

Any port in a storm

M. Hassell

April 6, 2017

1 Introduction

Here I describe the solution and methods for the “Any Port in a Storm” problem from PuzzlOR: http://puzzlor.com/2017-02_PortInAStorm.html.

2 Model

This can be cast as an integer programming model with binary decision variables X_{ij} for $i = 1 \dots 20$ and $j = 1 \dots 3$, where

$$X_{ij} = \begin{cases} 1 & \text{if boat } i \text{ docks at port } j, \\ 0 & \text{otherwise.} \end{cases}$$

This could be narrowed down to only 40 decision variables, since the determination of a ship going (or not going) to two ports determines the third. I’ll skip this optimization for now (mostly because I didn’t think of it!) The boats are numbered from 1 to 20 based on which group they are in: boats 1-8 are in the top left, 9-13 are in the middle, and the remainder are at the lower right.

The objective function we seek to minimize is simply

$$\sum_{i,j} c_{ij} X_{ij}$$

where c_{ij} is the distance that boat i has to travel to reach port j . If we arrange the variables and costs in two arrays, we simply have to compute their Frobenius product to get the objective. I didn’t do this since it was simply easier to do some simple index arithmetic and use PuLP’s built in dot product. We also have the constraints on the number of boats that can fit into a given dock. These are given by the equations

$$\sum_i X_{i1} = 4, \quad \sum_i X_{i2} = 7, \quad \sum_i X_{i3} = 9.$$

Finally, each boat needs to go to exactly one port, so we have $\sum_j X_{ij} = 1$ for all i .

3 Solution

I made use of PuLP's tools to solve the problem. It returns the value of the objective function to be 31, so the total distance travelled by all the boats in the shortest routing scenario is 31 units. We can also check which decision variables are nonzero to see what the optimal assignment would be.

In dock 1

Boats 1, 4, 7, and 8.

In dock 2

Boats 5, 9, 10, 14, 17, 19 20.

In dock 3

Boats 2, 3, 6, 11, 12, 13, 15, 16, 18.

4 Conclusion

This was a fun way to learn some PuLP. It would be interesting to consider the speed of the boats and instead compute the minimum time required for everyone to reach a dock. This would only impact the objective function. In addition, some boats may be too large to fit in a given dock, and this would introduce other constraints that could be handled in a simple manner. Since IP is NP-hard, it would be challenging to verify this solution is the optimal one (unless $P = NP$). One could run some simulations whereby the boats are randomly assigned to a dock just to check that we don't find a smaller value. The dual problem could also be solved, but this only provides a lower bound on the optimal solution. This would be a reasonable sanity check for our model, but doesn't tell us if we can do better.