Solution strategy based on Gaussian mixture models and dispersion reduction for the capacitated centered clustering problem

Santiago-Omar Caballero-Morales Corresp. 1

1 Postgraduate Department of Logistics and Supply Chain Management, Universidad Popular Autonóma del Estado de Puebla, Puebla, Puebla, Mexico

Corresponding Author: Santiago-Omar Caballero-Morales
Email address: santiagoomar.caballero@upaep.mx

The Capacitated Centered Clustering Problem (CCCP) - a multi-facility location (MFL) model - is very important within the logistics and supply chain management fields due to its impact on industrial transportation and distribution. However, solving the CCCP is a challenging task due to its computational complexity. In this work, a strategy based on Gaussian mixture models (GMMs) and dispersion reduction is presented to obtain the most likely locations of facilities for sets of client points considering their distribution patterns. Experiments performed on large CCCP instances, and considering updated best-known solutions, led to estimate the performance of the GMMs approach, termed as Dispersion Reduction GMMs (DRG), with a mean error gap smaller than 2.6%. This result is more competitive when compared to VNS (Variable Neighborhood Search), SA (Simulated Annealing), GA (Genetic Algorithm), and CKM (CKMeans) and faster to achieve when compared to the best-known solutions obtained by Tabu-Search (TS) and Clustering Search (CS).

Solution Strategy based on Gaussian Mixture Models and Dispersion Reduction for the Capacitated Centered Clustering Problem

Santiago-Omar Caballero-Morales

Postgraduate Department of Logistics and Supply Chain Management, Universidad Popular Autonoma del Estado de Puebla, Puebla, Mexico

Corresponding author:

Santiago-Omar Caballero-Morales

Email address: santiagoomar.caballero@upaep.mx

ABSTRACT

The Capacitated Centered Clustering Problem (CCCP) - a multi-facility location (MFL) model - is very important within the logistics and supply chain management fields due to its impact on industrial transportation and distribution. However, solving the CCCP is a challenging task due to its computational complexity. In this work, a strategy based on Gaussian mixture models (GMMs) and dispersion reduction is presented to obtain the most likely locations of facilities for sets of client points considering their distribution patterns. Experiments performed on large CCCP instances, and considering updated best-known solutions, led to estimate the performance of the GMMs approach, termed as Dispersion Reduction GMMs (DRG), with a mean error gap smaller than 2.6%. This result is more competitive when compared to VNS (Variable Neighborhood Search), SA (Simulated Annealing), GA (Genetic Algorithm), and CKM (CKMeans) and faster to achieve when compared to the best-known solutions obtained by Tabu-Search (TS) and Clustering Search (CS).

INTRODUCTION

Facilities are very important infrastructure within the supply chain as they support production, distribution and warehousing. Due to this, many operative processes associated to facilities are subject to optimization. Fields such as facility layout planning are crucial for smooth material handling and production flow (Mohamadghasemi, A. and Hadi-Vencheh, A., 2012; Hadi-Vencheh, A. and Mohamadghasemi, A., 2013; Niroomand, S. et al., 2015).

On the other hand, where to locate facilities within specific regions is a central problem for strategic decisions of transportation and distribution (Chaves, A.A. et al., 2007). This is because the distance between the facilities and the customers (demand or client points) is crucial to provide efficient transportation and distribution services.

Within this context, the Capacitated Centered Clustering Problem (CCCP) is one of the best-known and most challenging multi-facility location (MFL) problems in the fields of operations research, logistics and supply chain management (Chaves, A.A. and Nogueira-Lorena, L.A., 2010; Mahmoodi-Darani, N. et al., 2013). The CCCP is focused on determining clusters or groups of demand or client points which can lead to minimum average distances to their centroids (where facilities are to be located). The cumulative demand of the client points assigned to a cluster cannot exceed the capacity of the facility located at its centroid (Chaves, A.A. and Nogueira-Lorena, L.A., 2010, 2011; Negreiros, M. and Palhano, A., 2006).

The mathematical formulation of the CCCP is presented as follows (Chaves, A.A. and Nogueira-Lorena, L.A., 2011; Negreiros, M. and Palhano, A., 2006):

$$\min \sum_{i \in I} \sum_{k \in K} \|x_i - m_k\|^2 y_{ik} \qquad (1)$$

subject to

$$\sum_{k \in K} y_{ik} = 1 \;\; \forall i \in I \tag{2}$$

$$\sum_{i \in I} y_{ik} = n_k \;\; \forall k \in K \tag{3}$$

$$\sum_{i \in I} x_i y_{ik} \leq n_k m_k \;\; \forall k \in K \tag{4}$$

$$\sum_{i \in I} d_i y_{ik} \leq C_k \;\; \forall k \in K. \tag{5}$$

$$m_k \in \Re^l, \; n_k \in N, \; y_{ik} \in \{0,1\} \; \forall i \in I, \; \forall k \in K \tag{6}$$

where (a) I is the set of all demand points or clients (N=number of points); (b) K is the set of clusters (group of clients assigned to a facility) with $|K|$=number of facilities; (c) x_i is the geometric position of the i-th point in the $\Re^l$ space (l=2 for a two-dimensional space and $x_i = [cx_i \; cy_i]$ where cx_i is the x-coordinate and cy_i is the y-coordinate of x_i); (d) m_k is the geometric position of the centroid of a cluster k (i.e., the location of the k-th facility, and if l=2, m_k=$[cx_k \; cy_k]$ where cx_k is the x-coordinate and cy_k is the y-coordinate of m_k); (e) y_{ik}=1 if the point i is assigned to cluster k and y_{ik}=0 otherwise; (f) n_k is the number of points in the cluster k; (g) d_i is the demand of the point i; and (h) C_k is the capacity of each cluster k.

In this formulation Eq. (1) defines the non-linear objective function which consists on minimizing the total distance between each point and the centroid of the cluster where the point is assigned. As mentioned in (Chaves, A.A. and Nogueira-Lorena, L.A., 2010) the geometric position of the centroid depends on the points that compose the cluster. Eq. (2) and Eq. (3) are restrictions that define that each point is only assigned to one cluster and provides the number of points in each cluster respectively. Eq. (4) is the restriction that locates the centroid of each cluster at its geometric center while Eq. (5) is the restriction that defines that the total demands of the points assigned to a cluster k must not exceed its capacity. Finally, Eqs. (6) define the nature of m_k, the decision variable y_{ik} and the upper limits for the number of individuals or points per cluster (n_k). n_k is obtained from the values of the decision variables y_{ik} as these determine the number of points to be assigned at each cluster (i.e., if for k=1 the optimal solution is $y_{20,1}$=$y_{13,1}$=$y_{46,1}$=1, then n_1=3). This is explained in more detail in Section *Standard EM Algorithm.*

The CCCP has been approached with different solving methods (particularly meta-heuristics) due to its combinatorial nature and NP-hard computational complexity (hence, it is difficult to solve it with exact methods) (Chaves, A.A. et al., 2007; Chaves, A.A. and Nogueira-Lorena, L.A., 2010; Herda, M., 2015; Negreiros, M. and Palhano, A., 2006). While there are works reported in the literature that have achieved very competitive results for small, medium and large instances of the CCCP, their performance is dependent of the size or scale of the instances.

In this work the technique of Gaussian mixture models (GMMs) is proposed to estimate clusters of maximum probability in order to provide feasible solutions for large instances of the CCCP ($N > 1000$ points). This is accomplished by the meta-heuristic termed as DRG (Dispersion Reduction GMMs) which integrates the following algorithms:

- an adapted Expectation-Maximization (EM) algorithm to estimate the parameters of GMMs and generate clusters of points of maximum likelihood with the capacity requirements of the CCCP;

- a dispersion reduction algorithm to compress large CCCP data and achieve near optimal results within faster computational times.

Both, the application of the GMMs and the validity of the Expectation-Maximization (EM) algorithm for the CCCP, have not been explored in previous works. Experiments of the DRG on large instances of the CCCP considering the benchmarks of the Clustering Search (CS) algorithm (Muniz-de-Oliveira, A.C. et al., 2013) and Tabu-Search (TS) (Fernandes-Muritiba, A.E. et al.,

80 2012) led to a mean error less than 2.6%. In general, the DRG algorithm performed better than
81 Variable Neighborhood Search (VNS), Simulated Annealing (SA), Genetic Algorithms (GA) and
82 CKMeans (CKM). The advances of the present work can provide a better understanding of
83 Gaussian modeling and the EM algorithm for their application in facility location problems.

84   The contributions of this paper are presented as follows: First, in Section *Clustering* an
85 overview of clustering and the mathematical foundations of GMMs are presented. Also, the
86 findings regarding the specific dispersion reduction process required by CCCP data in order
87 to make it suitable for the EM algorithm are presented and discussed. Then, in Section *DRG*
88 *Meta-heuristic* the details of the DRG meta-heuristic are presented while the results regarding
89 its performance are discussed in Section *Results and Assessment*. Finally, conclusions and
90 improvement recommendations are discussed in Section *Conclusions and Future Work*.

CLUSTERING

92 Most of the solving algorithms for the CCCP perform clustering within the search process for
93 initial partitioning of the set of demand points. This partitioning is then improved by exchanging
94 certain points between the most promising partitions or clusters (Hansen, P. and Jaumard, B.,
95 1997; Negreiros, M. and Palhano, A., 2006).

96   Formally, clustering involves the grouping of data points in a way that homogeneity of points
97 within a group, and the heterogeneity of points between groups, are maximized (Chaves, A.A.
98 and Nogueira-Lorena, L.A., 2010; Negreiros, M. and Palhano, A., 2006). For the CCCP the
99 technique used for clustering has an important role in the quality of the solutions, which may
100 lead to significant differences from best-known results (i.e., errors from to 15.0% (Radiah, S.
101 et al., 2013) to 50% (Negreiros, M. and Palhano, A., 2006)).

Gaussian Mixture Models (GMMs)

103 Gaussian Mixture Models (GMMs) have been widely studied for data modeling within the field
104 of pattern classification based on Bayes decision theory (Theodoridis, S. and Koutroumbas,
105 K., 2010). For more accurate modeling of multi-dimensional patterns a mixture of Gaussian
106 distributions can be used. Each mixture component is defined by two main parameters: a
107 mean vector and a covariance matrix (Forsyth, D., 2012; Theodoridis, S. and Koutroumbas, K.,
108 1979, 2010). Within this context, "clusters" can be characterized by each Gaussian component
109 (mixture) which can model sub-sets of the whole set of patterns with maximum likelihood.

110   There are important differences between clustering that can be obtained with GMMs and
111 centroid-based clustering techniques (such as K-Means or K-Nearest Neighbors). The following
112 can be mentioned:

113 - Over-fitting (e.g., the model "overreacts" to minor fluctuations in the training data for
114   prediction purposes) can be avoided with Gaussian distribution-based clustering.

115 - Clusters defined with Gaussian distributions can have different shapes and sizes. By
116   contrast, centroid-based clustering algorithms tend to find clusters of comparable size of
117   (more or less) symmetrical shape (Mohammed, M. et al., 2016).

118 - At each iteration, Gaussian distribution-based clustering performs, for a given point, a
119   "soft-assignment" to a particular cluster (there is a degree of uncertainty regarding the as-
120   signment). The centroid-based clustering performs a hard-assignment (or direct assignment)
121   where a given point is assigned to a particular cluster and there is no uncertainty.

Due to these differences, the GMM-based clustering was considered as an alternative to
generate feasible solutions for the CCCP. In terms of the CCCP formulation described in
Section *Introduction* a cluster can be modeled by a single Gaussian probability density function
(PDF). Hence, the location "patterns" of a set of clients X can be modeled by a mixture of K
Gaussian PDFs where each PDF models a single cluster. If the set contains N clients, then
$X = [x_{i=1}, x_{i=2}, ..., x_{i=N}]$ and the mixture can be expressed as:

$$p(X) = \sum_{k=1}^{K} P_k p(X|k) = \sum_{k=1}^{K} P_k \mathcal{N}(X|m_k, S_k), \tag{7}$$

where $k = 1, ..., K$ and $|K| = p$ is the number of Gaussian PDFs, $p(X|k)$ represents the probabilities of each Gaussian PDF describing the set of clients X (Theodoridis, S. and Koutroumbas, K., 2010), and P_k is the weight associated to each Gaussian PDF (hence, $\sum_{k=1}^{K} P_k = 1.0$). Each Gaussian component can be expressed as:

$$p(X|k) = \mathcal{N}(X|m_k, S_k) = \frac{1}{\sqrt{2\pi|S_k|}} e^{-\frac{1}{2}(X-m_k)^T S_k^{-1}(X-m_k)} \ \forall k \in K, \tag{8}$$

122 where m_k is the mean vector and S_k is the covariance matrix of the k-th Gaussian PDF or
123 k-th cluster. Note that m_k and each element of X (i.e., any x_i) must have the same size or
124 dimension which is defined by l (in this case, $l=2$ because each point consists of a x-coordinate
125 and a y-coordinate). Finally, S_k is a matrix of size $l \times l$.
126   For clustering purposes, m_k can model the mean vector of a sub-set of points in X which is
127 more likely to be described by the Gaussian PDF k as estimated by Eq. (8). If the points in this
128 sub-set of X are clustered, then m_k can represent the "centroid" of the cluster k and S_k can
129 model the size and shape of the cluster k (Bishop, C.M., 2006; Theodoridis, S. and Koutroumbas,
130 K., 2010). Observe that $X - m_k$ defines a distance or difference between each point in X and
131 the centroid (located at m_k) of each cluster k. Thus, the estimation of probabilities considers
132 the distance between each point x_i and each cluster k.
133   The parameters of the Gaussian PDFs (P_k, m_k and S_k) that best model (describe) each sub-set
134 of the whole pattern X can be estimated by the iterative algorithm of Expectation-Maximization
135 (EM) (Bishop, C.M., 2006; Theodoridis, S. and Koutroumbas, K., 2010).
136   The advantage of this Gaussian approach for clustering is that faster inference about the
137 points x_i that belong to a specific cluster k may be obtained considering all points. In this
138 context, it is important to mention that due to the probabilistic nature of the inference process,
139 a single point x_i is not directly assigned to a specific cluster (as it is required by the CCCP)
140 because all points have probabilities associated to all clusters (i.e., "soft-assignment"). Also,
141 this approach does not consider the capacity of each cluster and the demand associated to each
142 client point. Thus, restrictions about the quantity of points x_i associated to each cluster are
143 not integrated in this algorithm. In order to consider these requirements and restrictions a
144 modification on the EM algorithm was performed. This is described in the following sections.

145 **Dispersion Reduction in GMMs Performance for the CCCP**
146 CCCP data, which consists of x and y coordinates, represents a particular challenge for GMMs.
147 This is because the values of coordinates can affect the computation of the exponential element
148 of Eq. (8). Also, dispersion of data may affect convergence of the EM algorithm for P_k, m_k
149 and S_k. Thus, specific re-scaling or coding of CCCP data was required to reduce the effect of
150 dispersion and coordinates' values on the computations of GMMs.
151   In order to reduce dispersion of the CCCP data the compression algorithm presented in
152 Figure 1 was performed. It is important to mention that compression is only performed on the
153 coordinates' values, and not on the number of data points (thus, the size of the instance remains
154 the same).
155   As presented, the original x-y coordinates were replaced by their unique indexes. This led to
156 elimination of empty spaces between data points. Coding of the compressed data was performed
157 within the interval $[0, 1]$ as presented in Figure 1. This coding made the compressed data more
158 suitable for fast computation of Eq. (8).
159   An important issue regarding the use of GMMs for the CCCP is the restriction on the number
160 of Gaussian components. In this regard, extensive research has been performed to determine
161 the most suitable number of Gaussian components for different sets of data (McLachlan, G.F.
162 and Rathnayake, S., 2014). However, available research does not consider the capacity aspect of
163 clustering which is the main feature of CCCP data. Nevertheless, a restriction on the number of
164 Gaussian components must be considered because, as discussed in (Fraley, C. and Raftery, A.,
165 1998), the EM algorithm may not be suitable for cases with very large numbers of components.
166   Due to this situation a maximum number of 30 Gaussian components was considered. This
167 restriction was based on the *DONI* database, which has some of the CCCP instances with the
168 largest number of client points (Fernandes-Muritiba, A.E. et al., 2012) (i.e., $N=13000$ client
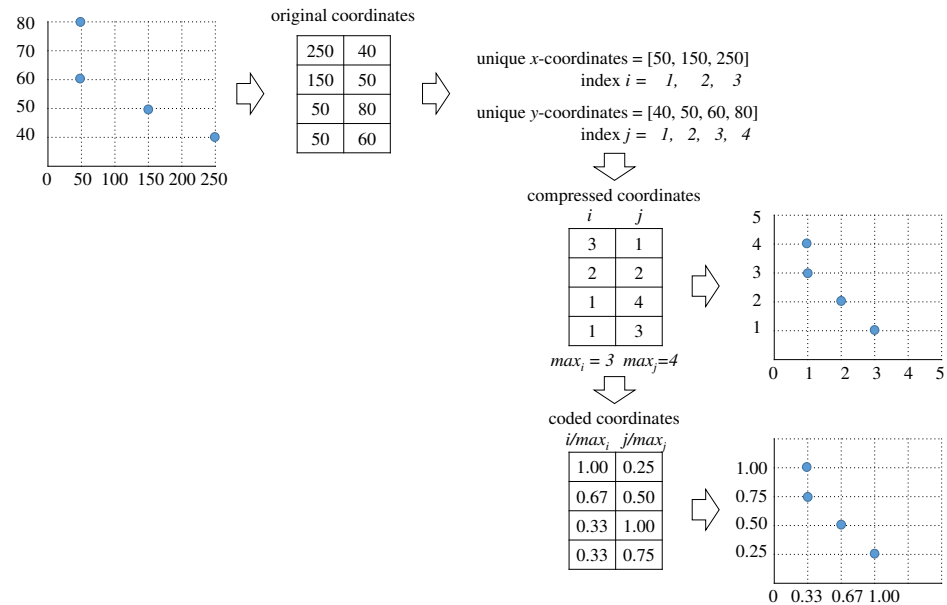
Figure 1. CCCP data compression and coding algorithm (dispersion reduction algorithm).

points with p=30 facilities). A discussion on future extensions for the Gaussian approach and the EM algorithm to address larger number of components is presented in the final section.

DRG META-HEURISTIC

Standard EM Algorithm

Figure 2 presents the structure of the standard EM algorithm (Bishop, C.M., 2006; Theodoridis, S. and Koutroumbas, K., 1979, 2010) considering the variables defined by Eq. (7) and Eq. (8). In this case, X represents the array of compressed and coded $x - y$ coordinates of all points x_i (the array structures presented in Figure 2 and 3 follow the EM formulation described in (Theodoridis, S. and Koutroumbas, K., 1979)).

As presented in Figure 2, the locations of the N clients are stored into the array X which consists of a matrix of dimension $l \times N$ where: (a) N is the number of client points, and (b) l=2 as each column vector of X consists of the values cx_i and cy_i that identify the compressed and coded $x - y$ coordinates of the client point x_i where i=1,...,N.

The EM algorithm starts with initial values for m_k, S_k and P_k. Values for m_k and S_k were randomly generated as follows:

$$m_k = [random(cx_{min}, cx_{max}), random(cy_{min}, cy_{max})]^T \quad \forall k \in K, \tag{9}$$

$$S_k = random(0.0, 0.1) \times I_l \quad \forall k \in K, \tag{10}$$

where (cx_{min}, cx_{max}) and (cy_{min}, cy_{max}) are the minimum and maximum values throughout all compressed and coded x and y coordinates respectively, and I_l is the identity matrix of size $l \times l$.

For P_k a lower bound for K was obtained by considering the total demand of the points x_i and the capacity of the clusters C_k. Because all clusters have the same capacity, $C_k = C$. Then, K and P_k were obtained as follows:

$$K = \frac{\sum_{i=1}^{N} d_i}{C}, \tag{11}$$

$$P_k = \frac{1}{|K|} \quad \forall k \in K. \tag{12}$$

The stage of Expectation starts with these initial values for m_k, S_k and P_k. An initial computation of assignment or "responsibility" scores $\gamma(z_{ik})$ is performed to determine which
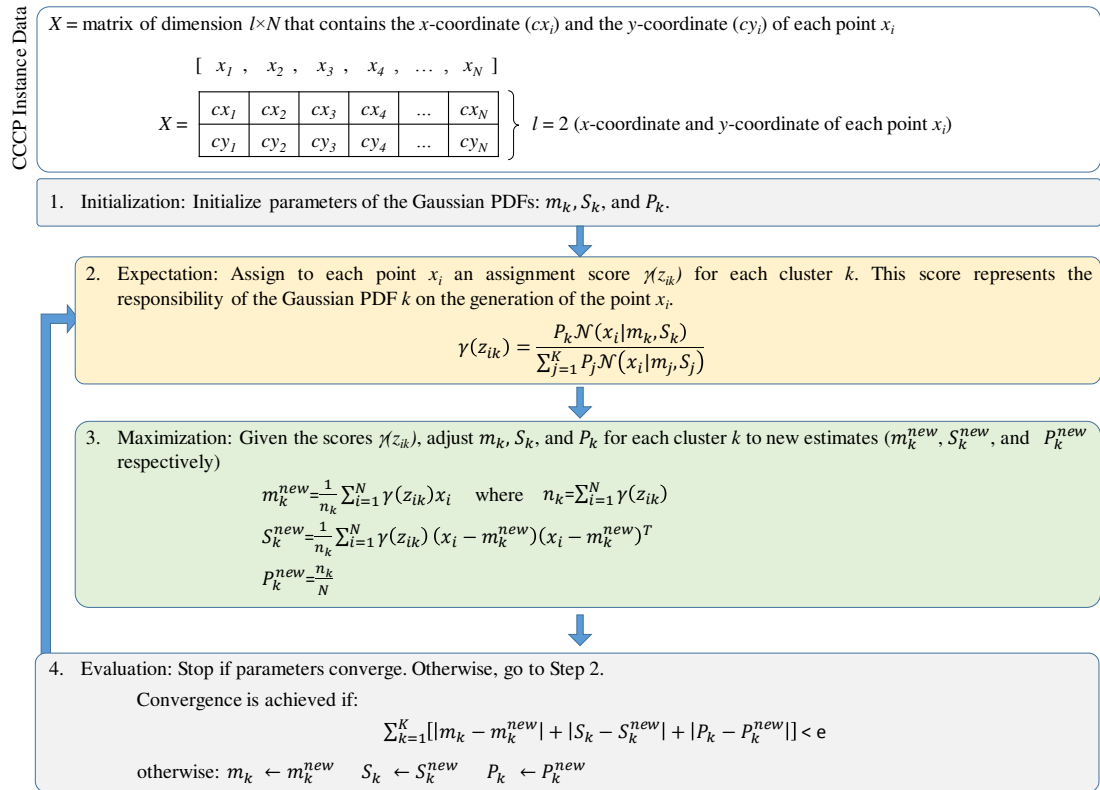
Figure 2. Structure of the Standard Expectation-Maximization (EM) Algorithm.

x_i is more likely to be associated to a particular cluster (and thus, to belong to this cluster) with parameters m_k, S_k and P_k (Bishop, C.M., 2006). Observe that, as presented in Figure 2 (Step 2), $\gamma(z_{ik})$ is computed by means of Eq. (8). These scores can lead to provide values for the decision variable y_{ik} of the CCCP objective function (see Eq. (1)). This process will be discussed in the following section.

Then, the stage of Maximization integrates the scores $\gamma(z_{ik})$ into the re-estimation of the cluster's parameters which leads to m_k^{new}, S_k^{new} and P_k^{new}. Convergence is achieved if the total error or absolute difference between the previous and new estimates is less than a given threshold e (in this case, $e=0.5$). If convergence is not achieved, then $m_k \leftarrow m_k^{new}$, $S_k \leftarrow S_k^{new}$ and $P_k \leftarrow P_k^{new}$.

Modified EM Algorithm with Direct Assignment and Capacity Restrictions

As discussed in Section *Gaussian Mixture Models (GMMs)* the advantage of the GMMs approach is that faster inference about the points x_i that belong to a specific cluster k can be obtained. This inference is performed based on the probabilities defined by Eq. (8) where the parameters of the Gaussian PDFs are estimated with the standard EM algorithm. However under this approach a single point x_i is not directly assigned to a specific cluster (as it is required by the CCCP) because all points have probabilities associated to all clusters (i.e., "soft-assignment"). Also, while the standard EM algorithm can determine the most likely sub-sets of client points in X to be covered by each cluster k, these sub-sets are not restricted by the capacity of the cluster and the demand of each assigned client point. Hence, the standard EM algorithm was modified in order to comply with the requirements and restrictions of the CCCP.

As previously mentioned, the score $\gamma(z_{ik})$ computed in Step 2 of the EM algorithm (see Figure 2) represents the likelihood or responsibility of the cluster k on the generation of the point x_i (Bishop, C.M., 2006). As presented in Figure 3 all scores are stored into a matrix γ of dimension $K \times N$, where each column vector $[\gamma(z_{i1}), ..., \gamma(z_{iK})]^T$ contains the assignment scores

of the point x_i to all clusters (thus, $\sum_{k=1}^{K} \gamma(z_{ik}) = 1.0 \; \forall i \in N$). These scores represent the basis
for defining the decision variable y_{ik}.
From Section *Introduction* it was explained that $y_{ik}=1$ if the point x_i is assigned to cluster k
and $y_{ik}=0$ otherwise (each point x_i can be assigned to only one cluster). Based on the scores
of $\gamma(z_{ik})$ it was determined that for each vector x_i the cluster assignment would be defined by
the cluster k with maximum $\gamma(z_{ik})$ value. If two or more clusters share the same maximum
likelihood, then one of them is randomly assigned. An example of this assignment process is
presented in Figure 3.
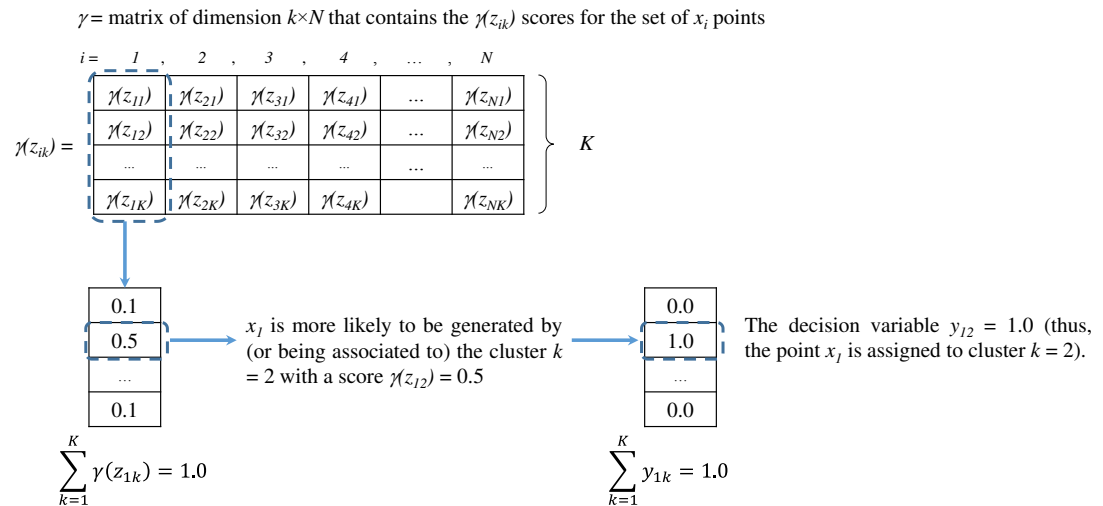
Figure 3. Assignment of values for y_{ik} from the scores of $\gamma(z_{ik})$.

By determining the unique assignment of each point x_i to each cluster k at Step 2 of the EM
algorithm (see Figure 2), the number of points assigned to each cluster is obtained. This leads
to determine the cumulative demand of the points assigned to each cluster. This information is
stored into the vector:

$$Demands = [D_1, D_2, ..., D_K], \tag{13}$$

where D_k represents the cumulative demand of the points assigned to cluster k and it must
satisfy $D_k \leq C_k$. This vector is important to comply with the capacity restrictions because it
was found that homogenization of the cumulative demands D_k contributes to this objective.
Homogenization is achieved by minimizing the coefficient of variation between all cumulative
demands:

$$\min CV = \frac{Standard\ Deviation(Demands)}{Mean(Demands)}. \tag{14}$$

The objective function defined by Eq. (14) is integrated within the evaluation step of the
standard EM algorithm (see Step 4 of Figure 2). This leads to the modified EM algorithm with
capacity restrictions where convergence is based on two objectives:

- minimization of the error (less than a threshold of $e=0.5$) between the estimates of m_k, P_k
  and S_k;

- minimization of the coefficient of variation of *Demands*.

An important issue regarding the minimization of CV is that there is the possibility that
$\min CV$ may not lead to comply the condition $D_k \leq C_k$ for all clusters k. This was addressed
by the following strategy: if $\min CV$ leads to a cluster k where the condition $D_k \leq C_k$ is not
complied, then the farthest assigned points are re-assigned to other clusters with closer centroids
and available capacity.

Figure 4 presents an example of the clustering achieved without and with the objective function defined by Eq. (14). As presented, the clusters are more accurately defined with the integration of Eq. (14). Also, clusters comply with the capacity restrictions of the CCCP.

It is important to remember that the proposed DRG meta-heuristic consists of the modified EM algorithm which provides the cluster assignation for each point x_i considering compressed and coded CCCP data. Thus, Figure 4 presents the decoded and uncompressed points x_i (i.e., original cx_i and cy_i coordinates) assigned to each cluster based on the assignments of the DRG meta-heuristic with the modified EM algorithm on compressed and coded data.
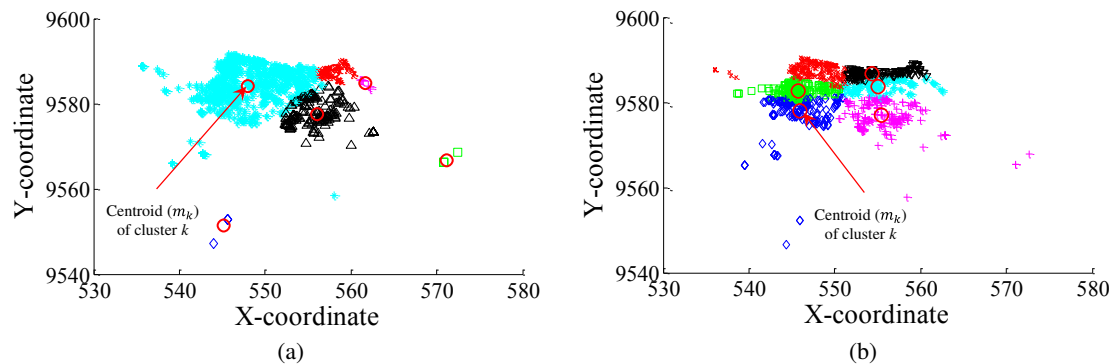
Figure 4. Clustering with (a) Standard EM Algorithm and (b) DRG Meta-heuristic (Modified EM Algorithm with decoded and uncompressed CCCP data).

Further improvement of the assignments and the centroids at m_k are achieved by a Greedy algorithm that verifies that all points x_i are assigned to the closest cluster. This leads to additional exchange and insertion/deletion of complying client points x_i between clusters. These operations must comply with the capacity restrictions of each cluster. The assignments are updated if the insertion/deletion is valid. This leads to re-estimation of the locations of the m_k centroids.

RESULTS AND ASSESSMENT

Implementation of the DRGMMs meta-heuristic was performed with the software *MATLAB* on a laptop computer with the following hardware: Intel CORE i7-4500U CPU at 2.40 GHz with 8 GB RAM. For testing purposes the libraries presented in Table 1 were considered.

For comparison purposes the methods and best-known results presented in (Chaves, A.A. and Nogueira-Lorena, L.A., 2010, 2011; Fernandes-Muritiba, A.E. et al., 2012; Muniz-de-Oliveira, A.C. et al., 2013; Negreiros, M. and Palhano, A., 2006; Palhano, A. et al., 2008; Pereira, M.A. and Senne, L.F., 2008) for the CCCP were considered. These methods are the following:

- **Best-known solutions***. The benchmarks reported in (Muniz-de-Oliveira, A.C. et al., 2013) have been used for the assessment of the most efficient methods for the CCCP. For this work, these benchmarks were updated with the best-known results obtained with Tabu-Search (TS) (Fernandes-Muritiba, A.E. et al., 2012) which is considered as the current state of the art algorithm for the CCCP (Carvalho, R. et al., 2017; Pereira, W.M. and Carvalho, R., 2017).

- **Clustering Search (CS)**. Hybrid method which combines meta-heuristics and local search heuristics in which the search is intensified only in areas of the search space that deserve special attention (promising regions). The main idea of CS is to identify promising areas of the search space by generating solutions through a meta-heuristic and "clustering" them into groups that are further explored with local search heuristics (Chaves, A.A. and Nogueira-Lorena, L.A., 2010, 2011; Muniz-de-Oliveira, A.C. et al., 2013). The results reported in (Muniz-de-Oliveira, A.C. et al., 2013) were considered for comparison with DRG.

Table 1. Libraries of CCCP instances considered for testing and assessment.

Instance	N	K
doni1	1000	6
doni2	2000	6
doni3	3000	8
doni4	4000	10
doni5	5000	12
doni6	10000	23
doni7	13221	30
SJC1	100	10
SJC2	200	15
SJC3a	300	25
SJC4a	402	30
TA25	25	5
TA50	50	5
TA60	60	5
TA70	70	5
TA80	80	7
TA90	90	4
TA100	100	6

- **A two-phase heuristic using Variable Neighborhood Search (VNS)**. The results reported in (Negreiros, M. and Palhano, A., 2006) were considered for comparison with DRG. These results were also reviewed by (Fernandes-Muritiba, A.E. et al., 2012) for comparison with TS.

- **CKMeans (CKM)**. The results reported in (Palhano, A. et al., 2008) were considered for comparison with DRG. These results were also reviewed by (Chaves, A.A. and Nogueira-Lorena, L.A., 2011) for comparison with GA.

- **Two-Column Generation (TCG)**. The results reported in (Pereira, M.A. and Senne, L.F., 2008) were considered for comparison with DRG.

- **Simulated Annealing (SA) and Genetic Algorithm (GA)**. CS performs a clustering strategy on solutions generated by a meta-heuristic, and the research in (Chaves, A.A. and Nogueira-Lorena, L.A., 2010) and (Chaves, A.A. and Nogueira-Lorena, L.A., 2011) reported the comparison of results of the CS method with the SA and GA meta-heuristics respectively. For assessment purposes, (Chaves, A.A. and Nogueira-Lorena, L.A., 2010) and (Chaves, A.A. and Nogueira-Lorena, L.A., 2011) also reported the comparison of results of SA and GA without the CS strategy (e.g., independent performance of SA and GA). These independent results were considered for comparison with DRG.

- **Tabu-Search (TS)**. The TS algorithm reported in (Fernandes-Muritiba, A.E. et al., 2012) is currently considered state of the art by Rodrigo de Carvalho *et al.* for the CCCP (Carvalho, R. et al., 2017; Pereira, W.M. and Carvalho, R., 2017). The results reported in (Fernandes-Muritiba, A.E. et al., 2012) were considered for comparison with DRG.

In order to compute the error, gap or deviation from the updated best-known solutions the error metric presented by (Yousefikhoshbakht, M. and Khorram, E., 2012) was considered:

$$Error(\%) = 100 \times \left[\frac{a-b}{b} \right], \tag{15}$$

where a is the cost or distance of the best solution found by the algorithm for a given instance while b is the best known solution for the same instance. In this case it is important to mention that the best-known solution is not necessarily the optimal solution due to the NP-hard complexity of the CCCP. Initially, this metric was computed for the DRG, VNS, SA, CS, TS and GA methods because the reference data was available for all sets of instances.

DRG vs. VNS-SA-CS-TS-GA

Table 2 presents the best results of the DRG meta-heuristic for the considered instances. Information regarding the runs performed by each method to report the best result is also presented when available. Also, information regarding the programing language and the hardware used by the authors of the reviewed methods were also included.

As reported in the literature, CS and TS are the most competitive methods to solve the CCCP with a mean error gap of 0.78% and 0.61% respectively (and thus, were considered to update the benchmark solutions). Then, the proposed DRG meta-heuristic stands as the next most competitive method with a mean error gap of 2.58%. The VNS, SA and GA methods show a more significant mean error gap with 4.69%, 13.79% and 7.03% respectively. Also, VNS, SA and GA show a higher variability in performance which is characterized by an increased standard deviation when compared with their mean error gap. The DRG shows a balanced mean and standard deviation, thus its performance is more robust and consistent through all different CCCP instances.

DRG vs. CKM-TCG

Tables 3 and 4 present the results of the DRG meta-heuristic for the instances where reference data of the CKM and TCG methods were available. As presented in Table 3, the DRG meta-heuristic is more competitive than the CKM method. Also, as previously observed, the DRG is more consistent.

When compared to the TCG method, this is more competitive than the DRG approach even though the error gaps are minimal (less than 1.5%).

Error and Speed vs. Size of the Instance

Figure 5 presents the graphical review of the error gaps of all methods based on the size of the test instance. TS and CS are located on the x-axis as they are the benchmark methods. It can be observed that, as the size of the instance increases, the error gap of SA, GA and VNS significantly increases. TCG presents very small error gaps with small instances (less than 1000 points) and CKM reports error gaps comparable to SA for small-medium size instances (less than 5000 points). In contrast, DRG performs consistently through all instances, decreasing its error gap as the instance grows from medium to large size (up to 130221 points).

Regarding speed, Figure 6 presents the computational (running) times reported by the reviewed methods. While TS and CS are the benchmark methods, these take over 25000 seconds to reach the best-known solution for the largest instance. Note that for these methods, their computational times exponentially increase for instances larger than 6000 points.

In contrast, SA is very consistent with a computational time of approximately 1000 seconds through all instances. GA significantly increases for instances larger than 6000 points (up to 7000 seconds for the largest instance). However, these methods have the largest error gaps as reviewed in Figure 5. The speed pattern of DRG is very similar to that of GA, however, as reviewed in Figure 5, its error gap is the closest to the benchmark methods for instances larger than 6000 points.

It is important to mention that this comparison may not be fair due to the differences in the programming language and the hardware used for implementation and testing of all the methods. In order to compare running speed all methods should be tested with the same hardware and be implemented with the same programming language by the same software developer (the developer's expertise may also affect the speed performance of the software). Due to the difficulty of achieving this task, in practice the comparison is commonly performed on the best results obtained by other methods as performed in (Chaves, A.A. and Nogueira-Lorena, L.A., 2011; Fernandes-Muritiba, A.E. et al., 2012). Running time is measured in order to determine if the proposed method can provide a solution within reasonable time considering standard resources of hardware and software. Due to this situation, the DRG can provide very suitable results (< 2.6% error gap) within very reasonable time.

Table 2. Performance of DRG, VNS, SA, CS, TS and GA on CCCP instances when compared to updated Best-known solutions*: (a) MATLAB, Intel CORE i7 at 2.4 GHz and 8 GB RAM, (b) AMD ATHLON at 1.6 GHz and 512 MB RAM, (c) C++, Pentium 4 at 3.0 GHz, (d) C++, Pentium 4 at 3.0 GHz, (e) C++, Intel CORE 2 Quad Q9550 CPU at 2.83 GHz and 4 GB RAM, (f) C++, Pentium 4 at 3.0 GHz.

Instance	Best Known*	N	K	DRG (a) (10 runs)	Error (%)	VNS (b)	Error (%)	SA (c) (10 runs)	Error (%)	CS (d) (20 runs)	Error (%)	TS (e) (25 runs)	Error (%)	GA (f) (20 runs)	Error (%)
TA25	1251.44	25	5	1256.62	0.41	1251.44	0.00	1273.46	1.76	1251.44	0.00	1251.45	0.00	1273.46	1.76
TA50	4474.52	50	5	4476.92	0.05	4476.12	0.04	4478.15	0.08	4474.52	0.00	4474.52	0.00	4474.52	0.00
TA60	5356.58	60	5	5356.58	0.00	5356.58	0.00	5370.05	0.25	5356.58	0.00	5356.58	0.00	5356.58	0.00
TA70	6240.67	70	5	6270.45	0.48	6241.55	0.01	6267.89	0.44	6240.67	0.00	6240.67	0.00	6267.89	0.44
TA80	5515.46	80	7	5748.30	4.22	5730.28	3.89	5780.55	4.81	5730.28	3.89	5730.28	3.89	5775.69	4.72
TA90	8899.05	90	4	9069.85	1.92	9103.21	2.29	9069.85	1.92	9069.85	1.92	9069.85	1.92	9133.35	2.63
TA100	8102.04	100	6	8122.36	0.25	8122.67	0.25	8153.64	0.64	8102.04	0.00	8102.04	0.00	8189.44	1.08
SJC1	17359.75	100	10	17492.77	0.77	17696.53	1.94	17363.47	0.02	17359.75	0.00	17359.75	0.00	17363.47	0.02
SJC2	33181.65	200	15	33317.03	0.41	33423.84	0.73	33458.40	0.83	33181.65	0.00	33181.65	0.00	33324.04	0.43
SJC3a	45354.38	300	25	46395.96	2.30	47985.29	5.80	46847.61	3.29	45354.38	0.00	45356.35	0.00	46682.60	2.93
SJC4a	61931.60	402	30	62701.95	1.24	66689.96	7.68	64981.66	4.92	61931.60	0.00	61993.66	0.10	65978.89	6.54
DONI1	3021.41	1000	6	3074.97	1.77	3021.41	0.00	3138.67	3.88	3022.26	0.03	3025.12	0.12	3122.02	3.33
DONI2	6080.70	2000	6	6456.14	6.17	6080.70	0.00	6985.30	14.88	6372.81	4.80	6384.84	5.00	6394.96	5.17
DONI3	8343.49	3000	8	8911.87	6.81	8769.05	5.10	9653.27	15.70	8438.96	1.14	8343.49	0.00	8945.88	7.22
DONI4	10777.64	4000	10	11453.68	6.27	11516.14	6.85	13328.16	23.66	10854.48	0.71	10777.64	0.00	11130.16	3.27
DONI5	11114.67	5000	12	11776.59	5.96	11635.18	4.68	13920.49	25.24	11134.94	0.18	11114.67	0.00	11341.52	2.04
DONI6	15610.46	10000	23	16362.38	4.82	18443.50	18.15	29102.49	86.43	15722.67	0.72	15610.46	0.00	19226.96	23.17
DONI7	18484.13	13221	30	18974.23	2.65	23478.79	27.02	29484.66	59.51	18596.74	0.61	18484.13	0.00	29915.77	61.85
				Mean=	2.58	Mean=	4.69	Mean=	13.79	Mean=	0.78	Mean=	0.61	Mean=	7.03
				StdDev=	2.45	StdDev=	7.18	StdDev=	23.43	StdDev=	1.41	StdDev=	1.48	StdDev=	14.68

Table 3. Performance of DRG and CKM on CCCP instances when compared to Best-known solutions*.

Instance	DRG	Error (%)	CKM	Error (%)
SJC1	17492.77	0.77	20341.34	17.18
SJC2	33317.03	0.41	35211.99	6.12
SJC3a	46395.96	2.30	50590.49	11.54
SJC4a	62701.95	1.24	69283.05	11.87
DONI1	3074.97	1.77	3234.58	7.06
DONI2	6456.14	6.17	6692.71	10.06
DONI3	8911.87	6.81	9797.12	17.42
DONI4	11453.68	6.27	11594.07	7.58
DONI5	11776.59	5.96	11827.69	6.42
	Mean=	3.52	Mean=	10.58
	StdDev=	2.70	StdDev=	4.36

Table 4. Performance of DRG and TCG on CCCP instances when compared to Best-known solutions*: (a) MATLAB, Intel CORE i7 at 2.4 GHz and 8 GB RAM, (b) C++, Intel CORE 2 Duo at 2 GHz with 2 GB RAM.

Instance	DRG (a)	Error (%)	TCG (b)	Error (%)
TA25	1256.62	0.41	1280.49	2.32
TA50	4476.92	0.05	4474.52	0.00
TA60	5356.58	0.00	5357.34	0.01
TA70	6270.45	0.48	6240.67	0.00
TA80	5748.30	4.22	5515.46	0.00
TA90	9069.85	1.92	8899.05	0.00
TA100	8122.36	0.25	8168.36	0.82
SJC1	17492.77	0.77	17375.36	0.09
SJC2	33317.03	0.41	33357.75	0.53
SJC3a	46395.96	2.30	45379.69	0.06
SJC4a	62701.95	1.24	61969.06	0.06
	Mean=	1.10	Mean=	0.35
	StdDev=	1.28	StdDev=	0.71
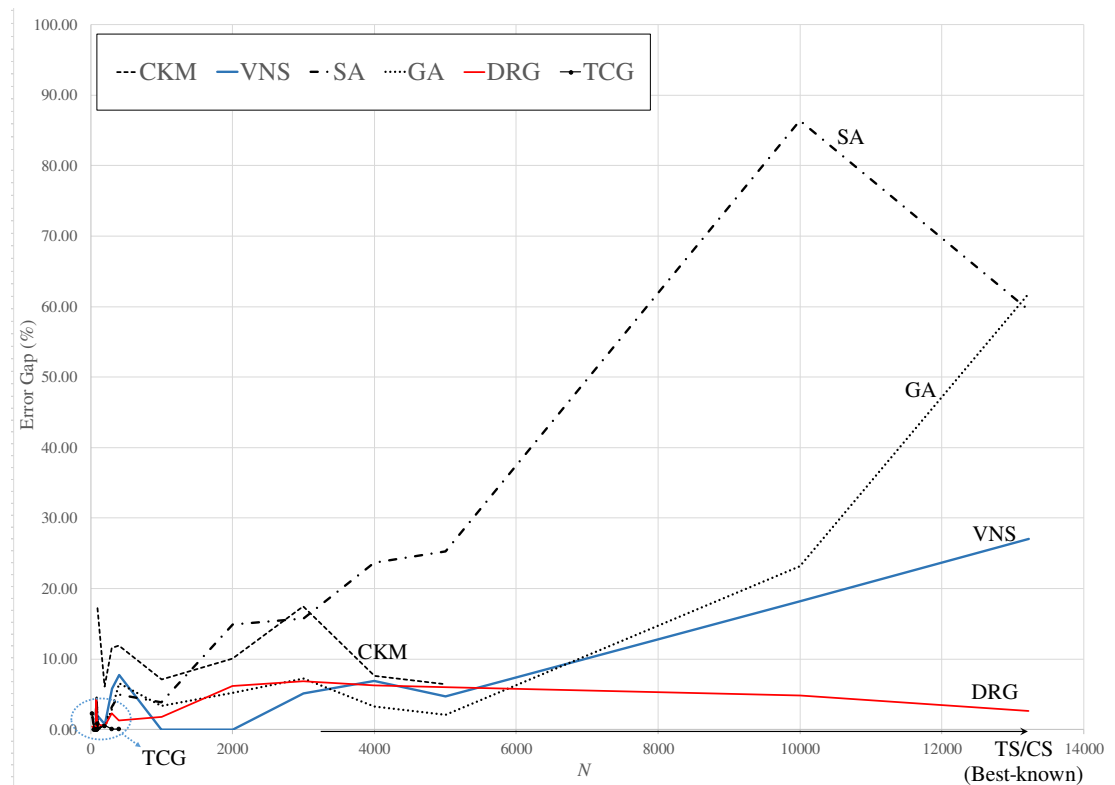
Figure 5. Comparison of Error Gap vs. Size of the Instance.
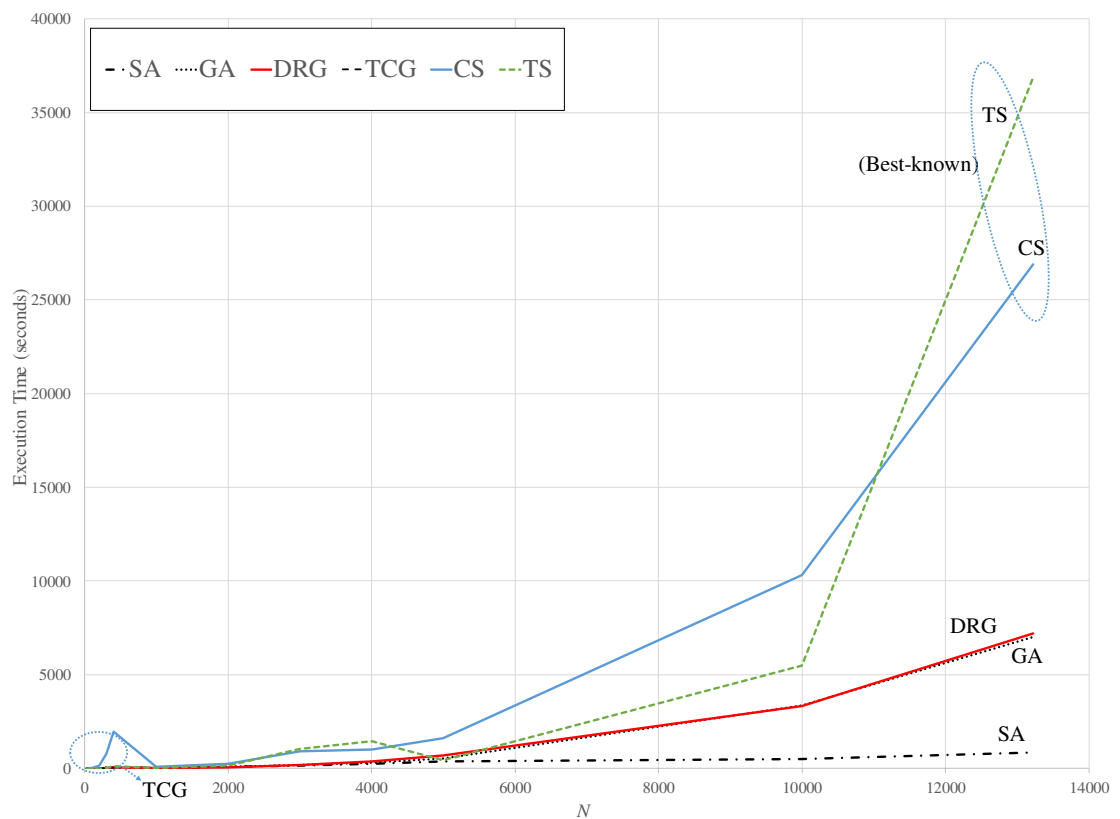
Figure 6. Comparison of Speed vs. Size of the Instance.

CONCLUSIONS AND FUTURE WORK

Both, the application of Gaussian probability functions and the Expectation-Maximization (EM) algorithm have not been explored in the literature as solving techniques for facility location problems (e.g., CCCP). An important aspect for the application of GMMs is the reduction of dispersion to accomplish more efficient clustering and convergence of the EM algorithm. Hence, the proposed DRG meta-heuristic provides important insights about the suitability of these techniques for the CCCP and the challenges to improve its performance.

Regarding performance, TS and CS are the most competitive solving methods for the CCCP and thus, were considered as benchmark methods with mean average error of 0.61% and 0.78% respectively. The proposed DRG meta-heuristic performed as the closest best method with a mean error gap smaller than 2.6% and there is evidence than it can provide these results faster when compared to TS and CS for large instances (> 6000 points). Hence, the DRG meta-heuristic can be a suitable alternative when compared to CS and TS regarding time, and a more efficient method when compared to VNS, SA, GA, and CKM.

The future work is focused on improving the performance of the DRG based on the following key aspects:

- The EM algorithm was found to be functional with up to 30 Gaussian components for the clustering process (see discussion on Section *Dispersion Reduction in GMMs Performance for the CCCP*). In this case the analysis of the Infinite Gaussian Mixture Model described in (Rasmussen, C., 2010) may lead to overcome this restriction.

- While most optimization methods such as those based on Mixed Integer Linear Programming (MILP) or meta-heuristics are purely quantitative, modeling of qualitative criteria may improve the optimization task. In example, in (Hadi-Vencheh, A. and Mohamadghasemi, A., 2013) a methodology that integrated the Analytic Hierarchy Process (AHP) and a Nonlinear Programming Model (NLP) provided very suitable solution for a facility layout problem. Such approach may lead to improve the solving methods for other logistic problems such as the facility location problem.

- Integration with state-of-the-art meta-heuristics such as Migrating Birds Optimization (MBO) (Niroomand, S. et al., 2015) and mat-heuristics (Sartori, C.S. and Buriol, L.S., 2020).

STATEMENT

"The author declares that there is no conflict of interest regarding the publication of this paper."

REFERENCES

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer Science and Business Media, LLC, New York, United States of America.

Carvalho, R., Mendes, G., and Azeredo-Chaves, F. (2017). A heuristic for the capacitated clustering problem. In *In Proc. of the XLIX Simposio Brasileiro de Pesquisa Operacional*, pages 1–12.

Chaves, A.A., Correa, F.A., and Lorena, L.A.N. (2007). Clustering Search Heuristic for the Capacitated *p*-Median Problem. *Springer Advances in Software Computing Series*, 44:136–143.

Chaves, A.A. and Nogueira-Lorena, L.A. (2010). Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, 37:552–558.

Chaves, A.A. and Nogueira-Lorena, L.A. (2011). Hybrid evolutionary algorithm for the Capacitated Centered Clustering Problem. *Expert Systems with Applications*, 38:5013–5018.

Fernandes-Muritiba, A.E., Negreiros, M., Gois-Oria, H.L., and Ferreira de Souza, M. (2012). A Tabu Search Algorithm for the Capacitated Centred Clustering Problem. In *2012 Congreso Latino-Iberoamericano de Investigacion Operativa (CLAIO) - 2012 Simposio Brasileiro de Pesquisa Operacional (SBPO)*, pages 2344–2359.

Forsyth, D. (2012). *Lecture Notes of CS 498 Signals AI: Gaussian Classifiers.* University of Illinois, Urbana-Champaign, IL, United States.

392 Fraley, C. and Raftery, A. (1998). How Many Clusters? Which Clustering Method? Answers
393   Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588.

394 Hadi-Vencheh, A. and Mohamadghasemi, A. (2013). An integrated AHP–NLP methodology for
395   facility layout design. *Journal of Manufacturing Systems*, 32:40–45.

396 Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathe-
397   matical Programming*, 79:191–215.

398 Herda, M. (2015). Combined Genetic Algorithm for Capacitated *p*-Median Problem. In *Proc.
399   of the 16th IEEE International Symposium on Computational Intelligence and Informatics,
400   CINTI 2015*, pages 151–154.

401 Mahmoodi-Darani, N., Ahmadi, V., Saadati Eskandari, Z., and Yousefikhoshbakht, M. (2013).
402   Solving the Capacitated Clustering Problem by a Combined Meta-Heuristic Algorithm. *Journal
403   of Advances in Computer Research*, 4(1):1–13.

404 McLachlan, G.F. and Rathnayake, S. (2014). On the number of components in a Gaussian
405   mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*,
406   4(5):341–355.

407 Mohamadghasemi, A. and Hadi-Vencheh, A. (2012). An integrated synthetic value of fuzzy
408   judgments and nonlinear programming methodology for ranking the facility layout patterns.
409   *Computers & Industrial Engineering*, 62:342–348.

410 Mohammed, M., Badruddin, M., and Mohammed, E. (2016). *Machine Learning: Algorithms and
411   Applications*. CRC Press, Taylor & Francis Group, Milton Park, United Kingdom.

412 Muniz-de-Oliveira, A.C., Chaves, A.A., and Nogueira-Lorena, L.A. (2013). Clustering search.
413   *Pesquisa Operacional*, 33(1):105–121.

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

416 Niroomand, S., Hadi-Vencheh, A., Sahin, R., and Vizvari, B. (2015). Modified migrating birds
417   optimization algorithm for closed loop layout with exact distances in flexible manufacturing
418   systems. *Expert Systems with Applications*, 42(19):6586–6597.

419 Palhano, A., Negreiros, M., and Laporte, G. (2008). A constrained k-median procedure for the
420   capacitated centered clustering problem. In *Proc. of the XIV Congreso Latino Ibero Americano
421   de Investigacion de Operaciones, CLAIO 2008*.

422 Pereira, M.A. and Senne, L.F. (2008). A column generation method for the capacitated centred
423   clustering problem. In *Proc. of the VI ALIO/EURO Workshop on Applied Combinatorial
424   Optimization*, pages 1–6.

425 Pereira, W.M. and Carvalho, R. (2017). ILS-PR: An efficient heuristic for the Capacitated
426   Centered Clustering Problem. *Proceeding Series of the Brazilian Society of Computational
427   and Applied Mathematics*, 5(1):0104271–0104272.

428 Radiah, S., Hasnah, N., and Omar, M. (2013). An Alternative Heuristic for Capacitated *p*-median
429   Problem (CPMP). In *Proc. of the 2013 IEEE Business Engineering and Industrial Applications
430   Colloquium, BEIAC 2013*, pages 916–921.

431 Rasmussen, C. (2010). The Infinite Gaussian Mixture Model. In Solla, S.A., Leen, T.K., and
432   Müller, K.R., editors, *Advances in Neural Information Processing Systems*, volume 12, pages
433   554–560. MIT Press.

434 Sartori, C.S. and Buriol, L.S. (2020). A study on the pickup and delivery problem with time
435   windows: Matheuristics and new instances. *Computers & Operations Research*, 124:1–17.

436 Theodoridis, S. and Koutroumbas, K. (1979). *Pattern Recognition*. Elsevier Academic Press,
437   San Diego, United States of America, second edition.

438 Theodoridis, S. and Koutroumbas, K. (2010). *An Introduction to Pattern Recognition: A
439   MATLAB Approach*. Academic Press: UK.

440 Yousefikhoshbakht, M. and Khorram, E. (2012). Solving the vehicle routing problem by a hybrid
441   meta-heuristic algorithm. *Journal of Industrial Engineering International*, 8(11):1–9.