Marcel Meschenmoser

Student	Marcel Meschenmoser
Examiner	Prof. Stefan F. Keller
Subject Area	Software and Systems
Project Partner	Rolf Sutter, Zünd Systemtechnik AG, SG

Cutting TSP

Combinatorial Optimization Problem to Sort Geometry Objects

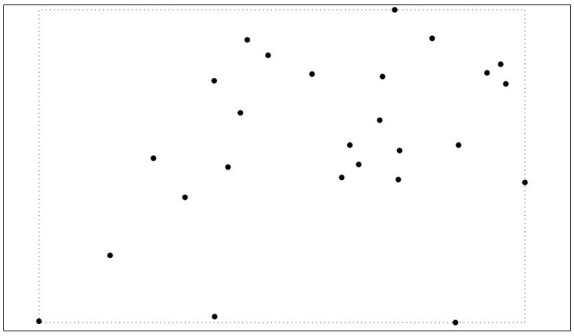

Figure 1: The 26 cantonal capitals of Switzerland.
Own presentment

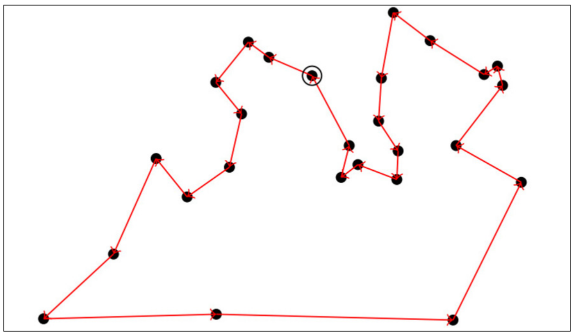

Figure 2: The fastest way to visit each city once. Found by the branch and bound algorithm.
Own presentment

algorithm	solution	runtime [ms]
Random	2860059.824	0
NearestNeighbor	1090345.418	0
BestOfNearestNeighbor	951531.322	0
BestGlobalEdge	986636.951	0
MaxRegret	1084832.851	0
BestOfMaxRegret	989304.364	4
BestInsertion	1050247.201	0
PilotNearestNeighbor	951531.322	1
BeamSearchNearestNeighbor5	937099.250	10
Genetic100	950611.438	223
Genetic1000	937099.250	1723
SimulatedAnnealing1000	1063239.761	0
AntColony100	1151057.384	360
BranchAndBound	937099.250	275886

Figure 3: Evaluation of the constructive algorithms. The fastest way is 937,099 kilometers.
Own presentment

Initial Situation: The company Zund Systemtechnik AG develops digital cutting machines. With these cutting machines, a lot of different products with 2.5 dimensions can be produced. The materials can be registered, thru-cut, creased, perforated, routed, graved, notched, punched, kiss-cut, v-cut, scored, and drawn. This wide range of methods can be used in a lot of segments like graphics, packaging, leather, textile, composite and technical textiles.
Customers all over the world want to produce individual parts for their use as fast as possible. To do this, both an optimal order of the registration points and geometry is necessary. This problem is called the Traveling Salesperson Problem, short TSP. Such a problem is illustrated in fig. 1 and 2. It shows the 26 cantonal capitals of Switzerland. The TSP is one of the most analyzed combinatorial problems and (with today’s knowledge) not solvable within polynomial time. Here, it even gets more complex, because cutting objects instead of points have to be sorted. Until now two different algorithms are used. A very simple one to sort before production and one which the user has to start explicitly.

Objective: The project aims to analyze this problem with different heuristics on different chosen test jobs. 14 different jobs, like the capitals of the United States or the cantonal capitals of Switzerland, random points, or packaging jobs have been taken. And these will be analyzed with constructive algorithms, like Nearest Neighbor, Random, Best of Random, Best Global Edge, Maximum Regret, Best Insertion, Pilot Method, Beam Search, Genetic Algorithm, Ant Colony, Simulated Annealing, Branch and Bound (see fig. 3). In addition there are also improving algorithms, like Two-Opt, Three-Opt, Node Shift, Node Swap and Change Start. Thereby the quality of the needed calculation time should be considered as quality to be measured. The output is an algorithm so that Zund Systemtechnik AG can offer faster workflows to its customers.

Result: After implementing all the constructive algorithms, it became clear that the complexity is too high and therefore the run time too long and not useful for this application. Sometimes the optimal solution has been found but only with test jobs having few points. Next, we implemented the improved algorithms, and they resulted in a good solution with a short run-time. The Two-Opt resulted often in better solutions than the Three-Opt. The Node Swap and Node Shift resulted with bad solutions in a similar time as the Two-Opt. But all of these algorithms have a different neighborhood, with all the solutions which can be reached from the current one. Therefore the best solution was to combine them and repeatedly run them in sequence. This results in a larger neighborhood to escape local minimums.
For smaller jobs a smaller number of iterations is sufficient, but for bigger jobs, more iterations are needed. Therefore we suggest to make one million iterations for each run, but abort if not a better solution has been found in the last ten thousand iterations. On a Windows Surface Book 2 the total run time is a most 100ms, while most jobs stop earlier. With this solution, we ensure that the customer gets a good solution on every run but can also search for the optimum for a given time.