

Trabalho de Implementação

Valor: 30 pontos

Data: 24/11/2025 pelo Canvas. Não serão aceitos trabalhos com atraso.

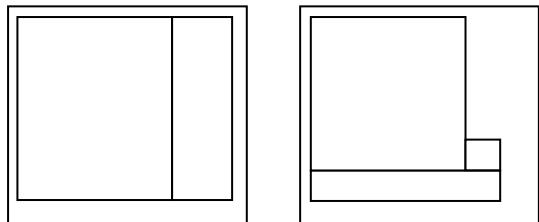
Trabalho em grupos de até 4 alunos

Descrição

Problemas de otimização são bastante comuns em vários setores produtivos. Um exemplo de aplicação é o planejamento de recorte e transporte de peças de metal para a produção de eletrodomésticos. Uma vez que o custo total do processo depende do grau de utilização da matéria prima, um problema relevante consiste em definir a sequência de peças a serem recortadas, de modo a evitar ao máximo o desperdício de metal e energia elétrica.

1^a Parte

Nesta parte do trabalho você deverá implementar uma solução para o problema de corte de peças que minimize o custo da operação, considerando a sobra de material e gasto de energia. As peças são cortadas de placas com dimensão fixa de 300 cm x 300 cm que custam R\$ 1000,00 cada. Vamos considerar que as peças têm formato retangular com lados medidos em múltiplos de 1 cm e que não podem ser rotacionadas por motivos estéticos (o metal é escovado). As peças são cortadas por laser e o custo é de R\$ 0,01 por cm. Os cortes têm que ser feitos deixando uma margem mínima de 10 cm nas placas, pois elas são presas na máquina de corte. Assim, o custo total da operação é o número de placas usadas para produzir todas as peças, mais o custo do corte a laser. Deseja-se obter o posicionamento de cada peça nas placas, de modo a minimizar o custo total de produção. A figura abaixo mostra um exemplo de posicionamento de 5 peças de dimensões (hxl) 250x200, 50x50, 200x200, 250x70, 50x250 e que usam 2 placas para serem produzidas, gastando R\$2000,00 de matéria prima e $((270*2 + 200 + 50 + 250*2) + (250*4 + 200 + 100)) * R\$0,01 = R\$25,90$ de energia.



1. Programe uma interface gráfica para o projeto para exibir o posicionamento das peças como na figura anterior. A descrição das peças a serem cortadas deve ser lida de um arquivo texto, onde a primeira linha contém o número de peças e as linhas restantes contêm a altura e largura em cm de cada peça separadas por espaço em branco. Exemplo:

```
5
250 200
50 50
```

200 200
250 70
50 250

2. Escreva uma função que, dadas as dimensões de uma peça e o estado de uma placa, indique se a peça cabe naquela posição e qual o custo de cortá-la.
3. Escreva uma solução baseada em força-bruta para o problema. Use o algoritmo de permutações para gerar todas as sequências de peças e calcule o custo. Observe que a posição da peça na placa influencia o resultado. Guarde o posicionamento de peças que produz o menor custo e determine o tempo de processamento, para diversos exemplos. O programa deve mostrar de forma gráfica como fica a disposição das peças.
4. Escreva uma solução por branch-and-bound, a partir da solução por força-bruta, que elimine ramos da árvore de permutações cujo lower bound indicar custo maior que o já obtido até aquele momento. Verifique se o resultado é o mesmo que o obtido por força-bruta e compare os tempos de processamento das duas soluções.
5. Proponha uma heurística para o problema. Compare os resultados com os obtidos pela solução ótima.

2^a Parte

Nesta parte do trabalho você deverá implementar uma solução para o transporte das peças em um navio que contém 2 porões. Por questões de segurança, as peças devem ser divididas em 2 grupos com pesos iguais ou o mais próximo possível.

6. Escreva uma solução baseada em força-bruta para o problema. Use o algoritmo de combinações para gerar todos os conjuntos possíveis e encontre a solução que minimiza a diferença das somas dos pesos dos 2 conjuntos. O programa deve mostrar como fica a divisão das peças e peso dos conjuntos.
7. Escreva uma solução por branch-and-bound, a partir da solução por força-bruta, que elimine ramos da árvore de soluções cujo lower bound indicar diferença maior que o já obtido até aquele momento. Verifique se o resultado é o mesmo que o obtido por força-bruta e compare os tempos de processamento das duas soluções.
8. Proponha uma heurística para o problema. Compare os resultados com os obtidos pela solução ótima.

Documentação

Documente as soluções e os testes, na forma de um relatório técnico em formato PDF com no máximo 20 páginas, segundo o padrão da SBC, contendo as seguintes seções:

- a. Introdução: Descrever o problema e o objetivo do trabalho.
- b. Solução proposta: Descrever os algoritmos usados para a solução do problema através de pseudocódigo.
- c. Implementação: Descrever detalhes dos programas implementados, principalmente aqueles utilizados para melhorar a eficiência da solução.

- d. Relatório de testes: Descrever os testes realizados e seus resultados, mostrando como ficou o mapa em alguns exemplos para variados número de linhas. Registrar o tempo de execução de cada um deles.
- e. Conclusão: Discutir os resultados obtidos, comparando as soluções por força-bruta, branch-and-bound e heurística, quanto à sua ordem de complexidade de tempo e memória, e quanto ao tempo de execução medido.
- f. Bibliografia segundo o padrão ABNT.

Considerações gerais e critérios de avaliação

1. O trabalho deverá ser feito em grupos de até quatro alunos, sem qualquer participação de outros grupos e/ou ajuda de terceiros. Também não é permitido uso de geradores de código ou documentação por IA. Cada aluno deve participar ativamente em todas as etapas do trabalho pois a avaliação é individual. Os componentes dos grupos devem ser informados até o dia 14/09/2025 através da planilha em
https://docs.google.com/spreadsheets/d/1K28PFzuxBUDvSwK0OkYL8Okc1GcD500dX-PlfK_JCbY/edit?usp=sharing
e não poderão ser alterados. Os alunos que não tiverem feito grupos até esta data serão considerados grupos individuais.
2. A codificação do trabalho deve ser feita em linguagem Java, C++ ou Python em um arquivo único. A apresentação poderá ser feita em um notebook levado pelo grupo ou em um computador do laboratório. Neste caso, não poderão ser utilizados bibliotecas gráficas ou qualquer recurso que não esteja instalado oficialmente nos laboratórios do ICEI. O grupo que desejar apresentar no laboratório deve informar o professor com uma semana de antecedência.
3. Os trabalhos (código e relatório) devem ser postados na forma de um arquivo compactado no padrão ZIP **por apenas um dos componentes**, com **tamanho máximo de 5 MB**, e seu nome deve ser o número do grupo fornecido pelo professor (Gnn) (Ex: G01.zip). **O arquivo-fonte e o relatório devem estar no diretório raiz** e devem conter o nome de todos os componentes do grupo no início do código. A apresentação será feita a partir do código postado no Canvas. **Trabalhos com mais de 5 MB não serão corrigidos**.
4. **Trabalhos iguais, na sua totalidade ou em partes, gerados por IA, copiados, “encomendados” ou outras barbaridades do gênero, serão severamente penalizados. É responsabilidade do aluno manter o sigilo sobre seu trabalho, evitando que outros alunos tenham acesso a ele. No caso de cópia, ambos os trabalhos serão penalizados, independentemente de quem lesou ou foi lesado no processo.**
5. Será pedida ao Colegiado uma advertência formal no caso de cópia por má fé.

6. Durante a apresentação poderão ser feitas perguntas relativas ao trabalho, as quais serão consideradas para fim de avaliação. Todos os componentes devem comparecer e serem capazes de responder a **quaisquer perguntas e/ou alterar o código de qualquer parte do trabalho**. A avaliação será individual e poderá conter parte escrita. Apenas os grupos que tiverem postado o trabalho poderão fazer prova escrita.
7. A avaliação será baseada nos seguintes critérios:
 - Correção, robustez e eficiência dos programas quanto ao tempo de processamento e uso de memória
 - Conformidade às especificações
 - Clareza e estilo de codificação (comentários, endentação, escolha de nomes para identificadores, parametrização)
 - Relatório
 - Avaliação individual