African Journal of Mathematics and Computer Science Research Vol. 5(11), pp. 176-178, September 2012
Available online at http://www.academicjournals.org/AJMCSR
DOI: 10.5897/AJMCSR11.151
ISSN 2006-9731 ©2012 Academic Journals

Review

Comparison of simulated annealing and hill climbing in the course timetabling problem

Kenekayoro Patrick

Department of Mathematics and Computer Science, Niger Delta University, Wilberforce Island, P. M. B. 071, Amassoma, Bayelsa State, Nigeria. E-mail: patrick.kenekayoro@outlook.com.

Accepted 12 June, 2012

Course timetabling is a task that must be performed by all higher institutions. It is very difficult doing this manually and even classified as nondeterministic polynomial (NP) complete in five independent ways. Several methods (heuristics) are used to solve this problem including local search optimization methods like simulated annealing and hill climbing. This paper compares these methods used to solve the university course timetabling problem.

Key words: Hill climbing, simulated annealing, course timetabling, local search optimization.

INTRODUCTION

Designing any type of schedule manually is an arduous task, an example is the university course timetables among others like duty rosters, job shops etcetera. Doing this automatically has proven to be difficult as well. It is classified as a nondeterministic polynomial (NP) hard problem (Even et al., 1976), and proven to be NP complete in five independent ways (Cooper and Kingston, 1995). In NP complete problems, as the size of input increases linearly, the time it takes to find a solution increases exponentially. It is difficult to find a solution to these types of problems in worst case, and a popular way to solve these problems is with heuristics. Course timetabling falls within this group of problems. Course timetabling involves assigning events to time slots and venues while meeting several constraints; constraints could either be hard or soft. A feasible timetable is reached when all hard constraints are met, while a good timetable meets all the hard constraints as well as most of the soft constraints.

Several ways to solve this timetabling problem have been published; some of these methods involve the use of local search optimization techniques like the simulated annealing and hill climbing. Hill climbing and annealing have different variants, depending on the way successive nodes are chosen. This paper compares these methods when used to solve a university course timetabling problem. Subsequent sections describe the course timetabling problem, current techniques used to solve it, with emphasis on simulated annealing and hill climbing, and then results gotten when compared are shown.

COURSE TIMETABLING PROBLEM (CTP)

"Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives" (Wren, 1996). A university course timetable is a function of students, rooms, lecturers, events and time slots. The constraints used to compare results are same as published by Dawood et al. (2011) that a student cannot attend more than one event simultaneously.

The room size must be able to contain all students. At any given time slot, only one event can take place in a room. The timetable to be generated is similar to the ones used in Nigerian universities. It has 5 events a day, hence 25 time slots. The problem instance is similar to the problem in the School of Mathematics and Computer Science, Niger Delta University with 10 groups of students taking 70 courses in 6 available rooms.

CTP SOLVING TECHNIQUES

One of the earliest methods to solve timetabling problems was with graph coloring. A graph consists of a set of vertices and edges. Two vertices are said to be adjacent if they are connected by an edge. Solving the CTP with graph coloring involves assigning colors to vertices where all adjacent vertices are assigned different colors. Each vertices is a course and all vertices adjacent to it are courses that should not be placed in the same time slot. The CTP problem has been solved by Burke et al. (1995) and Kenekayoro (2011). Although they solved the examination timetabling problem (ETP), which is similar to the CTP, in the ETP, students may be in different rooms but in CTP all students must be in the same room. Other methods are local search optimization methods like hill climbing; where an initial solution is chosen randomly and then gradually improved. Hill climbing has a tendency to get stuck in a local maxima but with exception of known methods like stochastic hill climbing, a variation of hill climbing as well as simulated annealing improving result significantly (Bertsimas and Tsitsiklis, 1993), Tabu Search (Alvarez-Valdes et al., 2002). Other methods are ants colony optimization (Mayer et al., 2007) and others based on the theory of natural selection; genetic algorithms (Yu and Sung, 2002), evolutionary approach (Dawood et al., 2011) and (Datta et al., 2007), fuzzy genetic heuristics (Chaudhuri and De, 2010).

In genetic algorithms, a group initial solutions are generated randomly or using a local search method. The group is called a population and each solution in the population is a chromosome. New chromosomes (offspring) are generated by mutation (modifying a chromosome in a generation) or crossover (merging chromosomes in a generation). A fitness function evaluates each chromosome, and a new generation is formed by some of the parents and offspring based on the fitness functions. A relative new approach is hyper heuristics. Meta heuristic approaches like the ones mentioned earlier are problem specific too, a slight change in the problem domain causes massive change in the results. The aim of hyper heuristic is to be able to get a solution that may not be as good as the best meta heuristic approach but it is good enough and also generic. Hyper heuristic in a broad sense is using a meta-heuristic technique to select other lower level heuristics to solve a problem. Hyper heuristic could have top level meta-heuristic as simulated annealing (Bai et al., 2006), Tabu Search (Burke et al., 2003), case based reasoning (Burke et al., 2006). These techniques do not perform as good as the best heuristic approach in a specific problem domain but across several domains they outperform meta-heuristic approaches.

HILL CLIMBING AND SIMULATED ANNEALING

Hill climbing gradually improves a solution recursively by selecting the best neighbor based on an evaluation function recursively, until there is not a neighbor better than the current. If there is more than one best successor, a random from the set of best successors is selected.

Popular variants of hill climbing are: (i) First choice: the first successor better than the current is selected and (ii) Stochastic: any random uphill move is selected. The successor is not necessarily the best neighbor. Hill climbing is incomplete because it usually gets stuck in a *local maxima*; a state that is not the optimal but no neighbor is better than the current, *ridges*; a state where neighbors are series of local maxima, *plateau or flat top*; a state where all successors are equal. To escape from a local maxima random restart hill climbing is a good choice. In this case, the hill climbing algorithm is run several times with a randomly selected initial state. The random restart hill climbing algorithm is proven to be quite efficient, it solves the N queen problem almost instantly even for very large number of queens.

Hill climbing always gets stuck in a local maxima because downward moves are not allowed. Simulated annealing is technique that allows downward steps in order to escape from a local maxima. Annealing emulates the concept in metallurgy; where metals are heated to very high temperature and then gradually cooled so its structure is frozen at a minimum energy configuration. Applying this to the hill climbing optimization method, the probability of allowing a downward move is high (at very high temperatures) and this probability is gradually reduced with time (as it cools). The idea behind annealing is that, at high temperatures the algorithm should jump out of a local maxima.

CTP SOLUTION

For the CTP problem as earlier specified, a penalty of 1 (one) was given for a violation of each of the constraints in the final schedule. For each iteration, 25 neighbors were generated by moving a course in time_slot[i] to time_slot[i+1]. In the last time slot, 25; the neighbor is time slot 1. In the initial state, a random time slot and room that can contain students is assigned to a course.

Simulated annealing (random) where the successor is a randomly selected neighbor of the current as suggested by Russel and Norvig (2003) performed poorly in this case. It rarely outperformed the initial state. On the other hand, simulated annealing (best) where the successor is the best neighbor produced good results. At over 50 iterations, simulated annealing (best) performs better than hill climbing, although there were a few instances when hill climbing performed better (Figure 1).

Random restart hill climbing always performs better than the regular hill climbing, which is no surprise as random restart is the best out of successive regular hill climbing's. Random restart also outperforms simulated annealing because with random restart, a larger space is searched. It is worth noting that the best result was achieved with simulated annealing.

Although, no solution with penalty = 0 was reached, simulated annealing (best) got a penalty of less than 6 in 5 instances and a penalty = 1 (one) in an instance. Random restart always had a penalty less than 10. The best solution from random restart and regular hill climbing

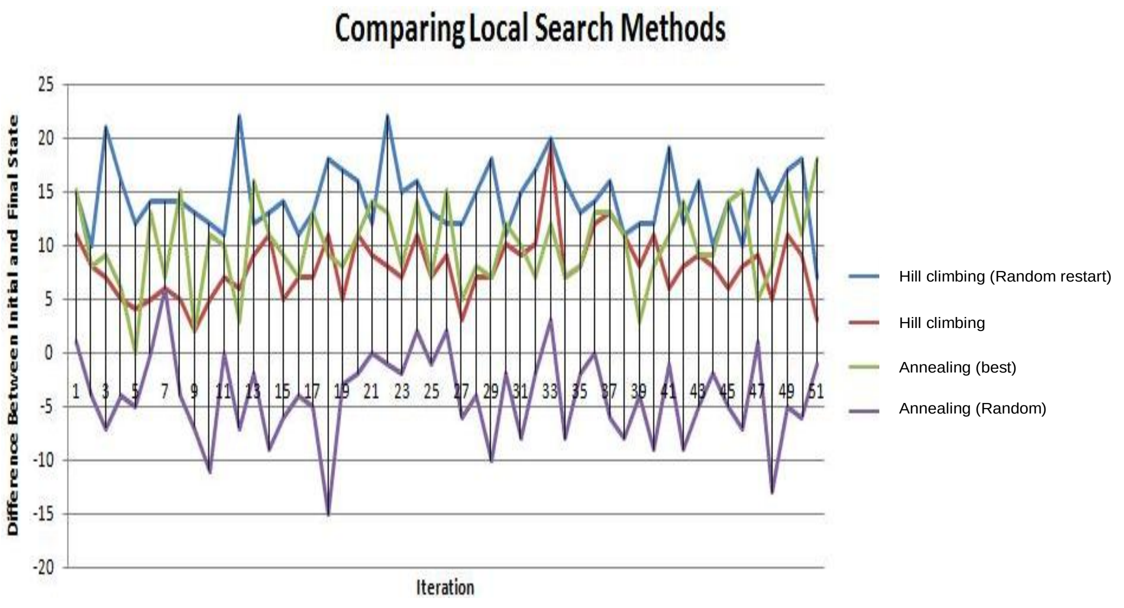

Figure 1. A comparison of hill climbing and simulated annealing for 50 iterations.

was penalty = 4.

When a solution with such low penalty is reached, a simple repair method that removes a course in a time slot that causes a collision and places it in a new time slot that does not increase the penalty produces a feasible solution. The solutions were reached almost instantly for regular hill climbing and simulated annealing with random restart hill climbing taking longer time to run.

CONCLUSION

Using simulated annealing or hill climbing is a good choice because of its ease of implementation. A solution was reached by generating only 25 neighbors. Alternative methods should be used only when simulated annealing or hill climbing is not good enough. Other methods like the genetic algorithm and hyper heuristics techniques still use these method at some point in their algorithm; either to generate the initial population or as a higher level heuristic.

The poor performance of simulated annealing (random) shows how problem specific meta-heuristic techniques are. It highlights importance of the new direction in operational research; hyper heuristics – *one size fits all* as opposed to meta-heuristics – *tailor made*. Tests were run on Windows Vista 32 Bit Operating system, MinGW c++ compiler.

REFERENCES

Alvarez-Valdes R, Crespo E, Tamarit J (2002). Design And Implementation of A Course Scheduling System Using Tabu Search. Eur. J. Oper. Res. 137:512-523.

Bai R, Burke E, Kendall G, McCollum G (2006). A Simulated Annealing Hyper-heuristic for Univ. Course Timetabling Problem. Nottingham Pract. Theory Autom. Timetabling pp. 345-350.

Bertsimas D, Tsitsiklis J (1993). Simulated Annealing. J. Stat. Sci. 8(1):10-15.

Burke EK, Petrovic S, Qu R (2006). Case-Based Heuristic Selection for Timetabling Problems. J. Scheduling 9(2):115-132.

Burke E, Elliman DG, Weare RF (1995). The Automation of the Timetabling Process in Higher Education. J. Educ. Technol. Syst. 23(4):257-266.

Burke E, Kendall G, Soubiega E (2003). A Tabu-Search Hyperheuristic for Timetabling and Rostering. J. Heuristics pp. 451-470.

Chaudhuri A, De K (2010). Fuzzy Genetic Heuristic for University Course Timetable Problem. Int. J. Adv. Soft Comput. Appl. 2(1):100-123.

Cooper TB, Kingston JH (1995). The complexity of timetable construction problems. Proceedings of PATAT. Springer-Verlag. pp. 283-295.

Datta D, Deb K, Fonseca CM (2007). Multi-objective evolutionary algorithm for Univ. Class Timetabling Problem. In Dahal, K. P., Tan, K. C., and Cowling, P. I. (eds.), Evolutionary Scheduling, Springer. pp. 197-236.

Dawood A, Awad Y, Badr A (2011). An Evolutionary Immune Approach for University Course Timetabling. Int. J. Comput. Sci. Netw. Secur. 11(2):127-136.

Even S, Itai A, Shamir A (1976). Complexity of Timetable and Multicommodity Flow Problems. SIAM J. Comput. 5(4):691-703.

Kenekayoro PT (2011). Automation of Examination Timetabling using Graph Coloring. West Afr. J. Sci. Technol. Soc. Sci. 1(1):53-56.

Mayer A, Nothegger C, Chwatal A, Raidl GR (2007). Solving the Post Enrolment Course Timetabling Problem by Ant Colony Optimization. Austria.

Russel SJ, Norvig P (2003). Local Search Algorithms and Optimization Methods. In Artificial Intelligence, A Modern Approach (2nd Edition) Prentice Hall, pp. 110-119.

Wren A (1996). Scheduling, timetabling and rostering - a special relationship? (E. Burke, and P. Ross, Eds.) Practice and Theory of Automated Timetabling, volume LNCS 1153 of Lecture Notes in Comput. Sci. pp. 46-75.

Yu E, Sung KS (2002). A genetic algorithm for a university's weekly courses timetabling problem. Int. Trans. Oper. Res. pp. 703-717.