# Estratégia Mista (Branch and Bound seguido de TSP)

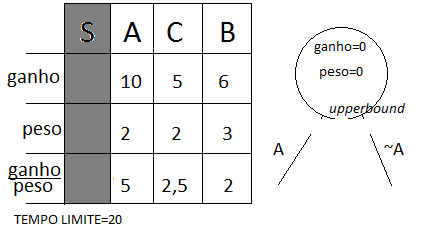
Esta abordagem procura um trajecto que cumpra as condições de limite de tempo e que tenha potencial para satisfazer a condição de maximização dos valores (preferências). Não apresentando uma solução óptima o algoritmo faz uso de heurística para conseguir obter uma solução aceitável em menos tempos que algoritmos mais precisos.

O algoritmo funcionaria em 3 etapas distintas.

Em primeiro lugar devia encontrar a aresta de maior peso do grafo.

Note-se para qualquer conjunto de solução *I* possível temos que:

Seguidamente é usado um algoritmo *de branch and bound*. Para este algoritmo teríamos:

-O limite de tempo como o peso máximo;

-As cidades (vértices) como itens;

-A preferência dada pela Rita como o ganho;

-O peso seria dado pela soma do tempo de visita da cidade com o tempo encontrado na etapa 1. Assim sendo garante-se que o resultado obtido é uma combinação válida (que pode ser visitada dentro do tempo limite).

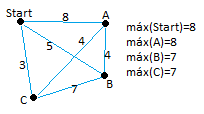
Uma vez já encontrada uma combinação de cidades que sabemos ser possível de visitar em qualquer ordem dentro do tempo dado podemos abordar o problema como um TSP para arranjar o melhor trajecto possível (que demora menos tempo) entre as cidades seleccionadas.

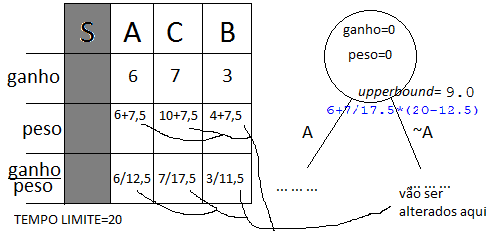
# (… not solid yet i think =/) Estratégia Mista (Branch and Bound Hibrido seguido de TSP)

Dados de entrada iguais ao anterior.

Sugestão de Algoritmo:

Em 3 etapas

1)Primeiramente seriam somados os tempos máximos de viagem para cada vértice. Denominemos esta soma de *maxTT (max travel time)*. Olhando para o grafo exemplo apresentado ao lado, o *maxTT* daria 30. Este valor representa o valor máximo que o tempo de viagem pode durar caso a Rita passe em todos as cidades (vértices).

2)Seguidamente é usado um algoritmo *de branch and bound híbrido*. Para este algoritmo teríamos:

-O limite de tempo como a capacidade máxima;

-As cidades (vértices) como itens;

-A preferência dada pela Rita como o ganho;

-O peso iria variar consoante o ponto do *bnb*. Seria dado por:

, onde *tv* é o tempo que demora visitar a cidade, *n* é o número de cidades que ainda podem ser escolhidas e *maxTTlocal* é o resultado do somatório realizado no passo anterior apenas para as cidades contempladas no branch que se verifica (pode ser calculado rapidamente fazendo ).

Tendo em conta que , e em caso de ser de arredondamentos nos cálculos que garante-se que o algoritmo de *bnb* encontra uma combinação de cidades possível de ser visitada.

3) usar TSP como na Estratégia Mista apresentada anteriormente.