Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 7 No. 2.June, 2016 - www.cisdijournal.net

Comparative Performance of Modified Genetic Algorithm with Standard Genetic Algorithm for Solving University Examination Timetabling Problem

Alowosile, O.Y. [1], Oyeleye, C.A.[2], Omidiora, E.O.[2], Aborisade, D.O.[3] & Odumosu, A.A.[1]
[1] Department of Computer Science, Abraham Adesanya Polytechnic, Ijebu-Igbo, Ogun State, Nigeria.
[2] Department of Computer Science and Engineering, Ladoke Akintola University of Technology,Ogbomoso, Nigeria.
[3] Department of Computer Science, Federal University of Agriculture Abeokuta, (FUNAAB) Nigeria.
[1] alowosile78@gmail.com [2]caoyeleye@lautech.edu.ng

Corresponding Author: [2]caoyeleye@lautech.edu.ng

ABSTRACT

Although a number of existing works abound in the area of algorithm for solving University Examination Timetabling Problem, there are still issues of exam clashes and inability to keep memory of best solution(s) encountered. In this paper, an algorithm is proposed for solving University Examination Timetabling problem (ETP). The variations that are observed in the standard genetic algorithm (SGA) performance in generating possible timetables are studied. In the quest for greater effectiveness as far as convergence to the optimum solution is concerned, we therefore designed a modified Genetic Algorithm (MOGA) based approach. In other to prove the point that MOGA possess better convergence abilities than standard genetic algorithm, a methodology initially based on standard GA and later on hybridization with particle swarm optimization has been developed during this research. The Modified GA was used to schedule the 2013/2014 rain semester examination of Ladoke Akintola University of Technology, Ogbomoso, Nigeria with a task involving 19,127 students, 200 courses, 53 examination venues for 2 weeks excluding Saturdays and Sundays. Computational experience shows that Modified GA utilized least simulation time to return feasible results compared to the standard GA and maintains its accuracy level with increase in problem size, whereas standard GA looses its effectiveness as the problem size grows.

Keywords: Modified Genetic Algorithm, Standard Genetic Algorithm, Particle Swarm Optimization, Examination Timetabling, and Simulation Time.

CISDI Journal Reference Format
Alowosile, O.Y., Oyeleye, C.A., Omidiora, E.O, Aborisade, D.O. & Odumosu, A.A. (2016): Comparative Performance of Modified Genetic Algorithm with Standard Genetic Algorithm for Solving University Examination Timetabling Problem. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 7 No 2. Pp 53-64 Available online at www.cisdijournal.net

1. INTRODUCTION

Examination timetabling deals with the assignment of exams into a limited set of specific timeslots and rooms subject to a variety of hard and soft constraints (Al-Milli, 2011). Hard constraints have to be fulfilled under all circumstances, and soft constraints may be fulfilled if possible. Examination scheduling is a very important process in educational institutions. The main challenge is to schedule a number of exams for a set of students into timeslots and rooms over a fixed period of time while satisfying a set of constraints on both candidates and invigilators. It therefore follows that an exam overlaps must be avoided, and ensuring that exams are spread as much as possible. In Examination timetabling, there is usually a range of period such as two weeks of which exams are scheduled within the period. Also, the exam timetable must meet the students and lecturer preferences as much as possible; room could be shared by more than one resource such that different courses could be held in the same room at the same timeslot. Examination timetabling represents a difficult computational problem due to the strong inter-dependencies between exams caused by the many-to-many relationship between students and exams. ETP schedules the examinations to meet room availability constraints and to avoid overlapping examination times for individual student.

Timetabling researchers have investigated various methods of exploring the large combinatorial search space to generate timetables. These methods were typically formulated as heuristics assignment algorithms. However, it has been observed that no single heuristics that can be used to solve all timetabling problems because of the incorporation of problem-specific features in the heuristics (Burke et al., 1994). Qu et al. (2009) emphasized the specialization of the timetabling research into sub-areas of educational timetabling, nurse scheduling, transport scheduling, sports timetabling, etc. Scheduling is a process or a way of organizing time according to arrangement of work order plan.

It also means a list or activity table or activity plan with a detailed execution time. In university terminology, this scheduling problem is known as University Timetabling Problem (Banowosari and Valentine, 2011). However, according to Qu et al. (2009) the most studied and researched timetabling problem is the educational timetabling and in particular, examination timetabling. The families of related heuristics deployed in the solution of timetabling problems include: graph heuristics, meta-heuristics, constraint based methods, multi-criteria techniques, hybridization, and methods that focus on the investigation of neighborhood in the solution space.

In exam timetabling problems, the constraints are normally different from one institution to the other, which make it difficult to define the "standard timetabling problem'. However, at a general level the exam timetabling can be thought of as a process of ensuring that all students are able to take their exams and that the schedule of examinations for each student is designed so as to maximize the gap between consecutive examinations. Each ETP has its own set of hard constraints and soft constraints. Hard constraints must be satisfied at all times by the timetable. For example, a student cannot write two examinations at the same time, i.e. there must not be any clashes. Another hard constraint that needs to be obeyed is the room capacity; i.e. there must be enough spaces in a room to accommodate all students taking a given exam.

A timetable, which satisfies hard constraints of the problem, is called a feasible timetable. Soft constraints, on the other hand, are not critical but their satisfaction is beneficial to students and/or the institution. They are those that should be obeyed only if possible, and more often than not will describe what it is for a timetable to be good with respect to the timetable policies of the University. An example of a soft constraint is the requirement to spread out exams taken by individual students so that they have sufficient revision time between the exams they are enrolled on. Another soft constraint is that examinations with a larger number of enrolments must be scheduled early in the timetable to allow sufficient time for them to be marked. All soft constraint cannot be met and we claim to minimize the soft constraint cost for a timetable. Examinations are usually allocated to periods so as to ensure that hard constraints are not violated and soft constraint costs are minimized.

There are two versions of the examination timetabling problem, namely, the capacitated ETP and the uncapacitated ETP. In the capacitated version of the problem room allocation is taken into consideration while in the uncapacitated version it is not. Examination timetabling problem has been considered in different researches. According to Qu et al. (2009), they considered both theoretical and practical researches done in a ten year period. Some researches focused on room assignment to exams in order to minimize total movement of students between rooms during two consecutive exams (Dammak et al., 2006; Ayob and Malik, 2011). Ayob et al. (2007) solved a model with the aim of improving the quality of the timetable. It tries to minimize the number of students with two consecutive exams in a day. Burke et al. (2008) defined a model including seven objectives. They grouped the objectives in such a way that each group satisfied the specific party (students, markers, invigilators and estates). MirHassani (2006) developed a model based on a predefined exam timetable in order to maximize paper spread. Cheong et al. (2009) developed a model to minimize the length of timetable and objective model and solved it using an evolutionary algorithm. Sagir and Ozturk (2010) formulated invigilator assignment to exams as multi-objective model and calculated the weights of objectives using analytic network process. Kahar and Kendall (2010) applied a model in a real case regarding some new constraints such as distance between rooms. At the end they used a heuristic to solve the problem. McCollumm et al. (2012) considered an integer programming model with a cost penalty objective function. It tried to satisfy soft constraints as much as possible and if it failed, a penalty is accounted.

In this work, we compare performance of modified genetic algorithm with standard genetic algorithm for solving University examination timetabling problem. In the quest for greater effectiveness as far as convergence to the optimum solution is concerned, we designed a modified Genetic Algorithm (MOGA) based approach with better convergence abilities than standard genetic algorithm by incorporating a methodology initially based on standard GA and later on hybridization with particle swarm optimization. The remaining parts of this paper are organized as follows. Section two discusses related works/previous techniques applied to ETP, section three discusses methodologies used for solving the ETP problems, section four presents the results and result analysis while section five concludes the paper.

2. RELATED WORKS

The approach taken by earlier attempts at solving the ETP generally involved sorting examinations according to the difficulty associated with scheduling the examination, and allocating the most difficult examination first so as to ensure that clashes did not occur. In the case of clashes, re-allocation of examinations was performed. A low-level heuristic was used to assess the difficulty of an examination. Low-level heuristics generally used for this purpose include largest degree, largest enrolment, largest weighted degree and saturation degree (Pillay and Banzal, 2010). Research in this field was initiated by the study conducted by Carter (1986), which employed such a heuristic-based sequential technique with backtracking to generate examination timetables for 13 different academic institutions. These 13 real-world problems are now referred to as the Carter benchmarks.

Carter and Laporte (1996) updated the previous research and examined it either on real data or implemented it for real world problems. They divided the method into four models Cluster methods, sequential methods, meta-heuristics method and constraint based techniques. Burke et al. (1996) and Bardadym (1996) worked on timetabling problems by gathering all the required information. Odeniyi et al. (2015) presented a modified simulated annealing approach to the process of solving a typical high school timetabling problem.

Burke and Petrovic (2002) and in a follow up paper in 2004 worked on course and examination scheduling and used algorithms including hybrid evolutionary, meta-heuristics, multi-criteria, case-based reasoning techniques and adaptive approaches. Burkey and Landa Silva (2004) used a memetic algorithm in order to solve examination scheduling problems. Also, they put much effort on self-adaptive memetic algorithm design. Landa Silva et al. (2004) did research on multi-objective meta-heuristic techniques for educational timetabling problems. Their paper covered multi-phased approaches and multi-criteria evolutionary methods. Oyeleye et al. (2012) hybridized Simulated annealing and genetic algorithm in solving university examination timetabling problem. Schaerf and Di Gaspero (2006) proposed a new approach in University. Alade et al. (2014) presented Timetabling problem as an open ended problem that constitutes a class of difficult-to-solve combinatorial optimization problems that lacks analytical solution methods.

Their paper developed a university lecture timetable using genetic algorithm with modification made by introduction of replacement, test and repair strategy. Their modified genetic algorithm approach implementation of the timetabling problem was characterized by crossover, mutation and selection scheme. Arogundade et al. (2010) presented real world examination timetabling data set of the University of Agriculture, Abeokuta to solve a genetic algorithm approach for Examination timetabling problem using hierarchy of constraints. This hierarchy was used to incorporate individual request or organizational requirement by weighing them according to some criteria. In Egwali and Imuokhome (2012) a universal algorithm called Synthesis-Algo for solving complex and highly constrained timetable problem was introduced using hybridizing concept from evolutionary algorithm and Tabu searching techniques. They also programmed Synthesis-Algo using Matlab. Their results simulations were performed using simulated and real data (Egwali and Imuokhome, 2012).

3. METHODOLOGY

The processing algorithm used is the combination of Genetic algorithm and particle swarm optimization.

3.1 The Developed Modified Genetic Algorithm (MoGA)

The developed Modified Genetic Algorithm (MoGA) is a combination of PSO and Standard GA. The PSO phase is used only for the worst solutions and it assists the GA phase essentially through improving the efficiency and performance of the overall procedure. In the first step, the population is randomly initialized over the search space with a uniform distribution. The population is then moved through two sequential phases to find the best feasible solution. The first phase involves the enhancement of population with worst fitness using the PSO. The objective function and constraints violation of the population are evaluated, and the individuals are ranked using pair-wise comparison as described below (Dhadwal et al., 2014).

- A feasible solution is preferred when compared to an infeasible one.
- When two feasible solutions are compared, the one with better objective function is preferred.
- When two infeasible solutions are considered, the one with small constraint violation is preferred.

The population then proceeds to the next phase through the standard GA. The parents are selected using a binary tournament selection scheme. The crossover and mutation operations are then performed, and the population is again ranked according to the values of the objective function and constraint violation. The solutions are directed to the stagnation check phase, to avoid the local optimum solutions where a solution is randomized if there is little or no change in the value of the corresponding objective function. The iteration loop is continued until the termination criteria are satisfied.

The flowchart for the developed MoGA is shown in Figure1. The particle swarm optimization phase and the genetic algorithm employed in the current approach are described in detail in the next sections.

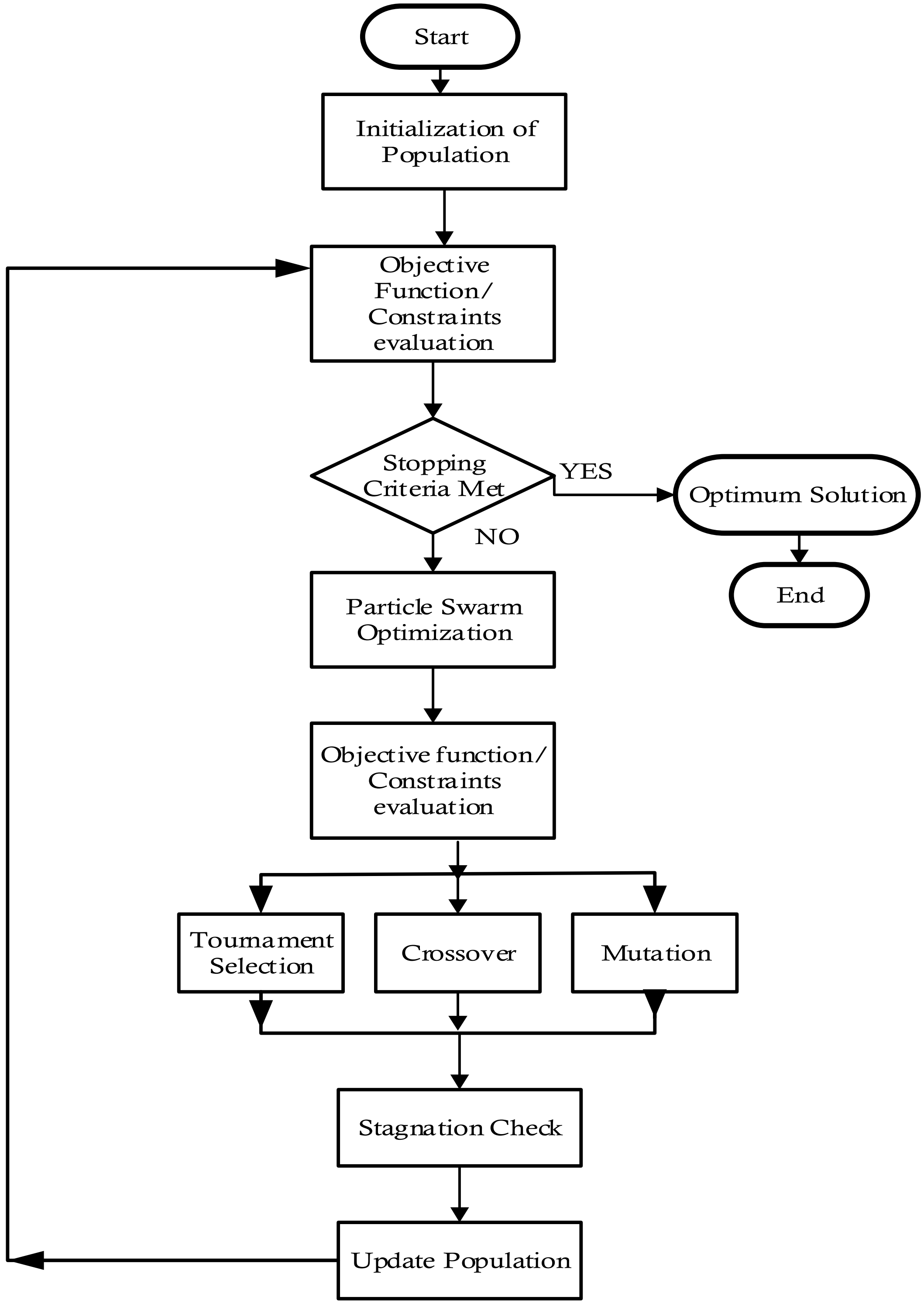

Figure 1: Flowchart of the Developed Modified Genetic Algorithm (Oyeleye et al., 2016)

3.1.1 Particle Swarm Optimization Phase

Particle swarm optimization (PSO) also known as swarm intelligence is an algorithm developed by kennedy and Eberhart in order to solve problems with continuous search space (Eberhart and Kennedy,1995). PSO is a stochastic optimization technique based on the movement and intelligence of swarms. PSO applies the concept of social interaction and communication, such as bird flocking and fish schooling for problem solving. This algorithm can be easily implemented and its computationally inexpensive, since its memory and CPU speed requirements are low. In PSO, a bird of a flock is represented as a particle, and the swarm is composed of a group of particles.

These particle are provided with initial velocities and certain learning constants and values at the beginning. The position of each particle can be represented as the candidate solution to an optimization problem. Every particle is given a fitness function designed in correspondence with the corresponding problem. When a particke moves to a new position in the search space, it will remember its personal best(P_{best}) known as the best position of the particle. In addition to remembering its own information, each particle will also exchange information with the other particles and remember the global best (G_{best}) also known as best position of the swarm.

Then, each particle will revise its velocity and direction in accordance with its P_{best} and G_{best} to move toward the optimal value and find the optimal solution(Ruey-Maw and Hsiao-Fang,2013). To begin a PSO algorithm, the initial velocity and position of each particle in a group of particles are randomly determined. At each iteration, the particle moves around according to its velocity and position; the cost function to be optimized is evaluated for each particle in order to rank the current location. The velocity and position of the particle is then stochastically updated equations 1

$$v_{id}^{t+1} = \omega v_{id}^{t} + c_1 r_1^t (p_{id}^t - x_{id}^t) + c_2 r_2^t (p_{gd}{}^t - x^t{}_{id})$$
$$x_{id}^{t+1} = x_{id}^t + \mathrm{v}_{id}^{t+1} \tag{1}$$

In the equation,

v_{id} is the velocity component of the ith particle in the dth dimension .

x_{id} is the position component of the ith particle in the dth dimension.

c_1 is the cognitive parameter.

c_2 is the social parameter.

p_{id} is the position component of the P_{best} of the ith particle in the dth dimension.

p_{gd} is the position component of the G_{best} in the dth dimension.

r_1 and r_2 represent random numbers in the range [0,1]

ω is the inertia weight.

The latter term (ω) plays an important role in the PSO convergence behavior since it is employed to control the exploration abilities of the swarm. It directly affects the current velocity, which in turn is based on the previous history of velocities. Large inertia weight provide for global exploration of the search space, while small inertia values concentrates the velocity update to nearby regions of the design space.

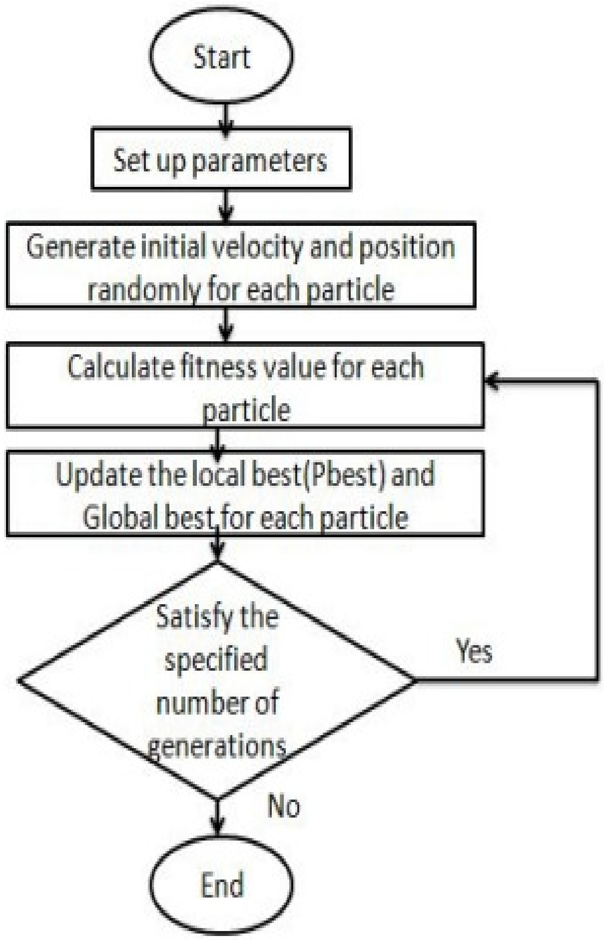

Figure 2: Flowchart of PSO (Ezgi and Sodik,2014)

3.1.2 Genetic Algorithm

Genetic algorithm is one of the most important meta-heuristic methods applied for examination scheduling problems that try to find solutions to NP-hard problems through evolution.GA is a population-based evolutionary heuristic, where every possible solution is represented by a specific encoding, often called Individual (Colorni *et al.,* 1993). The underlying principle of GA is the survival of the fittest individual, i.e. the best solution. The feature of exploration and intensification are employed to continually find the best solution. However, Standard GA generally suffers from slow convergence and to overcome these difficulties Modified GA is developed by incorporating certain modifications using PSO.

3.1.3 Stagnation Check

The stagnation check is performed to avoid the solutions getting trapped in local optimal. If the difference of the objective functions between the current and the previous iteration is less than 0.1% for a successive specified number of iterations, the solution is re-initialized in the given search space.

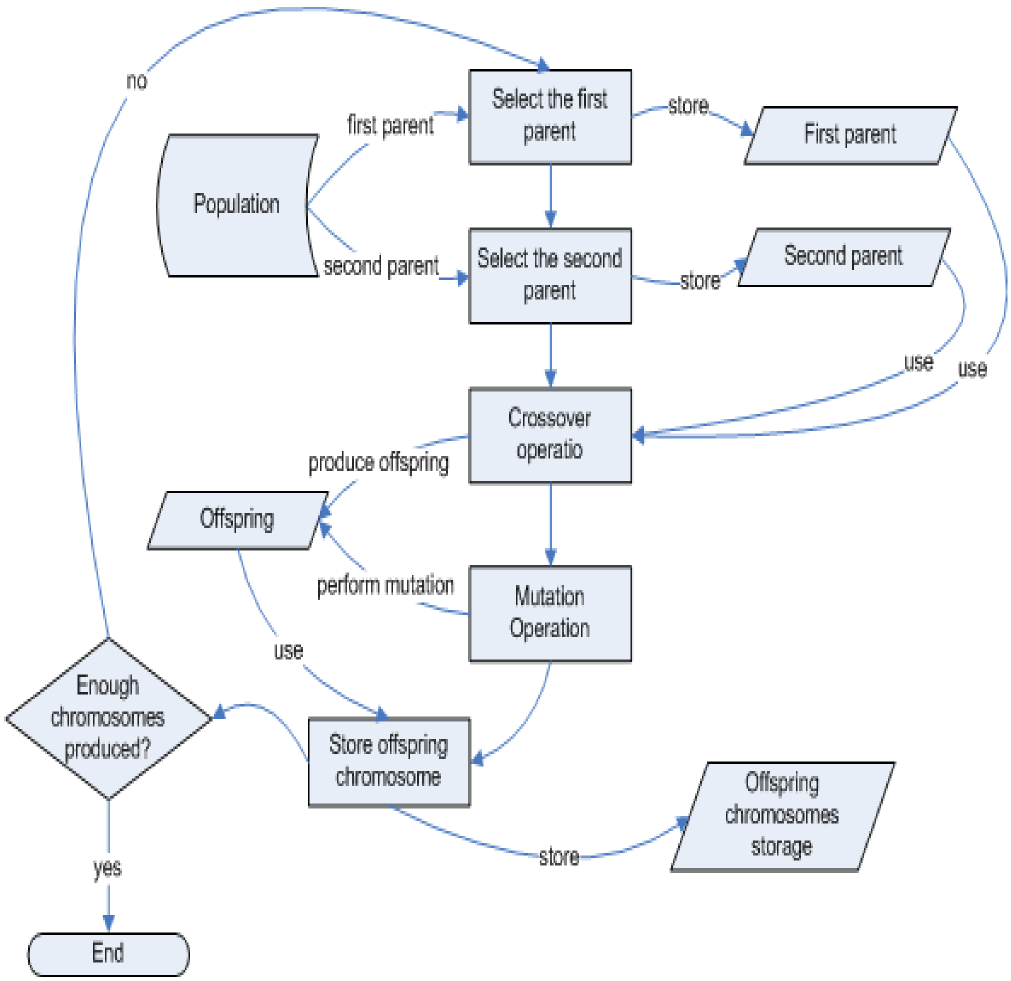

Figure 3: Flowchart of Genetic Algorithm (Mladen, 2012)

Table1: Summary of Constraints Considered

Constraints	Description
HC1	There cannot be any student sitting for more than one exam at the same time.
HC2	The total number of students assigned to each room cannot exceed the room capacity.
HC3	The duration(length) of an exam assign to each timeslot should not violate the timeslot length.
HC4	The number of classes to be invigilated by a lecturer at a time
SC1	Exams must be spread out to give students more time to study. i.e. no student must have exams in two consecutive timeslots.
SC2	Examination splitting over similar rooms within same timeslot must beminimized. i.e. total number of consecutive classes a lecturer should invigilate.

HC=Hard Constraints SC=Soft Constraints

3.2 Data used for the work

The following are the set of data used to automatically generate the examination timetable:

- List of examinations to be attempted
- List of available invigilators
- Duration of each examination in hour
- Number of examination days/weeks
- Venues and their corresponding capacity
- Number of registered students per examination

3.3 Performance Evaluation Metrics

In order to compare the performance of the coded algorithms, software complexity evaluation of the two algorithms was considered in terms of Halstead complexity measures, such as program volume, program effort, program level and intelligent content of the program. The second complexity measure considered is the line of codes (LOC).The simulation time is the parameter which measures the time utilized by an algorithm to run until the result is produced. The constraints violation is the metric which determines the feasibility or validity, and the goodness of the solution produced by an algorithm.

4. RESULT AND DISCUSSIONS

4.1 Analysis of Results

After the implementation of the two algorithms, Table 2 shows the measured metrics and the various values used in comparing the performance of the two algorithms. As presented in table 1, it is clearly shown that the two algorithms produced feasible solutions since both the hard and Soft constraints were not violated.

Table 2: Parameters for measuring the Computational Complexity

Measured Metrics	SGA	M_oGA
No of Unique operators (n1)	16	20
No of Unique operands (n2)	32	60
Total Number of operators (N1)	48	80
Total Number of operands (N2)	96	240
N =(N1+N2)	144	320
n = (n1+n2)	48	80
Line of codes	157	69
Program size(B)	4340	2752
Program Volume(V)	2787.034	1251.031
Program Effort(E)	30024.75	111481.3
Intelligent Content	24	40
Difficulty	52.1263	69.6759
Simulation Time(Seconds)	22.035	20.136
Computational Cost	1.0295	0.030409
Numberof Hard Constraints violated	0	0
Number of Soft Constraints Violated	0	0
Number of Invigilators double booked per time	0	0

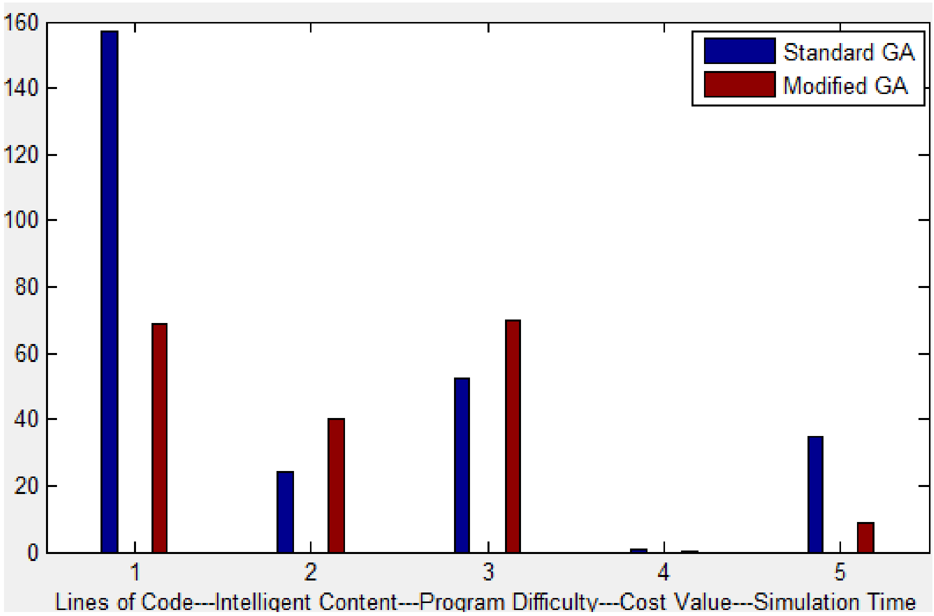

Figure 4: A diagram comparing the two algorithms in terms of the parameter metrics

4.2 Discussion of Results

The results obtained in table 2 show that the modified genetic algorithm is more efficient than the standard genetic algorithm. The Comparative performance of both modified genetic algorithm and standard genetic algorithm was carried out using MATLAB R2012a on Hp Probook system with window 7, 2.40GHz and 4.0GB. The simulation time of GA and $\mathbf{M_oGA}$ are 22.035secs and 20.136secs respectively to return a feasible timetable. The experimental results also show that Modified genetic algorithm utilizes less CPU time than the Standard genetic algorithm. Also, the complexity comparism of both the program volume and programming effort which are 2787.034 and 30024.75 for Standard GA and 1251.031 and 111481.3 for Modified GA show that modified genetic algorithm is less in value than the standard genetic algorithm. The program volume shows that Modified GA occupies lesser memory space in term of volume than the Standard GA while the program effort is an indication that the Modified GA requires lesser effort in the algorithmic implementation.

The line of code of Standard GA and Modified GA are 157 and 69 respectively which is an indication that the Modified GA has the smallest implementation time and effort. The program size of standard GA and Modified GA are 4340 and 2752 which is a measure that clearly shows that Modified GA occupies lesser disk space. The computational cost of standard GA and Modified GA are 1.0295 and 0.030409 respectively to return a feasible solution. This is a clear indication that the modified GA used less computational cost and converges faster than the standard GA. This is as a result of the enhancement provided by the particle swarm optimization algorithm. However, the two algorithms did not violate any of the hard and soft constraints as shown in the summary of results in table 2. Figure 4 shows the relationship between the two algorithms in terms of the measured parameter metrics.

5. CONCLUSIONS

The efficient creation of examination timetable is a recurring and important problem for universities worldwide. Good timetables are characterized by balanced distances between consecutive exams for all students. In this contribution an approach for the examination timetabling problem based on the modified genetic algorithm with particle swarm optimization was proposed to produce feasible exam timetables. The computation of PSO is easy and adds only a slight computation load when it is incorporated into the Standard GA. Furthermore, the flexibility of PSO to control balance between local and global exploration of the problem space helps to overcome premature convergence in GA and also enhances searching ability.

The behaviour of the developed modified genetic algorithm and the standard genetic algorithm was examined with respect to parameter variations in order to carry out the comparative analysis of the two algorithms in a bid to test the performance of the proposed system. The modified genetic algorithm based technique was successfully developed to generate a conflict free, more efficient and effective examination timetable with respect to all the considered metrics.

The developed M_oGA thus make provisions for a robust examination timetabling system that will ease administrators of the stress usually associated with manual timetabling with drastic reduction in the time spent in its preparation. This work therefore recommends the use of the developed M_oGA in solving University examination timetabling problems subject to highly reduced simulation time and program effort, less disk and memory space as well as a very high intelligent content of the program.

REFERENCES

[1] Alade, O., Omidiora, E.O., and Olabiyisi, S.O.(2014). "Development of a University Lecture Timetable using Modified genetic algorithms approach". *International journal of Advanced research in Computer Science and Software Engineering*. Vol. 4, Issue 9.

[2] Al-Milli, R.N.(2011). "Hybrid genetic algorithms with Simulating Annealing for University course Timetabling problems". *Journal of theoretical and applied Information Technology, Islamabad Pakistan.* Vol. 29.

[3] Arogundade, O.T., Akinwale A.T., and Aweda O.M. (2010). "A genetic algorithm approach for a real- world University Examination Timetabling problem". *International Journal of Computer applications.* Vol. 12, No. 5.

[4] Ayob, M., and Malik, A.,(2011). "A new model for an examination-room assignment problem". *International Journal of Computer Science Network and Securities.* Vol. 11(10): Pp 187-190.

[5] Ayob, M.,Abdullah and Malik A.,(2007). "A practical Examination timetabling problem at the Universiti kebangsaan Malaysia". *International Journal of Computer Science Network Security and Security.* Vol. 7(9):Pp198-204

[6] Banowosari, Y.,L., and Valentine, V.(2011). "University Timetabling Algorithm considering Lecturer's workload". *The sixth international multi-conference on computing in the Global information technology.*

[7] Bardadym, V.A.(1996)."Computer aided School and University Timetabling", the new wave. In: Practice and Theory of Automated Timetabling, Springer Lecture notes in Computer Science. Vol.1153: Pp 22-45.

[8] Burke, E.K and Petrovic, S (2002). "Recent research directions in automated timetabling". *European Journal of operation research.* Vol. 140(2) : Pp266-280.

[9] Burke, E.K, Elliman, D.G. and Weare, R.F. (1994). "A University Timetabling System based on Graph Colouring and Constraint Manipulation". *Journal of Research on Computing in Education*, 27(1): Pp 1-18.

[10] Burke, E.K., and Landa Silva, J.D., (2004) "The design of Memetic algorithms for scheduling and timetabling problems". In: recent advances in memetic algorithms and related search technologies, studies in Fuzziness and Soft computing 166, Springer, Berlin, New York : Pp 289-312.

[11] Burke, E.K., Elliman, D.G., Ford, P. and Weare, R.F. (1996). "Examination timetabling in british Universities". A Survey. In: Practice and Theory of Automated Timetabling, Springer Lecture notes in computer science, Vol.1153: Pp 76-90.

[12] Burkey, E.K., McCollum, B., McMullan, P. and Parkes A.J.,(2008). " Multi-objective aspects of the Examination Timetabling competition track". *Proceedings of PATAT 2008*, Montreal, Canada.

[13] Carter, M.W, Larporte, G. (1996). "Recent developments in practical examination timetabling". In: Practice and theory of Automated Timetabling, Springer Lecture notes in computer science, Vol.1153: Pp 3-21.

[14] Carter, M.W.(1986). "A survey of practical applications of examination timetabling algorithms". *Operation research*. 34(2): Pp 193-202

[15] Cheong, C.Y., Tan, K.C. and Veeravalli, B.(2009). "A multi-objective Evolutionary algorithm for examination timetabling". J.Sched., 12(2): Pp 121-146.

[16] Colorni, A., Dorigo, M., Maniezzo, V. and Trubian, M. (1993). "Ant system for job-shop Scheduling". *Belgium Operations.Research, Statics.and Computer Sci.* 34: Pp 39-53.

[17] Dammak,A., Elloumi,A.,and Kamoun(2006). "Classroom assignment for Exam Timetabling". Advanced Engineering software, 37(10): Pp 659-666

[18] Dhadwal, M.K., Jung, S.L., Kim, and C.J. (2014). "Advanced particle swarm assisted genetic algorithm for constrained optimization problems". *Comput Optim.* Vol. 58: Pp 781-806.

[19] Eberhart, RC., and Kennedy, J. (1995). "A new optimizer using particle swarm theory". In: *Proceedings of the 6th International symposium on micro machine and human science*. Nagoya, Japan, Pp 39-432

[20] Egwali A.O and ImouokhomeF.A(2012). "Synthesis-Algo: an inclusive timetable generating algorithm". *African journal of computing and ICTs.* Vol 5, No. 4. Pp 92-100.

[21] Ezgi D.Ü and Sadık U(2014). "Application Of Particle Swarm Optimization to Microwave Tapered Microstrip Lines", Computer Science & Engineering: *An International Journal (CSEIJ)* Vol. 4, No. 1 : Pp 59-64.

[22] Holland, J. (1975). "Scheduling Adaptation in Natural and Artificial Systems". The University of Michigan Press, MI, Pp 60.

[23] Kahar, M.N.M and Kendall, G.(2010). "The Examination Timetabling Problem at University Malaysia Pahang : Comparison of a constructive heuristic with an existing software solution". *European Journal of Operation research*. 207(2): 557-565.

[24] Landa Silva, J.D., Burke, E.K., and Petrovic, S. (2004) . " An introduction to Multi-Objective Meta-heuristics for Scheduling and Timetabling". In: lecture notes in Economics and Mathematical Systems, Springer, Vol. 535: Pp 91-129.

[25] McCollum, B., McMullan, P., Parkes, A.J., Burke, E.K., and Qu, R.(2012). " A new model for automated Examination Timetabling". *Annual Operation research*. 194(1): 291-315.

[26] Mirhassani, S.A.(2006) . "Improving paper spread in Examination Timetables using Integer programming". Applied Mathematics and computer. 179(2): Pp 702-706.

[27] Mladen, J.(2012). " Genetic Algorithm Library".

[28] Odeniyi,O,A., Omidiora,E.O., Olabiyisi, S.O., and Aluko, J.O.(2015) . "Development of a Modified Simulated Annealing to School Timetabling problem". *International Journal of Applied Information Systems*. Vol. 8, No. 2

[29] Oyeleye,C.,A., Olabiyisi, S. O., Omidiora, E., O., and Oladosu, J.B., (2012). "Performance evaluation of Simulated Annealing and genetic algorithm in solving Examination Timetabling problem". *Scientific research and essays*. Vol.7(17): Pp.1727-1733.

[30] Oyeleye, C.A., Omidiora, E.O., Alowosile, O.Y., Aborisade, D.O.,and Odumosu, A.A., (2016). "Comparative Performance of Modified Genetic Algorithm with Standard Genetic Algorithm for solving University Examination Timetabling Problem". In Proceedings of the fourth International Conference on Engineering and Technology Research. Vol 4 Pp 422-429.

[31] Pillay, N., Banzhal, W. (2010) . " An informed genetic algorithm for the Examination Timetabling Problem". *Journal of applied Soft Computing*.Vol 10: Pp 457-467.

[32] Qu,R., Burke, E.K., McCollum, B., Merlot, L.T.G and Lee, S.Y.,(2009). "A survey of search methodologies and automated system development for Examination Timetabling". J.Sched., 12(1): Pp 55-89.

[33] Reeves, C.R. and Yamada, T. (1995) . " Genetic algorithms: path relinking and the flowshop sequencing problem". Evolutionary Computation 6: Pp 230–234.

[34] Ruey-May, C., and Hsiao-Fang, S., (2013). "Solving University course timetabling problems using constriction particle swarm with local search". Open access algorithms: Pp 227-244

[35] Sagir,M. and Ozturk Z.K.(2010). "Exam scheduling, Mathematical Modelling and parameter estimation with the analytic network process approach". Math. Comp. Modl., 52(5-6): Pp 930-941

[36] Schaerf, A., and Di Gaspero, L.,(2006) ." Measurability and reproductivity in Timetabling research". In: *Proceedings of the 6th International Conference on practice and theory of automated Timetabling*. Pp 53-63.