

MAT004 - Example Coursework

Geraint Palmer - c1016865

1 Introduction & Problem

Josie's shed is a chain garden stores based in France. They would like to start an radio advertising campaign that will reach the following 30 cities in France:

- Amiens
- Avignon
- Biarritz
- Bordeaux
- Brest
- Caen
- Calais
- Dijon
- Grenoble
- La Rochelle
- Le Havre
- Le Mans
- Lille
- Limoges
- Lyon
- Marseilles
- Montpellier
- Nancy
- Nantes
- Nice
- Orleans
- Paris
- Perpignon
- Poitiers
- Reims
- Rennes
- Rouen
- Stranbourg
- Toulon
- Toulouse

Each city has it's own radio station whose broadcast can reach different distances, and so reach different cities, according to an adjacency matrix. Figure 1a shows the reach of the broadcast in different cities, with an edge connecting cities if they can hear each others' broadcasts. Josie's Shed would like to know the minimum number of cities they need to broadcast from in order to reach all thirty French cities. This is an example of the *dominating set problem*.

2 Methodology

A simple heuristic neighbourhood search algorithm will be used. Here neighbourhood operator is selecting a random set of cities. It is evaluated by checking if that set is a dominating set, if so, then the cost of that set is the size of the set. Otherwise the cost of that set is 30, the number of cities, as it would be better to choose every city (and so reach every city) than to choose a non-dominating set. If the cost of that set is smaller than the previous best solution, then that is labelled as the best solution. This is repeated a number of times. This algorithm is shown in Figure 1.

Algorithm 1: A neighbourhood search algorithm.

Data: \mathbf{A} an adjacency matrix; n the number of iterations

Result: \mathbf{s}_* , a minimal dominating set.

$m \leftarrow$ size of \mathbf{A} ;

$i \leftarrow 0$;

$c \leftarrow$ rank(\mathbf{A});

$b \leftarrow m$;

Let \mathbf{s}_* be a vector of m ones.;

while $i \leq n$ **do**

 Let \mathbf{s} be a random vector containing c ones and $m - c$ zeros;

$\mathbf{r} \leftarrow \mathbf{A}\mathbf{s}$;

if $0 \notin \mathbf{r}$ **then**

$b \leftarrow m$;

$\mathbf{s}_* \leftarrow \mathbf{s}$;

$c \leftarrow c - 1$;

end

end

Here \mathbf{A} is the adjacency matrix, n is the number of iterations, m is the number of cities, c is the current number of cities to try to find a dominating set, i is the current iteration, b is the current best cost, and \mathbf{s}_* is the current best solution.



(a) Map of cities who can mutually hear each others' radio broadcasts.

(b) Map of solution with 7 cities in the dominating set (orange).

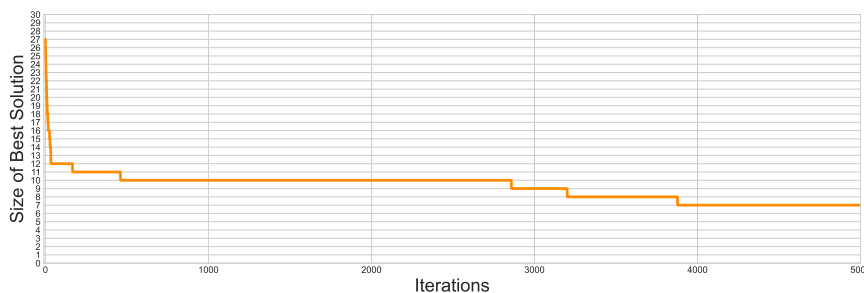


Figure 2: Plot of the best solution b over time.

Note that the vector $\mathbf{r} = \mathbf{A}\mathbf{s}$ is a vector giving, for each city, the number of cities in \mathbf{s} they see. Therefore if this contains a 0 then there is at least one city that is not seen by the cities of \mathbf{s} , and so if it doesn't contain a 0 then this is a dominating set.

Note also that we begin with $c = \text{rank}(\mathbf{A})$. The rank of a matrix is the number of linearly dependent rows, and so in an adjacency matrix, whose elements are either 0 or 1, it is the number of identical rows. If two rows are identical, then the two corresponding cities see the same cities, and should never be chosen together. Therefore we know we can find a solution of size $\text{rank}(\mathbf{A})$.

3 Results

When run for 5000 iterations we get a solution containing just 7 cities in the dominating set. This is shown in Figure 1b. The chosen cities are: Avignon, Bordeaux, Limoges, Nancy, Nice, Rennes, and Rouen.

The number of iterations, 5000, was chosen arbitrarily. The higher the number of iterations, the better the solution, as we are more likely to randomly find the dominating sets. Figure 2 shows the relationship between iterations and number of cities in the dominating set. We see that very quickly a good solution of 10 cities is found, but it takes a very large number of iterations before any better solutions are found.