

# A Voting-Based Agent System for Course Selection in E-Learning

Ali M. Aseere  
*School of Electronics and  
Computer Science  
University of Southampton,  
UK*  
ama07r@ecs.soton.ac.uk

Enrico H. Gerding  
*School of Electronics and  
Computer Science  
University of Southampton,  
UK*  
eg@ecs.soton.ac.uk

David E. Millard  
*School of Electronics and  
Computer Science  
University of Southampton,  
UK*  
dem@ecs.soton.ac.uk

## Abstract

*Agent technology is a good approach for solving a number of problems concerned with personalized learning. In personal learning contexts individual students are given an environment that takes into account of their needs, interests and aspirations, and this is intended to lead to an enhanced learning experience. The aim of this paper is to show how agent systems can not only form a good framework for distributed e-learning systems, but also how they can be applied in personal learning contexts where the learners are autonomous and independent. We present an e-learning scenario where students try to register for their preferred courses but where courses will only run if enough students register. In this context, we introduce, for the first time, a prototype of an agent-based voting system in e-learning, where autonomous software agents vote on behalf of the students. We present a number of different voting strategies that student agents could use. Finally, through simulation we empirically investigate the resulting satisfaction of the students in the system.*

## 1. Introduction

Multi-agent systems are being used in a wide variety of applications, ranging from comparatively small systems for personal assistance, to open, complex, systems for industrial applications [2]. In e-learning, multiagent systems appear to be a promising approach to deal with the challenges in educational environments. They can provide new patterns of learning and applications, such as personal assistants, user guides and alternative help systems, which are helpful for both students and teachers [6]. It has been argued that using multiagent systems to design

educational systems can lead to more versatile, faster and lower cost systems [10].

Agent technologies could allow us to take this personalization to new levels. In particular, consider an online university that has an open enrolment for adult learners to work towards a qualification (or a given set of skills needed for a particular job). Adults seek courses to match their own requirements, but the university can only run courses that have sufficiently high interest. An agent framework enables the students and university to negotiate which courses students select, and therefore which courses will run.

In this paper, we present a multiagent system for this e-learning scenario based on voting theory (where the number of candidates corresponds to the number of courses available that student can vote for), where an autonomous software agent votes on a student's behalf according to the student's preferences. In so doing, we are the first to apply voting procedures in an e-learning scenario. In particular, we introduce a novel voting protocol, consisting of multiple rounds that allows the student agent to accurately represent the student's preferences, and that can learn from previous rounds. Furthermore, we introduce a number of different voting strategies that student agents could use, and examine the resulting student satisfaction (which measures how well the courses that are running match the preferences of an individual student). Through a simulation, we evaluate our multiagent system and the proposed strategies. The objective is to investigate whether voting procedures in particular and multiagent technology in general could potentially replace a centralized infrastructure (where the selection of courses is determined directly by the university), and to explore the impact of agents using different strategies.

In the rest of this paper, Section 2 describes the background, focusing on voting approaches and their relevance to agent technology. Section 3 introduces the

prototype of the voting agent based. Section 4 describes a number of experiments to test the performance of three different voting strategies and analyses the results. Section 5 concludes the paper.

## 2. Background

Voting theory is part of the general area known as *social choice*, which is concerned with procedures for making collective decisions that maximize the *social welfare* (the sum of welfare of individual agents), while at the same recognizing that agents are self-interested and act in a way that maximizes their own individual preferences. Likewise, in our e-learning scenario, the university would like to choose the best overall set of courses, while each student would like their most preferred courses to be selected. In the remainder of this section we first provide the necessary background on voting theory. We then go on to review related papers that use agent-based technology in e-learning.

### 2.1. Voting procedures

Social choice theory is an active area of research in multiagent systems that enable decisions in a decentralized way. The designer of a voting system is concerned with analyzing and designing the mechanisms that are used for collective decision-making. In the last two decades this field has increasingly been the area of investigation by computer scientists, and especially researchers in multiagent systems [8]. This is because agents are inherently autonomous, and may have different and conflicting goals. At the same time, each agent would like to maximize its own utility. In such a setting voting systems provide an appropriate solution to reach a socially desirable decision, while taking into account individual preferences [11].

A voting system applies a set of rules that govern how votes are cast in an election, how they are aggregated, and how winners are determined. In the simplest system, each voter has one vote and the single candidate who receives the most votes, irrespective of the percentage of these votes among the total number of votes cast, is declared the winner. This system, known as *first past the post*, is used for example in elections for the UK Parliament. However, there are many alternative voting systems, such as the *Borda count* (where each voter is given multiple points, and the candidate with the most points wins), and approval voting (where a voter can vote for as many candidates as they want).

The above are examples of voting systems where there is only a single winner. However, in our scenario

we need a voting system that allows for *multiple* winners (since multiple courses need to be selected). Again, many possible procedures exist. For example, in the *single transferable vote* (STV) system, each voter provides a ranked list of candidates according to its preferences [3]. The winner determination process then proceeds in several rounds. In each round, all votes for the most preferred candidate are counted, and the candidate who has received the least number of votes is eliminated. Then, anyone who has voted for the eliminated candidate as their first preference now has their second preference allocated as their first preference. The process then repeats until the required number of winners remain [4]. The aim of this procedure is to minimize wasted votes and to promote proportional representation.

Another voting procedure used for multiple winners is *cumulative voting*. Here, each voter receives a number of points (usually the number of points is equal to the number of candidates), and they are free to choose how many points to allocate to each candidate. The candidates with the highest cumulative points are selected as winners. The advantage of cumulative voting (compared to e.g. STV) is that it allows an agent to better express its preferences. For example, if an agent prefers only the first candidate but has no interest in any of the others, then it can allocate all its points to the first one. This is not possible with a system that only allows an agent to express the *order* of the candidates (like in the case of STV voting).

In this paper, we introduce a novel voting procedure which combined features from both STV and cumulative voting. Specifically, we take advantage of the features of cumulative voting to express the preferences using points, and at the same time allow for multiple rounds to avoid wastage by allowing the transfer of points in a similar way to the transfer of votes in STV. Having multiple rounds also allows a student to learn from previous rounds and to adjust its voting behavior accordingly. The details of our voting procedure are presented in Section 3. In addition to the voting procedure, we also introduce and evaluate a number of reasonable voting strategies that can be used by the student agents in the system.

### 2.2. Agent Technology for e-Learning

In e-learning, multiagent systems appear to be a promising approach to deal with the challenges in educational environments. They can provide new patterns of learning and applications, such as personal assistants, user guides and alternative help systems, which are helpful for both students and teachers in their computer-aided learning-teaching process [6]. Using multiagent systems to design educational systems could lead to more versatile, faster and lower

cost systems [10]. A number of researchers have applied agent technology to e-learning. De Meo et al. [7] proposed the X-Learn system which is a XML based multiagent system for adaptive e-learning based on user preferences and requirements. However, they focus on the adaptation and how to exploit XML technology facilities for handling and exchanging information related to e-learning activities. Shi et al. [9] designed an integrated multiagent systems for computer science education that focuses on two introductory courses where the learning process is student-centered, self-paced and highly interactive. They use Java RMI, JavaSpace and JATLite to create a web-based system; in this case they use personal agents to manage student's data and their interactions with course material.

Although these papers apply agents in e-learning, none of these papers apply any fundamental agent theories, such as mechanism design or social choice theory, to guide their design choices. In contrast, our approach is to apply voting mechanisms to an e-learning scenario where the candidates represent all the courses available, and where students can vote in any way he or she prefers. Thus our work explores, for the first time, voting procedures in an e-learning setting.

### 3. The Multi-Agent System

This section describes the architecture of the system (Section 3.1), the protocol and voting procedure that the agents follow (Section 3.2), how we model the preferences of students (Section 3.3) and three different voting strategies (Section 3.4).

#### 3.1. Architecture

The entities and objects that exist in the system are shown in Figure 1. Each agent is autonomous, that is, it is in control of its own actions and responses. The system consists of two types of agents: student agents (SAs) and the university agent (UA). First the student expresses his or her preferences to the student agent (see Section 3.3), and also chooses an appropriate voting strategy (discussed in Section 3.4). Then the SAs and the UA use a voting procedure to interact with each other and to choose which courses to run. To this end, the UA manages the votes cast by the student agents and decides, based on the voting procedure and the votes received, which courses will be cancelled. Furthermore, after completing the entire process, it will provide the SAs with a final list of running courses.

#### 3.2. Voting Procedure

In general, a protocol is the set of rules that controls the interactions between agents and determines the

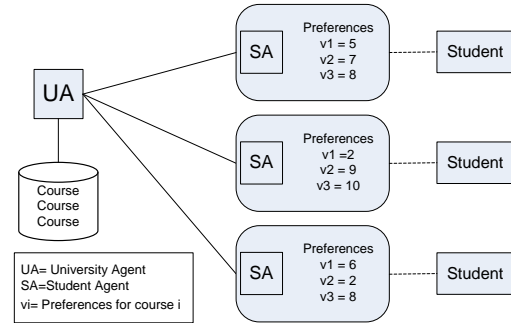


Figure 1: System architecture

beginning and end conditions of a given conversation [1]. In our system, the protocol works in several stages. In each stage, the student agents cast their votes for the courses by allocating points to each course. The course that receives the lowest number of cumulative points is cancelled, and the points that were allocated to the cancelled course are refunded. In the next round, the student agents can use these points (and any points that they did not use in the previous rounds), to vote again. Furthermore, in each round, the students are informed about which course is cancelled and the total number of points that have been allocated to the remaining courses so far. Note that, once allocated, a student cannot retrieve its points, unless the course is cancelled. The advantage of this iterative approach is that votes are not wasted since points allocated to the cancelled course can be reused for the remaining courses. Furthermore, the student can use the information about the current “popularity” (i.e. the current cumulative points) of the courses to guide its voting behavior (we discuss this in more detail in Section 3.4 where we discuss the voting strategies of students). In more detail, the protocol proceeds as follows:

1. Each student initially receives an equal and fixed number of points,  $IP$ , from the UA that they can use to cast their votes.
2. Each student allocates some or all of their available points to the available courses (they do not have to allocate all their points, but cannot allocate more than they have).
3. The UA calculates the cumulative points for each course.
4. The UA cancels the course with the lowest cumulative points.
5. The UA refunds the points for the cancelled course.
6. The UA informs all the SAs about the cancelled course, and the current cumulative points allocated to the remaining courses.

7. Now SAs can vote again using their remaining points (this includes the refunded points as well as any points which were not allocated in the previous rounds), and the process is repeated until the desired number of courses is remaining.

For example, if there are 40 courses available in total, but the university only has sufficient resources (e.g. staff and lecture rooms) to run 30 courses, then the voting will proceed for 10 iterations or rounds. At the end of each of these rounds, the course with the least number of cumulative points is cancelled.

### 3.3. Student Preferences

Each student has its own preferences for the different courses that the university offers. In our work, these preferences are modeled using a simple scoring model that describes a student's preference for each module as a number between 0 and 10, where 0 means that the student has no interest in the course. After the voting process has completed we can use these preferences to calculate a given *student's satisfaction* for the running courses. This is calculated by summing the preferences for courses that are running, as a fraction of the total preferences.

For example, if there are 7 courses available in total, and the preferences of a student are as follows:  $v_1 = 7, v_2 = 7, v_3 = 3, v_4 = 1, v_5 = 10, v_6 = 4, v_7 = 8, v_8 = 3$ , where  $v_i$  is the preference for the  $i^{\text{th}}$  course. The university decides to only run courses 2, 4 and 6. The student satisfaction is then calculated as follows. The sum of the preferences for the running course (21) is divided by the sum of the preferences for all courses (36). So the student satisfaction  $S$  is:

$$S = \frac{21}{36} \cdot 100 = 58.33$$

### 3.4. Voting Strategies

Abstractly, a strategy determines the agent's plan of action to achieve a particular goal. It specifies the way in which an agent behaves in a given environment [11]. In our scenario, the strategy determines the number of points to allocate to the courses in each voting round, given the preferences of the agent and the information received by the UA about the voting process. In this paper we introduce and compare three different strategies for the SAs, namely: proportional, equal share and intelligent. In what follows we describe each of these strategies in detail.

**Proportional:** The proportional strategy is included as an example of a simple but sensible strategy. Consequently, it provides a good benchmark that we can use to compare the performance of more sophisticated strategies. The main idea behind a proportional strategy is that, in each round of voting,

the student agent distributes its points proportionally to the student's preferences for each course. This strategy is simple in that it does not consider the information received by the UA about the current number of points allocated to the courses.

In more detail, the number of points allocated to course  $j$  is calculated as follows. Let  $RP$  denote the total number of points remaining (in the first round  $IP=RP$ ),  $m$  is the total number of available courses available, and the vector  $\vec{v} = \{v_1, v_2, \dots, v_m\}$  denotes the student preferences. Then, the total number of points to be allocated to course  $j$ ,  $b_j$  is:

$$b_j = \frac{RP}{\sum_{i=1}^m v_i} \cdot v_j$$

**Equal share:** The equal share strategy is included as an example of a very simple and ineffective strategy, and provides a good lower bound on the performance of the system. An equal share strategy is based on the principle that the SA gives all modules an equal number of votes, regardless of the student's preference. The following formula was used to calculate voting points each course:

$$b_j = \frac{RP}{m}$$

**Intelligent:** The intelligent strategy is included as an example of what can be achieved with a more sophisticated strategy that learns as the voting procedure progresses from one round to the next. Its effectiveness can be gauged by comparing it to the proportional strategy and the lower bound given by the equal share strategy. The main idea behind this strategy is that, in each round, the agent tries to predict the probability that a course will be cancelled based on the number of points currently awarded to each course from previous rounds. Then, based on this probability, it can calculate its *expected satisfaction* for a given allocation of points, and it will allocate the points such that the expected satisfaction is maximized.

In more detail, the probability of a course being cancelled is estimated using a *softmax* function, which is commonly used in discrete choice theory to make decisions in the case of incomplete information [5]. The probability that a course  $i$  is going to be cancelled in the future is given by:

$$ProbCancel_i(\vec{b}) = \frac{e^{\frac{-(cp_i+b_i)}{\sum_{j=1}^m (cp_j+b_j)} \beta}}{\sum_{k=1}^m e^{\frac{-(cp_k+b_k)}{\sum_{j=1}^m (cp_j+b_j)} \beta}}$$

Where  $cp_i$  is the *cumulative* number of points which have so far been allocated to course  $i$ , and  $b_i$  is the number of points that the student agent is planning to allocate to course  $i$  in the current voting round, and  $\vec{b}$  is the vector of points to be allocated. Furthermore,  $\beta$  is constant which enables a range of different strategies. For example, if  $\beta = 0$ , then each course is equally likely to be cancelled, irrespective of the cumulative number of points currently allocated. At the other extreme, as  $\beta \rightarrow \infty$ , the course with the lowest total number of points will be cancelled with probability 1, and all other courses will be cancelled with probability 0. All other cases fall somewhere in between. In our experiments we tune the parameter  $\beta$  such that it performs well in practice.

We now show how we can use this probability to calculate the *expected satisfaction*,  $ES$ , of the student, and how to find the allocation which maximizes this expected utility. The expected satisfaction is given by:

$$ES(\vec{b}) = \sum_{i=1}^m (1 - ProbCancel_i(\vec{b})) \cdot v_i$$

Note that the expected utility depends on  $\vec{b}$ , i.e. the number of points it is going to allocate to each course in the next round. The next step is then to find the allocation that maximises this expected utility. We estimate this using a search algorithm based on *random sampling*, which proceeds as follows:

1. We randomly generate an allocation vector  $\vec{b}$  subject to the constraint that the total number of points is equal to the maximum number of points that we would like to spend in the current round.
2. The student agent calculates the expected satisfaction.
3. If the current solution has a higher expected satisfaction than any previous solution, then keep the solution. Otherwise, discard it.
4. This process is repeated for 1000 times and the solution with the highest expected utility is kept.

Finally, to complete the strategy, we need to specify (1) how many total points to allocate in each round, and (2) how to allocate the points in the very first round (since at this point we have no information about the probability of a course being cancelled). To address the first question, note that we do not necessarily want to spend all points in the first round, since then we have no more points left to use in subsequent rounds to take advantage of the information that we receive in these rounds. We tried a number of different settings, and we found that allocating half of the points (including any points refunded from cancelled courses) in each round to perform well. In

the last voting round we allocate all remaining points. To address the second question, in the first round we use the proportional strategy (but only use 50% of the available points).

## 4. Evaluation

The objective of this section is to evaluate our novel voting procedure and to explore the impact of the three strategies described in the previous section on the overall student satisfaction. In particular, we consider the *social welfare* as the performance measure, which is calculated as the sum of the satisfactions of all students. We do this by simulating the multiagent system, as well as the student preferences. In this way we can see, for example, if the voting approach might advantage those students who use more intelligent agents. In addition, we compare the results to the *optimal social welfare*, i.e. the social welfare when the overall most preferred courses are selected.

There are many factors that influence the behavior of the agents. In order to evaluate the strategies, we now identify the variables and present a number of meaningful scenarios. The following variables were identified:

*Number of courses (m)*: This is the total number of courses that the university provides and in which the student can vote for.

*Number of running modules (r)*: This is the remaining total number of courses after the ones with the lowest student interest have been cancelled.

*Number of students (n)*: This is the total number of students in the system.

We vary the above values for different scenarios, which are explained in detail in Section 4.3. In addition to the above variables, we also have a number of constants:

*Initial points (IP)*: This is the number of points that each student initially receives. Without loss of generality, we set this value to 100 in the experiments.

$\beta$ : This is used when calculating the probability of a course being cancelled for the intelligent strategy (see Section 3.4). Throughout our experiments, we set this value to 5 since it was shown in initial tests to perform well.

### 4.1. Initialization of Student Preferences

The most straightforward approach to initialize the preferences is by randomly setting the satisfaction. However, in our initial experiments we found that, when the student population is large and when the preferences are initialized completely randomly, voting

has little effect because it does not really matter which courses are selected: for any subset of courses, there are many student who have a high satisfaction. Consequently, even the equal share strategy performed close to optimal. Furthermore, in practice preferences are not independent but there are groups of students with similar interests. To address these issues, we introduced a *bias* in the preferences.

In detail, for each student and each course we start by randomly generating preferences from a uniform distribution between 0 and 10. Then, for a *subset* of students we multiply the preferences of the subset of courses by a factor of 2. Then we limit all the preferences to be no more than 10. The result is that, because of the limit, this subset of students all favor a particular subset of courses. At the same time, their preferences are not the same.

### 4.2. Scenarios

In our experiment we considered three different scenarios. These differ in terms of the number of students, the number of total courses, and the number of running courses. We choose these scenarios to reflect the kind of courses typical in UK computer science department. We consider a large (undergraduate), medium (smaller undergraduate) and small (postgraduate) course. Table 1 shows the settings for these scenarios.

Table 1. Different setting to the scenarios

Scenario	#courses (m)	#running courses (r)				#students (n)
		10	20	30	40	
1	51	10	20	30	40	100
2	33	9	18	27		60
3	15	4	8	12		20

### 4.3. Analysis

We now proceed with discussing the results. In doing so, we consider two different cases. First, we consider the case where all students use identical voting strategies, and we compare the results of different voting strategies. In the second case, different proportions of students in the population use different strategies and we compare the resulting student satisfaction. In each of the experiments that follow, each scenario was run 30 times with different student preferences. Thus, the results shown are the average results over these runs.

#### 4.3.1 Identical Voting Strategies

First, we measure the performance of each strategy separately and compare this with the optimal solution (this is calculated by assuming that the university knows all the student preferences and selects the courses which maximize the social welfare).

Figures 2, 3 and 4 show the results for scenarios 1, 2 and 3 respectively. Here, the y axis shows the *percentage of student satisfaction*. This is calculated by the total satisfaction of the running courses, as a percentage of the total satisfaction if all the courses would be running. Furthermore, on the x axis we vary the total number of running courses (while keeping the other parameters in the scenarios fixed). The graphs show the differences in the satisfaction of the agents using different strategies and also compares this with the satisfaction of the optimal solution.

These results show that the outcome of the proportional strategy is almost identical to the optimal strategy (although this is not visible in the figure, there is some difference but this is not statistically significant), and the intelligent strategy does slightly less well but is still very close to optimal. On the other hand, we see that the equal share strategy does significantly worse. This suggests that a decentralized solution using voting results in high quality solutions that are comparable to optimal.

Interestingly, in this setting the intelligent strategy performs slightly worse than the simpler proportional strategy. This can be expected because proportional is similar to voting the “true” preferences, i.e. it does not try to outsmart the system. However, it may be possible for a group of students to outsmart the simple proportional strategy and increase their satisfaction at the expense of those students using the proportional strategy. To analyze this, in the next we consider a setting where students use different strategies.

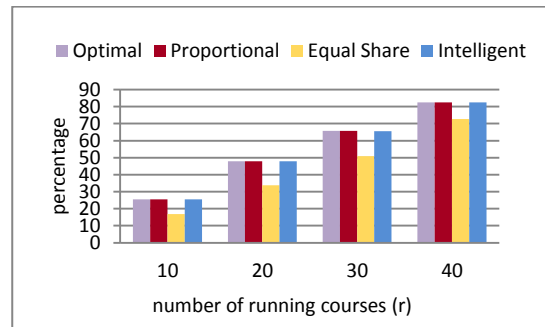


Figure 2. Scenario 1

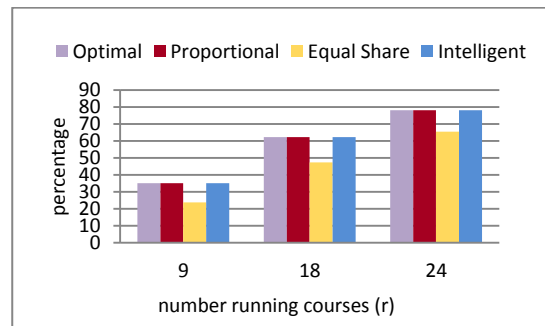


Figure 3. Scenario 2

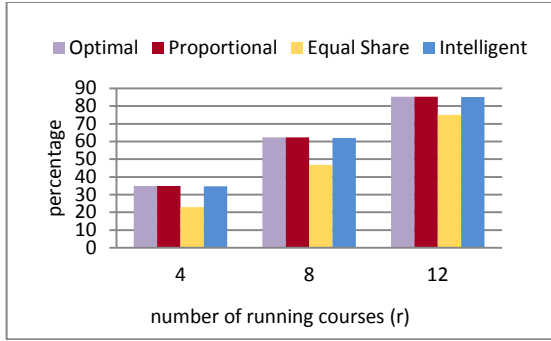


Figure 4. Scenario 3

#### 4.3.2 Combination of Strategies

In the next set of experiments we compare the case where a proportion of the students use one strategy, and the remainder of the students uses another strategy. Furthermore, we bias the preferences (as described in Section 4.1) in such a way that the students using the same strategy are also likely to have similar preferences. In the results that follow, the y axis shows the percentage of satisfaction for each group of agents using a particular strategy. Furthermore, on the x axis we vary the proportion of students using a particular strategy. For example, in Figure 5, 90-10 means that 90 students use the proportional strategy, and 10 students use the equal share strategy. In the experiments that follow, we set the number of running courses to 40 for scenario 1, 18 for scenario 2 and 4 for scenario 3. However, the results are very similar for the other settings in Table 1, and there are no qualitative differences.

The results in figures 5 and 6 show that the intelligent and proportional strategies are both significantly better than the equal share, irrespective of the proportion of students that use this strategy. On average, the improvement is around 8% compared to the equal share strategy. The results for other scenarios are very similar and not shown to avoid repetition.

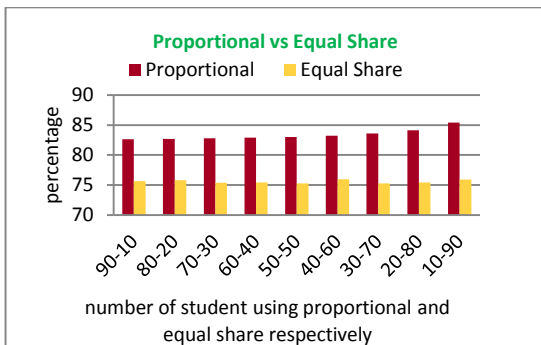


Figure 5. Scenario 1: Proportional vs. Equal Share

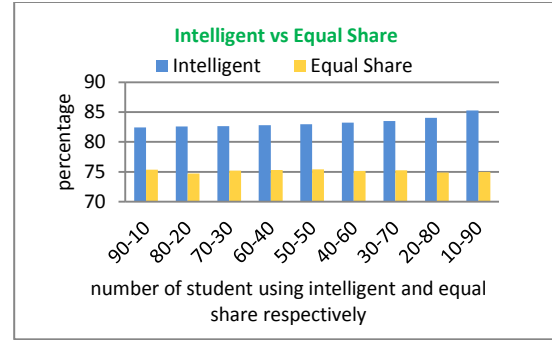


Figure 6. Scenario 1: Intelligent vs. Equal Share

Figures 7, 8 and 9 show the results with both the intelligent strategy and the proportional strategy for the 3 different scenarios. The result show that, as the number of students allocated to a particular strategy increases, the student satisfaction for these students also increases. However, this is mainly because of the bias that has been introduced; since students with the same strategy have similar preferences, when more students have these preferences they have greater voting power since they act as a group.

Comparing the intelligent and proportional strategies, it can be seen that there is not much difference between them. Although in some cases, as in Figure 9, the intelligent strategy slightly outperforms the proportional strategy (given the same number of students are using that strategy), in the other two scenarios, the proportional strategy outperforms the intelligent strategy. This suggests that the system cannot be easily exploited by an intelligent strategy. We have also tried to vary the parameters of the intelligent strategy (such as the beta parameter), but the results do not change significantly.

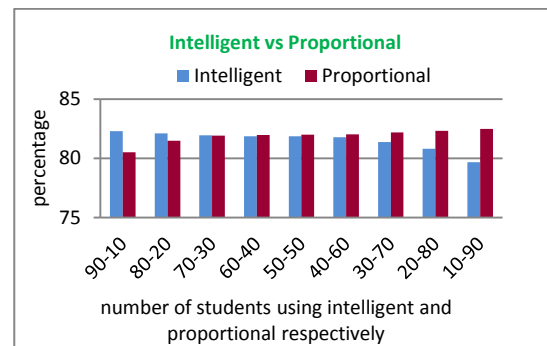


Figure 7. Scenario 1: Intelligent vs. Proportional



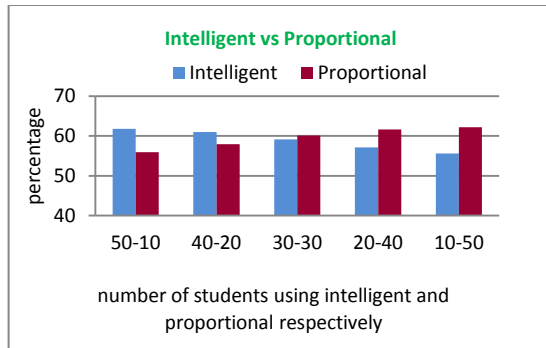


Figure 08. Scenario 2: Intelligent vs. Proportional

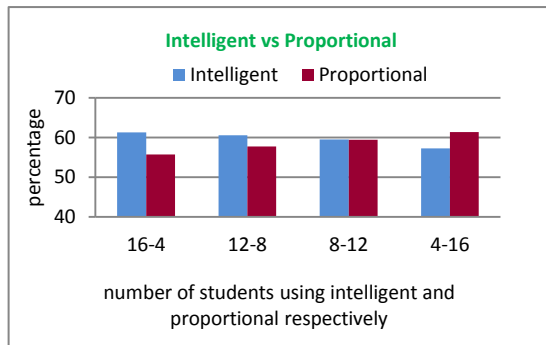


Figure 9. Scenario 3: Intelligent vs. Proportional

## 5. Conclusion

In this paper we presented a multiagent system based for this e-learning scenario based on voting theory. We first introduced a novel voting procedure where agents allocate points to different courses and voting occurs in several rounds. This way the agents are able to freely express their preferences and at the same time use the information provided from previous rounds to vote intelligently and strategically. We then introduced three different voting strategies, and evaluated their performance in a range of scenarios. The results show that even a simple voting strategy provides outcomes which are close to optimal. Furthermore, our intelligent strategy was unable to exploit other, more naïve voters. This is encouraging for the e-learning domain, where institutions are often required to be equitable.

Our future work consists of two parts. First of all, we intend to consider combinatorial preferences. Whereas the student agents in this paper have simple additive preferences, in practice the preferences are often *interdependent*. That is, the preferences for one course depend on whether or not another course is running. We believe that our voting procedure is particularly powerful for such combinatorial settings, especially since calculating the optimal solution becomes computationally intractable. Furthermore, we intend to consider other intelligent voting strategies and alternative voting procedures, and explore constraints such as limitations on the number of

courses that a student can take, and having prerequisite courses.

## 6. References

- [1] M. Beer, M. Inverno, M. Luck, N. R. Jennings, C. Preist and M. Schroeder, *Negotiation in Multi-Agent Systems*, Knowledge Engineering Review, 14 (1998), pp. 285-289.
- [2] F. L. Belfemine, G. Caire and D. Greenwood, *Developing multi-agent systems with JADE*, John Wiley, Chichester, 2007.
- [3] S. J. Brams, *Alternative Voting Systems*, Political Parties and Elections in the United States: An Encyclopedia, 1 (1991), pp. 23-31.
- [4] V. Conitzer, M. Rognlie and L. Xia, *Preference functions that score rankings and maximum likelihood estimation*, Proceedings of the 21st international joint conference on Artificial intelligence (2009), pp. 109-115.
- [5] D. A. Hensher, J. M. Rose and W. H. Greene, *Applied choice analysis : a primer*, Cambridge University Press, Cambridge, 2005.
- [6] P. Kommers and L. Aroyo, *Special Issue Preface - Intelligent Agents for Educational Computer-Aided Systems*, Journal of Interactive Learning Research, 10 (1999), pp. 235-242.
- [7] P. D. Meo, A. Garro, G. Terracina and D. Ursino, *Personalizing learning programs with X-Learn, an XML-based, "user-device" adaptive multi-agent system*, Information Sciences: an International Journal, 177 (2007), pp. 1729-1770.
- [8] A. D. Procaccia, *Computational Voting Theory: Of the Agents, By the Agents, For the Agents*, PhD thesis, The Hebrew University of Jerusalem, 2008.
- [9] H. Shi, Y. Shang and S.-S. Chen, *A multi-agent system for computer science education*, In Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE 2000), 32 (2000), pp. 1-4.
- [10] R. Silveira and R. Vicari, *Developing Distributed Intelligent Learning Environment with JADE — Java Agents for Distance Education Framework*, In the Proceedings of the 6th International Conference in Intelligent Tutoring System, ITS 2002 (2002), pp. 105-118.
- [11] M. J. Wooldridge, *An introduction to multiagent systems*, Wiley ; Chichester, Hoboken, N.J., 2009.