



**42°CSBC**  
CONGRESSO DA SOCIEDADE  
BRASILEIRA DE COMPUTAÇÃO

Instituto de

**computação**



**uff Science**

Laboratório Virtual de Pesquisa em e-Science



# Uma Heurística Híbrida para o Problema de Roteamento de Viaturas Policiais em Grandes Centros Urbanos

Raphael Leardini, Eduardo Canellas, Bruno Sá, Wagner Santos,  
Yuri Frota, **Daniel de Oliveira**, Isabel Rosseti

`danielcmo@ic.uff.br`

3 de agosto de 2022

1 Motivação

2 Heurística Proposta

3 Resultados

4 Conclusões

1 Motivação

2 Heurística Proposta

3 Resultados

4 Conclusões

- ▶ Crise econômica tem assolado o Brasil nos últimos anos
- ▶ Alta na taxa de desemprego
- ▶ Deslocamento de parte da população para grandes centros urbanos
- ▶ Aumento da população morando em áreas de risco (e.g., favelas e bairros mais perigosos)
- ▶ Aumento da violência nas grandes cidades

**G1** G1

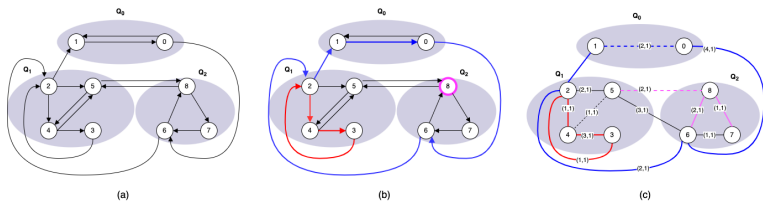
**Mesmo com quarentena, Brasil tem alta de 6% no número de assassinatos no 1º semestre**

Após dois anos seguidos de recordes na queda de mortes, país volta a registrar aumento de violência nos primeiros seis meses de 2020.

Aug 21, 2020



- ▶ A autoridade policial tem feito uso de ferramentas para criação de políticas públicas eficazes
- ▶ Policiamento ostensivo é uma delas!
  - ▶ Policiais e viaturas são posicionados em pontos estratégicos da cidade
  - ▶ A presença da força policial evita a ocorrência de certos tipos de crime
- ▶ **Problema!** Os recursos disponíveis (e.g., policiais e viaturas) não são suficientes para cobrir todas as áreas com maior ocorrência de crimes em uma cidade.
- ▶ Parte de um projeto acadêmico de colaboração entre a Polícia Militar do Estado do Rio de Janeiro e o IC/UFF
- ▶ Abordagem heurística para otimização do patrulhamento
- ▶ Baseado no histórico das ocorrências de crimes na cidade



**Figura:** (a) Grafo direcionado (b) Exemplo de solução (c) Arestas cobertas no grafo

Função Objetivo:

$$\max \sum_{\{i,j\} \in E} f_{ij} \cdot c_{ij} \quad (1)$$

- ▶ A função maximiza o fator de crime das arestas cobertas pelas unidades móveis e fixas.
- ▶ O fator de crime é calculado a partir da definição de pesos para cada tipo de crime

1 Motivação

2 Heurística Proposta

3 Resultados

4 Conclusões



*Iterated Local Search (ILS) e Variable Neighborhood Descent (VND)*

---

**Algorithm 1:**  $ILS_{Urb}(G, \maxIterILS)$

---

```
1  $s \leftarrow Construction(G)$ 
2  $s \leftarrow LocalSearchVND(G, s)$ 
3  $s^* \leftarrow s$ 
4  $iterILS \leftarrow 0$ 
5 while  $iterILS < \maxIterILS$  do
6    $s \leftarrow Perturb(G, s, iterILS)$ 
7    $s \leftarrow LocalSearchVND(G, s)$ 
8    $iterILS \leftarrow iterILS + 1$ 
9   if  $f(s^*) > f(s)$  then
10     $s^* \leftarrow s$ 
11     $iterILS \leftarrow 0$ 
12  end
13 end
14 return  $s^*$ 
```

---

- ▶ *Construction*: geração de solução viável
- ▶ *LocalSearchVND*: exploração da vizinhança melhorando a solução atual
- ▶ *Perturb*: função para escapar de ótimos locais

- ▶ Três conjuntos de rotas e um conjunto de vértices: *ROUTES* ( $INTER_{E_k}$ , *INTRA* e  $INTER_{\emptyset}$ ) e *STATIC*
- ▶ Quatro métodos de inserção distintos: *FindINTER<sub>E<sub>k</sub></sub>Route*, *FindINTRARoute*, *FindINTER<sub>∅</sub>Route* e *FindSTATICVertices*
- ▶ Gerar rotas para cada conjunto de acordo com as unidades de polícia disponíveis
- ▶ **Ordem de construção:** do mais restrito para o menos restrito

- ▶ **Limites:** distância máxima a ser coberta e um grupo de arestas prioritárias diferente de vazio
- ▶ **Construção:**  $FindINTER_{E_k} Route$
- ▶ **Ideia do algoritmo:** para cada aresta prioritária que ainda precisa ser coberta pela rota, tenta-se encontrar o menor caminho, do último vértice adicionado a rota até a aresta prioritária, usando *Dijkstra*.
- ▶ Se, pelo menos, uma aresta prioritária não estiver coberta, a solução é considerada inviável
- ▶ Um dos gargalos do algoritmo é o tempo de execução

- ▶ **Limites:** distância máxima intrazona, com todos os vértices sendo da mesma zona
- ▶ **Construção:** *FindINTRARoute*
- ▶ **Ideia do algoritmo:** guloso, insere arestas com maiores fatores de crime
- ▶ Uso de memória para caminhos descartados: se esta memória estiver cheia, ela e a atual rota em construção são descartadas, e um novo vértice inicial é escolhido

- ▶ **Limites:** distância máxima e grupo de arestas prioritárias igual a vazio
- ▶ **Construção:** *Find $INTER_{\emptyset}$ Route*
- ▶ **Ideia do algoritmo:** idêntico ao *Find $INTRARoute$* , sem a restrição de vértices na mesma zona

- ▶ **Construção:** *FindSTATICVertices*
- ▶ **Ideia do algoritmo:** guloso, insere unidades fixas em vértices com maiores fatores de crime ainda não cobertos

- ▶ Busca exaustiva em um conjunto pré-definido de movimentos
- ▶ Primeiro movimento que melhora a solução
- ▶ Três tipos de movimento:
  - ▶ SP-1 e SP-2 para  $INTER_{E_k}$ ,  $INTRA$  e  $INTER_{\emptyset}$
  - ▶ MOV<sub>v</sub> para  $STATIC$
- ▶ Aplicação dos movimentos: mais restrito para menos restrito
- ▶ Ordem de aplicação:
  1. SP-1 para  $INTER_{E_k}$ ,  $INTRA$  e  $INTER_{\emptyset}$
  2. MOV<sub>v</sub>
  3. SP-2 para  $INTER_{E_k}$ ,  $INTRA$  e  $INTER_{\emptyset}$



# Movimento SP-1

- ▶ Remoção de uma aresta aleatória, exceto aresta prioritária em  $INTER_{E_k}$
- ▶ Uso de *Dijkstra* ou busca em largura com fila de prioridade para reconstrução



Figura: Aresta selecionada em vermelho



Figura: Trecho que substitui a aresta selecionada, melhorando o fator de crime



Figura: Nova rota com a inserção do trecho

- ▶ Remoção de duas arestas aleatórias (essas duas arestas estão em sequência), exceto aresta(s) prioritária(s) em  $INTER_{E_k}$
- ▶ Uso de *Dijkstra* ou busca em largura com fila de prioridade para reconstrução



Figura: Arestas selecionadas em vermelho



Figura: Trecho que substitui as arestas selecionadas, melhorando o fator de crime



Figura: Nova rota com a inserção do trecho

- ▶ **Melhor** melhoria encontrada
- ▶ Pode alterar a posição de **todas** as unidades fixas.

- ▶ Impedir que o algoritmo de busca local retorne com facilidade ao ótimo local
- ▶ Movimentos parecidos com SP-1 e SP-2:
  - ▶ São definidos os números de rotas (de qualquer um dos conjuntos) que serão modificadas e o número de modificações que serão realizadas para cada rota. Ambos baseados na iteração atual do ILS
  - ▶ Rotas e arestas escolhidas aleatoriamente
  - ▶ Movimentos não são de melhoria, apenas de substituição

1 Motivação

2 Heurística Proposta

**3 Resultados**

4 Conclusões

- ▶ Geração de instâncias: dados extraídos da página da Secretaria de Segurança Pública do Estado de São Paulo
  - ▶ *OpenStreetMap*
  - ▶ *QuickOSM*
  - ▶ *QGIS*
  - ▶ Nomenclatura: "local-"número de vértices"
- ▶ Ambiente computacional: Notebook Lenovo ideapad 320-15IKB Windows 10 Pro 64 bits CPU Intel(R) Core(TM) i5-7200U CPU 2.50 GHz e C++ (8.1.0)
- ▶ 10 execuções com 10 sementes distintas



NOTA EXPLICATIVA

TAXA DE HOMICÍDIO

ESTATÍSTICA

HOMICÍDIO DOLOSO

FEMINICÍDIO

LATROCÍNIO

LESÃO CORPORAL SEGUIDA DE MORTE

REGISTRO DE ÓBITOS - IML

MORTE DECORRENTE DE INTERVENÇÃO POLICIAL

MORTE SUSPEITA

FURTO DE VEÍCULO

ROUBO DE VEÍCULO

FURTO DE CELULAR

ROUBO DE CELULAR

Secretaria de Estado da Segurança Pública  
Rua Líbero Badaró, 39, Centro, SP

CEP: 01009-000  
Telefone: (11) 3291-6500

Tabela: Resultado do ajuste de parâmetros

Instâncias	Parâmetro	Intervalo	<i>elite config</i>
cjn2-41 parque-peruche-710 jardim-centenario-1117 santo-amaro-2627 parque-santa-madalena-3578	<i>maxIterILS</i>	[100, 400]	330



Instância	Ótimo	Melhor $_{ILS_{Urb}}$	Média $_{ILS_{Urb}}$	$T_{Modelo}(s)$	$T_{ILS_{Urb}}(s)$
cjn1-18	126.74	123.85	123.85	0.037	0.042
cjn1-24	169.47	<b>169.47</b>	165.70	0.081	0.059
cjn1-33	270.96	269.96	264.72	0.190	0.271
cjn1-40	335.52	278.43	275.94	0.243	0.124
cjn1-46	398.25	385.20	382.23	0.740	0.296
cjn2-18	120.77	<b>120.77</b>	<b>120.77</b>	0.008	0.033
cjn2-25	211.84	<b>211.84</b>	208.64	0.146	0.085
cjn2-35	305.36	<b>305.36</b>	304.16	0.057	0.155
cjn2-50	408.41	<b>408.41</b>	399.41	0.284	0.219
mini1-64	390.97	357.23	357.23	13.390	0.260
mini2-75	399.94	<b>399.94</b>	388.97	5.680	0.193
mini3-20	108.48	<b>108.48</b>	100.71	0.029	0.048

- ▶ Experimento considerou instâncias de tamanho menor (pequenas áreas dentro de bairros, e.g., quatro quarteirões em torno do MASP)
- ▶ Heurística encontra bons resultados, encontrando 7 das 12 soluções ótimas geradas
- ▶ Para a maioria das instâncias o tempo da heurística se manteve próximo ao do modelo
- ▶ Modelo foi capaz de encontrar soluções ótimas em instâncias com até 75 vértices



Figura: Instância cjn2-50: (a) Modelo e (b) Heurística  $ILS_{Urb}$

Tabela: Resultados para instâncias de bairros

Instância	Melhor $ILS_{Urb}$	Média $ILS_{Urb}$	$T_{ILS_{Urb}}$ (s)
bosque-da-saude-405	645.40	631.57	0.087
favela-dos-anjos-922	2010.82	1922.07	2.703
fazenda-itaim-864	886.14	886.14	0.361
jardim-japao-879	2529.98	2485.23	2.215
jardim-primavera-682	1106.28	1074.27	0.424
mirandopolis-509	598.16	586.74	0.142
planalto-paulista-567	994.23	962.60	0.612
vila-guarani-625	1189.16	1159.60	0.324
vila-renato-645	663.66	662.65	0.263
alto-de-santana-1152	1253.94	1232.54	1.002
belenzinho-1548	2201.91	2067.07	4.209
bom-retiro-1542	1498.33	1388.70	4.668
bras-1674	2241.53	2078.45	9.189
cidade-ipava-1608	2937.02	2764.01	6.495
jardim-aricanduva-1430	3440.90	3246.16	3.065
jardim-maristela-1372	3668.07	3549.50	7.558
pompeia-1617	2865.16	2699.09	9.611
rio-pequeno-1779	3421.86	3301.80	9.403
vila-amelia-1112	1740.10	1606.32	3.334
vila-antonio-1912	5538.27	5249.20	12.244
vila-leonor-1130	1861.08	1756.68	1.230
vila-leopoldina-1258	2743.53	2596.59	2.499
vila-progresso-1988	4961.00	4841.09	23.221
americanopolis-2897	8336.42	8119.13	38.019

Tabela: Resultados para instâncias de bairros (cont.)

Instância	Melhor $ILS_{Urb}$	Média $ILS_{Urb}$	$T_{ILS_{Urb}}$ (s)
baixa-grande-2223	5223.87	5075.37	15.109
campo-belo-2515	8114.80	7774.61	38.897
campo-limpo-2488	7352.29	6996.18	26.582
interlagos-2318	4220.88	4070.60	23.487
jardim-botucatu-2244	5002.92	4809.81	20.668
jardim-campinas-2485	5320.40	5087.91	24.564
jardim-paulista-2408	1747.67	1663.55	9.142
jardim-tupi-2745	3927.32	3747.76	17.033
mooca-2572	7137.80	6599.82	36.846
morumbi-2942	4627.24	4279.62	29.515
parque-cisper-2006	2537.41	2438.27	5.601
sossego-2463	4238.18	4060.44	25.312
sumare-2250	3377.24	3141.53	17.192
cidade-tiradentes-3562	6456.89	6263.75	34.814
itaim-paulista-3209	9545.04	8833.24	64.788
jardim-tres-marias-3282	12805.70	12079.60	61.397
limoeiro-3472	9639.66	9176.26	60.360
sacoma-3745	12745.90	12264.10	103.881
vila-formosa-3460	7029.78	6581.77	60.500
vila-matilde-3454	9708.57	9318.87	75.710
vila-santa-teresa-3222	11127.80	10558.70	74.798
vila-solange-3372	11319.30	10952.60	54.207
aeroporto1-1301	6193.76	5958.76	30.183
perdizes1-1793	5712.98	5554.95	33.755

- ▶ Instâncias de maior escala que não possuem resultado ótimo conhecido, *i.e.*, instâncias em que o modelo não é capaz de gerar resultados viáveis
- ▶ As instâncias utilizadas nesse experimento englobam bairros inteiros, *e.g.*, Brás, Campo Belo, Morumbi, Itaim Paulista, *etc.*
- ▶ Tempo de execução cresce devido ao aumento do tamanho das instâncias (*i.e.*, número de vértices)

1 Motivação

2 Heurística Proposta

3 Resultados

**4 Conclusões**

- ▶  $ILS_{Urb}$  alcançou resultados ótimos em diversas instâncias
- ▶  $ILS_{Urb}$  encontrou soluções para instância que o modelo não conseguiu em um tempo de execução aceitável
- ▶ Os resultados obtidos se encontram em análise por parte dos especialistas em segurança pública
- ▶ Aplicação com dados do Estado do Rio de Janeiro



- ▶ Os autores gostariam de agradecer ao CNPq e a FAPERJ pelo apoio financeiro
- ▶ Os autores gostariam de agradecer aos oficiais da PMERJ que apoiam o desenvolvimento desse trabalho
- ▶ **OBRIGADO!**