PROGRESS OUTCOME 7

Solving unsolvable problems

Context

The students have been investigating possible answers to the question "How could you solve an unsolvable problem?" An algorithm is a set of instructions for solving a problem, so students were asked to consider whether people could solve all their problems using computer-based algorithms.

Kalani has decided to investigate how well algorithms can solve the complex problem of efficient international parcel delivery.

Insight 1: Complex problems

I did some research and discovered that because of their complexity, some problems are intractable. In an ideal world, computers would always be able to use an algorithm to find the best solution to any problem within moments. However, in reality many problems are so complex that there are billions of possible solutions and a computer cannot solve these problems in a useful amount of time.

From my research, I determined that the complexity of a problem is often described using the Big O notation. This describes the performance of the algorithm in the worst-case scenario by using an expression of n, where n can be the number of objects within a problem or the number of steps a computer must take to solve the problem.

Insight 2: Problem-solving mechanisms

Computers do not approach problems in the same way as people. Because a computer doesn't have the ability to be 'spatially aware', it cannot approximate a solution in the same way we can. To solve a problem, a computer has to 'brute-force' its way through the problem by calculating every possible candidate for a solution.

'Brute-force searching' a problem will deliver the best solution, but it usually takes impractical amounts of time to do so. For example, in an algorithm set up to efficiently manage an international delivery system, for every new city added to the list of delivery addresses, the number of solutions increases by another order of magnitude, as there are n! (1 ×2 ×3 × … n) solutions for a problem of size n (where n = the number of addresses in the system).

Insight 3: Application and impacts

I decided to see whether algorithms could improve efficiencies in the transport industry and have a positive impact by reducing emissions. For example, logistics and distribution companies regularly need to improve the efficiency of the algorithms they use for their deliveries. A multinational courier company delivering hundreds of thousands of mail items every day must manage its fleet of courier vans and aeroplanes so mail is delivered as quickly and cheaply as possible to the correct addresses. To do this, the routes the couriers take must be organised to minimise the total distance travelled.

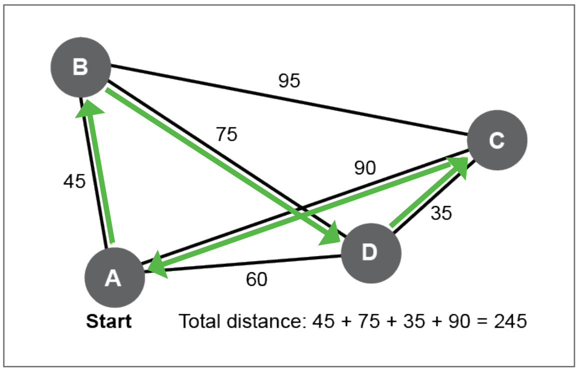
The less distance a courier travels, the lower the cost of fuel and wages and the more profit the company will make from its deliveries. An added benefit is that fewer fuel emissions will cause less harm to the environment. Any cost savings will compound over time into major increases in profit, putting the company ahead of its competitors. However, the courier company can't wait for a computer to take months or years to find the best route. At the same time, relying on human interpretation to find the best route is unlikely to be as beneficial as using a computer.

Insight 4: Solving the unsolvable

I researched this concept and learnt about heuristics, which in computer science are techniques designed to help computers solve problems efficiently and quickly. The "nearest neighbour" heuristic is commonly used to find a solution to complex problems. Nearest neighbour is a "greedy" algorithm, where the algorithm attempts to make the best possible choice at each stage, rather than looking for an overall best solution.

In the example below, a company has to send a courier from the depot (A) to three different houses (B, C and D) to deliver parcels and then return to the depot. Using the nearest neighbour heuristic, the total distance travelled is 245 units. This route appears to be efficient, but we don't know whether this is the optimal route because the other routes have not been calculated.

The nearest neighbour heuristic is one of the fastest algorithms for solving intractable problems like this, because at each step it simply calculates the distance from a point to all other points and chooses the point the minimum distance away.

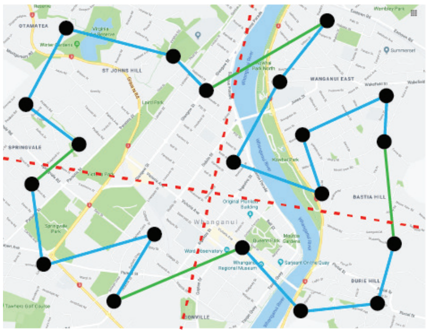


Insight 5: The wider implications

In real-world cases, typically more than one heuristic is used on a problem to ensure the best possible solution is found within a short amount of time.

For example, a company needs to deliver to the 20 addresses shown on this map. If it were to brute-force this problem, approximately 2.4×10^{18} obtained solutions would need to be calculated. This would take an impractical amount of time, and so the company needs to use a better way to find a solution.

In this image , the company has used the divide and conquer heuristic to divide the 20 addresses into smaller problems of more manageable size. The company can then brute-force these individual sections and put them back together. However, with a larger problem, brute-forcing could again run into time issues. In that case, the company could use other heuristics to find an efficient solution for each small sub-problem.



Once all the small solutions are joined together, the company has a final solution to its initial problem. This efficient and optimal problem-solving approach enables the company to deliver their parcels more quickly, decreasing their running costs and increasing profits.

This method of problem solving has implications for many kinds of organisations. It helps them reduce costs and solve ‘unsolvable’ problems, as well as benefitting the environment.

ISBN: 978-1-77669-239-2

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA