A new approach to clustering service orders in electric utilities

Vinícius Jacques Garcia*, Daniel Pinheiro Bernardon*,
Maurício Sperandio*

* Campus Alegrete
Universidade Federal do Pampa - UNIPAMPA
Av. Tiaraju 810 - 97546-550
Alegrete - RS - Brasil
{viniciusgarcia,danielbernardon,mauriciosperandio}@unipampa.edu.br

***Abstract*—Managing the staff to carry out service orders related to electric power distribution systems usually requires high available maintenance teams to execute those typical procedures, some of them with strict defined deadlines established by agency regulators. Additionally, this amount of orders often demand even more resources than those available on the utilities. In this scenario, there is a promise scope to be explored in order to improve productivity of teams in the sense that the time spent in the route is reduced. The methodology proposed in this paper defined this problem as the well-known capacitated clustering problem and includes a heuristic algorithm to cluster even hundreds of orders in reasonable execution times. Actual instances from a utility have shown the suitability of the proposed algorithm.**

***Index Terms*—electric distribution systems, clustering problem, vehicle routing problem, heuristics.**

I. INTRODUCTION

There is a considerable number of orders in electric utilities demanded by customers or due to system maintenance. These services are named as service orders and can be classified as follow:

- Regulated: those orders which have associated due dates and predefined execution time;
- Commercial: those related to connecting or disconnecting customers from the distribution network;
- Technical: characterized by periodic maintenance activities and inspection on distribution network equipments;
- Emergency: involved in any kind of service with a strict and short due time.

There are regulations from the Brazilian National Agency of Electric Energy (ANEEL) which establish the maximum time to execute regulated orders, with specific sanctions in case of violation. Another aspect of great relevance is the diversity in the sense of different procedures associated with orders and the required skills associated with maintenance teams.

Managing these resources involves storing and processing a considerable amount of information, including the customers' data, the human and material resources from the company. Some available commercial systems generally do not offer the integration with corporate systems already adopted by utilities and some considerations related to company management policies have not been carried out.

When faced with the problem of deciding which set of orders each team will carried out, one can refer to the capacited clustering problem [1], [2]. Constraints frequently imposed are related to the work time associated with each team in the three first classes of orders presented: regulated, commercial and technical orders.

Assuming the considerable demand for these services and the limited resources available in utilities, one suitable approach would be improve the attendance, the classification and the scheduling of orders on a centralized dispatch system. Such proposal makes easy the standardization of attendance procedures in order to improve the global productivity.

In this work is proposed an approach to classify a set of service orders and assigned each group to an available team. The next section (II) defines the problem of clustering service orders, followed by section III which includes the algorithm developed. Finally sections IV and V contain the results and final remarks.

II. PROBLEM DEFINITION

Clustering service orders involves classifying them into a given number of groups which is equal to the number of available teams. One day before, the staff from utilities are faced with planning the schedule of the set of orders for the next 24 hours, considering only the commercial and technical ones known so far.

The classical problem of group elements geographically distributed is commonly refer as the *capacitated p-medians problem* [3], [1], [4], [5], [6]. Heuristic methods are incorporated in a great number of approach because this problem is **NP-complete** [7]

The developed model to the clustering problem considered in this paper includes two kind of fundamental elements (orders and teams) put together in a complete graph $G = (V, E)$, where:

- V is the set of vertices which represent the service orders; related with each vertice $i \in V$ there is a order execution time a_i (the time required to execute the procedures associated with) and a priority p_i;
- $E = \{(i,j) : i,j \in V\}$ corresponds to the edges. A distance d_{ij} associated with each edge $(i,j) \in E$ is defined by the euclidean distance between each $(i,j) \in E$.

The next the mathematical model is related to the clustering problem with two set of variables: one to denote the assignment between vertices and medians (x_{ij}) and the other to defined the median location (y_j).

$$x_{ij} = \begin{cases} 1 & \text{if the vertice } i \text{ is assigned to the median } j, \\ & \forall i,j \in V \\ 0 & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if the vertice } j \text{ is median,} \\ 0 & \text{otherwise} \end{cases}$$

In the hypothetical example of Figure 2, 12 orders have to be classified in 2 groups. The decision variables x_{ij} and y_j assume values shown in Figure 1.

x_{ij}

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	1	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	1	0	0

y_j

		2	3	4	5	6	7	8	9	10	11	12
y_j	0	0	0	1	0	0	0	0	0	1	0	0

Figure 1. Values for x_{ij} and y_j variables in the example of Figure 2.

The following constraints are assumed:

- each service order should be attended by only one team;
- The number of medians, or groups, is equal to the number of available teams;
- Every team should have its workday hours respected, in the sense of the number of hours involved.

$$Min \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \quad (1)$$

Subject to:

$$\sum_{j \in J} x_{ij} = 1, \ \forall i \in I \quad (2)$$

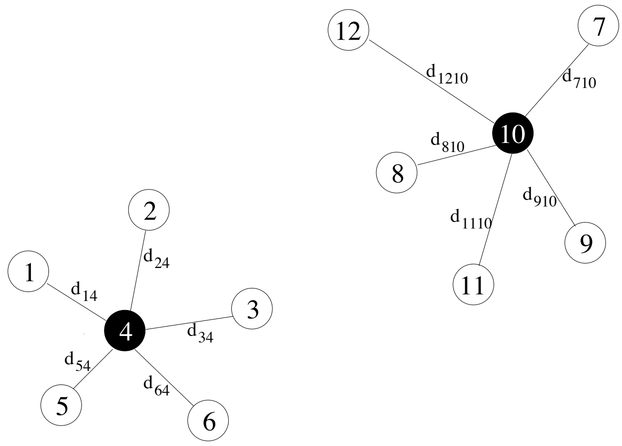

Figure 2. Hypothetical clustering of 12 orders into 2 groups.

$$\sum_{j \in J} y_j = p \quad (3)$$

$$\sum_{i \in I} a_i \frac{d_{ij}}{v} x_{ij} \leq \mu C_j, \ \forall j \in J \quad (4)$$

$$x_{ij}, \ y_j \in \{0,1\}, \ \forall i \in I, \ \forall j \in J \quad (5)$$

The objective function defined in Equation 1 minimizes the total distance, calculated by the sum of every distance between each order and its corresponding median.

Constraints 2 do not allow that a given order could be attended by more than one team. They also define that each order should be associated to at least one team.

The set of constraints 3 require that p medians or groups may be created, from the assumption that there are p available teams to serve the set of orders V.

The respect of workday hours is guaranteed by constraints 4, where for each group j is checked if the sum of all order execution times and of the distance traveled do not exceed the predefined number of hours times a μ factor. This factor should be defined in such a way that must be feasible to assign all orders to the given teams.

Finally constraints (5) define x_{ij} e y_j as binary variables.

One can note that in this model one peculiarity that makes it distinct from the classic models for *capacited p-medians problems*: the set 4. These constraints define the capacity of each group as the travel time, since is assumed that workday hours of each team must be respected.

III. The heuristic algorithm

Solving the problem defined in the previous section must first consider that the sum of all team workday hours is not necessarily enough to attend the whole set of orders, specially because the travel time should be also assumed.

Due to all these aspects, a heuristic algorithm is proposed in this paper to first define the necessary capacity of each group in hours in order to be able to classify all orders into the predefined groups.

Additionally, one aspect must be consider: the priority of each order. Utilities normally require that the sum of order execution times would be equivalent to all teams, in such a way that all of them may have almost the same chance to execute emergency orders. This situation leads to a reassign process and whenever priority orders are involved in, the regulations associated with these orders must be respected.

The approach proposed in this paper to solve the problem defined on section II is based on the algorithm developed by Ahmadi and Osman [8], manly due to the effectiveness on identifying and evaluating high concentrate regions in the sense of the number of orders. The methodology developed is a constructive heuristic algorithm which embody Ahmadi's ideas and the estimation of factor μ, described in Equation 4, section II. Additionally, two other important aspects are also included in the order clustering: the balance between the workday hours of all teams and the balance in terms of the time that each team will attend priority orders.

Figure 3 illustrates the algorithm developed, described from its main functions. Input parameters are assumed as the set of vertices and the set of edges with all associated data. Function *CreateInstance* create the data structure required in the algorithm, while in Step 2 the distance matrix is constructed. The workday hours of all teams is adjusted in Step 3, in an attempt to make possible the assignment of all orders. Another approach would be assuming non assigned orders, however it would be inconvenient from the definition of the objective function (Equation 1): letting all orders unassigned could give the best value of this objective function.

The solution is generated in Step 5, what involves creation of a number of groups equal to the number of teams and defining the assignment to all orders. The procedure included in this function holds the same structure as that of Ahmadi and Osman [8]: "Density Search Constructive Method (DSCM)". Parameters are defined as follow: the first ($inst$) corresponds to the problem instance; the second refers to reference solution[1], it is useful when is necessary to maintain median locations and modify only the order assignments; the third parameter corresponds to the consideration of the orders with priority equal to its value. The feasibility is checked on Step 7: in the case of feasible solution, all constraints defined on equations 2-5 have been respected and steps 8-10 are executed: in this phase, the assignment is changed by considering at each value of i one subset of orders with the corresponding priority value, always maintaining the feasibility of the solution constructed. The subset of regulated orders is carried on when $i = 1$, commercial orders are considered when $i = 2$ and finally the technical orders are assigned when $i = 3$.

When observing this approach, one concludes that the separation in the assignment phase of steps 8-10 makes possible to create different levels of priority associated with the assignment phase: since it must be respected the workday hours of each team, some orders may classified first than others and this will help to choose the subset of feasible orders that respect the workday hours of a team.

[1]When one intend to create a new solution and defining new median locations, as in Step 5, value 0 should be assumed for this parameter.

GroupOrders(V, E)
1. $inst$ = **CreateInstance**(V, E);
2. **CreateDistanceMatrix**$(inst)$;
3. μ = **CalculateFactor**$(inst)$;
4. **inst.setFator**(μ);
5. sol = **ConstructSolution**$(inst, 0, 0)$;
7. **If**$(sol.feasible())$
8.   **EmilinateDesignation**(sol);
9.   **For**$(i = 1 \ldots 3)$
10. sol = **ConstructSolution**$(inst, sol, i)$;
11. **return**(sol)

Figure 3. The heuristic algorithm proposed.

After defining the algorithm *GroupOrders* of Figure 3, all the procedures involved in functions *CalculateFactor* and *ConstructSolution* are detailed in the next sections.

A. The algorithm for calculating μ

Defining a value for μ is one of the contribution of this work, since the mathematical model described in section II makes an assumption that the capacity previously defined do not make the problem feasible. The travel time of each team is obtained only after deciding which subset of orders will be assigned to and this time must be less than or equal to the workday hours defined for each one of these teams.

Therefore, when the given workday hours of each team do not hold the feasibility of the problem, a μ factor for increasing the workday hours of each team is defined.

The algorithm developed, shown in Figure 4 is much simple and requires a reasonable computational effort. From the given workday hours furnished for all teams, the algorithm *ConstructSolution* (Step 3) is executed and is checked if a feasible solution have been obtained. When it occurs, the execution ends and $\mu = 1$ is returned. Otherwise, μ is increased in 10% at each iteration of the loop 4-8 until one feasible solution is obtained or the maximum number of iterations is reached (constant MAX).

B. The constructive algorithm proposed

As already mentioned, the procedure involved in function *ConstructSolution* is based on the method of Ahmadi and Osman [8]. Some phases were simplified in order to reduce the computational time required.

Figure 5 shows the algorithm developed from Ahmadi's ideas. Following the selection structure from step 1, two main blocks can be identified: one to construct solutions without pre-defined median locations (steps 2-6) and other to proceed only the assignment of the subset of orders defined by the priority given as parameter (steps 7-11). Those calculations related to the measure proposed by Ahmadi and Osman [8] are carried out in function *CalculateDensity* and the

```
CalculateFactor(inst)
1.  counter = 0;
2.  μ = 1;
3.  sol = ConstructSolution(inst, 0, 0);
4.  While(Not(sol.feasible) and
          (counter < MAX))
5.     μ = μ * 1, 1;
6.     IncreaseWorkday(inst, μ);
7.     counter = counter + 1;
8.     sol = ConstructSolution(inst, 0, 0);
9.  return(μ)
```

Figure 4. Algorithm *CalculateFactor*.

results are stored in the *sol* structure. Function *CreateMedian* defines a new group with the corresponding median location, based on the maximum density value calculated in function *CalculateDensity*. Function *Assign* also makes use of density, specially on step 10 when, for those existent medians, it is assumed that only the subset of orders with non-exceed limit of a pre-defined execution time will be assigned. On step 11, those non-assigned orders are included in *Or* and the following procedure of step 12 is related to recalculation of median locations of each group, with the corresponding re-assignment of orders, according to "findBestClusters" of [8]

```
ConstructSolution(inst, solref, prior)
1.  If(solref = 0)
2.     sol = CreateSolution(inst);
3.     While(NumMedians(sol) <
             NumTeams(inst))
4.        sol = CalculateDensity(inst, sol);
5.        sol = CreateMedian(sol);
6.        sol = Assign(sol);
    Else
7.     Or = SelectOrders(inst, prior);
8.     While(Or ≠ e sol.feasible())
9.        sol = CalculateDensity(inst, sol);
10.       sol = Assign(sol);
11.       Or = VerifyNonAssigned(sol);
12. sol = RedefineGroups(sol);
13. return(sol)
```

Figure 5. Algorithm *ConstructSolution*.

IV. Computational results

The proposed algorithm was applied to two instances, both related to actual scenarios from a Brazilian utility with service orders to be executed. The available service orders of one typical day were used to create each instance, with the corresponding data shown in Table I. The first column contains the instance number, the second the number of orders, the third the number of teams and finally the fourth column shows the total time required to attend the whole set of orders, without considering the travel time. Next, it will be show the results for both instances.

Table I
The instance set considered.

#Inst.	#Orders	#Teams	Time(hours)
1	102	5	25.5
2	430	10	107.5

Figures 6 and 7 describe the distribution of groups defined by the algorithm developed, for instance 1 and 2, respectively. Each point is one order and has associated one color of the corresponding team.

When observing Figure 6, a high concentrate regions are identified and the overlapping between groups is almost nonexistent. This overlapping is more expressive for results of instance 2, however one can note some regions of high density of orders.

One possible explanation for the problem identified in instance 2 is due to the given emphasis on balancing the work time of each team, what imposes a decision a little inadequate in the sense of overlapping groups and in terms of the whole travel distance.

Tables II and III have shown the effectiveness in obtaining a balance between the workday hours of each team, to detriment of a better concentration of orders. An expressive undesired result refers to teams 6-8, of instance 2: in this case was observed a high variance when compared to the remaining teams. One possible hyphotesis to explain this phenomenon may be the algorithm conception to identifying high concentrate regions.

Table II
Clustering data for instance 1.

Team	#Orders	Execution time	Travel time	Total time
1	10	2.50	11.13	13.63
2	42	10.50	1.93	12.43
3	2	0.50	5.49	5.99
4	16	4.00	8.90	12.90
5	32	8.00	4.97	12.97
			Average:	11.59
			Standard deviation:	2.82

V. Final remarks

This paper has shown an approach to clustering service orders in electric utilities. The proposed solution was based on constructive heuristic procedures of Ahmadi and Osman [8], manly due to the effectiveness of this procedure in identifying high concentrate regions in a geographical order space.

In addition of the proposed algorithm, another contribution of this work refers to the mathematical modeling of the clustering problem to be applied to service orders in the context of electric utilities, assumed as the well-known *capacitated p-medians* problem. Specially by considering individual and restrict workday hours of each team, a definition of capacity

Table III
CLUSTERING DATA FOR INSTANCE 2.

Team	#Orders	Execution time	Travel time	Total time
1	26	6.50	4.58	11.08
2	25	6.25	5.57	11.82
3	23	5.75	5.90	11.65
4	20	5.00	4.50	9.50
5	23	5.75	5.01	10.76
6	83	20.75	2.85	23.60
7	87	21.75	1.90	23.65
8	84	21.00	2.37	23.37
9	25	6.25	4.78	11.03
10	34	8.50	5.82	14.32
			Average:	15.08
		Standard deviation:		5.66

for each group was needed in order to have the problem feasible when faced with constraints of section II. One can note that the proposed model do not permit unassigned orders.

The computational results have shown that the algorithm developed was suitable to the given actual instance set considered. Further investigations should consider other methods in order to reduce the total travel time, while maintaining the balance between the workday hours of each team.

ACKNOWLEDGMENTS

The authors would like to thank the AES SUL Distribuidora Gacha de Energia SA for financial support provided to the project "Gestão Estratégica operacional - otimização de rotas e despachos de ordens de serviço".

REFERENCES

[1] I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search," *International Transactions in Operational Research*, vol. 1, no. 3, pp. 317–336, 1994.

[2] R. Z. Ríos-Mercado and E. Fernández, "A reactive grasp for a commercial territory design problem with multiple balancing requirements," *Computers & Operations Research*, vol. 36, no. 3, pp. 755–776, 2009.

[3] J. M. Mulvey and M. P. Beck, "Solving capacitated clustering problems," *European Journal of Operations Research*, vol. 18, pp. 339–348, 1984.

[4] P. M. França, N. M. Sosa, and V. Pureza, "An adaptive tabu search algorithm for the capacitated clustering problem," *International Transactions in Operational Research*, vol. 6, pp. 665–678, 1999.

[5] L. A. Lorena and E. L. F. Senne, "A column generation approach for the capacitated p-median problems," *Computers & Operations Research*, vol. 31, pp. 863–876, 2004.

[6] S. Ahmadi and I. H. Osman, "Greedy random adaptative memory programming search for the capacitated clustering problem," *European Journal of Operations Research*, vol. 162, pp. 30–44, 2005.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[8] S. Ahmadi and I. H. Osman, "Density based problem space search for the capacitated clustering *p*-median problem," *Annals of Operations Research*, vol. 131, pp. 21–43, 2004.

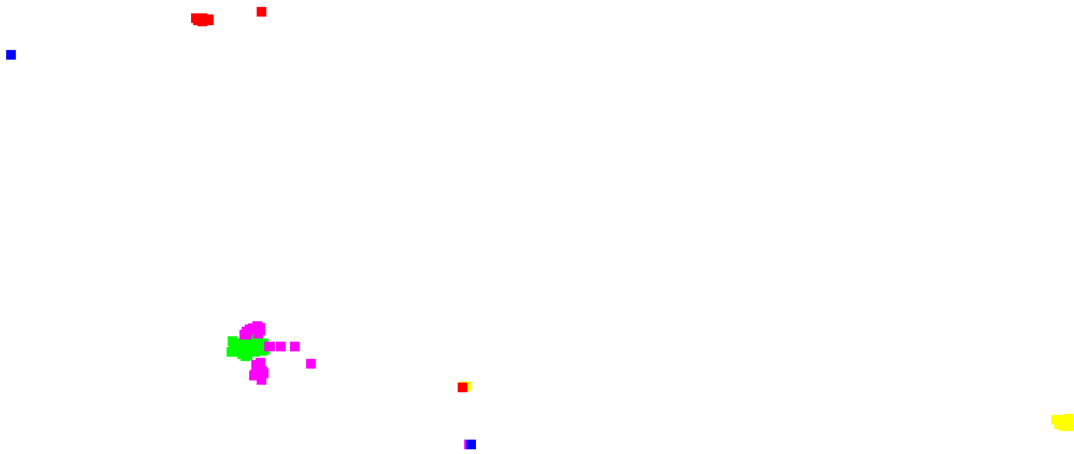

Figure 6. Results for instance 1.

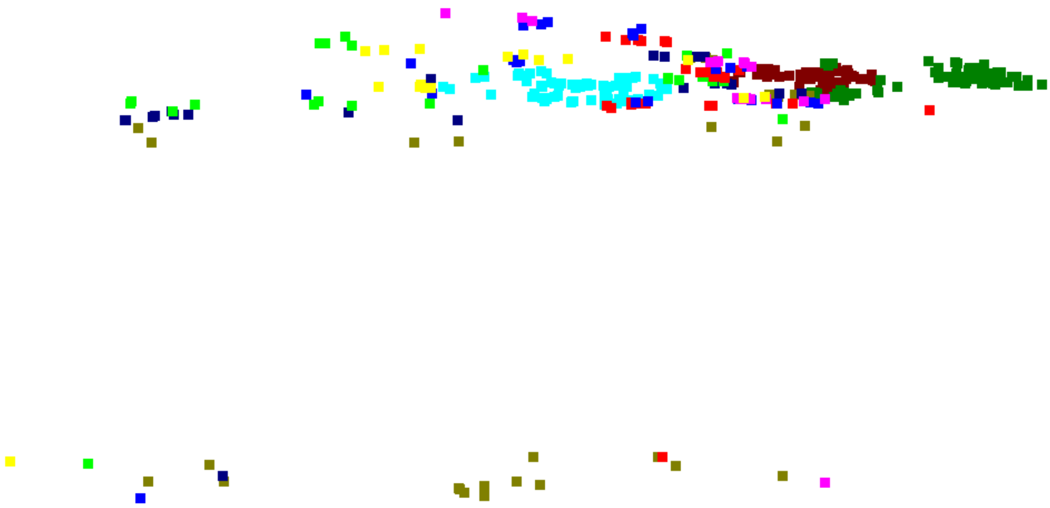

Figure 7. Results for instance 2.