

Metaheurística GRASP para o Problema de Agrupamento

Paulo Morelato França

Depto de Matemática, Estatística e Computação- FCT, UNESP
19060-900, Presidente Prudente, SP
E-mail: paulo.morelato@fct.unesp.br

Laura Silva de Assis

Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica e de Computação
13083-970, Campinas, SP
E-mail: laura.assis@gmail.com

Fábio Luiz Usberti

Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica e de Computação
13083-970, Campinas, SP
E-mail: fusberti@yahoo.com

***Resumo:** É proposto um método de resolução para um problema de agrupamento capacitado - PAC- baseado na metaheurística Grasp. O problema consiste em encontrar p repartições de um grafo conexo em que as arestas têm pesos ou demandas associados. As repartições devem ser tal que os desvios alcançados pela soma dos pesos das arestas que as compõem em relação a um nível alvo (ou capacidade) sejam minimizados. Além disso, é desejável que as repartições apresentem características de compacidade. Um método inspirado na metaheurística Grasp é usado para encontrar soluções que buscam minimizar uma função utilidade definida como uma composição ponderada dos desvios em relação ao alvo e de uma medida da compacidade. Resultados de experimentos computacionais realizados em uma série de exemplares gerados aleatoriamente e com controle de parâmetros comprovam a adequação e eficiência da proposta.*

Introdução

No contexto dos problemas de otimização combinatória, o PAC desperta grande interesse, quer seja pela dificuldade em se desenvolverem métodos eficientes, quer seja pelas várias aplicações ao mundo real. Dentre tais aplicações destacamos duas das mais encontradas: os problemas de distritamento político [8] e os problemas de definição de distritos comerciais [11]. No primeiro deseja-se dividir uma região geográfica em um número fixo de distritos eleitorais incumbidos de eleger um representante político. Os distritos devem possuir certas características impostas por lei eleitoral, tais como: todos os distritos devem ter o mesmo número de eleitores (os distritos são, pois, capacitados), devem ser contíguos e os mais compactos possíveis. Essa última característica busca evitar a formação de distritos com formas geográficas manipuladas de modo a favorecer indevidamente certos partidos, candidatos ou agrupamentos sociais. No distritamento com objetivos comerciais, os problemas mais comuns referem-se à repartição de regiões em territórios que devem ser designados para equipes de vendedores ou para equipes que desempenham algum serviço público ou privado como entrega de correspondência, leitura de medidores, etc. Os distritos devem também ter certas características tais como carregamento semelhante, contigüidade e compacidade.

Os problemas de agrupamento com as características acima descritas são reconhecidamente problemas complexos de otimização combinatória. Embora haja propostas de métodos exatos [7], eles têm aplicação limitada a problemas de pequena dimensão. Aplicações do mundo real usam em geral metaheurísticas como os algoritmos genéticos [9, 10], busca tabu [6, 8] ou Grasp [1, 11]. Os enfoques diferem também quanto ao número de critérios a otimizar, dando maior ou menor relevância a cada um dos muitos objetivos envolvidos nesse tipo de problema. Dependendo do caráter da decisão a tomar, pode ser impositiva a necessidade de ser apresentada

ao decisor uma curva aproximada de Pareto, o que implica em se usar uma técnica multicritério para solução [2].

Neste artigo é apresentada uma nova abordagem Grasp para um determinado PAC em que se deseja otimizar a soma ponderada de dois critérios. Um deles requer que os agrupamentos sejam os mais homogêneos possíveis em relação a duas atividades associadas às arestas do grafo: a demanda por um serviço executado ao longo das arestas e o número de clientes que demandam o serviço. Além desse critério quer-se que a forma geográfica de cada agrupamento seja a mais compacta possível de modo a facilitar a operacionalidade do serviço requerido.

Modelagem matemática

O PAC focado neste artigo consiste em uma entrada que é um grafo não-orientado conexo $G(V,E)$ onde o conjunto V é constituído de n vértices, cada qual associado a uma unidade básica geográfica bem definida da região sob estudo e E é um conjunto de arestas unindo cada par i, j de nós adjacentes pertencentes a V . Os nós $i \in V$ têm coordenadas (x_i, y_i) associadas e um conjunto de atividades $a \in A$ tais como demanda, tempo de atendimento, número de eleitores, etc. com nível de atividade conhecido e dado por inteiros w_i^a . No caso em estudo, $|A| = 2$. As arestas $(i, j) \in E$ podem também ter atividades associadas que devem ser convertidas para atividades nos nós dado que o problema e a metodologia propostos neste artigo restringem sua aplicação a atividades associadas apenas aos nós. Quer-se achar um dado número de agrupamentos (territórios ou distritos) $1 < p < n$ que seja uma partição conexa $V_k \subset V, k = 1, 2, \dots, p$ e que garanta que cada nó i seja designado a apenas um agrupamento. Deseja-se que os agrupamentos sejam os mais compactos possíveis, garantindo que deslocamentos dentro de cada agrupamento sejam minimizados. Além disso, o problema requer que os agrupamentos sejam os mais balanceados possíveis em relação às atividades a , ou seja, o nível de cada atividade $a \in A$ de cada agrupamento V_k , dado por $w^a(V_k) = \sum_{i \in V_k} w_i^a$, deve ser a mais próxima possível do nível

médio da atividade, calculado como $\mu^a = \sum_k w^a(V_k) / p$. Tal nível médio pode ser visto como uma capacidade máxima ou como um alvo a ser atingido.

Entretanto, devido ao caráter discreto do problema, é bastante improvável que se obtenham agrupamentos perfeitamente balanceados. Para se poder controlar o nível de desbalanceamento, são criadas restrições que limitam superior e inferiormente o nível de cada atividade a em relação ao seu nível médio μ^a ,

$$(1 - \tau^a) \mu^a \leq w^a(V_k) \leq (1 + \tau^a) \mu^a,$$

onde $0 \leq \tau^a \leq 1$ é uma tolerância relativa à atividade a . A partir do conceito de desbalanceamento é possível definir uma função que mede o nível de inviabilidade da atividade a em cada agrupamento V_k como sendo

$$g^a(V_k) = (1/\mu^a) \max \{ w^a(V_k) - (1 + \tau^a) \mu^a, (1 - \tau^a) \mu^a - w^a(V_k), 0 \}.$$

Define-se então o nível de inviabilidade alcançada por uma partição $S = \{V_1, \dots, V_k\}$ como sendo a soma das inviabilidades dos agrupamentos

$$G(S) = \sum_{k=1}^p \sum_{a \in A} g^a(V_k).$$

O requisito de compacidade exigido para cada agrupamento pode ser obtido por meio de uma função de dispersão do tipo $f(V_k) = \max_{i, j \in V_k} d_{ij}$, onde d_{ij} é a distância euclidiana entre cada par

(i, j) obtida a partir de suas coordenadas. Uma partição S pode então ter uma medida de compacidade dada por

$$F(S) = \left(\frac{1}{d_{\max}} \right) \max_{k=1, \dots, p} \left\{ \max_{i, j \in V_k} d_{ij} \right\},$$

onde d_{\max} é a maior distância euclidiana entre os pares de nós de V , usada para normalizar a medida de compacidade da partição. Note que maximizar a compacidade é equivalente a minimizar a dispersão.

O modelo matemático do PAC pode então ser apresentado. Inicialmente é preciso definir as variáveis

$$x_{ij} = \begin{cases} 1, & \text{se o nó } j \text{ é atribuído ao agrupamento com centro em } i, i \in V \\ 0, & \text{caso contrário} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{se o centro do agrupamento está localizado no nó } i \\ 0, & \text{caso contrário} \end{cases}$$

e o conjunto auxiliar $N^i = \{j \in V \mid (i, j) \in E \vee (j, i) \in E\}$ como sendo o conjunto dos nós adjacentes ao nó i . O objetivo é minimizar uma função utilidade $\psi(S)$ representada como a combinação convexa das funções $F(S)$ e $G(S)$, sendo $0 \leq \lambda \leq 1$ uma ponderação escolhida pelo decisor.

$$\min \quad \psi(S) = \lambda F(S) + (1 - \lambda)G(S) \quad (1)$$

$$s.a. \quad \sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (2)$$

$$\sum_{i \in V} y_i = p \quad (3)$$

$$x_{ij} \leq y_i \quad i, j \in V \quad (4)$$

$$\sum_{j \in \bigcup_{i \in D} N^i \setminus D} x_{ij} - \sum_{j \in D} x_{ij} \geq 1 - |D| \quad i \in V; \quad D \subset V \setminus (N^i \cup \{i\}) \quad (5)$$

$$x_{ij}, y_i \in \{0, 1\} \quad i, j \in V \quad (6)$$

A função objetivo (1) minimiza a soma ponderada dos critérios de dispersão geográfica e de violação das atividades, as restrições (2) garantem que cada nó j seja designado a um único agrupamento, a igualdade (3) determina que a solução tenha p agrupamentos e as restrições (4) asseguram que se um centro não for designado para o nó i , então nenhum nó pode ser designado a ele. As restrições (5), propostas por Drexl e Haase [4], garantem a conectividade dos agrupamentos enquanto (6) definem que as variáveis são binárias.

Método de resolução

O PAC representado pelo modelo (1)-(6) foi solucionado pela metaheurística Grasp [5]. Grasp tem se mostrado um enfoque de grande efetividade na resolução de problemas de otimização combinatória por aliar, num procedimento de duas fases, a simplicidade das heurísticas construtivas com o poder de prospecção das buscas locais. Em uma iteração do Grasp, alternam-se uma fase construtiva e uma de melhoria. Na primeira, uma solução factível é gerada passo a passo através da incorporação criteriosa de elementos constitutivos da solução. Na segunda fase, chamada de pós-processamento, uma busca em vizinhança é aplicada à solução da primeira fase. Ao longo do procedimento a melhor solução encontrada até o momento é mantida em S^{best} , que se constitui na saída do algoritmo.

```

function GRASP(LimitIt,  $\alpha$ ,  $\rho$ ,  $p$ )

Entrada: LimitIt := Número máximo de iteções do GRASP;
         $\alpha$  := Parâmetro de qualidade da RCL;
         $\rho$  := Parâmetro de fechamento do território;
         $p$  := Número de territórios.

Saída: A melhor solução factível encontrada -  $S^{best}$ .

1.  $S^{best} \leftarrow \{\}$ ;
2. for ( $l = 1, \dots, LimitIt$ ) do
3.    $S \leftarrow Construtivo(\alpha, \rho)$ ;
4.    $q \leftarrow |S|$ ;
5.   if ( $q \neq p$ ) then
6.      $S \leftarrow Ajustamento(S)$ ;
7.      $S \leftarrow Pós - Processamento(S)$ ;
8.     if ( $S$  melhor que  $S^{best}$ ) then
9.        $S^{best} \leftarrow S$ ;
10.  endfor;
11. return  $S^{best}$ ;
end GRASP.

```

Figura 1: Algoritmo Grasp para o PAC

A fase construtiva começa a partir da escolha de um nó semente (escolhe-se o nó de menor grau) ao qual se vai agregando um novo nó a cada iteração, até formar um agrupamento (ou território no jargão comercial) que respeite o limite imposto ao nível das atividades. O parâmetro ρ controla o momento em que um agrupamento atinge o limite com respeito a alguma atividade, o que determina o seu fechamento e enseja a formação de novo agrupamento, sendo para isso selecionado novo nó semente. Os nós candidatos são escolhidos dentre os nós vizinhos da solução corrente.

Mais especificamente a escolha recai sobre um nó de uma lista restrita de candidatos (RCL), que é um subconjunto dos melhores vizinhos. A escolha do novo elemento da solução é feita aleatoriamente dentre os elementos de RCL, de modo a conferir maior flexibilidade ao método. O parâmetro α modula o tamanho da RCL: se ele é zero a lista é unária e o elemento escolhido é a solução gulosa; na medida em que α cresce, a lista RCL aumenta, com isso aumentando a aleatoriedade da escolha. A solução construtiva pode finalizar com um número q de agrupamentos, diferente dos p requeridos. Nesse caso é preciso aplicar um procedimento de ajustamento que, ou funde o agrupamento de menor número de nós com o seu menor agrupamento vizinho, no caso de $q > p$, ou separa o maior agrupamento em dois outros conectados, no caso de $q < p$. O último procedimento é bem mais complexo do que o primeiro já que equivale a resolver o próprio problema de agrupamento com $p = 2$, mas felizmente ocorre muito raramente. Já o primeiro é muito freqüente, porém rápido.

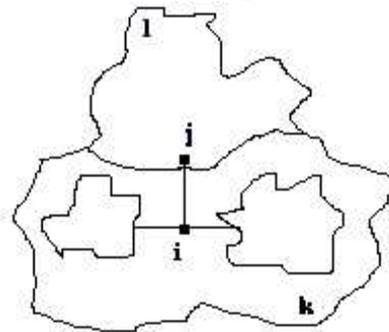


Figura 2: Nó de articulação

De posse de uma solução factível - de p agrupamentos -, a fase de pós-processamento consiste em aplicar uma busca local com o objetivo de melhorar a função $\psi(S)$. Foi utilizada uma vizinhança $N(S)$ composta por todas as soluções que podem ser encontradas a partir de S , através do movimento de um nó i de seu território corrente k , para um território vizinho l que possua um nó j tal que $(i; j) \in E$, sem criar um território descontínuo. A desconexão do grafo ocasionada por um movimento (i, k, l) é mostrada na Figura 2. Ao transferir o nó i para o agrupamento l , o movimento cria dois subgrafos disjuntos dentro do agrupamento k , o que deve

ser evitado. A ocorrência desses enclaves é um dos maiores problemas encontrados pela maioria dos métodos já propostos para solucionar o PAC. A forma encontrada para evitar movimentos que causam enclaves é testar, a cada movimento, se o nó i é um nó de articulação, ou seja, passível de gerar enclaves. Para isso basta executar uma busca em largura no território k excluído do nó i e de suas arestas incidentes, e tendo como nó origem um nó qualquer desse agrupamento. Se a remoção do nó i não desconecta o agrupamento k , ou seja, ele não é um nó de articulação, então se consegue encontrar uma árvore de caminhos entre o nó semente e todos os outros nós do território k [3].

A fase de pós-processamento segue a seguinte estratégia: primeiro tenta realizar movimentos que melhorem exclusivamente a função $G(S)$ sem piorar a função $F(S)$, e depois tenta uma série de movimentos com os objetivos trocados. Essas séries se alternam até que a busca não consiga encontrar mais nenhum movimento de melhoria. Tome, por exemplo, a série de melhoria de $G(S)$. A busca seleciona o agrupamento com maior nível de inviabilidade e, realizando uma busca nos nós de sua fronteira com outros agrupamentos, tenta realizar movimentos de forma a melhorar $G(S)$ sem piorar $F(S)$. O primeiro movimento de melhoria encontrado é o realizado. As inviabilidades dos agrupamentos envolvidos na troca são recalculadas e o processo se repete até que não exista mais nenhum movimento de melhoria. Passa-se, então, para a outra série da busca, tentando melhorar $F(S)$ sem piorar $G(S)$. Quando uma passada pelas duas séries da busca não encontra nenhum movimento de melhora, ou caso se atinja um limite predefinido de movimentos, a fase de pós-processamento da iteração se encerra. Em seguida testa-se a atualização de S^{best} e inicia-se nova iteração do Grasp, até um número máximo de iterações previamente fixado. Duas observações merecem destaque: (1) uma estrutura de dados especial guarda os nós de fronteira e realiza atualizações de forma eficiente, permitindo à busca realizar os testes rapidamente; (2) o teste de nó de articulação só é executado se o movimento for de melhoria, o que economiza tempo de execução.

Testes computacionais

O algoritmo Grasp para o PAC foi testado em uma série de instâncias geradas aleatoriamente com os n nós do grafo $G(V,E)$ sendo gerados dentro de um quadrado. Como a aplicação refere-se a um problema urbano, a geração das coordenadas dos nós procurou semelhança com cruzamentos de ruas, sem, porém, seguir uma distribuição perfeitamente uniforme. As instâncias têm quatro diferentes características:

- **Grupo:** refere-se ao tamanho do grafo; grupo 1 com 512 nós e grupo 2 com 1024 nós.
- **Família:** refere-se ao intervalo de geração da atividade 1. Utilizaram-se duas famílias de intervalos: família 1 (intervalo apertado) [16, 24] e família 2 (intervalo folgado) [12, 28].
- **Classe:** refere-se ao intervalo de geração da atividade 2. Duas classes foram definidas: classe 1 (intervalo apertado) [160, 240] e classe 2 (intervalo folgado) [120, 280].
- **Tolerância:** Define 3 diferentes valores para o parâmetro τ^a .
 - 1) tolerância apertada = 5%
 - 2) tolerância média = 10%
 - 3) tolerância folgada = 30%.

A combinação dessas características ensejou 24 tipos de instâncias, mostradas na Tabela 1.

Tabela 1: Características das instâncias de teste

Instância	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Grupo	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
Família	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	2	2
Classe	1	1	1	2	2	2	1	1	1	2	2	2	1	1	1	2	2	2	1	1	1	2	2	2
Tolerância	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3

Para cada uma das 24 instâncias foram gerados 10 exemplares, totalizando 240 instâncias usadas nos experimentos computacionais. Os resultados mostram sempre a média dos 10 exemplares. Os testes foram realizados em um PC Intel, Core2 Duo, 3 GHz, 3061 Mb de RAM, Windows Vista. De início foram feitos testes preliminares para ajustar os parâmetros do Grasp,

definidos nos valores seguintes: $\rho = 0,8$; $\alpha=0,3$; $\lambda=0,2$; limite de iterações do Grasp = 1000 e limite de iterações da busca local = 500. A Tabela 2 ilustra os resultados obtidos, sendo que S, M e I significam, respectivamente, o resultado superior (pior) encontrado entre as 10 execuções, o médio e o inferior (melhor). A última coluna traz o tempo de execução do Grasp em segundos. Note que o valor reportado para $G(S)$ corresponde à soma das inviabilidades relativas de todos os agrupamentos, ou seja, ao se tomar, por exemplo, o valor 1,80 - referente à média da soma das inviabilidades inferiores observadas para os 10 exemplares da instância 22 – está-se referindo na realidade às inviabilidades totais dos 20 agrupamentos. Isso dá uma inviabilidade média para os agrupamentos de 9% em relação à média μ das duas atividades consideradas. Como para a instância 22 a faixa de tolerância é de 5% (vide Tabela 2), deduz-se que o erro das atividades alcance em média 14% nos agrupamentos definidos pela solução S . Do mesmo modo, um erro de 0,00 em $G(S)$ deve ser interpretado como um erro médio (de cada agrupamento) cujo limite superior é da magnitude da tolerância considerada para a instância.

Tabela 2: Resultados computacionais

Inst	$\Psi(S)$			F(S)			G(S)			t(s)
	S	M	I	S	M	I	S	M	I	
1	0,47	0,27	0,06	0,56	0,36	0,28	0,51	0,25	0,00	8,75
2	0,07	0,06	0,05	0,34	0,30	0,27	0,00	0,00	0,00	10,02
3	0,33	0,15	0,06	0,33	0,31	0,29	0,34	0,12	0,00	10,33
4	0,59	0,31	0,06	0,43	0,34	0,29	0,63	0,30	0,00	8,68
5	0,08	0,06	0,05	0,42	0,31	0,27	0,00	0,00	0,00	9,71
6	0,36	0,21	0,06	0,38	0,32	0,28	0,36	0,18	0,00	10,36
7	0,49	0,34	0,06	0,43	0,32	0,28	0,51	0,34	0,00	9,02
8	0,08	0,06	0,06	0,34	0,31	0,28	0,02	0,00	0,00	9,16
9	0,39	0,17	0,06	0,45	0,34	0,30	0,41	0,13	0,00	10,57
10	0,84	0,44	0,06	0,39	0,35	0,28	0,96	0,46	0,01	9,09
11	0,19	0,08	0,06	0,55	0,36	0,28	0,13	0,01	0,00	8,49
12	0,36	0,26	0,07	0,45	0,34	0,29	0,38	0,25	0,01	10,35
13	1,85	1,65	1,23	0,37	0,32	0,26	2,35	1,99	1,45	30,47
14	0,62	0,42	0,05	0,42	0,30	0,25	0,69	0,45	0,00	31,74
15	0,09	0,06	0,04	0,27	0,24	0,22	0,05	0,01	0,00	32,14
16	1,87	1,66	1,20	0,33	0,30	0,26	2,27	2,00	1,43	30,56
17	0,53	0,30	0,09	0,39	0,32	0,29	0,59	0,30	0,04	30,44
18	0,09	0,06	0,05	0,31	0,25	0,23	0,06	0,01	0,00	32,48
19	1,75	1,55	1,34	0,34	0,29	0,27	2,10	1,86	1,42	29,68
20	0,59	0,33	0,06	0,46	0,30	0,24	0,67	0,34	0,00	28,51
21	0,08	0,05	0,04	0,26	0,24	0,22	0,04	0,00	0,00	31,37
22	1,92	1,76	1,50	0,35	0,29	0,26	2,65	2,13	1,80	30,61
23	0,59	0,38	0,15	0,35	0,30	0,24	0,66	0,40	0,11	28,94
24	0,07	0,05	0,05	0,29	0,26	0,24	0,02	0,00	0,00	32,07
Média	0,59	0,44	0,27	0,38	0,30	0,26	0,68	0,48	0,26	20,14

Os resultados mostram que, do ponto de vista da compacidade, as soluções obtidas apresentam bastante regularidade, com disparidades geográficas bem aceitáveis para os objetivos almejados na prática. Lembrando que em média tem-se aproximadamente 50 nós por agrupamento, os valores alcançados para $G(S)$ mostram-se dentro dos padrões obtidos pelos vários enfoques já desenvolvidos para o PAC, o que demonstra a dificuldade imposta para encontrar boas soluções para este difícil problema de otimização combinatória. Quanto aos tempos de CPU, a simplicidade do enfoque Grasp aliada à qualidade da busca local proposta neste trabalho conseguiram tempos bem mais baixos do que os obtidos nos demais enfoques. Finalmente é preciso salientar que uma contribuição importante desta pesquisa é a criação do teste de nó de articulação que garante soluções sem enclaves, uma dificuldade, pelo que sabem os autores, presente em todos os enfoques já propostos na literatura.

Dentre as possíveis ações para a continuidade da pesquisa uma que se poderia destacar é o investimento em uma busca local mais sofisticada do que a simples transferência de um nó para agrupamentos vizinhos. Por exemplo, uma vizinhança definida pela troca de nós entre agrupamentos vizinhos pode levar a soluções com desbalanceamentos menores e topologias melhores.

Agradecimentos

Esta pesquisa contou com apoios financeiros dos projetos 474099/2006-7 do CNPq e 2007/02604-0 da Fapesp.

Referências

1. Ahmadi, S. and Osman, I. H., Greedy random adaptive memory programming search for the capacitated clustering problem, *European Journal of Operational Research*, 162(1), pp.30-44, (2005).
2. Chinchuluun, A. and Pardalos, P. M., A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, 154(1), pp. 29-50, (2007).
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., “Algoritmos: Teoria e Prática”, Elsevier, 2002.
4. Drexler, A. and Haase, K., Fast approximation methods for sales force deployment. *Management Science*, 45(10), pp. 1307-1323, (1999).
5. Feo, T. A. and Resende, M. G. C., Greedy randomized adaptive search procedures, *Journal of Global Optimization*, 6(2), pp. 109-133, (1995).
6. Franca, P.M., Sosa, N. G. and Pureza, V.M., An adaptive tabu search approach for solving the capacitated clustering problem, *International Transactions in Operational Research* 6, 665–678, (1999).
7. Garfinkel, R.S. and Nemhauser, G.L., Optimal political districting by implicit enumeration techniques, *Management Science* 16, pp. 495–508, (1970).
8. Laporte, G., Bozkaya, B. and Erkut, E., A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1), pp. 12-26, (2003).
9. Pereira, F. T., Figueira, J. R., Mousseau, V. and Roy, B., Multiple criteria districting problems - the public transportation network pricing system of the Paris region, *Annals of Operations Research*, 154(1), pp. 69-92, (2007).
10. Ricca, F. and Simeone, B., Local search algorithms for political districting, *European Journal of Operational Research*, 189(3), pp. 1409-1426, (2008).
11. Rios-Mercado, R. Z. and Fernandez, E., A reactive Grasp for a commercial territory design problem with multiple balancing requirements, *Computer & Operations Research*, 36(3), pp. 755-776, (2009).