[3] C. K. Pang, G. B. Sheble', F. Albuyeh, " Evaluation of Dynamic Programming Based Methods and Multiple Area Representation for Thermal Unit Commitments ", *IEEE Trans. on PAS*, Vol. PAS-100, No. 3, 1981, pp. 1212-1218.

[4] A. K. Ayoub, A. D. Patton, "Optimal Thermal Generating Unit Commitment ", *IEEE Trans. on PAS*, Vol. PAS-90, No. 4, 1971, pp. 1752-1756.

[5] J. F. Bard, " Short-Term Scheduling of Thermal-Electric Generators Using Lagrangian Relaxation", *Operation Research*, Vol. 36, No. 5, 1988, pp. 756-766.

[6] A. Turgeon, "Optimal Scheduling of Thermal Generating Units ", *IEEE Trans. on Automatic Control*, Vol. AC-23, No. 6, 1978. pp. 1000-1005.

[7] S. K. Tong, S. M. Shahidepour, Z. Ouyang, "A Heuristic Short-Term Unit Commitment ", *IEEE Trans. on Power Systems*, Vol. 6, No. 3, 1991, pp. 1210-1216.

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

[9] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, " A Simulated Annealing Algorithm for Unit Commitment ", *IEEE Trans. on Power Systems, Vol. 13, No. 1, February 1998, pp. 197-204.*

[10] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "Unit Commitment by Tabu Search ", *IEE proceedings -Generation, Transmission and Distribution*, Vol. 145, No. 1, January 1998, pp. 56-64

[11] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A Genetic Algorithm with Local Search for Unit Commitment " *Proceeding of the intelligent System*

Application to Power Systems (ISAP'97) July 6-10, 1997, Korea, pp. 170-175.

[12]A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A New Genetic Algorithm Approach for Unit Commitment " *Proceeding of the Genetic Algorithms in Engineering systems: Innovations and Applications (GALSIA'97) July 2-4, 1997, Conference Publication No. 446, pp. 215-220.*

[13]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 System Research* Vol. 34, 1995, pp. 29-36.

[14]S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, " A Genetic Algorithm Solution to the Unit Commitment Problem", *IEEE Trans. on Power Systems*, Vol. 11, No. 1, Feb. 1996, pp. 83-92.

[15]S. O. Orero and M. R. Irving, " A Genetic Algorithm for Generator Scheduling in Power Systems", *Electrical Power & Energy Systems* Vol. 18, No. 1, 1996, pp. 19-26.

[16]A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A New Simulated Annealing-Based Tabu search Algorithm for Unit Commitment " *Proceeding of the IEEE International Conference on Systems, Man and cybernetics, SMC'97, October 12-15, 1997, Orlando, Florida.*

[17]A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "A New Genetic-Based Tabu Search Algorithm for Unit Commitment " *Accepted for publication in the International Journal of Electric Power Systems Research.*

[18]F. Żhuang , F. D. Galiana, " Unit Commitment by Simulated Annealing ", *IEEE Trans. on Power Systems*, Vol. 5, No. 1, 1990, pp. 311-318.

[19]D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, Mass., *Addison Wesely*, 1989.

[20]L. Davis (ed), Handbook of Genetic Algorithms, *Van Nostrand*, N. York, 1991.

[21]John J. Grefenstette, " Optimization of Control Parameters for Genetic Algorithms", *IEEE Trans. On Systems, Man, and Cybernetics*, 16, No. 1, 1986, PP. 122-128.

[22]B. Awadh, N. Sepehri, and O. Hawaleshka, " A Computer-Aided Process Planning Model Based on Genetic Algorithms", *Computer Ops. Res.*, Vol. 22, No. 8, 1995, pp. 841-856.

[23]J. A. Bland and G. P. Dawson, "Tabu Search and Design Optimization", Vol. 23, No. 3, pp. 195-201, April 1991.

[24]F. Glover, " A User's Guide to Tabu Search ", *Annals of Oper. Reas.* 41(1993) 3-28

[25]F. Glover, "Artificial Intelligence, Heuristic Frameworks and Tabu Search", *Managerial and Decision Economics*, Vol. 11, 365-375 (1990).

[26]F. Glover , H. J. Greenberg, " New Approach for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", *European Journal of Operational Research* 39 (1989) 119-130.

[27]F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence", *Comput. & Ops. Res.* Vol. 13, No. 5, pp. 533-549, 1986.

[28]F. Glover, "Tabu Search-Part I", *ORSA Journal on Computing*, Vol. 1., No. 3, pp. 190-206, Summer 1989.

[29]E. Aarts and Jan Korst, " Simulated Annealing and Boltzman Machines", A Stochastic Approach to Combinatorial Optimization and Neural Computing. John Wiley & Sons Ltd. 1989.

[30]V. Cerny, " Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm". *Journal of Optimization Theory and Applications*, Vol. 45, No. 1, 1985.

[31]Shokri Z. Selim and K. Alsultan, " A Simulated Annealing Algorithm for The Clustering Problem ", *Pattern Recognition*, Vol. 24, No. 10, 1991, pp. 1003-1008.

[32]N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, " Equations of state calculations by fast computing machines", *J. Chem. Phys.* 21, 1953, pp. 1087-1982.

APPENDIX A: THE GA APPROACH

A.1 Overview

Genetic Algorithms are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings [19-22].

Generally, GA comprises three different phases of search:
Phase 1: Creating an initial population.
Phase 2: Evaluating a fitness function.
Phase 3: Producing a new population.

A.2 The GA Operators

There are usually three operators in a typical genetic algorithm [21]. The first is the production operator (elitism) which makes one or more copies of any individual that posses a high fitness value; otherwise, the individual is eliminated from the solution pool.

The second operator is the recombination (also known as the "crossover") operator. This operator selects two individuals within the generation and a crossover site and performs a swapping operation of the string bits to the right hand side of the crossover site of both individuals. Crossover operations synthesize bits of knowledge gained from both parents exhibiting better than average performance. Thus, the probability of a better performing offspring's is greatly enhanced.

The third operator is the "mutation" operator. This operator acts as a background operator and is used to explore some of the invested 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.

APPENDIX B: THE TS METHOD

B.1 Overview

Tabu Search is characterized by an ability to escape local optima by using a short-term memory of recent solutions. This is achieved by a strategy of forbidding certain moves. The purpose of classifying certain move as forbidden - i.e. "tabu"- is basically to prevent cycling. Moreover, TS permits backtracking to previous solutions, which may ultimately lead, via a different direction, to better solutions [24].

The main two components of TS algorithm are the Tabu List (TL) restrictions and the Aspiration Level (AV) of the solution associated with the recorded moves. Discussion of these two terms is presented in the following sections.

B.2 Tabu List (TL)

TL is managed by recording moves (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 [23]. Best sizes of TL lie in an intermediate range between these extremes. In some applications a simple choice of TL size in a range centered around 7 seems to be quite effective [25].

B.3 Aspiration Criteria (AV)

Another key issue of TS arises when the move under consideration has been found to be tabu. Associated with each entry in the tabu list there is a certain value for the evaluation function called Aspiration Level (AV). Roughly speaking, AV criteria are designed to override tabu status if a move is "good enough" [25].

APPENDIX C: THE SA ALGORITHM

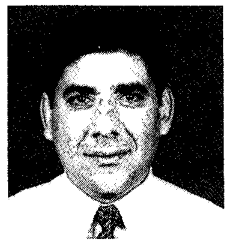
C.1 Concepts of the SA

Annealing [29-32], physically, refers to the process of heating up a solid to a high temperature followed by slow cooling achieved by decreasing the temperature of the environment in steps. At each step the temperature is maintained constant for a period of time sufficient for the solid to reach thermal equilibrium.

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

C.2 Cooling Schedule

A finite-time implementation of the SA algorithm can be realized by generating homogenous Markov chains of finite length for a finite sequence of descending values of the control parameter. To achieve this, one must specify a set of parameters that governs the convergence of the algorithm. These parameters form a cooling schedule. The parameters of the cooling schedules are: an initial value of the control parameter decrement function for decreasing the control parameter and a final value of the control parameter specified by the stopping criterion, and a finite length of each homogenous Markov chain.



Abdel-aal H. Mantawy was born in Cairo, Egypt. He received his B.Sc. and M.Sc. degrees from Ain Shams University, Egypt and the Ph.D. degree from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1982, 1988, 1997 respectively, all in Electrical Engineering. From 1982 to 1992 he was a graduate assistant then, a lecturer at Ain Shams University. His is currently working as a postdoctoral fellow at Electrical Engineering Department, King Fahd University of Petroleum and Minerals.
Dr. Mantawy's research interests includes the application of Artificial Intelligence and optimization techniques to power systems operation and planning.

Youssef L. Abdel-Magid (M'74, SM'87) received the B.Sc. (Honors) degree from Cairo University, Egypt, and the M.Sc and Ph.D. degrees from the University of Manitoba, Canada, in 1969,1972, and 1976, respectively, all in Electrical Engineering. From 1976 to 1979, he was with Manitoba Hydro as a telecontrol engineer. In 1979, he joined King Fahd University of Petroleum and Minerals , where he is currently an Associate Professor. During the 1990-1991 academic year, he was a visiting scholar at Stanford University, USA.
Dr. Abdel-Magid's research interests include power system control and modeling, optimization, adaptive control, and application of Intelligent systems to power systems.

Shokri Z. Selim received the B.Sc. degree in Mechanical Engineering, the M.Sc degree in Industrial Engineering from Cairo University, Egypt, and the Ph.D. degree in operations research from Georgia Institute of Technology, USA, in 1970,1973,1979, respectively. He is an Associate Professor in the Department of Systems Engineering at King Fahd University of Petroleum and Minerals. Dr. Selim's research interests are in the areas of cluster analysis, simulation of large systems, nonconvex optimization and scheduling.