A capacitated clustering heuristic for large datasets

Jacoba H. Bührmann and Ian Campbell
School of Mechanical, Industrial and Aeronautical Engineering
University of the Witwatersrand
Johannesburg, 2000, RSA
Joke.Buhrmann@wits.ac.za

M. Montaz Ali
School of Computational and Applied Mathematics (CAM),
University of the Witwatersrand
Johannesburg, 2000, RSA
Montaz.Ali@wits.ac.za

Abstract

A regret order based clustering heuristic is proposed to create capacitated clusters for spatial datasets with equal weighting. This heuristic can be used in a vast number of applications, including the Capacitated P-Median problem (CPMP) and Capacitated Facility Location Problem (CFLP). The heuristic is scalable and therefore also useful to cluster large instances. The heuristic builds on an initial uncapacitated clustering solutions the can be generated by various clustering methods and has proven to provide good quality end solutions independent of the quality of the initial solution. Different types of regret proximities are tested for the Capacitated P-Median problem (CPMP) in particular. The results along with visual plots on a map of geometrical data are shown to illustrate the impact of the different proximity rules.

Keywords
Clustering, Capacitated clustering, CPMP, CCP, p-median, hierarchical clustering, k-means, k-median, regret order based heuristic, iterative partitioning, graph-based clustering, nearest neighbor clustering, density-based clustering, concentrator clustering

1. Introduction

Most applications do not only need data points to be clustered but also adhere to capacitated constraints. An area where capacitated clusters are needed is the design of supply chain network where spatial data need to be clustered based on supply and capacity limitations, for example the vehicle routing (VRP), capacitated facility location (CFLP) and capacitated location-routing problems (CLRP). The capacitated p-median problem forms a central part of more complicated supply chain network problems like the VRP, CFLP and CLRP. It was therefore chosen to measure the effectiveness of the proposed regret-order based method.

Numerous methods have been suggested in the past but few were tested and found to work well for big instances. Approximate algorithms (Harks et al., 2013) that take scalability into consideration has only become the focus in recent years. Many of the suggested approaches from literature do not have a time complexity that can deal with large instances.

The goal of clustering methods is to group items together to minimize the total dissimilarity between the points in a cluster (Everitt et al., 2011, Hartigan, 1975). There are clustering methods that can effectively deal with large datasets, for example the hierarchical, iterative partitioning methods like the k-means, nearest neighbor and density-based methods. Even though a clustering method can be fine-tuned to create the correct amount of clusters, the capacity per cluster are not guaranteed. A technique is needed to create capacitated clusters from the unconstrained clustering solutions for single-source problems.

2. The capacitated p–median problem (CPMP)

The capacitated clustering problem (CCP) is a clustering problem with n spatial data points and a set of m candidate centers to select from to create exactly p capacitated clusters, Mulvey and Beck (1984). In the case of single-source constraints, each point must be assigned to a single cluster center (sometime referred to as a facility). The problem is considered NP-hard, Mulvey and Beck (1984), Min (1996).

The capacitated p–median problem (CPMP) is a specific case of the CCP where the capacity constraints for all the clusters are homogeneous and the coefficients of the objective function are distances, (Negreiros and Palhano, 2006). The CPMP is referred to as a concentrator problem because all points are considered potential facilities, Ceselli et al., (2009). The objective is to allocate p medians (also known as facilities) and minimize the sum of the within-clustered Euclidean distances between the chosen p facilities to the points assigned to them, (Drezner and Hamacher, 2001).

The objective function of the model is then:

$$\min z = \sum_{i \in V} \sum_{j \in V} d_{ij}\, x_{ij}, \qquad (1)$$

where x_{ij} is a binary decision variable with $x_{ij} = 1$ if point i is allocated to cluster j, (Han and Kamber, 2006). Each point i consists of two decimal coordinates (longitude and latitude) in $\Re^2$ Euclidean space using Euclidean distances.

3. Capacitated clustering methods used in the past

Mladenović et al. (2007) and Drezner and Hamacher (2001) provide a list of heuristic approaches used to solve the p–median and fixed charge FLP problems. Mladenović et al. (2007) also discuss using aggregation, where points are clustered together based on a commonality, as a method to make the problem smaller. In-line with this, clustering methods that can cluster large datasets with small time complexity have also become an area of interest. The time complexity for hierarchical clustering methods, for example, is O(n^2log(n)) when using the Lance-Williams formula to calculate inter-cluster distances, Everitt et al. (2011).

Min (1996), Barreto et al. (2007) and Lam et al. (2009) have investigated the use of hierarchical capacitated clustering methods in the past. Hierarchical methods need a stopping rule to stop clustering when an acceptable clustering solution has been reached. Min (1987) explore the use of capacity as a stopping rule to ensure that clusters do not exceed capacity limits, creating more clusters than are needed that are not filled to capacity. Barreto et al. (2007) suggest excluding a cluster if it becomes fully capacitated, and continue with the merging of clusters that still have capacity available. This resulted in far-off clusters being merged. This causes a "jumping" effect (Bührmann, 2016), where an under capacitated cluster overlaps a full cluster, which gives suboptimal solutions. Lam et al. (2009) suggests using the ratio of inter-cluster distance variation and between-cluster distance variation as a stopping rule. This is called the pseudo F–statistic, but do not guarantee that clusters meet capacity. Results showed that, of all the tested hierarchical clustering methods, Ward's method returned the minimum objective function value for Eq. (1) when using Euclidean distances. It is therefore considered the most effective hierarchical clustering method (Min, 1996, Barreto et al., 2007, Lam et al., 2009 and Bührmann, 2016).

A concern when using stopping rules for hierarchical clustering, is that it cannot guarantee balanced equal numbered clusters as is needed in the case of homogenous capacity constraints in the CPMP. Zhou et.al. (2002) mention that the balanced allocation problem has received little attention in the past. They suggest the use of a star-spanning tree formulation in order to solve the problem using a genetic algorithm. Unlike the hierarchical clustering methods, there are other clustering methods that do not make use of stopping rules. These methods create clusters based on the number of clusters specified beforehand or other input parameters like the number of nearest neighbors to consider.

Iterative partitioning methods are quite popular and has a time complexity of O(n) per iteration (Negreiros and Palhano, 2006). The methods make use of a k parameter to determine the amount of clusters. Although these methods have proven to provide more balanced clusters they also cannot guarantee adherence to capacity constraints. Geetha et al., (2009) describes the k-means method, also called the centroids of the clusters. Alternatively, the k-median method calculates the median each dimension of the points as the cluster centers, Han and Kamber (2006).

Nearest neighbors clustering, described by Jain and Dubes (1998), and density-based clustering make use of a density factor or the number of nearest neighbors to consider as input parameters. The best input parameters to consider can often only be determined using trial-and-error to create the correct amount of clusters.

A technique is therefore needed where an unconstrained clustering solution using any clustering method can be modified to a constrained solution. This method needs to adhere to capacity constraints that can be used in conjunction with all clustering methods. A regret order based approach is suggested to create capacitated clusters. Other regret order based functions have been proposed by various authors in the past, including Mulvey and Beck (1984), Negreiros and Palhano (2006) and Barreto et al. (2007). However, these authors do not make use of an iterative regret order based function. They also do not consider the visual attractiveness of the solution to ensure clusters remain points in close proximity of each other. An improvement phase is introduced to this effect. The iterative nature ensures that points stay close to their assigned clusters while preventing the creation of overlapping clusters.

4. The two-phase PROBUC heuristic

A two-phase proportional regret order based unconstrained to constrained (PROBUC) heuristic is introduced to create capacitated clusters. This method is used to modify an unconstrained clustering solution to a constrained version and can be applied to any clustering solution.

Figure 1. illustrates the algorithmic flowchart for the two-phase PROBUC method. The two phases are a re-assignment and an improvement phase, and PROBUC alternates between them until no further improvements are found. In the re-assignment phase, points are re-assigned to other clusters using a regret order based function until a feasible solution is found, i.e. one that has no overcapacitated clusters. Then PROBUC moves to the improvement phase. PROBUC forms mutually exclusive clusters that do not have overlapping boundaries.

4.1. The re-assignment phase

The procedure for the re-assignment phase is illustrated in Figure 2. It consists of multiple iterations consisting of improvements made using “neighborhood moves”. A “neighborhood move” is the process where a point in one cluster is re-assigned to another cluster, normally a neighboring cluster. Iterations continue until there are no more overcapacitated clusters. In each iteration, there are three steps, as follows:

Step 1: Identify clusters which are overcapacity

We define three types of clusters, where “capacity” means the cluster capacity, or the number of points assigned to the cluster. The points can also have demands associated to them. In this case the clusters have to adhere to the sum of the demands associated to the points assigned to each cluster. The clusters are defined in terms of percentage capacity, where 100 % means the cluster is full (“at capacity”). In practice, parameter P is used to indicate when a cluster is approaching capacity, and is varied between iterations as required.

1) Under capacitated: capacity < (100 - P)% .
2) Near-to-capacity: (100-P) % ≤ capacity ≤ 100 %.
3) Overcapacitated: capacity > 100 %.

If there are no overcapacitated clusters, the solution is feasible and the re-assignment procedure exits. If there are overcapacitated clusters, the procedure identifies a list of clusters which are either overcapacitated or near-to-capacity. These clusters are clusters from which points can be moved from when considering neighborhood moves in Step 2. The reason near-to-capacity clusters are considered, as well as overcapacitated clusters, is that the contents of these near-to-capacity clusters can be reduced to make way for points from nearby, overcapacitated clusters.

The parameter P can and should be varied between iterations, and when returning to the re-assignment phase after an improvement phase, as the number of overcapacitated clusters reduces. For example, in this work, P = 0 is used for the first re-assignment phase, first iteration. This means that only overcapacitated clusters will be considered for moves that remove points in step 2. Thereafter, P = 10 is used for the first re-assignment phase, second iteration. For subsequent iterations, the value of P is increased by 10 for each iteration. However, a maximum value of P = 90 is used as a limit to prevent clusters becoming empty. In the second and subsequent re-assignment phases, use P = 10 initially, and thereafter P = 0. A Tabu List is created to prevent the reversal of moves in subsequent iterations.

Step 2: Create a list of possible moves for each overcapacitated cluster

Cluster centers are calculated as the centroid of all points assigned to the cluster. Then a list of possible moves is created for each cluster, where a move is a point that can be re-assigned to another cluster.

The desirability of each move is calculated differently for a homogeneous demand case and a case where the point demands are different. For the homogeneous demand case, the impact of each move is calculated based on a regret function with a proximity measure. Different proximity measures used in the regret function cause different types of groupings, Bührmann (2016). The outcome of the different proximity measures is discussed in the computational study (Section 5).

Step 3: Implementing the best moves

The procedure creates an "ordered list of moves", which places the most desirable moves first, and the least desirable moves last. In the homogeneous demand case, this list is created by ordering moves according to the regret values for all clusters from smallest to largest. The smallest regret value move is considered the most desirable. In the varying demand case, the list of moves are ordered according the demand ratio values of the moves from largest to smallest, and the move with the largest demand ratio is considered most attractive.

The procedure then checks:

- whether the move is a reversal of a previously implemented move (i.e. on the Tabu List).
- whether the move can be feasibly implemented, that is, the new cluster has sufficient capacity.

If the move is not Tabu and can be feasibly implemented, the move will be implemented, and the point is re-assigned. The move is added to the Tabu List, such that any move that would re-assign the point to the original cluster will be prevented. Otherwise the procedure will move to the next point move on the ordered list of moves.

This re-assignment procedure will continue implementing moves until either all clusters are no longer overcapacity, or until there are no more moves on the ordered list of moves. If all clusters are no longer overcapacity, the procedure has found a feasible solution and exits the reassignment phase. Otherwise it returns to Step 1. However, now it will also consider moves involving moving points from clusters which are P = 10% below the cluster capacity limit.

Note that if the procedure goes through the ordered list of moves and cannot find a non-Tabu, feasible move to implement, it will return to the beginning of the list and implement all the Tabu, infeasible moves anyway. This enhances the possibility of finding a feasible solution in future iterations.

4.2. The improvement phase

The improvement phase is used to improve the quality of clusters, and to prevent clusters from overlapping. Similar to the k–near clustering method introduced by Jain and Dubes (1998), a point is assigned to the same cluster as its k nearest neighbors. In the case where the k nearest neighbors belong to different clusters, the point is assigned to the cluster to which most of the k nearest neighbors are assigned. The procedure goes through every point in the dataset and applies the following two steps:

Step 1: Identify k nearest neighbors

The value of k is calculated as follows:

$$k = \max\left\{b, \left\lceil \frac{n}{p} \times \sigma_a \right\rceil\right\}, \qquad (2)$$

where n is the total number of points, p is the number of clusters, $\frac{n}{p}$ is the average number of points per cluster, b and σ_a are parameter values that can be varied per instance. The k closest neighbors (points) are then identified and put in the nearest neighbors set.

Step 2: Remove neighbors based on the ratio of edge lengths

A fraction of the k nearest neighbors are removed from the nearest neighbor set based on the ratio of edge lengths criteria, Bührmann (2016):

$$\frac{l_{ij}}{\overline{l_i}} \geq l_T \text{ and } \frac{l_{ij}}{\overline{l_j}} \geq l_T, \quad \text{or} \qquad (3)$$

$$\frac{l_{ij} - \overline{l_i}}{\sigma_i} \geq \sigma_T \text{ and } \frac{l_{ij} - \overline{l_j}}{\sigma_j} \geq \sigma_T. \qquad (4)$$

Here l_{ij} is the length of the edge between points i and j. $\overline{l_i}$ and $\overline{l_j}$ are the averages of the length of the edges around points i and j respectively for a sub-tree of depth D. σ_i and σ_j are the sample standard deviations of the lengths of all the edges at points i and j for a sub-tree of depth D. σ_T and l_T are predefined cut-off values.

Any point which satisfy Eq. (3) or (4) are excluded from the set of nearest neighbors. A distance cut-off is used to ensure that no neighbors further than the cut-off distance is considered. The point is then assigned to the same cluster as all or most of the points in the nearest neighbor set. The improvement phase can cause a solution to become infeasible, that is, to cause clusters to become overcapacitated. Therefore, to restore feasibility, the re-assignment phase is repeated after every execution of the improvement phase.

Improvement and re-assignment phases are repeated until no further improvement in solution quality is obtained.

5. Computational Study

The experiments were carried out on an Intel(R) Core(TM) i7-3610QM CPU, 2.30GHz computer with 8GB RAM and using a Windows 7 operating system.

5.1 Instances

The two instances used in this study are described as follows:

D7681 – an instance with 7 681 spatial data points from a food distribution company in South Africa. The instance was used to test the use of different proximity measures to calculate the regret function in the two-phase PROBUC heuristic. The data points consist of two dimensional decimal Euclidean coordinates. The instance was clustered into ten clusters with a homogenous capacity constraint of 1 000 per cluster. The single and complete linkage hierarchical clustering methods were used to create starting solutions.

USA13509 – an instance from the TSP library given by Reinelt (2015) containing 13 509 points from the USA. The instance was clustered into 50 clusters. Ward's method, an effective hierarchical clustering method, was used to create starting unconstrained clustering solutions. This was compared to using the k-means and k-median methods, well-known iterative partitioning methods, to create unconstrained starting solutions. For the k-means and k-median methods batch updating was used – i.e. cluster centers were only updated once per iteration after all the points have been assigned to their new cluster, as explained by Han and Kamber (2006). Only ten iterations were used for the iterative partitioning methods. A homogeneous capacity constraint of 400 per cluster was then applied and using the different proximity measures for the regret function were compared to create capacitated clusters.

5.2 Parameter Estimation

The numerical experiments were done using the instances to find suitable parameters to use for the PROBUC heuristic. For the re-assignment phase, the Tabu list length was kept short and set to one. For the improvement phase, see Eq. (2), (3, (4), the following values were found to produce good results and were therefore used (Bührmann, 2016):
b = 2, σa = 0.01, lT = 0.5, σT = 0.1, D = 25 and all neighbors further than a distance cut-off of 10 km were ignored.

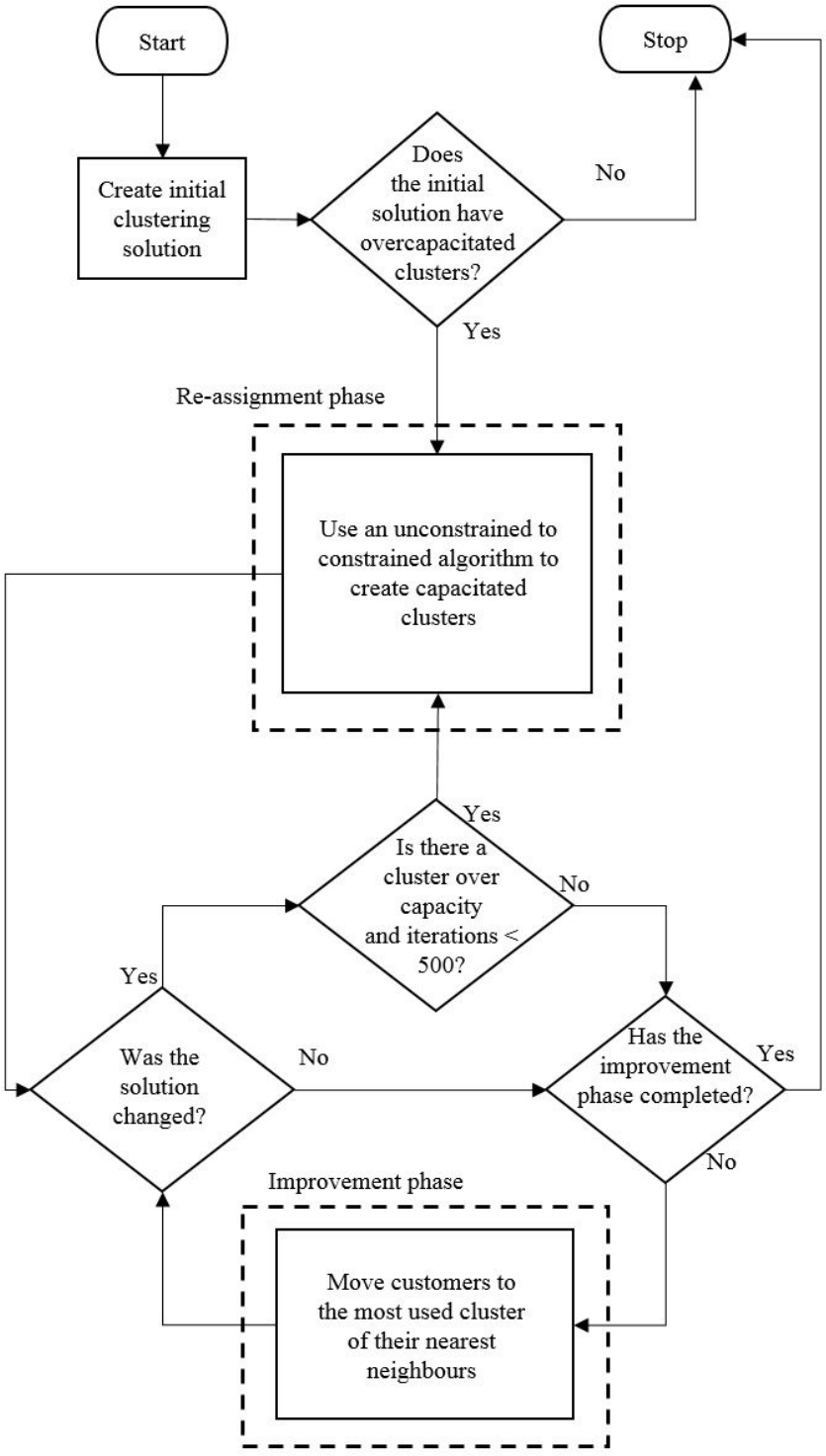


Figure 1. Work flow diagram of the two-phase PROBUC method

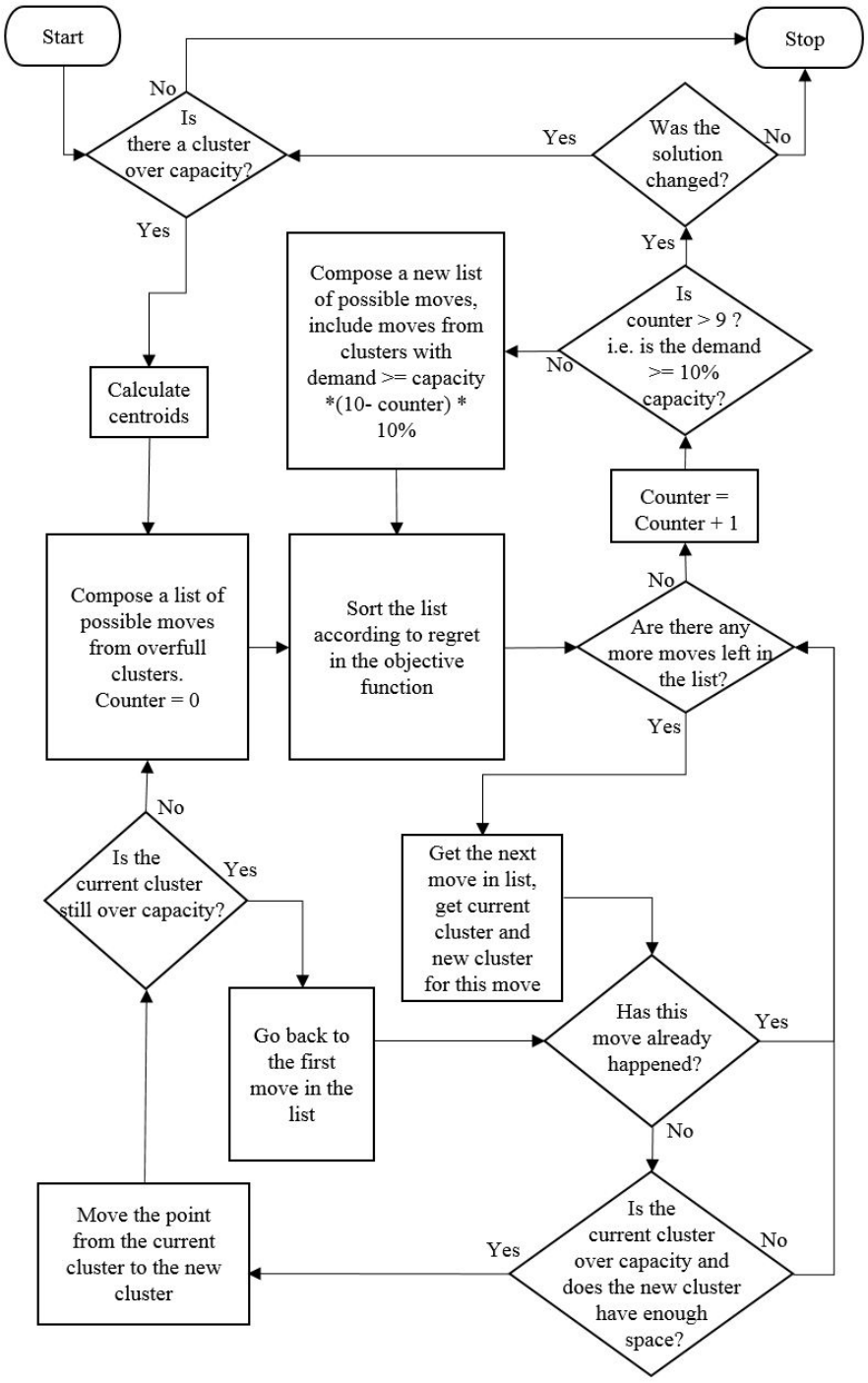


Figure 2. The re-assignment phase of the suggested two-phased PROBUC method

6. Results

It was observed that different proximity measures used in the regret function caused different types of groupings. Seven variations of the proximity measures for the regret function were identified based on previous literature. These variants were tested using both instances D7681 and USA13509 to determine the best one to use in the re-assignment phase.

It is assumed that a homogeneous unit demand is used for all the points, so that the impact of the different measures under equivalent circumstances can be tested. Since only movements from the point's current assignment and the next closest cluster are considered, the following definitions can be made:

- The current proximity, d_{cur}, refers to the distance between the point and the means of the currently allocated cluster center.
- The second best proximity, d_{2nd}, is the distance between the point and the means of the closest cluster center, excluding the current assignment.
- The total proximity, d_{tot}, is the sum of the distances from the point to all cluster centers, including the current assignment.

These proximity variables differ per point and can be used to calculate a regret value r(i) associated with moving point i to the second closest cluster center. It should also be noted that the current assignment of a point is not necessarily the closest cluster center. This is because in the different clustering methods, points can be assigned to facilities other than the closest depending on the clustering algorithm used. As such, it also means that the regret values can be negative.

The formulations of the different proximity measures are given in Eq. (5) – (11).

$$r1(i) = d_{2nd} \quad (5)$$
$$r2(i) = d_{2nd} - d_{cur} \quad (6)$$
$$r3(i) = d_{tot} - d_{2nd} \quad (7)$$
$$r4(i) = d_{2nd} / d_{cur} \quad (8)$$
$$r5(i) = d_{2nd} / d_{tot} \quad (9)$$
$$r6(i) = (d_{2nd} - d_{cur}) / d_{cur} \quad (10)$$
$$r7(i) = (d_{2nd} - d_{cur}) / d_{tot} \quad (11)$$

The results for the two instances are now discussed in further detail:

6.1 Instance D7681

The different proximity measures were tested using two initial solutions created by the single linkage and complete linkage respectively. These two were chosen because of the relative ease to compute the initial solutions and because the single linkage is known to provide notoriously bad unbalanced solutions while the complete linkage should return visually better but only average CPMP results compared to other clustering methods. The results using the CPMP objective function given in Eq. (1) and CPU times can be seen in Table 1. The CPU times of the unconstraint clustering using the single linkage and completer linkage methods were 48 and 46 seconds respectively.

Solutions were also compared for visual attractiveness, a concept defined by Poot et al. (1999). With the advances in visualization of spatial data came the need for a solution to also look acceptable to the client especially in the supply chain environment, (Kant et al., 2008). Typical acceptability rules for facility locations include that there should be clearly defined point to cluster (facility) assignment regions with no overlap. The point to cluster assignments for the different proximity measures are illustrated in Figures 3.a) – 3.h) and 4.a) – 4.h). The maps were illustrated using a free Google API for GoogleMaps as described by Bührmann (2016). Each cluster were illustrated using a different color and symbol. Circles were drawn to highlight obvious problems with regards to the clusters that were visually detected.

The unconstrained solutions for the single linkage and complete linkages are shown in Figures 3.a) and 4.a) respectively. The distance to the second best cluster being considered for assignment, d_{2nd} is used quite frequently in the literature, Eq. (5). This method of re-assignment tends to create concentric circular clusters around the cluster centers and allows points to easily fall between the circular clusters when capacity is reached. These points will then

remain assigned to the original unconstrained clusters because there was not enough capacity to re-assign them as well. This measure does not tend to create tightly clustered assignments, as illustrated by the red circles in Figures 3.b) and 4.b).

Mulvey and Beck (1984) suggested using the difference between the current proximity and the second best proximity, Eq. (6). In this case, the regret values will be negative where points were not assigned to the closest cluster centroids and these will be ordered above the positive values. Solutions using this proximity measure are illustrated in Figures 3.c) and 4.c). Although this method gives tighter clusters, there are still points that can fall between cluster assignments and are assigned to a third cluster.

One can also use the difference between the total proximity and the second proximity, Eq. (7), as illustrated in Figures 3.d) and 4.d). This method also left points furthest from the cluster centers to remain assigned to the original clusters. The values in Table 1. does not compare well against the other proximity measures either.

Instead of using the difference in distance, another type of calculation is to use proportional regret, see Eq. (8) and Eq. (9). An example is the ratio between the second best proximity and the current proximity, as suggested by Barreto et al. (2007). The solutions of using a proportional ratio as regret calculation are shown in Figures 3.e) – 3.f) for the single linkage and Figures 4.e) – 4.f) for the complete linkage methods. Figures 3.g) – 3.h) and Figures 4.g) – 4.h) illustrate combining both the difference and proportional ratios in the regret calculations, Eq. (10) and Eq. (11).

Different proximity measures were effective for the single - and complete linkage starting methods and the results were inconclusive. For single linkage, Eq. (6) and Eq. (9) returned the best objective function value and for complete linkage it was Eq. (6) and Eq. (11). When comparing the visual attractiveness of the single linkage solutions, measures using Eq. (10) and Eq. (11) - Figures 3.g) and 3.h) returned solutions with the least cluster overlap that could easily be solved in the improvement phase. For the complete linkage no obvious problems could be found when using the measure for Eq. (11) - Figure 4.h).

6.2 Instance USA13509

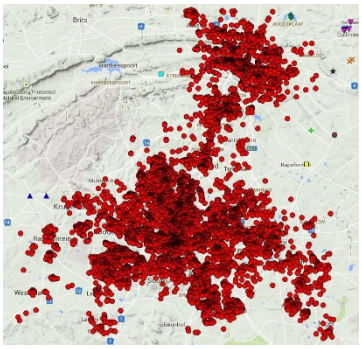
From the tests conducted on instance USA13509, the results of the objective function - Eq. (1) - can be seen in Table 2. Different proximity measures resulted in the best solution for each different unconstrained starting solution. The results are therefore inconclusive.

Three unconstrained starting solutions were created using Ward's method, k-means and k-median methods. These methods took 161, 65 and 65 seconds respectively. The CPU times for the two-phased PROBUC method were quite similar. In all cases the refinement phase took up the longest time at 189, 186 and 188 seconds respectively, while the re-assignment phase took only two seconds and executed before and after the refinement phase.

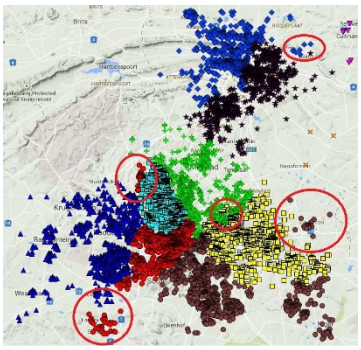
From the table it is clear that the starting solution do have an impact on the objective function of the capacitated clustering results. It is therefore better to start off with a good unconstrained clustering solution. A second benefit from using a clustering method that creates balanced clusters, is that the CPU time to create a capacitated solution is less because there are less points to re-assign. The two-phased PROBUC heuristic compose a sorted list of possible moves to make. If only a few points need to be re-assigned, the number of items in the list will be reduced.

The best objective function were however found using the k-means iterative partitioning method and the proximity measure calculation from both Eq. (6) and Eq. (11). This solution also gives a good visual attractive solution, illustrated in Figure 5. Similar to Figures 3. and 4, the map in Figure 5. was also produced using a free GoogleAPI on Google Maps.

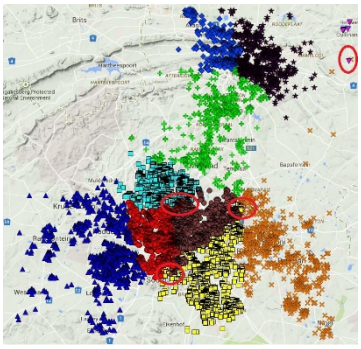
From the different proximity measures used in the regret function of the PROBUC heuristic, Eq. (6) and Eq. (11) produced consistent good objective function results with good visual attractive solutions. It does however not guarantee the best objective function. In all cases Eq. (7) returned the worst results, Eq. (8) and Eq. (10) returned the same results.



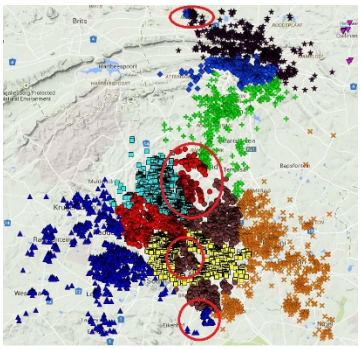
3.a) Single linkage initial solution



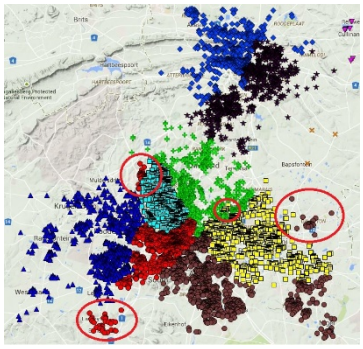
3.b) Proximity measure r1(i)



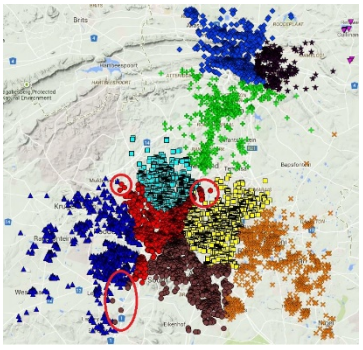
3.c) Proximity measure r2(i)



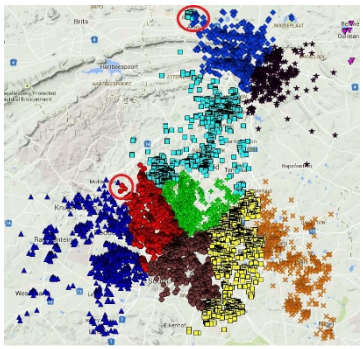
3.d) Proximity measure r3(i)



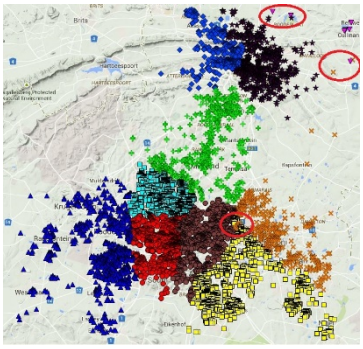
3.e) Proximity measure r4(i)



3.f) Proximity measure r5(i)

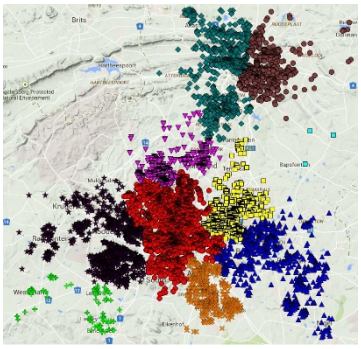


3.g) Proximity measure r6(i)

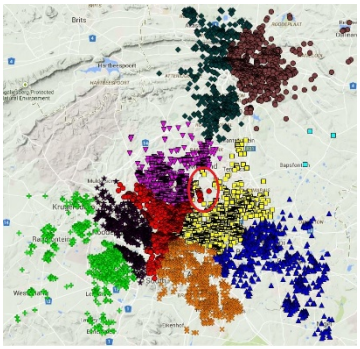


3.h) Proximity measure r7(i)

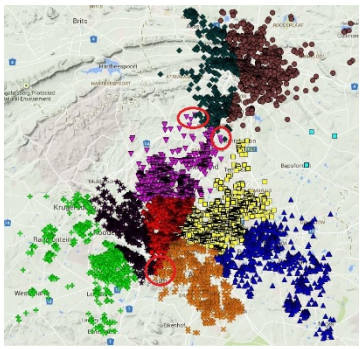
Figure 3. Different regret functions for the single linkage method using D7681



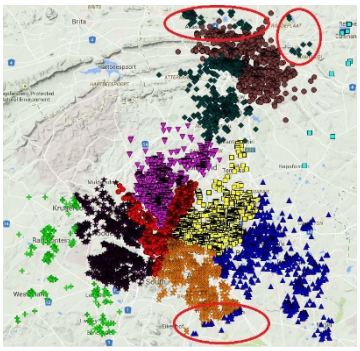
4.a) Complete linkage initial solution



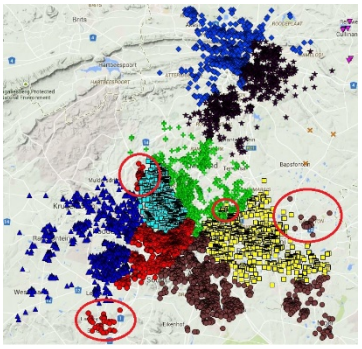
4.b) Proximity measure r1(i)



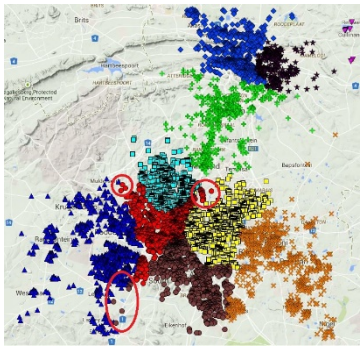
4.c) Proximity measure r2(i)



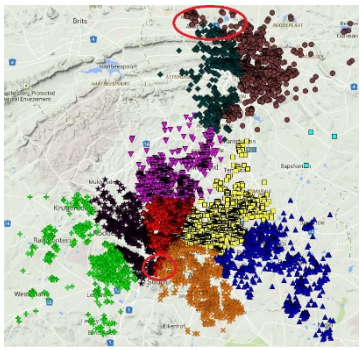
4.d) Proximity measure r3(i)



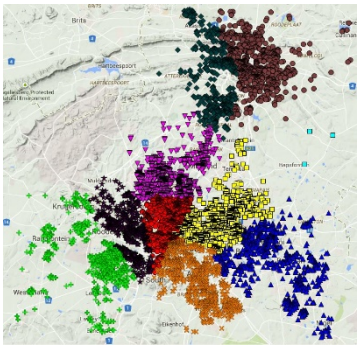
4.e) Proximity measure r4(i)



4.f) Proximity measure r5(i)



4.g) Proximity measure r6(i)



4.h) Proximity measure r7(i)

Figure 4. Different regret functions for the complete linkage method using D7681

Table 1. Different proximity measures in the two-phase PROBUC heuristic using D7681

	Objective function – Eq. (1)		**CPU time (seconds)**	
	Single linkage	**Complete linkage**	**Single linkage**	**Complete Linkage**
Original unconstrained solution	166 720.73	55 308.55		
r1(i) = d_{2nd}	59 424.95	53 520.17	53	46
r2(i) = $d_{2nd} - d_{cur}$	53 112.95	52 979.63	56	25
r3(i) = $d_{tot} - d_{2nd}$	60 305.91	55 581.12	47	45
r4(i) = d_{2nd}/d_{cur}	55 443.70	53 276.40	49	25
r5(i) = d_{2nd}/d_{tot}	52 678.53*	53 473.31	51	47
r6(i) = $(d_{2nd} - d_{cur})/d_{cur}$	55 443.70	53 276.40	48	26
r7(i) = $(d_{2nd} - d_{cur}) /d_{tot}$	55 803.87	52 958.50*	54	25

*** Best solution found associated with the specific unconstrained clustering solution**

Table 2. Different proximity measures in the two-phase PROBUC heuristic using USA13509

	Objective function – Eq. (1)		
	Ward's method	**k-means iterative partitioning**	**k-median iterative partitioning**
Original unconstrained solution	17 502.56	16 227.08	16 533.02
r1(i) = d_{2nd}	18 017.92	16 516.57	16 776.73
r2(i) = $d_{2nd} - d_{cur}$	17 980.76	16 486.70*	16 726.52
r3(i) = $d_{tot} - d_{2nd}$	18 311.33	16 624.76	16 825.34
r4(i) = d_{2nd}/d_{cur}	17 973.29*	16 487.95	16 752.26
r5(i) = d_{2nd}/d_{tot}	18 030.03	16 513.70	16 777.03
r6(i) = $(d_{2nd} - d_{cur})/d_{cur}$	17 973.29*	16 487.95	16 752.26
r7(i) = $(d_{2nd} - d_{cur}) /d_{tot}$	17 979.94	16 486.70*	16 725.86*

*** Best solution found associated with the specific unconstrained clustering solution**

7. Conclusion

When creating capacitated clusters for the single source CPMP with homogenous capacity constraints, traditional methods often do not take time complexity into consideration and are too slow to solve instances with a large number of data points. Cluster-based approaches are more practical as it is able to create clusters with reasonable time complexity. These methods can however not guarantee that capacity constraints are adhered to.

The two-phase PROBUC method is an effective heuristic to create capacitated clusters for converting uncapacitated clusters to capacitated clusters. However, certain uncapacitated clustering methods tend to create unbalanced solutions where many points need to be re-assigned to create capacitated clusters, for example, the single linkage hierarchical clustering method (Bührmann, 2016). In these cases, points are reassigned to neighboring clusters, and these neighboring clusters quickly reach capacity. In other work, points are assigned to another cluster with enough capacity where the cluster center is not necessarily the closest or second closest one to the point. This causes a "jumping" effect, where an under capacitated cluster overlaps a full cluster (Bührmann, 2016). In the PROBUC method, this effect is avoided by not allowing points in overcapacity clusters to move to any other cluster than the closest. Instead, PROBUC allows the re-assignment of points in clusters that are within a percentage (typically P = 10%) of the capacity limit to their next closest cluster center. This makes space for transferal of points from overcapacitated clusters. The algorithm is also designed to prevent clusters becoming empty, by not allowing points to move out of clusters with only 10% of the capacity limit remaining.

Results from the two instances indicate that solutions producing the best objective function do not necessary result in the best visual attractive solution. It is therefore important to also consider visual attractiveness when studying the CPMP and other related network problems as suggested by Poot et al. (1999) and Kant et al., (2008).

Starting with balanced unconstrained clustering solutions do provide an advantage when using the PROBUC heuristic. Most of the time spend in the PROBUC heuristic method is spent in the improvement phase. If the solutions produced by the unconstraint starting solution is of good quality, the number of points that need to be re-assigned are so few that the need of the improvement phase becomes less significant. Overall the method was able to effectively create capacitated constraint solutions in a very short CPU time of less than 200 seconds for 13509 data points.

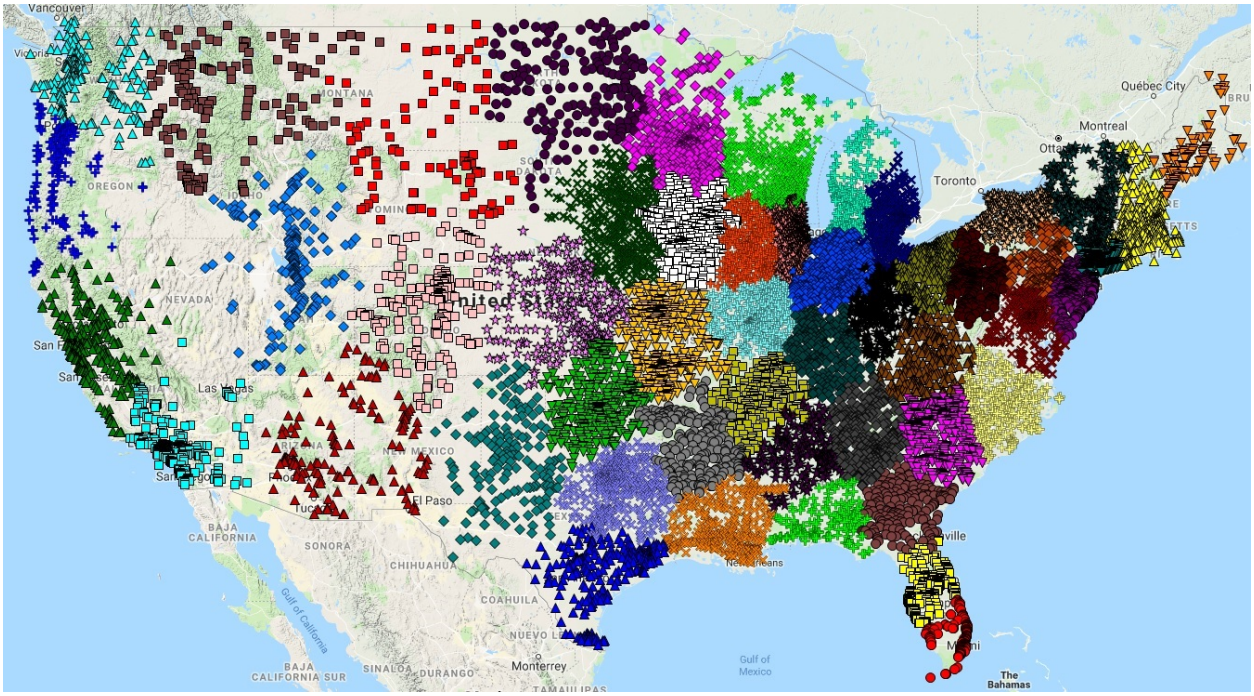


Figure 5. Best capacitated clustering solution for instance USA13509

References

Barreto, S., Ferreira, C., Paixão, J., and Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3):968–977.

Bührmann, J. H., *The effects of clustering on the medium and large-scale Capacitated Location-Routing problem*, Phd thesis, The University of Witwatersrand, South Africa (2016).

Ceselli, A., Liberatore, F., and Righini, G. (2009). A computational evaluation of a general branch-and-price framework for the capacitated network location problems. *Annals of Operations Research*, 167:209–251.

Drezner, Z. and Hamacher, H., editors (2001). *Facility Location: Applications and Theory*. Springer-Verlag, Heidelberg, Germany, 1st edition.

Everitt, B., Landau, S. L., Leese, M., and Stahl, D. (2011). *Cluster analysis*. John Wiley & Sons, Ltd, West Sussex, United Kingdom, 5th edition.

Geetha, S., Poonthalir, G., and Vanathi, P. T. (2009). Improved K–Means Algorithm for Capacitated Clustering Problem. *INFOCOMP Journal of Computer Science*, 8(4):52–59.

Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kauffman Publishers, Elsevier Inc., 2nd edition.

Harks, T., König, F. G., and Matuschke, J. (2013). Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22.

Hartigan, J. (1975). *Cluster algorithms*. John Wiley & Sons, Ltd, New York, USA.

Jain, A. K. and Dubes, R. C. (1998). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey.

Kant, G., Jacks, M., and Aantjes, C. (2008). Coca-cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces*, 38(1):40–50.

Lam, M., Mittenthal, J., and Gray, B. (2009). The impact of stopping rules on hierarchical capacitated clustering in location routing problems. *Academy of Information and Management Sciences*, 12(1):13–29.

Lance, G. N. and Williams, W. T. (1967). A general theory of classificatory sorting strategies: 1. Hierarchical systems. *Computer Journal*, 9:373–380.

Min, H. (1987). *The Vehicle Routing Problem with Product/Spatial Consolidation and Backhauling*. Phd thesis, Ohio State University.

Min, H. (1996). Consolidation terminal location–allocation and consolidated routing problems. *Journal of Business Logistics*, 17(2):235–263.

Mladenović, N., Brimberg, J., Hansen, P., and Moreno-Pérez, J. (2007). The p–median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179:927–939.

Mulvey, J. and Beck, M. (1984). Solving capacitated clustering problems. *European Journal of Operational Research*, 18:339–348.

Negreiros, M. and Palhano, A. (2006). The capacitated centred clustering problem. *Computers & Operations Research*, 33(6):1639–1663.

Poot, A., Kant, G., and Wagelmans, A. (1999). A savings based method for real-life vehicle routing problems. Technical report, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute.

Reinelt, G. (2015). Universität Heidelberg, Germany. INTERNET. https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/. Cited 9 March 2015, Last updated 2015.

Zhou, G., Min, H. and Gen, M., 2002. The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach. *Computers & Industrial Engineering*, 43(1-2), pp.251-261.

Biographies

Dr. Jacoba H. Bührmann is currently a senior lecturer at the School of MIA, specialising in Operations Research and Machine Learning. She has industry experience in areas including business analytics, data and optimisation analysis, business intelligence and application development. Her areas of interest include metaheuristic and approximation algorithms, mathematical programming, machine learning and data analytics.

Prof. M. Montaz Ali currently works at the School of Computational and Applied Mathematics (CAM), University of the Witwatersrand. Montaz does research in Global Optimization, Optimal Control, Operations Research, Statistics and Applied Mathematics. His area of specialism is currently in mixed integer nonlinear programming problems and optimization in big data.'

Dr. Ian Campbell was a senior lecturer at the School of MIA until 2017. His PhD was in construction heuristics for the airline taxi problem. He was particularly interested in the use of simulation modeling together with metaheuristics to solve NP complete problems.