Lesson: Greedy Algorithms

Bookmark this page

By the end of this lesson, you will be able to **define greedy algorithms and apply them to find approximate solutions to the Traveling Salesman Problem**.

Video - Greedy Algorithms

Start of transcript. Skip to the end.

Hi, it's me again.

I can see you're hungry for some more graph theory.

Me too!

Sometimes, when a problem is too complex to solve, approximate algorithms are used.

Approximate algorithms provide a solution close to the exact solution, but in a much

shorter time.

Sometimes, when a problem is too complex to solve, approximate algorithms are used. Approximate algorithms provide a solution close to the exact solution, but in a much shorter time. One example of approximate algorithms are greedy algorithms.

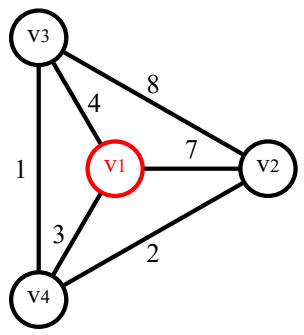
Principle

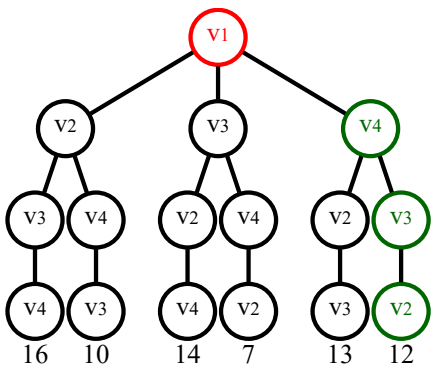
The purpose of a greedy algorithm is to approximate the solution to a problem by a succession of locally optimal solutions. To illustrate this, let's consider the traveling salesman problem, the TSP, which we've seen is a complex problem to solve.

Instead of looking for an exact solution, we can choose to always go to the nearest unexplored vertex from the vertex we are currently at.

This is a greedy algorithm because, at each point, the distance to travel is minimized, one step at a time. So we've made locally optimal decisions.

However, this solution is not necessary optimal. Take the following graph as an example, for which we explore all possible paths, as depicted with the corresponding tree:





Let's put the greedy algorithm into practice. Starting from vertex v_1, we'll first go to v_4, because the corresponding edge is the shortest of all possibilities. Using the same principle, we'll then move to v_3 and finally to v_2. In this example, the obtained path has a total length of 12. However, the optimal solution is a path that has a length of 7, which is the path $\{v_1, v_3\}, \{v_3, v_4\}, \{v_2, v_4\}$.

Don't forget, this algorithm is only one example of a greedy algorithm to answer the TSP. As well as systematically choosing the nearest neighbor, which was the example we've just seen, other heuristics can be used. Here are some examples:

- Choose the path of minimum length by exploring k remaining vertices ($k = 2, 3, \ldots$), $k = 1$ is the simple greedy algorithm we have just seen,
- Choose to go towards a dense area of a maze where there's many pieces of cheese to be found,
- Randomly launch several possible searches, and retain the best...

A case where the greedy algorithm is optimal

In some cases, greedy algorithms may be optimal. Consider the problem of returning coins after a payment. We wish to return a certain amount of change in euros, and we've got the following pieces: 1c, 2c, 5c, 10c, 20c, 50c, 1€, 2€. The aim is to use as few coins as possible.

A typical greedy algorithm is to select the coin with the highest possible value inferior to the sum that needs to be reimbursed, up to the point where the remaining change to be given is 0. So, if you want to return 3.27€, then the algorithm will first return 2€ (leaving 1.27€ to be returned), then 1€ (leaving 27c to be returned), 20c (leaving 7c), 5c (leaving 2c), and finally 2c.

By looking for all possible combinations, we can see that this is indeed the optimal solution. To show this, it's enough to notice that each coin has a value at least equal to twice the coin of lower value. This means that it's always going be more efficient to give priority to the higher value piece, because one piece will be returned instead of several.

Now let us consider the following pieces: 1c, 2c, 5c, 10c, 20c, 40c, 50c, 1€, and in this case we want to return 80c. The greedy algorithm will return 50c (30c left), then 20c (10c left) and 10c, i.e., three pieces. However, returning 80c can also be done by returning two 40-cent pieces. In this case, a greedy algorithm is not optimal.

To recap, this lesson has introduced you to greedy algorithms to find approximate solutions to complex problems such as TSP.

‹ Previous

Next ›

Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate