Modified Bellman-Ford algorithm for arbitrage searching

Péter Gyimesi

Matematika MSc
ELTE TTK

January 8, 2026

LaTeX

Problem

- Main goal: find an arbitrage in real world tests
- Given: the currencies and money exchangers (unknown functions)
- The program should return with a simple cycle and a starting amount.
- We can ask a point of a function
- Input: the ID of the function and a point
- Output: the amount we get back from the other currency and an additional cost (usually a constant)

Algorithm

- Each function is monotonic and concave
- If everything is linear an arbitrage is equivalent with finding a negative cycle after some transformation.
- We can do that with Bellman-Ford algorithm.
- We can try to calculate the best path from the starting node with a fixed starting amount, with a similar idea.

- If the graph is non conservative the algorithm might not terminate
- Because of this we do not check the edges from a descendant to its ancestor
- With this method we get a spanning tree, of the shortest paths (or from good candidates)
- We can all of the edges, back to the starting node, and choose the best one

- The best starting amount is not obvious for a cycle.
- The function of the final profit is usually concave, so we can find that with a ternary search
- This way the algorithm finds a cycle. It can be extended if there are several beneficial edge disjoint cycles.

Results

- For the running time the bottleneck is finding the tree of the shortest paths
- In practice each edge only needs to be checked a few times
- Works quickly, in larger graphs too
- For real life tests usually the algorithm could not find any arbitrage
- With smaller additional cost the results were better

Thank you for your attention!