A NEW SIMULATED ANNEALING-BASED TABU SEARCH ALGORITHM FOR UNIT COMMITMENT

A. H. Mantawy, student 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

In this paper we propose a new hybrid algorithm for solving the Unit Commitment Problem (UCP). The algorithm integrates the use of Simulated Annealing (SA) and Tabu Search (TS) for solving the UCP. The algorithm includes a step to find an initial control parameter, at which virtually all trial solutions are accepted. It uses a polynomial-time cooling schedule that is advocated in the SA literature. Furthermore the short-term memory procedures of the TS method are embedded in the SA test to prevent cycling of accepted solutions. Numerical examples from the literature are solved. Results showed an improvement in the solutions costs compared to previously obtained results.

1. INTRODUCTION

Unit commitment is the problem of optimal scheduling of the generating units during a specified time horizon. The committed units must meet the system load and reserve requirements at minimum operating cost, in addition 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].

The Unit Commitment Problem (UCP) can be considered as two linked optimization problems, the first is a combinatorial problem and the second is a nonlinear programming problem. The exact solution of the UCP can only be obtained by a complete enumeration of all feasible combinations of generating units, which could be a very huge number. Then, the economic dispatch problem is solved for each feasible combination. Basically, the high dimension of the possible solution space is the real difficulty in solving the problem.

The solution methods being used to solve the UCP can be divided into four categories[1-13]:

- Classical optimization methods such as: dynamic programming, integer programming, branch and bound and Lagrangian relaxation.
- Heuristic methods such as priority list.
- Artificial Intelligence methods such as: neural networks, expert systems, genetic algorithms, tabu search, and simulated annealing.
- Hybrid algorithms.

Classical optimization methods are well documented in the literature as a direct means for solving this problem. Some of these methods give good results, like Lagrangian relaxation, while others face the problem of dimensionality, especially in case of large-scale systems, as in dynamic programming.

In heuristic methods, a suboptimal solution may be obtained due to the incomplete search of the solution space besides the lack of a mathematical proof for reaching the optimal solution.

Artificial Intelligence methods seem to be promising and are still evolving. Genetic algorithms and neural networks are inspired by principles derived from biological processes, while simulated annealing is derived from material science. 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[14].

Hybrid algorithms are also well known techniques for solving optimization problems. They try to make use of the merits of different methods. Hence, the aim of hybridization is to improve the performance of algorithms that are based on a single method.

Simulated Annealing (SA), is a powerful technique for solving combinatorial optimization problems. It has been theoretically proved to converge to the optimum solution. SAA has also many strong features [15-18]. The most important features are; finding a high quality solution that does not strongly depend on the initial solution and it does not need large computer memory.

Tabu Search (TS) is an efficient optimization procedure that has been successfully applied to a number of combinatorial optimization problems[19-25]. It has the ability to avoid entrapment in local minima. TS employs a flexible memory system (in contrast to 'memoryless' systems, as SA and GA, and rigid memory systems as in branch-and-bound). Specific attention is given to the short-term memory component of TS, which has provided solutions superior to the best obtained by other methods for a variety of problems[19].

0-7803-4053-1/97/$10.00

In previous works for the authors to solve the UCP, a SA algorithm (SAA) and a TS algorithm (TSA) have been implemented [11-13].

The proposed hybrid algorithm is based mainly on the SAA while TS is used to prevent the repeated solutions from being accepted by the SA test.

In the next section, the formulation of the UCP is presented. Sections 3 and 4 give a general explanation of the SA and the TS methods. Section 5 presents the detailed description of the proposed algorithm. In Section 6 the computational results along with a comparison with previously published work are presented. Section 7 outlines the conclusions.

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 [1-13].

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:

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

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:

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

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]:

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

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:

2.2.1 System Constraints:

1- Load demand constraints:

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

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

2- 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.

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

2.2.2 Unit Constraints:

1- Generation limits

$$U_{it}Pmin_i \leq P_{it} \leq Pmax_i U_{it}\ \forall i,t \tag{6}$$

Wh\ere, $Pmin_i, Pmax_i$ is minimum and maximum generation limit (MW) of unit i.

2- Minimum up/down time

$$\begin{aligned} Toff_i &\geq Tdown_i \\ Ton_i &\geq Tup_i \end{aligned}\ ;\forall i \tag{7}$$

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.

3-Unit initial status

4-Crew constraints

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

6-Unit derating

3. SIMULATED ANNEALING ALGORITHM

3.1 Concepts of Simulated Annealing:

Annealing physically [15,17,18], 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.

3.2 Application of Simulated Annealing to Combinatorial Optimization Problems:

In applying the SAA, to solve the combinatorial optimization problem, the basic idea is to choose a feasible solution at random and then get a neighbor to this solution. A move to this neighbor is performed if either it has a 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. The effect of decreasing Cp is that the probability of accepting an increase in the objective function value is decreased during the search [16,17].

4. TABU SEARCH METHOD

4.1 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[23].

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

4.2 Tabu List

Tabu List (TL) is managed by recording moves (trial solutions) in the order in which they are made. These recorded solutions are considered tabu for certain number of iterations equal to the TL size[22,24].

4.3 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"[24].

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

5. THE PROPOSED ALGORITHM FOR THE UNIT COMMITMENT PROBLEM

5-1 Main Idea of the Algorithm:

The new proposed algorithm (we call it ST algorithm) integrates the main features of the SA and TS algorithms to solve the combinatorial optimization part of the UCP and a quadratic programming routine solves the nonlinear part.

The main feature of the TS method is to prevent cycling of solutions by using short term memory. This could be explored in refining the SAA, which is a memoryless technique. The main idea in the proposed algorithm is to use the TS algorithm to prevent the repeated solutions from being accepted by the SAA. This will save CPU time and improve the quality of the solution obtained. This is achieved by applying the TS test for all trial solutions. The accepted solution from the tabu test is then tested by the SAA.

In brief the proposed ST algorithm may be described as a SAA with the TS algorithm used as a filter to prevent cycling of accepted solutions.

The main steps of the ST algorithm are described in the flow chart of fig.(1).

5.2 Generating Trial Solution (Neighbor):

The neighbors should be randomly generated (block 4, fig(1)), 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.

In this work we use our rules to obtain randomly feasible solutions. These rules are described in details in [11].

5.3 Operating Cost Calculation:

Once a trial solution is obtained, the corresponding total operating cost is determined (block 4). 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 using (2).

5.4 Tabu Search Part in the Algorithm:

In the TS part of the ST algorithm the short-term memory procedures are implemented. In this work we use a new approach for implementing the TL for the UCP. In this approach the solution vector for each generating unit (which is 0-1 values) is recorded as its equivalent decimal number. Hence the Generating Unit Tabu List (GUTL) is a one dimensional array of length Z. Each entry records the equivalent decimal number of a specific trial solution for that unit. By using this approach we record all information of the trial solution in minimum memory.

The implementation of the TSA (block 5) can be described as follows:

- Sort the set of trial solutions (neighbors) in an ascending order according to their objective functions.
- Apply the acceptance test in order until one of these solutions is accepted.
- <u>Tabu test</u>: Accept the trial solution if:
  - The trial solution is not in the TL, or
  - In the TL but satisfies the AV.

Otherwise, apply the test to the next solution.

5.5 Simulated Annealing Part in the Algorithm:

In the SA part of the ST algorithm the polynomial-time cooling schedule is implemented (described in the appendix) to decrement the control parameter during the search (block 9).

The SA test (block 6) is described as follows:

- Accept the trail solution:
  if: $F_j < F_i$ or
  $[\exp(F_i - F_j)/C_p] \geq U(0,1)$,
  where F_i, F_j are the objective function of the current and trial solutions respectively
- Otherwise the trial solution is rejected.

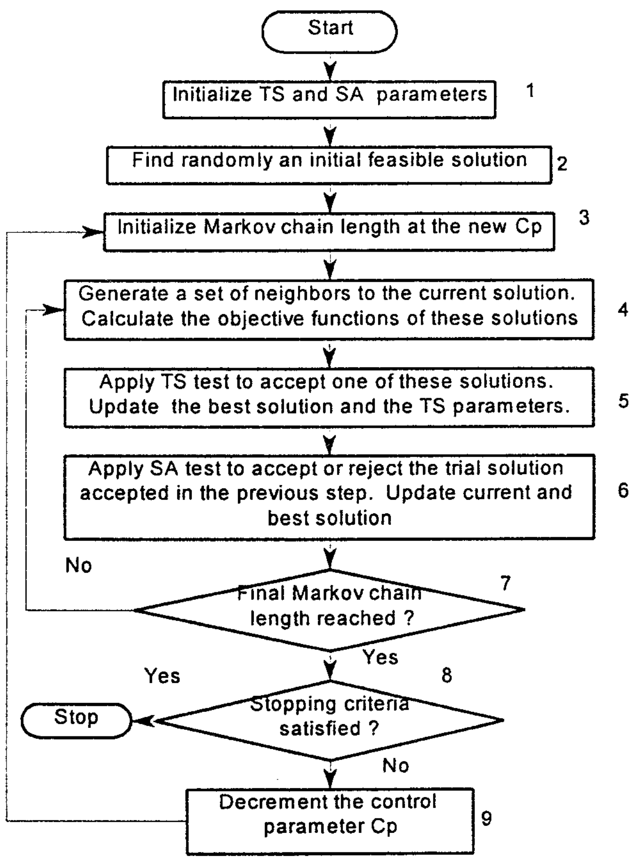

Fig.(1) The proposed SATS algorithm for UCP

5.6 Stopping Criteria:

There may be several possible stopping conditions for the search. In our implementation we stop the search if one of two conditions are satisfied respectively;

-The best solution is not changing for a prespecified number of iterations.

-Maximum allowable number of iterations is reached.

6. NUMERICAL EXAMPLES

Considering the proposed ST algorithm for solving the UCP, a computer program has been implemented and tested.

In order to test the proposed algorithm three different examples are solved. The first two examples include 10 generating units while the third contains 26 units. The scheduling time horizon for all cases is 24 hours. Example 1,[5], was solved by Lagrangian Relaxation(LR), Example 2,[6], was solved by Integer Programming(IP) while Example 3,[7,8], was solved by Expert Systems.

To emphasis the effectiveness of hybridizing the SA and the TS algorithms, a comparison is made with the results of those algorithms. The three examples are solved by the SA algorithm (SAA) and the TS algorithm (TSA) reported by the authors in [11,13].

The following control parameters have been chosen after running a number of simulations: maximum number of iterations=3000, tabu size=7, initial acceptance ratio during the heating process=0.9, chain length=150, ε =0.00001, and δ =0.3.

Table (6.1) shows a comparison of the results obtained for the three examples 1,2, and 3 as solved by the SAA, the TSA and the ST algorithms. It is obvious that the ST algorithm achieves reduction in the operating costs for the three examples. Also, the number of iterations is less.

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

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

Table (6.1) Comparison with the SAA and the TSA

	Example	SAA[11]	TSA[13]	ST
Total Cost ($)	1	536622	538390	536386
,,	2	59512	59512	59385
,,	3	662664	662583	660596
Iterations No.	1	384	1924	625
,,	2	652	616	538
,,	3	2361	3900	2829

Table (6.2) Comparison with LR and IP methods

	Example	LR [5]	IP [6]	ST
Total Cost ($)	1	540895	-	536386
,,	2	-	60667	59380
% Saving	1	0	-	0.83
,,	2	-	0	2.11

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

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

** Units 1 and 5 are OFF all hours

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

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

Total operating cost = $536386.

7. CONCLUSIONS

In this paper we proposed a new hybrid algorithm for the UCP. The algorithm integrates the main features of two of the most efficient methods for solving combinatorial optimization problems, simulated annealing and tabu search. The algorithm is based mainly on the SA, while the TS method is used to reject the repeated solutions before being tested by the SA method.

The current SAA implementation [11] is based on a polynomial time cooling schedule which uses statistics calculation during the search. It also provides a methodology for determining an initial temperature value, by simulating the heating step of the annealing process, at which a prespecified initial acceptance ratio is achieved. The TS part of the algorithm is implemented using the short term memory procedures[13].

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 ST algorithm) are better than those obtained using the individual SA and TS in [11,13].

A basic advantage of the proposed algorithm is the high speed of convergence besides the high quality of solutions compared to those obtained by 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.

8. REFERENCES

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

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

[3] C. K. Pang and 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 and 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", *Operations 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 and Z. Ouyang, " A Heuristic Short-Term Unit Commitment ", *IEEE Trans. on Power Systems*, Vol. 6, No. 3, 1991, pp. 1210-1216.

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

[9] F. Zhuang and F. D. Galiana, " Unit Commitment by Simulated Annealing ", *IEEE Trans. on Power Systems*, Vol. 5, No. 1, 1990, pp. 311-318.
[10] A. H. Mantawy, " Optimal Scheduling of Thermal Generation in Electric Power System", A Master Thesis, Ain Shams University, Cairo, Egypt, 1988.
[11] A. H .Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, " A Simulated Annealing Algorithm for Unit Commitment ", Accepted for publication in the *IEEE Trans. on Power Systems*.
[12] A. H. Mantawy, Youssef L. Abdel-Magid, Shokri Z. Selim, and Mazen A. Salah, " An Improved Simulated Annealing Algorithm for Unit Commitment-Application to Sceco-East", *3rd International Conference on Intelligent Applications in Communications and Power Systems, IACPS'97, pp. 133-139, April, 1997, UAE.*
[13] A. H. Mantawy, Youssef L. Abdel-Magid, and Shokri Z. Selim, "Unit Commitment by Tabu Search ", revised and resubmitted to the *IEE proceedings -Generation, Transmission and Distribution*, January 1997.
[14] F. Glover and H. J. Greenberg, " New Approach for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", *European Journal of Oper. Res.* 39, 1989, pp. 119-130.
[15] E. Aarts and Jan Korst, " Simulated Annealing and Boltzman Machines", A Stochastic Approach to Combinatorial Optimization and Neural Computing. John Wiley & Sons Ltd. 1989.
[16] V. Cerny, " Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm". *Journal of Optimization Theory and Applications*, Vol. 45, No. 1, 1985.
[17] Shokri Z. Selim and K. Alsultan, " A Simulated Annealing Algorithm for The Clustering Problem ", *Pattern Recognition*, Vol. 24, No. 10, 1991, pp. 1003-1008.
[18] 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.
[19] J. A. Bland and G. P. Dawson, "Tabu Search and Design Optimization", Vol. 23, No. 3, pp. 195-201, April 1991.
[20] F. Glover, " A User's Guide to Tabu Search ", *Annals of Oper. Res.* 41(1993)3-28
[21] F. Glover, "Artificial Intelligence, Heuristic Frameworks and Tabu Search", *Managerial and Decision Economics*, Vol. 11, 365-375 (1990).
[22] F. Glover and H. J. Greenberg, " New Approach for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", *European Journal of Operational Research* 39 (1989) 119-130.
[23] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computer. & Oper. Res.* Vol. 13, No. 5, pp. 533-549, 1986.
[24] F. Glover, "Tabu Search-Part I", *ORSA Journal on Computing*, Vol. 1., No. 3, pp. 190-206, Summer 1989.
[25] F. Glover, "Tabu Search-Part II", *ORSA Journal on Computing*, Vol. 2., No. 1, pp. 4-32, Winter 1990.

9. APPENDIX

THE POLYNOMIAL-TIME COOLING SCHEDULE

A finite-time implementation of the SAA 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, a cooling schedule should be designed to govern the convergence of the algorithm.

In this work, a Polynomial-Time schedule is implemented. This cooling schedule leads to a Polynomial-Time execution of the SAA, but it can not give any guarantee for the deviation in cost between the final solution obtained by the algorithm and the optimal cost. In the following we describe these parameters [15].

9.1 Initial Value of the Control Parameter Cp

The initial value of Cp, is obtained from the requirement that initially virtually all proposed trial solutions should be accepted.

It can be shown that the Cp is calculated by[15]:

$$Cp = \overset{(+)}{\Delta f} / \ln(m_2 \,/\, (m_2 . X - m_1(1 - X)) \qquad (8)$$

where: m_1, m_2 denote the number of trials that having less and excess objective function values than that of the current solution respectively. $\overset{(+)}{\Delta f}$: is the average difference in cost over the m_2 trials. X: is the acceptance ratio.

9.2 Decrement of the Control Parameter

Cp^{k+1} at iteration k+1 is related to the current value, Cp^k, by the following function[15]:

$$Cp^{k+1} = Cp^k \,/\, (1 + (Cp^k . \ln(1+\delta) \,/\, 3\sigma Cp^k \qquad (9)$$

where: σ is the standard deviation of the cost values generated at the $k\underline{th}$ Markov chain, corresponding to Cp^k. δ is a constant called distance parameter. Small δ-values lead to small decrements in Cp. Typical values of δ are between 0.1 and 0.5.

9.3 The Final value of the control parameter

Termination in the Polynomial-Time cooling schedule is based on an extrapolation of the expected average cost at the final value of control parameter. Hence the algorithm is terminated if for some value of k we have [15] :

$$\left.\frac{Cp^k}{\langle f\rangle_\infty} \cdot \frac{\partial \langle f\rangle_{Cp}}{\partial Cp}\right|_{Cp = Cp_k} < \varepsilon \qquad (10)$$

where, $\langle f\rangle_\infty \approx \langle f\rangle_{Cp_0}$ is the average cost at initial value of control parameter Cp_0. $\langle f\rangle_{Cp}$ is the average cost at $k\underline{th}$ Markov chain. $\left.\frac{\partial \langle f\rangle_{Cp}}{\partial Cp}\right|_{Cp=Cp_k}$ is the rate of change in the average cost at Cp^k. ε is some small positive number. In our implementation ε =0.00001.

9.4 The Length of Markov Chains:

In [15], it is concluded that the decrement function of the control parameter, (9), requires only a 'small' number of trial solution to rapidly approach the stationary distribution for a given next value of the control parameter. In general, a chain length of more than 100 transitions is reasonable[15]. In our implementation good results have been reached at a chain length of 150.