

UM ALGORITMO PATH RELINKING COM BUSCA TABU PARA O PROBLEMA DE AGRUPAMENTO CAPACITADO EM CENTRO GEOMÉTRICO

Hedley Luna Gois Oriá¹,
Marcos Negreiros¹, Albert Einstein Fernandes Muritiba²

¹ Universidade Estadual do Ceará (UECE)
Curso de Bacharelado em Ciências da Computação (CBCC)
Av. Paranjana, 1700 – Campus do Itaperi
CEP: 60740-000 – Fortaleza – CE – Brazil

²Universidade Federal do Ceará (UFC)
Departamento de Estatística e Matemática Aplicada (DEMA)

hedleygois@gmail.com, negreiro@graphvs.com.br, einstein@ufc.br

Resumo. O PACCG - Problema de Agrupamento Capacitado em Centro Geométrico consiste em particionar n pontos no \mathbb{R}^2 , com peso definido, em p grupos disjuntos respeitando que a soma dos pesos dos pontos em um grupo não ultrapassa uma capacidade máxima. O objetivo do problema é minimizar a soma das distâncias entre cada ponto e o centro geométrico do seu grupo. Trata-se de um problema NP-Árduo, justificando-se o uso de heurísticas para tratá-lo. Nesse artigo, propomos um novo algoritmo combinado para o PACCG, que usa as meta-heurísticas path relinking e busca Tabu. O método mostrou-se eficiente para 23 das 25 principais instâncias da literatura, tendo contribuído com 10 novos limites superiores.

Palavras Chave: Agrupamento Min-Sum, Agrupamento Restrito, Meta-heurísticas.

Abstract. The capacitated centred clustering problem (CCCP) is a problem of partitioning a set of n individuals in \mathbb{R}^2 , each with a specified capacity, in p distinct clusters such that the capacity of each cluster formed is limited to a maximum value (Q). Its objective is to minimize the sum of distances of the individuals to their allocated cluster center. It is a NP-Hard problem in which justifies the use of heuristics to solve its instances. In this article, we propose a new combined algorithm to the CCCP, that uses path relinking previously to a tabu search schema. The method showed to be efficient to 22 of the 25 principal instances from the literature, and obtained 10 new upper bounds.

Keywords: Min-sum-square Clustering, Constrained Clustering, Metaheuristics.

1 Introdução

Um problema bastante estudado na computação é o agrupamento de indivíduos em grupos disjuntos, a esses grupos se dá o nome de *clusters*. Problemas de agrupamento podem ser classificados de diversas maneiras, de acordo com as limitações e atributos desejáveis dos *clusters*.

Duas variações destes problemas podem ser assim descritas:

- PAC - O Problema de Agrupamento Capacitado (*Capacitated Clustering Problem - CCP*) foi proposto por Mulvey & Beck (1984) e é definido simplifcadamente como o problema de agrupar n indivíduos em p *clusters* mutuamente exclusivos tal que, cada *cluster* tem uma limitação de capacidade. O objetivo do PAC é minimizar a soma das distâncias entre cada indivíduo e a mediana do seu grupo, [Mulvey (1984)], [Chistofides (1994)];
- PACCG - O Problema de Agrupamento Capacitado em Centro Geométrico (*Capacitated Centred Clustering Problem - CCCP*), foi proposto por Negreiros & Palhano (2006), sendo muito parecido com o PAC. A diferença básica é que o centro de cada *cluster* é a média aritmética das coordenadas (centro geométrico) de cada *cluster*, [Negreiros (2006)].

O PACCG tem muitas aplicações, variando desde como sub-problema para problemas de roteirização, até aplicações no Marketing, cobertura territorial de força de vendas, agentes de saúde, localização de viaturas de polícia, entrega de jornais a assinantes, etc. [Negreiros (2003)], [Negreiros (2006)].

Neste artigo, na seção 2 apresentamos uma descrição matemática para o PACCG que será aqui considerado, na seção 3 mostramos uma nova heurística para o PACCG que usa Busca Tabu operando antes com melhorias iterativas de um método *Path Relinking*. Na seção 4 fornecemos dados comparativos de desempenho entre os resultados presentes na literatura e a abordagem sugerida, e mostramos que a nova abordagem supera muitas das soluções obtidas pelas demais metodologias até então descritas para este problema.

2 Descrição do Problema

Análise de *cluster* consiste no problema de dividir um conjunto de entidades (indivíduos) em conjuntos, de modo que cada grupo formado seja mutuamente exclusivo, ou seja, nenhum indivíduo pode estar em mais de um *cluster* ao mesmo tempo. Outra definição é dada por Negreiros & Palhano (2006), consiste em agrupar indivíduos de forma a maximizar a homogeneidade das relações dos grupos e, ao mesmo tempo, que se maximiza a heterogeneidade entre indivíduos de *clusters* diferentes, [Negreiros (2006)].

O critério usado para dividir as entidades pode ser muito variado, dependendo da aplicação em que a análise será feita, há um critério apropriado. *Clusters* também são conhecidos como hipervértices de um hipergrafo. Neste trabalho, o critério de heterogeneidade entre *clusters* de indivíduos é tomado como a soma das distâncias euclidianas entre cada indivíduo de um grupo e seu centro (denominado como a soma dos mínimos quadrados entre *clusters*).

O PACCG é uma maneira de se fazer agrupamentos esféricos com restrições. Existem duas variações do PACCG, o PACCG Genérico (*gPACCG*) e o *p*-PACCG (tratado simplesmente como PACCG). A diferença básica entre elas é que no *gPACCG* o número de *clusters* não é conhecido **a priori**, e, com isso, pode-se abrir e fechar *clusters* quando convier limitando-se a uma penalidade à abertura de cada novo grupo adicionada na função objetivo. Já no PACCG o número de *clusters* é estipulado previamente. Discutimos aqui apenas o *p*-PACCG. O modelo matemático do PACCG pode se proposto como segue:

$$(p - \text{PACCG}) \text{Minimizar} \sum_{i \in I} \sum_{j \in J} \|a_i - \bar{x}_j\| y_{ij} \quad (1)$$

$$\text{sujeito a : } \sum_{j \in J} y_{ij} = 1, \forall i \in I \quad (2)$$

$$\sum_{i \in I} y_{ij} \leq n_j, \forall j \in J \quad (3)$$

$$\sum_{i \in I} a_i y_{ij} \leq n_j \bar{x}_j, \forall j \in J \quad (4)$$

$$\sum_{i \in I} q_i y_{ij} = Q, \forall j \in J \quad (5)$$

$$\bar{x}_j \in \mathbb{R}^l, n_j \in \mathbb{N}, y_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J \quad (6)$$

Onde:

l - representa a dimensão do espaço onde está definido o problema;

I - é o conjunto dos indivíduos;

J - é o conjunto dos centros de *clusters*;

p - é o número de (*clusters*);

a_i - representa o vetor de dimensão l contendo as coordenadas do indivíduo i ;

q_i - é a oferta de cada indivíduo;

Q - é a capacidade máxima associada à mediana;

n_j - é o número de elementos no cluster j ;

$$\bar{x}_j = \begin{cases} 1, \text{ é o vetor de dimensão } l \text{ representando as coordenadas do centro do cluster } j \\ 0, \text{ caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, \text{ se o indivíduo } i \text{ está atribuído ao cluster } j \\ 0, \text{ caso contrário} \end{cases}$$

A função objetivo 1 visa minimizar a dissimilaridade entre os indivíduos dos *clusters*. A restrição 2 trata que cada indivíduo está em apenas um *cluster*. A restrição 3 diz respeito à quantidade máxima de indivíduos por *cluster*. A restrição 4 refere-se ao centro do *cluster*, que no PACCG é o centro geométrico. A restrição 5 diz que a soma dos indivíduos atribuídos a um *cluster* está limitado à capacidade máxima do grupo. A restrição 6 refere-se às variáveis de decisão do modelo.

A resolução do PACCG foi em sua maioria tratada pela literatura considerando apenas métodos heurísticos e metaheurísticos para resolvê-lo. Métodos meta-heurísticos constituem-se de um subcampo da otimização estocástica, [LUKE (2009)]. Otimização estocástica é uma área da otimização que trata uma classe de algoritmos que tentam de forma randômica, mas perto da ideal, resolver problemas difíceis. As meta-heurísticas consistem nas técnicas mais gerais de se aplicar esses algoritmos de forma independente do problema que está sendo estudado. As meta-heurísticas usadas neste trabalho foram a busca tabu (*tabu search*) e a reconexão de caminhos (*path relinking*).

A busca tabu (*tabu search*) é uma meta-heurística que guarda um histórico (memória) das boas soluções (soluções candidatas) já visitadas, esse histórico é conhecido como lista tabu (*tabu list*). Com o decorrer da busca, o algoritmo chega a um ótimo local,

quando isso acontece, guarda-se essa solução na lista e volta-se a solução anterior ou vai-se para uma solução distante, dependendo da implementação. Mesmo indo para uma solução pior, deve-se garantir que seja proibitivo voltar àquele ótimo local previamente visitado. Percorre-se outro caminho no espaço de busca, isso então garante que o algoritmo não pára em um ótimo local, [LUKE (2009)].

O procedimento *path-relinking* foi proposto inicialmente por Glover (1999) para compor a fase de diversificação da meta-heurística *Scatter Search*, e possui a estratégia de gerar novas soluções ao se explorar um trajeto que conecta soluções de boa qualidade (soluções elite), [Glover (1999)].

Esse procedimento exploratório inicialmente seleciona uma solução conhecida por solução inicial. Posteriormente é construído um caminho pelo espaço de soluções (vizinhança) que a conecta até uma outra solução, chamada de solução guia, gerando assim novas soluções (intermediárias) a cada movimento executado. Este movimento é realizado inserindo características ou atributos da solução guia na solução intermediária.

3 Um Método Combinado para o PACCG

O método construído contém as fases: construtiva, de busca local e de melhorias por meta-heurísticas, como pode ser visto no algoritmo 3. A fase construtiva é baseada no algoritmo Melhor Primeiro Randomizado (RBFC), proposto por [Muritiba (2012)]. Aqui se inclui uma estratégia de *path relinking* (PR) numa fase intermediária, [Glover (1999)].

O RBFC cria uma solução viável a partir de pontos sementes (*seeds*) de forma gulosa. Após isto aplica-se uma busca local (BL) para melhorar os resultados obtidos. A busca local, procura por movimentos que melhorem a solução final, e, os realiza, quando achados. Caso a solução gerada pelo RBGC não seja melhor que a até então melhor solução obtida, inicia-se o algoritmo de *path relinking* (PR), religando a solução corrente à melhor solução até então obtida.

O algoritmo PR é baseado nos primeiros vértices a entrarem nos *clusters* definidos como os primeiros p vértices do vetor (*seed*). Faz-se várias trocas entre os pivôs assim como em outros vértices da solução corrente, de forma a tentar igualar aos vértices da melhor solução. Após essa fase de trocas, chama-se mais uma vez o RBGC para gerar uma nova solução inicial e, por fim, inicia-se mais uma busca local na tentativa de melhorar a solução atual. Esse processo é feito sempre que o RBFC não consegue gerar uma nova solução guia.

Mais detalhadamente, temos um vetor com os vértices da solução guia em memória, e a cada geração da solução inicial preenchemos um vetor com os vértices da solução corrente. As trocas funcionam iterando-se sobre todos os vértices do conjunto, através de passos de tamanho p , onde p é a quantidade de *clusters* da instância.

Por exemplo, considerando a instância *TA50* figura 1, temos 5 *clusters* com 50 vértices, buscamos inicialmente as posições 0, 5,10,15 e assim por diante nos vetores de vértices. Comparamos os dois vetores de vértices, *Seed*, referente a solução atual e *BSeed*, referente a solução guia. Caso sejam diferentes, o índice é guardado, digamos m . Itera-se então sobre o vetor dos vértices da solução corrente a fim de encontrarmos o vértice da solução atual que seja igual ao vértice de índice m da solução guia. Ao encontrarmos o vértice, trocamos a posição dos dois no vetor da solução atual. Então, dá-se o passo de tamanho p e faz-se mais uma vez a busca.

Algorithm 1 MCombPACCG

Procedure MCPACCG

p - quantidade de clusters;
 It - número de iterações do método combinado;
 $RBFC()$ - Melhor Primeiro Randomizado
 $PR()$ - algoritmo de Path Relinking
 $BL()$ - Algoritmo de Busca Local para o PACCG
 $Seed$ - Vetor de pontos randômicos (p-sementes)
 $BSeed$ - Vetor contendo as sementes da melhor solução obtida
 $GeraRandom()$ - Gera um vetor $seed$ com p-sementes
 $SolAtual$ - Solução Atual
 $MelhorSol$ - Solução Guia
 x - Solução corrente
 x^* - Melhor solução obtida

```

1: for  $i \leftarrow 1$  to  $It$  do
2:    $GeraRandom(p, Seed)$ ;
3:    $x := RBFC(Seed)$ ;
4:    $x := BL(x)$ ;
5:   if  $SolAtual > MelhorSol$  then
6:      $x^* := PR(x, x^*)$ ;
7:      $MelhorSol := SolAtual$ ;
8:   else
9:      $x^* := PR(x, x^*)$ ;
10:     $BSeed := Seed$ ;
11:     $MelhorSol := SolAtual$ ;
12:   end if
13: end for
14:  $BT(x^*)$ 

```

O processo de geração de soluções iniciais ocorre até que a quantidade máxima de iterações seja atingida, ou caso atinja um limite de tempo pré-estabelecido. Na figura 1 temos o processo de mudança na solução conforme indicado anteriormente.

Após a etapa de construção da solução inicial, dá-se início à Busca Tabu. O algoritmo 3 apresenta o método de melhorias meta-heurístico proposto.

4 Resultados Computacionais

A seguir temos a descrição das instâncias usadas para gerar os resultados, e em seguida faremos a comparação com os resultados até então obtidos pela literatura.

4.1 Instâncias

As instâncias estudadas estão disponíveis na literatura relacionada ao PACCG. Temos sete instâncias denominadas *TA*, seis do tipo *sjc*, cinco instâncias do tipo *p3038* e, por fim, temos sete instâncias do tipo *doni*. Estas instâncias podem ser obtidas através da OR-library, [ORL (2012)].

A tabela 1 mostra estatísticas sobre as instâncias. As colunas indicam os nomes das instâncias *Nome*, o número de pontos n , o número de *clusters* p , a capacidade de cada *cluster*, Q , a média da demanda dos vértices (**w_Avg**) e o desvio padrão da demanda de cada vértice, (**w_Dev**).

As instâncias do tipo *TA* têm um pequeno número de vértices, no máximo 100, e também um pequeno número de vértices por *cluster*. Uma característica interessante das

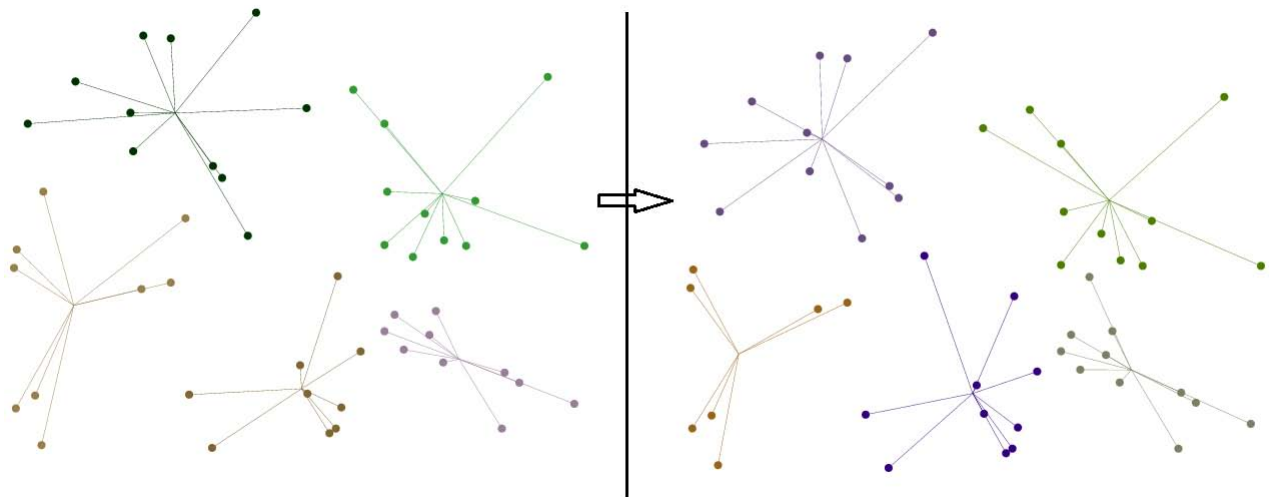


Figura 1. Solução antes e depois de executado o *Path Relinking* na instância TA50.

Algorithm 2 Path Relinking

Procedure PR

p - quantidade de clusters

$size$ - quantidade de vértices no total

$Seed$ - vetor de vértices da solução atual

$BSeed$ - vetor de vértices da solução guia

$RBFC()$ - Melhor Primeiro Randomizado

$BL()$ - algoritmo de busca local

$SolAtual$ - solução atual

$MelhorSol$ - solução guia

```

1: for  $i \leftarrow 1$  to  $p$  do
2:   for  $m \leftarrow 1$  to  $size$  step  $p$  do
3:     if  $Seed[m] \neq BSeed[m]$  then
4:       for  $j \leftarrow 1$  to  $size$  do
5:         if  $Seed[j] = BSeed[m]$  then
6:           troca( $Seed[j]$ ,  $BSeed[m]$ )
7:         end if
8:       end for
9:     end if
10:  end for
11:   $SolAtual := RBFC(Seed)$ 
12:   $BL(Seed)$ 
13:  if  $SolAtual < MelhorSol$  then
14:     $best \leftarrow curr$ 
15:    Exit;
16:  end if
17: end for

```

instâncias dessa família, é que todos os vértices têm demanda igual a um. Foram criadas artificialmente por [Negreiros (2006)] para servirem de base para os algoritmos de pesquisa sobre este problema.

As instâncias *sjc* têm pontos bem distribuídos. Isso nos indica que a separação dos pontos nos *clusters* é difícil. Essa família, assim como a *TA*, têm relativamente poucos pontos, no máximo 402, e poucos *clusters* são formados, no máximo de 40. Estas instâncias foram originalmente concebidas para o PAC, referentes à cidade de São José dos Campos

| Instância | n | p | Q | w_Avg | w_Dev |
|------------|-------|------|-----|-------|-------|
| TA25 | 25 | 5 | 6 | 1 | 0 |
| TA50 | 50 | 5 | 11 | 1 | 0 |
| TA60 | 60 | 5 | 13 | 1 | 0 |
| TA70 | 70 | 5 | 17 | 1 | 0 |
| TA80 | 80 | 7 | 12 | 1 | 0 |
| TA90 | 90 | 4 | 23 | 1 | 0 |
| TA100 | 100 | 6 | 17 | 1 | 0 |
| SJC1 | 100 | 10 | 720 | 58.07 | 51.86 |
| SJC2 | 200 | 15 | 840 | 46.34 | 37.15 |
| SJC3a | 300 | 25 | 740 | 37.51 | 29.58 |
| SJC3b | 300 | 30 | 740 | 37.51 | 29.58 |
| SJC4a | 402 | 30 | 840 | 39.76 | 32.87 |
| SJC4b | 402 | 40 | 840 | 39.76 | 32.87 |
| p3038_600 | 3038 | 600 | 321 | 50.85 | 24.75 |
| p3038_700 | 3038 | 700 | 273 | 50.33 | 25 |
| p3038_800 | 3038 | 800 | 238 | 50.26 | 25 |
| p3038_900 | 3038 | 900 | 216 | 51.29 | 25.34 |
| p3038_1000 | 3038 | 1000 | 191 | 50.44 | 24.87 |
| doni1 | 1000 | 6 | 200 | 1 | 0 |
| doni2 | 2000 | 6 | 400 | 1 | 0 |
| doni3 | 3000 | 8 | 400 | 1 | 0 |
| doni4 | 4000 | 10 | 400 | 1 | 0 |
| doni5 | 5000 | 12 | 450 | 1 | 0 |
| doni6 | 10000 | 23 | 450 | 1 | 0 |
| doni7 | 13000 | 30 | 450 | 1 | 0 |

Tabela 1. Instâncias de teste da literatura relacionadas ao PACCG.

- SP, [Lorena (1994)], [Lorena (2003)].

As instâncias *p3038*, assim como as *sjc*, também têm pontos bem distribuídos, entretanto as duas famílias diferem radicalmente na quantidade de pontos e *clusters*. As *p3038* têm uma quantidade significativa de vértices, 3038. Nesta família, a quantidade de *clusters* é grande, variando entre 600 e 1000. São instâncias muito conhecidas, pois são provenientes do problema do Caixeiro Viajante.

As instâncias *doni* são as mais "mal comportadas" relativamente à distribuição dos vértices e de suas demandas, e portanto são as mais difíceis de serem resolvidas. Assim como as *TA*, todos os pontos têm demanda igual a um. A principal característica dessa família é a elevada quantidade de vértices e poucos *clusters*. A quantidade de pontos varia de 1000 à 13221 e a quantidade de *clusters* fica na faixa de 6 a 30. Muitos pontos têm a mesma coordenada. Estas instâncias foram extraídas de uma aplicação real na cidade de Fortaleza/CE, ligada à distribuição da força de vendas de um atacadista de produtos secos do mercado cearense.

4.2 Resultados Computacionais para o PACCG

As implementações computacionais foram feitas usando a linguagem C++(4.5.2) no sistema operacional Ubuntu Linux 11.04. O hardware usado foi: Intel Core 2 Quad Q9550 CPU 2.83Ghz (por núcleo, 4 núcleos), 4 GB de RAM. Os tempos de CPU foram medidos com precisão de segundos, e a implementação foi feita para rodar em um único núcleo, de forma sequencial.

Para fazer os testes, foram criadas, à priori, 100 soluções iniciais para a busca tabu, também limitado o processo inteiro em 48 horas, sendo 24 horas para a fase construtiva e

24 horas para a fase de refinamento. A busca tabu foi limitada a 4000 iterações.

A tabela 2 compara os resultados obtidos usando o Método Combinado (PR+TS) com os até então existentes na literatura, provenientes de [Negreiros (2006)], [Pereira (2008)], [Palhano (2008)] [Chaves (2010)], [Chaves (2011)].

Na tabela 2, temos uma coluna mostrando o nome da instâncias **Instância**), o custo final encontrado do método combinado (**PR-TS**) e o tempo total necessário para completar o método (**TComb**), o custo da melhor solução reportada na literatura (**MS Ant**) e o tempo reportado na literatura para obter esta melhor solução (**MTp Ant**). A medida do tempo é em segundos. Na última coluna, (**GAP**), temos o percentual das soluções encontradas pelo método combinado à melhor solução obtida, ou ótimo "putativo" da instância.

| Instância | PR-TS | TComb (s) | MS Ant | MTp Ant (s) | GAP |
|------------|---------------------|-----------|-----------------|-------------|------|
| TA25 | 1251.44 | 0 | 1251.44 | 0 | 0.00 |
| TA50 | 4474.52 | 0 | 4474.52 | 0 | 0.00 |
| TA60 | 5356.58 | 0 | 5356.58 | 0 | 0.00 |
| TA70 | 6240.67 | 0 | 6240.67 | 0 | 0.00 |
| TA80 | 5730.28 | 0 | 5515.46 | 0 | 3.90 |
| TA90 | 9069.85 | 0 | 8899.05 | 1 | 1.91 |
| TA100 | 8106.76 | 1 | 8102.04 | 0 | 0.00 |
| SJC1 | 17359.75 | 7 | 17359.75 | 1 | 0.00 |
| SJC2 | 33181.65 | 41 | 33181.65 | 3 | 0.00 |
| SJC3a | 45356.35 | 221 | 45356.35 | 24 | 0.00 |
| SJC3b | 40661.94 | 284 | 40661.94 | 21 | 0.00 |
| SJC4a | 61993.95 | 477 | 61931.60 | 3026 | 0.01 |
| SJC4b | 52202.48 | 585 | 52202.48 | 153 | 0.00 |
| p3038_600 | 124863.78(*) | 78456 | 126567.31 | 810 | 0.00 |
| p3038_700 | 112510(*) | 58644 | 115168.49 | 1022 | 0.00 |
| p3038_800 | 103016.15(*) | 60039 | 105352.33 | 2411 | 0.00 |
| p3038_900 | 95410.45(*) | 72031 | 97319.54 | 1650 | 0.00 |
| p3038_1000 | 88443.70(*) | 77043 | 89896.55 | 1017 | 0.00 |
| doni1 | 3024.99 | 1121 | 3021.41 | 0 | 0.01 |
| doni2 | 6384.39 | 5093 | 6080.70 | 0 | 4.99 |
| doni3 | 8342.37(*) | 44930 | 8363.49 | 1032 | 0.00 |
| doni4 | 10761.30(*) | 39758 | 10777.64 | 1450 | 0.00 |
| doni5 | 11109.10(*) | 80707 | 11114.67 | 437 | 0.00 |
| doni6 | 15542.30(*) | 150486 | 15610.46 | 5476 | 0.00 |
| doni7 | 18253.90(*) | 85322 | 18484.13 | 36878 | 0.00 |

Tabela 2. Comparativos entre diferentes estratégias heurísticas criadas para resolver o PACCG

A tabela 2 mostra que o método combinado atingiu a melhor solução em 22 das 25 instâncias reportadas, estando a no máximo 4.99% da melhor solução, para o caso da instância doni2, sendo as duas outras do grupo de instâncias **TA**. A tabela também mostra que foram estabelecidos dez novos limites superiores para estas instâncias, indicados por (*), todas relativas ao grupo **p3038** e cinco das sete do grupo de instâncias **doni**.

Apesar do resultado promissor encontrado, o método combinado realizou a tarefa de busca, na maioria das vezes, em tempo muito superior ao obtido para o melhor resultado da literatura em tempo. Um maior investimento no tratamento desta componente deve ser feito proximamente.

5 Conclusão

Como pode ser visto, a estratégia PR resultou em bons resultados superando, em várias instâncias, os anteriormente obtidos pela literatura. Entretanto, também foi observado um aumento significativo no tempo de execução do método usado para obter tais resultados.

O ganho nos resultados ficou mais claro nas instâncias maiores e mais complexas. Nas instâncias *sjc* foram alcançados os mesmos melhores resultados obtidos pela literatura. Nas instâncias *p3038* conseguiu-se novos limites superiores. Nas instâncias *doni* também foi possível estabelecer novos limites superiores.

A estratégia combinada PR+TS, apesar de ter apresentado melhores resultados aos que os da literatura, pode ser melhorada quanto ao tempo de execução.

Referências

- [ORL (2012)] (2012). Orlibrary. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>.
- [Chaves (2010)] Chaves, A. A.; Lorena, A. (2010). Clustering search algorithm for the capacitated centred clustering problem. *Computers and Operations Research*, 37:552–558.
- [Chaves (2011)] Chaves, A. A.; Lorena, A. (2011). Hybrid evolutionary algorithm for the capacitated centered clustering problem. *Expert Systems with Applications*, 38:5013–5018.
- [Chistofides (1994)] Chistofides, N.; Osman, I. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *Internacional Transactions in Operational Research*, 1:316–317.
- [Glover (1999)] Glover, F. (1999). Scatter search and path relinking. In: D Corne, M Dorigo and F Glover (eds.), *New Ideas in Optimisation*, Wiley.
- [Lorena (2003)] Lorena, A.; Senne, E. (2003). Local search heuristics for capacitated p-median problems. *Networks and Spatial Economics*, 3:407–419.
- [Lorena (1994)] Lorena, A.; Senne, E. (1994). A column generation approach to capacitated p-median problems. *Computers and Operations Research*, 31:863–876.
- [LUKE (2009)] LUKE, S. (2009). Essentials of metaheuristics. <http://cs.gmu.edu/~sean/book/metaheuristics/>.
- [Mulvey (1984)] Mulvey, J.; Beck, M. (1984). Solving capacitated clustering problems. *European Journal of Operational Research*, 18:339–348.
- [Muritiba (2012)] Curitiba, A.E.F.; Negreiros, M. O. H. S. M. (2012). A tabu search for the capacitated centred clustering problem. *Technical Report - UFC/DEMA*, page pp 12.
- [Negreiros (2003)] Negreiros, M.J.; Palhano, A. (2003). Uma aplicação para o problema generalizado de percurso de veículos. *Anais XXV SBPO-Natal/RN*.
- [Negreiros (2006)] Negreiros, M.J.; Palhano, A. (2006). The capacitated centred clustering problem. *Computers and Operations Research*, 33:1639–1663.
- [Palhano (2008)] Palhano, A.W.C.; Negreiros, M. G. (2008). A constrained k-median procedure for the capacitated centred clustering problem. *Anales del XIV CLAIO*, cd rom.
- [Pereira (2008)] Pereira, M. S. E. (2008). A column generation method for the capacitated centred clustering problem. *Annals of VI ALIO/EURO*, pages 1–6.