

COM10393 – MÉTODOS DE OTIMIZAÇÃO

TRABALHO I

O objetivo geral deste trabalho consiste em iniciar a modelagem e implementação de meta-heurísticas para resolução do **Problema de Alocação de Berços (PAB)**.

O PAB consiste em atribuir os navios que chegam a um determinado porto para as “posições” de atracação disponíveis ao longo de um cais (berços). As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar. A Figura 1 ilustra um cenário típico do PAB.

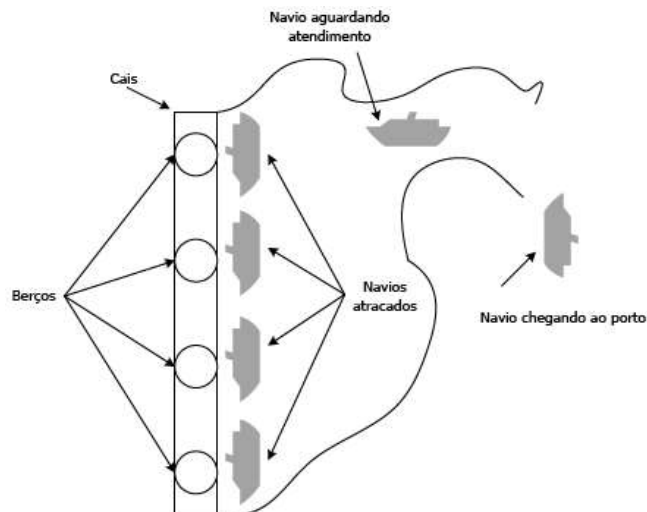


Figura 1 – Ilustração do PAB.

O objetivo do PAB é minimizar os custos referentes ao porto e ao navio, que é relacionado, entre outros, ao tempo de serviço dos navios. Logo, busca-se minimizar o tempo de serviço de todos os navios, ou seja, o tempo desde a chegada do navio até a conclusão de seu atendimento. A Figura 2 apresenta os principais dados a serem considerados para resolução do PAB, considerando um navio i e um berço k .

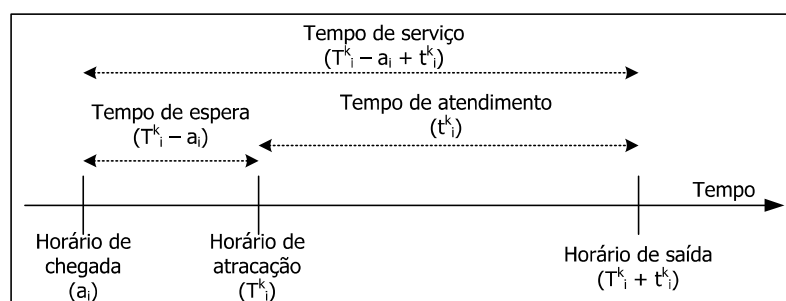


Figura 2 – Principais dados do PAB.

O PAB considera a existência de um conjunto N com os navios programados para serem atendidos por um conjunto K de berços. Devido ao tipo de equipamento disponível em cada berço, ao tamanho, tipo e quantidade de carga de cada navio, etc., o tempo para atendimento de cada navio em cada berço pode ser diferente. Logo, o tempo de atendimento do navio i ($i \in N$) no berço k ($k \in K$) é dado como t_i^k , e é conhecido a priori. O tempo previsto de chegada de cada navio i , a_i ($i \in N$), também é conhecido a priori. Logo, deve-se determinar qual berço k ($k \in K$) cada navio i ($i \in N$) irá atracar, e o respectivo tempo de atracação T_i^k . Caso um navio chegue ao porto e não exista um berço disponível, o mesmo deverá ficar aguardando até a sua atracação (tempo de espera). Cada navio i possui um tempo limite b_i para sua partida do porto, ou seja, o serviço deve obrigatoriamente ser concluído até esse tempo. E ainda, cada berço k ($k \in K$) possui tempos de abertura o^k e fechamento c^k , ou seja, os navios devem ser atendidos dentro desses tempos.

O objetivo é minimizar o tempo de serviço de cada navio. Então, uma solução s para o PAB possui a seguinte função objetivo:

$$\text{Minimizar } f(s) = \sum_{i \in N} (T_i^k - a_i + t_i^k) \quad \text{se } i \text{ atracou em } k$$

Para que uma solução seja considerada viável, deve-se considerar as seguintes restrições:

- Todos os navios devem ser atendidos;
- Cada navio só pode ser atendido por um único berço;
- Um berço não poder atender mais de um navio ao mesmo tempo;
- Cada navio só pode atracar após a sua chegada no porto ($T_i^k \geq a_i$);
- O atendimento de cada navio deve terminar antes do seu tempo limite ($T_i^k + t_i^k \leq b_i$);
- Todos os navios devem ser atendidos no período de funcionamento dos berços, ou seja, $o^k \leq T_i^k \leq c^k$ e $o^k \leq T_i^k + t_i^k \leq c^k \forall k \in K$ e i atracado em k .

As restrições podem ser tratadas na modelagem e/ou por meio de penalização. Nesse caso, os valores dos pesos devem ser definidos e calibrados, e apresentados junto com o trabalho na data prevista para apresentação do mesmo.

Como entrada de dados, deverão ser consideradas instâncias com até 100 navios ($|N| \leq 100$) e 20 berços ($|K| \leq 20$), no formato descrito a seguir. Nessas instâncias, o valor zero para o tempo de atendimento de um navio em um berço ($t^k_i = 0$) indica que o berço k não possui condições (equipamento, tamanho, profundidade, etc.) para atender o navio i , ou seja, o navio i não poderá atracar no berço k , caso contrário, a solução será inviável.

Número de navios ($ N $)	Número de berços ($ K $)
Tempo de atendimento de cada navio (coluna) no berço 1 ($t^1_i \forall i \in N$)	
Tempo de atendimento de cada navio (coluna) no berço 2 ($t^2_i \forall i \in N$)	
...	
Tempo de atendimento de cada navio (coluna) no berço $ K $ ($t^{ K }_i \forall i \in N$)	
Abertura do berço 1 (o^1)	Fechamento do berço 1 (c^1)
Abertura do berço 2 (o^2)	Fechamento do berço 2 (c^2)
...	
Abertura do berço $ K $ ($o^{ K }$)	Fechamento do berço $ K $ ($c^{ K }$)
Tempo de chegada de cada navio ($a_i \forall i \in N$)	
Tempo limite para saída de cada navio ($b_i \forall i \in N$)	

Para auxílio na realização deste trabalho, são disponibilizadas 5 instâncias (anexadas a este trabalho): i01.txt, i02.txt, i03.txt, i04.txt e i05.txt. Essas instâncias deverão ser utilizadas nos experimentos a serem realizados neste trabalho e no Trabalho II.

A descrição do PAB e as demais informações apresentadas até então deverão ser utilizadas para realização tanto deste trabalho (Trabalho I) quanto do Trabalho II. Neste trabalho, deverá ser iniciada a modelagem para resolução do PAB por meio de meta-heurísticas. Para isso, deverá ser implementado um código, preferencialmente na linguagem C/C++, que contemple as seguintes características:

- Parâmetro: instância a ser considerada.
- Procedimento ou função para leitura da instância.
- Estruturas de dados necessárias para armazenar os dados de entrada.
- Estruturas de dados para armazenar uma solução. O código deve ser capaz de armazenar várias soluções diferentes.
- Procedimento ou função para criação de uma solução inicial para o problema (heurística construtiva). *
- Procedimento ou função para clonar uma solução qualquer.

- Procedimento ou função para calcular a função objetivo de uma solução qualquer. **
- Procedimento ou função para “escrever” (em tela e arquivo) uma solução qualquer com o máximo de informações possível: número de berços utilizados, número de navios atendidos (total e por berço), tempo de serviço (total e por berço), sequência de navios atendidos por cada berço com o tempo de início e fim do atendimento de cada navio, violações na abertura e fechamento dos berços (total e por berço), violações no tempo limite dos navios (total e por berço), e o valor da função objetivo da solução.

O método marcado com * deverá ser executado 1 vez para a instância i05 (sem semente aleatória), e os resultados obtidos deverão ser colocados na terceira e na quarta coluna da Tabela 1. Os métodos marcados com * e ** deverão ser executados separadamente, também para a instância i05, por 10000 vezes, e os resultados deverão ser colocadas nas duas últimas colunas da Tabela 1. A Tabela 1 está preenchida com um exemplo. Ela deverá ser preenchida e exibida na apresentação do trabalho.

Tabela 1 – Exemplo de tabela de resultados.

Descrição do Computador	α (seg.)	Solução Inicial (1 vez)		10000 vezes	
		FO	Tempo (seg.)	Tempo Sol. Inicial (seg.)	Tempo Calc. FO (seg.)
i7 3.5GHz – 8GB ram – Windows 7	169	29200,00	0,82300	10,35	2,13

O valor de α na Tabela 1 deverá ser obtido por meio da execução do programa *benchmark_machine* (anexo a este trabalho), que mede a “velocidade” do computador.

O trabalho deverá ser realizado **INDIVIDUALMENTE** e apresentado (código-fonte e Tabela 1 completa) em, **no máximo**, 10 minutos na aula do dia **27/10/2020**. Todos os arquivos gerados deverão ser enviados (via *Classroom*) em .zip (não enviar .rar e nem .exe) até o dia **26/10/2020** (até as 23:59hs).

O valor do trabalho será **5,0 pontos** (50% da nota final). Trabalhos copiados total ou parcialmente terão nota **ZERO**.

O trabalho que apresentar melhor relação entre a FO da solução inicial e o tempo para construí-la ganhará um bônus de 0,75 pontos (15%), o segundo 0,5 (10%) e o terceiro 0,25 (5%).

O trabalho que apresentar o menor tempo para o cálculo da FO (se estiver correto) por 10000 vezes ganhará um bônus de 0,75 pontos (15%), o segundo 0,5 (10%) e o terceiro 0,25 (5%).

O valor de α será considerado para equiparação dos tempos obtidos em computadores diferentes e, conseqüentemente, definir os vencedores do bônus.

Dúvidas relativas ao trabalho podem ser sanadas via *Classroom* ou via e-mail. Além disso, em todas as aulas será disponibilizado um tempo para discussão sobre o trabalho, então o mesmo deverá ser iniciado o quanto antes.