MÉTODOS DE SOLUÇÃO PARA O PROBLEMA DA LOCALIZAÇÃO-ALOCAÇÃO DE RECURSOS DE SAÚDE [1]

Tailini Schultz[2], Marcelo de Souza[3]

[1] Vinculado ao projeto intitulado "Projeto automático de algoritmos".
[2]Acadêmica do Curso de Engenharia de Software – CEAVI – Bolsista PROIP/UDESC.
[3] Orientador, Departamento de Engenharia de Software – CEAVI – macelo.desouza@udesc.br.

1. Contexto

A localização e alocação de recursos de saúde é um problema de otimização combinatória que busca determinar a localização de instalações e a atribuição de pacientes a essas instalações, de modo que a distância total entre os pacientes e as instalações seja minimizada. Para isso, é considerada a demanda que cada localidade possui e a capacidade de atendimento de cada instalação. Este estudo se divide em três etapas: i) a formulação matemática do problema e a exploração de um *solver* matemático para resolvê-lo de forma exata; ii) o desenvolvimento da meta-heurística *iterated greedy* (IG) para a solução de instâncias maiores, cuja solução exata é impraticável; e iii) um estudo de caso com instâncias do estado de Santa Catarina.

2. Modelagem matemática

O modelo matemático proposto é baseado na versão capacitada do problema das p-medianas (Hakimi, 1964). Sua modelagem é apresentada a seguir.

minimiza

$$\sum_{i,j\in[n]} d_{ij} x_{ij} + M\lambda \qquad (1)$$

sujeito a

$$\sum_{i\in[n]} x_{ij} \leq 1, \qquad j\in[n] \qquad (2)$$

$$y_i \geq x_{ij}, \qquad i,j\in[n] \qquad (3)$$

$$\sum_{i\in[n]} y_i = p, \qquad (4)$$

$$\sum_{j\in[n]} P_j x_{ij} \leq C\, y_{ij}, \qquad i\in[n] \qquad (5)$$

$$\sum_{i,j\in[n]} x_{ij} = n - M, \qquad (6)$$

$$x_{ij} \in \{0,1\} \qquad (7)$$

$$y_i, M \in \mathbb{Z} \qquad (8)$$

A função objetivo (1), a ser minimizada, calcula o somatório da distância entre cada ponto de demanda e o recurso de saúde alocado a ele. Além disso, é somada uma penalização para os pontos de demanda sem atendimento, que corresponde à soma das distâncias entre cada ponto e seu ponto mais distante. Dessa forma, a pior solução que atende a todos os pontos será melhor que a melhor solução que deixa algum ponto desatendido. A restrição (2) restringe a alocação de pacientes a no máximo um recurso. A restrição (3) garante a existência de recursos suficientes para atender todas os pacientes a ele alocados. A restrição (4) determina que exatamente *p* recursos sejam localizados. A restrição (5) assegura que a capacidade de cada recursos deve ser respeitada. A restrição (6) determina que o total de pontos atendidos deve ser igual ao total de pontos menos a quantidade de pontos não atendidos. Finalmente, as restrições (7) e (8) definem os domínios das variáveis de decisão.

3. Solução heurística

Como instâncias do mundo real tendem a ser grandes, i.e. com muitos pontos de demanda, a solução exata requer um tempo impraticável. Para resolver essas instâncias, foi implementada a metaheurística *iterated greedy* (Algoritmo 1), proposta por Ruiz e Stutzle (2007). Esse algoritmo explora o espaços de soluções ao destruir parcialmente as soluções encontradas e reconstruí-las, gerando novas soluções. Inicialmente, o algoritmo gera uma solução inicial aleatória e, por *N* iterações, a solução atual é parcialmente destruída e então reconstruída. Caso a busca local esteja habilitada, a solução sofre um processo de refinamento. Além disso, em caso de estagnação o algoritmo pode ser reiniciado. Por fim, a solução a ser usada na próxima iteração é escolhida conforme o critério de aceitação: solução atual (S’) ou incumbente (S*).

Algoritmo 1. Metaheurística *Iterated Greedy*.

procedimento Iterated Greedy (N, α, D)
    S ← SoluçãoAleatória()
    S* ← S
    para i ← 1 **até** N **faça**
        se reinicialização está habilitada **e** iterações estão estagnadas **então**
            S ← SoluçãoAleatória()
        fim se
        S’ ← Destruição(S, d_1, d_2)
        S’ ← Construção(S’, α, β)
        se busca local está desabilitada **então**
            S’ ← BuscaLocal(S’)
        fim se
        S ← Aceitar(S’, S*)
    fim para
    retorna S*
fim procedimento

Na etapa de construção aleatória, a localização dos recursos de saúde e a alocação dos paciente são realizadas aleatoriamente. Já na reconstrução, os recursos são localizados de forma semi-gulosa e então atribuídos a pacientes. Durante a seleção semi-gulosa, os pontos são classificados de acordo com sua distância para todos os outros pontos ainda não atendidos, e um ponto aleatório é escolhido entre os α melhores. Esse processo é repetido até todos os recursos serem localizados. No caso da alocação de pacientes, os pontos são classificados de acordo com a proximidade com os recursos. Repetidamente, um dos β melhores pontos é escolhido e alocado ao recurso mais próximo. Esse processo é repetido até que todos os pacientes sejam alocados, ou até não haver mais recursos com capacidade disponível para atender pacientes.

A etapa de destruição leva em consideração os parâmetros d_1 *e* d_2. O primeiro deles determina o percentual de recursos que serão removidos da solução, enquanto o segundo define o percentual de pontos de demanda que serão desalocados. Esse processo pode ocorrer de forma totalmente aleatória ou seguindo uma estratégia guiada, que consiste em remover com maior probabilidade os recursos de saúde com maior capacidade ociosa, e desalocar com maior probabilidade pontos de demanda com maior demanda.

No processo de busca local, a seleção de vizinho segue a estratégia de primeira melhoria, sendo selecionada a primeira solução que melhora o valor da função objetivo. Para gerar a vizinhança, um equipamento por vez é removido da solução e alocado a cada outro ponto. Os pontos que ficaram sem atendimentos são alocados aleatoriamente aos recursos de saúde existentes.

4. Resultados e discussões

Para o estudo de caso, foi realizada a geração de instâncias baseadas nos dados do sistema público de saúde do estado de Santa Catarina, com o objetivo de comparar os resultados obtidos através da solução exata do modelo matemático e do algoritmo *iterated greedy*. As instâncias geradas consideram as 295 cidades do estado, dividindo-as nas seis mesorregiões: Grande Florianópolis, Norte, Serrana, Sul, Vale do Itajaí e Oeste. No total, há uma instância para cada região e uma contendo todas as cidades do estado.

Para obter a melhor combinação de componentes e parâmetros para o algoritmo IG, foi explorado o configurador *irace* (Lopez-Ibáñez et al., 2016). O *irace* executa o algoritmo de *racing* de forma iterada, avaliando diferentes combinações de parâmetros em instâncias de treinamento, descartando aquelas de desempenho inferior e ajustando os modelos de amostragem para a geração de novas configurações. Ao final de 2000 experimentos, a melhor configuração encontrada pelo configurador usa a estratégia de destruição guiada, aceitação da solução incumbente, busca local habilitada, reinício ativado, $r = 0{,}10$, $d_1 = 0{,}05$, $d_2 = 0{,}02$, $\alpha = 0{,}57$ e $\beta = 0{,}77$. Esses valores representam uma baixa aleatoriedade para a destruição e uma maior aleatoriedade para as reconstruções.

O algoritmo IG sob a configuração obtida foi avaliado com 10 replicações, com critérios de parada definidos tanto pelo número máximo de iterações (igual a 1000) quanto pelo tempo máximo de execução (600 segundos). A Tabela 1 apresenta as informações das instâncias, os valores ótimos encontrados ao resolver o modelo matemático usando o *solver* GLPK (também com 600 segundos de tempo máximo de execução), e os resultados obtidos pelo IG. Observa-se que a abordagem exata (MIP) alcançou a otimalidade em 5 das 7 instâncias, que são aquelas de menor tamanho. Por outro lado, apesar do algoritmo IG ter alcançado a solução ótima em somente 3 das 7 instâncias, conseguiu produzir soluções viáveis para as duas instâncias maiores em tempo razoável.

Os resultados mostram que o uso da metaheurística *iterated greedy* para a resolução do problema da localização e alocação de recursos de saúde é uma abordagem promissora e supera a abordagem exata em instâncias de maior complexidade, uma vez que é capaz de alcançar soluções viáveis em tempo sem exigir grandes recursos computacionais e de tempo. Para uma melhor compreensão da distribuição dos recursos, a Figura 1 ilustra como seria uma solução para a instância da Grande Floripa. Nessa solução, as localidades são representadas por pontos azuis, sendo que aqueles que possuem equipamento são representados por estrelas. A distribuição dos equipamento às cidades é evidenciada pelas retas ligando os pontos e as estrelas.

Tabela 1. *Resultados obtidos na execução do MIP e do algoritmo IG.*

Instância			MIP	*Iterated greedy* (IG)			
Mesorregião	*N*	*p*	Ótimo	Melhor	Média	Melhor Tempo	Tempo médio
Grande Floripa	21	12	**244**	**244**	253.5	3.1	7.2
Norte	26	12	**388**	**388**	398.9	3.6	12.2
Serrana	30	4	**1460**	**1460**	1490.2	3.6	6.0
Sul	46	10	**628**	647	659.8	20.1	37.3
Vale do Itajaí	54	16	**707**	710	731.0	51.0	73.7
Oeste	118	12	-	**3439**	3499.7	290.6	379.5
SC (todos)	295	62	-	**6294**	6486.5	423.1	600.0

Figura 1. *Representação visual de uma possível solução para a instância da Grande Florianópolis*

Palavras-chave: Problemas de localização. Otimização combinatória. Automação de algoritmos.

Referências

HAKIMI, S. Louis. **Optimum locations of switching centers and the absolute centers and medians of a graph**. Operations research, v. 12, n. 3, p. 450-459, 1964.

RUIZ, Rubén; STÜTZLE, Thomas. **A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem**. European journal of operational research, v. 177, n. 3, p. 2033-2049, 2007.

LÓPEZ-IBÁÑEZ, Manuel et al. **The irace package: Iterated racing for automatic algorithm configuration**. Operations Research Perspectives, v. 3, p. 43-58, 2016.