Volume 3, Issue 3, March-2016, pp. 112-118 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Course Scheduling in University using Agent Exploration in Timetable Media

Jamaludin Hakim[a,1], Sri Hartati[b,2]

[a]YAPIS Papua University, Jl. Sam Ratulangi Dok V Yapis,Jayapura 99111, Indonesia
[b]Gadjah Mada University, Address,Yogyakarta, Indonesia
[1]j2hakim@yahoo.com; [2]shartati@ugm.ac.id

Abstract—Course scheduling in University is still a study area that is interesting for researchers to explore and search its optimization [4]. Scheduling optimization still depends on the needs of a University. Manual scheduling is still the most commonly used option, but it requires a skilled and expert person or person who has been accustomed to do so. However, the result still needs a lot of corrections. Course scheduling in College/University in Indonesia using semester credit system (SKS) is a challenge in making course schedule. Some scheduling methods have been proposed in several studies such as genetic methods [10], integer programming [1] and other methods using other agents [2] [7]. In the previous studies, the generated schedules must still be adjusted and or inflexible when there are changes in the schedule or its pattern, and its model cannot be applied to all Colleges/Universities in Indonesia using semester credit system (SKS). This study offers a method which is referring to the constrains/conditions in Universities but still flexible in its formation and change. Agent will explore timetable media to locate time cell/slot corresponding to the scheduling using credit system (SKS).

Index Terms—Agent Exploring, Schedule, Timetable Media, Rules, Conditions

1. INTRODUCTION

Course scheduling is still an interesting thing for researcher to study its optimization [4]. The complexity of course scheduling depends on the number of constrains/conditions to be met. Each College has its own conditions/rules in making schedule. Such rules/conditions will make the method for making schedule slightly different. Those rules/conditions will change into constrains/rules in making course schedules.

Schedule resources such as room and timing will be things that affect the making of course schedule to comply with the conditions. The lack of rooms to be used in the course schedule will make the course schedule more complex. Similarly, if available timing in the form of everyday time shift/slot is inadequate, it will cause that not all conditions can be met.

Manually course scheduling is still regarded as the most optimal method, thus, many Colleges/Universities still apply it; or the results of automatic method will be adjusted manually. Manual scheduling requires personnel having the required expertise or persons who have done it for many times and know how to make course schedule in each semester. Considering that all conditions that must be applied to the course scheduling should be done for each schedule component to be scheduled, the problem will be the accuracy when it is done manually.

Timetable media exploration method will make course scheduling manually. All conditions used in making schedule will be agent behaviors, thus, it is expected that problems in remembering conditions that are carried out manually can be overcome. The results using media exploration schedule by agent will be close or equal to the optimization conducted manually.

2. RELATED WORK

There have been many studies conducted by researchers to identify the optimization in course scheduling. Various methods have been used to formulate the course scheduling. Genetic Algorithm (GA) is one of the most popular methods in finding an optimization, including for course scheduling. One of techniques using GA is by changing course schedule components into chromosome, and then, sub-chromosome, and finally, mod-chromosome [10]. Integer Programming is also one of methods that convert course schedule components into numbers 1 and 0, in order to obtain the optimal combination in the course scheduling [1]. However, a mathematical method mentioned earlier is used to obtain the most optimal combination in order to place the course schedule component according to the applied conditions. These methods are effective in generating course schedule but some adjustments are still needed and it will inflexible when there is a change in the course schedules. In case of any adjustments or changes in the course schedule, all schedule components will be rearranged, thus, the unchanged or unmoved schedule components will still be affected by these changes.

Agent has been used in some studies related to scheduling. In the scheduling for transportation, agent will act as operator confirming the passenger pick-up and carrying in taxi scheduling in real time scheduling [3]. From these studies, agent is assigned to replace the role of taxi operator. On the activities in factories, agents can also be assigned to manage resources to be used[8]. Agent has replaced the human roles but it did not has ability as human in regulating. For the case of room booking in hospital, agent will replace the role of receptionist to arrange the booking order and the use of rooms in surgery and post-surgery process [6].

In the same case, agent is used for exam scheduling. Schedule is arranged for students in a university to be present in different semester exams. Schedule shall be set in such a way so that students will not have more than one exam at the same time [7]. This technique does not use time slot referring to the semester credit system (SKS) SKS in which each SKS is allocated with 50 minutes in classroom, on the other hand, the use of Utility-Base-Agent-System is less optimal in many spread time slots. In Tambuk University, multi-agent system is used to form a pattern of course type to be scheduled [2]. This technique has not been able to accommodate course scheduling with SKS models, in addition, this modeling cannot be applied to the courses in Indonesia.

3. AGENT EXPLORATION METHOD

Colleges/Universities in Indonesia using credit system (SKS) as its learning model requires courses scheduling in accordance with that system. Each subject has a weight which is also an indicator for the amount of time required in teaching in classroom. According to regulations, 1 credit is equal to 50 minutes in the classroom, so that if a subject has 3 credits, it takes 3x50 minutes = 150 minutes per week.

Manually, schedule draft will be formed in row and column model that represents day, time slot (1 slot = 1 credit = 50 minutes) and classroom. Schedule draft will be filled according to the available time and rooms based on course weight/credits, which means that course having 3 credits will get 3 time slots on the schedule draft. To perform these activities, personnel making schedule will manually conduct visual exploration to find a time slot corresponding to subjects and meet the conditions/constrains.

3.1. Conditions/Constraints

Indicators for the successful of making course schedule is how to meet the conditions, particularly the hard constrains. In scheduling, there are two types of constrains, the first is hard constrain and the second one is soft constrain [7]. Hard constrain is mandatory conditions that must be met so that a schedule can be used. If the hard constrain cannot be met then the schedule cannot be used. Hard constrain usually consists of three items:

a. A lecturer cannot be scheduled more than once at the same time period

b. A classroom cannot be scheduled more than one course at the same time period

c. Students at same level/semester cannot be scheduled at the same time period.

Sometimes, at the time of making course schedule, a College/University has rules or regulations other than the hard constrains that govern lecturers, use of classroom, and the students in the course scheduling. The generated schedule will meet the users' desire, which means that it is optimal from the perspective of the users if it can comply with the conditions from *the University. These conditions are called as* soft constrains. Some examples of soft constrains that are generally exist in College/Universities in Indonesia are:

a. Lecturers are not scheduled consecutively in one day, there is a lag time between one course with the other, in order to give opportunity to break or prepare materials for the next class.

b. A room can only be scheduled twice consecutively, to give opportunity for arranging or cleaning the room.

c. Students at the same level/semester should not be scheduled more than three times in one day

d. Elderly lecturers or lectures having health problems shall not be scheduled on rooms located on the top floor

e. The scheduled room should be within its capacity; otherwise, the room capacity must be greater than the number of participants.

For the conditions of choice, not all universities are same, but there are some conditions that are often used by universities based on the research result and health, such as providing opportunity for lectures to take enough break (not lag time for class movement) to refresh their mind and energy and or to prepare materials and or equipments. Conditions of choice are not an absolute requirement to be met because it will depend on resources being owned and how the College/University overcomes these problems.

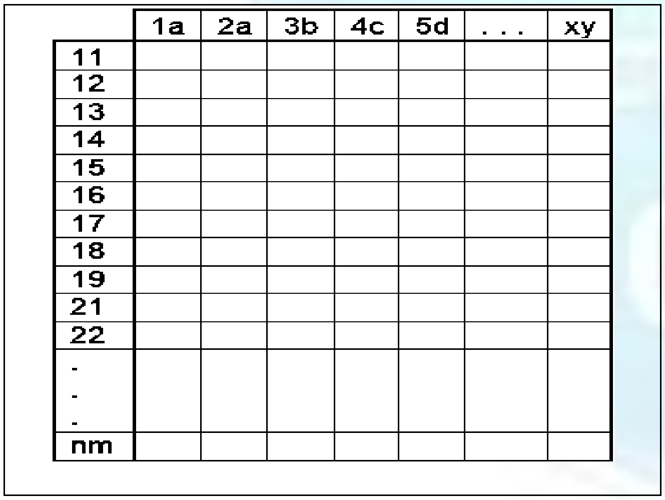
On manual scheduling, personnel will conduct a visual exploration on schedule draft to find the suitable time slot [4]. In the proposed method, agent will perform the task to explore the schedule draft and to find a time slot suitable with the applied conditions. Schedule draft becomes a very important media in using this method, thus, the schedule draft will be the media used by agent to conduct exploration, hereinafter referred to as "Timetable media"

3.2. Timetable Media

The timetable media will be like a chessboard in a chess game, while agent will be like the pawn that move from one field to another based on the rules. The timetable media will consist of rows and columns. The rows will contain information about the time slots (day and time shift) and will contain information about rooms. The visualization of timetable media is seen in Figure 1.

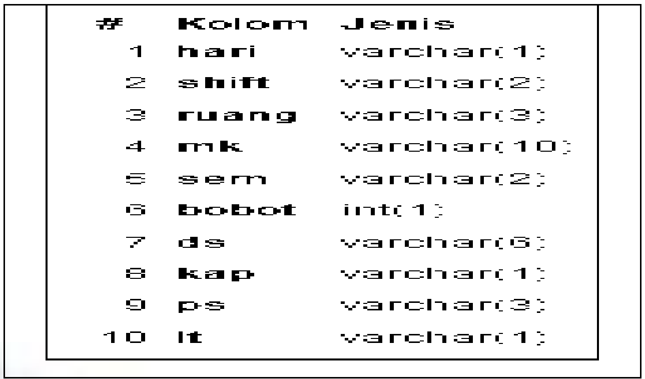
Each cell in timetable media represents a timing unit in Semester Credit System/SKS (50 minutes). The amount of shift time on the media schedule depends on each college/ university, but the average is nine or ten shifts which spread from morning until afternoon. If colleges/universities do course until night then the schedule shift will be more and there are more cells that can be used.

Referring to the visualization of timetable media in which the information will be saved and will generate a schedule, the structure of timetable media table structure will contain information mentioned previously. In addition to information about slot time and classroom, timetable media will also contain information about course, lecturers, course credit, semester/level, classroom capacity, study program and the room floor. The structure of timetable media can be seen in Figure 2



	1a	2a	3b	4c	5d	. . .	xy
11							
12							
13							
14							
15							
16							
17							
18							
19							
21							
22							
- - -							
nm							

Fig. 1. The visualization of timetable media



#	Kolom	Jenis
1	hari	varchar(1)
2	shift	varchar(2)
3	ruang	varchar(3)
4	mk	varchar(10)
5	sem	varchar(2)
6	bobot	int(1)
7	ds	varchar(6)
8	kap	varchar(1)
9	ps	varchar(3)
10	lt	varchar(1)

Fig. 2. Structure of Timetable media

3.3. Agent Exploration Model

Exploration model consists of four components. The first is user who will enter the schedule components (course code, lecturer code, course level/semester, study program and number of participants). The second one agents that will explore timetable media by themselves; the third is timetable media that will contain information about the results of agent exploration; and the last is the course schedule. Agent exploration model can be seen in Figure 3

In agent exploration model, user will enter the schedule components in which the information will be used by agents to trace cell by cell in timetable media to find a time slot in accordance with the schedule and following the conditions. All schedule components will firstly be kept in the storage space. The schedule components will be taken by agents in accordance with the order being stored.

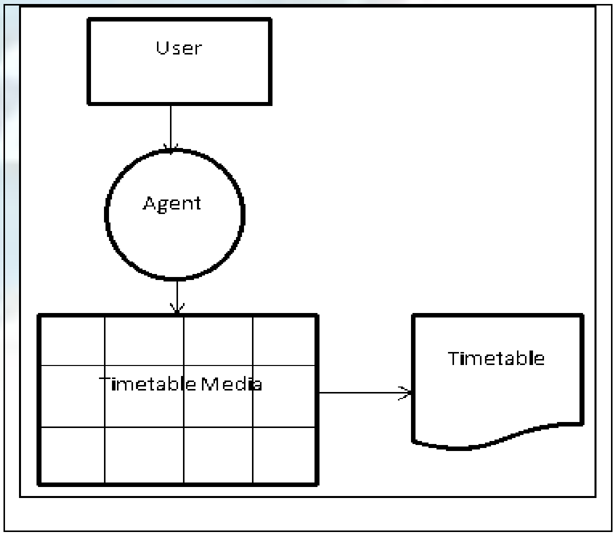


Fig. 3. Agent exploration Model

Agent will check cell by cell based on the conditions specified as agent behavior. Agent checking will be carried out section by section from the media schedule. Agent does not directly check all parts of its timetable media/environment. Section by section check if for efficiency in the condition checking. Based on agent checking activities carried out to some cells of timetable media, the type of agent used is Simple-Reflex-Agent [9]. Agent only reacts to the conditions faced at that time and takes action in accordance with the rules/conditions.

3.4. gent Exploration Algorithm

Agent exploration will be carried out from cell to cell in timetable media starting from the initial cell. Cells on media schedule will be marked with day, shift and room (d,s,r). The initial cell will be marked with 1,1,1 which means that the course is on cell of first day (Monday), first shift, and first room. The movement of agent in exploring timetable media is determined by the conditions and provisions that must be executed. Room/ horizontal movement will add room value (d, s, r ++), shift/vertical movement will add shift value (d, s ++, r), and day/vertical movement will add day value (d++, s, r).

Course will be placed by agent on the timetable media based on credits of the course. Course having three credits will occupy three shifts vertically, as well as courses having two credits. The course placement in the cells of timetable media will refer to the course placement algorithm that are adjusted with the conditions. The course placement flow is depicted in Figure 4.

Agent will execute condition by conditions owned by agent if it is in the particular cell position. Each constrain will experience with a condition of being met or not. If being met, agent will check the next conditions. If it is not met then the agent will move to other cell in accordance with the existing condition, whether it only moves to other room, switch to other room or move to other day. Here are the criteria for condition and agent movement:

1. Checking room capacity, if the room capacity does not match, then, it will be moved to the next room (d, s, r++)
2. Checking the possible/empty rooms, if impossible/not empty, then it will move to other room (d,s,r++)
3. Checking lectures who teach for all rooms as much as the weight/credits, if a lecturer was found in one room for all shifts as much as weight/credits, then the agent will move to other shift/cell vertically as much as weight/credits of the current agent position (d,s+credits/bbt,r).
4. Checking student semester/level, if the same student semester/level is found in all rooms as much as the course weight/credits, then agent will move to the next shift vertically as much as course weight/credits (d, s+credits/bbt, r)
5. Checking whether any lecturers who teach two times respectively, if lecturers are found to be scheduled at a position before the agent cell for all rooms, then the agent will move to other shift vertically (h,s++, r), if the lecturers cell is found after the current agent position as much as total weight/credits for all rooms, then the agent will move shift as much weight/credits course plus one (h, s+credits/bbt +1, r), if it is already in its final position shift, it will move to other day (d++, s, r)

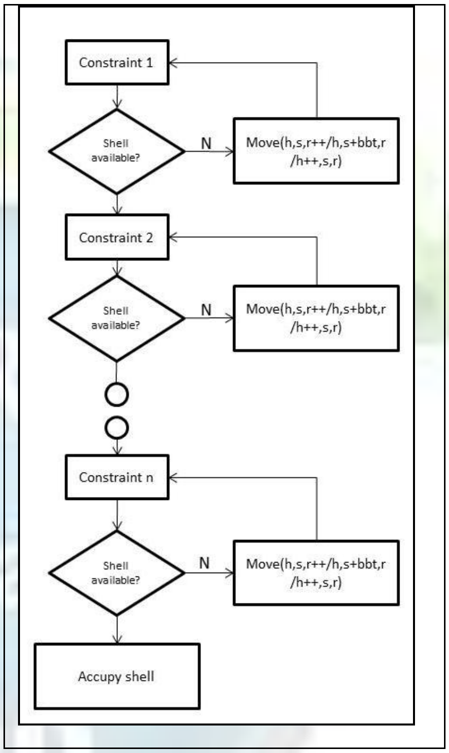


Fig. 4. Agent Exploration Flow

Based on constrains and agent movement, the following methods can be formed:

Agent movement

```
MoveRoom(room);
{
        ifroom=maxroom thenroom=1;
        room←room++;
        return room;
}

MoveShift(shift, credit);
{
        localday;
        ifshift=maxshiftthen
        {
                shift=1;
        MoveDay(day);
    }
    Shift←shift+credit;
    returnshift;
}

MoveDay(day);
{
    ifday=maxdaythenday=1;
    day←day++;
    returnday;
}
```

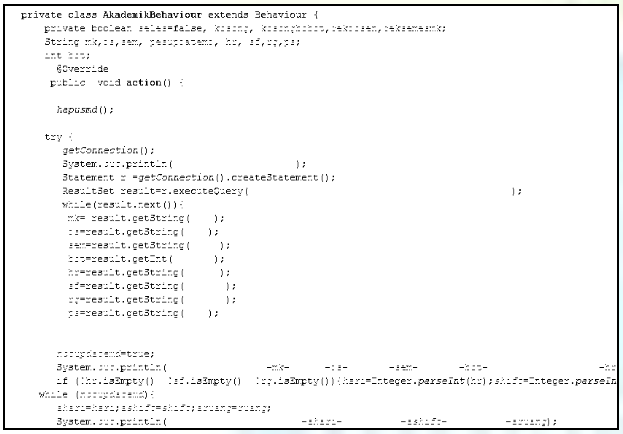
Constrains

```
RoomCapacity(day, shift, room,cap)
asboolean;
{
localroomcapasstring;
localcapresultas boolean;

    roomcap←taking room capacity in
position (day, shift, room);
    ifroomcap=capthencapresult←truee
    lsecapresult←false;
    returncapresult;
}

RoomAvailable(day, shift, room)
asboolean;
{
localroomresultas boolean;
localroomstatas string;

    roomstat←taking room volume in
position (day, shift, room);
    ifroomstat =
emptythenroomresult← false
    elseroomresult←true;
    returnroomresult;

  }

LecturerAvailable(day, shift, room,
idlec) asboolean;
{
locallecresultasboolean;
locallecstatasstring;

    Lecstat← search for lecturer
code in cell starting from (day, shift,
room) until the end room (roomax) as
much as course weight/credits (shift++
as much weight/credits);
    iflecstat =
idlecthenlecresult←false
    elselecresul←true;
    returnlecresult;
}

CourseLevel(day, shift, room, sublev)
as boolean;
{
localsublevresultasboolean;
local sublevstatasstring;

    sublevstat← search for course
level/semester in cell starting from
(day, shift, room) until the end room
(roomax) as much as course
weight/credits (shift++ as much
weight/credits);
    ifsublevstat =
sublevthensublevresult←false
    elsesublevresult←true;
    returnsublevresult;
}

LecTeach(day, shift, room, idlec)
asboolean;
{
Locallectearesultasboolean;
locallecteastatasstring;

    Lecteastat←search for lecture
code at the beginning of shift before
the current shift (day, shift--, room)
until the end room and return to the
room before this shift;
    iflecteastat =
idlecthenlecstearesult←false
    else
    {
        Lecteastat = empty;
        Lecteastat← searching for
lecture code in shift position after
the current shift is added with weight
(shif+weight+1), until the end room and
return to the room before this shift;
    iflecteastat =
idlecthenlectearesult←false
    elselecteastat←true;
    }
    returnlectearesult;
}
```

4. RESULT AND DISCUSSION

Methods that have been made will be encoded using JADE (Java Agent Development Environment) Framework. Neatbean Editor will help in making the program. JAVA programming will be the basis in making exploration agent, but the object class will use JADE library. In JADE Framework, agent will be created in a container, and each container can consist of several agents, but in this paper, the container only consists one single agent.

The conditions in which the methods have been made will be entered into object class of agent behavior. Each condition will be made separately into methods in JAVA. These conditions will be invoked in agent behavior class. In agent behavior class, a central program will be built to sort and test each condition methods. Each condition method will be run continuously as long as it generates false value and will continue to the next method if it has true value. If all method conditions have been fulfilled, then, the agent will stop by itself. It means that all value components have occupied cells on the timetable media by meeting the existing conditions. The quotation of the program code of agent behavior is shown in Figure 5.



```
private class AkademikBehaviour extends Behaviour {
    private boolean seles=false, kssmng, kssmngkskst,tekrsen,teksemesmk;
    String mk,ds,sem, pesuptatemi, hr, sf,ry,ps;
    int kst;
      @Override
     public  void action() {

      hapusmd();

    try {
       getConnection();
       System.out.println(                        );
       Statement r =getConnection().createStatement();
       ResultSet result=r.executeQuery(                                                   );
       while(result.next()){
        mk= result.getString(    );
        ds=result.getString(    );
        sem=result.getString(     );
        kst=result.getInt(       );
        hr=result.getString(      );
        sf=result.getString(       );
        ry=result.getString(       );
        ps=result.getString(    );


      ncrupdatemd=true;
      System.out.println(                   -mk-      -ds-       -sem-       -kst-                   -hr
      if (!hr.isEmpty()  !sf.isEmpty()  !ry.isEmpty()){hari=Integer.parseInt(hr);shift=Integer.parseIn
   while (ncrupdatemd){
      ahari=hari;ashift=shift;aruang=ruang;
      System.out.println(                         -ahari-          -ashift-          -aruang);
```

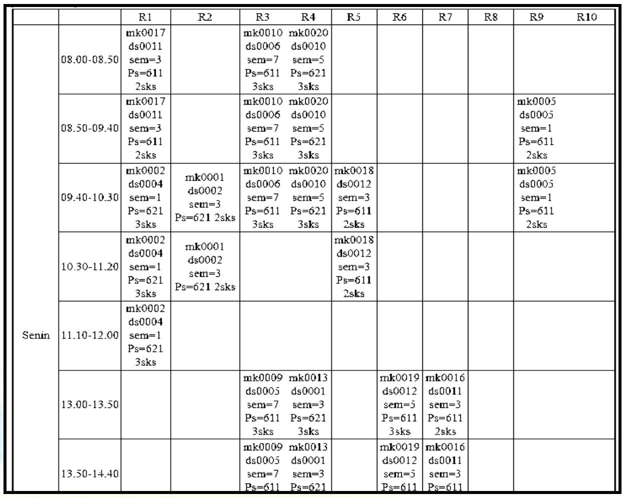
Fig. 5. Quotation of agent behavior code

In this paper, an experiment has been done using agent exploration to make a course schedule by using 10 classrooms, six days and 9 shift per day and 20 courses, 7 lecturers as the schedule components to be scheduled.

The time required for executing 5 condition is 40 seconds. The time required will vary depending on the schedule component order to be scheduled. Ordering schedule components can either accelerate or decelerate the scheduling process.

In this study, the time required to make schedule is not an indicator for the success or optimization of scheduling. The accommodated rules applied to a schedule will be an indicator for the success or optimization of scheduling using agent exploration method. The more conditions to be applied, the more complex the agent exploration process and the more time needed.

Ignore a condition of choice will be one of alternatives to make agent exploration not repeated because it does not meet its condition. The use of schedule component order to be scheduled may also be an option. The results of agent exploration experiment can be seen in Figure 6.



		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Senin	08.00-08.50	mk0017 ds0011 sem=3 Ps=611 2sks		mk0010 ds0006 sem=7 Ps=611 3sks	mk0020 ds0010 sem=5 Ps=621 3sks						
	08.50-09.40	mk0017 ds0011 sem=3 Ps=611 2sks		mk0010 ds0006 sem=7 Ps=611 3sks	mk0020 ds0010 sem=5 Ps=621 3sks					mk0005 ds0005 sem=1 Ps=611 2sks	
	09.40-10.30	mk0002 ds0004 sem=1 Ps=621 3sks	mk0001 ds0002 sem=3 Ps=621 2sks	mk0010 ds0006 sem=7 Ps=611 3sks	mk0020 ds0010 sem=5 Ps=621 3sks	mk0018 ds0012 sem=3 Ps=611 2sks				mk0005 ds0005 sem=1 Ps=611 2sks	
	10.30-11.20	mk0002 ds0004 sem=1 Ps=621 3sks	mk0001 ds0002 sem=3 Ps=621 2sks			mk0018 ds0012 sem=3 Ps=611 2sks					
	11.10-12.00	mk0002 ds0004 sem=1 Ps=621 3sks									
	13.00-13.50			mk0009 ds0005 sem=7 Ps=611 3sks	mk0013 ds0001 sem=3 Ps=621 3sks		mk0019 ds0012 sem=5 Ps=611 3sks	mk0016 ds0011 sem=3 Ps=611 2sks			
	13.50-14.40			mk0009 ds0005 sem=7 Ps=611	mk0013 ds0001 sem=3 Ps=621		mk0019 ds0012 sem=5 Ps=611	mk0016 ds0011 sem=3 Ps=611			

Fig. 4. Agent Exploration Flow

The entire result of schedule test could not be displayed because of the resulted schedule display is quite long. In the display of the resulted schedule, it can be seen that schedule components consist of course code, lecturer code, semesters, study program code and course weight/credit. The display is still in the form of code in order to observe whether the placement of schedule is conforming to the stipulated conditions. From these results it appears that certain courses will occupy cells as many as the course weights/credits being owned, similarly, a lecturer will not be scheduled to teach at the same time, so did students of the same level/semester in the same study program will not be scheduled at the same time.

One of advantages of this agent exploration method is in the schedule change or adjustment. In that process, the schedule will not undergo major changes given that most of schedule components have already obtain the right timing and room. The step for changing schedule will be the same with the process for making schedule in which only component values to be changed will be excluded from timetable media by recording the position that has been occupied, so that at the time of exploration, it will search for the time slot and the appropriate room, a position that has been recorded will be ignored.

5. FUTURE WORK

This study is still very far from complete, even, it can be said as a preliminary study. This method is also highly dependent on the component ordering to be scheduled. Ordering using certain methods such as larger weight/sks or the lectures being scheduled in advance can be additional research.

6. REFERENCES

[1] 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.

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

[3] 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.

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

[5] 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

[6] 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

[7] 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.

[8] 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

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

[10] 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