Statim: Um Otimizador de Rotas

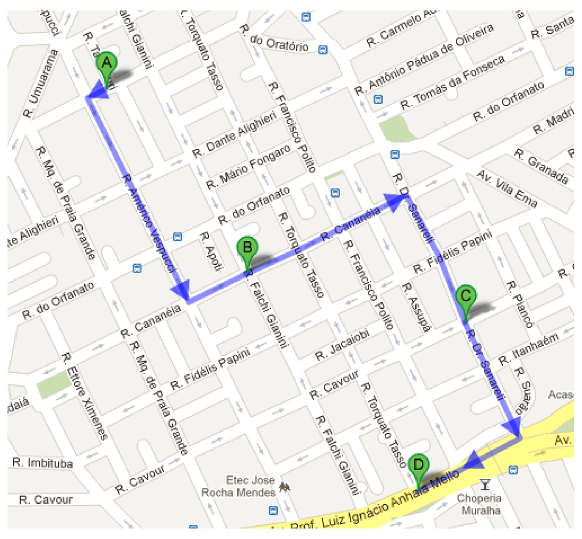
Felipe de Godoi Torres, Geraldo Castro Zampoli

Instituto de Matemática e Estatística - IME/USP

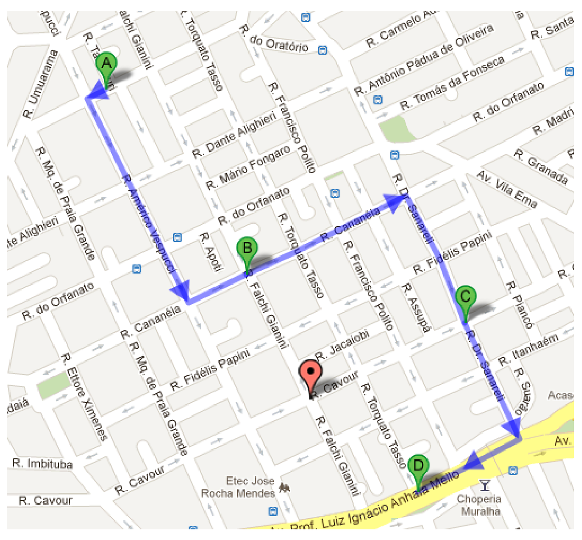
felipe.godoi.torres@usp.br, geraldo.zampoli@usp.br

O problema

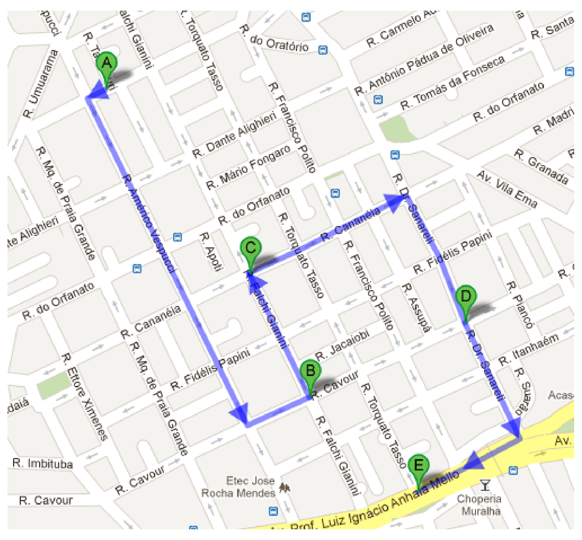
Suponha um conjunto de entregadores, cada um com o seu respectivo itinerário. É bastante comum que, ao decorrer do dia de trabalho, apareçam novos pontos de destino (entregas ou retiradas de documentos, por exemplo) que não foram previstos nos itinerários iniciais. Como atribuir esse novo destino a um entregador, para que haja um menor desvio da rota original?



Trajeto do entregador até o momento



Aparece um novo ponto de entrega/coleta



O caminho mínimo com esse ponto é encontrado e a rota é refeita

Modelagem

Considere o conjunto $C = \{I_1, I_2, ..., I_m\}$ onde $I_i = (P_{i1}, P_{i2}, ..., P_{iN})$ é uma n-úpla ordenada de pontos (x, y) em $\mathbb{R}^2$.

Definamos:

Um ponto P é da forma (x, y) com $x, y \in \mathbb{R}$.

(a) $d_{a,b}$ = distância, não-euclideana, entre os pontos a e b.

Seja P^* um novo ponto a ser inserido em alguma posição k de um itinerário I de tamanho n. Então, usando **(a)**, temos:

(b) $d_{I_i} = \sum_{j=1}^{j<n} d_{P_{ij},P_{ij+1}}$: distância total do itinerário I_i.

(c) $d_{I_i}[P^*]_k = d_{I_i} + [d_{p_{ik-1},P^*_k} + d_{P^*_k,p_{ik+1}}] - d_{p_{ik-1},p_{ik+1}}$: distância total de I_i com P^* na posição k.

Queremos que essa posição k seja tal que, fixando-se o itinerário, ao inserir o novo ponto P^*, nos dê a menor distância total. Sendo assim, usando **(c)**, temos:

(i) $d_{I_i}[P^*]_k \leq d_{I_i}[P^*]_h \forall k \neq h$, I_i fixo

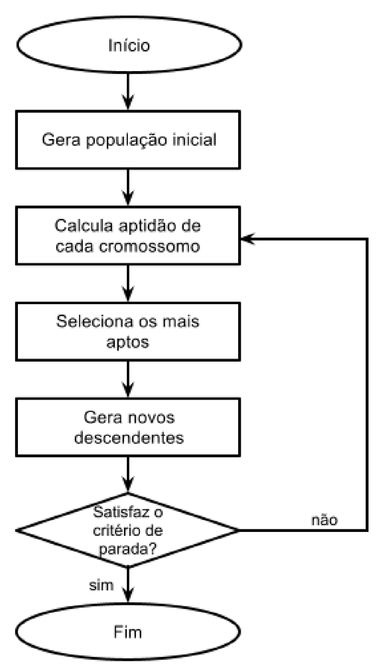
Após obtermos todos os itinerários satisfazendo **(i)**, queremos aquele que tiver a menor distância total, ou seja:

(ii) $d_{I_i}[P^*]_k \leq d_{I_j}[P^*]_h \forall j \neq i$; h, k fixos

Dessa forma, teremos o itinerário procurado.

Algoritmos Genéticos

O funcionamento básico de um algoritmo genético segue o fluxograma abaixo:



Fluxograma de um algoritmo genético

Operadores genéticos

Chamamos de operadores genéticos os algoritmos para seleção dos mais aptos e geração dos descendentes de cada geração. Neste trabalho, estudamos os operadores *Partially Mapped Crossover (PMX)*[1][2] e *Cycle Crossover (CX)*[3], para a geração de descendentes, e *Ranking* e *Wheel*, para a seleção dos mais aptos.

Critérios de parada

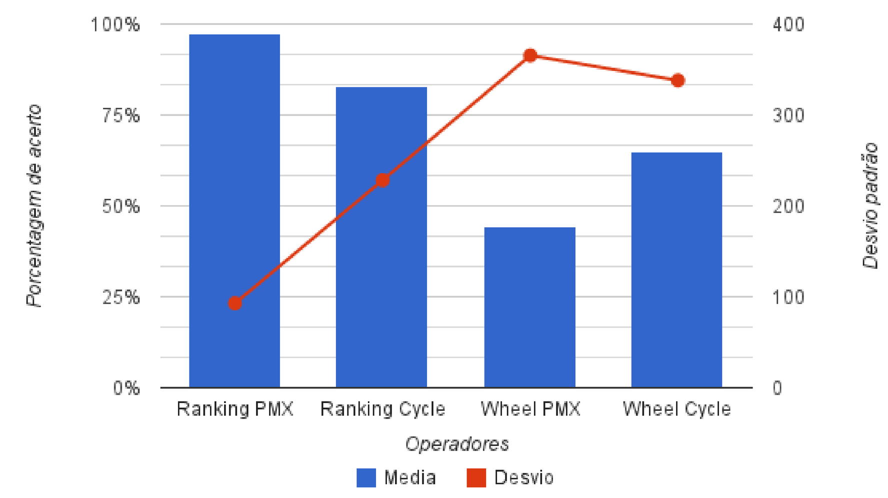
A cada iteração de um algoritmo genético, uma verficação é feita para decidir se há uma nova iteração ou o seu término. Para este problema, implementamos dois algoritmos de parada:

Generation: define-se o número de gerações que serão criadas. Ao término da última geração, o cromossomo mais apto é devolvido.

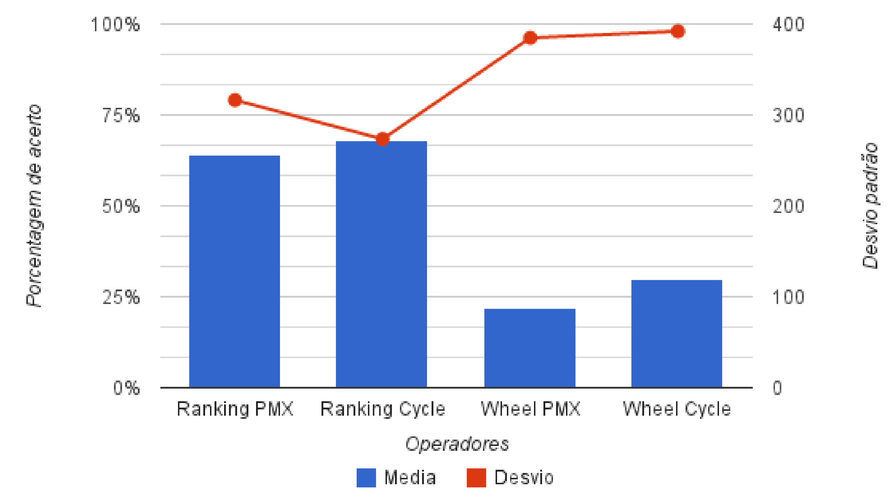
Popularity: um valor de popularidade, p, é definido. O algoritmo para quando a ocorrência de um cromossomo na população for maior ou igual a p.

Resultados

Usando o *Generation*, cada configuração foi executada 5000 vezes alterando o tamanho da população inicial, i, número de gerações, n e quantidade de cromossomos selecionados para reprodução, f. Os resultados obtidos foram:



5 pontos, $i = 100$, $n = 50$, $f = 50$



10 pontos, $i = 1000$, $n = 500$, $f = 500$

Note que, o *Ranking* obteve um melhor resultado em ambos os casos de teste. No entanto, para instâncias maiores, o *Ranking–PMX* indica um pior desempenho, motivado pela rápida convergência dessa combinação em comparação com o *Ranking–Cycle*.

Já o *Wheel*, implementado sem nenhuma memorização, favorece a perda do melhor resultado encontrado, explicando um pior desempenho.

Referências

[1] T. Starkweather and S. Mcdaniel and D. Whitley and K. Mathias and C. Whitley. *A Comparison of Genetic Sequencing Operators*. Proc of the 4th Int. Conf. Genetic Algorithms. p. 69–76. 1991.

[2] ÜÇOLUK, Göktürk. *Genetic Algorithm Solution of the TSP Avoiding Special Crossover and Mutation*. Intelligent Automation and Soft Computing, 3(8), TSI Press. p. 265–272. 2002.

[3] I. Oliver, D. Smith, and J. R. Holland. *A study of permutation crossover operators on the traveling salesman problem.* Genetic Algorithms and their Applications: Proc. of the 2nd Int. Conf, John Grefenstette. p. 224–230. 1987.