# Hybrid Erosion And Deposition Metaheuristic University Teaching Timetabling Framework

Stanley Murairwa
murairwas@africau.edu
Africa University, Box 1320, Mutare, Zimbabwe

Abstract
The research developed a Hybrid Erosion And Deposition Metaheuristic University Teaching Timetabling Algorithm (HEADMUTTA). The hybrid erosion and deposition metaheuristic university teaching timetabling algorithm constructs a university teaching timetable with very few iterations. The Hybrid Erosion And Deposition (HEAD) metaheuristic university teaching timetabling algorithm adopts its behaviour from the HEAD metaheuristic with some concepts adapted from Tabu Search, Simulated Annealing, and Ant Colony metaheuristics. The HEADMUTTA constructs a draft university teaching timetable and further improves it by searching for the best feasible solution that satisfies the predetermined soft and hard rules or constraints. The research also proposed a Hybrid Erosion and Deposition Metaheuristic University Teaching Timetabling framework for implementation. The results indicate that the use of hybrid metaheuristics to solve university teaching timetabling problems improves the quality of the produced university teaching timetables. The HEAD metaheuristic algorithm has a unique feature that allows it to further improve the draft university teaching timetable. Further improvements in the framework may reduce the complexity of NP-hard combinatorial teaching timetabling problems.

**Keywords**: Heuristics, Hybrid Erosion And Deposition, Combinatorial Optimisation, Constraints, Metaheuristics, University Teaching Timetable
**JEL:** C6, C61
**Research classification:** Research paper

## 1. Introduction

The university teaching timetabling problems are special types of timetabling problems that handle the scheduling of events to critical limited resources such as lecturers, venues, days, and time slots. The teaching timetabling problems have soft and hard constraints or rules that should be satisfied to generate a feasible teaching timetable. The timetable generation process is a time-consuming, laborious, and repetitive activity for university administrators (Muklason, Irianti, & Marom, 2019; Mittal, Doshi, Sunasra, & Nagpure, 2015). Most universities in Zimbabwe are facing challenges in developing teaching timetables. Generally, coming up with a non-conflicting event timetable is proving to be a mammoth task for most education institution administrators in Zimbabwe. Currently, university administrators use manual methods to construct teaching timetables and this results in conflicting teaching timetables that affect the quality of services offered to stakeholders. A study found that 11% of the learning institutions were struggling to construct feasible teaching timetables during the Covid-19 pandemic (PhephauFUNDE, 2020). The teaching timetabling problems are NP-hard or NP-complete combinatorial optimisation problems (Murairwa, 2020; 2010; de Werra, Asratian, & Durand, 2002) that require the application of heuristics, especially metaheuristics and their hybrids to develop perfect teaching timetabling algorithms.

The third-generation heuristics such as the metaheuristic hybrids are key to the successful development of event teaching timetabling algorithms because they are capable of handling NP-hard or NP-complete timetabling problems. Murairwa (2020; 2010) proposed a third-generation heuristic that was named the Hybrid Erosion And Deposition (HEAD) metaheuristic that imitates the erosion and deposition processes during the development of a mature river. The HEAD metaheuristic can handle the university teaching timetabling problems because it adapted some imperative concepts from the Tabu Search (Glover, 1986), Ant Colony (Deneubourg, Aron, Goss, & Pasteels, 1990), and Simulated Annealing (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953), which most researchers employed successfully to solve the event timetabling problems. According to Muklason, Irianti, and Marom (2019), manual teaching timetabling is still creating recurring problems when generating university teaching timetables. The general university teaching timetabling problem is presented in Figure 1.

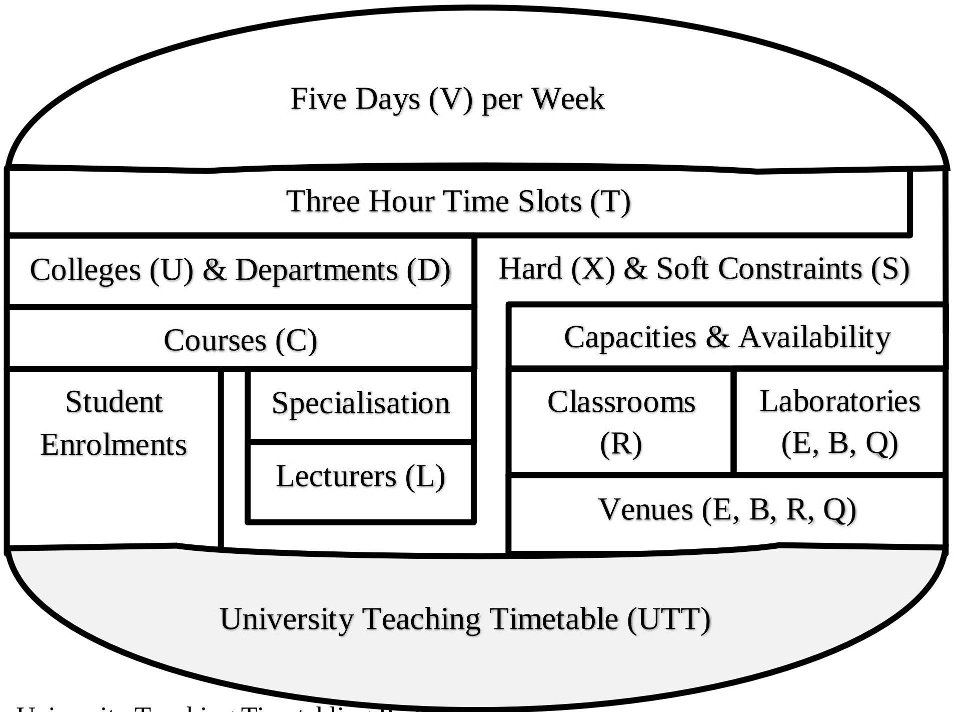



**Figure 1:** University Teaching Timetabling Problem

Figure 1 presents the university teaching timetabling problem and its vital stochastic and deterministic variables. The increasing university new programmes and student enrolments against the standard time slots and fixed number of days per week demand high-quality service, flexibility, and dynamic university teaching timetabling systems. A university teaching timetabling problem demands the consideration of the lecturers' areas of specialisation (courses taught) and preferences, student course enrolments, venues' capacities and availability (classrooms for non-practical courses and laboratories (computer and medical laboratories, and moot court for practical courses), time slots, and the number of days per week. The main challenge is to allocate all courses (stochastic variable) to the available venues, timeslots, and days per week (deterministic variables) without creating conflicts. The availability and flexibility of teaching resources are limited despite the increasing demand (Oude Vrielink, Jansen, Hans, & van Hillegersberg, 2019) for evolving university education. The lecturers include sabbatical, visiting, graduate teaching assistant, volunteering (Murairwa, 2015; 2014), and full-time and part-time teaching staff members. Cooper (2018) disclosed that some teaching timetabling systems are dedicated while some are not and many institutions suffer from instruments that have been surpassed many years ago in timetabling technology innovation. Many universities in Africa and Zimbabwe in particular are still using manual teaching timetabling systems. The timetabling problem also considers the soft and hard rules or constraints. These soft and hard rules or constraints affect the whole university teaching timetabling problem as shown in Figure 1. The objective of this research was to develop the HEAD metaheuristic university teaching timetabling algorithm (HEADMUTTA) for solving the university teaching timetabling problem.

## 2. Literature Review

Heuristics are general search and optimisation algorithms that are used to solve NP-hard or NP-complete real-life problems (Murairwa, 2020; 2010) such as university teaching timetabling problems. Elloumi, Kamoun, Jarboui, and Dammak (2014), and de Werra, Asratian, and Durand (2002) proved the event timetabling problems to be NP-hard problems. The most powerful and dominant search heuristics are metaheuristics (Murairwa, 2022). Teaching timetabling is a challenge that is encountered in most academic institutions (Abdellahi & Eledum, 2017). The most popular metaheuristics for solving timetabling events are the Genetic Algorithm (Mittal, Doshi, Sunasra, & Nagpure, 2015) and, Tabu Search (Muklason, Irianti, & Marom, 2019; Arntzen & Løkketangen, 2005). Some researchers such as Awad, Al-kubaisi, and Mahmood (2022) and, Lu and Hao (2008) apply the Adaptive Tabu Search to generate feasible timetables. The teaching timetabling problem involves the distribution of resources to avoid clashes between or among them (Abdellahi & Eledum, 2017). The institutions' teaching timetabling instruments should be in line with the current teaching timetabling technology innovation (Cooper, 2018). This revelation indicates that the current teaching timetabling tools are archaic to handle the current complexity of allocating the available timetable variables that include part-time and full-time lecturers, available and preferred time slots, hybrid-flexible (hyflex) teaching methods, improved technologies, and ever-changing disruptive, volatile, uncertain, complex, and ambiguous (DVUCA) (Murairwa, 2022) learning environment.

The hard constraints are compulsory while the soft constraints are optional (Alencar, Dantas do Nascimento, Soares, & Longo, 2019). When the timetabling problem is NP-complete, it involves lecturers who are teaching at least three groups of courses (de Werra, Asratian, & Durand, 2002). A survey conducted found that 11% of learning institutions were struggling to come up with teaching timetables that reflect the true reality on the ground since the outbreak of the Covid-19 pandemic (PhephauFUNDE, 2020). The Covid-19 pandemic brought in a social distancing challenge that increases the complexity of solving the teaching timetabling problem. Hambali, Olasupo, and Dalhatu (2020) combined the Simulated Annealing and Genetic Algorithm to form a metaheuristic hybrid named Heuristic Approach that was applied to solve the teaching timetabling problem. The teaching timetabling problem is a search problem (Al-Jarrah, Al-Sawalqah, & Al-Hamdan, 2017) that can only be handled by metaheuristics and their hybrids. The teaching resources are limited to the extent that classrooms are shared many times but the demand for availability and flexibility is increasing (Oude Vrielink, Jansen, Hans, & van Hillegersberg, 2019). These conditions make the teaching timetabling problem more complex and byzantine to solve.

Murairwa (2020; 2010) proposed a Hybrid Erosion And Deposition (HEAD) metaheuristic that adapts some concepts from Tabu Search (Glover, 1986), Simulated Annealing (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953), and Ant Colony (Deneubourg, Aron, Goss, & Pasteels, 1990) metaheuristics. The HEAD metaheuristic has a feature for further enhancing the draft university teaching timetable (UTT). The feature is very useful in constructing the final feasible UTT. A mature river to the Ocean is a successful allocation of the course to the lecturer, time slot, venue, and day with all the soft and hard rules or constraints satisfied. However, other heuristics such as dual bi-directional (multi-start) heuristics (Murairwa, 2021) can also handle teaching timetabling NP-hard problems because of the search mechanism used by these algorithms.

## 3. Methodology

The university teaching timetabling system requirements and notations are

- Colleges (U) with u number of colleges;
- Departments (D) with d number of departments;
- Courses (C) with n number of courses. Each course must have a student enrolment number and unique identifiers of degree and semester level. Each course must be identified as either a practical or non-practical course. A practical course must also be identified by its practical discipline area. A course cannot be offered more than once per week. Lecture attendance is mandatory and that plays an important role in the development of the university teaching timetabling system;
- Lecturers (L) with l number of lecturers. A lecturer must teach all courses allocated per week. A lecturer can repeat as long he or she is not scheduled at the same time slot on the same day;
- Venues (R, Q, B, E) must be identified as non-practical or practical venues. Classrooms (R) with r number of classrooms. Each classroom must have a capacity. The classrooms are for non-practical courses from all colleges and departments. Computer laboratories (Q) with q number of computer laboratories. Each computer laboratory must have a capacity. The computer laboratories are for computer-related practical courses. Medical laboratories (B) with b number of medical laboratories. The medical laboratories are for medical-related practical courses. Law practical (E) with e number of moot courts and law clinics. The law practical courses are scheduled in the moot court or law clinic;
- Days (V) with v number of days per week (Monday, Tuesday, Wednesday, Thursday, and Friday). A decision can also be made on whether to include Saturdays and Sundays to decongest the traditional five working days. National holiday days are excluded. Each day has several available three-hour time slots excluding the 1 – 2 pm lunch hour;
- Time slots (T) with t number of days and h number of hours. Under normal circumstances, lectures commence at 9 am and end at 5 pm daily. However, the 8 am up to 8 pm could be considered for the smooth implementation of the university teaching timetabling system. Each time slot must be three hours excluding the 1 – 2 pm lunch hour every day;
- Degree levels (M) with m number of degree levels. Each course code must identify the degree level of the course. This may be taken care of by the course code formatting approach used;

- Soft and Hard rules or constraints. Soft constraints (S) with s number of soft constraints and hard constraints (X) with x number of hard constraints. Each rule or constraint has a penalty value for breaching it.

The edge, which is the mature river, is an allocation of the course to the venue, day, time slot, and lecturer with all the hard and soft rules or constraints satisfied. The search duration ($SD$) in seconds for c cycles of the HEAD metaheuristic enhancement process (Murairwa, 2020; 2010) is computed with

$$SD = \sum_{i=1}^{c} (\lambda + \gamma_1 + \gamma_2)_i = \sum_{i=1}^{c} \left(\frac{T_n}{\beta} + \gamma_1 + \gamma_2\right)_i \quad (1)$$

The search rate function is computed with

$$\beta = \left(\frac{\rho^2 g^2 \theta S^2 K}{3w\eta}\right) h^3 \quad (2)$$

The best parameters for the HEAD metaheuristic to construct the feasible UTT are the erodibility factor (K = 0.5), gradient angle (θ = 0.7071), flow power to erode ($\delta = 8.2570gm^{-2}$), erosion rate ($\beta = 1.51gs^{-1}$), tolerable soil loss level ($T_n$ = 100kg), $T_0$ = 0, upslope (ϑ = $45^0$), depth of the river (h = 1m), length/slope (S = 0.366m), density ($\rho = 0.9982gm^{-3}$), viscosity (η = 1.002mPas.sec), gravity ($g = 9.81ms^{-2}$), cover-management factor (C = 1), flow rate ($\alpha = 2.303m^3s^{-1}$) and support practice factor (velocity/speed/friction/carried particles) (P = 1) (Murairwa, 2020; 2010). The HEAD metaheuristic university teaching timetabling algorithm (HEADMUTTA) terminates when either β or $T_n$ is reduced to $T_0$ = 0. The HEADMUTTA search concept is shown in Figure 2.

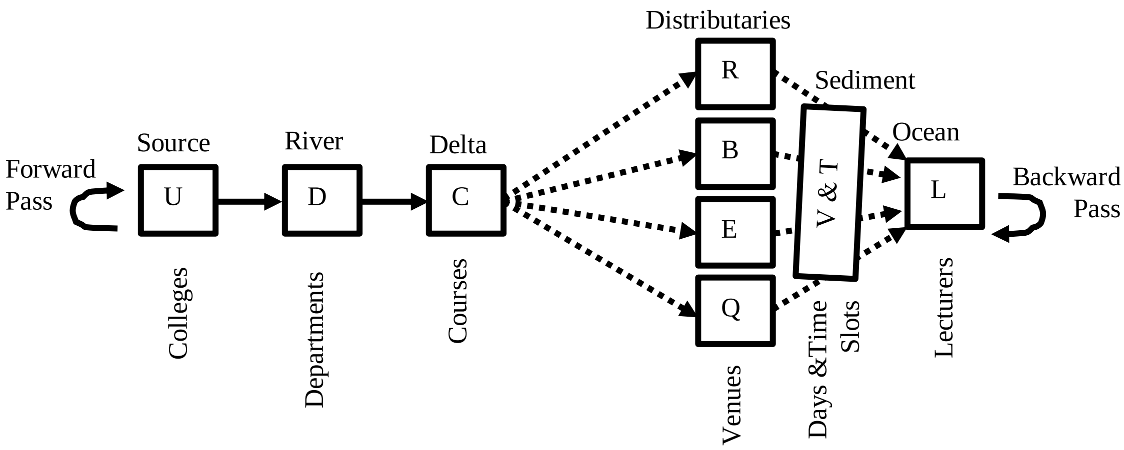



**Figure 2:** The HEAD Metaheuristic Search Concept

Figure 2 shows the HEAD metaheuristic search concept for scheduling the courses to the lecturers, venues, time slots, and days during the construction of the feasible university teaching timetable (UTT). The HEADMUTTA selects the best combination of the variables of the university teaching timetabling problem to schedule a course. The HEADMUTTA constructs a tour through forward and backward passes. The algorithm's forward pass starts from the source (colleges) to the Ocean (lecturers) and the reverse process is the backward pass to construct a round tour as shown in Figure 2. There are predetermined soft and hard rules or constraints that are applied during the construction of the feasible UTT. However, there is a penalty for breaching each of the rules or constraints. The conditions that are important for the HEADMUTTA to successfully construct a feasible UTT are

- A student must attend a lecture per time slot per day.

- The capacity of the venue (R or Q or B or E) must be greater than or equal to the number of students enrolled in the course being scheduled. The practical courses are scheduled in the discipline-related practical venues {computer laboratories (Q) or medical laboratories (B) or moot court (E)}. The non-practical courses across all colleges are scheduled in the classrooms (R) that meet the timetabling criterion.
- A lecturer per venue per time slot per day. A lecturer can have more than one lecture per day with different courses and time slots.
- A course is scheduled in a venue (R or Q or B or E) if its enrolment is equal to or greater than three–quarters of the venue's capacity but less than the capacity of the venue; $\frac{3}{4}\,Venue\ Capacity \leq Course\ enrolment \leq Venue\ Capacity$.
- A course is assigned to a venue (R or Q or B or E) at a time slot that is available and meets the capacity requirement criterion. A course is offered once per week. A course is offered for at least fifteen weeks per semester.
- University-wide courses should be available to all students. The university-wide courses cannot be scheduled at the same time slot as courses of the same degree level.
- All courses (C) registered must be scheduled per week. Thus, each course must be allocated to a venue, lecturer, and time slot every week.
- Lectures start at 9 am and end at 5 pm excluding the 1 – 2 pm lunch period. Nevertheless, this condition can be adjusted according to the university's operating hours per day.

### 3.1 Tabu moves

The tabu moves are the NP-hard or NP-complete or simply hard rules or constraints. The tabu (forbidden or restricted) moves (Murairwa, 2020; 2010) during the construction of the UTT are

- A course must be assigned to a venue if and only if its enrolment is less than or equal to the venue's capacity but greater than or equal to three-quarters of the venue's capacity i.e. $\frac{3}{4}\,Venue\ Capacity \leq Course\ enrolment \leq Venue\ Capacity$.
- The practical courses must be assigned to practical venues and the non-practical to classrooms.
- A selected course or venue must not be selected again for the same three-hour time slot per day despite meeting other timetabling criteria. This is a selection without replacement (Murairwa, 2021; 2019) and the concept creates a pool of tabu moves. The concept reduces duplications and conflicts of the timetabling problem variables.
- The national holidays must be respected by not timetabling courses. The national holidays must be identified and classified as tabu moves.

### 3.2 Soft Rules or Constraints

- A student must have at most a lecture for the same three-hour time slot per day. The HEADMUTTA does not consider repeated courses.
- A lecturer must take only one lecture per three-hour time slot per day. Thus, a lecturer can take morning and afternoon time slots per day for different courses.
- The capacities of the venues must be respected.
- A lecturer must teach all allocated courses per week.
- The lectures start at 9 am and end at 5 pm. The lunch hour (1 – 2 pm) should be respected.
- Lecturers take courses from all colleges and departments in their specialisation areas.
- The university teaching timetable (UTT) must eventually be generated.

### 3.3 HEADMUTTA Approach

The HEADMUTTA starts by creating rivers through erosion and deposition processes from the colleges (U) through the departments (D), courses (C), venues (R, B, E or Q), days (V), time slots (T) to the lecturers (L) and return to the starting point to complete a tour as presented in Figure 2 above. The best river that meets the soft and hard rules or constraints of the teaching timetabling problem is selected as a successful course allocation. The scheduled course and time slot are stored in the update manager memory as tabu moves in the next process of creating the rivers. The process continues until all the courses are allocated the venues and time slots to produce the initial feasible university teaching timetable (UTT). The update manager memory stores all tabu moves for reference purposes during the construction of the feasible UTT. The general structure of the university teaching timetabling system adapted from Mittal, Doshi, Sunasra, and Nagpure (2015) is presented in Figure 3.

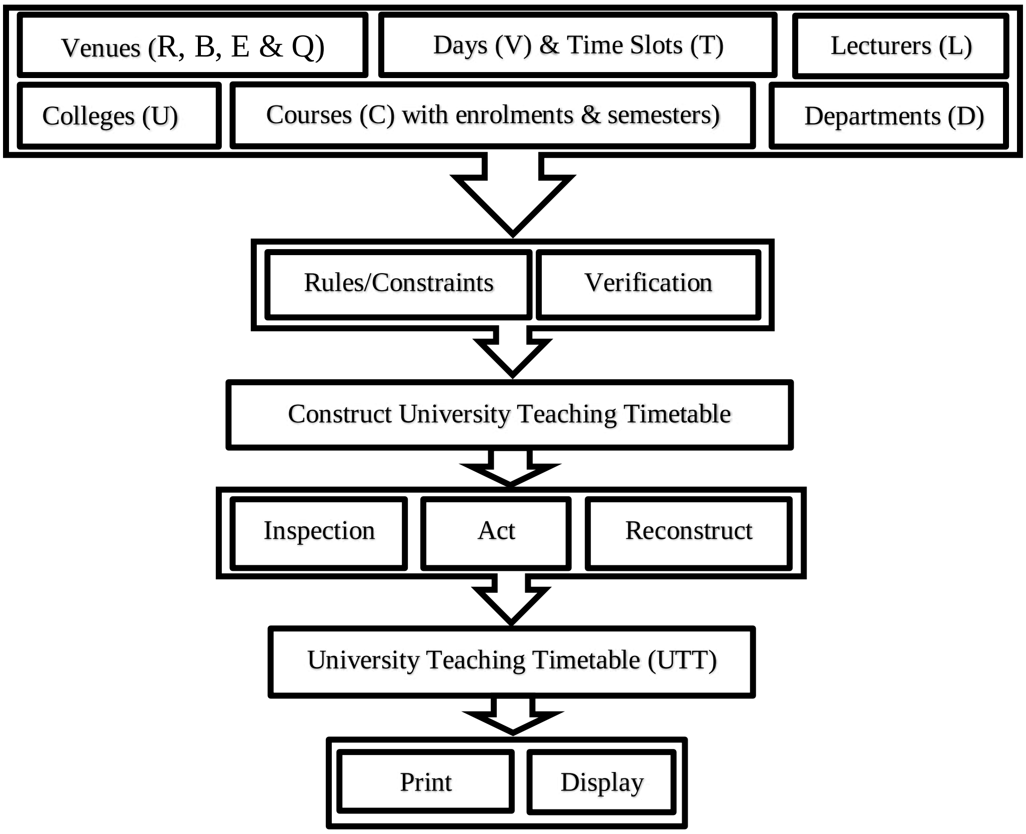



**Figure 3:** HEAD Metaheuristic University Teaching Timetabling Structure

Figure 3 presents the structure of the HEAD metaheuristic university teaching timetabling algorithm (HEADMUTTA). The HEAD metaheuristic university teaching timetabling structure has three phases, namely, the Input, Construction, and Output (ICO) phases. The Input phase loads all the requirements of the university teaching timetabling algorithm. The inputs include the names of the colleges, departments, courses with student enrolments and degree levels, lecturers, days, time slots, and venues (classrooms, computer laboratories, moot court, and medical laboratories) with capacities and locations. The Construction phase allocates all courses to the lecturers, days, and time slots and checks whether all the predetermined soft and hard rules or constraints are satisfied. The Construction phase is divided into two sub-phases and these are the construction of the draft and final university teaching timetables. Therefore, the Construction phase develops and improves the draft feasible university teaching timetable into the final feasible UTT for users through the HEAD Metaheuristic University Teaching Timetable Enhancement (HEADMUTTE) process. The Output phase displays and prints the draft and the final university teaching timetables for the users. The HEADMUTTE process adapted from Murairwa (2020; 2010) is presented in Figure 4.

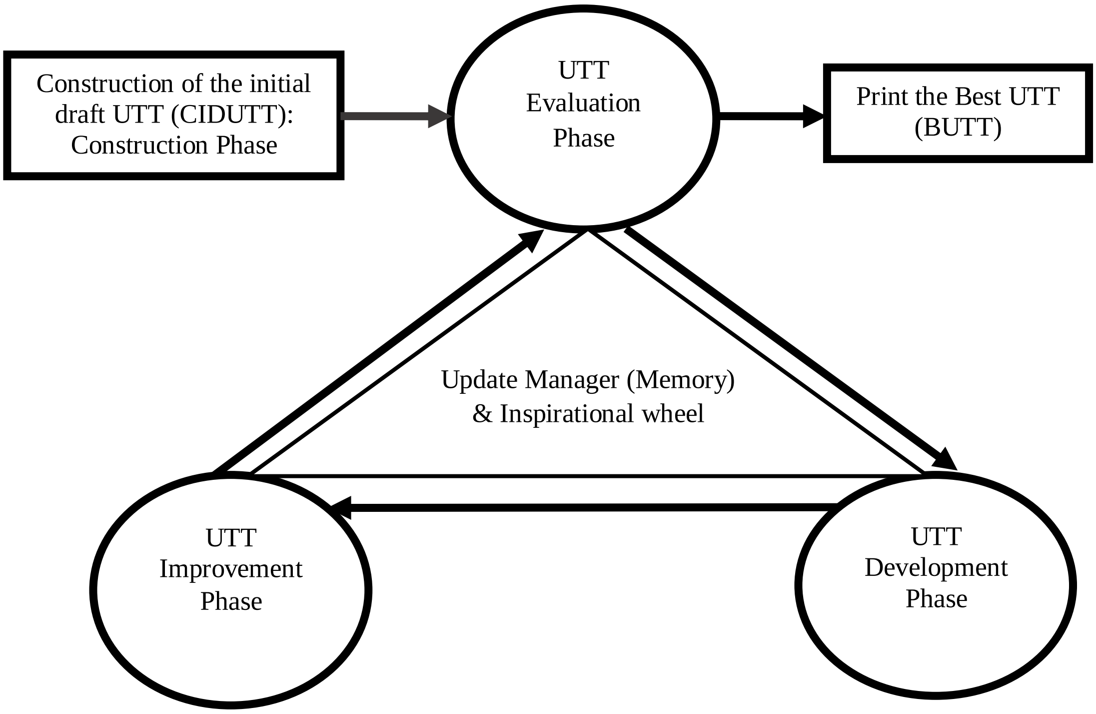


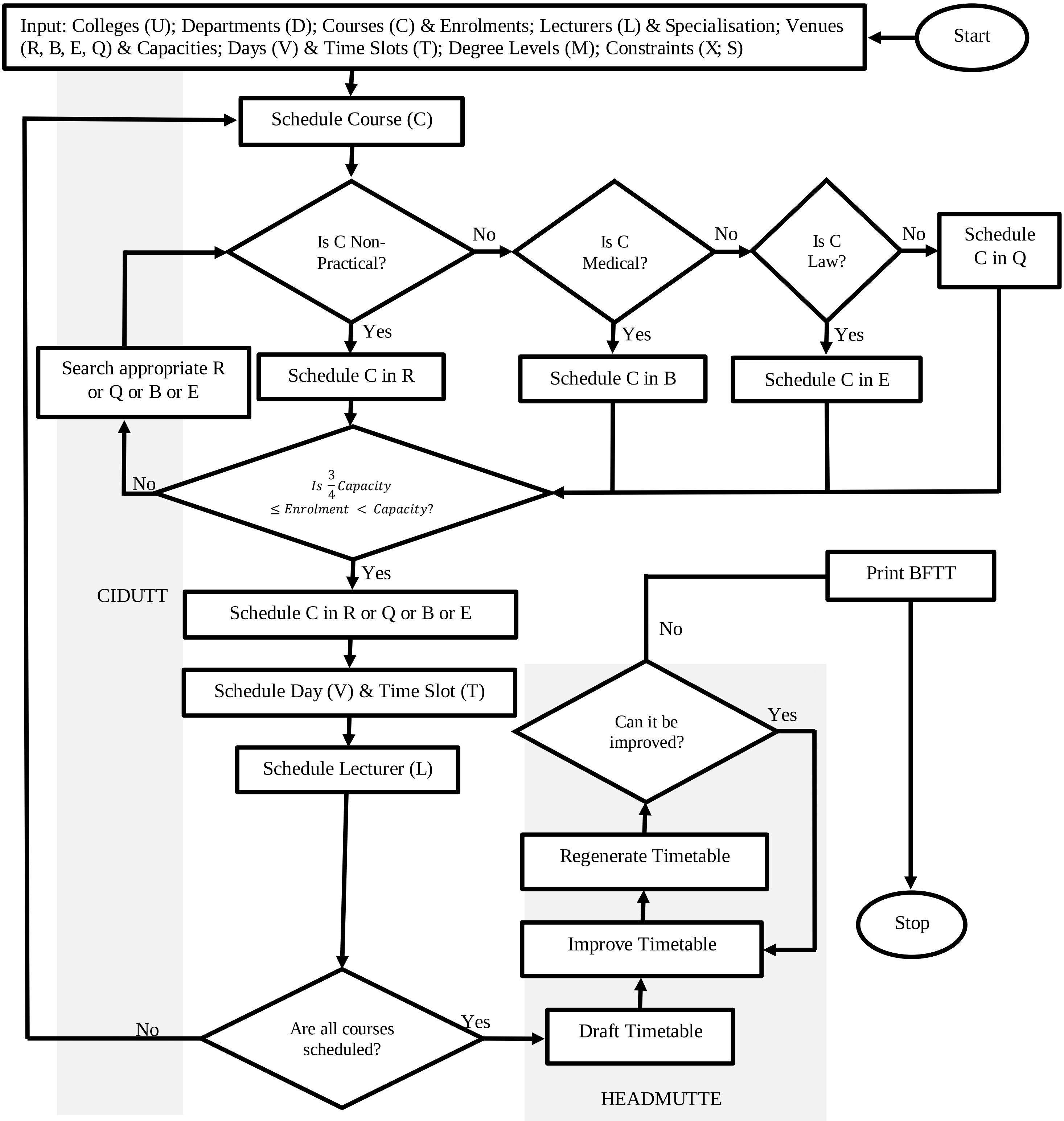

**Figure 4:** HEAD Metaheuristic University Teaching Timetable Enhancement Process

Figure 4 presents the process of improving the quality of the draft feasible university teaching timetable (UTT) produced by the Construction phase. The update manager has a memory that stores the selected teaching timetabling problem variables as tabu moves. The HEAD metaheuristic university teaching timetable enhancement process (HEADMUTTEP) is the quality assurance engine of the HEAD metaheuristic university teaching timetabling algorithm. The algorithm generates a draft UTT in the Construction phase. The quality assurance engine improves the draft UTT by comparing the capacities of the venues to the class enrolments to make sure that high course enrolments are allocated to venues with high capacities. The Evaluation phase checks whether improvements to the feasible UTT have been done. The phase also checks whether all the predetermined soft and hard rules or constraints have been met and switches off (or terminates) the enhancement process. The Development phase receives the draft (or initial or feasible) UTT from the Evaluation phase. It develops another feasible UTT and compares it with the draft UTT it received from the Construction phase. A better feasible UTT as guided by the objective function is forwarded to the Improvement phase for further improvement. The Improvement stage employs the gradient change ($\Delta T_n$) method in the erosion and deposition processes to improve the feasible UTT it received from the Development stage.

## 4. Results

### 4.1 University Teaching Timetabling Framework

The HEAD metaheuristic concept, the university teaching timetabling structure, and the metaheuristic university teaching timetabling enhancement process (MUTTEP) are integrated to form the HEAD metaheuristic university teaching timetabling framework. The HEAD metaheuristic university teaching timetabling framework is presented in Figure 5.




• Construction of Initial Draft University Teaching Timetable (CIDUTT)
• HEAD Metaheuristic University Teaching Timetable Enhancement Process (HEADMUTTEP)

**Figure 5:** HEAD Metaheuristic University Teaching Timetabling Framework

Figure 5 shows the ICO phases of the proposed HEAD metaheuristic university teaching timetabling framework. In the first phase, the HEADMUTTA loads the required inputs. The inputs of the university teaching timetabling algorithm include the colleges (U), departments (D), courses (C) (name and course code) with their enrolments and degree levels (M), venues {R, Q, B, and E} with their capacities, and locations and lecturers (L) with their areas of specialisation (allocated courses) and days (V) with time slots (T). The venue selection loop can be adjusted according to the number of practical courses that are offered by the learning institution. The second phase involves the construction of the initial draft university teaching timetable (CIDUTT). The HEADMUTTA selects the college, department, and course and decides on the venue (R or Q or B, or E) to schedule the selected combination as guided by the type of course, whether practical or non-practical. In this phase, all the soft and hard rules or constraints must be satisfied to produce an initial draft UTT. All the courses (C) must be allocated to days (V), time slots (T), and lecturers (L) while the soft and hard constraints (X and S) are satisfied. The third phase is the output. In this phase the HEADMUTTA displays and prints the final feasible UTT for the users.

### 4.2 HEADMUTTA Pseudo Code

```
Procedure GenerateInitialDraftFeasibleSolution(IDFS)
FeasibleSolution = 0;
Read Input();
Function Allocate Course to time slot();
Select C;
     Check
          if C is Non-practical;
          Select R;
else
if C is Medical practical;
          Select B;
else
C is Law practical;
          Select E;
          else
          Select Q;
          endif
          endif
     Check
if 3/4 Capacity≤ Enrolment ≤ Capacity;
          Schedule C;
          else
          Select a new venue (R or Q or B or E);
          endif
          Schedule V;
          Schedule T;
          Schedule L;
   while C≤ n
          Schedule the next C;
   endwhile
   Solution = CP(IDFS_CP)
endprocedure
Function improve IDFS_CP
Procedure HEADMUTTA
Best_FeasibleSolution = 0;
Read_Input();
Function Allocate Course to time slot();
For k = 1, 2, ......, MAX_ITERATIONS do
Solution = EP(FS_EP);
Solution = DP(FS_DP);
Solution = IP(FS_IP);
If(FS_IP is better than the best_feasibeSolution) then
UpdateFeasibleSolution (FS_IP, Best_FeasibleSolution);
endif
endfor
end improveInitialDraftFeasibleSolution();
return (Best_FeasibleSolution)
print Best_FeasibleSolution
end HEADMUTTA
```

## 5. Conclusion and Recommendations

The hybrid erosion and deposition metaheuristic was implemented to solve the university teaching timetabling problem. The variables of the university teaching timetabling problem are the details of the colleges, departments, courses (names and course codes) with their enrolments and degree levels, venues (practical and non-practical) with their capacities and locations, and lecturers with their areas of specialisation (allocated courses) and days with time slots. The proposed hybrid erosion and deposition metaheuristic university teaching timetabling framework develops and improves the feasible draft and final teaching timetable respectively in three phases, namely, Input, Construction, and Output phases. Thus, the proposed Hybrid Erosion And Deposition Metaheuristics University Teaching Timetabling Algorithm (HEADMUTTA) has a feature for improving the draft feasible university teaching timetable. The best feasible university teaching timetable is produced as the final deliverable of the HEADMUTTA. There is a need to implement the proposed hybrid erosion and deposition metaheuristic university teaching timetabling framework to construct the university teaching timetables. Further improvement of the framework can be done to perfect the development.

### Areas for further studies

The researchers can improve the developed university teaching timetabling framework by introducing different hybrid metaheuristics in solving the university teaching timetabling problem. The improvement of the university teaching timetabling involves further fine-tuning the enhancement process for further improving the draft feasible university teaching timetable produced in the Construction phase.

### Acknowledgement

The researcher wants to thank Dr Agrippah Kandiero and Mr Lorence Dhlakama for their useful contributions during the writing of this article.

### References

Abdellahi, M., & Eledum, H. (2017). The University Timetabling Problem: Modeling and Solution Using Binary Integer Programming with Penalty Functions. International Journal of Applied Mathematics and Statistics, 56(6), 164 - 178.

Alencar, W., Dantas do Nascimento, H., Soares, F., & Longo, H. (2019). Visualization Methods for Educational Timetabling Problems: A Systematic Review of Literature. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019} (pp. 275-281). Lola: SCITEPRESS – Science and Technology Publications. doi:10.5220/0007375802750281

Al-Jarrah, A. M., Al-Sawalqah, A. A., & Al-Hamdan, F. S. (2017). Developing A Course Timetable System for Academic Departments Using Genetic Algorithm. Jordanian Journal of Computers and Information Technology, 25 - 36. doi:10.5455/jjcit.71-1465735040

Arntzen, H., & Løkketangen, A. (2005). A Tabu Search Heuristic for a University Timetabling Problem. In T. N. Ibaraki, Metaheuristics: Progress as Real Problem Solvers (Vol. 32). Boston: Springer. doi:doi.org/10.1007/0-387-25383-1_3

Awad, F. H., Al-kubaisi, A., & Mahmood, M. (2022). Large-scale timetabling problems with adaptive tabu search. Journal of Intelligent Systems, 31(1), 68 - 176. doi:10.1515/jisys-2022-0003

Cooper, C. (2018, August 1). The Challenges Of Timetabling Part-Time Staff. Retrieved October 7, 2022, from Education Technology Solutions: https://educationtechnologysolutions.com/2018/08/challenges-timetabling-part-time-staff/

de Werra, D., Asratian, A. S., & Durand, S. (2002). Complexity of some special types of timetabling problems. Journal of Scheduling, 2, 171-183. doi:10.1002/jos.97

Deneubourg, J.-L., Aron, S., Goss, S., & Pasteels, J.-M. (1990). The self-organising exploratory pattern of the Argentine ant. Journal of Insect Behavior, 3, 159.

Elloumi, A., Kamoun, H., Jarboui, B., & Dammak, A. (2014). The classroom assignment problem: Complexity, size reduction and heuristics. Applied Soft Computing, 14, 677 - 686. doi:10.1016/j.asoc.2013.09.003

Glover, F. (1986). Future paths for inter programming and links to artificial intelligence. Computer and Operations Research, 13, 533 – 549.

Hambali, M. A., Olasupo, A. Y., & Dalhatu, M. (2020). Automated University Lecture Timetable Using Heuristic Algorithm. Nigerian Journal of Technology, 39(1), 1 - 14. doi:10.4314/njt.v39i1.1

Lu, Z., & Hao, J.-K. (2008). Adaptive Tabu Search for Course Timetabling. European Journal of Operational Research, 1 - 35. doi:10.1016/j.ejor.2008.12.007

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics, 21(6), 1087–1092.

Mittal, D., Doshi, H., Sunasra, M., & Nagpure, R. (2015). Automatic Timetable Generation using Genetic Algorithm. International Journal of Advanced Research in Computer and Communication Engineering, 4(2), 245 - 248. doi:10.17148/IJARCCE.2015.4254

Muklason, A., Irianti, G. R., & Marom, A. (2019). Automated Course Timetabling Optimization Using Tabu-Variable Neighborhood Search Based Hyper-Heuristic Algorithm. The Fifth Information Systems International Conference 2019 (pp. 656 – 664). Elsevier B.V. doi:10.1016/j.procs.2019.11.169

Murairwa, S. (2010). A Hybrid Heuristic Research Framework: The Travelling Salesperson Problem. PhD Thesis, Universiti Utara Malaysia (UUM), Kedah.

Murairwa, S. (2014). Voluntary Sampling Design. International Journal of Futuristic Trends in Engineering and Technology, 1(7), 13 - 19. doi:30/08/2014

Murairwa, S. (2015). Voluntary sampling Design. International Journal of Advanced Research in Management and Social Sciences, 4(2), 185 - 200. Retrieved February 7, 2022

Murairwa, S. (2019). RESEARCH AND STATISTICS with application procedures in statistical package for social sciences (First Edition ed.). Zimbabwe: Media Essentials.

Murairwa, S. (2020). Third Generation Heuristic: Hybrid Erosion And Deposition (HEAD) Metaheuristic. Mauritius: LAP Lambert Academic Publishing.

Murairwa, S. (2021). RESEARCH AND STATISTICS with Application Procedures in Statistical Package for the Social Sciences. Harare: Media Essentials.

Murairwa, S. (2021). Study on the Dual Bi-Directional Heuristic Development Framework. In M. Basu, Recent Advances in Mathematical Research and Computer Science. Book Publisher International.

Murairwa, S. (2022). A Framework for Analysing Heuristics Data for Performance Comparison to Solve a Travelling Salesperson Problem. Global Scientific Journals, 10(1), 1461 - 1477.

Murairwa, S. (2022). The Effects of Covid-19 pandemic on the Micro, Small, and Medium Enterprises in Manicaland, Zimbabwe. Global Scientific Journals, 10(8), 1282 - 1294. Retrieved from https://www.globalscientificjournal.com/researchpaper/The_Effects_of_Covid_19_pandemic_on_the_Micro_Small_and_Medium_Enterprises_in_Manicaland_Zimbabwe.pdf

Oude Vrielink, R., Jansen, E., Hans, E., & van Hillegersberg, J. (2019). Practices in timetabling in higher education institutions: a systematic review. Annals of Operations Research, 275, 145 – 160. doi:10.1007/s10479-017-2688-8

PhephauFUNDE. (2020, October 10). Leadership Challenges in Managing Timetabling. Retrieved from PhephauFUNDE: https://www.phephaufunde.org.za/themes/leadership-challenges-in-managing-timetabling/