International Journal of Computer & Information Technologies (IJOCIT)
www.ijocit.ir & www.ijocit.org
ISSN = 2345-3877

Generation of E-Courses According to Educational Needs of Each Student Using Automated Planning

Fatemeh Zolfaghari[1], Amin Rahati[2]
MS graduate of computer sciences[1], Assistant Professor of computer sciences department[2]
nady.zolfaghari@gmail.com[1], A.rahati@cs.usb.ac.ir[2]

Abstract: Generation of an educational pathway according to learning features of each student is one of the important issues in electronic education, because it reduces the wastage of time and increases the usefulness. Achievement of the aims of such an issue is difficult considering limited sources such as education time. In this research, we present a method for solving this problem, using automated planning technique and one of the educational theories. The planning actions in our suggested method, include each course learning activities each of which has usefulness (according to each considered student's characteristics) and cost (education time). Also, we have classified the activities in a way that, for teaching each course, accomplishing at least one activity of each class is necessary. The experiments suggest the planner high ability in finding educational plans which are the most useful for the considered student.

Keywords: Artificial intelligence, E- learning, Automated Planning, Educational pathway

1. Introduction

What is of special importance in an electronic education system is to provide the required educations to the users of the system and to make sure that students have received these educations. The main problem in today education is not the availability of more information; in fact one of the students' challenges is to make the context they face meaningful and to absorb all the information objectively. Because of the high amount of information and its progresses as well as lack of time in education, we need new approaches in order to find the best educational pathway [1].

Automatic planning is one of the new approaches in generating educational pathway in electronic learning. Automated planning is a branch of artificial intelligence science in which the planner finds a sequence of applications the accomplishment of which results in the program goals. Therefore, the necessity of finding such a plan is searching in a space involving all possible states which, in case of real world problems, is very large and complex. The planner major problem is the large searching space that researchers have tried to decrease it by different techniques [3].

In this study, we have generated plans well-adjusted to the students learning characteristics using automated planning. The limited sources problem was first introduced by Mr. Smith in 2004 [18]. In such cases, planners do not seek for a plan to achieve all the goals, but rather they find a plan whose achieved goals have the most possible number of profits. Many of the real world problems belong to the limited sources scope. For example, a planet rover follows many scientific goals in each mission. But it can perform only some of the given orders in each tour around the planet surface, because of limitations in energy and time [15]. Our problem also describes the kind of limited sources problem that indicates there is not the possibility of choosing all learning activities in course plan, because of time limitation.

Our problem is similar to limited sources problems for the following reasons:

- Each activity in the teaching course has reward and time of doing.
- The reward of giving learning activities to the student depends on his/her learning characteristics.
- The performance time for all learning activities in final plan must be less than the considered time for the course.
- The student must do some activities in the learning process, before some other activities. So in planning, activities are modeled to actions and the reasoning relation between them comes at the precondition of each practice.
- We have classified course learning activities in a way that activities having the same educational results are ascribed to one class and the final plan involves at least one activity of each class.

The problem we want to solve by automated planning is generating educational pathway according to the learning characteristics of each student. Therefore, each course has been defined by a set of different learning activities. So in the automated planning, learning activities have been converted to actions in planning. We sought for a plan that considers the relations among learning activities and the contrast between benefit and cost in each action, in addition to the applicability and the reward of the plan. This objective has been achieved via defining the appropriate criteria in planning.

In previous educational researches, appropriate methods were presented for evaluating the reward of each learning activity. Each activity reward rate has been calculated by using these methods and identifying each student's learning style.

The rest of the article is as follows:

The second part introduces the automated planning and the planning language. The third part expresses modeling method of the problem in planning. In the fourth part, experiments related to evaluating the correctness of this method are reported. And, at last, the fifth part describes the conclusion.

2. Automated Planning

A planning is choosing an arranged sequence of actions in a way that the actions performance respectively results in the planning goals. The planner input includes a set of actions, a set of goals and the initial state. The planner output is a sequence of choosing actions by the planner their respective execution of which turns the system from its initial state to the determined goals. Planning is defined by two elements: first, the planning domain which includes a set of actions, next, the planning problem which includes the initial state and a set of planning goals.

In the problem world, each state is described by a set of fundamental predicates and functions, and each action is expressed via the parameters, preconditions and effects of the action. Preconditions have predicates which must have a truth value before doing the activity and effects include predicates which take a truth value after doing the activity. Therefore, executing each activity adds to or lessens predicates from the problem world and changes the status of the problem world.

PDDL is the standard performing language for the automated planning. This evolved language is the primary language of planning (strips). ADL volume, related to PDDL, has expressed the activities with negative conditions and conditional effects [9]. Negative conditions can be indicated by adding "not" before conditions. Conditional effects will be used by adding the keyword "when" as follows:

When *(condition) (effects of predicates).*

In the PDDL volume 2/1 the possibility of using numerical values and metrics and fluents were provided [16]. The metrics of the plan allow for the analysis of the plan results. Fluents are functions the values of which increase or decrease during the execution of each activity. In the present article, we have defined fluents for calculating the course performance time the initial value of which is zero and the amount of the considered fluent value increases by executing each activity proportional to the time for its learning activity. Also the PDDL volume 3, has considered logistics for defining soft goals. Soft goals are those the achievement of which is not necessary by the final plan. Each soft goal is indicated like a priority as follows[7]:

(preference <id> <goal>)

Here <id> stands for the priority identification and <goal> stands for the goal. So we rank the goals in this way so that goals most useful for the plan are accomplished first. In this article, we have used classic display status. For example, we point to blocks problem in figure 1. In this example, we have blocks A, B, C, T table and R robot.

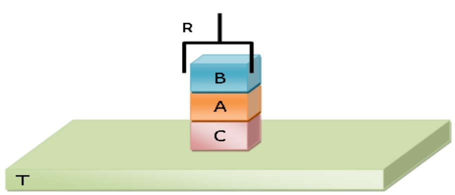

Figure1: An example of the Planning Problem (blocks)

As an instance, we will show the **unstack** activity defined in table1 in the planning domain

Action	The result of execution this action
Unstack ?x_block ?y_block	block x is on the block y and takes block x robot
Stack ?x_block ?y_block	Robot R has taken block X and put it on block Y
Pickup ?x_block	robot R picks up block X from the table
Putdown ?x_block	robot R which has taken block X put it down on the table.

Table 1: Applicable Actions in Blocks Problem

as follows:

```
(:action  unstack
  :parameters (?x_block ?y_block ?r_robot)
  :precondition (and(on ?x ?y)(empty ?r)(clear ?x))
  :effect (and(not(on ?x ?y))(not(empty ?r))(hold ?x)
(clear ?y)(not(clear ?x))))
```

If we define the initial state and the goal in the planning problem as follows:

```
(:init (ontable c)(on A C)(on B A)(clear B (empty R))
(:goal (and(ontable A)(on B A)(on C B)(clear C)))
```

Then, the following plan will be a solution for the above problem:

```
(unstack B A)(putdown  B)(unstack A C)
(putdown A)(pickup B)(stack B A)(pickup C)
(stack C B)
```

3. Modeling a Problem in PDDL

The first step in electronic education is modeling a course. Each course is described via learning activities. Here, we will express learning activities modeling in PDDL language. First, we will classify learning activities in a way that, for learning a course, the students must do an activity from each learning activity class. Also, more activities can be chosen from that class to learn a subject better.

Each student has its own learning style for learning a course so that the proportionality of the course learning activities for him/her depends on these features. So the educational theory of Richard Felder as shown in table 2 is used for realizing the proportionality of activities with the considered student's learning style [16]. Felder has classified people's learning style in four dimensions, namely the perceptual dimension (sensitive and intuitive), processing information dimension (operative and reflective), input dimension (verbal and visual) and comprehension dimension (sequential and Global).

Learning styles / Learning source types	Processing		Perceptional		Comprehension		Input	
	Active	**Reflective**	**Intuitive**	**Sensitive**	**Sequential**	**Global**	**Visual**	**Verbal**
Lecture	0	2	2	0	0	0	0	2
Narrative text	1	2	2	0	0	0	0	2
Slide	0	1	1	1	1	2	2	1
Table	0	0	0	1	1	1	1	1
Index	0	0	0	0	1	2	0	1
Diagram	0	1	1	1	1	2	2	0
Figure	0	1	1	1	1	2	2	0
Graph	0	1	1	1	1	2	2	0
Exercise	1	1	2	1	1	0	1	2
Simulation	2	0	0	2	0	1	2	0
Experiment	2	1	1	2	1	0	2	0
Questionnaire	1	0	0	0	1	0	0	0
Problem statement	2	1	1	2	0	0	1	2
Self assessment	1	0	0	0	1	0	0	0
Exam	1	0	0	0	1	0	0	0

Table2: Rating Felder's Learning Styles in 15 Different Learning Sources

According to table 2, we have calculated the reward rate for each learning activity based on the learning source and the considered student's characteristics. So each activity has an education time and reward which constitute the planning problem inputs. Each activity has been modeled practically in the planning domain with its own reward and education time. The planner chooses actions which provide the maximum reward set in the time scope of doing the course for the considered student. The modeling a course as a planning problem will be explained in 3 steps, namely modeling the student's learning style in the planning problem, modeling activities in the planning domain and defining the appropriate metrics in planning.

3.1. Modeling the Student's Learning Style

At first, we obtain the student's learning style by presenting a questionnaire to him/her. After realizing his/her learning style, we model each style by a predicate in PDDL problem file as follows [7]:

(sequential ?s_student p_< profile_level_type >)

Instead of <profile-level-type>, strong, moderate or balanced levels can be put. For example, if a student is strong in the comprehension dimension, in sensitive style, and in input dimension, in visual style, the following predicates will be added in the initial state in the planning problem:

(Visual student strong)(Sensitive student strong)

In table 2, each activity belongs to one of the learning sources like slide, graph, text and etc. The educational effect of each learning activity depends on its learning source and the student's learning style. For example, if a student has verbal style, an activity with lecture learning source is more appropriate for him, while this activity is not so much useful for a student with visual learning style. We have defined reward function to determine the proportion of one activity with the student's learning features. For each learning style, a conditional effect will be added in each action in the planning domain file, which adds the reward function to the calculated rate based on the considered action as follows:

(when (<style> ?s strong)

(increase (reward_student ?s <v>)

Where the student's learning style is put instead of <styles> and the calculated reward is put instead of <v>. We have indicated the reward function with the fluent (reward-student? s). We have used Felder's table for calculating the reward value of each action. In this table, rows show the learning sources and columns show the learning styles and they are expressed by the 3 values 0, 1 and 2. For example, at the first row with lecture learning source, the verbal learning style has the value 2 and the active style has the value 0 which equal to very good and ineffective, respectively. In this study, we have assumed the maximum reward of 120 for each learning activity and the reward value of very good in Felder's table is twice as good and the reward rate of ineffective has been considered as zero. For calculating the reward of an activity, at first, we have counted non-zero values at the row related to the learning source and have saved it in the variable **d**. In each learning style with very good value, the reward increase rate equals to the result of 120 divide by **d**, with good value, it is half of this amount and with ineffective value, the reward is zero. For example, suppose that a student is strong in reflective, intuitive and visual learning styles. If we choose the graph learning source of educational activity, then in Felder table, in the row related to the graph learning source **d**=6 and the reward in reflective style is 10, it is 10 in the intuitive style and is 20 in the visual style. So, the conditional effects appropriate to this action will be written as follows [7]:

(when (reflective ?s strong) (increase(reward_ student ?s) 10))

(when (Intuitive ?s strong) (increase(reward_ student ?s) 10))

(when (visual ?s strong) (increase(reward_ student ?s) 20))

3.2. Defining the Planning Domain

For each learning activity we define one action in the planning domain. We will prevent the repeated choosing of actions in the plan by using the following predicate:

Task_ action name_done

The fluent (total_time_student ?s) has been used for calculating the plan performance time by the student. Choosing each action increases the value of this fluent as much as the learning time of its proportional action. The fluent (reward_student ?s) is used for calculating the total reward obtained from the chosen actions. The value of this fluent increases as much as the calculated reward of its proportional action by choosing each action in the plan, (based on the student's learning style and the action learning source). There are two types of relation between the learning activities of the course, namely pre-conditional and co-conditional. If the relation between some learning activities is of the type pre-conditional, all the learning activities in the pre-conditional relation must be done in order to accomplish a new learning activity. The co-conditional relation between several learning activities expresses that in order to do a new action, doing one of the learning activities is enough? We have expressed the co-conditional relation with "or" and the pre-conditional relation with"and" in the action precondition in the planning domain. Figures 2 and 3 show these relationships for the learning activities in PDDL.

```
(:action  Gmp_6
    :parameters (?s_student)
    :precondition (and (not(task_Gmp_6_done ?s)
                         (or(task_unification_6_done ?s )
                            (task_unification_5_done ?s )
                            (task_unification_4_done ?s )
                            (task_unification_3_done ?s )
                            (task_unification_2_done ?s )
                            (task_unification_1_done ?s)))
    :effect (and (task_Gmp_6_done ?s)
                 (increase(total_time_student ?s) 10)
                (when(Reflective ?s )
                     (increase (reward_student ?s) 30))
                 (when(active ?s )
                     increase (reward_student ?s) 15))
                (when(Intuitive ?s )
                     (increase (reward_student ?s) 30))))
```

Figure 2: An Example of PDDL Action With Co-Conditional Relation

```
(:action  Exam_1
   :parameters (?s_student)
   :precondition (and (not(task_Exam_1_done  ?s))
                      (task_Resolution_1_done  ?s))
   :effect (and (task_Exam_1_done  ?s )
                (increase(total_time_student ?s)10)
                (when(active ?s )
                     (increase (reward_student ?s) 30))
                (when(sequential ?s )
                     (increase (reward_student ?s) 30))))
```

Figure 3: An Example of PDDL Action with Pre-Conditional Relation

In figure2, GMP action is for teaching the generalized modus ponens of the artificial intelligence course. To avoid the repetitive choosing of an action in the plan, the predicate *(not (task_Gmp6_done))* is used. A student must have learned unification in order to learn this problem. Unification is explained in six learning activities with different sources, of which doing one is enough for the student. This has come with "or" in the action precondition. The considered time for GMP action is 10 and the fluent time value increases with the following predicate:

(increase(total_time_student ?s)10)

The reward of each learning activity in the action will increase in accordance with the learning source and the student's learning style. For example, if a student benefits the intuitive learning style, the reward function with *(increase (reward_student ?s))* increases up to 30. Figure 3 indicates the exam action in whose precondition, the pre-conditional relation is considered. For choosing this action in the plan, both precondition predicates should have a truth value. The last action in the planning has been called the finish_course. This action is shown in figure 4. We put tasks required for the course education in the precondition of this action.

```
(:action  finish_course
    :parameters (?s_student)
    :precondition (and (not(task _finish_course_done ?s))
                       (task_Exam_1_done ?s)
                       (>= (reward_student ?s) (total_reward ))
                       (<=(total_time_student ?s)(total_time)))
    :effect (and (task_ finish_course_done  ?s )))
```

Figure 4: An Example of The Last Action in PDDL Domain

The equation *(<=(total_time_student ?s) (total_time))* is considered to avoid the plan time exceed from the course time. The fluent *(total_time_student)* is calculated during choosing actions in the plan and total_time (total course learning time) will be first measured in the problem file.

The predicates *(>(reward_student ?s) (total_reward)* are used to compare the obtained reward by the plan with the determined minimum. The minimum reward *(total_reward)* is obtained based on the chosen actions with the maximum reward of each class which will be explained in the next section. Our planning problem has one goal *(task_finish_course_done)* which will be achieved by the last action.

4. Defining the Appropriate Metrics in the Planning

In the mentioned planning problem, we have used conditional effects and fluents to find a solution. Planners such as the Metric-FF which use PDDL volume2.1, present acceptable, but not the best, solutions [10], because these planners work based on the minimum chosen actions. Our problem objective is increasing the reward of educational pathway by choosing the maximum possible learning activities based on the course learning time limitation. On this basis, we have created a metric in the planning to make the planner choose actions with the maximum reward. A metric, is a variable defined in the metric planning that the planner tries to choose a plan among the possible plans in which the metric is in its minimum rate.

Learning styles in table 2 have three value levels, namely very good, good, and ineffective for different learning sources. If an activity with lecture type is very good for a student with verbal learning style, there is no reason for it to be very bad for a student who does not have the verbal style. It means that we can not comment about each learning style contrary. So we could not creat an appropriate metric by reversing the reward function or minimizing the time in the planning. Thus, we added the fluent penalty_student to the planning problem as a metric. This fluent increases by one unit if the chosen action in the plan, has a value other than 2 in Felder's table, based on the student's learning characteristics and the action learning source. We have defined the plan metric with the predicate *(minimize (penalty_student ?s))* in the planning domain. So the planner tries to choose the maximum actions which have the most reward for a particular student. Another metric is the minimum reward the plan must contain which is defined by the formula number 4. We choose actions from each class that have the maximum reward for the student and put them in the set B.

(**m** indicates the class number of activities)

(1) $B=\{b_1,b_2,\ldots,b_m\}$

(2) $\forall b_i \in B,\ b_i=<t_i,r_i>$

(3) $MaxR=\sum_{i=1}^{m} r_i$

Since 120 is the highest reward rate for each activity, total_reward rate used in the precondition of the last action in the planning domain is calculated by using MaxR.

(4) $total\_reward=MaxR-(120\times penalty)$

5. Experiments

We have executed the experiments on three different courses of artificial intelligence with different rates as follows:

- The first course, intelligent agents with 54 actions in 20 clusters.
- The second course, problem solving by searching, with 86 actions in 30 clusters.
- The third course, constraint satisfaction problems, with 20 actions in 14 clusters.

For each cluster, we have considered maximum 6 actions with different learning sources, in a way that the defined actions include teaching each part of the course by different styles such as slides, practice, reading a text and etc. In these experiments, we have used three students with different learning styles, intuitive, verbal and intuitive-verbal. After creating the course actions, we solved the planning problem by using the Metric-FF planner, according to the created actions [11]. This planner has used the

Hill climbing algorithm without defining the metric and has solved the problem. By defining the metric, the metric-FF planner uses Best First Search algorithm (BFS) which is a costly algorithm. Based on the domain rate and the PDDL features used (fluent, numerical value in preconditions, “or” and “not” in preconditions and conditional effects), one among the desired planners must be chosen. Despite BFS algorithm cost the metric-FF is still the best existed metric planner for solving our problem by using the metrics. We have done the experiments in four states and have compared the results. Experimental cases are as follows:

1. FF-EHC: the metric-FF planner with hill climbing algorithm without using a metric.
2. FF-BFS-penalty: the metric-FF with BFS algorithm and using the fluent “penalty” as the metric.
3. FF-BFS-time: the metric-FF planner with BFS algorithm and using total-time as the metric.
4. FF-BFS-reward: the metric-FF planner with BFS algorithm and using the metric “penalty “and the minimum reward.

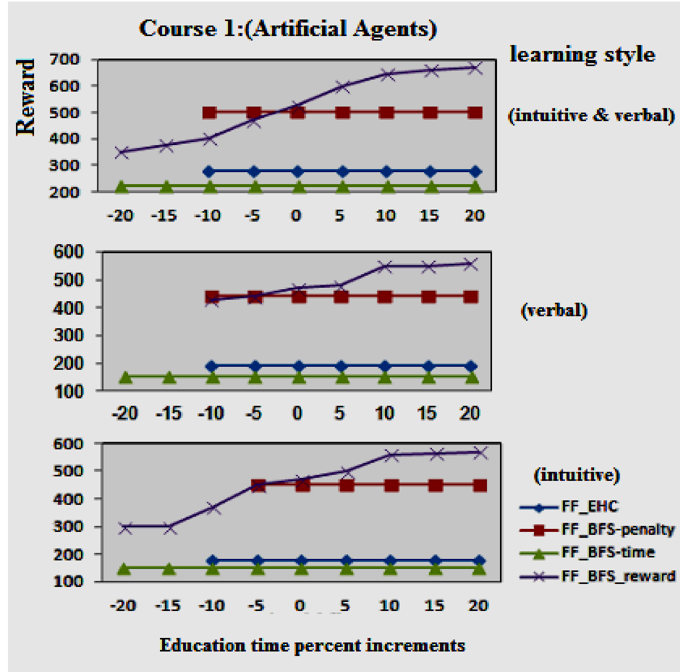

Figure 5: **Total Reward Obtained for Course1**

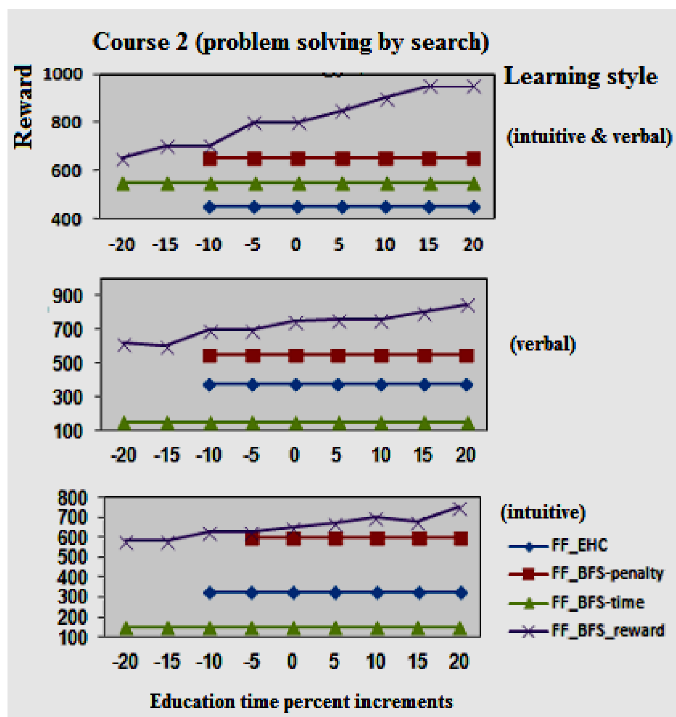

Figure 6: **Total Reward Obtained for Course2**

International Journal of Computer & Information Technologies (IJOCIT)
Corresponding Author: *Fatemeh Zolfaghari*
August, 2014 Volume 2, Issue 3

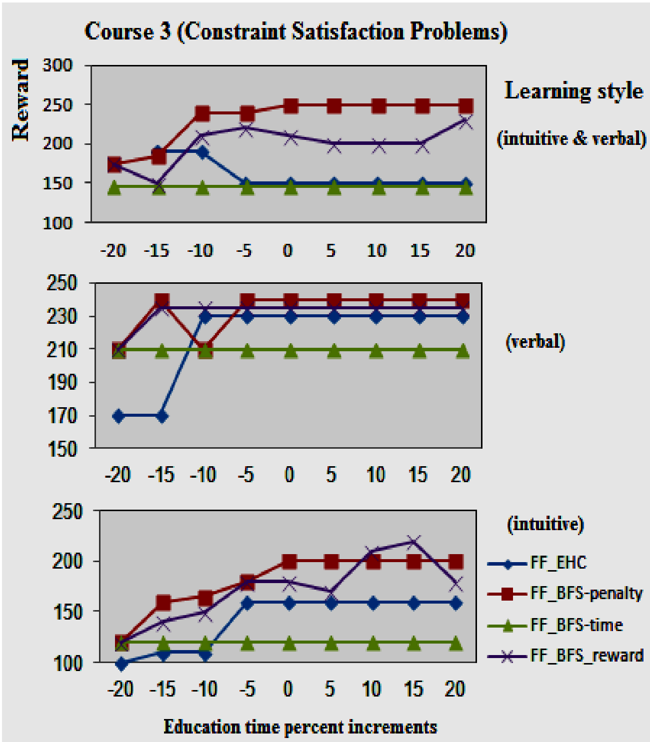

Figure 7: Total reward Obtained for course 3

We have considered a special educational time for each course. For example, Artificial Agents course has first 90 minutes time. We have increased or decreased the time to 5, 10, 15 and 20 percent to study the effect of course educational time. The experiments results show that information obtained from the minimum efficiency and penalty metric has increased the system reward significantly, especially when the course education time is large. The greatest difference among the experimented systems was seen in courses with the greatest number of actions and clusters. The third course which is shown in figure 7 has only 20 actions with 14 clusters and when the course time is low, the planner has not found an appropriate solution. By using the minimum reward, the planner does not find any solution if the course activities are many and the time scope is low.

This issue is shown in figure 5, course1. The best results have been obtained in courses with many activities and clusters and a planning with the maximum selectable activities and by considering the minimum reward to which the plan must reach. This is shown by graphs in figure 5 and 6.

6. Conclusion

The present research result is to generate the educational pathway according to different students' learning features by using the automated planning technique. Our goal is to find a course plan with the most benefit (educational reward) and the least cost (in terms of time) for the considered student. Therefore, we have modeled this problem like a limited sources classified problem and have considered the contrast between profit and cost of each learning activity by defining appropriate metrics in the planning. The planner output is an educational pathway including a systematic sequence of learning activities, based on the relations between them with the maximum reward for the student. When the plan includes one activity from each category, the metric-FF planner stops the searching without using the metric, although there is enough time for choosing more activities and increasing the student's knowledge. The planner continues to choose activities by using the minimum reward in the planning till it obtains the minimum determined reward. Even when the time of the course learning is low, this planner finds a solution for the problem. The best results are obtained by the metric-FF planner with penalty metric and by considering the minimum reward in the plan especially when the course size (the number of categories and activities) is large.

References

[1] Baldiris, S., Santos, O., Barrera, C., Gonzalez, J., Velez, J., & Fabregat, R.”Integration of educational specifications and standards to support adaptive learning scenarios in adaptaplan. Special Issue on New Trends on AI techniques for Educational Technologies”. International Journal of Computer Science andApplications(IJCSA),Vol.4,NO.3,2008.

[2] Bylander, T. “The computational complexity of propositional STRIPS planning.Artificial Intelligence”, Vol.69,1994,pp: 165–204.

[3] Castillo, L., Morales, L., González-Ferrer, A., Fdez-Olivares, J., Borrajo, D., & Onaindía, E. “Automatic generation of temporal planning domains for e-learning problems”.Journal of Scheduling, Vol.13,NO.4,2009,pp: 347–362.

[4] Feillet, D., Dejax, P., & Gendreau, M. “Traveling salesman problems with profits”.Transportation Science, VOL.39,NO.2, 2005,pp:188–205.

[5] Felder, R. M. “Matters of style”. ASEE Prism,VOL. 6,NO.4,1996,pp: 18–23.

[6] Fuentetaja, R., Borrajo, D., & Linares, C. (2009). A look-ahead B&B search for cost-based planning. InProceedings of CAEPIA’09(pp. 105–114).

[7] Fernandez, S., Borrajo, D., "using linear programming to solve clustered oversubscription planning problems for designing e-courses.”, Expert System and Applications , vol.39, 2012, pp: 5178–5188.

[8] A. Garrido, E. Onaindia, L. Morales, L. Castillo, S. Fernandez and D. Borrajo, "Modeling E-Learning Activities in Automated Planning", ICAPS -09 International Competition on Knowledge Engineering for Planning and Scheduling, Thessaloniki Greece, 2009.

[9] J.A. Lozano, J.A. G´amez, and J.A. Moreno “Planning and Execution in a Personalised E-Learning Setting”, CAEPIA, LNAI 7023, 2011, pp: 233–242.

[10] Gerevini, A., & Long, D. (2005). Plan constraints and preferences in pddl3 – The language of the fifth international planning competition. Tech. rep.

[11] Morgan Kaufmann. Hoffmann, J. “The Metric-FF planning system”: Translating ‘‘ignoring delete lists’’ to numeric state variables. JAIR, 20, 2003, pp:291–341.

[12] Sanchez Nigenda, R., & Kambhampati, S. “Planning graph heuristics for selecting objectives in over-subscription planning problems”. InProceedings of ICAPS-VOL.05,2005, pp: 192–201.

[13] Smith, D. “Choosing objectives in over-subscription planning”. In Proceedings of ICAPS, VOL.04,2004, pp: 393–401.

[14] Cristea, A., & Garzotto, F. (2004). “Adapt major design dimensions for educational adaptive hypermedia”. InED-Media ’04 conference on educational multimedia, hypermedia & telecommunications. Association for the advancement ofcomputing in education.

[15] Do, M. B., Benton, J., van den Briel, M., & Kambhampati, S. (2007).” Planning with goal utility dependencies”. InProceedings of the 20th international joint conference on Artificial Intelligence (IJCAI-07)(pp. 1872–1878). Hyderabad, India

[16] Pednault, E., (1989). “ADL: Exploring the middle ground between strips and the situation calculus”. InProceedings of the conference on Knowledge Representation and Reasoning (pp. 324–332)..

[17] Fox, M., & Long, D. “PDDL2.1: An extension to PDDL for expressing temporal planning domains”. Durham, UK: University of Durham,2002.

[18] Ghallab, M., Nau, D., & Traverso, P. “Automated planning – Theory and practice”, 2004.