Tanzania Journal of Science 46(1): 9-18, 2020 ISSN 0856-1761, e-ISSN 2507-7961

The Application of Late Acceptance Heuristic Method for the Tanzanian High School Timetabling Problem

Allen R. Mushi
Department of Mathematics, University of Dar es Salaam, P. O. Box 35062
Dar es Salaam, Tanzania. E-mail: allenmushi66@gmail.com

Received 10 Oct 2019, Revised 30 Dec 2019, Accepted 1 Jan 2020, Published 31 Mar 2020

Abstract

High School timetabling is the problem of scheduling lessons of different subjects and teachers to timeslots within a week, while satisfying a set of constraints which are classified into hard and soft constraints. This problem is different from university course timetabling problem because of the differences in structures including classroom allocations and grouping of subject combinations. Given the scarce education resources in developing countries, high school timetabling problem plays a very important role in optimizing the use of meager resources and therefore contribute to improvement of quality of education. The problem has attracted attention of many researchers around the world; however, very little has been done in Tanzania. This paper presents a solution algorithm known as Late Acceptance heuristic for the problem and compares results with previous work on Simulated Annealing and Great Deluge Algorithm for three schools in Dar es Salaam Tanzania. It is concluded that Late Acceptance heuristic gives results which are similar to the previous two algorithms but performs better in terms of time saving.

Keywords: Late Acceptance, High School Timetabling, Combinatorial Optimization, Heuristics, NP-Hard

Introduction

High School Timetabling Problem (HSTP) is the problem of scheduling teachers and lessons of various subjects to a set of timeslots within a week while satisfying a set of constraints. This problem differs from the university course timetabling problem. In high school timetabling, classrooms are fixed for each class where each class is assigned to a fixed set of class combination depending on the subjects taken by the class; while in university timetabling, rooms are not fixed to a program group and students in a program can take varying courses. High School timetabling is a very important activity in optimization of teaching and learning resources especially in developing countries where these resources are scarce. It plays an important role in improving the quality of education in secondary schools.

This problem is NP-Hard implying that there is no algorithm that is known to provide an optimal solution to such a class of problems within reasonable time (Even et al. 1976). Heuristic algorithms have been the most favorable options for such problems. A lot of attention has been placed by researchers especially in the developed world and vast numbers of papers are available on high school timetabling. The problem has many variants depending on the educational system of a country. International Timetabling Competition in 2011 collected instances of problems from different countries to support researchers in tackling challenging problems in high school timetabling (Post et al. 2013). They collected 35 problems from Australia, Brazil, United Kingdom, Finland, Greece, Italy, Netherlands and South Africa, and some groups presented solution algorithms for their specific problems. Clearly, there are many more variants of the problem and therefore calling for more instances with different features.

Approaches to the problem can be grouped into exact and heuristics. Exact approaches include the work by Burke et al. (2013) that gives a general classification of timetabling problems and presents modeling approaches using graph theory. Birbas et al. (1997) provides an Integer Programming model and tested successfully on Greek high schools. Simon et al. (2014) presented a generalized model for high schools timetabling through Integer Programming. The model was tested on instances from the ITC and reported improved solutions to some of the problems in the list. Ribic et al. (2015) modeled constraints of a high school timetabling problem using Integer Programming and reported successful results to some instances of the problem. There are many other research results on exact methods using Integer and Mixed Integer Programming models including Willemen (2002), Matias and Riis (2012) and Valouxis et al. (2012). Models that apply constraint programming technique are also common and include; Marte (2002), Muller (2005) and Demirovic and Stuckey (2018). However, these exact methods do not guarantee that optimal solution will always be found for a general problem and may get stuck when problem size grows; a typical characteristic of NP-Hard problems. Heuristic approaches have been reported in many cases in the literature. Simulated Annealing has been used for high school timetabling problems for specific instances (Abramson 1991, Zhang et al. 2010). Tabu Search has also been applied in several papers including the work by Schaerf (1996). A presentation of Tabu Search, Simulated Annealing, Genetic Algorithms coupled with Branch and Bound is studied by Wilke and Ostler (2008). Evolutionary algorithms are also applied in the problem including the work by Filho and Lorena (2001) and also Fernandes et al. (2002). Graph based heuristics have been reported, such as the work presented in Burke et al. (2007). Metaheuristics techniques which combine various heuristics into one algorithm have been tested in several cases including Colorni et al. (1998). Other specific heuristics which have been applied in high schools timetabling include; fix-and-optimize (Dorneles et al. 2014), Bacteria foraging (Kunthavai and Rajithaa 2018) and Local Search techniques (Schaerf 1999). Katsaragakis et al. (2015) performed a comparative study of modern heuristic techniques for high school timetabling by looking at population based methods; Particle Swam and Artificial Fish Swam applied to specific schools where they both performed well. In principle, many successful heuristic algorithms have been reported; however, they are all based on specific instances of applications, either from collected libraries of problems or specific schools. Since schools are different and have different features, it is necessary to investigate specific cases of schools which have not been explored.

Specific features of the selected schools

High school students in Tanzania are admitted to take a total of three courses called combinations. For instance a student may take Physics, Chemistry and Mathematics (PCM) or Physics, Chemistry and Biology (PCB) or History, Geography and Kiswahili (HGK) and many other possible choices. Each combination is considered to be a separate class and will occupy one room throughout their two years of high school studies, i.e., form V and form VI. Each subject has a set number of lessons which are to be taught to the students every week for a full term. A teacher may be assigned to teach more than one subject (maximum of two) in the same or different combinations. There are subjects that are shared among the students including General Studies, Religious Studies and some compulsory courses depending on combination such as Basic Applied Mathematics (BAM) for business, economics and biological sciences students.

A timetable is said to be feasible if it satisfies a set of hard constraints; these include the following:

1. No class can be taught more than one lesson at the same time;
2. No teacher can be assigned to teach more than one lesson at the same time;
3. Lessons of the same subject cannot be assigned to the same timeslot;
4. Compulsory lessons must have common times for all students (Crosscutting subjects);
5. All lessons of all subjects must be scheduled for a complete timetable.

Apart from these hard constraints there is a list of soft constraints; these have to be satisfied as much as possible and therefore form part of the objective function in the formulation of the problem. The following is a list of soft constraints as used in this work:

1. Teachers have to commute to and from work every day and face challenges of traffic jams. It is therefore preferred to minimize as much as possible the use of early morning and late evening timeslots.
2. Since a subject has several lessons, it is preferred to spread these lessons of the same subject as far as possible.
3. Some lessons and teachers have preferred free times during the day, for instance Biology students may need free times to collect and prepare samples for laboratory experiments.

Late Acceptance Heuristic

Late Acceptance Heuristic (LAH) is a relatively new global heuristic method and it is one of the *one-point* iterative techniques. Global heuristic techniques employ various strategies to avoid falling into local optima by accepting bad moves in anticipation of better moves in the future. LAH avoids falling into local optima by accepting a bad solution by comparing it with a solution that was current several iterations back. In this case, a list of moves of size L is created and a new solution is accepted if it is better than current or better (or equal) to a solution which was current L number of iterations back. Burke and Bykov (2017) presented the so called "*Late Acceptance Hill-Climbing Heuristic*" which describes the algorithm for a case of a maximization problem. Many global heuristics techniques use cooling schedules to guide convergence into optimal solutions like Simulated Annealing (Henderson et al. 2003) and Tabu Search (Glover and Laguna 1998); whose performance is largely dependent on the choice of parameters by the user. LAH does not have a cooling schedule and requires only one input parameter, the list length L and therefore more stable with less dependency on user input. The procedure is as shown in Algorithm 1 which has been adapted from Burke and Bykov (2017) to a minimization case.

Algorithm 1: Late_Acceptance_Heuristic
Produce an initial solution X
Calculate initial cost value $Z = f(X)$
Assign best solution so far $Z_b = Z$
Define fitness array $C(X)$
Specify L
First iteration $I = 0$;
For all $k \in \{0 \dots L-1\}$ $C(k) = Z$
While not stopping criteria (Fixed iterations)
Construct a candidate solution in neighborhood i.e., $X_c \in Neighbor(Z_b)$ as current solution
Calculate its cost function $Z_c = f(X_c)$ `
    $v = I \bmod L$
    If $Z_c < Z_b$ or $Z_c <= C(v)$
    Then accept the candidate ($Z_b = Z_c$)
    Else reject the candidate (Z_c)
    Insert the current cost into the fitness array $C(v) = Z_b$
    End if

Increment the iteration number $I = I + 1$
End While
Return best solution (Z_b)

The algorithm has been implemented in several applications with recorded success including university course timetabling (Marwa and Mushi 2013), examinations timetabling (Ozcan et al. 2009) and vehicle routing (Souza et al. 2019) among others. Only one paper has been found to the best of knowledge that addresses high school timetabling, that is the work by Fonseca et al. (2016), for timetabling competition instances. It is worth therefore to contribute more work by introducing the algorithm to new real application problems.

Mathematical Formulation

Mathematical programming model is formulated in Mushi (2011) and is summarized as follows:

Define a variable

$$x_{ijk} = \begin{cases} 1 & \text{if lesson } i \text{ of subject } j \text{ is assigned timeslot } k \\ 0 & \text{Otherwise} \end{cases}$$

Then

Minimize $f(x) =$

$$\lambda_1 \sum_{(i_1,i_2)\in H} \sum_{(k_1,k_2)\in K' \ni k_1 \neq k_2} \sum_{j\in J \ni x_{i_1 j k_1} + x_{i_2 j k_2} = 1} \frac{1}{(k_1 - k_2)^2}$$

$$+ \lambda_2 \sum_{k\in S_M} x_{ijk} + \lambda_3 \sum_{k\in S_A} x_{ijk} + \lambda_4 \sum_{k\in S_P} x_{ijk} \quad (1)$$

Subject to:

$x_{ipk} + x_{jqk} \le 1$ for all *(i,j)*∈H, *k*∈*K'*, for any lessons (*p*,q)∈*J*∋$T_p = T_q$

(2)

$\sum_{i\in H} \sum_{j\in J \ni G_j = c} x_{ijk} \le 1$ for all classes *c*∈*C*, *k*∈*K'*.

(3)

$x_{ujk} + x_{rjk} \le 1$ for all *j*∈*J*, *k*∈*K'*, *(u,r)*∈*H*∋ $u \neq r$ (4)

$$\sum_{i\in H} \sum_{k\in K} x_{ijk} = L_j \text{ for all subjects } j\in J \quad (5)$$

$$x \in \{0,1\},\ K' = K \setminus S_R \quad (6)$$

Objective function (1) represents all soft constraints and (2)-(6) represent hard constraints of the problem, where;

$H = \{l_1, l_2, \ldots l_n\}$: set of all lessons

$K = \{k_1, k_2, \ldots k_m\}$: set of all timeslots (periods)

$J = \{j_1, j_2 \ldots j_y\}$: set of all subjects groups

$C = \{c_1, c_2, \ldots c_z\}$: set of all classes

L_j = Number of lessons of subject *j*

T_j = A teacher who teaches subject group *j*

G_j = *A* class of a subject group *j*

S_M = A set of early morning timeslots

S_A = A set of late afternoon timeslots

S_R = A set of common religion timeslots

S_P = A set of slots which have restrictions due to other preferences and

$K' = K \setminus S_R$ is a set of all timeslots excluding religion times which are normally fixed.

Adapting Late Acceptance to HSTP

To be able to apply Late Acceptance Heuristic to the HSTP, it is necessary to define a number of configurations, and these are;

Solution data structure: this is defined using a 3-dimesional 0-1 matrix with entries $x_{ijk} \in X$ where x_{ijk}=1 if lesson i of subject group j is slotted in timeslot k and 0 otherwise. A typical Tanzanian high school has ten 40-minutes period timeslots per day, making a total of 50 timeslots per week and they are numbered from 1 on first period of Monday to 50 on the last slot on Friday of the week.

Teacher data structure: usually teachers are pre-assigned to teach in a given subject group (they may belong to a maximum of two subject groups), and are identified through subject groups. An array *T* is defined such that $t_j \in T$ is a teacher in the subject group j.

Neighborhood structure: a neighbor of the current solution $S \in X$ is obtained by swapping randomly selected pair of timeslots of S. The swapping may create infeasibilities which are penalized in the objective function.
Objective function: is a linear combination of all functions in the mathematical formulation including both soft and hard constraints. Hard constraints are penalized higher than soft constraints and a solution is feasible only when all hard constraints are satisfied. The objective function is summarized in equation (7);

$$f(x) = \sum_{i=1}^{8} \lambda_i f(x_i) \qquad (7)$$

Where λ_i= weight given to constraint i.

The functions $f_1(x)$ to $f_8(x)$ represent constraints of the problem, both hard and soft. As shown in the mathematical formulation which is adopted from Mushi (2011) the meanings the functions $f_i(x),\ i \in \{1,2,..8\}$ are as follows:

- $f_1(x)$ spread lessons of a subject group throughout the week, i.e., $f_1(x) = \lambda_1 \sum_{(i_1,i_2)\in H} \sum_{(k_1,k_2)\in K' \ni k_1 \neq k_2} \sum_{j \in J \ni x_{i_1 j k_1} + x_{i_2 j k_2} = 1} \frac{1}{(k_1 - k_2)^2}$.
- $f_2(x)$ is designed to minimize the use of early morning timeslots, that is $f_2(x) = \lambda_2 \sum_{k \in S_M} x_{ijk}$.
- $f_3(x)$ is designed to minimize the allocation to late afternoon timeslots, that is to minimize $f_3(x) = \lambda_3 \sum_{k \in S_A} x_{ijk}$.
- $f_4(x)$ minimizes the allocation of lessons to other non-preference timeslots $f_4(x) = \lambda_3 \sum_{k \in S_P} x_{ijk}$
- $f_5(x)$ is designed to sum up all cases where assigned lessons of a subject are incomplete i.e., $f_5(x) = \lambda_4 \left| \sum_{j \in J} \left(\sum_{i \in H} \sum_{k \in K'} x_{ijk} - L_j \right) \right|$.
- $f_6(x)$ is designed to count the number of times w, a class has been assigned more than one lesson in the same timeslot, i.e., $f_6(x) = \lambda_6 w$,
  where
  $$w = \begin{cases} 1 & \text{if } \sum_{k \in K'} \sum_{i \in H} \sum_{j \in J \ni G_j = c} x_{ijk} > 1 \\ 0 & \text{Otherwise} \end{cases}$$
- $f_7(x)$ counts the number of lesson collisions u, that is, $f_7(x) = \lambda_7 u$.
  Where
  $$u = \begin{cases} 1 & \text{if } \sum_{j \in J, k \in K'} \sum_{(u,r) \ni u \neq r} (x_{ujk} + x_{rjk}) > 1 \\ 0 & \text{Otherwise} \end{cases}$$
- $f_8(x)$ counts the number of teacher collisions v, that is, $f_8(x) = \lambda_8 v$.
  Where
  $$v = \begin{cases} 1 & \text{if } \sum_{(i,j) \in H, k \in K'} \sum_{(p,q) \ni T_q = T_q} (x_{ipk} + x_{jqk}) > 1 \\ 0 & \text{Otherwise} \end{cases}$$

As pointed earlier, a solution is feasible only if there are no violations of the hard constraints, i.e., the sum of objective values of all hard constraints is zero (equations 2 to 6).

Initial solution: is obtained by assigning lessons of a subject group sequentially into the timeslots followed by another subject group until all lessons are assigned to timeslots. This does not guarantee that the solution obtained is feasible but minimizes possible collisions for lessons within the same subject group. The initial solution is therefore infeasible but can be obtained in $O(n)$ time where n is the number of lessons.

Summary of Results

Data from three high schools have been collected with properties shown in Table 1.

Tambaza is the largest with 118 subject groups and 919 lessons followed by Azania with 62 subject groups and 462 lessons.

Table 1: Properties of the problem instances

Problem	Subject groups	Lessons	Timeslots	Teachers
1(Azania)	62	462	45	30
2(Jangwani)	46	332	40	26
3(Tambaza)	118	919	50	50

The algorithm was written in C++ and problems tested on a 3.3GHz processor in Windows platform. Through experience from stakeholders and types of constraints, the values of weights were selected as shown in Table 2 which concurs with previous work in Mushi (2011) for ease of comparisons.

Table 2: Values of weights used in LAH

Weight	Value	Description
λ_1	5	Lesson spread
λ_2	3	Early morning hours
λ_3	3	Late evening hours
λ_4	3	Non-preference timeslots
λ_5	10	Lesson completeness
λ_6	10	Class collision
λ_7	10	Lesson collision
λ_8	20	Teacher collision

After experimentation it was found that the best value of *L* (Late value) is 10. The three problems were tested by varying number of iterations and results are as shown in Table 3. Iteration 0 means initial solution while other iterations show final solution found and time in seconds used to obtain the solution. In both problems, the number of iterations matters in finding a good solution as expected. However, there is a threshold in the number of iterations since all problems converge to a minimum point after a given number of iterations. Further growth in iterations does not yield any significant improvement. For instance, in Tambaza case the problem converges after 4,000 iterations, while Azania converges after 2,000 iterations and Jangwani converges after only 1,000 iterations.

Table 3: Final solutions versus number of Iterations

Iterations	Tambaza		Azania		Jangwani	
	Solution	Time (Sec)	Solution	Time (Sec)	Solution	Time (Sec)
0	2190.67	0	630.670	0	500.67	0
1,000	909.14	97	306.556	29	249.004	17
2,000	669.14	195	303.003	59	249.004	35
3,000	598.14	292	303.003	88	249.004	57
4,000	549.007	390	303.003	118	249.003	69
5,000	549.007	487	303.003	147	249.003	87
10,000	549.005	974	303.003	296	249.003	174
20,000	549.003	1982	303.003	588	249.003	349

Clearly this indicates that the number of iterations varies with the size of the problem where more iterations and therefore more time is needed to search in the solution space for higher sizes of problems. The convergence properties can be visualized in Figure 1.

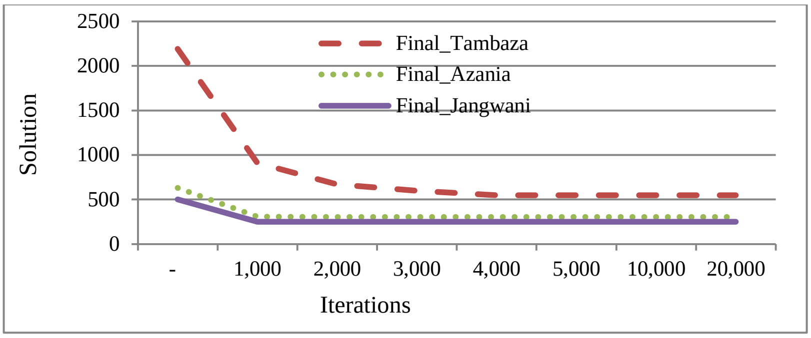

Figure 1: Solution versus number of iterations.

Since convergence varies with size of the problem it is important to fine tune the number of iterations for each problem for better results.

A comparison is made with results obtained from two other heuristics on the same data set; these are Simulated Annealing (SA) (Mushi and Batho 2011) and Non-Linear Great Deluge (NLGD) (Mushi 2011). The results are as shown in Table 4 where the three heuristics give similar solutions. However, Late Acceptance performs better in terms of time in both cases.

Table 4: Comparison of performances of three Heuristics

Algorithm	Tambaza		Azania		Jangwani	
	Solution	Time (Sec)	Solution	Time (Sec)	Solution	Time (Sec)
SA	549.003	649	303.003	220	249.003	127
NLGD	549.006	526	306.002	169	249.005	526
LAH	549.007	390	303.003	59	249.004	17

Late Acceptance performs as good as the other two heuristics but takes shorter to reach the best solution. It is therefore a good heuristic for the high school timetabling problem and better when time factor is a concern.

Table 5 shows a comparison of performances in terms of satisfactions of both hard and soft constraints for LAH as compared to the manually generated timetables and those generated in the previous work by NLGD algorithm. Hard constraints were satisfied in all cases indicating that feasible timetables were generated in all three systems. Satisfaction of soft constraints in NLGD and LAH are very similar, showing insignificant differences but they all performed better than the manual system.

Table 5: Comparison of constraints satisfactions between Manual, NLGD and LAH

Problem	**Property**	**Manual**	**NLGD**	**LAH**
Azania	Solution cost	339.108	303.002	303.003
	Lesson collision	0	0	0
	Teacher Collision	0	0	0
	Completeness violations	0	0	0
	Lesson spread	0.108306	0.00248	0.00329
	Morning time violations	171	153	153
	Evening time violations	168	153	150
Jangwani	Solution cost	311.014	249.005	249.003
	Lesson collision	0	0	0

Problem	Property	Manual	NLGD	LAH
	Teacher collision	160	0	0
	Completeness violations	0	0	0
	Lesson spread	0.01445	0.00489	0.00328
	Morning time violations	154	126	126
	Evening time violations	157	123	123
Tambaza	Solution cost	611.571	549.006	549.003
	Lesson collision	0	0	0
	Teacher collision	0	0	0
	Completeness violations	0	0	0
	Lesson spread	0.57111	0.00571	0.00271
	Morning time violations	372	276	276
	Evening time violations	239	273	273

Conclusion and Further Research Directions

This paper focused on implementing the Late Acceptance Heuristic for the High School Timetabling Problem and compare results with previous works on the same data set through Simulated Annealing and Non-Linear Great Deluge algorithms. Despite of its simplicity with single parameter choice, Late Acceptance has shown to perform well with the same solutions as the other algorithms in both cases. However, Late Acceptance has shown to perform better in terms of time and therefore more useful when time factor is concerned. The value of a single parameter *L* is dependent on the size of the problem for better convergence and therefore fine tuning is necessary for better results.

So far the data set used comes from three high schools; further extension by including data from more schools in the country may give better insights into the structure of the problem and give way for further exploration. Furthermore, the challenge of timetabling is not only in high schools, ordinary level secondary schools have more subjects, since each student is required to take at least seven subjects. Studies on these school timetables which are currently generated manually are areas for exploration. The data set has been tested on only three algorithms so far; more global heuristics algorithms can be implemented for further studies.

References

Abramson D 1991 Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Manage. Sci.* 37(1): 98-113.

Birbas T, Daskalaki S and Housos E 1997 Timetabling for Greek High Schools. *J. Operat. Res. Soc.* 48(12): 1191-1200.

Burke E, de Werra D and Kingston J 2013 Applications to timetabling. In Gross et al (Ed): *Handbook of Graph Theory*, Chapman & Hall/CRC, New York.

Burke E, McCollum B, Meisels A, Petrovic A and Qu R 2007 A Graph-based hyper-heuristic for educational timetabling problems. *Eur. J. Operat. Res.* 176: 177-192.

Burke E and Baykov Y 2017 The late acceptance hill-climbing heuristic. *Eur. J. Operat. Res.* 258(1): 70-78.

Colorni A, Dorigo M and Maniezzo V 1998 Metaheuristics for high-school timetabling. *Comput. Optim. Applicat. J.* 9(3): 275-298.

Demirovic E and Stuckey P 2018 Constraint programming for high school timetabling: a scheduling-based model with hot starts. In Hoeve WJ (Ed): *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26–29, 2018, Proceedings, Springer International Publishing.

Dorneles AP, de Araújo OCB and Buriol LS 2014 A fix-and-optimize heuristic for the high school timetabling problem. *Comput. Operat. Res.* 52(2018): 29-38.

Even S, Itai A and Shamir A 1976 On the complexity of timetable and multi-commodity flow problems. *SIAM J. Comput.* 5: 691-703.

Fernandes C, Caldeira J and da Rosa A 2002 Infected Genes Evolutionary Algorithm for School Timetabling. WSES International Conference, 2002, Interlaken.

Filho GR and Lorena LAN 2001 A Constructive Evolutionary Approach to School Timetabling. In *Applications of evolutionary computing*, Evo Workshops 2001: EvoCOP, EvoFlight, EvoIASP, EvoLearn, and EvoSTIM, Como, Italy, April 18–20, 2001. Proceedings (130-139).

Fonseca GH, Santos HG and Carrano EG 2016 Late acceptance hill-climbing for high school timetabling. *J. Schedul.* 19: 453-465.

Glover F, Laguna M 1998 Tabu search. In: Du DZ, Pardalos PM (Eds) *Handbook of Combinatorial Optimization* Springer Boston MA.

Henderson D, Jacobson SH, Johnson AW 2003 The Theory and Practice of Simulated Annealing. In: Glover F, Kochenberger GA (Eds) *Handbook of Metaheuristics*, International Series in Operations Research & Management Science 57 Springer Boston MA.

Katsaragakis IV, Tassopoulos IX and Beligiannis GN 2015 A Comparative study of modern heuristics on the school timetabling problem. *Algorithms* 8: 723-742.

Kunthavai A and Rajithaa R 2018 Auto-scheduler for school timetable using bacterial foraging optimization algorithm. *Int. J. Innov. Engin. Technol. (IJIET)* 10(1): 38-41.

Marte M 2002 *Models and algorithms for school timetabling–a constraint-programming approach*. PhD thesis, Faculty of Mathematics, Informatics and Statistics, Ludwig-Maximilians-Universität München.

Marwa Y and Mushi AR 2013 Late acceptance heuristic for University course timetabling problem. *Int. J. Adv. Res. Comput. Sci.* 4(2): 88-92.

Matias S and Riis TJ 2012 High School Timetabling: Modeling and solving a large number of cases in Denmark. Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012).

Muller T 2005 *Constraint-based timetabling*. PhD thesis, Charles University in Prague, Faculty of Mathematics and Physics.

Mushi AR 2011 Non-linear great Deluge algorithm for Tanzanian high schools timetabling. *Int. J. Adv. Res. Comput. Sci.* 2(4): 584-590.

Mushi AR and Batho P 2011 A simulated annealing heuristic for Tanzanian high schools timetabling. *GJCAT* 1(3): 425-431.

Ozcan E, Bykov Y, Birben M and Burke EK 2009 Examination Timetabling Using Late Acceptance Hyper-heuristics. *IEEE Congress on Evolutionary Computation* 997-1004.

Post G, Gaspero LD, Kingston JH, McCollum B and Schaerf A 2013 The Third International Timetabling Competition. *Pract. Theory Automat. Timetab. (PATAT 2012)*, 29-31 August 2012, Son, Norway

Ribic R, Turcinhozic and Muratovic-Ribic A 2015 Modeling constraints in school timetabling using integer linear programming. In 2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT) pp. 1–6.

Schaerf A 1996 Tabu search techniques for large high school timetabling problems. *IEEE Trans. Syst. Man Cybernet. - Part A Syst. Hum.* 29(4):368 – 377.

Schaerf A 1999 Local search techniques for large high-school timetabling problems. *IEEE Trans. Syst. Man Cybernet., Part A: Syst. Hum.* 29(4): 368-377.

Simon K, Matias S and Riis STJ 2014 Integer Programming for the Generalized (High) School Timetabling Problem. Proceedings of the 10th International Conference on the Practice and Theory of Automated Timetabling.

Souza AL, JB Chagas, Penna PH and Souza JF 2019 A late acceptance hill-climbing heuristic algorithm for the double vehicle routing problem with multiple stacks and heterogeneous demand. In: Abraham A., Cherukuri A., Melin P., Gandhi N. (Eds) *Intelligent Systems Design and Applications. ISDA 2018 Advances in Intelligent Systems and Computing*, 940 Springer, Cham.

Valouxis C, Gogos C, Alefragis P and Housos E 2012 Decomposing the High School Timetable Problem. Practice and Theory of Automated Timetabling (PATAT 2012), 29-31 August 2012, Son, Norway.

Wilke P and Ostler J 2008 Solving the School Time Tabling Problem using Tabu Search, Simulated Annealing, Genetic and Branch & bound Algorithms. In the proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008), Montreal.

Willemen RJ 2002 *School timetable constructions: algorithms and complexity*. PhD thesis, Technische Universiteit Eindhoven, Netherlands.

Zhang D, Liu Y, M'Hallah R and Leung SCH 2010 A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problem. *Eur. J. Operat. Res.* 203(2010): 550-558.