

Trabalho de Implementação

Valor: 35 pontos

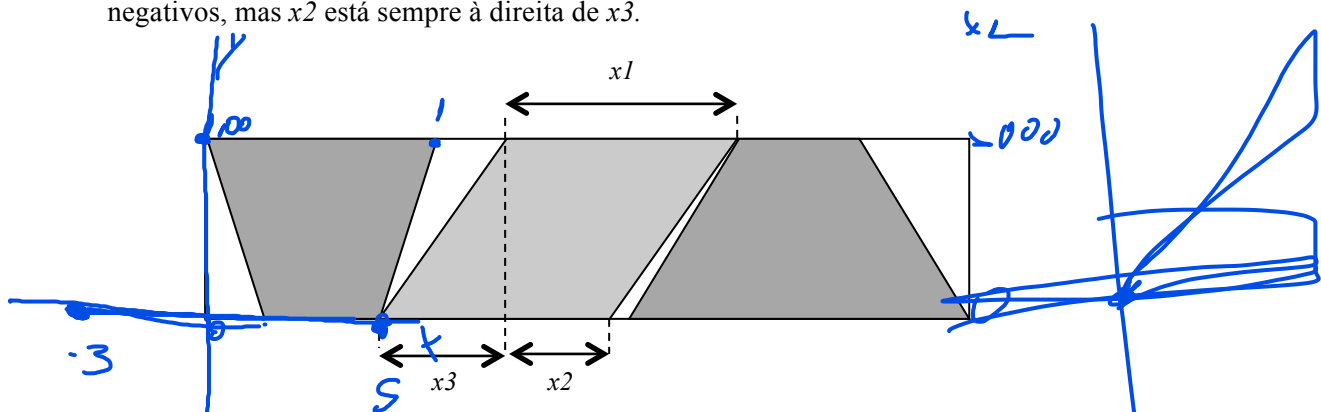
Data: 01/12/2022 pelo Canvas

Penalidade por atraso: 5 pontos por dia. Não serão aceitos trabalhos com atraso após 07/12/2022.

Descrição

Problemas de otimização são bastante comuns em vários setores produtivos. Um exemplo de aplicação é o planejamento de recorte de peças de tecido para confecção de roupas. 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 tecido.

Neste trabalho você deverá implementar uma solução para o problema de corte de peças que minimize o gasto total de tecido. Vamos considerar que as peças têm formato trapezoidal com altura h igual ao do tecido (100 cm) e que não podem ser rotacionadas por motivos estéticos. O gasto de tecido será calculado considerando o comprimento mínimo necessário para acomodar todas as peças. A diferença entre a área do tecido utilizado e a soma das áreas das peças é considerado desperdício. Como a altura é constante, cada peça pode ser descrita através de 3 coordenadas relativas à posição dos vértices, em cm, tomando-se o vértice superior esquerdo como referência. A figura abaixo mostra um exemplo de posicionamento de 3 peças, onde a parte em branco representa a sobra de tecido para essa solução. A peça central exemplifica a descrição da forma do trapézio através de 3 coordenadas. Os vértices que estiverem à esquerda do vértice de referência terão valor negativo. O vértice $x1$ é sempre positivo; $x2$ e $x3$ podem ser negativos, mas $x2$ está sempre à direita de $x3$.



1. Escreva um procedimento que receba a representação $(x1, x2, x3)$ de 2 peças, A e B, e calcule a posição do vértice superior esquerdo de B em relação ao vértice superior esquerdo de A que resulte no menor desperdício de tecido, considerando que B está à direita de A. O valor do desperdício também deve ser calculado em cm^2 .
2. Programe uma interface gráfica para o projeto para exibir uma sequência de 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 as coordenadas $x1, x2, x3$, em cm, dos vértices do trapézio. As coordenadas são números reais separados por espaço em branco. Exemplo:

3
10.0 20.0 -10.0
1.5 10.0 8.5
10.0 20.0 0.0

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 desperdício gerado. Guarde a sequência de peças que produza o menor desperdício 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 que se mostrarem com 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.
6. Documente as soluções e os testes, na forma de um relatório técnico em formato PDF, segundo o padrão da PUC (site biblioteca) ou da SBC, contendo as seguintes seções:
 - a) Introdução: Descrever de maneira geral o objetivo do trabalho.
 - b) Solução proposta: Descrever de forma resumida os algoritmos usados para a solução do problema e analisar a sua ordem de complexidade.
 - c) Implementação: Descrever detalhes dos programas implementados, principalmente aqueles utilizados para melhorar a eficiência da solução e como se organiza a interface gráfica.
 - d) Relatório de testes: Descrever os testes realizados e seus resultados, mostrando como ficou a interface em cada caso. Registrar o tempo de execução de cada um deles e o sistema computacional utilizado.
 - e) Conclusão: Discutir os resultados obtidos, comparando as soluções por força-bruta, branch-and-bound e heurística, quanto à sua ordem de complexidade e tempo de execução.
 - f) Bibliografia segundo o padrão ABNT.

Considerações gerais e critérios de avaliação

1. O trabalho deverá ser feito em grupos de dois ou três alunos, sem qualquer participação de outros grupos e/ou ajuda de terceiros. Cada aluno deve participar ativamente em todas as etapas do trabalho. Os componentes dos grupos devem ser informados no prazo de 2 semanas, através de e-mail para alexeimcmachado@gmail.com e não poderão ser alterados. Os alunos que não tiverem feito grupos até esta data serão agrupados pelo professor de maneira arbitrária, em grupos de 2 ou 3 alunos.
2. A codificação do trabalho deve ser feita em linguagem Python, Java, C ou C++, de forma independente de IDEs. Verifique se as bibliotecas gráficas e versões de linguagens estejam disponíveis nos laboratórios, caso contrário o grupo deve trazer um notebook para a apresentação. Apenas funções básicas das bibliotecas podem ser usadas, ou seja, os algoritmos de solução para o problema devem ser implementados exclusivamente pelos componentes do grupo.

3. Os trabalhos (código e relatório) devem ser postados na forma de um arquivo compactado em padrão zip, com **tamanho máximo de 5MB**, e seu nome deve ser o número de matrícula de um dos componentes (Ex:346542.zip). **Os arquivos fontes devem estar no diretório raiz** e devem conter o nome de todos os componentes do grupo no início do código.
4. **Trabalhos iguais, na sua totalidade ou em partes, 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. A utilização do Github deve ser feita de forma privada. 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.
7. A avaliação será baseada nos seguintes critérios:
 - Correção, robustez e eficiência dos programas
 - Conformidade às especificações
 - Clareza e estilo de codificação (comentários, endentação, escolha de nomes para identificadores, parametrização)
 - Relatório
 - Apresentação individual