A Musical Packing Problem

by Kevin DeVizia

I find it more challenging to make math seem important to my unmotivated Algebra I students than to any other group that I teach. In order to convince such students that mathematics is useful, I've often found that the best examples are all around me in everyday life, just waiting to be noticed. I found myself in the car one day stuck in a traffic jam, when the cassette tape to which I was listening stopped playing. After a long wait, the tape still was not playing. Finally, switching the tape to the other side revealed that there was a great disparity between the total lengths of songs on the two sides of the tape. Could there be a better way to organize the songs so that the "dead time" on the tape could be minimized? Great! Another application for my kids to think about.

Listed below are the times of the songs on the two sides of this cassette (in minutes:seconds).

Side 1: 4:10, 2:39, 3:59, 3:47, 3:20, 3:11 (Total = 21:06).
Side 2: 3:21, 4:38, 2:32, 3:49, 1:47, 3:40 (Total = 19:47)

The data above gives a disparity of 1:19 and a completion time of 21:06. Essentially, this is a bin-packing problem, in which the songs are items to be packed in two variable-sized bins. (See the sidebar on **Packing and Scheduling**.) While my Algebra I students had never heard of bin-packing, they could certainly relate to this application.

I asked my students how to arrange the songs to achieve the minimum disparity, and a lively discussion ensued, with students offering different methods of attack. Occasionally, a student would be sure that the perfect solution was found--only to find an error in computation or someone with a better answer. Eventually the students wanted to know, "Well, what is the answer?" Of course, in my excitement over the problem, I never did decide on a solution, and this was all for the best--it was up to the students to check whether we had found the best solution. Could we be sure? As one student pointed out, we could list all possible ways to place the songs on the tape and then be sure. No problem, except that there are 2048 different ways to do this (this includes ridiculous arrangements, like all 12 songs on side one). By hand, the best solution my class found has a disparity of only 3 seconds. Of course, this does not take into account any aesthetic concerns for arrangement of songs. An interesting variant would be to classify songs as "slow" or "fast" and require that each side have an alternating sequence of these two types. This problem inspired my class to share a wide variety of creative strategies, and allowed students to interact in a meaningful way with mathematics and with each other.

Note: an earlier version of this article appeared in the Newsletter of the Pennsylvania CTM in Spring, 1993.

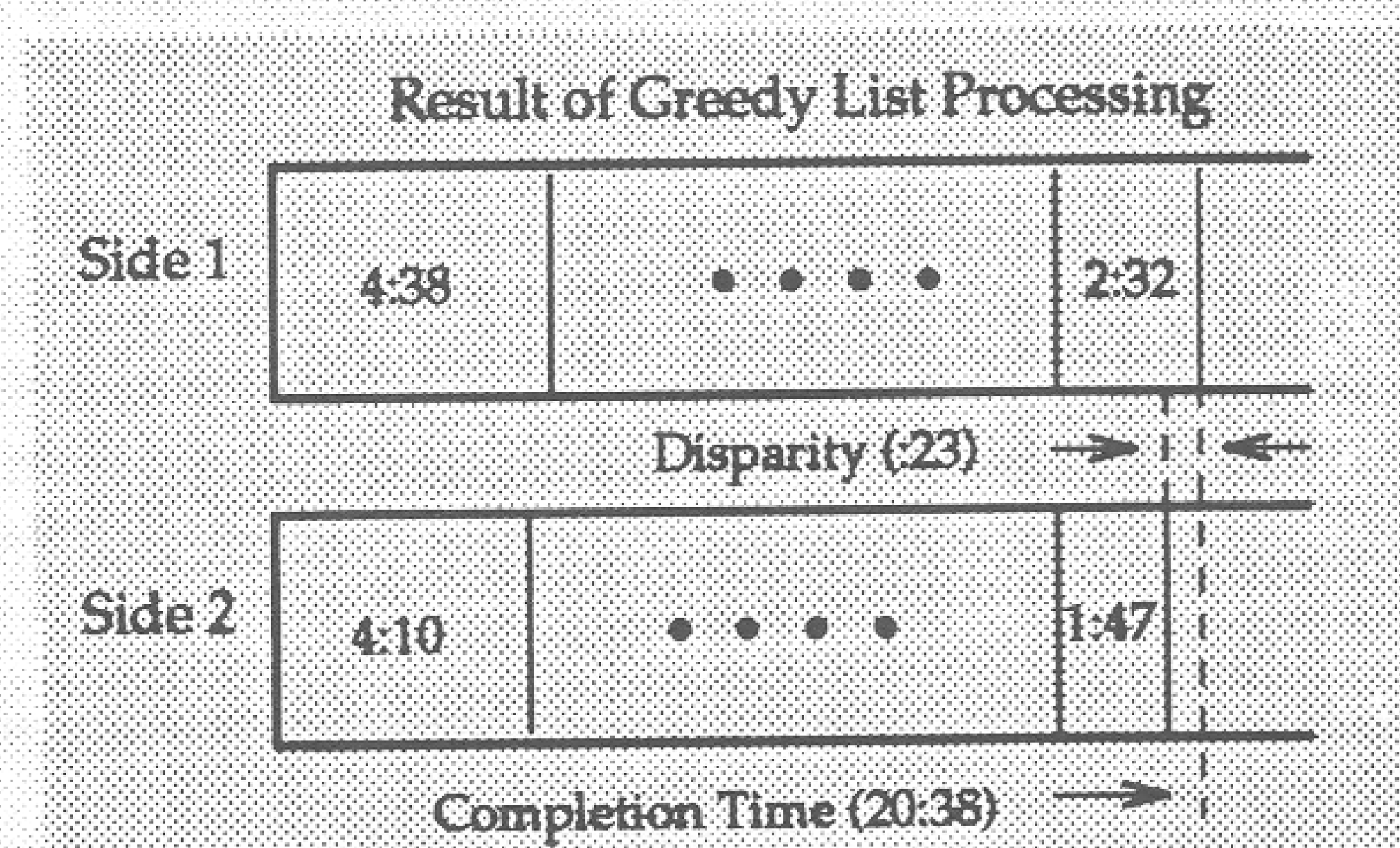
Packing and Scheduling

It may not be obvious that the musical packing problem has any relation to scheduling. However, the problem is essentially the same as "Two-processor Scheduling" [1]. In two-processor scheduling, you are given a list of tasks (in this case a list of songs to be played), along with the time needed to complete each task. You have two equivalent machines (one side of a cassette in this case) which can process the tasks; the problem is to assign the tasks to the machines to minimize the completion time. The two-processor scheduling problem is known to be NP-hard [1, 2] (which tells us that there is probably no fast algorithm to solve the problem exactly for large inputs). Thus, heuristic strategies, such as letting the class compete to get the best answer, may be quite practical!

It turns out that there is a simple, greedy "list processing" strategy which yields a completion time which is at worst 7/6 (1.17) of the best possible completion time (see [1, 2]):

(1) sort the tasks from longest to shortest;
(2) each time a machine becomes idle, assign the next task on the list.

If you use this strategy on the data above (converting to decimal notation first) you end up with two sides of approximately 20.63 and 20.24 minutes, giving a disparity of .39 min. (about 23 sec.) and a completion time of 20.63 min. The best possible completion time must be



at least half of the total time (40.87 min.), or 20.435 minutes; The ratio 20.63/20.435 ≅ 1.01, which is less than 7/6 as claimed.

References

1. Garey and Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman, 1979, p. 65, 238.
2. COMAP, *For All Practical Purposes,* 3rd Ed., W.H. Freeman, New York, 1994, pp. 83-88.