GLOBAL JOURNAL OF PURE AND APPLIED SCIENCES VOL. 10, NO. 4, SEPTEMBER 2004: 607 - 610
 ISSN 1118-0579

A NOTE ON SIMULATED ANNEALING TO COMPUTER LABORATORY SCHEDULING.

E. O. OMIDIORA and I. A. ADETUNDE

(Received 5 July, 2001; Revision Accepted 21 July, 2004)

ABSTRACT

The concepts, principles and implementation of simulated Annealing as a modem heuristic technique is presented. Simulated Annealing algorithm is used in solving real life problem of Computer Laboratory scheduling in order to maximize the use of scarce and insufficient resources.

KEYWORDS: Simulated Annealing (SA), Computer Laboratory Scheduling, Statistical Thermodynamics, Energy Function, and Heuristic etc.

INTRODUCTION

The Simulated Annealing theory is recently of much interest (Dowsland, 1993; Kirkpatrick et al, 1983). Simulated Annealing is a variant of the well-known modern heuristic strategy for finding good solutions to a wide variety of combinatorial optimization problems. It was first suggested as an optimization technique. The technique derives its name from its origin in statistical thermodynamics, in its analogy to the process of slowly cooling metals to improve strength, otherwise known as heat annealing where it forms the basis of a simulation of the energy function of material being cooled in a heat bath.

The advent of simulated annealing was as a result of a considerable amount of inadequacies observed in traditional local search techniques. It was discovered that most of these early techniques are problem specific. The design and implementation of such technique can be long and expensive process, and the result is often domain dependent and not flexible enough to deal with changes, which may occur over time. Consequently, there is considerable interest in general techniques, which can be applied to a variety of different combinatorial problems. A significant amount of research was then made into a new generation of heuristic techniques such as Genetic Algorithms, Tabu search and Simulated Annealing (SA).

Of these, Simulated Annealing had been tested on an increasing number of real life problems, which have been tackled successfully. Some of the earliest practical application of Simulated Annealing is the Very Large Scale Integration [VLSI], Design packages of thunderbird and thunderwolf. Recently, (Kelly et al, 1990) applied Simulated Annealing (SA) to the controlled rounding problem in 3 dimensions.

In this paper, we use Simulated Annealing approach to Computer Laboratory Scheduling. Computer Laboratory Scheduling involves assigning students to practical classes in a Laboratory with few machines to work on and a limited number of hours within a week. Each student is required to attend at least a practical class once in a week and at least a supervisor must be present in each of the classes. The large variation in the degree schemes taken by the students and the presence of options within these schemes means that there is a very little commonality among the lecture timetables of the student group. Thus, the two major constraints are the diversities in students' free periods and the capacity of the Laboratory. The practical classes for each student have to be assigned using a schedule subject to the given constraints.

In achieving this, Simulated Annealing, which is heuristic approach, is adopted to carry out the timetabling. The approach solves this problem by taking the free periods of each class of students and then fixed practical classes to the students based on the available slots. The approach involve moving or canceling an event or swapping the times of two events therefore adopting a search solution technique to allow all the students participate in the practical sessions within the time limit.

This paper focuses on the following:-

(a) To provide an understanding of the concepts, principles, flexibility and implementation of Simulated Annealing Algorithms in solving real life problems.

(b) To show the importance of scheduling most especially in maximizing the use of scarce and insufficient resources.

(c) To solve the problem of computer Laboratory using SA.

SIMULATED ANNEALING

Simulated Annealing is a variant of the well-known heuristic strategy of local search. SA is a mechanism for searching the feasible search space of an optimization problem (Kirkpatrick et al, 1983). Their algorithms were based on the controlled cooling of a material, by a process known as annealing. The Simulated Annealing technique is essentially local search in which a move to an inferior solution is allowed with the probability, which decreases as the process

E O OMIDIORA, Department of Computer Science and Engineering, Ladoke Akintola, University of Technology, Ogbomoso, Oyo State - Nigeria

I. A. ADETUNDE, Department of Mathematical Sciences, University of Agriculture, Abeokuta, Ogun State - Nigeria

progresses according to some Boltzman -type distribution.

In traditional local search an initial solution is gradually improved by considering small perturbation or changes. For combinatorial optimization problems in which a solution is defined by a set of zero-one variables, this usually involves changing the value of a single variable or swapping the values of two variables. If such a change results in a better solution the current solution is replaced. According to (Dowsland, 1993), Simulated annealing technique derives its name from its origins in statistical thermodynamics, where it forms the basis of a simulation of the energy function of material being cooled in a heat bath.

The basic method is shown using the algorithm below:

Select an initial solution S_0:

Select an initial temperature $t_0 > 0$;

Select a temperature reduction function α ;

Repeat

Repeat

Randomly select $S \in N(S_0)$:

$\delta = f(S) - f(S_0)$;

if $\delta < 0$ then

$S_0 = S$

else

generate random $x \in U(0,1)$

if $x < \exp(-\delta/t)$ then

$S_0 = S$;

Endif

Endif

Until iteration_count =nrep

$t = \alpha(t)$:

Until stopping condition =true.

S_0 is the approximation to the optimal solution

The parameter, t, is a control parameter which is usually referred to as the temperature, in line with the analogy of the physical cooling process. A solution that results in an increase of δ in the cost function is accepted with the probability $\exp(-\delta/t)$. Thus, if it is allowed to become sufficiently small no further uphill moves will be accepted and the algorithm will converge to a local optimum.

Several theoretical studies show that if t is decreased slowly enough then the process will converge to an optimum solution. However, a cooling rate which is slow enough to satisfy any performance guaranteed will usually require a prohibitive amount of computer time. In spite of this, many empirical studies using faster cooling schedules suggest SA as a very effective heuristic for a variety of combination optimization problems.

DESCRIPTION OF THE SA HEURISTIC

The SA procedure is described in a stepwise procedure by (Roland, 1995):

Step1: Initialization: Choose any starting feasible solution and set it as incumbent solution. The initial temperature is also determined and set as the current temperature while acceptance ratio is taken into consideration. The initial temperature and the acceptance ratio are given a fixed value or are determined by the SA heuristic.

Step 2: Stopping: If out of the set moves. no leads to a feasible neighbor of current solutions or if the solution index maximum. Then stop. The incumbent solution is the approximate optimum.

Step3: Provisional moves: Generate a neighborhood solution by randomly selecting a move from the current solution and compute S as the different in objective function value between the neighborhood and the current solution.

Step 4: Acceptance: If $\delta > 0$ and e is less than random number, which would have been predetermined: then step 3 is repeated else the neighborhood solution becomes the current solution. The neighborhood solution is saved if it is the best solution so far.

Step 5: Temperature Reduction: Decrease the current temperature t by means of the cooling function. If a sufficient number of iterations have passed since the last temperature change. The temperature will be reduced.

Step 6: The solution stored is accepted as the final best solution. In order to implement the algorithm above for a particular problem, a number of decisions must be made. These can be divided into generic and problem specific decisions. The genetic decisions are mainly concerned with controlling the temperature and include the determination of its starting value. The function by which it is decreased. And the condition under which the system will be declared frozen. The problem specific decision involves the definition of the solution space and its neighborhood structure. The form of the cost function and the way in which a starting solution is obtained. Both types of decision need to be made with care.

MODIFICATIONS

Strength = Number of students.
Temperature =Period.
Cost = Number of session.

PROBLEM FORMULATION

All students admitted into Ladoke Akintola University of Technology, Ogbomoso. Oyo State, with the exception of Medical Students are required to take a compulsory computer programming course titled Basic Computer Programming with the code CSE 201 either in their second year or third year as determined by their various Faculties and Departmental authorities. This consists of a weekly lecture, which is supposed to be supported by at least a practical session weekly in a Computer Laboratory but which could not be achieved due to an insufficient number of systems.

COMPUTER LABORATORY SCHEDULING

All students admitted into Ladoke Akintola University of Technology, Ogbomoso, Oyo State, with the exception of Medical Students are required to take a compulsory computer programming course title, Basic Computer Programming, with the course code, CSE 201, either in their second year or third year as determined by their various Faculties and Departmental authorities. This consists of a weekly lecture, which is supposed to be supported by at least a practical session weekly in a computer laboratory but which could not be achieved due to an insufficient number of systems.

Also, the large variation in the degree schemes taken by these students and the presence of option within these schemes mean that there is very little commonality among the lecture timetables of the student group and practical times must therefore be allocated on an individual basis. As each practical excludes other users from the laboratory and requires at least a supervisor to be present and at the same time the working hours of the systems have to be minimized in other to increase the life span of the system. Therefore it is important to minimize the total number of sessions used.

It is assumed that the laboratory, which is available for about thirty-five hours per week, can take thirty students. Details of these hours, which are free for each student in the class, are available and it is required to find allocation of students to a practical session in such away to minimize the number of utilized slots. In addition, there are a number of secondary objectives such as carry over and brought forward courses, many of which are subjective and may vary from year to year.

SIMULATED ANNEALING TO SOLVE COMPUTER LABORATORY SCHEDULING PROBLEMS

As earlier stated, SA can be used to solve various scheduling problems including Computer laboratory scheduling. This is a combinatorial optimization problem in the sense that the total number of practical sessions has to be minimized while the number of students per session is minimized.

In other to implement the SA Algorithm for computer laboratory scheduling problem, a number of decisions must be made. These can be divided into generic and problem specific decisions.

1. The generic decisions are mainly concerned with controlling the time and includes the determination of its starting value, the function by which it is increased, the number of iterations between increases and the conditions under which the scheduling will be declared satisfactorily.
2. The problem specific decisions involve the definition of the solution space (i.e. ideal occupancy of the laboratory) and its neighbor hood structure (i.e. the other feasible number of occupancies), the form of the cost function (i.e. the stipulated number of sessions) and the way in which a starting solution is obtained (this is obtained as a result of the maximum capacity of laboratory).

SIMULATED ANNEALING ALGORITHM FOR COMPUTER LABORATORY SCHEDULING

Initialization:

Select an initial solution S_9
S: = S_9
Assume S = (25...30)
Select an initial period. P
P: = (1.... 35)
P = 1
Select a Period increasing function
P: = P + 1

Stopping condition:

While not Finish (S) do

Provisional move:

Let n: number of iterations
Input n
For I: 1 to n do
Begin
Randomly select S' a neighbour of S
Improvement: = value (S') - value (S);

Acceptance:

If improvement >=0 then S: = S' else
Begin
Generate random number x ∈ U (0,1)
If X<esp (-improvement/period) then S: =S'
end(else)
end(for)

Period increment:

P: = P + 1
end(while).

DISCUSSION OF RESULTS OF SIMULATION

The simulator developed (as in figure 1) for this problem can only work for cases where scheduling is done periodically as provision is not made for any alteration after scheduling has been done. As the size of the laboratory is increasing, there is corresponding increasing tendency that some students may not be scheduled unless there is an increase in the number of systems to be used. In essence, this model points out the need to increase the size of the laboratory (including number of systems) as the number of students increase.

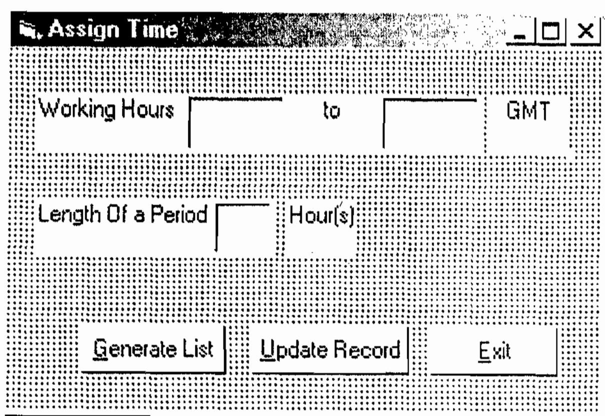

Figure 1: The Simulator that is used to assign students

This is an optimality criterion, if this is not done the laboratory and other materials involved may not be effectively utilized.

It is of paramount importance to know that the students to be rescheduled must specify more than two (2) free periods for the week as to increase his/her chances of being scheduled. Also, a change in a students slot after the scheduling will lead to a change in the optimality of the simulated annealing (SA).

CONCLUSION

This paper shows, that the principles and implementation of Simulated Annealing can be successfully employed for computer laboratory scheduling which is a real time problem. With the use of this Simulated Annealing approached, we are able to maximize the insufficient resources on the laboratory and also maximizing time from the limited time the students learn.

The Simulated Annealing approached as elucidated in the paper can be extended to classical combinational problems such as the travelling salesmen problem, quadratic assignment problem, held of Examination schedule etc. Simulation Annealing is an efficient technique for solving combinatorial optimization problems in general and Computer Laboratory scheduling in particular. Since it was first suggested as a solution for discrete optimization problems in the early 80's, SA has been subjected to a vast amount of testing on a wide variety of problems.

ACKNOWLEDGEMENTS

The authors wish to thank Faniyi B.J. and Aworeni K.O. of Ladoke Akintola University of Technology Ogbomoso for their contribution towards the successful completion of this paper work as well as the staff of Computer Laboratory LAUTECH Ogbomoso.

REFERENCES

Dowsland, K.A., 1993. Simulated Annealing, Canada, C.R. Reeves.

Kirkpatrick, S., Gallet, C. D., Vecchi, M. P., 1983. Optimization by Simulated Annealing, Science, 220: 671-680.

Kelly, J., Golden, B., Assad, A., 1990. Using Simulated Annealing to solve Controlled rounding problems, ORSA Journal on computing, 174-185.

Roland, L.A., 1995. Optimization in Operation Research.