A CS UNPLUGGED ACTIVITY FOR THE ONLINE CLASSROOM *

Vincent A. Cicirello
Richard Stockton College
101 Vera King Farris Drive
Galloway, NJ 08205
cicirelv@stockton.edu

ABSTRACT

In this paper, we present a group, asynchronous problem solving exercise designed to introduce non-computer science students to a variety of concepts from artificial intelligence and operations research. The exercise is designed to be a Computer Science (CS) Unplugged activity suitable for the online classroom. CS Unplugged is a methodology for teaching CS concepts using hands-on interactive activities without using a computer. CS Unplugged activities are often used to introduce CS concepts to novices, such as elementary, middle, or high school students; but can also be effective at engaging the non-CS undergraduate student. By definition, CS Unplugged seems at odds with online education, as the participants in the exercise are necessarily using computers. We present an interactive activity motivated by CS Unplugged, but implemented within an asynchronous threaded discussion board. The activity is based on the well-known Bin Packing problem, and is specifically an extension of an existing active learning exercise known as Collective Bin Packing but adapted to the online classroom. The activity is designed for an undergraduate interdisciplinary course as part of a liberal arts curriculum.

1. INTRODUCTION

Computer Science (CS) Unplugged is a methodology for teaching CS concepts using hands-on interactive activities without using a computer [6] [16]. CS Unplugged activities are often used to introduce CS concepts to novices, such as elementary [6],

---

middle [18], or high school students [7]; but can also be effective at engaging the non-CS undergraduate student. CS Unplugged is a form of active learning, and several have demonstrated the general effectiveness of active learning within the computing sciences, from introductory courses (e.g., [9][10]) to more advanced topics (e.g., [2][3][13][17]).

In prior work, we developed an active learning exercise, "Collective Bin Packing" [4], and a Java program for its facilitation, suitable for illustrating concepts in a range of courses, including artificial intelligence (e.g., heuristics), discrete math (e.g., partitioning sets), and operations research (e.g., combinatorial optimization). Its basis is bin packing [8], a combinatorial optimization problem, but adapted for a team of individuals to work collectively toward a solution taking turns manipulating the current state of the problem without directly communicating the intent of their individual actions. Bin packing offers a puzzle-like exercise, which others have observed can attract student interest and encourage students to think about algorithms independent of programming [12].

At its core, Collective Bin Packing is an "Unplugged" activity. Although in the classroom, the instructor uses a Java application to facilitate the activity [4], it is used solely to maintain and visualize the current state of the problem. The problem solving logic resides in the minds (and possibly scrap paper) of the classroom participants. We have adapted the Collective Bin Packing activity to the online classroom. Our aim in this endeavor is to reach the non-CS major in a course designed to introduce non-majors to interdisciplinary topics of CS [5]. Recently we have been offering this course online. Our objectives in integrating Collective Bin Packing into the course are: (a) to ensure that we retain a high degree of interactivity among the students in the course, and (b) to engage students in active "implementation" of CS problem solving strategies.

We proceed with an overview of the bin packing problem and its adaptation as a CS Unplugged activity (Section 2). We then describe its implementation within an online classroom (Section 3) and our experiences with it (Section 4). We conclude in Section 5.

2. BACKGROUND

2.1. The Bin Packing Problem

Bin Packing is an NP-Hard combinatorial optimization problem [8]. An instance of the bin packing problem consists of a finite set U of items, a function S(u) that maps each item from the set U onto a real-valued size, and a bin capacity C. The problem is to partition the items from the set U into bins $B_1, B_2, ..., B_n$, such that the sum of the size S(u) for all items u in a given bin does not exceed the bin capacity C. The optimization problem is to minimize the number of bins necessary to achieve this partitioning.

The bin packing problem has many applications, including placing computer files in fixed length memory blocks, packing advertisements into television breaks, preparing the collective works of a musical composer on a set of CDs, logistics problems such as packing trucks, among others [14][15]. Bin packing is also important for scheduling problems, such as job scheduling [11] and wireless network channel scheduling [19].

2.2. Collective Bin Packing

In Collective Bin Packing, the class is presented with a bin packing instance, and students take turns deciding which items to place into which bins. A turn consists of one of the actions: (1) placing an item in a bin with sufficient space; (2) removing an item from a bin; or (3) moving an item from a bin to a different bin with sufficient space.

The exercise is designed to actively involve all students in the problem solving process. To avoid students directly influencing the choices of others, the students must only indicate their chosen action, and should not explain the reasoning behind their choice. The exercise continues at least as long as necessary to have all items in bins. It may or may not be feasible to continue until the students converge upon the optimal solution depending on the difficulty of the specific instance. Generally, once all items are in bins, the exercise continues until all students have had at least one more turn.

3. ADAPTING COLLECTIVE BIN PACKING TO THE ONLINE CLASSROOM

When we set out to integrate the Collective Bin Packing activity into an online course, we wanted to ensure that we did not lose its key elements: (a) active involvement of all students, and (b) the student as "computer" simulating the algorithmic process.

We chose to "implement" the CS unplugged activity within an asynchronous threaded discussion. From our experience in online instruction, we have observed that the quality of student interaction within threaded discussions degrades with more than 7 or 8 students. Therefore, we organize the students into random groups of no more than 8 and give each group access to a dedicated discussion board for the activity. Students are not able to access those of other groups. Part of the students' course grade comes from a series of group discussions. This activity counts as one of those discussions.

The online course where we have integrated the activity is primarily for non-majors and attracts a wide variety of students. To keep a fair playing field, we initially mask the identity of the bin packing problem from those who may have encountered it before. Instead, we describe the problem as follows: *"You have a set of shelves in your basement. Each shelf can hold no more than 100 pounds. You have a set of 20 objects on the floor and you know how much each weighs. Your group's goal is to put all of the objects on as few shelves as possible without exceeding the weight limit on any of the shelves."* They are given a list of items initially on the "floor." This is simply a list of item names with the weights indicated in parentheses, e.g., "Floor: A(19), B(10), ..."

We give the groups the following set of rules:

1. In any single discussion post, you can choose at least one and no more than two actions where the allowed actions are: (a) Put one of the remaining objects on any shelf of your choice provided you do not exceed the weight limit. (b) Remove an object from a shelf placing it back on the floor. (c) Move an item from one shelf to another.
2. You cannot post twice in a row. Once you post, you must wait until at least one other student in your group takes a turn.
3. You must not explain the rationale for your actions within your post.

To motivate the students to begin and to remain engaged in the exercise, they are offered extra credit points toward their discussion grade. The team that uses the fewest "shelves" gets 3 extra points. The team with the 2nd fewest shelves gets 2 extra points. All other teams that find a valid assignment of items to bins without violating constraints receive 1 extra point. In the event of a tie, the team that takes the fewest actions to solve the problem is chosen. Each group is provided with the same bin packing instance. The instance is designed to have an optimal solution that maximally fills 5 bins, and such that it is not too easily solved by any of the more common heuristics.

I make the first post in each group as an example of what to do, and to provide a coherent structure for the posts, consisting of a list of what is left on the floor and lists of the items on each shelf with an indication of space usage. Students take their turns by: (a) replying to a post, (b) cutting and pasting objects between the lists that correspond to their chosen actions, and (c) updating the space usage. There is little need to worry regarding maintenance of accuracy and consistency. Most groups tend to have at least one or two particularly astute members who will notice and actively correct any mistakes in the current state of the problem during the exercise. Here is an example of the first post I make to start the exercise (an instance actually used in the course):

Floor: A(36), B(33), C(39), D(43), E(7), F(19), G(37), H(8), I(29), K(37), L(23), M(29), N(10), O(22), P(11), Q(33), R(9), S(17), T(30)
Shelf 1 (28 pounds): J(28)
Shelf 2 (0 pounds):
Shelf 3 (0 pounds):
Shelf 4 (0 pounds):
Shelf 5 (0 pounds):
Shelf 6 (0 pounds):

After the exercise is complete, we have the groups engage in a discussion about the problem solving process. In this particular course, we introduce non-majors to concepts from artificial intelligence. This exercise is used in this course to illustrate a few concepts, including: (a) swarm intelligence [1], and (b) heuristics. The follow-up discussion question is designed to connect the activity to these concepts and is as follows:

"In this exercise, you had minimal control over the final solution to the problem. Each of you were only able to make small modifications without explaining to each other your reasoning process. This is much like how colonies of social insects solve problems collectively while interacting only indirectly through changes in their shared environment. Your individual actions resulting from your independent thought processes collectively lead to the solution. Discuss the reasoning processes that you each took in deciding where to put objects. For example, how did you decide which object to put on which shelf? Did you try to anticipate what others in your group would do next? Did the actions of others in your group provide any clues as to what you should do next?"

4. EXPERIENCES

We have been using Collective Bin Packing in an online interdisciplinary course on artificial intelligence (AI) for several semesters. We present our experiences in two semesters where we kept the details of how the activity was presented to students and how it was conducted constant. In each semester, there were 34 students enrolled (total of 68 students). Each class was divided into 5 groups of 6 or 7 students. The course fulfills a general liberal arts requirement in science and math. Additionally, it includes significant coverage of the history of AI and fulfills a college history requirement. Thus, it attracts a wide variety of students (e.g., non-science students looking for a "gentler" science course, CS or science students looking for a "gentler" history course, Philosophy, Psychology, Biology, CS and IS students interested in the topic itself). Table 1 provides a summary of the majors in these 2 semesters grouped into categories.

By the end of the activity, 8 of the 10 groups successfully found the optimal 5-bin solution. The other 2 groups found 6-bin solutions. The best possible path to a solution involves 20 individual actions (e.g., if you knew the optimal solution, you would simply move each of the items directly to its optimal bin). The closest to that direct solution was a group that took 22 actions. This was not typical, and the other groups took 26, 32, 34, 46, 50, 58, and 70 actions. It was evident in the follow-up discussion that members of the other 2 groups believed they could do better than 6 bins, as was also evident in their continued collective activity leading to 70 and 108 actions for each group.

Table 1: Distribution of student majors

Major	N	Percentage
Arts & Humanities	3	4.4%
Business	17	25.0%
CS / IS	16	23.5%
Health Sciences	6	8.8%
Natural Sciences & Mathematics	12	17.6%
Social Sciences	12	17.6%
Unknown	2	2.9%

In the follow-up discussion, student commentary can be categorized as follows:

(1) Ordering Heuristics (items): The most common remarks by students pertains to how they select which object to place next. One student comment was: "... *my strategy was to use the largest remaining item at the start of the next shelf and go from there.*" Another student in a different group likewise said: "... *put the heaviest items on the shelf first and then supplement with the smaller items.*"

(2) Ordering Heuristics (bins): Other students commented on how they chose which bin to place items, which mostly focused on maximizing its contents, such as: "I thought that it would be best to fill each shelf to capacity (100 lbs)."

(3) Swarm Intelligence [1] Related: Other students brought up concepts related to the swarm intelligence concepts, such as indirect communication via the shared environment. E.g., one student made the following remark: "*It seems to me that the problem could be solved.... kind of like the pyramids, we knew what our end result would be (or look like) and we just put one block at a time until we reached our goal. Obviously we could make adjustments as we went along, but we were all working toward a common goal.*" This was a particularly interesting remark in that one of the next readings was an article

discussing (among other things) wasp nest building where the completed portion of the nest influences where the next wasp places material.

5. CONCLUSION

In this paper, we considered the possibility of integrating CS Unplugged activities into online courses. Although at first, CS Unplugged seems at odds with an online course environment since students are necessarily using computers, it is possible to retain the underlying principles and spirit of CS Unplugged. Students can still engage in interactive activities that demonstrate and explain the underlying CS algorithms and concepts independent of the actual programmatic implementation of those concepts.

REFERENCES

[1] Bonabeau, E., Dorigo, M., Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems*, New York, NY: Oxford University Press,1999.

[2] Budd, T.A., An active learning approach to teaching the data structures course, *ACM SIGCSE Bulletin*, 38, (1), 143-147, 2006.

[3] Cassel, L., Very active learning of network routing, *SIGCSE Bull.*, 34, (3), 195, 2002.

[4] Cicirello, V.A., Collective bin packing: An active learning exercise, *Journal of Computing Sciences in Colleges*, 24, (6), 117-123, 2009.

[5] Cicirello, V.A., An interdisciplinary course on artificial intelligence designed for a liberal arts curriculum, *Journal of Computing Sciences in Colleges*, 23(3), 120-127, 2008.

[6] CS Unplugged, 2012, http://csunplugged.org/, retrieved November 26, 2012.

[7] Feaster, Y., Segars, L., Wahba, S.K., Hallstrom, J.O., Teaching CS unplugged in the high school (with limited success), *Proceedings of ITiCSE '11*, ACM, 248-252, 2011.

[8] Garey, M. R. & Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York, NY: W. H. Freeman and Co, 1979.

[9] Gonzalez, G, A systematic approach to active and cooperative learning in CS1 and its effects on CS2, *ACM SIGCSE Bulletin*, 38, (1), 133-137, 2006.

[10] Lau, K., Active learning sheets for a beginner's course on reasoning about imperative programs, *ACM SIGCSE Bulletin*, 39, (1), 198-202, 2007.

[11] Leinberger, W, Karypis, G, & Kumar, V., Multi-capacity bin packing algorithms with applications to job scheduling under multiple constraints, *Proc of the 1999 Int Conf on Parallel Processing*, (pp. 404-412), Washington, DC: IEEE Computer Society, 1999.

[12] Levitin, A. Analyze that: puzzles and analysis of algorithms, *ACM SIGCSE Bulletin*, 37, (1), 171-175, 2005.

[13] Ludi, S., Natarajan, S., Reichlmayr T., An introductory software engineering course that facilitates active learning, *ACM SIGCSE Bulletin*, 37, (1), 302-306, 2005.

[14] Malkevitch, J., Bin packing and machine scheduling, *Feature Column from the AMS: Monthly Essays on Mathematical Topics*, June 2004, archived at: http://www.ams.org/featurecolumn/archive/packings1.html.

[15] Malkevitch, J., Bin packing. *Feature Column from the AMS: Monthly Essays on Mathematical Topics*, May 2004: http://www.ams.org/featurecolumn/archive/bins1.html

[16] Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., Kuno Y, A CS unplugged design pattern, *SIGCSE Bulletin*, 41, (1), 231-235, 2009.

[17] Pahl, C., Barrett, R., Kenny, C., Supporting active database learning and training through interactive multimedia, *ACM SIGCSE Bulletin*, 36, (3), 27-31, 2004.

[18] Taub, R., Armoni, M., Ben-Ari, M., CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. ACM Trans on Comp. Edu., 12, (2), 2012.

[19] Wang, C, Li, B, Sivalingam, K, & Sohraby, K., Scalable multiple channel scheduling with optimal utility in wireless local area networks, *Wireless Networks*, 12, (2), 189-198, 2006.