Algorithms Seminar
Airline Seat Swap

Notes by: Sean M Munson

August 31, 2012

Introduction / Motivation

M Bender brought in an idea from his summer journeys by plane. Travelling alone, he'd been asked by the flight crew to be swap seats so that a couple could sit together. He got to thinking if this process could be optimized ...

So in our conjecture, Stony Brook Air, like many airlines, prides itself in high-quality customer service. In order to improve the flying experience, they have begun a new service, allowing their flight staff to help arrange people once boarded on a flight. The flight crew will then help to arrange persons who are in couples or families or groups to sit together, allowing them to be 'happier' during the flight.

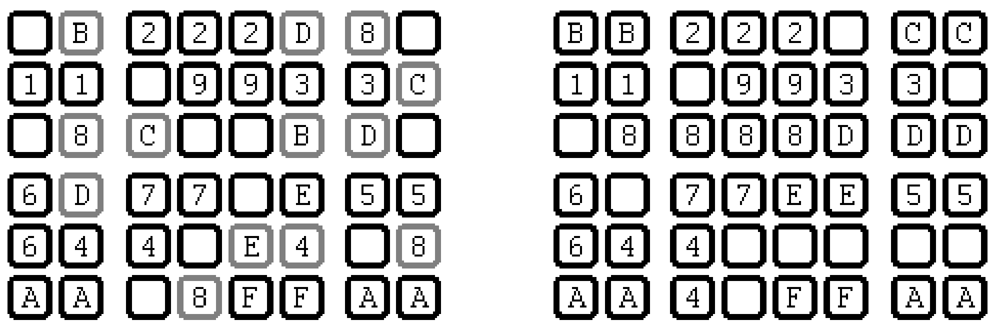

Figure 1: an initial arrangement, and a happier one (gray units are unhappy)

Q: By what process can the flight crew achieve the maximum level of happiness in the section while still disturbing the minimum number of seated people?

In the General case we also will want to preserve certain seat qualities (aisle, window, etc). This may make the problem significantly harder to solve.

Assumptions: We assume that the initial distribution of fliers is arbitrary. We also limit our problem by stating that the solution must be solved in-place once fliers are boarded. Singletons people are always 'happy' sitting next to anyone. Members of a group are Happier when surrounded by their group members.

Some proposals for initial analysis modes were put forward ; S Skiena suggested colouring groups and then seeking maximum color adjacency, or a graph structure with edges between members of the same group. J Mitchell proposed simplifying the problem first to a one-dimensional model of the problem, with only couples on board (maybe a plane full of honeymooners headed to a post-nuptial resort?).

We also restrict ourselves to a simple two-person swap. Now we have a clearly defined problem with strict limits on choices of actions. It also creates a sorting / Matching class of problem which we can attack with well-known ideas.

E Arkin asked for clarification whether we were seeking to minimize the number of swaps or the total swap distance. J Mitchell suggested that either could be sought as this was an optimal path type of problem. S Skiena commented that he felt we should think about disturbing the fewest number of persons.

J Zuber mentioned that this initially seemed like with complete unhappiness (a worst-case Scenario), this could be solved in $\frac{n}{2}$ operations.

M Biro commented that this bound is tight.

After some discussion, E Arkin proposed the following sweeping-line algorithm as an initial attempt : algorithmic

while $i \leq n-2$ **do** Read a pair at a time $(i, i+1)$
if $group(i) \neq group(i+1)$ **then**
get the jth element from the right with $group(j) = group(i)$
swap the jth and $i+1th$ element
end if
end while

For example :
1 3 1 2 3 2 Initial condition
1 1 3 2 3 2 swap position 2 with position 3
1 1 3 3 2 2 swap position 4 with position 5

J Zuber believes that this is optimal, and offers the following thumbnail proof:

Proof: Any swap that is more optimal will find two pairs to satisfy at once. Since the left-right sweep will find them at the same time, no other search can be more optimal. Additionally, if the pairs are odd-aligned, no single swap can satisfy both pairs at once.

J Yan points out the that last swap will satisfy both remaining unsatisfied pairs simultaneously, and therefore a slightly tighter bound is $\frac{n}{2} - 1$ is a better supper bound.

Discussion then started using the terms 0-swap to describe a swap that satisfies no pairs, a 1-swap to be a swap that satisfies one pair, and a 2-swap that satisfies two pairs at once. S Skiena then threw out the question of whether we could have a sitauation where two 0-swaps could then create the scenario where 3 2-swaps could happen, giving a more efficient algorithm.

L Walsh pointed out that each 0-swap could never make any more than 2 2-swaps possible.

Pairs with singletons we then moved on to the 1-dimensional model with the addition of possible single persons flying alone. These people are considered always happy whomever they sit next to.

E arkin challenged seminar members to find a counterexample demonstrating that the previous algorithm would be weaker, and Paul Fodor brought forth the following example:

1 2 3 3 4 2

S Munson pointed out that since 1 is unmatched, the algorithm would break as written. If it were amended to 'skip' singletons, then the algorithm would work out as :
1 2 3 3 4 2 Initial conditions
1 2 2 3 4 3 swap positions 3 and 6
1 2 2 3 3 4 swap positions 5 and 6

and clearly a more optimal solution is to simply swap positions 1 and 6.

S Skiena pointed out that the key here would be to detect 2-swaps consistently.

A Ban pointed out that the center element in a triplet of singletons would never be moved.

S Skiena Pointed out that there would never be a circumstance where both members of a pair were moved.

There was general discussion after this point but no more significant progress was made on the problem, and the meeting was closed with encouragement of the students to work more on this project, as it seemed to M Bender that this could result in a publishable work.