

Posicionamento de equipes policiais

Maria Eduarda Oliveira Miranda

Rafael Victor Costa Braz

Dezembro de 2019

1 Introdução

O problema do posicionamento de equipes policiais, formulado em (MENDES; SANTOS; GONÇALVES, 2014), consiste em posicionar um número fixo de unidades policiais ao longo de uma região de modo a adquirir a cobertura máxima da mesma, considerando um tempo máximo de ação TMAX. São definidas várias unidades policiais e cada uma com suas características, como velocidade, de modo que um policial em uma viatura cobre uma região maior do que um policial andando. Na modelagem do problema original, o objetivo era maximizar a função Z , figura 1, respeitando as restrições 2, 3 e 4.

$$Z = \sum_{r=1}^{|E|} l_r * a_r$$

Figura 1: Função objetivo do problema original.

$$\sum_{l=1}^{|V|} x_{il} \leq 1, \quad i = 1, 2, \dots, |U|$$

Figura 2: Restrição 1 do problema original.

$$a_r = \sum_{i=1}^{|U|} \sum_{l=1}^{|V|} \sum_{m=1}^{|Q|} (p_{rlm} * x_{il} * q_{im}), \quad r = 1, 2, \dots, |E|$$

Figura 3: Restrição 2 do problema original.

Sendo l_r o lucro de se cobrir a aresta r , x_{il} 1 caso a unidade i esteja no local l , e 0 caso contrário, a_r 1 caso a aresta r é coberta por um policial em um tempo TMAX, e 0 caso contrário, a'_r sendo o mesmo que a_r porém considerando um tempo $2 * TMAX$,

$$a'_r = \sum_{i=1}^{|U|} \sum_{l=1}^{|V|} \sum_{m=1}^{|Q|} (p'_{rlm} * x_{il} * q_{im}), \quad r = 1, 2, \dots, |E| \quad a'_r = \sum_{r=1}^{|E|} a'_r = |E|$$

Figura 4: Restrições 3 e 4 do problema original.

q_{im} 1 caso a unidade i é do tipo m , e 0 caso contrário, p_{rlm} se uma unidade do tipo m pode cobrir uma aresta r , se for posta em um vertice l , e p'_{rlm} o mesmo que p_{rlm} porém considerando um tempo de $2 * TMAX$.

O presente trabalho propõe soluções para uma versão simplificada do problema, onde só há um tipo de unidade, considerada uma viatura policial, considerando também uma função objetivo que busca minimizar a soma do menor tempo possível de uma viatura chegar em todos os locais da região 5 e respeitando somente a primeira restrição do problema original.

$$Minimizar Z = \sum_{i=1}^{|V|} \min_u (imp_i * t_{ui})$$

Figura 5: Função objetivo do problema simplificado.

2 Métodos

Para este problema, foram propostas duas heurísticas que serão descritas nesta seção: ILS e Scatter Search. Em ambas as heurísticas foi utilizado o mesmo mecanismo de busca local, descrito na seção 2.1.

Para a definição dos parâmetros de cada heurística, foi utilizado o Iterated Race for Automatic Algorithm Configuration (IRACE¹), um pacote que busca encontrar a melhor configuração de parâmetros para algoritmos de otimização de maneira automatizada.

2.1 Busca Local

Para a busca local foi utilizado o método de first improvement, onde um policial é realocado, iterativamente, para todos os locais que não estão cobertos por outro policial. O processo é repetido até que se ache uma solução que melhore a solução atual, ou até que todas as unidades tenham passado por esse processo sem que haja melhora na solução inicial.

2.2 Iterated Local Search

A primeira heurística proposta foi baseada na metaheurística Iterated Local Search (ILS) 1, que consiste em realizar uma busca local a partir de uma perturbação sobre o ótimo local, e esse resultado passa por uma fase de aceitação, que escolhe se ele deve ser passado para frente, ou se deve ser retornado ao estado da solução antes da perturbação. Ela é caracterizada pelo fato de explorar mais o espaço de busca sem utilizar de reinícios aleatórios (HOOS; STÜTZLE, 2000).

¹ <http://iridia.ulb.ac.be/irace/>

O método de perturbação desenvolvido escolhe aleatoriamente um subconjunto dos policiais definido por `disturb_size`, e os realocam aleatoriamente para locais onde não há nenhum outro policial alocado. Já o critério de aceitação é definido por uma probabilidade de `accept_chance` de aceitar soluções piores do que a solução antes da perturbação, e 100% de chance de aceitar soluções melhores.

Algoritmo 1: Iterated Local Search

ILS (*disturb_size*, *accept_chance*):
 $s \leftarrow busca_local(s_0);$
enquanto não alcançar o critério de parada **faça**
 $s' \leftarrow perturbação(s_*, disturb_size);$
 $s'_* \leftarrow busca_local(s');$
 $s_* \leftarrow aceitar(s_*, s'_*, accept_chance);$
fim
retorna *Melhor solução encontrada*

Com os resultados do IRACE, foi utilizado, em 10 iterações, o tamanho da perturbação (*disturb_size*) de 0.1 e a chance de aceitação (*accept_chance*) de 0.2.

2.3 Scatter Search

A segunda heurística proposta é baseada na metaheurística Scatter Search, em 2, cujo princípio consiste em combinar soluções de um conjunto de referência (RefSet) para a construção de novas soluções (GLOVER; LAGUNA; MARTI, 2003). Seu design, geralmente, é baseado no fluxo dos métodos apresentados na figura 6.

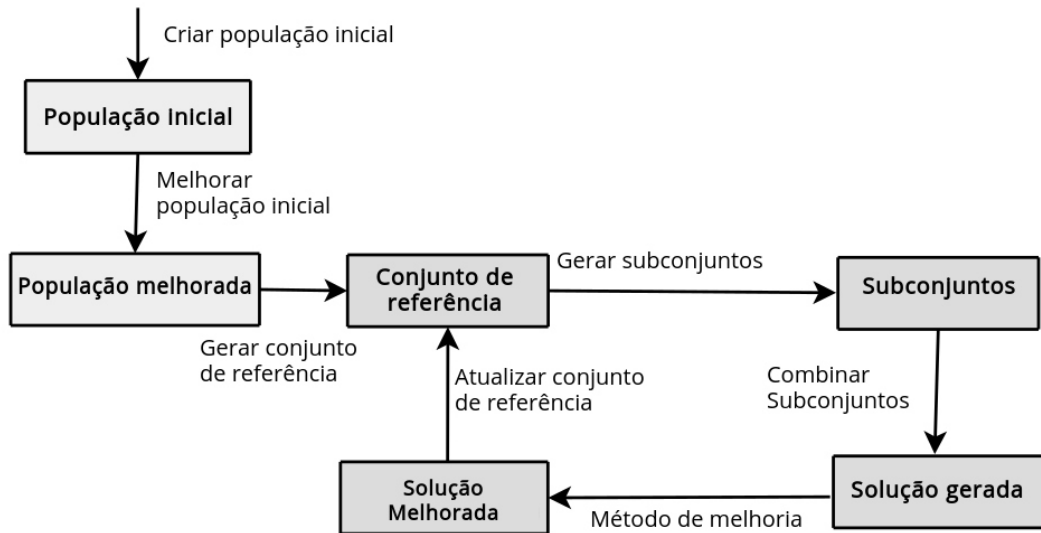


Figura 6: Componentes de busca de algoritmos scatter search (TALBI, 2009)

A construção de cada população inicial nesta heurística foi realizada da seguinte forma: para a geração de cada indivíduo, um dos algoritmos descritos abaixo é escolhido de forma aleatória: Os algoritmos construtivos utilizados seguem as seguintes propostas:

- Algoritmo 1: Posicionar as unidades nos vértices mais importantes do grafo ².
- Algoritmo 2: Posicionar as unidades nos vértices adjacentes aos vértices de maior importância.
- Algoritmo 3: Posicionar as unidades nos vértices de maior grau.

Com os critérios de escolha acima, os algoritmos são aleatorizados escolhendo uma dentre uma lista de tamanho *sizeRCL* das melhores opções a cada passo.

Partindo da população inicial, cada indivíduo foi melhorado utilizando a busca local descrita anteriormente e, com a população melhorada, foi construído um conjunto de referência com os *n* melhores indivíduos da população (intensificação) e *m* indivíduos aleatórios (exploração).

Ambos os critérios de parada utilizados foram baseados no número de iterações realizadas, e as soluções foram combinadas a partir de conjuntos de 2 a 2 indivíduos, utilizando o início e fim de cada um destes, removendo repetições.

Algoritmo 2: Scatter Search

```

ScatterSearch(SizeRCL):
  gerarPopulaçãoInicial(C);
  melhorarIndividuos();
  atualizarRefSet(populacaoInicial);
  enquanto não alcançar o critério de parada faça
    gerarSubconjuntos();
    enquanto não alcançar o critério de parada 2 faça
      combinarSoluções();
      melhorarIndividuos(individuosCombinados);
    fim
  atualizarRefSet(novaPopulacao);
fim

```

Com o resultado do IRACE, foram utilizados como parâmetros: uma população de tamanho 30, a lista de candidatos *SizeRCL* = 0.05, o conjunto de referência com *n* = 9 e *m* = 6. Além disso, os critérios de parada utilizados foram com 10 iterações em cada, buscando melhorar a velocidade do algoritmo.

3 Resultados

Os algoritmos propostos foram implementados utilizando C++ e os testes foram realizados em um computador com processador *Intel(R) Core™ i5-4210UCPU@1.70GHz* × 4 com 4GB de memória RAM e sistema operacional Ubuntu 19.10.

Para a realização dos testes foi utilizado o grafo da cidade de Viçosa - MG, variando o número de unidades disponíveis e com as ruas em contra-mão sendo consideradas corretas porém com uma distância 50% maior para serem percorridas.

É importante destacar que ao aumentar o número de unidades, o resultado sempre melhora, uma vez a probabilidade de haver uma unidade próxima à ocorrência de um crime aumenta. Neste contexto, o resultado apenas deixa de melhorar quando há pelo menos uma

² A importância de um vértice é dada pela soma das arestas incidentes a ele.

unidade em cada vértice. Apesar disso, geralmente, as cidades apresentam um número pequeno de unidades e, portanto, foram feitos testes de como seriam os resultados dos algoritmos para 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70 e 75 unidades. Tais resultados são apresentados na Tabela 7 e é possível observar a diferença dos algoritmos no Gráfico 8.

Número de Unidades	Greedy	Scatter Search	ILS
20	2999157	2509440	2196909
25	2759703	2380662	1924978
30	2451690	2451690	1766059
35	2373257	2227083	1760327
40	2284265	1852760	1573850
45	2083164	1860786	1478718
50	1984178	1817746	1653467
55	1893755	1662661	1535296
60	1691750	1615282	1429580
65	1623356	1574761	1363532
70	1558583	1558583	1330137
75	1536992	1478610	1283493

Figura 7: Tabela de resultados das heurísticas propostas



Figura 8: Gráfico de resultados das heurísticas propostas

4 Conclusões

Através dos dados apresentados na seção anterior, pode-se observar que o ILS apresentou melhores resultados que o Scatter Search em todos os testes. Uma maneira de melhorar este segundo algoritmo seria através da implementação de novas formas de combinação ou, também remover a limitação do número de subconjuntos gerados, mesmo que, com isso, o tempo de execução aumente consideravelmente. E uma outra possível melhoria, agora podendo ser aplicada às duas heurísticas, seria a implementação de um

path relinking, onde a cada passo um policial seria realocado para um local mais próximo, topologicamente, de uma solução elite.

Referências

GLOVER, F.; LAGUNA, M.; MARTI, R. Scatter search and path relinking: Advances and applications. In: *Handbook of metaheuristics*. [S.l.]: Springer, 2003. p. 1–35. Citado na página 3.

HOOS, H. H.; STÜTZLE, T. Local search algorithms for sat: An empirical evaluation. *Journal of Automated Reasoning*, Springer, v. 24, n. 4, p. 421–481, 2000. Citado na página 2.

MENDES, N. F. M.; SANTOS, A. d.; GONÇALVES, L. B. Métodos para o problema de posicionamento de unidades policiais. In: *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional, XLVI SBPO*. [S.l.: s.n.], 2014. p. 639–650. Citado na página 1.

TALBI, E.-G. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009. v. 74. Citado na página 3.