# COLLECTIVE BIN PACKING: AN ACTIVE LEARNING EXERCISE*

*Vincent A. Cicirello*
*Computer Science and Information Systems*
*School of Business*
*The Richard Stockton College*
*Pomona, NJ 08240*
*(609)-626-3526*
*cicirelv@stockton.edu*

**ABSTRACT**

Active learning has become an important part of effective computer science pedagogy. Active learning helps to keep students engaged in the learning process and is useful at all levels of computer science education from introductory courses on programming through more advanced upper level topics and graduate level coursework. In this paper, we present an active learning exercise that we developed for use in courses on artificial intelligence for demonstrating heuristic search concepts as well as concepts related to swarm intelligence. We have also found the exercise useful in an interdisciplinary course for non-majors to demonstrate some of the problem solving approaches of computer science. It is also applicable to courses on discrete mathematics. The exercise that we call Collective Bin Packing is an adaptation of the well-known combinatorial optimization problem known as bin packing. During the exercise, students take turns deciding upon individual problem solving steps. A follow-up class discussion draws out the algorithmic elements of the students' collective reasoning process. An interactive Java application is used to facilitate the group problem solving exercise.

---

* Copyright © 2009 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

## 1. INTRODUCTION

Active learning has been shown as an effective educational tool whose strength derives from the active engagement of students in the learning process. Active learning has been widely adopted within computer science from introductory level programming courses (e.g., [8][7]) to courses on intermediate and advanced topics (e.g., [11][2][15]) including at the graduate level (e.g., [3]). There are many approaches to active learning that have been suggested for use in the computer science classroom, including the use of interactive animations [17], classroom worksheets [8], small group exercises [7], "students acting-out" [14], interactive multimedia [15], among other approaches.

In this paper, we present an active learning exercise we have developed with applicability to courses on discrete mathematics, artificial intelligence (AI), as well as for non-major courses. We discuss our experience with using the exercise in both an upper-level course on artificial intelligence as well as our experience with its use in a non-majors course. Specifically, we have integrated its use into an interdisciplinary course for non-majors that fulfills a liberal arts elective requirement at our institution [4]. We have additional plans to adapt its use for a course on discrete mathematics this coming spring.

We call our active learning exercise "Collective Bin Packing". Our original objective was to provide an example for a course on AI of the sort of "collective problem solving" that natural multi-agent systems employ. Our first use (as well as subsequent uses) of the exercise have provided insight into a plethora of concepts that the exercise can be used to demonstrate. At the heart of the exercise is the bin packing problem [6]. The bin packing problem provides a puzzle-like exercise, which others have observed can attract a student's interest and encourage students to think about algorithms independent of programming and implementation [10]. The exercise involves students taking turns making individual actions working towards the solution to the problem. As a follow-up, the class engages in a discussion of the problem-solving process taken by the group.

In Section 2, we begin by summarizing the bin packing problem and then providing the details of our active learning exercise, "Collective Bin Packing". Next, in Section 3, we provide a brief overview of a Java application implemented to help facilitate the exercise. Then, in Section 4, we discuss its use in the classroom and our experience with it. We conclude in Section 5.

## 2. THE COLLECTIVE BIN PACKING EXERCISE

At the heart of our active learning exercise is the classic combinatorial optimization problem known as bin packing. We begin this section with an overview of the bin packing problem and its diverse applications. We then present the details of our exercise, "Collective Bin Packing".

### Bin Packing

Bin Packing is a combinatorial optimization problem that is NP-Hard in the strong sense [6]. 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 as described here is considered a one-dimensional bin packing problem. The size constraint can be generalized to have multiple dimensions (e.g., to generalize the problem to one of placing 2-dimensional or 3-dimensional objects within bins represented by a fixed 2-D or 3D shape). There are many important applications of bin packing problems, including of the 1-D type. Malkevitch outlines some of these including placing computer files in fixed length memory blocks, packing advertisements into breaks during television shows, preparing the collective works of a musical composer on a set of CDs, logistics problems such as packing trucks, among others [12][13]. He also provides a gentle overview of the connections between bin packing and scheduling problems [12]. Some of the scheduling related applications of bin packing include multiprocessor scheduling [5], job scheduling [9], channel scheduling in wireless networks [18], among many others.

## Collective Bin Packing

Our active learning exercise that we call Collective Bin Packing involves students taking turns making decisions on which items to place into which bins. When it is a student's turn, they can do one and only one of the following actions: (1) The student can take an item that is not currently in any bin and place it into the bin of their choice, provided they choose a bin with sufficient space for the item. (2) The student can remove an item of their choice from the bin of their choice. (3) The student can move an item of their choice from its current bin into a different bin, provided the bin they choose has sufficient space remaining.

The exercise is designed to actively involve all of the students in the class in the problem solving process. One of our objectives with the exercise is to engage all of the students in the problem solving process. During the first run of the exercise, we want to avoid students directly influencing the choices of the others. To accomplish this, the students are told that when it is their turn they should simply indicate their chosen action. When they make their individual action, we request that they do not explain the reasoning behind their choice.

The exercise continues for at least as long as necessary to have all items in bins. It may or may not be feasible to have the exercise continue until the students converge upon the optimal solution depending on the difficulty of the specific instance used. Generally, once all items are in bins, the exercise continues until all students in the class have had at least one more turn.

## Follow-up Discussion

During the exercise itself, the students are not allowed to explain why they chose a particular item or why they chose to place it in a particular bin. However, after the exercise, we have a class discussion of the problem solving process. Here is where the students are encouraged to explain their reasoning process behind why they chose certain items and why they placed them into certain bins. The instructor can guide the follow-up discussion, using the students' observations, to match the objectives of the course. Examples will be given in the next section specifically related to our experience with using the exercise in a course on AI.

## 3. INTERACTIVE BIN PACKING APPLICATION

In order to facilitate conducting the Collective Bin Packing exercise in the classroom, we have designed an interactive Java application for solving bin packing problems. The application graphically displays the problem instance and allows the user to specify items to move into bins. Its use is not limited to our Collective Bin Packing active learning exercise, and can be used by an individual student.

Figure 1 shows a screenshot of our interactive bin packing application from the middle of a problem solving session. If used by an individual student, that student could designate which items to move into which bins. For facilitating the class-wide active learning exercise, the instructor would remain in control of the application, receiving and carrying out the instructions given one-step at a time from students in the class. The items are labeled alphabetically. The number in parentheses after the item's label corresponds to the size of the item. The interactive Java application is available at http://loki.stockton.edu/~cicirelv/.

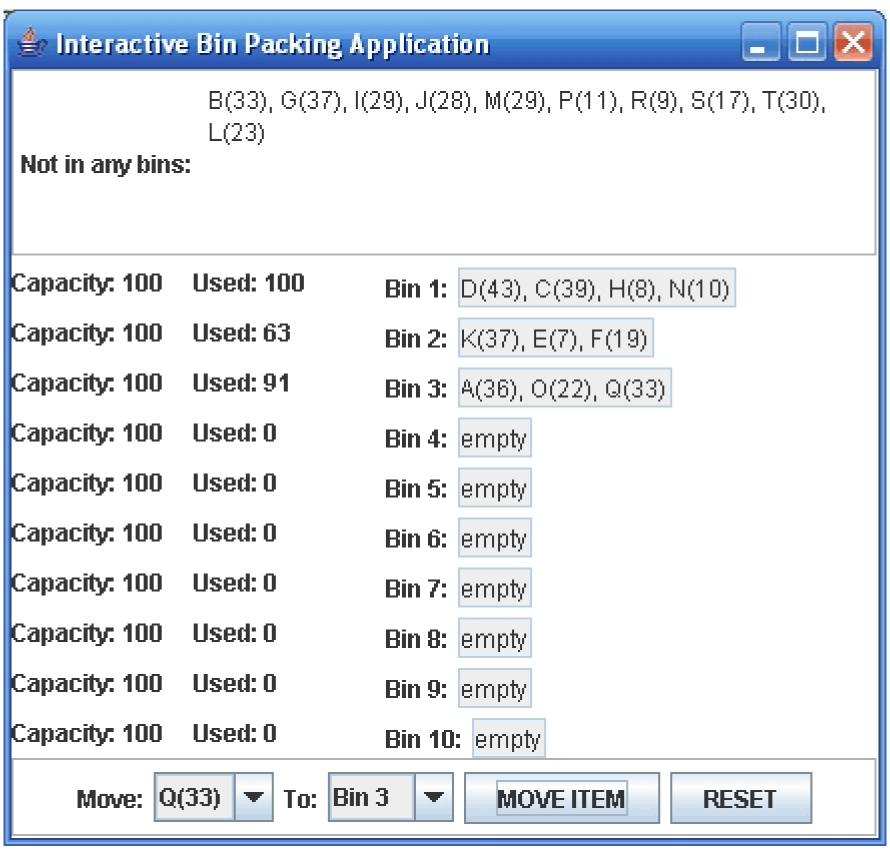


**Figure 1: A screenshot from our interactive bin packing Java application**

## 4. CLASSROOM USE AND EXPERIENCE

The Collective Bin Packing Exercise can potentially be used in a variety of courses to serve different objectives. For example, it can be used in a course on discrete mathematics as an example of a combinatorial optimization problem. The author intends to use it for this purpose within a course that we call Foundations of Computer Science, which is a second course on discrete mathematics that our computer science majors take.

The exercise can also be used within a course for non-majors as a way of introducing problem solving approaches of computer science. The author has used the exercise in two offerings of a course for non-majors designed to highlight the interdisciplinary nature of our field [4]. In the most recent offering, one business major in the class, commented during the follow-up discussion that the problem seemed like it was suitable for an approach he learned about in a course called quantitative business methods. The course the student referred to is closely related to operations research (OR). This provided motivation for discussing the relationship between OR and CS.

Of course, we can also use the exercise within a course on AI. This is specifically why we had originally designed the exercise. We wanted to offer the students an interactive example of the concept of swarm intelligence [1], the notion that the intelligence of a multi-agent system can emerge from the collective actions of many individual agents. It also provided a means of discussing elements of AI search algorithms [16], such as variable ordering heuristics and value ordering heuristics for constraint satisfaction search, as well as heuristic search in general.

For example, the follow-up discussion phase of the exercise in an AI course has been wide ranging. Some students happily admit to confusion and indicate that they chose items at random and placed them into an arbitrary bin. Others offer rationale for their actions that are either closely, or in some cases directly, related to some well-known heuristic algorithms for solving the bin packing problem. At this stage, the students would not be familiar with these heuristic solution approaches and are thus recreating them (or at least elements of them).

For example, each time that I have used this exercise in both majors and non-majors courses, there has always been at least one student who chose the "largest" item as their response. This is a feature of heuristic bin packing algorithms such as "first fit decreasing" and "best fit decreasing" that choose items for bin placement in decreasing order of item size. If the students are probed to further explain why they chose the largest item, the most common explanation given is that they believed the larger items would be the most difficult to fit into bins. These students are essentially describing the commonly used variable ordering heuristic for constraint satisfaction problem solving known as "most constrained variable first".

When the discussion is then steered towards the students' reasons behind their choice of bin for their chosen item, the most common explanation given by students at this point is that they either chose the bin arbitrarily or chose the first one that had space for the item. This response is held in common by both students who indicated the selection of the largest remaining item as well as those who indicate they chose an item arbitrarily, thus matching the behaviors of the "first fit decreasing" and "first fit" algorithms. These algorithms place the selected item in the first bin with sufficient remaining capacity for the item, with "first fit decreasing" ordering the items by size and "first fit" using the initial arbitrary ordering of the items. A few students in the class will occasionally indicate that their bin choice was based on how close to capacity the bins are. For example, among the minority of students who seem to use something other than a "first fit" based approach during the exercise; virtually all indicate that they focused on choosing a bin that could be brought closest to capacity (thus either a "best fit decreasing" or a "best fit" based approach).

In all of these cases, the exercise can be used to get students thinking about how to solve a problem like this and to engage the students in discovering common features of problem solving approaches. This can be a springboard to a deeper discussion of the relevant concepts, such as the use of variable ordering or value ordering heuristics to guide our search for a solution, heuristic problem solving approaches in general, etc.

## 5. SUMMARY AND CONCLUSIONS

In this paper, we presented an active learning exercise suitable for use in courses on artificial intelligence for demonstrating concepts such as variable/value ordering heuristics as well as swarm intelligence, in courses on discrete mathematics as an introduction to combinatorial optimization problems, and in courses for non-majors as a tool to demonstrate problem-solving approaches. The exercise, Collective Bin Packing, is an adaptation of the bin packing problem to a group problem solving exercise.

Our early experience with the exercise has shown that active learning can be used to help students derive for themselves features of algorithms for the problem, as well as general heuristic search principles.

## 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 Bulletin*, 34, (3), 2002.

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

[5] Coffman, E.G., Garey, M. R., & Johnson, D.S., An application of bin-packing to multiprocessor scheduling, *SIAM Journal on Computing*, 7, (1), 1-17, 1978.

[6] 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.

[7] 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.

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

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

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

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

[12] 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.

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

[14] McConnell, J.J., Active and cooperative learning: tips and tricks (part i), *ACM SIGCSE Bulletin*, 37, (2), 27-30, 2005.

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

[16] Russell, S., Norvig, P, *Artificial Intelligence: A Modern Approach*, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, 2003.

[17] Schweitzer, D., & Brown, W, Interactive visualization for the active learning classroom, *ACM SIGCSE Bulletin*, 39, (1), 208-212, 2007.

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