Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COURSE SCHEDULING USING MULTI-AGENT EXPLORATION METHOD

[1]JAMALUDIN HAKIM, [2]RETANTYO WARDOYO, [3]SRI HARTATI, [4]AHMAD ASHARI

[1] Department of Information System, Faculty of Engineering and Information System, YAPIS Papua University

[2,3,4] Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science, Gadjah Mada University

E-mail: [1]j2hakim@gmail.com, [2]rw@ugm.ac.id, [3]shartati@ugm.ac.id, [4]ashari@ugm.ac.id

ABSTRACT

The issue of course scheduling is a frequent topic, involving the use of new techniques with promising results. This research used exploration method on the Schedule Media with the agents providing a flexible schedule based on the resulted combination and a stable process. The constraints include lecturers' teaching preferences and students' different choices of courses in the Semester Credit System; which would be the problem being addressed in this study. Using multi-agents, the exploration method would be able to process lecturers' constraints and varying courses chosen by students from varying semesters. Agents' responses to requests during the scheduling process would determine the success of the scheduling.

Keywords: *Scheduling, Exploration Method, Constraints, Multi-Agent, Schedule Media*

1. INTRODUCTION

The many rules and constraints faced in university scheduling systems have caused difficulties in completing schedules. Lack of resources and emergence of constraints have added complexities to the process of schedule arrangement and constraints fulfilment [1].

The variation in the constraints faced by universities is the reason why varying scheduling methods continue to be developed. However, most universities adjust their schedules manually when current methods still fail to fulfil constraints. The manual method used by most operators to adjust or fix schedules consists of exploring the schedule draft.

The manual visual exploration is still a common option among operators. The method is used both in designing schedules from scratch and making adjustments to current schedules. Schedule adjustment occurs due to requests from lecturers or other changes. The adjustment mostly occurs due to lecturers' teaching preference. Lecturers' teaching preferences would be one of the constraints expected to be solved. The lecturers having administrative posts and duties would be the reason why those constraints have to be solved.

Universities using the Semester Credit System allow students to choose courses from different levels. All courses chosen by students must be accommodated by the schedule. Therefore, in addition to same-level courses, those from different levels must be made available at the same time for any given student.

Meeting constraints would be the sole challenge in scheduling using the exploration method. The combination of schedules produced would be influenced by the sequence of schedule-determining components (courses and lecturers). The sequence of schedule-determining components would strongly influence the number of lecturer constrains accommodated in the schedule.

The scheduling method would be implemented in rules and algorithms. The rules of the exploration would be input to the agents as behaviours in commencing the exploration process. The exploration process requires parameters of courses list and lecturer constraints that would be represented by different agents. A successful exploration process requires suitable communication among the agents.

This research is expected to construct multi-agents implementing the exploration method. The multi-agent exploration would produce a flexible scheduling based on schedule-determining components. The multi-agent exploration scheduling method is also expected to accommodate as many lecturers' constraints as possible as well as all courses from different levels.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. RELATED WORK

The purpose of using more than one agent (multi-agent) in scheduling is to accommodate a given number of human activities interacting within the schedule construction. The schedule is constructed based on lecturers' requests for teaching time adjustments, which the system would send responses as to whether or not they could be accommodated [2]. This method requires continuous responses from lecturers until all teaching preferences can be fulfilled.

Each university has varying constraints as the parameter of optimization expected. The use of agents conforms to a set of basic rules to accommodate non-linear level course selections. In solving scheduling issues, each university formulates constraints in accordance with their specific needs and conditions [3]. The method applied here cannot be used in universities that use CSS.

Schedule arrangements in different universities generally share similarities in the use of resources, but they may differ in defining their respective issues. This is because each university has pre-set rules addressing their unique conditions and customs. The issue of lecturers' specialization will influence course instruction. The requirement for different courses needing to be taught by same or different specialists must be taken into consideration. Agents are designated to be able to negotiate with the specialists a given course as necessary [4]. Neither lecturers' administrative posts/duties nor students choices of courses are taken into account in this method.

Genetic Algorithm (GA) is one of the optimization methods frequently used and developed in scheduling. One of its implementations is by turning chromosomes into sub-chromosomes and later turning them into mod chromosomes [5]. This method is concentrated on solving lecturers' requests for schedule alterations, but not for students taking courses from different levels.

3. MODEL OF SCHEDULING SYSTEM

The model consists of three parts. The first is interface, the second is data storage to initiate the scheduling process, and the third is agents commencing the scheduling process.

As seen on Figure 1, the model consists of three agents. The first agent is the academic agent that will run the exploration. The second one is the lecturer agent that will receive and transmit lecturer constraints. The third is the student agent that will provide the list of courses taken for the effective semester.

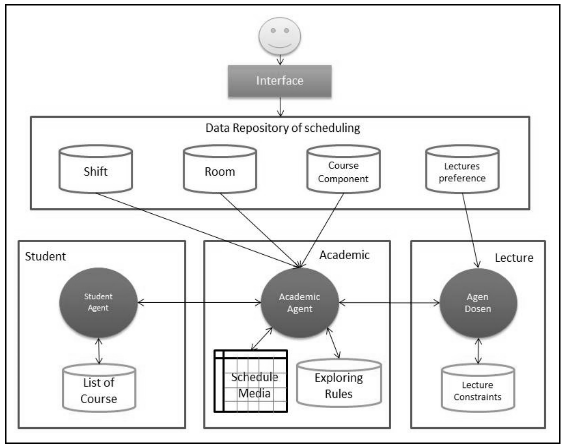

Figure 1: Model of Scheduling System

The academic agent requires parameters in running the exploration. The parameters will be requested from the lecturer agent student agent. Therefore, the academic agent must be able to communicate with the other two agents.

The model can be implemented in different systems or environments. The lecturer agent lecturer can be placed in a separate system or in lecturers' computers/laptops, while the student agent can be placed in the academic information system where course options are made available.

4. MULTI-AGENT EXPLORATION

The multi-agent is a flexible system that can be used for exploration in course scheduling. The exploration will be conducted by agents pre-assigned with a set of exploration rules. In principle, the exploration is conducted by one of the agents in the model referred to as the academic agent. In order to enable exploration, the academic agent is assigned a behaviour conforming to a set of exploration rules. The agent will then conduct the exploration in a medium constituting a draft schedule.

Specifically for the academic agent, a number of behaviours need to be assigned to it. In addition to behaviours conforming to exploration rules, other behaviours are necessary to enable communication with the other two agents. The purpose of communication with other agents is to obtain parameters required in the scheduling process.

During the scheduling process, the academic agent requires parameters of lecturers' constrains and the list of courses chosen by students. Academic agent must have the ability to

Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

communicate with other agents and make requests. The request for lecturer constraints will be sent to the lecturer agent and the request for a list of courses chosen by students will be sent to the student agent.

The lecturer agent will be assigned with a behaviour of capturing parameters sent by other agents. The aim of this behaviour is to capture and extract data being received.

The lecturer agent will be assigned with a behaviour of capturing parameter requests, another behaviour of searching for intended lecturer constraints, and another to transmit data being received in accordance with pre-set structures/ontology.

The student agent will have a behaviour enabling it to capture requests for parameters transmitted to it, another behaviour of selecting the list of courses needed, and another behaviour of transmitting the list of courses in accordance with pre-set structures/ontology.

At the start of scheduling, the academic agent will conduct a search throughout the entire detectible systems/environments for required agents. Once the agents are detected based on their identities, the academic agent will send them requests for required parameters.

The academic agent will commence its exploration on the schedule medium based on lecturer constraint parameters being received. If a lecturer constraint exists, the agent will immediately go to the intended cell and run explorations in accordance with exploration rules pre-set for each constraint. If no lecturer constraint is received, the agent will implement the exploration rules on the initial cell of the schedule medium (h=1, s=1, r=1).

5. COMMUNICATION AMONG AGENTS

The communication model shall conform to standards of FIPA (Foundation for Intelligent Physical Agents). The structure shall conform to the ACL (Agent Communication Language). Communication among agents not using the same data structure mandates a definition of ACL ontology/structure.

The academic agent will send requests for lecturer constraints to the lecturer agent. The lecturer agent will respond by sending the required lecturer constraints. Figure 2 shows the structure of lecture constraint request communication by the academic agent. Figure 3 shows the respond towards the lecturers' constraints request.

```
(QUERY-IF
 :sender ( agent-identifier :name AgenAkademik@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc ))
 :receiver (set ( agent-identifier :name AgenDosen@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc )) )
 :content "DS003"
)
```

Figure 2: Structure of Lecturer Constraints Request

```
(INFORM
 :sender ( agent-identifier :name AgenDosen@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc ))
 :receiver (set ( agent-identifier :name AgenAkademik@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc )) )
 :content "((KetentuanDosen (agent-identifier :name AgenDosen@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc)) (SlotJadwal :hari 1 :shift 1 :ruang 1)))"
 :language fipa-sl :ontology Jadwal-Ontology )
```

Figure 3: Structure of Respond to Lecturer Constraints Request

In order to verify that students' choice courses are not scheduled at the same time, the academic agent requires the list of courses chosen by the student. This list will be requested by the academic agent from the student agent. Figure 4 displays the communication structure of the academic agent requesting the list of courses. Figure 5 describes the communication structure of the student agent's response to said request.

```
(QUERY-IF
 :sender ( agent-identifier :name AgenAkademik@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc ))
 :receiver (set ( agent-identifier :name AgenMahasiswa@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc )) )
 :content "MK001"
)
```

Figure 4: Communication Structure of Request for List of Course

```
(INFORM
 :sender ( agent-identifier :name AgenMahasiswa@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc ))
 :receiver (set ( agent-identifier :name AgenAkademik@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc )) )
 :content "((ListMatakuliah (agent-identifier :name AgenMahasiswa@192.168.1.5:1099/JADE :addresses (sequence http://jaim-PC:7778/acc)) (sequence (MK :kode MK021) (MK :kode MK022) (MK :kode MK023) (MK :kode MK024) (MK :kode MK025))))"
 :language fipa-sl :ontology Jadwal-Ontology )
```

Figure 5: Communication Structure of Response to Request

6. SCHEDULE MEDIUM

The Schedule Medium is draft of schedule used in scheduling. The schedule medium will be explored by the agents in order to find cells that they will occupy based on the constraints. Each cell in the schedule medium will represent a unit of time in the semester credit system and classrooms. Figure 6 displays the visual design of the schedule medium [6].

Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	1a1	2a1	3b1	4c2	5b2	. . .	mno
11							
12							
13							
14							
15							
16							
17							
21							
22							
. . .							
Xy							

Figure 6: Visualization of Schedule Medium

Using an analogy of the game of chess, the schedule medium is like a chess board. Agents will explore cell by cell with certain patterns to check for pre-determined constraints. To conduct the exploration, agents will have to possess information on the addresses of schedule medium cells. The addresses will be set based on the days and shifts – on the vertical side of the schedule medium, and based on classrooms on the horizontal side of the schedule medium. Each cell address will require a variable of day, shift, and classroom (h,s,r).

Table 1 contains information stored by the schedule medium. On table 1, the schedule medium's coding is not only based on day, shift, and room but also on capacities and which floors classrooms are located [6].

Table 1: Schedule Medium Coding

Symbol	Explanation
1mo...nmo	• n = Room number, m = category of classroom capacity (a,b,c,d), and o = floor • 1student < = a < = 20students, 20students < b < = 50students, 50students < c < = 80students, d > 80students
11...xy	The first digit represents the day, and the second digit represents time of shift (50 minutes); e.g. code 11 means the first day (Monday) and shift 1.

The schedule medium contains time shifts used. Each unit of time shift used will refer to the timing of credit semester system in Indonesia. Each time unit of the credit system represents 50 minutes. Table 2 lists a sample timing scheme using 10 shifts a day.

Table 2: Sample Timing with 10 Shifts

Shift	Time
1	08.00-08.50
2	09.00-09.50
3	10.00-10.50
4	11.00-11.50
5	12.00-12.50
6	13.00-13.50
7	14.00-14.50
9	15.00-15.50
10	16.00-16.50

7. EXPLORATION RULES

The scheduling method will refer to the following hard constraints:

- A lecturer is not scheduled on the same period for different courses or parallel-classroom courses.
- A room is not scheduled for use for more than one course or class on the same period.
- Courses on the same semester and courses taken/opted by students are not scheduled for the same period.

In addition to aforementioned compulsory hard constraints, some constraints are optional. In this research room capacity are soft constraints that would be taken under consideration in scheduling courses based on the number of students taking them. Another optional constraint is floor position. This constraint is optional for the university that has rooms in different floors. An option is also available to place senior lecturers and those with health issues in rooms located on the ground floor. Lecturer preference would be made another optional constraint as soon as it can be accommodated.

In implementing the exploration, agents will refer to rules governing exploration status and condition. Exploration status means agents are currently checking for particular constraints. Exploration condition means agents are discovering some conditions when on cells referred to with perception. Table 3 outlines a number of rules and actions that will be implemented by agents in conducting exploration.

The constraint-checking status will change if expected conditions are met. Constraint-checking will be conducted sequentially. Each failure in meeting expected conditions will result in the constraint-checking status being reverted to the initial constraint.

The process of the writing schedule components will be executed once all constraints/checking statuses are fulfilled.

Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Rules of Exploration by Agents

No.	Rules
1	**if** LocationCell(h,s,r) **and** StatusCheck=Lecture() **and** Empty() **then** ChangeRoom()← room=room+1
2	**if** LocationCell(h,s,r) **and** StatusCheck=Lecture() **and** ¬Empty() **then** ManyFinding()←LectureAvailable(LecturerCode), Changeroom()←room=room+1
3	**if** LocationCelll(h,s,r) **and** StatusCheck=Lecture() **and** Room=RoomStart **and** PositionShiftt()≠shift+Courseweight-1 **then** ChangeShift()←shift=shift+1
4	**if** LocationCell(h,s,r) **and** StatusCheck=Lecture() **And** room=RoomStart **and** PositionShift()=shift+Courseweight-1 **and** Lectureavailable() **then** ChangeShift()←ManyFinding(), StatusCheck←Lecture()
5	**if** LocationCell(h,s,r) **and** StatusCheck=Lecture() **and** Room=RoomStart() **and** PositionShift()=shift+Courseweight-1 **and** ¬AvailableLecture() **then** StatusCheck←SemesterCourse()
6	**if** LocatioCell(h,s,r) **and** StatusCheck=SemesterCourse() **and** Empty() **then** ChangeRoom()←room=room+1
7	**if** LocationCell(h,s,r) **and** StatusCheck=SemesterCourse() **and** ¬Empty() **then** ManyFinding()←FindingCourseSemester(CourseSemester), ChangeRoom←room=room+1
8	**if** LocationCell(h,s,r) **and** StatusCheck= SemesterCourse() **and** room=RoomStart() **and** PositionShiftt()≠shift+Courseweight-1 **then** Changeshift←shift=shift+1
9	**if** LocationCell(h,s,r) **and** StatusCheck= SemesterCourse() **and** room=RoomStart() **and** PositionShift()=shift+CourseWeight-1 **and** FindingCourseSemeter() **then** Changeshift()←ManyFinding(), StatusCheck←Lecture()
10	**if** LocationCell(h,s,r) **and** StatusCheck= SemesterCourse() **and** room=RoomStart() **and** PositionShift()=shiftt+Weight-1 **and** ¬AvailAllCourse() **then** StatusCheck←ListCourse()

LocationCell (h,s,r) means that the agent is on a particular day (h), shift (s), and room (r). StatusCheck will question on which constraint the checking status is at the moment. Empty() will contain a "true" value if the current cell's position does not contain schedule-determining components (lecturers and courses).

There are several stages of action that will be undertaken. When the StatusCheck is on a particular constraint, the agent move from room to room (room=room+1) until all rooms are checked. Then the agent will move to the next shift (shift=shift+1). The number of shift changes will represent the number indicating course weight.

Each time a non-empty cell (¬Empty()) is found and a search criterion discovered, it will be recorded as a finding (ManyFinding()←AvalableLecture(LecturerCode)). The agent will then change shifts as many times as the number of criteria found (ChangeShift()←ManyFinding) in order to resume checking on the same constraint.

Constraints-checking is deemed complete when all constrains are met. Each failure in checking a constraint will result in the agent repeating the constraint-checking from the beginning. When all constraint-checks are finished, the agent will write the schedule-determining components (lecturers and courses) currently being scheduled on the cells that have passed the checks. The amount of cells written/occupied will represent the numbers indicating course weights.

8. ALGORITHM FOR PROCESSING LECTURER CONSTRAINTS AND LIST OF COURSES

As previously mentioned, while exploring, the academic agent requires lecturer constraints and the list of courses taken by the students as parameters. Lecturer constraints and the list of course taken by student will be produced each by the lecturer agent and the student agent, respectively. The algorithm for lecturer constraint extraction is shown in Table 4. The algorithm for selecting and the collecting of courses list is shown in Table 5.

The algorithms assigned to each agent will be one of the agents' behaviours in addition to inter-agent communications.

Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Algorithm for Lecturer Constraints Extraction

1	Taking code_lecturer
2	**If** code_lecture<> kd_lec_constraint **then** generate *constraints*
3	**If** code_lecturer= kd_lec_constraint **then**
4	**if** st <> 0 then to the next *constraint*
5	If the end of *constraint* **then** h=0,s=0,r=0
6	**If** st = 0 then *constraint* **then** taking (h,s,r), st = 1

Table 5: Algorithm for Selecting and the Collection of Courses List

1	Taking Course that will be scheduled(Code Course, semester Course)
2	Selection of student taking Code Course
3	For each course taken by the students selected
4	Check if the semester is the same as the component that determine the schedule
5	If not, add in to the list of the course
6	If yes, ignore
7	Check if the course has already been on the list
8	If not, add in to the list
9	If yes, ignore
10	Send the list of Course

9. EXPERIMENT

The scheduling model may produce a flexible schedule pattern/combination based on the sequence of the schedule-determining components being scheduled. For this, the test used a number of criteria for sequencing schedule components. Table 6 shows the number of experiments undertaken.

Table 6: Experiments Based on Categories of Schedule-Determining Components

No	Parameter	Category
1	Teaching Lecturer	Level of structural job position
2	Teaching Lecturer	The Amount of course taught
3	Course	The bigger course weight comes first
4	Course	The smaller course weight comes first
5	Semester	Initial Semester first
6	Semester	End of Semester first

Figure 7 is a capture of the result from one of the experiments, formatted to provide a visual display of the Schedule Medium (matrix/grid). Figure 8 displays the result of experiments based on the sequence of schedule-determining components being scheduled.

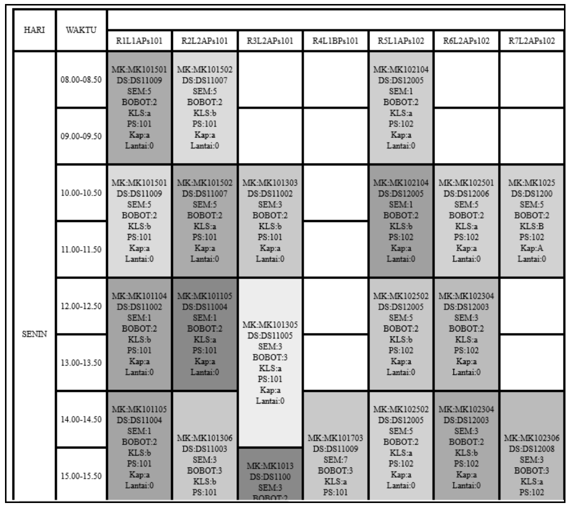

Figure 7: The Capture of the Result of the Schedule with Line and Column

Each experiment included changes to the sequence of schedule-determining components based on pre-set categories. Other parameters such as lecturer constraints and data of courses taken by students used the same data for all experiments.

The experiments only measured whether all schedule-determining components had been scheduled, no lecturers were scheduled at the same times, no courses taken by the students were scheduled at the same times, and how many lecturer constraints could be accommodated in the schedule.

```
No:2 MK:MK035 DS:DS018 KLS:a Lantai:0 Kap:b
hari=3 shift=1 ruang=6
hari=3 shift=2 ruang=6

No:3 MK:MK036 DS:DS018 KLS:a Lantai:0 Kap:a
hari=3 shift=3 ruang=1
hari=3 shift=4 ruang=1
hari=3 shift=5 ruang=1

No:4 MK:MK037 DS:DS018 KLS:a Lantai:0 Kap:a
hari=1 shift=1 ruang=1
hari=1 shift=2 ruang=1
hari=1 shift=3 ruang=1

No:5 MK:MK038 DS:DS019 KLS:a Lantai:0 Kap:a
hari=1 shift=4 ruang=1
hari=1 shift=5 ruang=1
hari=1 shift=6 ruang=1
```

Figure 8: The Capture of Experiment Results based on Sequence of Schedule-Determining Components

The experiments used 10 rooms, 10 shifts, and 6 days. The quantities of schedule components were 20 lecturers, 40 courses, and 180 students.

10. RESULT AND DISCUSSION

As previously mentioned, the results of experiments were expected to measure the fulfilment indicator. Four indicators were adopted

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

as the standard for success of the scheduling model. Three of them must be 100% fulfilled. All schedule components have been scheduled, have met all hard constraints, and no courses taken by students were scheduled at the same times. One indicator is a soft constraint to which fulfilment is optional. Nevertheless, if it had been fulfilled, this scheduling model would have gained an added value. The minimum limit for lecturer constraints achievement is 60%, but it does not mean this scheduling model is deemed a failure if the achievement were under that percentage. Table 7 displays the experiment results.

Table 7: The Experiment Result

no	All Course scheduled	Fulfilling Schedule Constraints	Fulfilling Lecturer Constraints	Courses accommodated 100% in schedule
1	√	√	60 (12/20)	√
2	√	√	55 (11/20)	√
3	√	√	60 (12/20)	√
4	√	√	75 (15/20)	√
5	√	√	70 (14/20)	√
6	√	√	75 (15/20)	√

From the experiment results, three indicators have been fulfilled for all experiment categories. Particularly for the indicator of lecturer constraints accommodation, the result varies.

The biggest values were reached on experiments number 4 and 6. This means that the sequence of schedule-determining components with the category of lowest course weight offered at the final semester was the one that most accommodated lecturer constraints in the schedule.

The smallest value was reached on experiment number 2 that involved sequencing lecturers teaching many courses (teaching more than one course). This result means that it is not recommended to use this particular category if the expectation is to accommodate as many lecturer constraints as possible. However, there is a possibility that some universities may have policies prioritizing the category.

This also means that this model can be flexible, since it can provide some categories of arranging schedules based on schedule-determining components.

11. CONCLUSION

The exploration method for schedule-formulation is still an ideal option due to its flexibility and capability to fulfil constraints sufficiently. The use of multi agents improves the capability of the exploration method. The exploration is better implemented since it is able to obtain constraints that are the parameters of the exploration process. Parameters are resulted from inter-agent communication.

The possibility of required constraints being in different systems/environments calls for the need for a good model for inter-agent communication structure. The communication structure must be built properly so as to enable agents to provide good responses and prevent them from losing data during communication.

Model Penjadwalan dengan multi agen dengan metode penjelajahan, dapat mengakomodir bermacam kategori komponen penentu jadwal. Sehingga dapat dikatakan model ini dapat menghasilkan jadwal secara fleksibel.

The sequence of schedule-determining components is key to ensuring optimum accommodation of lecturer constraints. The categories for the sequence of schedule-determining components are adjustable to different university needs. The multi-agent exploration model of scheduling can accommodate various categories for schedule-determining components. Therefore, it can be said that this model is able to produce flexible schedules.

12. FUTURE WORK

The movements of agents in searching for empty cells/slots will improve the efficacy of schedule components. The resulted schedule combinations will be affected by the development of method used in agent movements. The existing search methods can be applied to further develop the exploration method.

An alteration to the exploration method referring to lecturer constraints as a schedule-determining component can be undertaken in order to enable more thorough accommodation of lecturer constraints in the schedules. In addition, the composition of the sequence of schedule-determining constraints can be improved using certain optimization methods in order to further improve the exploration method.

REFERENCES:

[1] Hosny, M. and Fatimah, S.,"A Survey of Genetic Algoritms for University Timetabling Problem", International Conference on Future Information Technology, IPCSIT vol. 13 (2011) IACSIT Press, Singapore

[2] El-Helly, M., Abdelhamid, Y. dan Al-Wakeel, M.,"A Multi-Agent Pattern Based Timetabling System", Egyptian Computer Science Journal, Vol. 35, No. 3 September 2011

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] Mehmood, T. dan Farid, M. S."A Multi-agent Architecture for Task Scheduling In University Environment", IAES International Juornal of Artificial Intelligent, Vol. 1, No. 4, Des 2012, pp. 193-200, ISSN:2252-8938.

[4] Oprea, M., 2007, MAS_UP-UTC: A Multi-Agent System For University Course Timetable Scheduling, International Journals of Computers, Communications and Control, Vol II (2007), No. 1, pp. 94-102.

[5] Suryadi, D. and Philipus, R."Genetic Algorithm For University Timetable Planning in FTI", Proceeding of the 2012 International Conference and Industrial Engineering Operation Management, , July 3-6, 2012, Istambul, Turkey.

[6] Hakim, J and Hartati,S, Course Scheduling in University using Agent Exploration in Timetable Media, International Journal of Computer Engineering In Research Trends, Vol 3, Issue 3, March 2016, pp. 112-118.

[7] Bakir, M.A and Askop, C, "A 1-0 Integer Programming Approach to a UniversityTimetabling Problem", Hecettepe Juornal of Mathematics and Statistics, Vol 37(1), pp. 41-55, 2008.

[8] Glascheko, A., Ivaschenko, A., Rzevski, G. dan Skobelev, P., "Multi-Agent Real Time Scheduling for Taxi Companies", 8th International Conference on Autonomous Agents and Multiagent Systems, 10-15 may 2009, Budapest, Hungaria.

[9] Hakim, J and Wardoyo, R.,"Using Grid for Subject Scheduling with Multi Agent", International seminar of Information Technologi (ISIT), October 23th, 2010, Bandung, pp 147-149.

[10] Hutzschenreuter, A. K., Bosman, P. A. N. dan Blonk-Altena, I.,"Agent-based Anmission scheduling in Hospital", 7th International Conference on Autonomous and Multiagent System (AAMS 2008), pp 45-52, Estoril, Portugal

[11] Paulo, R. dan Rocco, P., "Adaptive Dynamic Scheduling in Agent Based Manufacturing Environment : A Budget Approach", Communication of IBIMA, Volume 7, 2009 ISSN : 1943-7765

[12] Russell, S. and Norvig, P., "Artificial Intelligence: A Modern Approach", 2nd ed. Prentice Hall. 2003, New Jersey, USA.

[13] Teoh, C. K., Wibowo, T. dan Ngadiman, M. S., 2013, Review of state of the art for metaheuristic techniques in Academic Scheduling Problems, Springer Science+Business Media Dordrecht 2013, 44:1–21, February 2013