Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. Rosetti, eds.

AN EFFICIENT SOLUTION FOR K-CENTER PROBLEMS

Peter Hillmann, Tobias Uhlig, Gabi Dreo Rodosek, and Oliver Rose

Universität der Bundeswehr München
Neubiberg, 85577, GERMANY

ABSTRACT

The facility location problem is a well-known challenge in logistics that is proven to be NP-hard. In this paper we specifically simulate the geographical placement of facilities to provide adequate service to customers. We analyze the problem and compare different placement strategies and evaluate the number of required centers. We simulate several existing approaches and propose a new heuristic for the problem.

1 INTRODUCTION

The typical geographical facility placement problem originates from the application area of logistic and transportation. Locations of central warehouses have to be determined with a short distance to customers to provide adequate service and convenient access. Facility location decisions have a strong long-term effect. The facility location problem is based on the classic k-center problem. We identified three fundamentally different constraints for the placement in relation to the infrastructure. These are:

- Free placement: The location can be determined completely free and without any restriction.
- Infrastructural placement: Existing infrastructure should be used for the placement.
- Node placement: Only a limited amount of given locations are viable to place a facility.

A company aims for short transportation paths from their warehouses to the customers to reduce transportation costs and transportation times. These costs are balanced against the operational costs of multiple warehouses. According to the application areas, we need to answer the following questions:

- Where should warehouses be placed geographically in order to obtain short transportation paths?
- What is the necessary amount of warehouses to minimize the cost of operation and transportation?
- How large is the performance difference between the constraint-free and the node placement scenario?

2 OUR APPROACH DRAGOON

Based on the k-Means strategy, we developed a new algorithm Dragoon (Diversification Rectifies Advanced Greedy Overdetermined Optimization N-Dimensions). In the preliminary stage of the initialization phase, an orientation node is placed at the optimal position comparable to the one center node case. Afterwards, the specified amount of center nodes is placed using the 2-Approx strategy. It chooses the node with the largest distance to their closest center node as new location of the next center node. After the initialization, the algorithm starts with the iterative refinement. In every iteration step, the vertices are (re)assigned to their nearest center nodes. Afterwards, an updated location is calculated for every cluster of vertices related to a center node. The algorithm tests all possible locations around the current center restricted by the current cluster. The new location is chosen after the best improvement. In each iteration, every center node is allowed to shift its location only once. This iterative optimization is repeated until all center node locations do not change any more. For the node and infrastructure placement constraint, the algorithm tests all locations of grouped vertices for a center. For the free placement constraint, our algorithm tests all points on a grid with a defined distance (ε). If one of the tested locations results in a better performance, this location is used for the next iteration step. If no location leads to an improvement, we successively increase the granularity of the grid ($\varepsilon_{new} := \varepsilon_{old}/2$). This process is repeated until the grid distance ε is smaller than the maximal accepted deviation.

3 SIMULATION AND ASSESSMENT

The evaluation of the algorithm is based on experiments using a prototypic implementation in Java. We use more than 10 scenarios. The set consists of a randomly generated and realistic test scenarios. As fitness function we calculated the distance parameter: maximum. Based on the 2-approximable results *s* of other algorithms we define a theoretical limit for the optimum.

Figure 1 shows the results of the algorithm comparison. For small center node amounts, our improved Dragoon algorithm is close to the theoretical optimum. For larger center node amounts the algorithms stagnate with their performance nearly at the same level, referring that we are already very close to the actual optimum. Our approach performs significantly better than the others and is much faster than a brute force approach. Figure 2 presents the difference between free placement and node placement. The distance deviations constraints are on average 4% and in the worst case 11%. We observe that the node placement approach needs on average 2 centers more to compensate the more flexible positioning of the free placement. In the worst case, 6 centers more are required. Based on the maximum distance of a vertex to its nearest center node, it shows that it is sufficient to set up about 6% of the vertices as center nodes. After we placed 58 center nodes in the normalized scenarios, the average improvement of maximum distance for an additional center node is less than 1%.

In line with our initial intention, to set up warehouses for a specific scenario, the costs for transportation as well as operating and setup costs have to be respected. With an increasing amount of center nodes, the transportation distance and cost is reduced, whereas the set-up and operating cost is increased. To find the optimal balance between these aspects, we use simulation based optimization to calculate the optimal amount and location of center nodes for specific scenarios, see Figure 3.

We tested all algorithm with more than 1000 nodes and 200 center nodes. The calculation time is less than one minute, except for evolutionary algorithms and lineare programming.

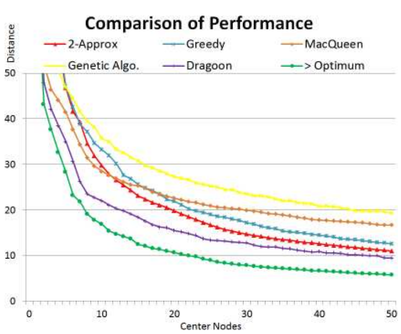

Figure 1: Node Placement: Performance comparison of the algorithms for maximum distance.

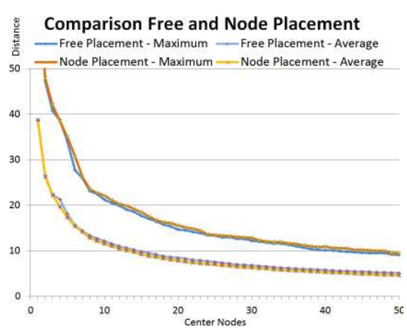

Figure 2: Deviation between free and node placement for maximum and average distance (Dragoon).

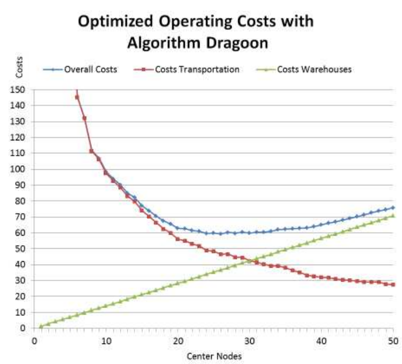

Figure 3: Optimized operating costs with our algorithm Dragoon.

4 CONCLUSION AND OUTLOOK

We propose the novel algorithm Dragoon to solve the k-center problem with geographical placement. Our strategy outperforms the other approaches, reaches very good results close to the global optimum. We calculated the distance deviations between the different placement constraints (free vs node). A slight difference on average of 4% is observed for the maximum distance. In the worst case, it can be up to 11% distance difference between the most flexible case and most restrictive case. Our analyses show that even the best placement strategy reaches less than 1% performance gain by adding an additional center node after an amount of a center node ratio of about 6% is reached.