Agent based integer programming framework for solving real-life curriculum-based university course timetabling

Mansour Hassani Abdalla, Joe Henry Obit, Rayner Alfred, and Jetol Bolongkikit

Knowledge Technology Research Unit, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
mansourabdalla22@gmail.com, joehenry@ums.edu.my, ralfred@umse.du.my, jetol@ums.edu.my

Abstract. This research proposes an agent-based framework for solving real-life curriculum-based University Course Timetabling problems (CB-UCT) at the Universiti Malaysia Sabah, Labuan International Campus (UMSLIC). Similar to other timetabling problems, CB-UCT in UMSLIC has its own distinctive constraints and features. The proposed framework deal with the problem using a distributed Multi-Agent System (MAS) environment in which a central agent coordinates various IP agents that cooperate by sharing the best part of the solution and direct the IP agents towards more promising search space and hence improve a common global list of the solutions. All agents are incorporated with Integer programming (IP) search methodology, which is used to generate initial solution in this, regards as well. We discuss how sequential IP search methodology can be incorporated into the proposed multi-agent approach in order to conduct parallel search for CB-UCT. The agent-based IP is tested over two real-life datasets, semester 1 session 2016/2017 and semester 2 session 2016/2017. The experimental results show that the agent-based IP is able to improve the solution generated by the sequential counterpart for UMSLIC's problem instance used in the current study impressively by 12.73% and 17.89% when three and six IP agents are used respectively. Moreover, the experiment also shows that increasing the number of IP agents lead to the better results.

Keywords: Integer Programming, Multi-Agent System, Asynchronous Cooperative Search.

1 Introduction

Curriculum based university course timetabling is an interesting topic to study because neither modeling nor solving them is a straightforward task to do. This is because each problem has its own unique characteristics and variations which differ from one university to another [3]. Besides, duplicating the previous timetable does not really solve the problems as university are growing with a great pace and the

R. Alfred et al. (eds.), *Computational Science and Technology*, Lecture Notes in Electrical Engineering 481, https://doi.org/10.1007/978-981-13-2622-6_7

teaching program is evolving towards more modular and distributed nature where students are able to choose course from other programs or even from other faculty. These fluctuations of teaching guidelines lead to the problem of constructing different timetables in every semester. Also as [3] highlight that, “poor quality course timetabling can lead to massive costs for the peoples affected by the timetable, for example students might not be able to attend all of their lessons if clashes exist”. The cost here is the student need to take the course in the future semester which ultimately will result for student to extend study time. A high quality timetable is the one in which all the peoples (students, lecturers and academic departments) affected by the timetable are satisfied. However, constructing timetable which satisfies students, lecturers and academic departments is not an easy task.

Basically, all CB-UCT is associated with constraints which are different from other universities. In addition, these constraints vary from time to time and are classified into two categories which are hard constraints and soft constraints [2]. Hard constraints must be satisfied at all circumstances while soft constructs are used to determine the quality of timetable and the more the soft constraints are satisfied the better the timetable produced. In order to solve this problem for the certain real-life case of CB-UCT, we have adopted the agent-based incorporated with IP search methodology.

Generally, in past ten to twenty years ago, agent-based technology has entered the scene of software industry and proves its suitability [4]. MAS fall into the area of distributed systems, where number of entities work together to cooperatively solve given problems. [1] Pointed out that, “MAS are concerned with coordinating behavior among a pool of autonomous intelligent agents (e.g. software agents) that work in an environment”. In this regard MAS is effective because it facilitates the agent to share the best part of the solution and hence guide the search process to more promising region. These agents can be cooperating to achieve common goals however generally on other systems agents are competing with each other to fulfil the delegated objectives [17] are commonly intended as computational systems where several autonomous entities called agents, interact or work together to perform some tasks[2]. Likewise, in CB-UCT, the MAS could find a high-quality and acceptable solution with minimal message passing as well. In this work, we are proposing agent-based framework incorporating IP search methodology where agents are working together by sharing the best part of the solution to achieve the delegated objectives which in this work is to improve the global solutions.

2 Problem Definition

In the current study agent-based framework (as shown in figure 1, section 4) incorporating IP search methodology that fulfills the requirements of zero hard constraints values and minimum values for soft constraints is proposed. The proposed framework is used to test on real-life datasets at UMSLIC as shown in table 1. The objective of the problem is to develop communication protocol that helps agents in the framework to share the best part of the solution, guide the agents towards more promising search space, and hence find the improved feasible timetable solutions. The problem involves several hard constraints and soft constraints. Hard constraints need to be satisfied in all circumstances, whereas soft constraint violations should be minimised to increase the timetable quality, and increasing the satisfaction of the people who are affected by the timetable. The constraints undertaken in this work are explained in the following subsections.

Essentially the problem in the current study involves allocating a set of 35 timeslots (seven days, with five fixed timeslot per day) according to UMSLIC teaching guidelines. Each lecturer teaching several courses in each semester and each course has at least one lecture of minimum two hours per week. In addition, UMSLIC's administration has a guideline as shown in table 2 for the compulsory, elective, center for promotion of knowledge and language learning (PPIB), and center for co-curriculum and student development (PKPP) courses to be enrolled by the students in each of the semesters throughout the students' university days

Our approach will consider certain lecturer's preferences, better utilization of appropriate room and improved evenly student's schedule. Moreover, our approach also fulfill university teaching guideline where there are some general preferences such as some courses particularly program and faculty courses cannot be scheduled on weekends and must be scheduled on the first or third timeslots of the weekdays. In addition, some course such as PKPP courses cannot take place on weekdays. In addition, some courses such as PPIB course must be scheduled on second, fourth, or fifth in timeslot.

Hence, this research concentrates on real-life CB-UCT. In fact, in CB-UCT there are five variables identified namely periods, courses, lecturers, rooms, and curricula. The objective is to assign a period and a room to all lectures of each course according to the hard and soft constraints based on UMSLIC teaching guidelines. This research work aims to implement agent-based incorporating an IP search methodology for solving real-life CB-UCT for UMSLIC.

Table 1. Summary of the dataset from UMSICL academic division

	Semester1 s2016/2017	Semester2 2016/2017
Number of student	2263	2224
Number of curriculum	65	49
Number of lectures	108	92
Number of courses	134	117

2.1 Hard Constraints

Listed below are all the predefined hard constraints considered in this work:

1. *Lectures.* Each course has a predetermined amount of lectures that must be given. Each lecture must be scheduled in distinct time slots and the total number of lectures cannot be exceeded.
2. *Room conflict.* Each Two lectures cannot take place in the same room in the same time slot.

3. *Main and PPIB courses.* All main (major) courses cannot be scheduled at weekend. This is according to UMSLIC teaching guideline. Main course involves program core and school course as well as some PPIB courses.
4. *Center for co-curriculum and student development (PKPP) courses.* All PKPP courses must be scheduled at weekend. There are some courses under PKPP which by default must be scheduled at weekend. Normally this course is taught at the early semesters of the students' university years.
5. *Room Capacity.* The size of the room must be larger or equal to the size of the course. The room where the course is scheduled should be large enough to accommodate the number of students registered for that course.
6. *Curriculum and lecturer conflicts.* Lectures of courses in the same curriculum or taught by the same lecturer must all be scheduled in different time slots

2.2 Soft Constraints

Listed below are all the predefined soft constraints considered in this work:

1. *Lecturer preferences.* The assignment of classrooms and periods of time must allow satisfying at best the preferences of lecturers. I.e. there should be a gap for lectures taught by same lecture as well as the lecturers can specify times when they prefer not to lecture.
2. *Appropriate room size.* The usage of appropriate room size i.e. does not schedule a lecture with 30 students in a room with capacity of 300 seats.
3. *Evenly timetable.* The Student should not have consecutive courses per any given day.

Table 2. UMSKAL teaching guideline. Where 1 stands for Faculty courses, Program courses or elective courses; 2 stand for PPIB courses; 3 stand for PKPP courses.

Day/Time	**Time groups**	
	Monday - Friday	Saturday & Sunday
08.00 AM – 10.00 AM	1	3
10.00 AM – 12.00 PM	2	3
02.00 PM – 04.00 PM	1	3
05.00 PM – 07.00 PM	2	3
07.00 PM – 10.00 PM	2	3

3 Related Works

In general, there are many techniques proposed in literatures for solving timetabling problems in particular curriculum-based course university timetabling. However, scholars in operational research and artificial intelligence acknowledge Metaheuristics as indispensable techniques to address difficult problems in numerous and diverse fields [5]. Likewise, recently hype-heuristics has been widely used to address the issues. Nevertheless, even meta-heuristics may reach quite rapidly the limits of what may be addressed in acceptable computing times for many problem settings for research and practice alike [6, 18]. Similarly, hyper-heuristics do not generally guarantee optimality, performance often depending on the particular problem setting and instance characteristics [7]. Therefore, this thought has led birth of the fertile field of cooperative search especially in the operational research and artificial intelligence research community.

Generally, cooperative search can be natural approach to address the issues resulted from meta-heuristics and heuristics alike. [16] Stated that, "instead of trying to design new algorithms without downside, a task that is quite difficulty if not impossible, scholars in operational research and artificial intelligent research community have been working on the ways to organize the existing techniques in order to suppress their weakness through cooperation, and together do what separately they might not be able to accomplish". Ultimately parallel implementations of sequential algorithms appear quite naturally as an effective alternative to speed up the search for approximate solutions of combinatorial optimization problems [8]. Moreover parallel implementations allow solving larger problems or finding improved solutions, with respect to their sequential counterparts, due to the partitioning of the search space and to more possibilities for search intensification and diversification [4, 8, 19].

However, even with recent enormous effort in cooperative search, [15] believes this area has been little explored in operational research. Also, as computers keep becoming very powerful nowadays, this present huge opportunity for researcher to do what was unable to be done in 20 to 30 years ago especially in parallel computational research area. Similarly, according to [9] in recent years, multi-core processors are widely used and cooperative search can easily benefit from parallel processing. Thus in the last few years research community have started to exploit the opportunity presented by multi-core processors and work on how to develop optimization technique that is faster, more robust and easier to maintain.

More research in combinatorial optimization is currently being devoted in cooperative search techniques. Several number of cooperative search approaches have been proposed in the literature [4, 10]. The key idea behind cooperative search is to combine the strengths of different (meta-) heuristics to balance intensification and diversification and direct the search towards promising regions of the search space [4, 11]. Essentially, by cooperating the chances of finding novel and greatly improved solutions are increased.

[12] Defined cooperative search as the "parallelization strategy for search algorithms where parallelism is obtained by concurrently executing several search programs". In general cooperation by these programs is to interact with one another directly or indirectly, synchronously or asynchronously. Therefore the communication and sharing of information is an important feature of cooperation in cooperative search field [15]. The need to interact in such systems occurs because programs (agents) solve sub problems that are interdependent, either through contention for resources or through relationships among the sub problems [13]. The benefit of this

approach is the fact that, it adds parallel computational resources and possibility of information exchange (exchange best part of the solution) among the agents [14]

However, so far most of cooperative search focus more on metaheuristics and heuristics. Interestingly, integer programming search naturally offers significant opportunities for parallel computing, yet not enough research has been devoted to parallel integer programming implementations. In this research, we propose asynchronous agent-based framework incorporating integer programming search methodology for solving real-life CB-UCT at UMSLIC.

4 Agent-Based CB-UCT IP Framework

Figure 1 present the proposed agent-based searches framework. In this research, a decentralized agent-based framework, which consist of given number of agents (n) is proposed. Basically a framework is a generic communication protocol for integer programming (IP) search methodology to share solutions among each other. Each IP is an autonomous agent with its own representation of the search environment. To this end they share complete feasible solution to enable each other to direct (move) towards more promising search space. Moreover, the communication or ability for the agent to exchange the solutions with one another via the central agents prevent individual agent from stacking on the local optima [8]. Essentially all agents in the distributed environment communicate asynchronously via the central agent. Additionally, it is worth mentioning that, the initial feasible solution is generated by the central agents as well. In clarity this framework will involve asynchronous cooperative communication as follow.

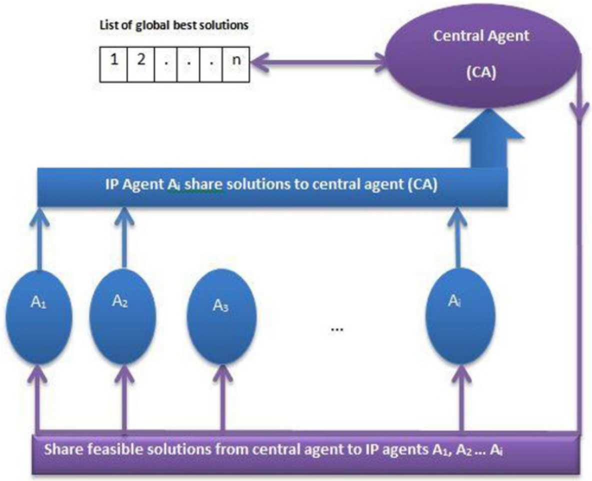

Fig. 1. Proposed Agent-based IP Search methodology Framework

4.1 Central agent (CA)

The central agent is responsible to generate the initial feasible solutions as well as to coordinates the communication process of all other agents involved in the proposed framework. The central agent acts as intermediate agent among other agents where it passes the feasible solution and other parameters to the IP agents asynchronously on top of FIPA-ACL communication protocol. On top of that, the central agent receives the improved solution from the IP agents and compares the objective function cost value of the received solution with the existing global solutions on the list, if the improved solution's objective is better or similar to any of the solutions on the existing solutions then the worse in the list is replaced. Else the received solution is discarded and the central agent randomly select other solution from the list of the global solutions and send back to that particular agent so in order for the agent to try to improve the new solution received from the central agents.

4.2 IP Agents (A_i)

All other agents' start from the complete solution received randomly from central agent and iteratively perform search to improve the solution autonomously (independently). In this case the agents have to maintain the feasibility of the solution i.e. do not violent hard constraint. After certain number of iterations according to the rules stated (after every 10 seconds and no improvement found) the agent passes the solution back to the central agent and request new solution from the central agent. The central agent accepts the solution if only the solution is better or similar to the existing global solutions in the list of the solutions else the solution is discarded. If the solution is accepted then the solution with higher objective cost function i.e. worse in the list will be replaced. The reason an IP agent's exchange solution is to make sure the agents are not stuck on local optima, moreover scholars highlighted on the literature that, by exchanging the solution the possibility of the agents (algorithms) changing the position towards more promising search space is increased [4, 8, 19].

Best solution Criteria. All of our agents are incorporated with integer programming search methodology. Each agent also is capable to compute the final objective function and return it along with the improved solution. The central agent places all the solutions obtained in a sorted list where the solution on top will be the best solution (the solution with minimum objective function value).

In this framework the value of the objective functions is used to determine the quality of the solution. The lower the cost value the better the solution. Hence for the solution which has improved by the IP agents to be considered better than or similar to the global existing solutions, the returned improved solution's objective function should be lower than or similar to the one of the available in the global solutions objective functions values. Else the solution is discarded. The objective is to enable the IP agent to escape from local optimal and more importantly to allow the agent to move towards the most promising search space by sharing the best part of solution.

The whole process stops when all the IP agents are not improving the solution any more in a given number of conversations. Conversation in this regards means number of communication between the central agent and improving. For example IP agent A_i request new solution from central agent try to improve the solution however the agent is unable to improve anymore for three consecutive conversations. In this case the agent has reach appoint where unable to improve the solution anymore.

Proposed Agent-framework's Commitments rules.

The communication of an agent is built on top of FIPA-ACL protocol. The send and receive massage mechanism is well explained in the subsequent sub-sections pseudo-code. The agents are in the agent society so each agent in the pack of agents follows the following commitments rules explained in as follow.

```
Commitments Rules (Pseudocode).
Let
Central agent is denoted as CA,
IP agents are denoted as Ai.

  {CA, REQUEST, DO (time, action)
          },;;; msg condition
          (B,
  [Now, Friend agent] AND
  CAN (self, action) AND
  NOT [time, CMT (self, anyaction)
  ),;;; mental condition
  DO (time, self, action)
  }
```

The proposed framework's commitments rules pseudocode may be paraphrased as follows:
If IP Agent *(A_i)* receives a message from central agent (*CA*) which requests A_i to do action (improve the solution) at time t, and *Ai* believe that; *CA* is currently a friend; and A_i can do the action; at time t, and A_i not committed to doing any other action, then A_i will commit to doing that action at time *t*. All agents in the framework are following this set of rules. These set of rules, guide agent in the framework on what to do on a given time to make sure agents do not interfere one action with another

5 Experimental Setup and Results

Now we discuss the performance of the proposed agent-based framework for CB-UCT, in which two-semester problem instances of different difficulty is tackled. For each semester (session one (s1) 2016/2017 and session two (s2) 2016/2017) datasets, the initial solutions generated by the central agent using pure 0-1 IP. In average the initial solutions are generated in five seconds. To determine the consistence of the al proposed framework, for each instance, we run the experiments 50 times and the average final costs are computed in table 3. In this experiment, first we use three IP agents (Ai), and then we increase the number of IP agents (Ai) from three to six IP agents (Ai).

The improvement from initial to final cost value when three IP agents (Ai) are used is 12.73% and 10.20% for s1 2016/2017 and s2 2016/2017 respectively. On the other hand, the improvement of the solution's cost value when six IP agents (Ai) is used are 17.89% and 15.58% % for s1 2016/2017 and s2 2016/2017 respectively. The main benefits of the agent-based approach adopted for CB-UCT are the possibilities of intensifying and diversifying the search space, where Ai is able to changes solutions among each other in the distributed MAS. This leads the IP agents to easily move towards the most promising search areas of the search space. Basically, by the analysis the results, the numbers of IP agents used in the framework determine the quality of the solution generated. In this regard we find out the quality of the solution in this framework proves to increase slightly as the number of IP agents (Ai) are increased

Table 3. Experimental results for the proposed agent-based search framework.

	No of agents	Semester1 s2016/2017	Semester2 S2016/2017
Initial cost	-	368.04	377.29
Final average cost	3	321.20	338.80
Final average cost	6	302.20	318.50
Average improvements (%)	3	12.73	10.20
Average improvements (%)	6	17.89	15.58

6 Conclusion and Future Work

The current study focuses on agent-based IP framework for the CB-UTT for real-life instances in UMSLIC. The proposed framework is able to produce an applicable solution for UMSLIC. Based on the methodology employed, it is discovered that the sharing of solutions among agent improved the overall performance of the framework as the number of agent increase the solution quality slightly improve.

The currents study recommends that the future work may include agent negotiation; the negotiation amongst the IP agents (A_i) may lead to better performances of the proposed agent-based search framework.

References

1. Oprea M.: Multi-Agent System for University Course Timetable Scheduling. The 1st International Conference on Virtual Learning, ICVL (2006)

2. Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, 43-59.
3. Obit. J. H., Ouelhadj, D., Landa-Silva, D., Vun, T. K.., Alfred, R.: Designing a multi-agent approach system for distributed course timetabling. IEEE Hybrid Intelligent Systems (HIS), 10.1109/HIS(2011)-6122088.
4. Obit, J. H., Alfred. R., Abdalla, M.H.: A PSO Inspired Asynchronous Cooperative Distributed Hyper-Heuristic for Course Timetabling Problems. Advanced Science Letters, (2017)11016-11022(7)
5. Crainic, T. G., Toulouse, M.: Parallel strategies for meta-heuristics. In Handbook of metaheuristics (pp. 475-513): Springer (2003).
6. Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing, 11(6), 4135-4151.
7. Crainic, T.G.: "Parallel meta-heuristic search", Tech. Rep. CIRRELT-2015-42, (2015)
8. Cung, V.-D., Martins, S. L., Ribeiro, C. C., Roucairol, C.: Strategies for the parallel implementation of metaheuristics. In Essays and surveys in metaheuristics (pp. 263-308): Springer (2002)..
9. Yasuhara, M., Miyamoto, T., Mori, K., Kitamura, S., Izui, Y.: Multi-objective embarrassingly parallel search for constraint programming. Paper presented at the Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference
10. Crainic, T. G., Gendreau, M.: A Cooperative Parallel Tabu Search for Capacitated Network Design, Technical Report CRT-97-27(1997).
11. Ouelhadj, D., Petrovic, S.: A cooperative hyper-heuristic search framework. Journal of Heuristics, 16(6) (2010)., 835-857.
12. Toulouse, M., Thulasiraman, K., & Glover, F. (1999, August). Multi-level cooperative search: A new paradigm for combinatorial optimization and an application to graph partitioning. In European Conference on Parallel Processing (pp. 533-542). Springer, Berlin, Heidelberg.
13. Lesser, V. R.: Cooperative multi-agent systems: A personal view of the state of the art. IEEE Transactions on knowledge and data engineering, 11(1) (1999), 133-142.
14. Silva, M. A. L., de Souza, S. R., de Oliveira, S. M., & Souza, M. J. F.: An agent-based metaheuristic approach applied to the vehicle routing problem with time-windows. Paper presented at the Proc. of the Brazilian (2014) Conference on Intelligent Systems-Enc. Nac. de Inteligência Artificial e Computacional (BRACIS-ENIAC 2014).
15. Martin, S., Ouelhadj, D., Smet, P., Berghe, G.V., Özcan, E.: Cooperative search for fair nurse rosters. Expert Syst. Appl. 40(16), 6674–6683 (2013).
16. Talukdar, S., Baeretzen, L., Gove, A., and de Souza, P.: Asynchronous teams: Cooperation schemes for autonomous agents. Journal of Heuristics, 4:295–321, 1998.
17. Wooldridge, M, Jennings, N.: Intelligent Agents, Lecture Notes in Artificial Intelligence 890 Springer-Verlag (eds.), 1995b.
18. Obit. J. H, Landa-Silva, D.: Computational Study of Nonlinear Great Deluge for University Course Timetabling, Intelligent Systems - From Theory to Practice, Studies in Computational Intelligence, Vol. 299, V Eds. Springer-Verlag, 2010, pp. 309-328
19. Obit, J. H.: Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems. Ph.D. Thesis, School of Computer Science University of Nottingham (2010).