A heuristic for the capacitated clustering problem

Rodrigo de Carvalho
Centro Universitário Una
Av. João César de Oliveira, 5.775, Beatriz, Contagem, Minas Gerais
carvalho.rodrigo@prof.una.br

Gisele Mendes
Centro Universitário Una
Rua dos Aimorés, 1451, Funcionários, Belo Horizonte, Minas Gerais
gisele.mendes@prof.una.br

Fernanda Azeredo Chaves
Prefeitura Municipal de Belo Horizonte
Av. Afonso Pena, 2236, Funcionários, Belo Horizonte, Minas Gerais
fechaves1000@gmail.com

RESUMO

O problema de agrupamento centrado capacitado (PACC) é uma importante variante do problema p-mediano, amplamente estudado na literatura. O PACC tem sido aplicado a vários problemas reais, tais como definição de áreas de coleta de lixo, planejamento de operações logísticas referentes a programas de combate à dengue, e entrega de jornais. Este artigo apresenta um algoritmo heurístico, utilizando a ideia de diferentes meta-heurísticas para tratar este problema. A heurística proposta superou os resultados da melhor heurística da literatura para um conjunto de instâncias.

PALAVRAS CHAVE. Heurística, Agrupamento, Otimização.

ABSTRACT

The capacitated centered clustering problem (CCCP) is an important variant of the p-median problem, which is widely studied in the literature. The CCCP has been applied to various real problems such as definition of garbage collection areas, planning of logistical operations referring to dengue combat programs and newspaper delivery. This article presents a heuristic algorithm, using the idea of different metaheuristics to deal with this problem. The proposed heuristic outperformed the well known heuristic in the literature for the selected instances.

KEYWORDS. Heuristic, Clustering, Optimization.

1. Introduction

Cluster analysis is an important unsupervised learning task, and aims to find similar structures in a collection of unlabelled data. In general, clustering problems lie in grouping a set of elements in such a way that the elements belonging to the same group have the greatest possible similarity, while the less similar elements are in separate groups.

Clustering techniques have been discussed frequently in the literature for the solution of various problems of practical application in different areas of knowledge, such as software engineering, in order to partition the modular structure of a system Doval et al. [1999]; Kohler et al. [2013]; data mining, applied in order to find groups characterized by similar interests Boginski et al. [2006]; Romanowski et al. [2006]; bioinformatics Kawaji et al. [2004]; Vlasblom e Wodak [2009]; formation of manufacturing cells Trindade e Ochi [2006]; segmentation of images Wu e Leahy [1993]; logistics Negreiros e Palhano [2005]; natural language processing Ushioda e Kawasaki [1996], among others.

The clustering problem addressed in this study, known in the literature as capacitated centered clustering problem, is part of the class of *NP-Hard* problems Pereira e Senne [2008]. For such problems, no deterministic method is known to find the optimal solution with polynomial-time complexity, regardless of the dimension of the problem. Because of this feature, the use of heuristic methods in the solution of this kind of problem is widely used in this area of research. Such methods give up the optimality guarantee in favor of strategies to find suboptimal solutions in a viable time.

Negreiros et al. Negreiros e Palhano [2006] proposed a two-phase heuristic algorithm to solve the CCCP. The first phase uses the Forgy algorithm Forgy [1965] for building an initial solution. In order to improve the performance of this algorithm, it was used a tree-based data structure. The second phase consists of a refinement using a VNS-based heuristic Mladenovic [1995]. Instances related to salesforce problems of a food industry and garbage collection area projects were used to test the performance of this algorithm. The results showed a great performance, where good solutions were achieved at relatively low computational times.

Chaves et al. Chaves e Lorena [2009] presented a procedure for obtaining solutions to the CCCP using Clustering Search (CS) metaheuristic, whose main idea is to identify promising areas of the search space, generating solutions and grouping them so that they can be subsequently explored with local search heuristics. CS showed the best results in most cases in relation to other metaheuristics such as Simulated Annealing and VNS Negreiros e Palhano [2006]. Later, in Chaves e Lorena [2011], the idea of CS was extended by inserting a genetic algorithm in order to increase the exploitation capacity of the search space. The results showed a very satisfactory performance.

An algorithm based on Tabu Search and path-relinking metaheuristics was proposed in Oriá et al. [2012]. This method consisted of two phases. In the first, solutions are created from a constructive heuristic, and then combined from the path-relinking technique. In the second phase, it is applied the Tabu Search method in order to perform a local search. Although this methodology has shown good results, the computational time was higher when compared to other methods of literature.

Muritiba et al. [2012] proposed an effective set of strategies joint with the classical Tabu Search metaheuristic. First, in the construction phase, some points are drawn to make up the initial clusters, later, the remaining points are allocated in the next group so as not to violate,or do the least possible harm, to the grouping capacity. This procedure is repeated 10 times, and the best solution is returned to be used as an initial solution by the local search procedure.Then, a local search is applied using three types of movements, called transfer, swap and wave. The swap movement aims to exchange points of different groups, provided that it does not violate the capacity constraints. Yet the transfer moves the cluster points (one by one), respecting the capacity of the groups. The wave movement exchanges a cluster point if the change results in an improvement in the objective function and the constraint capacity is not respected being applied a correction procedure. This procedure is done recursively until the capacities are no longer violated, being repeated at most 30 times. The results showed that this method was very efficient, with superior results in approximately 80% of instances when compared to other works in literature.

A variation of the CS metaheuristic was applied to the CCCP by Oliveira et al. [2013], the results were compared with two other methods based on CS found in literature, Chaves e Lorena [2011] and Chaves [2009]. The main difference of this work is the replacement of the methods of Simulated Annealing and Genetic Algorithm by the ILS metaheuristic Lourenco et al. [2003]. The results showed that the different approaches of CS achieved similar performance,with a slight advantage for the proposed method.

Maravilha et al. [2014] proposed a variation of the Differential Evolution (DE) algorithm for combinatorial optimization problems. It is worth mentioning that, originally, DE (Storn e Price [1995]) was designed for optimization with variables in the continuum space. The proposed method was applied to the CCCP and did not obtain good results when compared to other heuristics for this problem.

Carvalho et al. [2015] proposed a multi-start heuristic for the CPPP. The solutions were initialized from a method based on greedy randomized search procedure which provided good start solutions. In the performance analysis of the algorithms studied, were used three sets of benchmark problems with different characteristics, and other three new instances. The results founded shows that the proposed algorithm is competitive to other methodologies in literature, however these results are worst than Muritiba et al. [2012].

The method proposed in this paper to deal with the capacitated centered clustering problem consists of ideas of 3 metaheuristics known in the literature, Iterated Local Search (ILS) (Lourenco et al. [2003]), Random Variable Neighborhood Descent (RVND) (Mladenovic e Hansen [1997]) and Greedy Randomized Adaptive Search Procedure (GRASP) (Feo e Resende [1995]). The choice of this methodology is grounded in the good results achieved by this methodology when applied to problems of similar difficulty, as can be seen in several works Martins et al. [2012]; Barbosa et al. [2011]; Neto et al. [2011]; COELHO et al. [2008.]; Silva et al. [2011]; Munhoz et al. [2008]. Moreover, for the intensification phase, it is used the Path-Relinking technique, a generalization of the Scatter Search method Glover [1977].

2. Capacitated Centered Clustering Problem

The CCCP may be defined as follows: N is as set of points distributed in Euclidean space, so that each point i of this set has a demand d_i and a coordinate $\overrightarrow{a_i}$. Considering also p as the number of groups that must be created and c_j as the maximum capacity of each group j. The objective of this problem is to assign each element of the set N to one of the p groups, so that the capacity of the groups is not exceeded and the sum of the distances of the points belonging to each group and its representative is minimized. In this problem,the representative of each group is given by the geometric center. Figure 1 illustrate a solution to this problem, where the numbers indicate the group to which each point is associated,and the central points, in which all other points are attached, the representative of each group. Mathematically, the CCCP may be defined according to Eq. 1-6.

$$min \quad z \quad = \sum_{i=1}^{n} \sum_{j=1}^{p} |\overrightarrow{a_i} - \overrightarrow{q_j}|^2 y_{ij} \tag{1}$$

$$\sum_{j=1}^{p} y_{ij} = 1, \quad \forall\, i = 1, ..., n \tag{2}$$

$$\sum_{i=1}^{n} y_{ij} = n_j \quad \forall\, j = 1, ..., p \tag{3}$$

$$\sum_{i=1}^{n} \overrightarrow{a_i} y_{ij} \leq n_j \overrightarrow{q_j} \quad \forall\, j = 1, ..., p \tag{4}$$

$$\sum_{i=1}^{n} d_i y_{ij} \leq c_j \quad \forall\, j = 1, ..., p \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad n_j \in N, q_j \in \mathbb{R}^d \tag{6}$$

, where $\overrightarrow{q_j}$ is the centroid (geometric center) of group j and y_{ij} a variable that assumes the value 1 when point i is allocated to group j, and the value 0 otherwise. The objective function (1) minimizes the sum of the square of the difference between the coordinates of each point and the centroid of the group to which it belongs. Constraints (2) determine that a point should be exactly associated with one cluster. Constraints (3) and (4) provide the number of points in each cluster and the centroid coordinates, respectively. Constraints (5) determine that the capacity of each group must be respected. The integrity restrictions of the variables are imposed in (6). CCCP is a NP-hard problem and, moreover, the calculation of the objective function is given by a non-linear function, further hindering its solution by exact methods. Therefore, heuristic methods aiming to find effective solutions to this problem are often found in the literature.

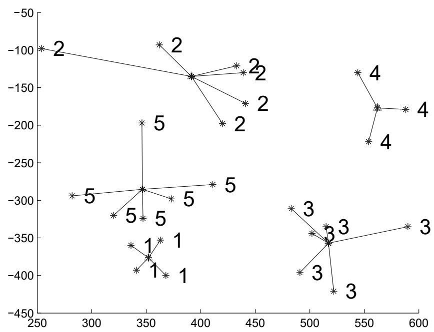

Figure 1: Example of a solution for CCCP

3. Proposed Heuristic

In this section, the proposed heuristic will be shown, called HH-CCCP (Hybrid Heuristic for the Capacitated Centered Clustering Problem).

3.1. Evaluation Function

A solution is evaluated from the sum of the distances of the points to the geometric center of the groups to which they belong. To avoid verifying the feasibility of a solution in relation to the capacity constraint, we worked with the unconstrained problem, inserting this constraint as a penalty term in the evaluation function. In this case, a penalty is applied in infeasible regions, or where the constraint is violated.

The Eq. 7-9 represent the evaluation function $f(y)$ used, where $g(y)$ is the accounting of distances and $h(y)$ is the capacity constraint violation. ϖ is a parameter that penalizes the evaluation function for each unit violated.The value of this penalty was obtained after a series of experiments using all instances of the problem, being set in 1000. It is worth mentioning that for the instances used in this study, this value was enough, but for other instances where the distance between points can overcome 1000, one should use a larger value so that the proposed method does not tend to converge to infeasible solutions.

$$g(y) = \sum_{i=1}^{n} \sum_{j=1}^{p} |\overrightarrow{a_i} - \overrightarrow{q_j}|^2 y_{ij} \tag{7}$$

$$h(y) = \varpi \sum_{j}^{p} \left[\max \left(0, \sum_{i}^{n} d_i y_{ij} - C_j \right) \right] \tag{8}$$

$$f(y) = g(y) + h(y) \tag{9}$$

3.2. Construction Procedure

The construction procedure proposed in this paper can be divided into two stages: The first seeks to create centroids, representatives of the groups, being distributed in the space; the second seeks to assign each point to the centroids created. Fig. 2 illustrates the first stage of this algorithm. Initially, a point $i \in N$ is randomly chosen to be part of the group Q (line 2). The next candidate points to become centroids are those who still do not belong to Q, i.e., $Q' = N \setminus Q$ (line 4). To each point belonging to Q', it is calculated the distance to all Q points, the one with the greatest sum of distances will be chosen to be part of Q (lines 5 and 6). New points are selected using the same logic, until $|Q|$=p.

PROCEDURE Create points
1**BEGIN**
2 $Q \leftarrow \overrightarrow{a_i}, i \in N$
3 DO
4 $Q' \leftarrow N \setminus Q$
5 $i \leftarrow max(\phi(Q, Q'))$
6 $Q \leftarrow \overrightarrow{a_i} \cup Q, i \in N$
7 WHILE $|Q| < p$
8**END**

Figure 2: Algorithm ILS-RVND.

In the second stage, each point $i \in N$ is assigned to the nearest centroid, one by one. In this step, the capacity of each group is not taken into consideration. After the allocation, in case the group's capacity has been exceeded, the point which is farthest from the centroid of this group is reallocated to another group. In this reallocation, all possibilities are considered, i.e., to the available Q groups. The group chosen will be the one contained in the RLC (Restrict List of Candidates), meeting the following condition $[f_{min}, f_{min} + \alpha(f_{max} - f_{min})]$. Where f_{max} and f_{min} are, respectively, the element of greater and the element of lower increase in the evaluation function of the solution and is an input parameter that can vary between 0 and 1. The closer to 1 is the α value more random are the solutions generated. A RLC element is chosen at random, characterizing the method as randomic. This random choice allows this procedure to be used several times to generate different solutions. In case the capacity of the group that received this new point has also been exceeded, again the farthest point will be reallocated using the same logic described previously. This happens until a viable solution is reached, or until the number of reallocations is equal to 20.

3.3. Local Search

When it comes to the CCCP, a wide variety of neighborhood structures can be exploited. The local search used in this work makes use of three different neighborhoods: Swap, Transfer and Wave. These structures are described below: Swap consists of exchanging two points from their groups; Transfer aims to reallocate a point in a group; Wave consists of finding the farthest point of a group j, and reallocating it in the nearest group k. If the value of the evaluation function does not improve, a new attempt to reallocate that point is performed, where the farthest point of the group k is reallocated to the nearest group l. This procedure is repeated for a maximum of 30 times, or until the evaluation function improves.

The solutions that are generated by these neighborhood structures can be exploited in two ways, known as: First improving and Best improving. In the first, the solutions are checked until it is found one that improves the current solution. In the second, Best improving, all solutions generated by a neighborhood structure are checked, returning the best among them. In this work, it was used the First-improving strategy, since it presented the best results in preliminary tests.

Moreover, in order to make the local search more efficient, only part of the neighborhood is exploited. A movement is performed only between points/groups who are close, avoiding unnecessary exchanges. For example, by applying the Swap neighborhood structure, the points considered as candidates are only those where the distance is smaller than a radius r.

The local search used is a variant of the VND procedure (Mladenovic e Hansen [1997]), known as RVND. The RVND consists in systematically explore different neighborhood sequences of a solution, using a descent local search in order to reduce the risk of becoming trapped in local optimal. Furthermore, it is also possible to exploit different areas in the search space, since the order of the neighborhood structures can lead to different solutions.

3.4. Perturbation Procedures

To allow the proposed heuristic to escape of local optimum, three perturbation strategies were proposed in this paper. The first, called P_1, consists of randomly deallocate a predetermined number of points and allocate them again in the nearest clusters that have capacity. If the allocation is not possible due to lack of cluster capacity, only the distance is considered as allocation criterion. When deallocated the points of the clusters, the centroids should be corrected, however, in the stage of allocation, the centroids will only be updated after all points are allocated. The predetermined number of points is set in proportion to the size of the instance, in this work, it was used 20% of the points N, value obtained from empirical tests.

The second strategy, called P_2, differs from the first by the fact that the deallocation/ allocation is made entirely at random. This method is used only when the algorithm stagnates in a local optimum for a few iterations, as it allows the metaheuristic to search for further areas of the search space. The number of points used in this disturbance is equal to 10% of the points.

Finally, the third strategy, named here P_3, deallocates 50% of the farthest points of each centroid. Subsequently, these points are entered in the nearest groups. The centroids are corrected in two moments: first, after the deallocation process of the points; then, after the allocation of all points.

At each iteration of the proposed algorithm, only one of the disturbances is applied, chosen according to the number of iterations without improvement (b) of the heuristic. Two thresholds, l_1 and l_2, are used, thus, until the threshold l_1 is reached($b < l_1$), the strategy P_1 is used. Reaching l_1, strategy P_2 is used until l_2 is reached ($l_1 < b < l_2$). When $b > l_2$, the strategy P_3 is used, subsequently, the number of iterations without improvement b is reset. The values of l_1 and l_2 were set, respectively, at 10 and 25.

3.5. Path-Relinking

Path-Relinking consists of generating all intermediate (nearby) solutions between an initial solution and a final solution. The basic principle of this technique is that between two quality solutions, there can be a third and better one. The solutions are chosen from a set of solutions called H elite solutions, with maximum size h. Initially, this set is empty, and at each iteration of the proposed heuristic, the current solution is considered a candidate to compose H. The updating of the set of H elite solutions is as follows: If $H < h$, i.e., is not yet full, the candidate solutions added. Otherwise, two aspects are considered for the candidate solution to replace an H solution: the quality of any candidate solution must be better than the worst solution contained in H and must be different from the solutions contained in this set. The difference is calculated from the coordinates of the centroids of each group. For example, consider Q_1 and Q_2 as the centroid of two different solutions s_1 and s_2, where $Q_1 = \{(1,2),(3,1),(5,4)\}$ e $Q_2 = \{(1,2),(2,2),(3,4)\}$. In this case,the difference between these solutions is equal to 2, because they have two different centroids.

Once defined the initial s_i and target s_f solutions, at each iteration, the points of the initial solution are moved, one by one, to other groups, according to the configuration of the target solution. At the end of the iteration, the best movement is applied even if s_i does not improve. The method ends when s_i and s_f are equal. Fig. 3 illustrates the ideia of this method; consider the bold arrows as the best movement of a given iteration, represented here as a figure line. Furthermore, the solution scheme shown in this figure is known as group-number, in which an integer n-vector is used to represent a group of n elements. In this vector, the i-th position of the vector indicates the cluster number of the i-th element.

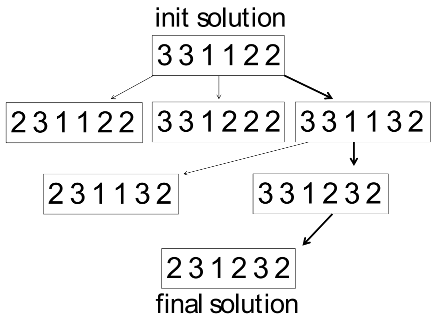

Figure 3: Example of Path-Relinking

Before applying this technique, it is necessary to use a procedure for normalization of labels of clusters in the solutions s_i and s_f. As the clusters, a priori, are not labeled, similar clusters may have different labels. Fig. 1 and 4 illustrates this case, where clusters with different labels, 3 in Fig. 1 and 4 in Fig. 4, are equal. To solve this problem, one must verify the similarity of groups, and give the same label on both solutions (s_i and s_f).

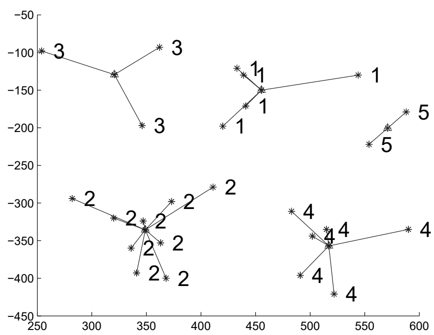

Figure 4: Example of a solution for CCCP (b)

The details of normalization is given by Carvalho e Pereira [2017]. The similarity among centroids of different solutions guide the process of the normalization. In a first step, the distances among centroids of different solutions, s_f and s_i, are calculated and stored in a vector v. The formula known as Euclidean distance is used for calculate the distance. Moreover, the following informations are stored in v: cluster of solution s_f; cluster of solution s_i; and the distance between them. In the second step, the same label is given for clusters based on distances, i.e., nearby groups receive the same label. The result obtained by the normalization process is shown in Figure 5.

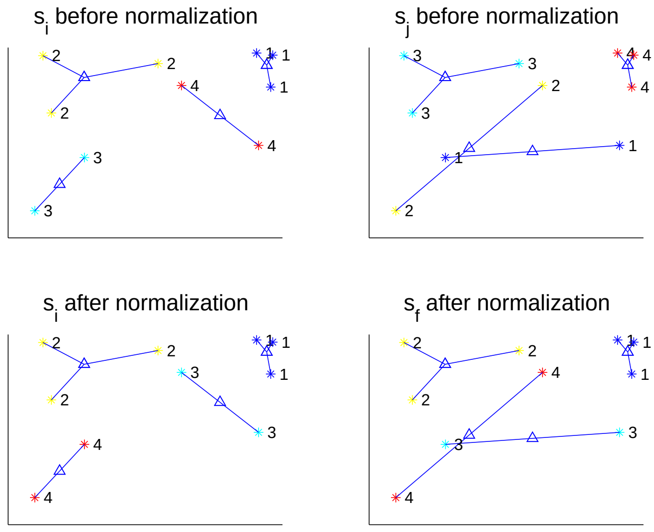

Figure 5: Result after application of the normalization procedure.

3.6. Heuristic

The Fig. 6 outlines the heuristic proposed in this paper. It is fundamentally composed of 3 phases: initialization, local search (intensification) and perturbation (diversification). In the initialization phase (lines 1 to 2), a solution is created and the List of Candidates for Path-Relink procedure is initialized. The lines 5 to 8 of the algorithm summarize the intensification process, where RVND and Path-Relink procedures are used. Finally, the diversification phase, represented in the line 10 of the algorithm, contains the perturbation procedure. The diversification and intensification procedures are run until a stop criterion is satisfied, in this paper was used the run-time limit.

PROCEDURE HH-CCCP
1**BEGIN**
2 Initialize H
3 $s \leftarrow$ Construction procedure(α)
4 DO
5 $s \leftarrow$ RVND(s)
6 Update H
7 Update(s, s^*)
8 Path-Relink(H)
9 $s \leftarrow s^*$
10 $s \leftarrow$ Pertubation(s)
11 WHILE $Not\ stopping\ criterion$
12**END**

Figure 6: Algorithm HH-CCCP.

4. Computational Experiments

The performance between HH-CCCP procedure and the well known literature heuristic Muritiba et al. [2012] are compared in this paper to the benchmark instances and a set of instances proposed here. The sizes of benchkmark instances has between 100 and 13221 nodes, and the proposed instances has between 1000 and 10000. Fig. 7 illustrates a sample of the benchmark instances and proposed instances. Note that the distribution of points (last three subfigures) is different from those proposed in the literature. In addition, the distribution of demands is made unequally and the relationship between demand/capacity is tight. For example, the demand standard deviation of a given instance of the literature is 62.41 and the ratio capacity/demand of groups equal to 2.1. The proposed instances in this paper has an average standard deviation set to 100 and the ratio capacity / demand equal to 1.2, it makes the problem even more difficult because it increases the number of solutions unfeasible. Although the complexity of the problem does not change, the heuristics techniques used to solve the problem may have convergence difficulties.

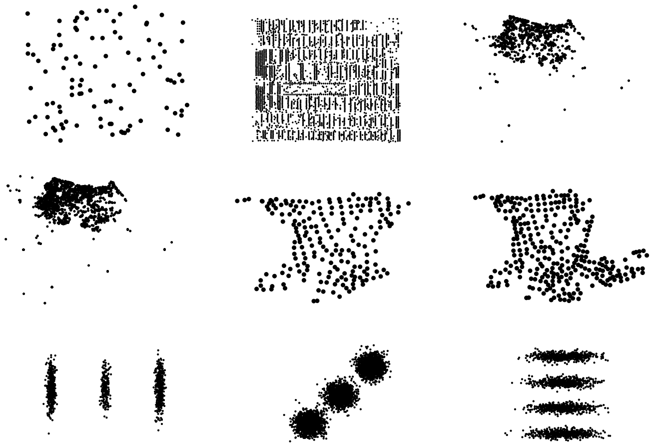

Figure 7: The images of examples extracted from the benchmark and proposed instances.

Each algorithm was run 10 times to all instances using distinct random seeds, but utilizing the same seed in each run. All computational tests were carried out on a computer with a Intel i7 processor at 2.0 GHz and 6 GB of RAM, running the Ubuntu operating system. Further, all algorithms were implemented in Matlab. The parameter value of the heuristic was set to: $\alpha = 0.2$. The stopping criterion for proposed method was set to $0.3|N|$ seconds, the Tabu Search parameters was maintained as in original paper.

Initially, the following indicators were computed to evaluate the performance of algorithms: DevMed, DevMin and #Best. Given BestValue was the best solution obtained among the considered algorithms for a given instance. For each method, DevMin and DevMed represents the average minimum and the average mean of the deviation between the best solution attained by the algorithm and the BestValue of each instance, respectively. Low values for DevMin indicate that the algorithm in the best case obtained close results to BestValue. As well as low values for DevMed indicate that the algorithm was able to find solutions with close quality of BestValue, considering all executions in all instances, demonstrating consistency. Further, #Best represents the number of times in which the algorithm returns to the BestValue. The results achieved by heuristics can be seen in Table 1.

Table 1: Compararison between HH-PACC and Tabu Search Heuristic.

Indicators	HH-PACC	Tabu SearchMuritiba et al. [2012]
#Best	378	321
DevMed	0.005	0.002
DevMin	0.002	0.001

The HH-CCCP outperforms the well known heuristic of the literature for the indicator used #Best. Although these results provide a good performance indicator, they cannot be used to acquire more general conclusions. The performance evaluation was performed using Wilcoxon statistical test. The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when comparing two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ (i.e. it is a paired difference test). Statistical tests are used to determine whether the observed differences are real or are due to chance. It was found that there is a significant difference among them, and HH-CCCP outperforms the heuristic studied.

In order to understand more about the behavior of the algorithm proposed and the well known heuristic of the literature (Tabu Search) Muritiba et al. [2012], their performance to find a target solution is studied. For this purpose, it was used Time-to-target (TTT) plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. The foundations of the construction of TTT plots, together with their interpretation and applications, can be found in Alex et al. [2007]. Fig. 8 shows the results obtained, the probability that heuristic proposed finds a target solution in a smaller computation time than algorithm Tabu Search is 87% for the instance SJC4b. The best solution found by Tabu Search for this instance is given as the target solution.

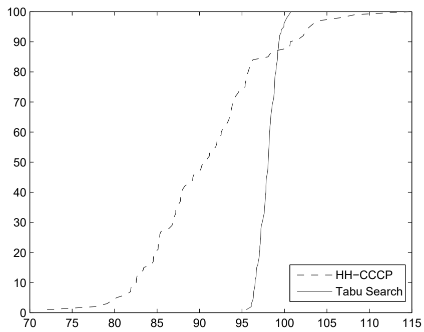

Figure 8: TTT plots

5. Conclusion

The objective of this study was to propose a competitive heuristic with respect to computational time and solution quality when compared with well known heuristic available in the literature for CCCP.

The construction phase based on GRASP was combined with the variations of three different structures of neighborhood guided by RVND procedure, three perturbation mechanism and path-relink. These combinations outperformed the well known heuristic of the literature, regarding running time and solution quality for the instances used.

It is worth of remarking that Path Relink is a very effective strategy for improving the solutions of the implemented heuristic. The attained results were only possible with the deployment of this strategies in terms of both computational time and solution quality.

In future work we intend to analyze the performance of the proposed algorithm applied to a real Health Agents problem. We aim will be determine the coverage area of a health agent and obtained the minimum length route for the attendance of all patient. Futhermore, one can investigate the adaptation of the heuristic proposed for other clustering problems.

References

Alex, R. M., Resende, M. G. C., e Ribeiro, C. C. (2007). TTT plots: a perl program to create time-to-target plots. *Optimization Letters*, 1:355–366.

Barbosa, H. S. C., de Souza, S., e Souza, M. J. F. (2011). Programação de cursos universitários baseada em currículos via uma meta-heurística híbrida grasp-ils utilizando o procedimento kempeswap. *Simpósio de Pesquisa Operacional & Logistica da Marinha*.

Boginski, V., Buteko, S., e Pardalos, P. M. (2006). Mining market data: A network approach. *Computer & Operations Research*, 33:3171–3184.

Carvalho, R. e Pereira, W. M. (2017). Normalization procedure for cccp. Código. URL `www.rclink.com.br/hub`.

Carvalho, R., Saldanha, R., e Martins (2015). Heurística aplicada ao problema de agrupamento centrado capacitado. *XLVII Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas*.

Chaves, A. e Lorena, A. (2011). Hybrid evolunionary algorithm for the capacitated centered clustering problem. *Expert Systems with Applications*, 5:5013–5018.

Chaves, A. A. (2009). Uma metaherística híbrida com busca por agrupamentos aplicada a problemas de otimização combinatória. Master's thesis, Instituto Nacional de Pesquisa Espaciais.

Chaves, A. A. e Lorena, L. A. N. (2009). Clustering search algorithm for the capacitated centred clustering problem. Technical report, LAC, INPE.

COELHO, I. M., RIBAS, S., e SOUZA, M. J. F. . (2008.). Um método híbrido baseado em grasp, ils e vnd para o planejamento operacional de lavra em minas a céu aberto. In *Anais do XVI Seminário de Iniciação Ciêntífica da UFOP*.

Doval, D., Mancoridis, S., e Mitchell, B. S. (1999). Automatic clustering of software system using a genetic algorithm. *Proc. of the Conf. on Software Tool and Enginnering Practice*, 1:72–91.

Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptative search procedures. *J. of Global Optimization*, 6:109–133.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 3:768.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Science*, 8:156–166.

Kawaji, H., Tarenaka, Y., e Matsuda, H. (2004). Graph-based clustering for finding distant relationships in a large set of protein sequences. *Bioinformatics*, 20:243–252.

Kohler, V. C., Fampa, M., e Araujo, O. (2013). Mixed-integer linear programming formulations for the software clustering problem. *Computational Optimization and Applications*, 55:113–135.

Lourenco, H. R., Martin, O., e Stutzle, T. (2003). *Iterated Local Search*. Handbook of Metaheuristics.

Maravilha, A. L., Ramirez, J. A., e Campelo, F. (2014). Combinatorial optimization with differential evolution: a set-based approach. *Proc. of the 2014 Conference on Genetic and Evolutionary Computation*, p. 69–70.

Martins, A. X., Duhamel, C., Souza, M. C., Saldanha, R., e Mahey, P. (2012). A vnd-ils heuristic to solve the rwa problem. *Lecture Notes in Computer Science*, 6701:577–582.

Mladenovic, N. (1995). A variable neighborhood algorithm - a new metaheuristics for optimization combinatorial. *Abstracts of Papers at Optimization Days*.

Mladenovic, N. e Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100.

Munhoz, P. L. A., Perché, M. H. P., e Souza (2008). Um algoritmo baseado em grasp, iterated local search, descida em vizinhança variável e reconexão por caminhos para o problema de sequenciamento em uma máquina com penalidades por antecipação e atraso. In *Anais do XI EMC. Volta Redonda (RJ) : Escola de Engenharia Industrial Metalúrgica de Volta Redonda/UFF,*.

Muritiba, A. E. F., Negreiros, M., Oriá, H. L. G., e Souza, M. F. (2012). A tabu search algorithm for the capacitated centred clustering problem. *Congresso Latino-Iberoamericano de Investigación Operativa*, 1.

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

Negreiros, M. J. G. e Palhano, I. C. V., A. W. C.and Silva (2005). Uma avaliação de modelos de cadeias logísticas de entrega de jornais e assiinantes e coleta de leite a granel. *Revista Produção*, 5:1876–1901.

Neto, A. J. S., Vianna, D., e Vianna, M. F. D. (2011). Heuristica grasp e ils para o planejamento de estocagem de contêineres em navios porta-contêineres. *Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional*.

Oliveira, A. C. M., Chaves, A. A., e Lorena, L. A. N. (2013). Clustering search. *Pesquisa Operacional*, 33:105–121.

Oriá, H. L. G., Negreiros, M., e Muritiba, A. E. F. (2012). Um algoritmo path relinking com busca tabu para o problema de agrupamento capacitado em centro geométrico. *Claio SBPO, Simpósio de Pesquisa Operacional, Rio de Janeiro*.

Pereira, M. A. e Senne, E. L. F. (2008). A column generation method for the capacitated centred clustering problem. *Alio/Euro workshop on applied combinatorial*, 1:1–6.

Romanowski, C. J., Nagi, R., e Sudit, M. (2006). Data mining in an engineering design environment: Or applications from graph matching. *Computer & Operations Research*, 33:3150–3160.

Silva, M. M., Subramanian, A., e Ochi, L. S. (2011). Uma heurística baseada em grasp e ils para o problema da mínima latência. In *Anais do XLIII Simpósio de Pesquisa Operacional*.

Storn, R. e Price, K. (1995). Diferential evolution - a simple e efficient adaptative schema global optimzation over continuos spaces. *Technical Report TR-95-012, ICSI*.

Trindade, A. R. e Ochi, L. S. (2006). Um algoritmo evoluttivo híbrido para a formação de células de manufaturas em sistemas de produção. *Pesquisa Operacional*, 26:255–294.

Ushioda, A. e Kawasaki, J. (1996). Hierarchical clustering of words and application to nlp tasks. *Fourth Workshop on Very Large Corpora, New Jersey, Association for com Computational Linguistics*, 1:28–41.

Vlasblom, J. e Wodak, S. (2009). Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics*, 10:1–14.

Wu, Z. e Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation. *IEE Transactions on Pattern Analysis and Machine intelligence*, 15:1101–113.