Pseudo Código

Heuristica principal:

initializeData();

CurrentCost : generateStartSolution();

BestCost : CurrentCost;

while itCount < maxRuns do

// Close facilities

for i : 1 to swapCount do

Drop();

// Open facilities

for i : 1 to swapCount do

ADD();

CheckTabuListSize();

if CurrentCost < BestCost then

BestCost : CurrentCost;

itCount : 0;

itDepth : 0;

else

if maxDepth = 0 then

itCount : itCount + 1;

else

itDepth : itDepth + 1;

if itDepth = maxDepth then

itDepth 0;

itCount : itCount + 1;

CurrentCost : BestCost;

swapCount : Max(1; Min(itCount \* swapModify; p - 2);

dontStop : true;

while dontStop : true

for i : 1 to N

if solMelhor[i] : true

for j : 1 to N

if solMelhor[j] : false

dontStop : SWAP(i,j);

if dontStop : true

break;

else

continue;

if dontStop : true

break;

return S

end function

ADD:

Para todo nó não presente na solução:

Calcula a solução se esse nodo estiver presente

Guarda solução calculada em um vetor

Para o menor valor deste vetor, adiciona nodo correspondente à solução e à lista tabu

Se não há melhora e todos os valores do vetor forem Zero

Escolhe um nodo aleatório para ser adicionado

SWAP:

Troca dois nodos e atualiza a resposta

Se melhor solução

Retorna true

Senão

Destroca

Retorna false

Drop antigo:

Estima uma mediana a ser retirada a qual seria menos confiável, para avaliar confiabilidade ele calcula a 1-mediana correspondente. O maior valor obtido do calculo das 1-medianas implica no nodo menos confiável, retirando-o da solução e atualizando a solução.

O(2pn)

Drop antigo faz uma atualização de resposta

Drop novo:

Garante que a mediana retirada gera a menor resposta resultante, já que calcula todas as possíveis respostas resultantes da retirada exclusiva de cada mediana na solução. A menor solução gerada indica a mediana a ser retirada. Assim atualiza-se a resposta.

O(p²n)

Obs.: Cada atualização de resposta é O(pn) e cada calculo de 1-mediana é O(n), por isso, Drop novo é mais complexo, pois faz p atualizações de resposta, enquanto o Drop antigo faz p 1-medianas e uma atualização de resposta.

Esse algoritmo é idêntico ao da referencia: <http://iwi.econ.uni-hamburg.de/IWIWeb/Uploads/Team/JP/MaroszekRettig-p-median-2008.pdf>

Nota-se somente que as estruturas de dados não são as mesmas, a heurística do nosso grupo utiliza somente vetores, uma tabela, variáveis para guardar as solução melhor e solução atual e acumuladores.

Devido à alta complexidade computacional do problema, muitas vezes este não pode ser resolvido em tempo aceitável. Uma forma alternativa de obter uma solução válida é a utilização de heurísticas. Heurística não resolver o problema da forma clássica de testar todas as soluções possíveis.

Eles inteligente tentar encontrar uma solução próxima da melhor forma possível rapidamente. A desvantagem das soluções obtidas usando a heurística é que não há maneira de medir a qualidade da solução sem ter o problema resolvido utilizando um método clássico antes. Heurística Meta fornecem quadros para a construção de heurísticas que são capazes de resolver diferentes problemas computacionais. Para o problema de p-Mediana heurística meta vários têm sido propostos. Voss [18] e Rolland et al. [17] discutiu busca tabu. Busca na vizinhança variável (VNS) tem sido proposto por Hansen e Mladenovic [6] e foi estendido para pesquisa variável bairro de decomposição (VNDS) por Hansen et al. [7]. Simulated Annealing (SA) tem sido discutido por Murray e da Igreja [14], Chiyoshi e Galvão [3] e Levanova e Loresh [11]. Uma abordagem multi-partida híbrido é proposto por Resende e Werneck [16] que combina características de heurísticas meta outros como caminho de religação como na ENV e várias gerações como em algoritmos genéticos com várias pesquisas locais a partir de soluções iniciais diferentes. Uma comparação detalhada dos métodos para resolver o problema de p-Mediana foi dada por Mladenovic et al. [13].

O algoritmo apresentado neste trabalho faz uso da heurística busca tabu meta  
introduzida por Glover [4, 5]. A idéia básica por trás de busca tabu é evitar óptimos locais  
lembrando-se das soluções anteriores e proibi-los em pesquisas subseqüentes. a idéia  
para as atualizações de custo incremental foi proposto por Michel e Van Hentenryck [12].  
A nossa contribuição para o algoritmo consiste no uso de instalações rankings previamente classificados  
que permitem a rápida localização de instalações abertas nas imediações do cliente.  
Mostramos também um método de dois sentidos que aumenta a precisão do algoritmo: Quando a al-  
gorithm fica preso em áreas de menor qualidade que a primeira gota / adicionar várias instalações em um  
iteração. No caso em que este procedimento não conduzir a um resultado novo melhor em um determinado número  
de iterações caímos de volta para o melhor resultado obtido antes. A lista tabu aqui impede  
o algoritmo de inspecionar o caminho mesma pesquisa novamente.

O algoritmo apresentado neste trabalho faz uso da meta heurística busca tabu. A idéia básica por trás da busca tabu é evitar ótimos locais, lembrando as soluções anteriores e proibi-las de aparecer em buscas subsequentes.

O algoritmo utiliza também um método para aumentar sua precisão: Quando o algoritmo fica preso em áreas de menor qualidade, primeiramente são testadas várias configurações de medianas. No caso em que este procedimento não conduzir a um resultado melhor em um determinado número de iterações, o melhor resultado obtido antes é restaurado. Dessa forma a lista tabu evita que o algoritmo de pesquise o mesmo ramo da árvore de busca novamente.

A idéia básica por trás de busca tabu é evitar óptimos locais  
lembrando-se das soluções anteriores e proibi-los em pesquisas subseqüentes.

Mostramos também um método de dois sentidos que aumenta a precisão do algoritmo: Quando a algorithm fica preso em áreas de menor qualidade que a primeira gota / adicionar várias instalações em um  
iteração. No caso em que este procedimento não conduzir a um resultado novo melhor em um determinado número  
de iterações caímos de volta para o melhor resultado obtido antes. A lista tabu aqui impede  
o algoritmo de inspecionar o caminho mesma pesquisa novamente.