# Algoritmo ótimo de procura dos taxistas

## Objetivos

Dado um ponto geográfico (latitude/longitude) encontrar N (N>=1) taxistas disponíveis para atender uma corrida.

## Descrição

1. A partir do ponto inicial do requisitante definir um retângulo geográfico para busca de taxistas disponíveis

2. Para cada taxista disponível, calcular um score de quão bem esse taxista atende à requisição levando em conta regras de negócio para cálculo do score, como: distância, tempo de chegada estimado (fatores externos como trânsito, protestos, etc. podem influenciar), preferências do requisitante (ex: um requisitante pode optar por esperar mais por um taxista que tenha um histórico de avaliação maior, outro requisitante pode preferir taxistas que tenham ar-condicionado, etc.)

3. Dependendo da complexidade das regras de negócio, o cálculo de score pode não ser tão trivial. Idealmente o cálculo do score deve ser feito em processos paralelos para uma sincronização final ao término do processamento

4. cutoff (min score): ignorar automaticamente taxistas que não atinjam um score mínimo, ou seja, que sabidamente não atendem às necessidades do requisitante

5. Para cada taxista que passar pelo cutoff, adicioná-lo em uma lista de resultados juntamente com seu score

6. Caso a lista de taxistas seja maior ou igual a N, ordená-la por ordem decrescente de score e retorná-la como resultado. O processamento está finalizado.

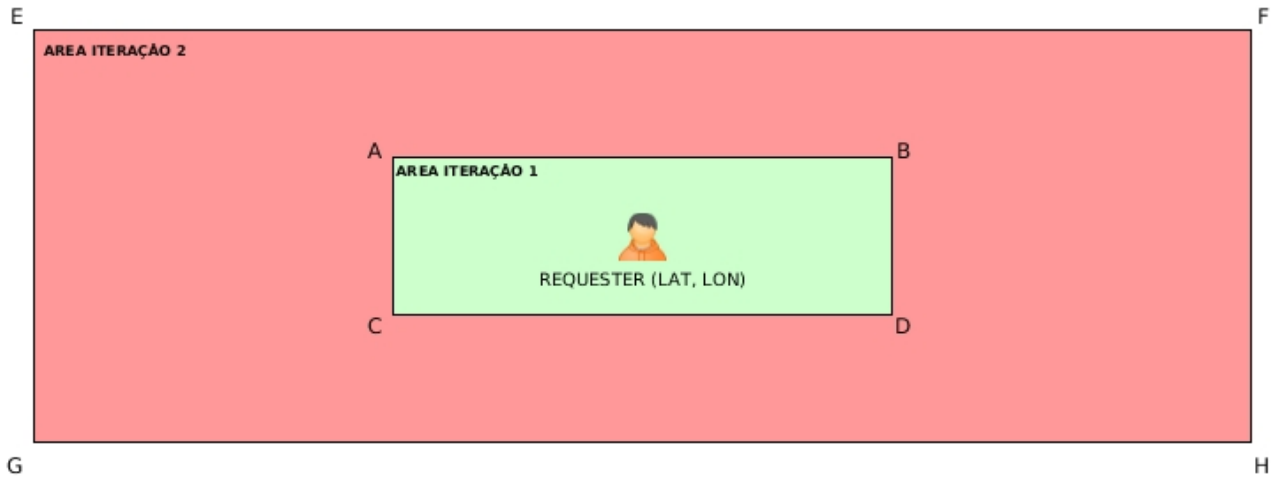
7. Caso a lista de taxistas seja menor que N, aumentar exponencialmente a área a ser pesquisada buscando taxistas contidos na nova área mas fora da área anterior (não precisamos reprocessar uma área que já foi processada). Retomar processamento a partir do item 2.

8. Repetir o item 7 quantas vezes forem necessárias, mas deve haver uma condição de parada (ex: limite de execuções, limite de tempo, etc.). Ao atingir o limite, a lista de taxistas deve ser ordenada por ordem decrescente de score e retornada (mesmo que não tenham sido encontrados N taxistas) e o processo é finalizado.

## Exemplo

Queremos encontrar 5 taxistas para atender o “requester” posicionado em (LAT/LON). Definimos então um retângulo geográfico inicial para listar os taxistas ativos (áreas em vermelho representam onde a busca deve ser feita):


Na primeira iteração conseguimos, por exemplo, apenas um taxista. A lista seria algo como [(taxi1,score1)]. Como 1<5, extrapolamos a área exponencialmente e fazemos uma nova busca:




A busca deve ocorrer somente na área em vermelho, já que a área do triângulo ABCD já foi processada na iteração anterior.

A nova busca retornou, por exemplo, 8 taxistas. A lista conterá portando 9 itens (1 da primeira iteração+8 da segunda iteração), ex: [(taxi1,score1).....(taxi9,score9)].

Como 9>5, a lista é ordenada por ordem decrescente de score e retornada. O processamento acaba. Exemplo de lista final: [(taxi3,score3),(taxi8,score8).....]

**obs.: um esboço de implementação pode ser encontrado em [ROOT]/taxi_api/helpers/driver_finder.py**