Capacitated Clustering Problem

Ahmad Sidani
Politecnico di Torino
S312919
ahmad.sidani@studenti.polto.it

Ali Yassine
Politecnico di Torino
S312920
ali.yassine@studenti.polto.it

Hadi Ibrahim
Politecnico di Torino
S313385
hadi.ibrahim@studenti.polto.it

***Abstract*—In this study, the Capacitated Clustering Problem (CCP) is addressed using mathematical programming and a heuristic approach. Node position, weight, distance, and cluster capacity are used to represent the problem. We will create solutions using Gurobi [1] and a customized heuristic algorithm then compare their outcomes in terms of processing time and objective function.**

I. Introduction

The Capacitated Clustering Problem (CCP) is an essential optimization problem that is commonly used in many domains, including waste collection zones, salesman regions, and hospital location. The CCP's goal is to partition a collection of nodes into numerous disjoint clusters so that the sum of node weights in each cluster fulfills a specified capacity limit. The topic has several real-world applications and is extremely important in operations research and management science.

The CCP has been described by mathematical model , where the objective function seeks to reduce the distance of each node from the center of the cluster plus a capacity that relies on the amount of weight not satisfied. The model takes into account a variety of characteristics, including node location, weight, distance between nodes, and cluster capacity.

The purpose of this study is to address the CCP utilizing mathematical programming and a heuristic approach. To find the best solution, we will use Gurobi optimization software, a cutting-edge mathematical programming solver, and a customized k-means algorithm [8]. The results of both techniques will be compared on the basis of computing time and objective function, and the stability and durability of the solutions will be assessed using in-sample [6] and out-of-sample [5] analyses, as well as the Value of Stochastic Solution (VSS) method [2] [3].

In this paper we will detail the methodology used to solve the CCP , including the processes used to generate instances of the issue and random weight realizations. In addition, we will offer a full explanation of the findings gained from executing the scripts, as well as a comparison of the solutions found using the Gurobi optimization software and the customized heuristic technique. Finally, we will end the report by summarizing the findings and commenting on the level of satisfaction with the results.

II. Mathematical Model

A. Parameters

The mathematical model describing the problem consider the following parameters:

- d_{ij} is the distance between point i and j,
- C_j is the capacity of cluster i,
- $w_i(\omega)$ is the weight of point i (it is a random variable),

B. Decision Variables

The mathematical model describing the problem consider the following decision variables:

- x_{ij} is 1, if point i is assigned to cluster j, and 0 otherwise.
- y_{ij} is 1, if node i is the center of the cluster and 0 otherwise.

C. Objective Function

The objective function seeks to minimize the distance between each point and the cluster's center, plus a number determined by the amount of weight not reached. The parameter λ weights the objective function's two terms. It is described by the

$$\min \sum_{i\in I}\sum_{j\in I} d_{ij}x_{ij} + \lambda \mathbb{E}\left[\sum_{j\in I}\left[C_j y_j - \sum_{i\in I}\omega_i(\omega)\, x_{ij}\right]^+\right]$$

D. Constraints

A: Each point must be assigned

$$\sum_{j\in I} x_{ij} = 1, \quad \forall i \in I$$

B: The maximum number of clusters is p.

$$\sum_{j\in I} y_j \leq p$$

C: Force x_{ij} to be one, only if point j is a cluster center

$$x_{ij} \leq y_j, \quad \forall i \in I, \quad \forall j \in J$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in I, \quad \forall j \in J$$

III. Methodology

A. Problem Instance and Weight Realization Generation

1) Generation of the Instance: The placements of the nodes and the capacities of the clusters are generated. The node placements are created by a 2D random uniform distribution, while the cluster capacity are generated by a normal distribution with a defined mean and standard deviation.

2) Generation of Weights: In this problem, the creation of random weights is also significant. The node weights are calculated using a normal distribution with a given mean and standard deviation.

B. Solving the Model

This part involves employing Gurobi and a heuristic approach to solve the problem.

1) Model using Gurobi: The Gurobi approach is utilized to tackle the optimization problem. The optimization challenge entails reducing the Euclidean distance between nodes and capacities while also taking weights and the lambda parameter into consideration. The Gurobi library is used to represent the optimization issue by establishing decision variables, defining the objective function, and adding constraints to the model. The Gurobi library is used to address the optimization problem by optimizing the objective function once the model has been specified. The solution to the optimization issue is obtained by extracting the values of the decision variables and grouping the values into clusters. After that, the result is returned, together with the time it took to solve the issue and the optimal objective value.

2) Model using Heuristic Algorithm: The customized heuristic method applied is k-means [8] with capacity constraints where each cluster is allocated a maximum capacity, and the algorithm seeks to find a solution that fulfills these requirements. The aim is to reduce the sum of squared distances (euclidean distance) between data points and their nearest centroid while meeting capacity restrictions.

IV. Evaluation Methods

A. In Sample Stability

In sample stability [6] is a metric employed to assess the reliability and consistency of the clustering results. The clustering technique is repeated several times on the same dataset in order to assess the in sample stability. The similarity between the clustering outcomes is then calculated. The Jaccard similarity [4] is a metric for comparing the similarity of two sets of data. It is determined by dividing the size of the sets' union by the size of their intersection.

B. Out Sample stability

The model's out sample stability [5] is assessed by computing the Rand Index [7] between cluster assignments achieved on a training dataset and those obtained on a validation dataset. The Rand Index compares the similarity of two sets of cluster assignments, with 1 representing perfect agreement and 0 representing complete disagreement. Out sample stability is determined by repeating this technique several times and then averaging the Rand Index scores over all runs. This provides a measure of the consistency of cluster designations obtained on new, previously unseen data.

C. Value of Stochastic Solution

The Value of Stochastic Solution (VSS) [2] [3] is a metric that calculates the difference between the expected value of the stochastic solution (EEVS) and the ideal value of the recourse problem (RP). The VSS is calculated by generating several instances of the problem then using the CCP model to determine the expected value of the solution. After that, the RP is computed by solving the issue for a single instance. The VSS is calculated as the difference between the EEVS and the RP. This metric provides a measure of the value of having a recourse option available in the event of an unexpected event or change in the problem conditions.

V. Results

A. Results from Gurobi

1) Random Data Results: Gurobi [1] requires that we first specify the problem, establish the parameters, and then execute the solver in order to solve an optimization problem. For example if you want to cluster a dataset of 10,000 nodes into 20 clusters. Gurobi can be used to determine the optimal solution by minimizing the objective function within the constraints of the clustering problem. The size of the dataset and the complexity of the problem will affect the duration needed to find the optimal solution, however in our example, it takes 36 roughly seconds. We can investigate the structure of the dataset and get insight into the problem by looking at the number of clusters and the number of nodes in each cluster that are returned in the model's output. Figure 1 shows the clustered nodes using the CCP-Gurobi algorithm, where the nodes have been divided into 20 distinct clusters based on their similarity.

The nodes in Figure 1 have been clustered using the CCP-Gurobi algorithm hence, split into different clusters based on objective and constrained mentioned before.

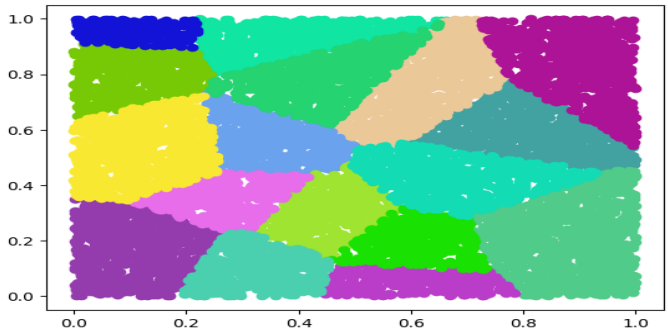

Fig. 1. Clustered Nodes using CCP-Gurobi Algorithm

Below in Figure 2, we provide a simpler example containing only 200 nodes to better visualize and understand how the algorithm partitions the nodes into clusters.

2) In Sample Stability: The results were found to be very similar, with a mean Jaccard Similarity Measure of [1.0].

3) Out of Sample Stability: The results demonstrated that the model was capable of producing stable solutions with a Rand Index score of 0.936.

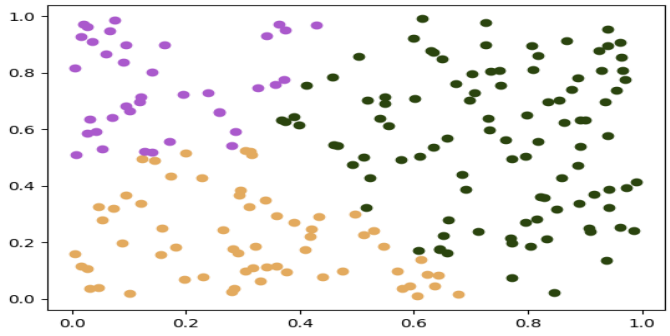

Fig. 2. Clustered Nodes using CCP-Gurobi Algorithm with Reduced Dataset

4) Value of Stochastic Solution: The result obtained from the VSS that is close to zero (-150).

B. Results from Heuristic Algorithm

1) Random Sample Results: The term "K Means with capacity constraints" (KMCC) refers to a variant of the standard K Means that includes extra constraints relating to cluster capacities. For demonstration, we want to employ KMCC to divide a dataset of 10,000 nodes into 20 clusters. By defining the nodes, capacities, and weights of the algorithm on the available dataset, we can accomplish this. The algorithm assigns each node in a cluster based on its distance from the cluster's centroid, which was generated at random while taking weights and capacities into account. Similar to CCP, the size of the dataset and the complexity of the problem affect how long the method takes to run. The KMCC algorithm is executed on the dataset in our case in roughly 120 seconds. When the process is complete, we can observe how many clusters there are in total and how many nodes in each cluster. We may use this information to deepen our analysis and to better comprehend the dataset's structure.

The nodes in Figure 3 have been clustered using the KMCC algorithm hence, split into different clusters based on objective and constrained.

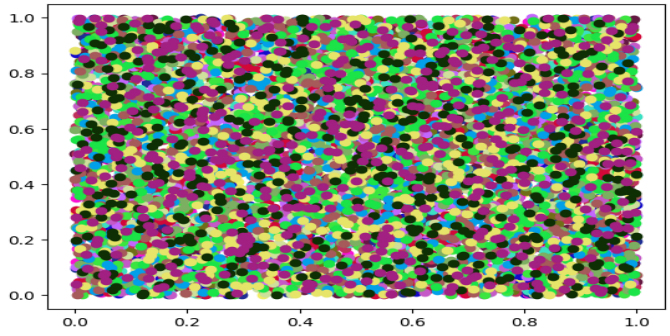

Fig. 3. Clustered Nodes using KMCC Algorithm

Below in Figure 4, we provide a simpler example containing only 200 nodes to better visualize and understand how the algorithm partitions the nodes into clusters.

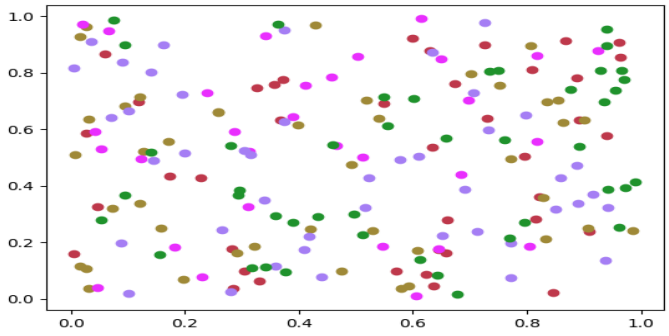

Fig. 4. Clustered Nodes using KMCC Algorithm with Reduced Dataset

2) In Sample Stability: The results obtained were underwhelming, with a mean Jaccard Similarity Measure of 0.036.

3) Out of Sample Stability: The model found consistent answers, as seen by the high Rand Index score of 1.0.

C. Comparison of Results

The Jaccard Similarity Measure [4] was used to assess in-sample stability, and it was discovered that gurobi had a mean Jaccard Similarity Measure of 1.0, which was much greater than that of k-means 0.036.

Out-of-sample stability was tested using the Rand Index, and gurobi scored 0.936 which is almost the same in k-means, with a Rand Index score of 1.0.

In terms of computing time, gurobi produced the results in 36 seconds, whereas k-means produced the findings in 120s.

In terms of the objective function, gurobi was able to generate solutions with a lower object value, -3800, than k-means 1662.

The comparison of results between the gurobi method and the customized k-means method is shown in Table I.

	Gurobi	**Customized K-Means**
In Sample Stability	1.0	0.036
Out Sample Stability	0.936	1.0
VSS	-150	N/A
Time	36s	120s
Objective Function	-3800	1662

TABLE I
COMPARISON OF THE EVALUATION METHODS ON GUROBI AND CUSTOMIZED K-MEANS

VI. DISCUSSION

The project findings demonstrate that Gurobi outperformed the customised k-means approach in terms of in sample stability, computation time, and objective function. Gurobi's in sample stability score suggested that it was capable of delivering consistent solutions inside the sample data. The customised k-means approach, on the other hand, did not produce consistent findings, implying a lack of consistency in the responses provided. In terms of the out sample stability, both methods scored well, indicating that the clustering solution is extremely stable and resilient, since it gives the same results regardless when the data or clustering methods are altered.

Gurobi was able to provide the findings significantly faster than the k-means approach in terms of computation time. Gurobi was also able to deliver more optimal solutions with a lower objective function value, according to the results. This implies that Gurobi was able to deal with the unpredictability of the data while maintaining the quality of the answers, something that k-means was unable to achieve.

A VSS of almost zero indicates that the RP is almost the same, but slightly minimized value compared to the EEV. This result suggests that the clustering solution is efficient and optimized, as it provides the minimum value required to solve the problem while still meeting the constraints. The VSS of zero also suggests that the expected cost of the solution is minimized, which is essential in capacitated clustering problems where resources are limited.

The project demonstrates Gurobi's efficacy in addressing the Capacitated Clustering Problem with stochastic optimization. The findings show that the model can tolerate unpredictability in data and produce stable and optimal answers. The study sheds light on the use of stochastic optimization in tackling real-world situations.

However, there is still potential for development in terms of the modified k-means method's stability and computing time. Future study might concentrate on designing and optimizing heuristic methods for this problem in order to attain both stability and computing speed. Furthermore, future research might look at the usage of additional optimization approaches to tackle the capacitated clustering problem, such as genetic algorithms or simulated annealing. Future enhancements would not only increase the accuracy and stability of the results, but would also make the technique more resilient and efficient.

REFERENCES

[1] Gurobi Optimization. Gurobi Optimizer. https://www.gurobi.com/.

[2] Birge, J. R., and Louveaux, F. (2011). "Introduction to stochastic programming." Heidelberg, Germany: Springer Science Business Media.

[3] Li C and Grossmann IE (2021) "A Review of Stochastic Programming Methods for Optimization of Process Systems Under Uncertainty." Front. Chem. Eng. 2:622241. doi: 10.3389/fceng.2020.622241

[4] Alessandro Tufano, Riccardo Accorsi, Riccardo Manzini, "Machine learning methods to improve the operations of 3PL logistics, Procedia Manufacturing", Volume 42, https://doi.org/10.1016/j.promfg.2020.02.023.

[5] Leonard J. Tashman,"Out-of-sample tests of forecasting accuracy: an analysis and review", International Journal of Forecasting, Volume 16, Issue 4, 2000, https://doi.org/10.1016/S0169-2070(00)00065-0.

[6] Stephan M. Bischofberger, Munir Hiabu, Enno Mammen, Jens Perch Nielsen, "A comparison of in-sample forecasting methods", Computational Statistics Data Analysis, Volume 137, 2019, https://doi.org/10.1016/j.csda.2019.02.009.

[7] Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods". Journal of the American Statistical Association, 66(336), 846-850

[8] Garassini, L. and Marzano, A. (2010). "K-means with capacitated constraints: An experimental comparison with a modified GRASP algorithm." European Journal of Operational Research, 201(3), 767-773. doi: 10.1016/j.ejor.2009.08.007