The p/q-ACTIVE Facility Location Problem: Investigation of the solution space and an LP-fitting heuristic

Anders Dohn   Søren Gram Christensen   David Magid Rousøe
Informatics and Mathematical Modelling, Technical University of Denmark
Kgs. Lyngby, Denmark

7th July 2005

Abstract

The p/q-ACTIVE Uncapacitated Facility Location Problem is the problem of locating p out of n possible facilities each serving at least q out of m given clients at a minimum cost. The problem is an extension of the Uncapacitated Facility Location Problem (UFL) also considering constraints on the number of facilities and their minimum activity. An example of the use of this formulation could be the opening of p new schools where each must have at least q pupils. p/q-ACTIVE is NP-hard like the UFL.

In this paper we present a thorough investigation of the p/q-ACTIVE UFL and propose a heuristic solution method. Different geometric and random cost problem instances are considered. Experiments show that 60% of the problems can be solved to optimality just by solving the corresponding LP-relaxation. Using a simple local search heuristic, the geometric problems are solved with an average gap of 0.1% to the lower bound of the LP-relaxation. An effort is put into isolating problem types that are hard to solve. Problems with low p, pq close to m combined with clustered clients or a low variation in the facility opening cost are most likely to give results worse than average. Gaps up to 8% are observed in the worst cases.

Keywords: (p/q-ACTIVE, Uncapacitated Facility Location, Heuristic Solution Methods, LP-relaxation, LP-fit, MIP-heuristics)

1 Introduction

The *p/q-ACTIVE Uncapacitated Facility Location Problem* (p/q-ACTIVE) is the problem of locating p out of n possible facilities each serving at least q out of m given clients at the minimum total cost. The problem is a natural extension to the UFL which can be made p-ACTIVE by demanding that clients should be served by exactly p facilities. The open facilities should serve at least one client, making them *active*. If the problem is to locate schools in a city area it does not seem desirable to open a school serving only one pupil. Therefore, it is required that an active facility serves at least q clients. The total cost of the solution is the cost of opening the facilities plus the cost of serving each of the clients given the allocation of these.

In figure 1, an example is shown with 50 sites to choose from and 250 clients to serve. To the right the optimal solution to the problem is shown when $p = 5, q = 40$, the cost of opening facilities are randomly generated values and client costs are proportional to the distance.

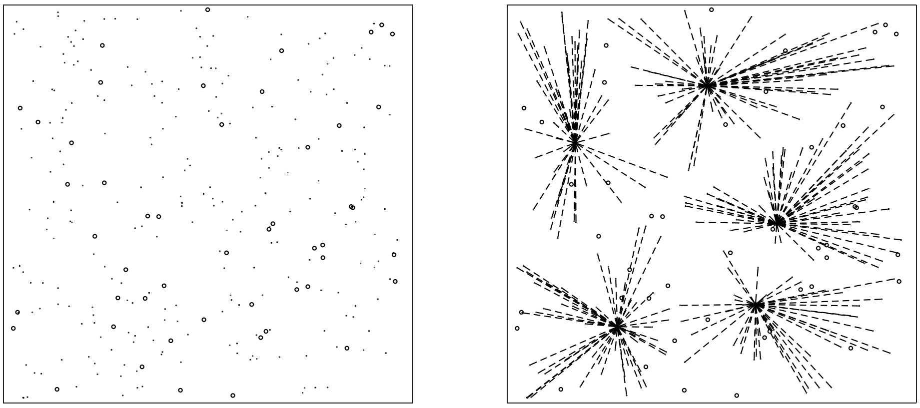


Figure 1: Example of a p/q-ACTIVE (left) and its optimal solution (right). Clients are marked with dots and possible facility locations are marked with circles. The values of p and q ($p = 5, q = 40$) are visible on the right map. Exactly 5 facilities have been opened and each of these serves at least 40 clients.

The problem was formulated at a conference by Krarup, Leopold-Wildburger and Pisinger [3]. M.Sc. J. B. Wanscher is the only one who has published actual research on p/q-ACTIVE [1]. He developed a branch and bound algorithm with bounds generated by a dual ascent heuristic. The main focus of his work was to produce good lower bounds.

Our first goal was to find close primal bounds to the problems created by Wanscher, using a metaheuristic approach. We found that all problems considered were easily solved to 0.5% from a lower bound obtained by an LP-relaxation. The problems were constructed at random with the Euclidean distance as the cost measure between facilities and clients. Facilities and clients were uniformly distributed on a square map. Moreover, it was shown that the LP-relaxation of problems with up to 300 sites and 3000 clients often results in feasible solutions to the integer problem and that there is a strong connection between the probability of finding IP-feasible solutions and the fraction $\frac{pq}{m}$, denoted the *coverage*.

The main focus of this paper is to investigate many different problem structures and find their properties with respect to the LP-relaxation. An algorithm that benefits from the good lower bounds obtained by the LP-relaxation is proposed.

In the next section we give the formulation of p/q-ACTIVE. A decomposition of the model, which will be used in our algorithm, is also proposed. In the subsequent section initial tests are performed. The goal of the initial tests is to examine the characteristics of problems with different cost structures, such as uniform geometric distribution, geometric distances with clustered distribution and even completely random costs. Finally, we describe the algorithm and test it on a wide range of problems, including random generated problems and problems known from the OR-library [5] and the TSP-library [6].

2 Mathematical Formulation

2.1 p/q-ACTIVE Uncapacitated Facility Location Problem

p/q-ACTIVE can be described as UFL with additional constraints.The following is given: Let $N = \{1, \dots, n\}$ be the set of potential facilities (also referred to as sites). Each facility j has an opening cost f_j. Furthermore let $M = \{1, \dots, m\}$ denote the set of clients where $c_{ij} \geq 0$ is the cost of serving client i from facility j. The two sets of binary variables y_j and x_{ij} are defined as follows:

$$
\begin{aligned}
y_j &= \begin{cases} 1 & \text{if facility } j \text{ is open} \\ 0 & \text{otherwise} \end{cases} \\
x_{ij} &= \begin{cases} 1 & \text{if client } i \text{ is served by facility } j \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

The problem is to satisfy the demand of all clients at the least total cost given that p facilities are opened and at least q customers are served from each facility. The p/q-ACTIVE model can hence be written as:

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j \tag{1}$$

s.t.

$$
\begin{aligned}
\sum_{j \in N} x_{ij} &= 1 && \forall i \in M && (2) \\
x_{ij} &\leq y_j && \forall i \in M, \forall j \in N && (3) \\
\sum_{j \in N} y_j &= p && && (4) \\
\sum_{i \in M} x_{ij} &\geq q y_j && \forall j \in N && (5) \\
x_{ij} &\in \{0, 1\} && \forall i \in M, \forall j \in N && (6) \\
y_j &\in \{0, 1\} && \forall j \in N && (7)
\end{aligned}
$$

Here, (4) and (5) are the additional constraints, compared to UFL, regarding the number of open facilities and the number of clients served from each facility.

We require that $p \geq 1$ and $q \geq 1$. Furthermore it is obvious that $p \leq n$ and $pq \leq m$ must hold. The coverage is defined as $\frac{pq}{m}$.

The UFL can be reduced to n p/q-ACTIVEs in polynomial time. This is done by setting $q = 1$ and $p = 1, \dots, n$. The UFL is an NP-hard problem [8] and consequentially p/q-ACTIVE is NP-hard.

2.2 Decomposition

If the locations of the facilities are known the allocation of the customers can be found in polynomial time, thus we may split p/q-ACTIVE into two problems: A master problem taking care of the location of the p facilities giving the subset of facilities $P \subseteq N$, and a subproblem allocating the clients to the open facilities in the least expensive way. So as soon as we have decided which sites are active, we only need to solve one subproblem. Mathematically, the subproblem can be expressed as:

$$\min \sum_{i \in M} \sum_{j \in P} c_{ij} x_{ij} \tag{8}$$

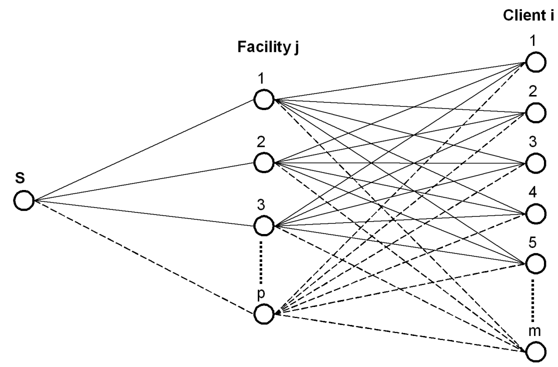
s.t.

$$\begin{aligned} \sum_{j \in P} x_{ij} &= 1 && \forall i \in M && (9) \\ \sum_{i \in M} x_{ij} &\geq q && \forall j \in P && (10) \\ x_{ij} &\in \{0,1\} && \forall i \in M, \forall j \in P && (11) \end{aligned}$$

This is a classic transportation problem which is easily solved as the constraint matrix is known to be totally unimodular. Thus, to this problem the LP-relaxation always yields integer solutions.

2.3 A network formulation

When solving the subproblem the LP-solver `CPLEX 9.0` is used, and as shown in [2] it turns out, that there are computational advantages of formulating the subproblem as a network problem enabling `CPLEX` to use network simplex.



Quantity	1	p	p	p·m	m
Supply	m		0		0
Demand	0		0		1
Cost		0		c_{ij}	
Bounds		[q ; m - q · (p-1)]		[0 ; 1]	

Figure 2: Network formulation of the subproblem.

In Figure 2, the network is represented. Below the graph, the number of nodes and arcs are shown as well as the supply/demand in the nodes and the costs and capacities of the arcs. The aim is to find the cheapest way to "send" the clients from the node s via a facility and "home". With the capacities and demands shown, it is clear that this model is equivalent to the subproblem. As shown on Figure 2 the total number of nodes in the network is $p + m + 1$ and the total number of arcs is $p(m + 1)$.

3 The cost structure

In the general formulation of the problem the structure of the allocation costs, c_{ij} and the location costs, f_j are not specified. These two measures define the cost structure of the problem. To make a thorough investigation of p/q-ACTIVE, we investigate different cost structure scenarios. For *geometric problems* the cost c_{ij} of allocating a client to a facility is measured as the distance between these multiplied by a weight assigned to the client. The average cost of opening a facility f_{avg}, is calculated as:

$$f_{avg} = Kc_{avg}\frac{m}{p}$$

where c_{avg} is the average allocation cost, $\frac{m}{p}$ is the average number of allocations to an open facility and K is a cost ratio between clients and facilities, which can also be varied. The individual facility costs are now chosen randomly from a uniform distribution between $\frac{1}{2}f_{avg}$ and $\frac{3}{2}f_{avg}$.

By varying c_{ij} and K, the following different problem instance types are obtained:

STD Standard problem structure. Euclidian norm distance, uniform distribution of clients and sites, $K = 1$ and unit weights on clients.

CLU As **STD** but with the distribution of clients and sites clustered. 15 different clustered distributions are considered, both varying size and density of client- and site-clusters. See section 6.1.1 for details.

RATIO As **STD** but with the cost ratio between site cost and client cost varying. $K = 0.001, 0.01, 0.1, 10, 100$

NORM As **STD** but with the type of norm varying, using: ℓ_1, $\ell_{1.5}$ and ℓ_∞. [1]

WEIGHT As **STD** but with weights on the clients varying uniformly between 1 and 1000.

RAND Problems with a totally random client cost matrix and $K = 1$.

A thorough computational investigation is now performed to reveal the influence of instance type on the difficulty of p/q-ACTIVE.

4 Initial tests with an LP-solver

The difficulty of the problem instances at hand is expected to depend on the particular cost structure of the instance and we aim at revealing such dependencies in the tests. The instance types that are harder to solve will be identified.

4.1 IP-feasible solutions to the LP-relaxation

To get a lower bound for the solution value of an instance the integer constraints are relaxed. If the solution of the LP-relaxation is feasible for the integer problem (IP-feasible), it is also optimal. The value is always a lower bound on the integer problem. The LP-relaxation is solved using the dual simplex method in `CPLEX`, since this method in average has shown to be the fastest for p/q-ACTIVE.

A test is made to examine the quality of the results from the LP-relaxation. First of all, it is interesting to find the number of solutions which are integer feasible and consequently optimal. In Table 1 it is shown that this is actually a large fraction of the solutions.

	STD	**CLU**	**RATIO**	**NORM**	**WEIGHT**	**RAND**
Problems	1440	1440	1440	1440	1440	1440
IP-feasible	882	760	956	881	1.041	1.052
	61.3%	52.8%	66.4%	61.2%	72.3%	73.1%

Table 1: Average IP-feasibility of p/q-ACTIVE.

All the problems have a fixed number of sites equal to 100. As demonstrated later the conclusions do not depend on this choice. In the tests the number of clients, m, varies between 100 and 2500, p takes values in the interval between 2 and 98 and q is chosen so the coverage is in the interval between 5% and 95%.

With a fixed number of sites a test is conducted on **STD**-Problems revealing a clear relation between the coverage and the procentage of IP-feasible problems (Figure 3 - left). A relation between p and the number of problems with IP-feasible optimum of the LP-relaxation is also observed (Figure 3 - right). For small values of p a larger percentage of the problems are IP-feasible. Data for $m = 100, 2500$ show the same results. Taking the average of the data on the right graph gives the relation shown on the left graph for $m = 500$. The

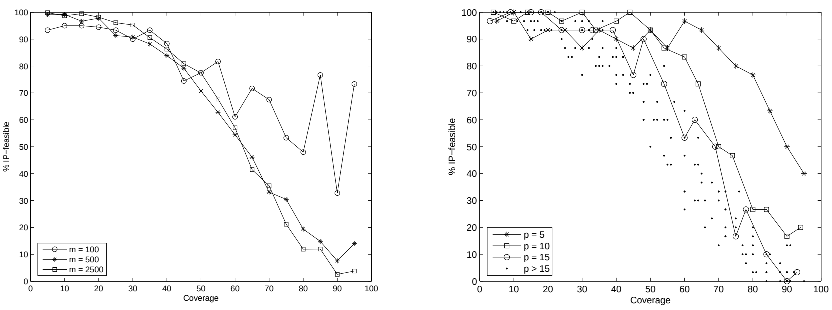


Figure 3: Relation between the coverage and the procentage of IP-feasible **STD**-problems. The average of all p-values (left). Values for $m = 500$ (right).

fluctuations for $m = 100$ on Figure 3 (left) is a consequence of the settings of p in the test.

Similar test have been made for the other problem instance types defined. The conclusions on the dependencies of coverage and p are the same for all the geometric problems as the ones stated above for the simple geometric problem. However looking at the **RAND**-Problems the IP-feasibility is almost independent of the coverage. Now the dependency is solely on the value of p. For $p > 20$ almost all problems are IP-feasible. This is demonstrated in Figure 4.

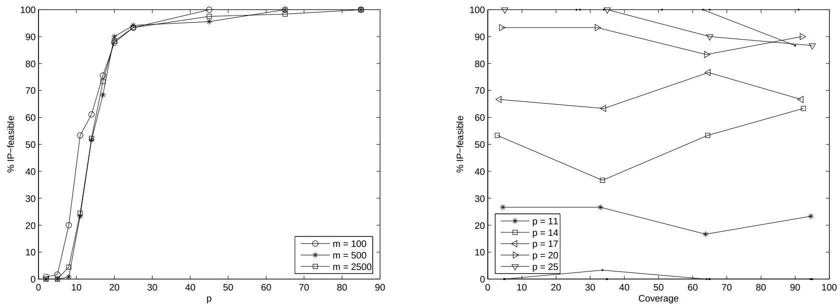


Figure 4: Relation between p and the procentage of IP-feasible problems for **RAND**-Problems (left). Relation between the coverage and the procentage of IP-feasible 100x500 **RAND**-Problems (right).

4.2 Quality of non-optimal LP-relaxations

The quality of the LP-relaxations are measured as the gap to the optimal integer solution. In the previous section it was shown that for many problems the LP-relaxation yields feasible solutions to the original problem. In this section the remaining LP-solutions will be compared to the optimal IP-solution.

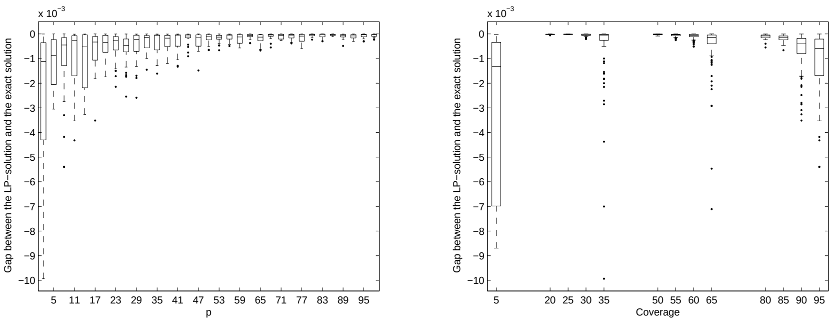


Figure 5: Gaps between the LP-value and the exact solution value for 100x500 **STD**-Problems.

Figure 5 shows an ordinary box and whisker plot, where the upper and lower lines of the "box" corresponds to the 25th and 75th percentile of the gaps. The line inside the box indicates the median. The whiskers are calculated as the minimum of 1.5 times the interquartile range[2] and the distance to the point furthest away. Points outside this range indicate outliers. In Figure 5 the gap size is observed to depend on both p and the coverage. p has the major effect on the gap as it is possible to get large gaps even when the coverage is low as long as p is small. The peaks on the right graph is a consequence of this. The figure shows that an instance with a relatively low p which does not have an IP-feasible solution to the LP-relaxation (which is seldom the case as shown on Figure 3 - right) is more likely to have a solution with a large gap. The values for coverage equal to 5% calls for an explanation. In the relevant problem the

[2]interquartile range: the distance between 25th and 75th percentile

only way to get low coverages is to keep the value of p relatively small. In Figure 6 (page 11) this effect will also be present.

More importantly the range on the axis indicates that the gaps are in general very small. The mean value for the **STD**-Problems observed on Figure 5 is 0.05%. The same behaviour is also observed for the other geometric types. There are, however, in the different instance types some outliers which have gaps up to 2%. This will be further investigated in section 6.1.

The initial tests give a strong indication that the solutions to the LP-relaxation of the problem give very tight lower bounds. Actually, more than 60% of the LP-solutions were proven optimal. Advantage of this can be taken when considering the feasible solutions (upper bounds).

5 A heuristic solution method

In this section a local search heuristic that fast and efficiently solves p/q-ACTIVE is developed. In section 2.2 it was shown that the problem of allocating clients can be solved in polynomial time when the facility locations are given. Therefore the heuristic will only deal with the location of facilities. Whenever we refer to a solution only by the facility locations the allocation of clients in the solution is optimal with respect to the locations. This is achieved by solving the subproblem whenever a solution to the master problem is considered.

When an initial solution is known, the heuristic searches part of the solution space, the *neighborhood*, and by some criteria a new solution is chosen until a stopping criteria has been reached. The first step is to define the solution space.

5.1 Solution space

The solution space S of the master problem can be described by the location variables, and it is defined as the solutions where exactly p facilities have been opened:

$$S = \mathbb{B}^n \text{ with the number of 1's equal to } p$$

The size of the solution space is then:

$$\binom{n}{p} = \frac{n!}{(n-p)!p!}$$

5.2 Initial solution: LP-relaxation with integer fit

In the preceeding section it was demonstrated that the LP-relaxed problem often yields integer solutions and if not, gives a good lower bound.

This gives rise to the idea of fitting the infeasible relaxed solution to the feasible space and thereby hopefully find a good initial solution to the problem. The most obvious way to find a feasible solution is to choose the p sites that have the largest values of y_j. As the process of choosing sites to the initial solution based on the LP-solution is very fast, other selection strategies are considered as well. Instead of just choosing the p sites with largest y_j values, a site will be chosen if the y_j value is greater than or equal to a threshold $\hat{y}$ $(0 < \hat{y} \leq 1)$, that is: A facility is chosen if $y_j \geq \hat{y}$. The set of sites chosen by the selection criteria above is called P_{LP}. If this criterion is used there is no guarantee that the correct number of facilities will be opened. Therefore strategies for choosing extra sites and eliminating sites are necessary.

Three cases have to be considered:

- $|P_{LP}| = p$. The right number of facilities has been chosen and the subproblem can be solved. This is the same as opening the p facilities with largest y_j-values as described above.

- $|P_{LP}| > p$. Too many sites are opened. Now the problem is to determine which of the sites in P_{LP} to keep open. A greedy algorithm is used to choose between the selected sites. Like the method to find initial solutions discussed in [2], this greedy algorithm chooses the best of several solutions based on either locating cheapest facility first, allocating the cheapest costumers first or allocating expensive costumers first.

- $|P_{LP}| < p$. Too few sites have been selected by the LP-fit and extra sites should be added. This is done by using the same greedy algorithm as above, but selecting from the sites in $N \setminus P_{LP}$. If the greedy algorithm is used without further changes, the extra sites that in the cheapest way can service all clients will be chosen. Because the sites in P_{LP} should also serve some clients (probably most clients), the extra sites chosen by the greedy algorithm do not have to serve all the clients. It is investigated if it is a better strategy to disregard some of the clients when using the greedy algorithm. It is not known which clients should be assigned to the facilities in P_{LP}. Therefore it is tested to exclude different fractions of the clients. Disregarded clients are considered assigned to a facility in P_{LP}.

  The following disregarding strategies, B_{type}, that determines the number of clients, m_{nb}, available to the facilities in $N \setminus P_{LP}$, are tested:

1.	m	4.	$(p - \|P_{LP}\|)q$
2.	$m - \|P_{LP}\|q$	5.	pq
3.	$(p - \|P_{LP}\|)m/p$	6.	$m/2$

  To find out which clients to block they are sorted by their x_{ij} values, where $j \in P_{LP}$. Then the $m - m_{nb}$ clients with the highest values are considered as already allocated. This is done as they most likely will be assigned to one of the facilities in P_{LP}.

Extensive tests are made both varying $\hat{y}$ and B_{type} strategy. The best combinations are chosen. The gap between the objective value and the lower bound is used as a measure of the quality of the solution.

In the test $\hat{y}$ is varied between $0.05 - 1.00$ with a step of 0.05 and for each step all B_{type} strategies are tested, to choose the best combinations of these parameters. The test is made on 1187 problems. 64.8% of the problems are best solved by simply choosing the p sites that have the largest y_j values. These are removed. This is done so that the methods will complement each other, i.e. different methods find good solutions to different types of problems. Of the remaining 418 problems, 256 are best solved by setting $\hat{y} = 0.15$. In this case too many sites are almost always selected and the B_{type} has minor relevance. Of the 162 problems left, 55 are best solved by $\hat{y} = 1$ and $B_{type} = 4$. In this way the following 8 choices of parameters are made:

1. $\hat{y} = 0.15,\ B_{type} = 4$	5. $\hat{y} = 0.55,\ B_{type} = 5$
2. $\hat{y} = 0.35,\ B_{type} = 4$	6. $\hat{y} = 0.75,\ B_{type} = 3$
3. $\hat{y} = 0.50,\ B_{type} = 5$	7. $\hat{y} = 1.00,\ B_{type} = 2$
4. $\hat{y} = 0.55,\ B_{type} = 4$	8. $\hat{y} = 1.00,\ B_{type} = 4$

The B_{type} is only used when too few sites are selected. In the other case the best of the selected sites are chosen, as described earlier. Using these 8 methods only 37 of the 1187 problems can be solved better by applying a method not already selected and the improvement is minimal.

5.3 Neighborhood

For a feasible solution s, the neighborhood $N(s)$ is defined as the solutions $s' \in S$, that can be constructed by closing a facility in s and opening one not in s:

$$N(s) = \{s' : s' \in S \wedge D_H(s, s') = 2\}$$

where $D_H(s, s')$ is the *Hamming-distance* between the two solutions. The size of the neighborhood is $p(n-p)$ as there are p possibilities of choosing an open facility and $n-p$ possibilities of choosing a closed facility.

5.4 *First Better Admissible* search

With the initial solution and the neighborhood defined, a local search can be performed. As shown in [2] the *steepest decent* approach produces good and often optimal solutions but it is also very time consuming due to the size of the neighborhood and the solution time of the subproblem. To reduce the solution time, a *First Better Admissible* (FBA) search is used. This strategy is a modification of the steepest descent algorithm, where instead of searching through the entire neighborhood to find the best solution, the first better solution found in the neighborhood is chosen. This means that more neighborhoods will be searched in the same amount of time, but of course each neighborhood is not investigated completely. A local minimum has been reached when the whole neighborhood has been searched without improvements, just as for the steepest descent.

6 Test of the algorithm

The proposed algorithm is tested on a wide range of different problem types, covering all of the specified structures (section 3). In the following only problems that are not IP-feasible are used, bearing in mind that this is less than 40% of all the problems generated. Also the gaps to the LP-relaxation and not to an exact solution are considered.

In Table 2 the mean gaps for the problems listed in Table 1 (section 4) are shown. Only the problems that were not IP-feasible are considered and gaps for both the LP-fitter alone and with the FBA-search are calculated. In this way the calculated means are rather pessimistic measures as they do not take all the instances in the test into account.

Type	STD	CLU	RATIO	NORM	WEIGHT	RAND
Problems	1440	1440	1440	1440	1440	1440
IP-feasible	882	760	956	886	1.041	1.052
	61.3%	52.8%	66.4%	61.5%	72.3%	73.1%
Gap of FIT	0.083%	0.061%	0.413%	0.080%	0.045%	3.884%
Gap of FBA	0.070%	0.055%	0.178%	0.068%	0.029%	3.705%
Std deviation FBA-gap	0.120%	0.332%	0.240%	0.134%	0.058%	1.822%

Table 2: The mean gap for the LP-fitter alone and with the FBA improvement. Standard deviation of FBA-gap.

From Table 1 it is observed that the average gaps for all the geometric problem types are around 0.1%. It is also clear that the FBA-search improves the fitted solution. Thus, the fitter does not always give locally minimal solutions. The completely randomized problems yield higher gaps than the other problem types. We also observe that **RATIO**-Problems yields higher gaps than the **STD**-Problem.

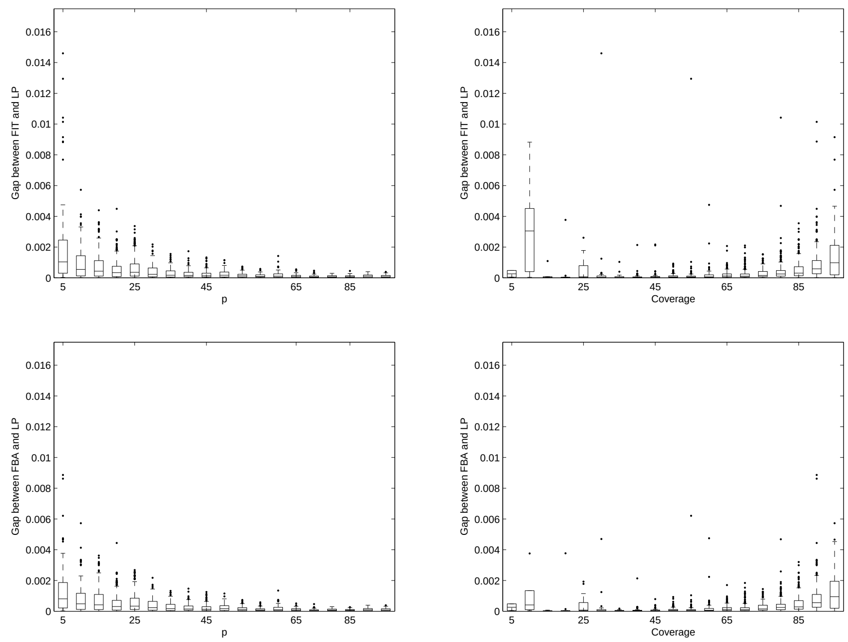


Figure 6: 100x500 **STD**-Problems. The relation between p and the gaps (left). The relation between the coverage and the gaps (right).

If we take a closer look at the **STD**-Problem, Figure 6 shows that there is a clear connection between the gaps and the value of p. The link between the coverage and the gaps is also evident. Large gaps are encountered when p is low or the coverage is close to 100%. This is evident for all the 6 different problem types. Remembering that the same connection was observed between the exact solution and the lower bound obtained by the LP-relaxation (Figure 5 page 7), this may indicate that the deviation between the gaps solely originates from the lower bounds. When comparing to exact solutions this is seen not to be the case. A part of the gap of course is from the lower bound, but the gap from the primal value to the exact solution does have the same dependency as what can been interpreted from Figure 6.

6.1 The hard problems

In the following the different problem instance types are considered one at a time in a search for instances that yield worse results than the average case shown in Table 2.

6.1.1 CLU - Clustered problems

Regarding clustered problems we introduce the settings used for the tests. The clusters can be either small or large and the number of clients/facilities in each cluster is varied so that they are either dense or sparse from the following definition:

· **Small:** Having a width and a height in the interval $\left[\frac{1}{16};\ \frac{1}{8}\right]$ of the total width and height.

○ **Large:** Having a width and a height in the interval $\left[\frac{1}{4};\ \frac{1}{2}\right]$ of the total width and height.

⊡ **Sparse:** Each cluster contains a fraction of the clients/facilities chosen randomly in the interval $\left[\frac{1}{16};\ \frac{1}{8}\right]$.

■ **Dense:** Each cluster contains a fraction of the clients/facilities chosen randomly in the interval $\left[\frac{1}{4};\ \frac{1}{2}\right]$.

The height and width of clusters are drawn at random from the same interval giving the clusters an almost quadratic shape. In Table 3 all the problem types constructed from the above definition are illustrated. The problems with large and sparse clusters for both clients and facilities are omitted because this setting produces problems much like the **STD**-Problem.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Clients	○	○	○	·	·	·	·	○	○	○	○	·	·	·	·
	■	⊡	■	■	⊡	■	⊡	■	⊡	■	⊡	■	⊡	■	⊡
Facilities	○	○	○	○	○	○	○	·	·	·	·	·	·	·	·
	■	■	⊡	■	■	⊡	⊡	■	■	⊡	⊡	■	■	⊡	⊡

Table 3: The defined cluster types.

Among the clustered problems created, the problems differing the most from the **STD**-Problem are the ones with small and dense clusters.

In Figure 7 the results from the test are illustrated. From the right graph it is seen that the cluster types causing larger gaps than the **STD**-Problem are types: 3, 4, 5 and 13. The common property for these problems is that they all have small client clusters. In type 3, 5 and 13 where the results differ

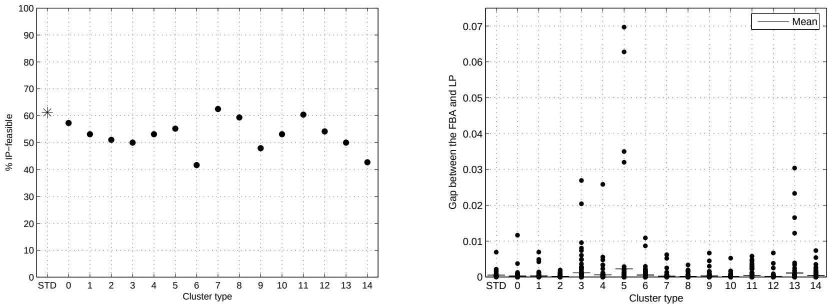


Figure 7: Quality of the heuristic solution for different cluster types. Connection between cluster types and IP-feasibility (left) / FBA-search gaps (right).

the most from the **STD**-Problem the client clusters are small and dense. This test indicates that the most difficult cluster problems are the ones that have small and dense clusters of clients and a spread out distribution of facilities. It is important to notice that these conclusions are based on the outliers in the graph. When generating problems randomly, a large part of the problems will not differ from the **STD**-Problem and one have to look for the problems standing out. In particular, the 500 client problem giving the worst result was captured during the test. This problem was the second worst problem plotted in Figure 7 (of cluster type 5). In Figure 8 the problem is shown. As discussed in connection with Figure 6, the highest gaps are obtained when p is low and if the coverage is high. This is also the case in the problem shown on Figure 8, where $p = 5$ and the $q = 95$ resulting in a coverage of 95%.

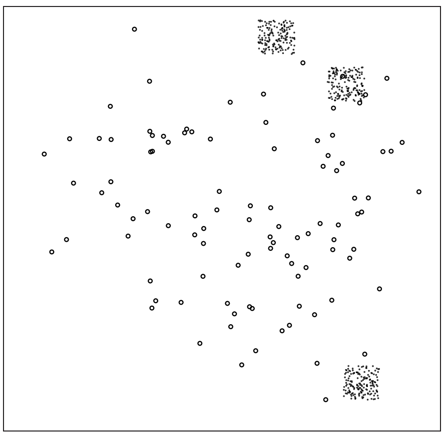


Figure 8: Randomly generated clustered 100x500 problems. Client clusters are small and dense while the facility clusters are large and sparse. $p = 5, q = 95, Gap = 6.28\%$.

Figure 7 also demonstrates that it is not the same types that are performing poorly concerning IP-feasibility as the ones giving the large gaps. Types 6 and 14 are significantly less IP-feasible than the **STD**-Problem. An interesting point about the IP-feasibility is that the "contrary" problem types to type 6 and 14 are the types 7 and 0 respectively. These two types reveal some of the best results regarding IP-feasibility, indicating that problems with a large number of small clusters are harder to solve by LP-relaxation.

6.1.2 RATIO - Variation in the cost ratio between clients and facilities

As it was observed in the beginning of this section the problems where the ratio is varied seems harder to solve than the **STD**-Problem. We now examine whether some settings are worse than others. Figure 9 shows that the dependency on the factor K is very high. If K is large the chance of IP-feasibility increases, and for the problems that are not IP-feasible the heuristic reveals small gaps. This means that if the facilities are much more expensive to place than it is to assign the clients the problem is easy. It has the natural explanation that if it is really expensive to open facilities, it is just a matter of opening the cheaper ones and then worry about the clients afterwards.

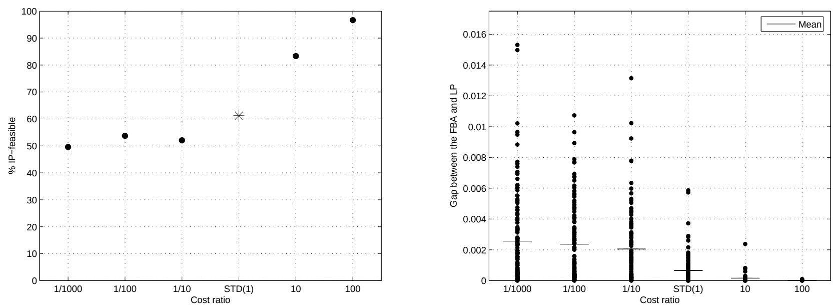


Figure 9: Quality of the heuristic solution for different cost ratios.

6.1.3 NORM - Different norms

The test of the various norms shows that there is no difference between the tested norms, neither on the LP-feasibility nor the gaps obtained from the FBA-solution. This is illustrated on Figure 10.

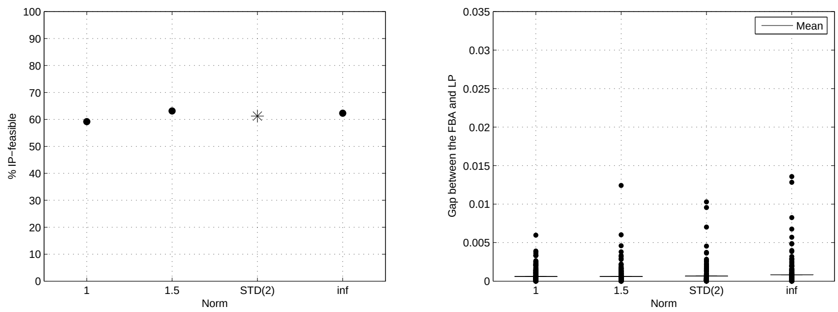


Figure 10: Quality of the heuristic solution for different norms.

6.1.4 WEIGHT - Weighted clients

Table 2 indicates, that this structure “helps” the solver and the results for the weighted problems are better. More problems are IP-feasible and those that are not yield smaller gaps. This result is connected to the findings from the **RATIO**-Problems, that clearly indicate that if some facilities are very expensive the problem is easy. If the clients are weighted, the very heavy clients are the ones to be assigned first; all others can be dealt with afterwards.

6.1.5 RAND - Completely random cost matrix

It can be seen in Table 2 that the **RAND**-Problems are IP-feasible more often than the **STD**-Problems. However the **RAND**-Problems that are not solved to optimality by the LP-relaxation yield a much higher gap than the **STD**-Problems. It is known from section 4.1 that only problems with small p-values are interesting. If p is large it is almost certain that the LP-solution will be feasible for the original problem.

6.1.6 Test summary

The overall result of the tests is that small gaps are found for almost all problems. For special structures there may however be large gaps, particularly if the problem has a low p and a high coverage combined with clustered clients or a low facility opening cost. The conclusion of the test is that the problems are in general easy to solve and yield small gaps, but there are nonetheless outliers.

Preliminary tests have shown that if the settings yielding large gaps in each of the instance types above are combined, the gaps increase dramatically.

6.2 Other problem sizes

To inspect the effect of the number of sites, n, a test is conducted with $n = 200$ and $m = 200, 1000$. p and q are set to vary in the same way as in the previous tests. Table 4 shows the tendency. Having a larger number of sites seems to affect the results in a slightly negative direction. The various dependencies of problem structure and parameters discovered earlier in this section still hold for these new problems, but it is worth noting that there are slightly fewer IP-feasible problems. The gaps have not changed significantly, which is also an important result.

Type	**STD**	**CLU**	**RATIO**	**NORM**	**WEIGHT**	**RAND**
IP-feasible	51.9%	38.0%	58.7%	51.5%	67.3%	76.0%
Gap of FIT	0.09%	0.13%	0.62%	0.11%	0.06%	4.27%
Gap of FBA	0.06%	0.11%	0.17%	0.08%	0.04%	3.95%

Table 4: Test of problems with 200 sites.

6.3 Solution time

A very important aspect that we have neglected in the preceding sections is the solution time. There is actually two parts of the solution time. First the time it takes to solve the LP-relaxation, which in many cases is enough to solve the original problem. If a non-feasible solution is found, the time it takes to perform the LP-fit with FBA-search is also of interest.

6.3.1 LP-solver

In our experiments a state-of-the-art LP-solver is used as described earlier, and this limits the possibility of lowering the solution time. It is however interesting to examine how the problem types and the p/q settings affect the solution time for the LP-solver. We look initially at the **STD**-Problem. As shown in Figure 11 there is a strong correlation between p and the solution time. The connection to the coverage is also evident. The combination of a low p and a high coverage leads to higher computational times.

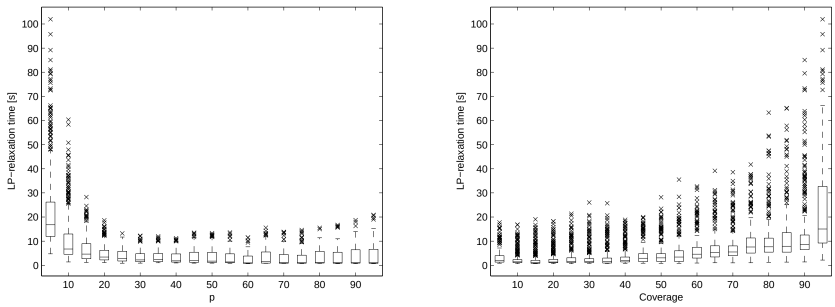


Figure 11: Solution times for the LP-solver for the 100x500 **STD**-Problem.

The number of variables also has a high impact on the solution times. Looking at the values on the y-axis of Figure 12 it is clear that solution times increase dramatically when the number of clients is increased. This is a natural consequence as the number of variables increase with the number of clients.

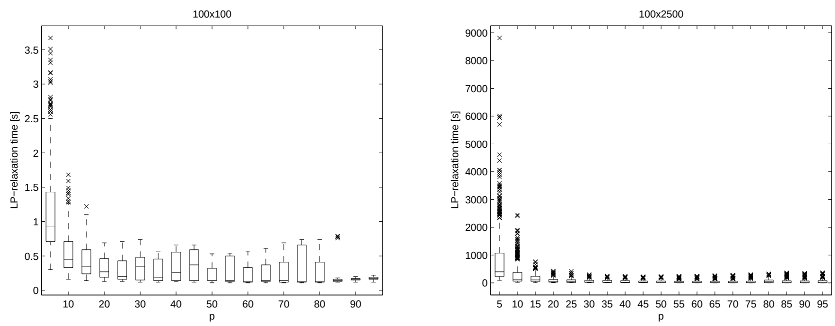


Figure 12: Solution times for the LP-solver for the 100x100 problems (left) / 100x2500 problems (right).

Turning to the other problem types considered in this article one significant difference is observed. The completely randomized problems are much more time consuming for the LP-solver. Table 5 presents an overview. For each problem type the average and median solution time for problems of size 100x500 has been calculated. The difference between these two numbers is small for the randomized problems, showing that the problems in general use the average time. For all the geometric problems the average is much higher than the median, indicating some outliers with very high solution time.

Type	GEO	CLU	RATIO	NORM	WEIGHT	RAND
Average [s]	13.4	5.0	15.7	10.8	9.9	130.5
Median [s]	3.2	3.2	7.1	2.2	4.0	108.2

Table 5: Average and median of the solution times for 100x500 problems.

6.3.2 FBA-search

The time used to make an FBA-search depends on three things. The size of the neighborhood, the number of neighborhoods searched in total and the solution time of the individual problems in the neighborhood. In Figure 13 the dependencies on p and the coverage are shown (100x500 **STD**-Problem). The left graph has a curved shape with a maximum around $p = 65$. This is not surprising as the neighborhood is largest for $p = \frac{n}{2} = 50$. The FBA-search however is slower for p-values slightly larger than the $p = 50$. This is due to the fact that the subproblem grows as p increases and thus is more time consuming to solve.

There is a clear dependency on the coverage as well. The solution time increases with the coverage. The explanation here should be found in Figure 6 (page 11) where it is seen that a higher coverage yields larger gaps for the LP-fit. The higher gaps potentially lead to a higher number of neighborhoods to search before reaching a local minimum and the search hence requires more time.

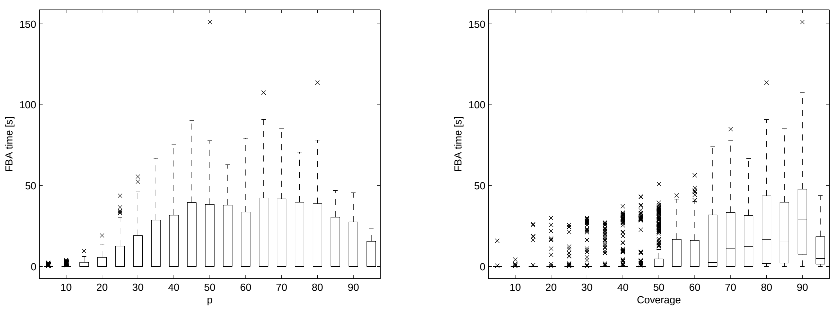


Figure 13: Solution times for the FBA-heuristic for the 100x500 **STD**-Problems.

When considering the quality of the heuristic, it is also relevant to investigate the distribution of the time used on the different segments of the algorithm. The left graph of Figure 14 shows that the major part of the solution time in general is used on the FBA search. On average it is 10 times slower than the LP-relaxation. It is also seen that the LP-relaxation time has many outliers with very high solution time. The picture is even clearer for the larger problems on the graph to the right. Again, in general the LP-solver is fast but in the worst cases even the most time consuming FBA-search uses less time than the LP-solver. The test displayed on Figure 11 and 13 shows that the LP-solver uses most time for small p but the FBA-search uses most time when p is slightly more than $\frac{n}{2}$. Hence slow LP-relaxations will often be linked to fast FBA-searches. Not much time is used to LP-fit compared to the other parts, but further optimization of the data structures can reduce the fit-time even more.

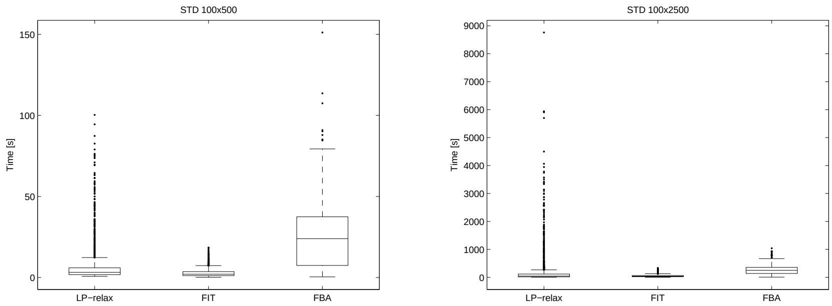


Figure 14: Solution times distributed on LP-relaxation, LP-FIT and FBA-Search. 100x500 **STD**-Problems (left) / 100x2500 **STD**-Problems (right).

7 Final tests

After the exhaustive testing it is interesting to try the heuristic on a number of problems described in the literature. Some of the data sets are derived from real life data. As mentioned earlier there has not been much research on p/q-ACTIVE, but the UFL problems in the OR-library [5] can be given values of p and q and solved as p/q-ACTIVE. Some TSP-problems from the TSP-library [6] have also been tested. Besides the parameters p and q it is also necessary to split the nodes in two: one group representing clients and one representing sites. This is done for $n = 100$ and the nodes chosen as sites (at random) has been saved for each of the problems[3]. All these test problems have been solved by the FBA-heuristic. For the OR-library problems, if the LP-relaxation is not IP-feasible, the optimal solution has been found by standard tools in `CPLEX`.

7.1 OR-library

Three problems have been tested (there are only three large UFL problems in the OR-library). All three problems have $n = 100$, $m = 1000$. The test has been carried out with p taking the values $p = 2, 5, 8, 15, 25, 65$ and q having values giving a coverage of $10\%, 50\%, 80\%, 90\%, 95\%$. Hence $6 \cdot 5 = 30$ problems are tested for each of the three problems in the OR-library. In Table 6 the results from the tests of the OR-library problems are displayed. To save space all problems having IP-feasible solution to the LP-relaxation have been omitted. This leaves only 38 of the 90 problems. From the number of IP-feasible solutions it is clear that the observation about the high quality of the LP-solution is still sound.

The table shows that all of the problems having $p = 2$ are IP-feasible. On the other hand having $p = 5$ gives the largest gaps for the LP-solver for both problem `capa` and `capc`. These findings correspond well to what was concluded from the preliminary tests. For coverage close to 100% less IP-feasible LP-solutions are found, but the FBA-heuristic still finds very good solutions each time. As seen in this test, even when optimality cannot be guaranteed, it is often the optimal solution that has been found. Only 11 of the 90 results of the FBA are not optimal and these have an average gap of 0.19% to the optimal solution.

[3]For further work on these UFL problems the site numbers can be downloaded from the website: http://www.student.dtu.dk/~s011566/

UFL Prob	p	q	Cov.	GAP LP - Fit	GAP LP - FBA	GAP exact - LP	GAP exact - FBA	Time [s] LP	Time [s] Fit	Time [s] FBA
capa	5	20	10.0%	0.32%	0.30%	-0.30%	0	59.97	2.79	1.7
capa	5	100	50.0%	0.32%	0.30%	-0.30%	0	73.53	7.16	2.15
capa	5	160	80.0%	0.31%	0.31%	-0.30%	0.01%	358.39	10.92	2.47
capa	5	180	90.0%	0.36%	0.36%	-0.25%	0.11%	126.31	11	3.2
capa	5	190	95.0%	0.80%	0.53%	-0.19%	0.35%	197.23	11.98	4.41
capa	8	100	80.0%	0.00%	0.00%	0.00%	0	258.64	5.06	3.94
capa	8	112	89.6%	0.01%	0.01%	-0.01%	0	451.19	4.09	4.95
capa	8	118	94.4%	0.06%	0.06%	-0.06%	0	514.52	5.96	5.56
capa	15	60	90.0%	0.01%	0.01%	-0.01%	0	43.14	0.6	19.73
capa	15	63	94.5%	0.05%	0.05%	-0.05%	0	61.42	5.05	21.4
capa	25	36	90.0%	0.02%	0.02%	-0.02%	0	17.49	4.11	51.27
capa	25	38	95.0%	0.05%	0.05%	-0.05%	0	16.37	8.17	97.08
capa	65	14	91.0%	0.00%	0.00%	0.00%	0	4.4	17.57	155.71
capb	8	100	80.0%	1.06%	0.41%	-0.01%	0.40%	208.51	5.91	6.35
capb	8	112	89.6%	1.57%	0.12%	-0.12%	0	119.54	12.62	4.87
capb	8	118	94.4%	1.17%	0.43%	-0.29%	0.14%	691.37	12.92	5.59
capb	15	53	79.5%	0.01%	0.01%	-0.01%	0	36.18	2.12	15.12
capb	15	60	90.0%	0.09%	0.09%	-0.07%	0.02%	74.58	9.09	17.36
capb	15	63	94.5%	0.12%	0.11%	-0.11%	0	78.71	8.77	24.45
capb	25	32	80.0%	0.02%	0.02%	-0.02%	0	22.38	6.2	39.51
capb	25	36	90.0%	0.03%	0.03%	-0.03%	0	34.61	6.84	47.22
capb	25	38	95.0%	0.06%	0.06%	-0.06%	0	37.47	10.07	49.35
capb	65	12	78.0%	0.00%	0.00%	0.00%	0	2.65	8.75	120.91
capb	65	13	84.5%	0.00%	0.00%	0.00%	0	2.68	9.08	132.36
capb	65	14	91.0%	0.00%	0.00%	0.00%	0	4.91	24.96	147.04
capc	5	20	10.0%	0.36%	0.36%	-0.10%	0.25%	142.94	1.97	1.06
capc	5	100	50.0%	0.52%	0.36%	-0.10%	0.25%	155.33	5.07	1.76
capc	5	160	80.0%	0.35%	0.35%	-0.10%	0.26%	122.16	7.33	2.63
capc	5	180	90.0%	0.31%	0.31%	-0.10%	0.20%	172.61	8.35	3.34
capc	5	190	95.0%	0.45%	0.25%	-0.18%	0.06%	259.04	8.81	5.3
capc	8	100	80.0%	0.00%	0.00%	0.00%	0	97.67	0.39	4.13
capc	8	112	89.6%	0.01%	0.01%	-0.01%	0	430.67	0.97	4.97
capc	8	118	94.4%	0.01%	0.01%	-0.01%	0	91.32	0.9	5.81
capc	15	60	90.0%	0.04%	0.04%	-0.04%	0	51.74	1.92	19.21
capc	15	63	94.5%	0.16%	0.16%	-0.16%	0	64.2	3.8	20.61
capc	25	32	80.0%	0.03%	0.03%	-0.03%	0	17.58	6.02	39.97
capc	25	36	90.0%	0.06%	0.06%	-0.06%	0	24.95	10.64	45.53
capc	25	38	95.0%	0.09%	0.07%	-0.07%	0	28.15	12.76	97.69

Table 6: Results for the OR-library tests. $n = 100$, $m = 1000$. All problems having an IP-feasible solution have been omitted.

7.2 TSP-library

Finally, problems from the TSP-library are tested - this time with four different combinations of p and q. Because of the large size of some of the problems, no optimal values have been calculated. The results from the test are given in Table 7. These results are very interesting. The gaps found are significantly larger than expected. One gap is close to 8%, which is more than what was observed in any of the preliminary tests. This gap is found in the solution of problem fl1577. The other setting of p for problem fl1577 still having a coverage close to 100% also yields a rather large gap of approximately 4%. It is notable that not only do the solutions to this problem get better with a low coverage, they are actually both optimal.

Prob	m	p	q	Cov.	GAP LP - Fit	FBA	LP	Time [s] Fit	FBA
dsj1000	900	5	90	50.0%	0	0	77.3	-	-
dsj1000	900	5	171	95.0%	2.61%	1.66%	263.13	3.01	6.12
dsj1000	900	25	18	50.0%	0	0	9.19	-	-
dsj1000	900	25	34	94.4%	2.19%	2.19%	41.25	4.36	46.76
pcb3038	2938	5	293	49.9%	0	0	1861.31	-	-
pcb3038	2938	5	558	95.0%	0	0	3638.85	-	-
pcb3038	2938	25	58	49.4%	1.67%	0.42%	427.91	27.44	550.57
pcb3038	2938	25	111	94.5%	0.44%	0.44%	3565.23	52.89	614.86
gr431	331	5	33	49.8%	0.01%	0.01%	15.95	0.33	0.25
gr431	331	5	62	93.7%	0.26%	0.26%	40.99	0.41	0.41
gr431	331	25	6	45.3%	0.79%	0.65%	6.72	0.78	3.48
gr431	331	25	12	90.6%	0.45%	0.17%	7.54	1.52	10.84
fl1400	1300	5	130	50.0%	0	0	68.84	-	-
fl1400	1300	5	247	95.0%	0.64%	0.64%	436.08	1.76	4.58
fl1400	1300	25	26	50.0%	0	0	26.89	-	-
fl1400	1300	25	49	94.2%	2.44%	1.88%	65.61	9.35	225.98
fl1577	1477	5	147	49.8%	0	0	168.11	-	-
fl1577	1477	5	280	94.8%	3.57%	3.57%	1363.74	6.42	8.09
fl1577	1477	25	29	49.1%	0	0	52.44	-	-
fl1577	1477	25	56	94.8%	10.69%	7.89%	141.26	12.98	308.63
fl3795	3695	5	369	49.9%	0	0	1551.16	-	-
fl3795	3695	5	702	95.0%	0.15%	0.07%	22333.96	55.91	332.63
fl3795	3695	25	73	49.4%	0	0	577.39	-	-
fl3795	3695	25	140	94.7%	3.11%	1.51%	1422.1	44	1531.55
fl417	317	5	31	48.9%	0	0	5.01	-	-
fl417	317	5	60	94.6%	1.89%	1.67%	25.47	0.45	0.62
fl417	317	25	6	47.3%	1.66%	1.66%	2.69	0.76	1.95
fl417	317	25	12	94.6%	6.59%	4.02%	3.65	1.12	6.89
fnl4461	4361	5	436	50.0%	0	0	2845.03	-	-
fnl4461	4361	5	828	94.9%	0	0	13680.38	22.86	68.48
fnl4461	4361	25	87	49.9%	0.52%	0.12%	1541.88	57.07	721.97
fnl4461	4361	25	165	94.6%	0.86%	0.45%	5927.34	100.86	1550.09

Table 7: Results for the TSP-library tests. $n = 100$.

Solving problem fl417 in the test also seems to cause some problems. To analyze this further, visualizations of the two problems have been created to investigate the problem structure. In Figure 15 the two problems mentioned are visualized. It is seen that these problems have a structure which is not found in any of our problem structure definitions described in section 3. They do have all clients and facilities clustered, but these clusters are not like the ones defined earlier. The clusters are very dense and with a very flat rectangular shape.

The important conclusion to be drawn from this test is that even though numerous problems with many different settings and structures have been tested in this work, it is still possible to create specific problems with large gaps.

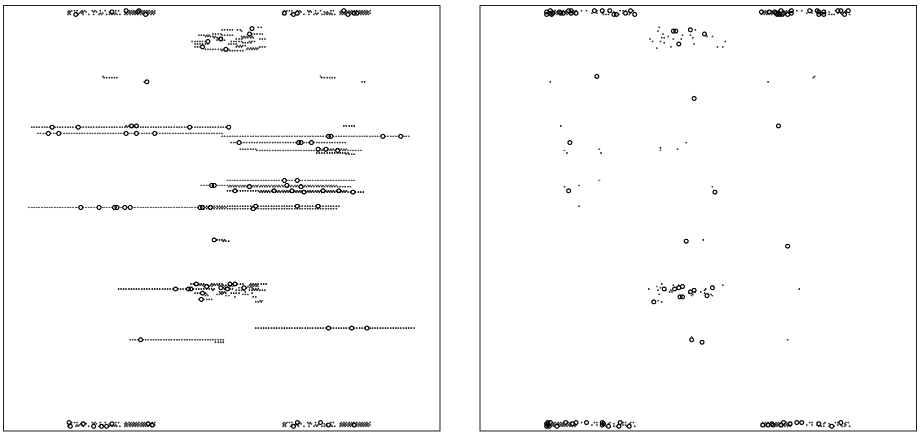


Figure 15: Problem fl1577 (left) and problem fl417 (right) from the TSP-library. The partition of facilities and clients was not specified in the TSP-library.

8 Possible enhancements

During our work with p/q-ACTIVE we have focused on examining the solution qualities when looking at different problem structures and characteristics. Further research could be focused on the following areas.

- The basis of our heuristic is the solution to the LP-relaxed problem. The tests have shown that in a limited number of cases the LP-solver uses a lot more time than in the average case. This can lead to that no solution is found in the time given. To make sure that a solution is always found a mechanism that stops the LP-solver after a certain amount of time can be implemented. When the LP-solver stops, the non-optimal LP-solution can be fitted to an initial feasible solution. In our case we can use the value found by the LP-solver at the time of the break as lower bound, as we are using the dual simplex algorithm. If using primal simplex the non-optimal solution to the relaxed problem could be fitted just as it is done with the optimal solution.

- The neighborhood structure and the chosen decomposition lead to heavy calculations in each neighborhood. Another neighborhood structure can ease the computations and be used as a basis for other heuristics. An example is a reversed neighborhood structure, defined by a fixed number of client-to-facility reallocations. This definition leads to a larger neighborhood, but the calculations in each neighborhood is faster.

- Due to the many calculations needed in the local search heuristic, it can gain in speed if the less promising solutions in the solution space (and thereby in the neighborhood) are excluded. An exclusion can be done with respect to the LP-solutions. If a site is not used in the LP-solution ($y_j = 0$) it is also left out in the primal solution space. This leaves only facilities that in the LP-solution have a fractional y_j or $y_j = 1$. The solution space can be reduced even further if the sites having $y_j = 1$ are fixed as well.

- The idea of using the primal and dual solution in combination can also be used as a guideline to the LP-solver which then resolves the problem a number of times. The ideas of Local Branching [11] and Relaxation Induced Neighborhood Search (RINS) [12] can be applied as the LP-solver usually reveal near-optimal solutions even without any altering.

- When solving the problem, `CPLEX` has been used for solving the decomposed transportation problem. To reduce computation time in this part a dedicated algorithm for the subproblem can be implemented.

- The algorithm used is designed to work well on p/q-ACTIVE in general. If the focus is on a real life problem or problems with a specific structure, a heuristic can be tailored taking advantage of problem dependent features.

9 Conclusion

More than 60% of the problems tested can be solved to optimality just by solving the corresponding LP-relaxation. This is the case for **all** the geometric problem types tested. The number of IP-feasible problems are very dependent on the coverage. Almost all problems with a low coverage are IP-feasible and the number decreases to almost 0 as the coverage increases. The remaining problems have LP-solution values that yield very tight bounds (around 0.05% to optimum).

A heuristic is introduced and tested on a wide range of problems. In general the heuristic is very effective, yielding an average gap for the geometric problems of less than 0.1%. The hardest problems to solve are found to be problems with one or more of the following characteristics: They either have a totally random cost matrix or they have a geometric structure with small and dense clusters of clients and the opening cost of facilities have a relatively small variation. In any case the value of p is small and the coverage is close to 100%.

Finally the heuristic is tested on a number of reference problems from the OR-library and the TSP-library. An important observation here is that it is possible to find problems that are hard to solve, but they must have a unique structure and even in that case most problems will be easy. We have encountered gaps up to 8% in the worst case.

In general the results from the proposed heuristic are very good and the optimal solutions are found in most cases. In more than 60% of the test problems optimality can even be proven. We conclude that the p/q-ACTIVE Uncapacitated Facility Location Problem is easy. Only in rare cases solutions far from optimum are encountered.

Acknowledgments

We thank our fellow student Esben Kolind for his comments on this paper and for his work on a project that led to this paper. Further we thank M.Sc. J. B. Wanscher for his guidance during our work and for supplying test results and source code. Last we thank our two advisors Professor Jens Clausen and Associate Professor Jesper Larsen for their constructive criticism and help throughout the writing process.

References

[1] J.B. Wanscher: *The p/q-ACTIVE Facility Location problem*, M.Sc. Thesis, Department of Computer Science, University of Copenhagen, Copenhagen 2002

[2] A.D. Hansen, D.M. Rousøe, E. Kolind, S.G. Christensen: *Simuleret udglødning og tabusøgning for p/q-active Facility Location Problem*, B.Sc Report, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby 2004

[3] J. Krarup: *The p/q-ACTIVE Uncapacitated Facility Location Problem*, Department of Computer Science, University of Copenhagen, Conference abstract,
http://www.diku.dk/forskning/algorithms/seminarabstracts/...
seminarabstracts00/jakob.html, 18. Feb 2000

[4] E. L. F. Senne, L. A. N. Lorena: *Lagrangean/Surrogate Heuristics for p-Median Problems*, In Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research, M. Laguna and J.L. Gonzalez-Velarde (eds.) Kluwer Academic Publishers, pp. 115-130, 2000

[5] J. E. Beasley: *OR-Library*,
http://people.brunel.ac.uk/ mastjjb/jeb/info.html, 1. Nov 2004

[6] G. Reinelt: *TSPLIB*,
http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, 16. Dec 2004

[7] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998

[8] G. Cornuéjols, G. L. Nemhauser, L. A. Wolsey, *The uncapacitated facility location problem*, In P. Mirchandani and R. Francis, editors, Discrete Location Theory, pages 119-171. Wiley, New York, NY, 1990.

[9] J. Brimberg, P. Hansen, N. Mladenović, E. D. Taillard: *Improvements and comparison of heuristics for solving the Uncapacitated Multisource Weber Problem*, Operations Research, 48 (3) pp. 444-460, 2000.

[10] R. F. Love, J. G. Morris, G. O. Wesolowsky: *Facilities Location, Models & Methods*, North-Holland, 1988

[11] A. Lodi: *Local branching: a tutorial*, MIC2003: The Fifth Metaheuristics International Conference, T4-1 - T4-8, 2003

[12] E. Danna, E. Rothberg, C. L. Pape: *Exploring Relaxation Induced Neighborhoods to improve MIP solutions*, Technical report, ILOG, 2003