P-Medianas com restrição de capacidade
Relatório Parcial

Paulo Viana Bicalho[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

p.bicalho@dcc.ufmg.br

1. Introdução

Em um problema de localização deseja-se estabelecer os locais onde serão posicionadas centros de facilidades (fábricas, depósitos, hospitais, etc.) para atender, da melhor maneira possível, um conjunto espacialmente distribuído de pontos de demanda. O problema de p-medianas com restrição de capacidade é um problema clássico de localização onde o objetivo é determinar os locais de p facilidades (medianas) em um grafo com N nós de modo a minimizar a soma das distâncias entre cada nó de demanda e a mediana mais próxima. Além disto, cada nó possui uma certa demanda e a soma das demandas de cada nó j ligado a um centro de facilidade i, não pode ultrapassar a capacidade total de i. O problema da P-media capacitada é um problema NP-Difícil [1].

Formalmente, podemos descrever este problema, como um problema de otimização inteira onde: Dados um conjunto de N pontos i_s, aos quais estão associada uma demanda dem_i e as distâncias d_{ij} entre o nó de demanda i e uma facilidade j, queremos encontrar:

$$Min \sum_{i=1}^{N} \sum_{j=1}^{N} dist_{ij} \times x_{ij} \tag{1}$$

Considerando as seguintes restrições.

$$\left(\sum_{j=1}^{N} x_{ij} \right) = 1, \quad \forall i \in N \tag{2}$$

$$\sum_{j=1}^{N} x_{jj} = p \tag{3}$$

$$\sum_{i=1}^{N} dem_i \times x_{ij} \leq cap_j \times x_{jj}, \quad \forall j \in N \tag{4}$$

Onde p é o número de centros de facilidades (medianas) a serem localizados, x_{ij} representa a aresta entre os vértices i e j e assume valor 1 se o vértice i for atendido pelo centro j e 0 caso contrário. Quando $i == j$, $x_{ii} = 1$ identifica o vértice i como uma mediana.

A restrição (2) garante que, para todos os vértices, ou o vértice é uma mediana, e logo não é atendido por nenhuma outra mediana; ou ele é atendido por uma (e apenas uma) mediana. A restrição (3) garante que o número de medianas seja igual a p. Sendo dem_i a demanda do vértice i e cap_j a capacidade de atendimento do centro j, se este for escolhido como mediana, a restrição (4) garante que a capacidade de alocação de uma mediana não seja ultrapassada.

2. Trabalhos Relacionados

Devido a grande utilidade deste problema para áreas telecomunicação, transporte e distribuição, este problema foi amplamente estudado nas ultimas décadas [2]. Em [3] são apresentados dois algoritmos exatos para resolver o Problema de p-Medianas e ambos apresentaram bons resultados para as diversas instâncias testadas.

Devido a complexidade do problema, várias heurísticas foram desenvolvidas e, atualmente, técnicas como *simulated annealing* [4] e *tabu search* [5] são amplamente utilizadas para solucionar o problema.

3. Conjunto de Dados

As bases de dados que serão utilizadas para o desenvolvimento deste trabalho estão disponíveis online através dos links:

1. `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedcapinfo.html`
2. `http://www.lac.inpe.br/~lorena/instancias.html`

O primeiro conjunto de dados foi utilizado em [1] e o segundo conjunto apresenta instâncias de tamanho grande (até ordem de milhares). Em ambos os sitios, as soluções ótimas para os problemas são conhecidas.

4. Modelagem

Pela definição, o problema das P-Medianas com restrição de capacidade é diretamente modelado através de um grafo ponderado G(V,E) onde

- Cada vertice $v \in V$ representa um ponto de demanda ou de oferta;
- Cada aresta ponderada $e_{ij} \in E$ indica a distância entre dois i e j;

A partir desta modelagem, o problema se resume a encontrar um subconjunto ótimo de vertices $V_p \subset V$ de tamanho p que representa as medianas que atendem as demandas de todos os nós e minimizam o somatorio das distancias

A figura **??** apresenta uma instancia deste problema bem como uma solução ótima.

5. Baseline

Vamos utilizar como baseline o algoritmo exato utilizando o solver Gurobi. A implementaçao é bastante simples, uma vez que o solver só necessita da função objetivo e das restrições (todas definidas e descritas na seção 1 desta documentação).

5.1. Resultados Preliminares

Testamos a solução exata com 4 das bases que serão utilizadas. Em nenhuma delas o solver terminou a execução em tempo habil (- de 12 horas) e portanto a execução foi encerrada e o melhor valor encontrado até o momento foi selecionado

O gurobi foi executado em uma máquina com sistema operacional Ubuntu 12.04, com processaodr Intel(R) Core(TM) i3-2375M CPU @ 1.50GHz e 4GB de RAM.

A tabela **??** sumariza os resultados encontrados:

Para a conclusão do trabalho o solver será executado em uma máquina com maior poder de processamento e durante mais tempo.

Table 1. Resultados Preliminares Gurobi.

Base	Gurobi (12h)	Ótimo
SJC1	18.368,78	17.246,53
SJC2	37.200,26	33.225,88
SJC3b	47.934,73	40.635,80
SJC4a	76.477,62	61.843,23

6. Heurísticas - Algoritmo Genético

Neste trabalho, propomos solucionar o problema das P-medianas capacitadas utilizando um algoritmo genético (GA). Uma modelagem simples do problema pode ser realizada da seguinte forma:

- Cada indivíduo é uma solução do problema;
- Cada indivíduo possui p genes onde cada gene indica uma mediana;
- A soma das capacidades dos nós em cada gene deve ser capaz de suprir a soma das demandas de todos os outros nós (garante viabilidade da solução);
- Os nós são atribuídos as medianas seguindo duas possíveis políticas definidas a seguir.

Através desta modelagem, o algoritmo genético será capaz de encontrar uma solução próxima da ótima para o problema. Na modelagem acima, cada individuo indica quais nós serão utilizados como medianas, mas a politica de atribuição dos nós de demandas a estas medianas ainda deve ser definida.

6.1. Política de Atribuição 1

A primeira politica a ser testada assume que o vértice que esta mais longe de alguma das medianas tem a maior capacidade de prejudicar a solução, uma vez que devido as restrições de capacidade este vértice pode ser atribuido a mediana mais longe dele, e desta forma damos prioridade a este vértice atribuindo-o a mediana viável mais próxima.

Algorithm 1: Politica de atribuição 1

```
M ← Medians()

for cada vertice v_i ∈ V − M do
    distances[v_i] ← [dist(v_i, m_1), dist(v_i, m_2), ..., dist(v_i, m_p)]
end

VPrioridade ← vertices ordenados reversamente utilizando max(distances[v_i]
como critério

for v_p ∈ Vprioridade do
    atribui v_p para a mediana mais próxima que seja capaz de suprir sua demanda
end
```

6.2. Politica de Atribuição 2

Esta política assume que os vertices que possuem maior demanda são mais problematicos durante a atribuição e por isso devem ter prioridade e serem atribuidos para medianas mais próximas.

Algorithm 2: Politica de atribuição 2

```
M ← Medians()

VPrioridade ← vertices ordenados reversamente utilizando demanda como critério;

for v_p ∈ Vprioridade do
    atribui v_p para a mediana mais próxima que seja capaz de suprir sua demanda
end
```

6.3. Fitness

A *Fitness* mede a qualidade do individuo e indica quão bom é a solução representada por este. O cálculo desta requer duas etapas:

1. associar os vértices, utilizando alguma política, às medianas do indivíduo e
2. calcular a soma das distâncias entre vértices e medianas.

Dessa forma, a *Fitness* fica sendo a soma das distâncias e quanto menor seu valor, melhor será o indivíduo e a solução que ele representa.

6.4. Operadores Genéticos

6.4.1. Mutação

Será utilizada mutação uniforme, no qual cada gene tem uma probabilidade Mut_{prob} de ser mutado. Caso um dado gene v_p (id de vértice) de um indivíduo I seja mutado, o operador genético irá selecionar o id de um vértice $v_i \in V$ onde $v_i \neq v_p$ e $v_p \notin I$.

6.4.2. Crossover

O crossover escolhido, assim como a mutação, será o crossover uniforme. Esse operador é binário, requerendo 2 indivíduos, I_1 e I_2 de tamanho p para serem cruzados. Cada posição i dos indivíduos pais tem probabilidade de $Cross_{prob}$ de ser trocada. Caso seja, os genes $I_1[i]$ e $I_2[i]$ serão trocados. Ao final desse processo dois novos indivíduos são gerados.

6.5. Seleção

Utilizaremos seleção por torneio. Este tipo de seleção consiste em escolher k indivíduos da população, de forma aleatória. Aquele com a maior *fitness* irá gerar descendentes. Esse processo é executado duas vezes, para que possamos obter dois pais.

Este tipo de seleção permite a regulação da pressão seletiva por meio do parâmetro k. Quanto menor o k, mais diversa será a escolha dos pais. Quanto maior o k, mais rápido a população converge.

6.6. Substituição de indivíduos

A substuição dos indivídus ocorre da seguinte forma: Cada par de pais geram dois filhos totalizando 4 individuos no final (os dois pais originais e os dois filhos). Deste conjunto de 4 individuos, somente os dois melhores passam para próxima geração.

7. Experimentos

Foram propostas duas heurísticas baseadas em GA. A diferença entre elas está na ordem de prioridade de atribuição dos nós do grafo às medianas encontradas. A primeira da prioridade a ponto que estão mais distantes das medias e a segunda a pontos com maior demanda.

Ambas as soluções possuem os mesmos parametros:

- Numero de Gerações;
- Tamanho da população;
- Probabilidade de mutação;
- Probabilidade de crossover;
- Tamanho do torneio;

Para avaliar o impacto de cada um dos parametros vamos realizar um projeto fatorial 2kr. Os valores limites de cada parametro serão:

- Numero de Gerações: 20 - 100;
- Tamanho da população: 500 - 2000;
- Probabilidade de mutação: 0.45 - 0.9;
- Probabilidade de crossover: 0.2 - 0.5;
- Tamanho do torneio: 2 - 20;

O projeto fatorial 2kr irá dizer se existe interação entre os parametros e se um projeto simples (variar um parametro e fixar os outros) pode ou não ser feito.

Encontrado a melhor configuração de parametros vamos executar as heuristicas para todas as bases, realizar um teste-t para comparar a eficácia e tempo de execução das duas e do algoritmo exato.

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] P. Mirchandani and R. Francis, *Discrete location theory*, ser. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1990. [Online]. Available: http://books.google.com.br/books?id=_GFRAAAAMAAJ

[3] A. Ceselli, "Two exact algorithms for the capacitated p-median problem." *4OR*, vol. 1, no. 4, pp. 319–340, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/4or/4or1.html#Ceselli03

[4] T. V. Levanova and M. A. Loresh, "Algorithms of ant system and simulated annealing for the p-median problem," *Autom. Remote Control*, vol. 65, no. 3, pp. 431–438, Mar. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:AURC.0000019375.23201.c1

[5] K. Rosing, C. ReVelle, E. Rolland, D. Schilling, and J. Current, "Heuristic concentration and tabu search: A head to head comparison," *European Journal of Operational Research*, vol. 104, no. 1, pp. 93 – 99, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037722179700310X