Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 60 (2013) 16504-16506

ISSN: 2229-712X

Variation of the Convergence Speed on Basis of the Local Search Method Used in a Memetic Algorithm. Applying to the Scheduling Tasks Problem

B.Selma* and S.Chouraqui*

Department of Computer Science, Faculty of Science, University of Science & Technology Mohamed Boudiaf of Oran, Algeria.

ARTICLE INFO

Article history:

Received: 6 May 2013;

Received in revised form:

10 July 2013;

Accepted: 20 July 2013;

Keywords

Memetic algorithm,
Genetic algorithm,
Search local,
Convergence speed,
Scheduling problem.

ABSTRACT

In this paper, we study the variation of the convergence speed on basis of the local search method used in a memetic algorithm (MA); for this reason we compared two memetic algorithms MA1 and MA2 implemented with two different local search methods are descent and simulated annealing successively applied to the scheduling tasks problem. Then we will make a comparison between these two algorithms (MA1, MA2) results, with the genetic algorithm and other metaheuristics previously used to solve the same problem to get an idea on the most efficient methods in term of completion time work for this problem and choose the most effective.

Introduction

The memetic algorithms are advanced metaheuristics; the main idea of this technique is make a genetic algorithm more efficient by adding a local search over the mutation. One of the general observations from the implementation of a basic genetic algorithm [14] is often low speed of convergence of the algorithm. The idea of Moscato is then to add a local search that may be a descent method or a more sophisticated local search (simulated annealing, descent or tabu search).

It is obvious that this simple modification leads to profound changes in the behavior of the algorithm.

There are many ways to design a genetic algorithm. Local search methods are also very numerous. The hybridization of these two approaches allows considering a number of considerable combinations.

Scheduling is an NP complete problem [1], concerning the large-scale resource and job scheduling, and the adoptive and efficient job scheduling algorithm is required. Genetic algorithm show good capability to solve the problem of the small-scale, but with the increase in the number of jobs and resources, genetic algorithm is hard to convergence or slow convergence.

As stated above, using Genetic Algorithm (GA) technique to solve scheduling problem relying on Darwin evolutionary theory is one of the most popular and widely used algorithms which has been used in numerous cases and produced notable results [4]-[5]-[6]-[7]-[15]; moreover, there are other methods for solving Timetabling Problems, of which memetic Algorithms, Graph Coloring Algorithms and Colony of ants are the most important [3]-[8]-[16]-[17].

Local search have shorter completion times for job scheduling, and then algorithms search ability and convergence speed were compared. The simulation results shown that the proposed algorithm can effectively solve the job scheduling problem.

Genetic Algorithm at the Department of computer Science, University of Nottingham, which only applies to acceptable time intervals and is related to user interactively [3]. Khaled Mahar introduced a genetic algorithm for solving courses scheduling problem in 2006. He applied it to a university and reported its results [4]. Wilhelm Erben & Jurgen keppler also designed and implemented optimized tables of weekly courses utilizing Genetic Algorithms in 1995 [9]. In his proposed method, Michael W.Carter (2001) divides a problem into several sections and uses a greedy algorithm and an algorithm based on Lagrange function for allocating time intervals and classrooms, respectively [2].

Considering computational complexity of Genetic Algorithms, measures have been taken to increase their speed. Abramson D. &Abela J (1992) have taken advantage of parallel Genetic Algorithms to solve the problem of designing timetables for schools, reporting a relatively high time advantage compared with normal genetic algorithms [10]. Varac J et al. (2002) have presented a complete review on the application of genetic algorithms for designing to designing time tables [11]-[13]. Also, Moschopoulosa et al. (2008), in Belgians, have proposed and used an adaptive algorithm on the base of evolutionary computations for designing schedules in Greek high schools [12].

Memetic Algorithm (MA)

Initialization: Generate initial population Pop of solutions with size = n

Improve each solution s of Pop: s ← LS (s)
Repeat
Select two solutions x and x' with the selection technique
Cross both parents x and x' → children C1, C2
For each child C do
Improve: C ← SL (C)
C Muter
Replace a solution of P by Pop C

Tele:
E-mail addresses: selma.boumediene@yahoo.fr

End for

Until (stopping criterion).

Intensification is produced by applying the local search and the operator of mutation provides diversification.

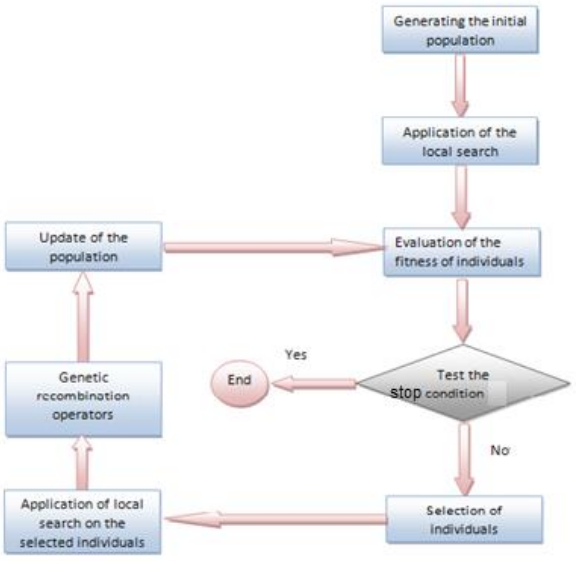

Figure 1. Flowchart of the hybrid algorithm.

Description of the algorithm

Step 1: Construct an initial population of | pop | random individuals or a greedy initialization.

Step 2: Apply the selection operator. The choice of these individuals can make different ways. It is desirable that parents have good properties to transmit to their sons.

Step 3: Apply the crossover operator which creates a new individual from two parent individuals.

Step 4: Apply the local search operator to set up new individuals (children). For n iterations and returns the best individuals found (Population of local minima).

Step 5: Apply the mutation operator is used to introduce new genes into the population. (To exit the local optimum).

Step 6: Update P replace the worst parent in C. The stopping criterion is different from one problem to another.

Obviously this change (the addition of local search) will change the behavior of the algorithm:

1. The first use of the local search procedure can work with intensified solutions and are among the best.
2. The second use of local search provides the best of children newly created generation.

Presentation of scheduling tasks problem

Production systems have experienced a tremendous development, where management production and scheduling tasks have become the element that create more and very important problems as the increased production and the decreasing costs, have become major objective in all companies.

Scheduling problems occur very often especially in optimizing the production management

The scheduling in production systems is to organize in time the realization of tasks so as to satisfy one or more target and taking into account the constraints of deadline, the constraints of concatenation.

Objective

Solving the problem is to find an order in which tasks input of the system and an allocation of different tasks on different machines in stage j+1, following the date of treatment to stage j.

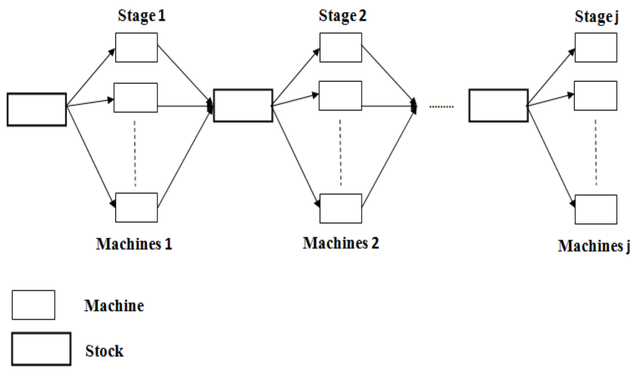

Figure 2. Representation of a production system

The scheduling problem: is to specify the order of execution of tasks by different machines in a given stage.

Comparison of MA with other methods

In the experiments we applied three comparisons

1. Memetic with (descent, simulated annealing): MA1 and MA2.
2. Genetic algorithms.
3. Local search (descent, simulated annealing).

Table 1 presents the values of CMAX (completion time work) depending on number of rooms of all the above algorithms. The results indicate that, AM2 values of the CMAX are always the shorter and it proves that he has the highest convergence speed and it is followed by MA1 then the other methods.

The memetic algorithms are the most powerful methods in term of completion time work and have shown their effectiveness as shown in Figure 2 and Figure 3.

Figure 1 shows the change of the local search method implemented in the memetic algorithm was influencing the results of CMAX.

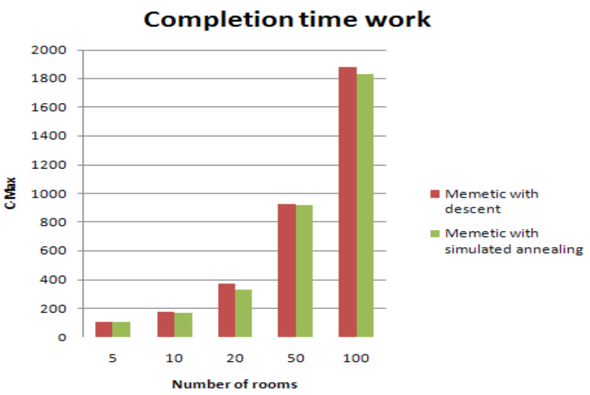

Figure 3. CMAX comparison between MA1 and MA2.

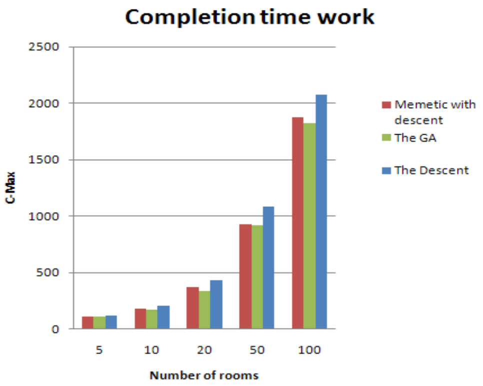

Figure 4. CMAX comparison between MA1, GA and Descent

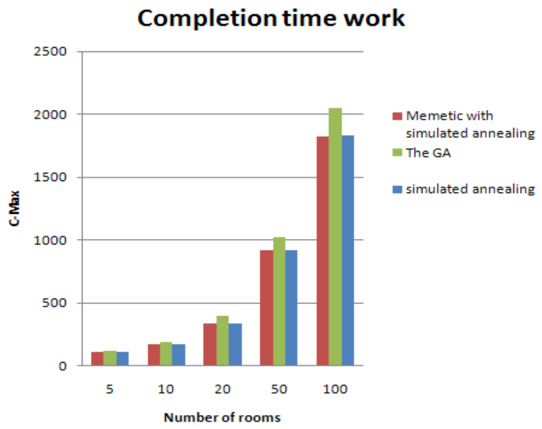

Figure 5. CMAX comparison between MA2, GA and Simulate annealing.

Table I

Comparison CMAX results

Methods / Number of rooms	MA1	MA2	GA	Descent	simulated annealing
N=5	105.2	104.7	113.4	110.4	106.4
N=10	173.4	163.8	179.4	198.6	168.3
N=20	366.8	329.2	389.2	430.8	330.1
N=50	920.4	910.3	1022.4	1081.6	911.2
N=100	1875.2	1823.9	2048	2073.4	1833.5

Conclusion

The present paper examined memetic algorithm properties as an appropriate tool to optimize CMAX responses. Then the problem of designing CMAX was an applied problem, and a Memetic algorithm solution was represented to it.

The memetic algorithms without doubt one of the most powerful methods and have shown their effectiveness in solving several problems by hybridization between two techniques "Genetic Algorithms" and "local search methods". And the combination of advantages of both techniques at the same time.

The memetic algorithms are more efficient than genetic algorithms, and also more efficient than local search methods such as simulated annealing and descent search.

The MA which gives the good rate success and offer less CMAX is MA2 while guaranteeing a good convergence to solve a given problem. Finally we conclude that the choice of the local search in MA plays an important role for the time o completion and of course that gives a good convergence speed.

References

[1] Cooper T., Kingston J., "The Complexity of Timetable Construction Problems, Lecture Notes in Computer Science", Vol. 1153, pp. 281-295, 1996.

[2] Carter M., "A Comprehensive Course Timetabling and Student Scheduling System at the University Lecture Notes in Computer Science, Vol. 2079, pp. 64-82, 2001.

[3] Burke E., Elliman D., Wearer R., "A Genetic Algorithm based University Timetabling System, Proceedings of the 2nd East-West International Conference on Computer Technologies in Education, pp. 35-40", 1994.

[4] Khaled Mahar, "AUTOMATIC GENERATION OF UNIVERSITY TIMETABLES: AN EVOLUTIONARY APPROACH", pp. 2-5, 2006.

[5] Aldy Gunawan, K. M. Ng, H. L. Ong, "A Genetic Algorithm for the Teacher Assignment Problem for a University in Indonesia",Volume 19, Number 1, pp. 1-16, 2008.

[6] Sadaf Naseem Jat, Shengxiang Yang, "A Guided Search Genetic Algorithm for the university Course Timetabling Problem", MISTA 2009.

[7] Z. Bratkovic, T. Herman, V. Omrcen, M. Cupic, D. Jakobovic, "University Course timetabling with Genetic Algorithm", a Laboratory Exercises Case Study 2008.

[8] Rossi-Doria O., Paechter B," A Memetic Algorithm for University Course Timetabling", Proceedings of the CO2004 Conference, Lancaster, UK, 2004,p. 65.

[9] Erben W., Keppler J., "A Genetic Algorithm Solving a Weekly Course-timetabling Problem", Proceedings of The First International Conference on The Practice and Theory of Automated Timetabling, Edinburgh, UK, 1995, pp. 198-211.

[10] Abramson D., Abela J., "A Parallel Genetic Algorithm for Solving the School Timetabling Problem of the 15th Australian Computer Science Conference, Hobart, Australia, 1992, pp.101.

[11] Vorac J., Vondrak I., Vlcek K., "School Timetabling Using Genetic Algorithm", Technical Report, VSB-Technical university of Ostrava, Czech Republic, 2002.

[12] Beligiannisa G., Moschopoulosa C., Kaperonisa G., Likothanassisa D., "Applying Evolutionary Computation to the School Timetabling Problem: The Case", Journal of Computers & Operations Research, Vol. 35, 2008, pp. 1265–1280.

[13] EnzheYa And Ki SeokSung , "A Genetic Algorithm For a University Weekly courses Time Tabling Problem", Intl. Trans in Op .Res. 9, 2002.

[14] Whitely D., "A Genetic Algorithm Tutorial", Journal of Statistics and Computing Vol. 4, 1994, pp. 65-85.

[15] Spyros Kazarlis, Vassilios Petridis and PavlinaFragkou, "Solving University Timetabling Problems Using Advanced Genetic Algorithms" pp.2-3.

[16] NysretMusliu, Werner Schafhauser, Magdalena Widl, "A Memetic Algorithm for a Break Scheduling Problem", MIC The VIII Metaheuristics International Conference, 2009.

[17] Olivia Rossi, Doria and Ben Paechter, "A Memetic algorithm for University Course Timetabling", Evolutionary Computation (GSICE),2002.