Paper ID: 506

Automate Timetable Scheduling with AI: A Review

C L T H Gajanayake#, W P J Pemarathne

Department of Computer Science, Faculty of Computing, Sir John Kotelawala Defense University, Sri Lanka

#chathushaka97@gmail.com

Abstract : Scheduling timetables are one of the complex and time-consuming process when constructing using manual methods. These manual methods don't always promise the optimum schedule plan and leads to countless conflicts. Recently, there are many states of the art systems proposed for the task scheduling using Artificial Intelligence (AI). This paper reviews the recently propose timetable scheduling systems with AI. The result of the analysis shown that the evolutionary techniques has been used in many studies to generate optimize timetable schedule specially using the Genetic Algorithm. Most of the studies proved that the Genetic Algorithm optimizes most of the constraint and fitted to automate timetable scheduling.

Keywords: Scheduling, Timetable Automation, Artificial Intelligence, Genetic Algorithm

Introduction

Time is more important in present world. There are endless opportunities, but we have only 24 hours a day. So time management is very important and timetable is most common method to manage time. Every institute has a schedule to do their operations. Most of the schedules are fixed. For example, in Sri Lanka typical office hours are between 8AM to 5PM. Most government and private sector institutes have fixed timetables or schedules. Those schedules cannot be changed, and flexibility is minimum. Those schedules haven't changed or updates since last 3 – 4 decades. But in present world population and human needs are growing exponentially. So we need a flexible scheduling systems that are not static or fixed. We need dynamic schedules that can evolve according to situations. Best example for this is Colombo morning and evening traffic jams. Since all institutes have static and not upgrading schedules, everyone needs to be at office on 8AM. Population and needs are growing. But infrastructure is not developing according to those needs. For example, let's consider about roads. In Sri Lanka we import lot of vehicles. But our main roads still only have two lanes. So it is clear we have limited resources. Because of that we have to spend more time on road. We have to sacrifice our limited time for that cause. But imagine if we have dynamic schedules that changes according to the traffic. In these days work from home is very popular. If we can change our working hours how easy it is. In practically this might be difficult.

Since we cannot apply dynamic schedule to whole country, we can experiment this in our university. My aim is to create dynamic scheduling system, which updates timetables according to the situation. For example if a lecturer cannot attend to the lecture tomorrow lecturer can request to change his lecture in to another time. Then the system should able to swap other lecture with this lecture without clashes. Then both students and lecturers can save their time. This paper focuses techniques and methods which are used in Scheduling and timetable projects and systems and analyze and review them.

In this study we pay focus on which areas to consider and which constraints should consider when creating a timetable system.

Since we focus only about faculty of computing in KDU, main constraints which are considered is time availability of lecturers, resources and subjects. These are some factors which we should avoid clashing each other.

Literature Review

When it comes to scheduling problems, we cannot create universally applicable application to solve every scheduling problem. (McCollum et al., 2012a) In this study my main aim is to create application that can automate faculty of computing timetable scheduling system. Currently timetable scheduling done by manually. It is less efficient and consume lot of time. And also, it is less flexible because timetables cannot change easily. It takes lot of time to recreate timetable without any clashes because there are lot of things to consider.

Most of researches are base on few technologies and algorithms. Among those algorithms Local Search Procedures are more often used in researches and projects. (Burke et al., 1995a). Before come to conclusion we should discuss about those technologies and algorithms first.

A. Local Search Procedures

This is one of common technology which used to create scheduling problems. Tabu Search, Simulated Annealing and Genetic Algorithm are Local Search Procedures. Genetic Algorithm is most commonly use algorithm. Let us discuss above methods.

B. Simulated Annealing (SA)

This method is based on probabilistic method that is applied like the global optimum of given function. When it comes to huge search space simulated annealing gives accurate global optimization (Pillay and Özcan, 2019a). This method is good for use when it comes to search space is different. This technique gives good results for an optimization dilemma. If our problem contains condition that we want to reduce or maximize parameters or other thing, that problem can be solved by Simulated Annealing in most cases. Simulated Annealing starts using initializing random solutions first. Main procedure of this method is generating random solutions which is neighbors of the current solution using loop. Structure of the problem define the definition of the neighboring solutions.

C. Tabu Search

Tabu search algorithm can handle huge relations of derivative approach that can handle and create memory structure in metaheuristics. Tabu search and parallel tabu search is examples. Tabu search is meta strategy or metaheuristics which can use for get calculations from surrounding heuristics. Ans it is also a global optimizing algorithm (Islam et al., 2016a). To solve optimization problem this approach is used mostly.

Fred Glover's ideas are based for tabu search. This method is used to find solutions using meta-heuristics approach which search and explore the solutions which are in beyond the local optimality. Over the last decades this method became very popular because this method could produce or give best possible answer or solution that near to the best possible solution. Since tabu search have adaptive memory feature, this method has very flexible search behaviors (Burke et al., 1995b).

D. Genetic Algorithm

This algorithm is invented by John Holland. He wrote a book about genetic algorithm called "Adaptation to natural and artificial systems.". Genetic algorithm is based on Evolutionary Algorithms (Al-Majmar and Al-Shfaq, 2016a). This algorithm uses natural collection principle to create and develop to give best solution as outcome. Genetic algorithm is also a heuristic search which have natural evolution features like mutation, inheritance, crossover and selection to generate solutions to optimize and get good solutions for a problem (AlMajmar and Al-Shfaq, 2016a; Rozaimee et al., 2017a; Salvi et al., n.d.).

Genetic algorithm is used commonly to develop scheduling systems. When scheduling a timetable there may be set of solutions that doesn't violate constraints. So, when using a genetic algorithm we get pool of good solutions. So, using this algorithm we cannot obtain best answer. Because of evolutionary features like mutation and crossover this algorithm is more efficient and take less time to search.

E. The Constraint Programming (CP)

The main feature of this method is, this method can clearly recognize the constraints as a part of the program. This pays the way to adaptability which is a need feature of scheduling timetables (Gervás and Miguel, n.d.). By backtracking search and condition passing this method can narrow down the search domain. This can minimize time to search. In present constraint programming languages do not need to plan functions explicitly (Department of CSE, SDMIT Ujire, Karnataka, India et al., 2017). There are some main drawbacks of this approach. They are, Hard to define soft constraints Potential problems with enhancing the initial feasible solutions.

F. Heuristic selection methodologies

In this methods solution are formed gradually. Initially this method starts with empty solution. Then this approach can intelligently select and construct a complete solution. This approach uses SCFG rules to create the system components needed to create schedule. Stochastic context-free grammar (SCFG) lot alike context free grammar (CFG). Heuristic selection contains hyper heuristic framework, which have many pre-existing constructive heuristics. Main challenge is to find relevant heuristic according to the problem. Until the complete solution obtained this cycle continues.

Methodology

In literature review, most projects used genetic algorithm for the timetable scheduling systems. For my project I also chose genetic algorithm. Since I design a timetable system for the faculty of computing in KDU, this application is based on the requirements of the faculty. There are soft and hard constraints when scheduling a timetable. Hard constraints cannot violate in any means. Soft constraints should try to avoid violations. Since there are lot of possible answers for a problem, sometimes soft constraints can be ignored. In computer faculty there are some main constraints identified. They are,

- Visiting lecturers get priority when scheduling timetables. They are assigned first according to their available time.
- Then inhouse lecturers are assigned according to their availability.
- No lecturer can have two lectures at same time and tow lecturers cannot have lectures at same time.
- There are Computer science, Software engineering and computer engineering majors in the computer faculty. They have separate selective subjects. Those selective lectures cannot be clashed.

And when consider about soft constraints they are,

- Inhouse lecturers cannot have two lectures have two lectures at one day.

After identifying constrains we can try to develop the chromosome representation of the genetic algorithm for the timetables. A single chromosome is a solution. We can generate those chromosomes randomly according to our need. Then we have to create population of chromosomes. Chromosome representation is one of hard tasks of this approach. KDU typically ends at 2.30PM. So there are 4 time periods and two sessions in most time. In rarely there are lectures at 3-5PM. Then we can assign value for each timeslot. This is considered as a chromosome. Chromosome contains genes. Those genes represent details about the lecture.

After designed chromosome representation we need a selection method. We build fitness function which can calculate the fitness of the chromosome. When we have more fitness, it is a better solution. We can implement fitness function according to the constraint violation of a chromosome. We can declare required accuracy for the fitness function.

Then we have to select chromosomes from the population which do not have required fitness level. Then we perform mutation and crossover until the required level is achieved.

Discussion

A. Limitations in Existing Systems

Allocating a timetable for a primary school may seem like easy task because there are only few subjects and most importantly they are fixed. So it is not a difficult task to schedule a timetable manually. But when comes to the university level there are many aspects to consider when scheduling a timetable. Many subjects and modules. Lot of elective subjects and limited space or lecture rooms etc. So, there are lot of constraints. Even automatic systems face difficulties when meet these types of constraints. So, it is very important to consider limitations in existing systems and technologies.

When considering Simulated Annealing it is very heavy computational heavy function. And also other major drawback is this method cannot select optimal solution by itself. But this have advantages also. Simulated Annealing is easy to code even for complex problems. And also it also gives good solutions

Genetic algorithm is most used in automated timetable systems. But it also has both advantages and disadvantages. So when considering disadvantages genetic algorithm might not find the most optimal solution to the defined problems in all cases. The first and most important consideration in creating a genetic algorithm is the definition of a representation of the problem. The language in which the solution candidates are specified must be robust. Hard to choose parameters like population size, generation number etc. A major obstacle in genetic algorithms is the coding of the fitness (evaluation) function to achieve higher fitness and better solutions to the problem at hand. A wrong choice of fitness function can lead to critical issues, such as the resolution of a problem can not be found or worse, and lead to a wrong solution of the problem. It is very hard to find a good heuristic reflects the algorithm we need. And also when we use genetic algorithm for a system it is computationally expensive. So we might need to upgrade our existing machine to implement this system. Genetic algorithm need less information on the problem. But drawback is designing an objective function, representation and operators right can be difficult.

But Genetic Algorithm has advantages also. Coding is really is when compared to other algorithms. GA can find a solution in a very less time. Concept is easy to understand.

We do not have a fully automated system to schedule timetable in universities in Sri Lanka. Existing manual systems are observed with some problems. Some parts of the existing system is not efficient enough. Some generated timetables have course clashing. So it need to fix. Other major problem in existing systems are vulnerability to error due to human factors like stress and fatigue. As mentioned above universities contain many courses and modules. Lot of elective subjects. So it is very hard to schedule a timetable. Doing such tasks may find very stressful for the employees in the university. Also unavoidable data omission because of too many data in the collection. Data redundancy is a inherent problem in manual system.

Conclusion

Information technology has evolved in recent decades. The schedule software applications have been adapted and enabled to generate and optimize more appropriate timetables in an automated environment for higher education institutions. Literature research revealed that applications from reputable vendors were gradually adjusting the use of spreadsheets, a database management system, schedule editors, web-based tools, and an improved graphical user interface. Timed software applications thus use modern techniques (prior art) and are therefore able to use them.

The other component includes the working mechanism, the solution method, and the algorithm used to create a work plan. The timing problem is very challenging and many possible combinations need to be explored to find a list of acceptable solutions. Since it is impractical to list all combinations, one will choose an approach that calculates a subset or part of it. Such algorithms can give an approximation that is considered an acceptable solution. Heuristic algorithms therefore seem to surpass traditional methods, and such algorithms are even combined to reinforce one another. The latest developments can be found in the field of hyperheuristics. Such solution strategies aim to generate timetables by choosing the right algorithms. However, commercial timetable products do not seem to focus on the actual implementation of such solution methods in timetable applications.

Reference

Al-Majmar, N.A., Al-Shfaq, T.H., 2016b. Solving of Lectures Timetabling Problem and Automatic Timetable Generation using Genetic Algorithm. International Journal of Advanced Research in Computer and Communication Engineering 5, 505–512. https://doi.org/10/ggbd7g

Almu, A., Muhammad, A.B., 2013. ISSN: 1597 – 9928 http://www.wiloludjournal.com doi:10.5707/cjapplsci.2013.8.3.23.30 AUTOMATING LECTURES TIMETABLE USING CONSTRAINT SATISFACTION TECHNIQUE 9

Burke, E., Elliman, D., Weare, R., n.d. The Automated Timetabling of University Exams using a Hybrid Genetic Algorithm 8.

Burke, E.K., Elliman, D.G., Weare, R.F., 1995b. The Automation of the Timetabling Process in Higher Education. Journal of Educational Technology Systems 23, 353–362. https://doi.org/10/fmpk5k

Delgado, A., Pérez, J.A., Pabón, G., Jordan, R., Díaz, J.F., Rueda, C., 2005b. An Interactive Tool for the Controlled Execution of an Automated Timetabling Constraint Engine, in: Van Roy, P. (Ed.), Multiparadigm Programming in Mozart/Oz. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 317– 327. https://doi.org/10.1007/978-3-540- 31845-3_26

Department of CSE, SDMIT Ujire, Karnataka, India, M, S., Vaze, P.K., Department of CSE, SDMIT Ujire, Karnataka, India, Pradeep, Department of CSE, SDMIT Ujire, Karnataka, India, N R, M., Department of CSE, SDMIT Ujire, Karnataka, India, 2017. Automatic Time Table Generator. IJARCSSE 7, 204–211. https://doi.org/10/ggbd68

Emmanuel, E., Friday, A., Onoja, Omolola, Y., Serah, 2019. Design of Automated Departmental Lecture Timetable System. Review of Computer Engineering Research 6, 24–34. https://doi.org/10/ggcb34

Gervás, P., Miguel, B.S., n.d. Sequential Building of Constrained Timetables Using Rule-Based Heuristics: An Expert System for Automated Timetabling at UEM 12.

Islam, T., Shahriar, Z., Perves, M.A., Hasan, M., 2016b. University Timetable Generator Using

Tabu Search. JCC 04, 28–37. https://doi.org/10/ggbd7k

Klevanskiy, N.N., Antipov, M.A., Krasnikov, A.A., 2017. Cognitive Aspects of Timetable Visualization: Support Decision Making. Procedia Computer Science 103, 94–99. https://doi.org/10/ggcb33

Mahgoub, H., Altaher, M., 2013. Automated Timetabling Using Stochastic Free-Context Grammar Based on Influence-Mapping. IJACSA 4. https://doi.org/10/ggbd63

McCollum, B., Burke, E.K., 2014. The practice and theory of automated timetabling. Ann Oper Res 218, 1–2. https://doi.org/10/ggcb36

McCollum, B., McMullan, P., Parkes, A.J., Burke, E.K., Qu, R., 2012b. A new model for automated examination timetabling. Ann Oper Res 194, 291–315. https://doi.org/10/dksvzq

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Gaspero, L.D., Qu, R., Burke, E.K., 2010. Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition. INFORMS Journal on Computing 22, 120–130. https://doi.org/10/bvhq4p

Meng, L., Muneeb Abid, M., Jiang, X., Khattak, A., Babar Khan, M., 2019. Increasing Robustness by Reallocating the Margins in the Timetable. Journal of Advanced Transportation 2019, 1–15. https://doi.org/10/ggcb35

Murray, K., Muller, T., n.d. Automated System for University Timetabling 7

Ojha, D., Kumar, R., Das, S., 2016. Automated Timetable Generation using Bee Colony Optimization. IJAIS 10, 38–43. https://doi.org/10/ggbd6z

Oude Vrielink, R.A., Jansen, E.A., Hans, E.W., van Hillegersberg, J., 2019. Practices in timetabling in higher education institutions: a systematic review. Ann Oper Res 275, 145–160. https://doi.org/10/ggcb37

Pillay, N., Özcan, E., 2019b. Automated generation of constructive ordering heuristics for educational timetabling. Ann Oper Res 275, 181–208. https://doi.org/10/ggbd6p

Rozaimee, A., Shafee, A.N., Hadi, N.A.A., Mohamed, M.A., 2017b. A Framework for University's Final Exam Timetable Allocation Using Genetic Algorithm 7.

Salvi, A., Khanvilkar, O., Balkhande, B.W., n.d. Automatic Time-Table Generation System using Genetic Algorithm 5, 3.

Sampebatu, L., Kamolan, A., n.d. TIMETABLE MANAGEMENT USING GENETIC ALGORITHMS 6.

Schaerf, A., 1999. [No title found]. Artificial Intelligence Review 13, 87–127. https://doi.org/10/ffkzrt

Schonberger, J., Mattfeld, D.C., Kopfer, H., 2000. Automated timetable generation for rounds of a table-tennis league, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512). Presented at the 2000 Congress on Evolutionary Computation, IEEE, La Jolla, CA, USA, pp. 277–284. https://doi.org/10/chdq47

Xu, T., Zhou, L., Bai, Z., Li, W., Guo, B., Wang, Z., 2019. The Research on Big Data Platform based on Timetable Management. IOP Conf. Ser.: Mater. Sci. Eng. 569, 052081. https://doi.org/10/ggcb38