

Trabalho Prático Final
GCC118 - Programação Matemática
PCC540 - Linear and Integer Programming
Universidade Federal de Lavras

Profa. Andreza C. Beezão Moreira
Prof. Mayron César O. Moreira

21 de outubro de 2025

1 Introdução

O *Problema de Balanceamento de Linhas de Produção e Designação de Trabalhadores* (ALWABP, do inglês: *Assembly Line Worker Assignment and Balancing Problem*) completa 18 anos de sua primeira publicação, atribuída a “Miralles, C., Garcia-Sabater, J. P., Andres, C., & Cardos, M. (2007). *Advantages of assembly lines in sheltered work centres for disabled. A case study. International Journal of Production Economics*, 110(1-2), 187-197”. Trata-se de uma primeira iniciativa, do ponto de vista da Pesquisa Operacional, em trabalhar de forma eficiente a presença de trabalhadores com deficiência em uma linha de produção. Trabalhos posteriores mostraram que o problema pode ser generalizado em um contexto amplo, ao qual cada trabalhador pode ser unicamente considerado devido aos tempos distintos de execução das tarefas.

Formalmente, a versão clássica do ALWABP considera uma linha de modelo único, do tipo *straight line*. Considere um conjunto de estações $S = \{1, \dots, m\}$ linearmente ordenadas, e um conjunto de trabalhadores $W = \{1, \dots, k\}$, com $|S| = |W|$. Cada tarefa do conjunto $N = \{1, \dots, n\}$ deve ser designada a uma estação da linha. As tarefas a serem executadas e suas relações são estabelecidas por um grafo direcionado e topologicamente ordenado $G = (V, E)$, em que $i \preceq j$ indica que a tarefa i deve ser executada em uma estação de mesmo índice ou de índice inferior à estação que executa a tarefa j . Cada trabalhador deve ser alocado a exatamente uma estação, e cada estação deve possuir um único trabalhador. Define-se $t_{wi} \in \mathbb{R}_+$ como o tempo de execução da tarefa i pelo trabalhador w . Além disso, considera-se $I_w \subseteq N$ como o conjunto de tarefas que o trabalhador w é incapaz de executar ($t_{wi} = \infty, \forall i \in I_w$). Deseja-se a minimização do tempo de ciclo da linha de produção, ou seja, a minimização do tempo de execução da estação mais carregada, respeitando todas as restrições do problema.

2 Objetivos

Essa atividade avaliativa visa verificar o aprendizado do discente em termos dos conteúdos de Programação Linear e Inteira vistos no curso de GCC118/PCC540. Para tanto:

1. Formule o problema como programa linear inteiro.
2. Resolva as instâncias propostas com um solver genérico (CPLEX, Gurobi, Cbc, Coin-OR, GLPK, ou SCIP).
3. Defina e implemente a meta-heurística definida logo abaixo, no item 3.
4. Resolva as instâncias definidas com a meta-heurística.
5. Documente e analise os experimentos em um relatório.
6. Apresente os resultados em aula.

3 Regras

1. Metaheurística: **a ser definida em aula para cada grupo.**
2. O trabalho deve ser feito em até 3 pessoas. Sugiro que façam em grupos.
3. O objetivo do trabalho é conhecer uma meta-heurística profundamente e ganhar experiência prática para aplicá-la em novos problemas.
4. Todas escolhas feitas para aplicar a meta-heurística para o problema em questão devem ser claramente relatadas. Isso inclui a representação do problema, a função objetivo, a geração da solução inicial, a vizinhança e a estratégia de escolha em caso de buscas locais, os operadores (crossover, mutação) em caso de algoritmos genéticos, outros parâmetros do métodos (temperatura, lista tabu e tenure), critério de parada.
5. Cuidado com a documentação das instâncias, tempo de execução, parâmetros, número de experimentos, semente do gerador randômico, dados experimentais, etc. Apresente o método de escolha de parâmetros.
6. Em particular, para métodos estocásticos, os valores apresentados devem ser médias de pelo menos 5 replicações de cada experimento com sementes diferentes.
7. Todas implementações devem aceitar uma instância no formato do problema na entrada padrão (stdin) e imprimir a melhor solução encontrada na saída padrão (stdout).
8. Os principais parâmetros do método devem ser definíveis pela linha de comando. O primeiro parâmetro da linha de comando é o nome de um arquivo para gravar a melhor solução encontrada.

9. Qualquer tentativa de plágio será devidamente punida pelos docentes com medidas administrativas cabíveis.
10. Todo código externo deve ser devidamente citado.

4 Entregáveis

O grupo deverá criar um repositório no GitHub, e nesse repositório, apresentar:

- Código fonte e README bem documentado.
- Um relatório com a documentação da solução com resultados e discussão (veja um exemplo). Elementos obrigatórios: Introdução, Formulação, Algoritmo proposto, Resultados obtidos com análise, Conclusão e Bibliografia.
- Uma implementação do modelo matemático e da heurística. Para a codificação do método heurístico, não é permitido o uso de bibliotecas proprietárias.
- Nos resultados computacionais, deve-se criar uma tabela informando, para cada instância: valor da solução inicial (SI), valor da solução final (SF), desvio percentual da solução final em relação à inicial calculado como $100 \times (SI - SF)/SI$, desvio percentual da SF em relação à solução ótima, tempo computacional da metaheurística, tempo computacional da resolução via solver.
- Protocolos das execuções do *solver* e da meta-heurística e outros dados experimentais detalhados.
- No relatório, teste apenas as instâncias indicadas no *link* https://drive.google.com/file/d/1Sua_gXlRM_1Ck-jQBqjztg_eZti2kGwg/view?usp=sharing.
- Uma apresentação aos professores da disciplina.
- **Forma de entrega:** apenas um membro do grupo enviará um e-mail para **ambos os professores** (mayron.moreira@ufla.br e andreza.moreira@ufla.br) como link do repositório do trabalho. Dediquem-se ao fazer que o repositório seja bem documentado, e que seu código seja facilmente executado.
- Data de envio do trabalho: até às 23h55 do dia 07/12/2025.

5 Instâncias teste

As instâncias estão disponíveis em: https://drive.google.com/file/d/1CveaBibixwLSQh_1vx4IFYMtUKhBhIt/view?usp=sharing. O formato das instâncias pode ser conferido no Campus Virtual. Além disso, os melhores *upper bounds* de cada instância constam em https://drive.google.com/file/d/1f9-oysuJZU_0Mzq42cRoMZU17mgnHQBK/view?usp=sharing, coluna “UB”.

6 Desafio extra

A análise do espaço de instâncias é uma ferramenta muito útil para identificar características das instâncias que possam explicar o comportamento de alguns algoritmos. Trabalhos tais como “Smith-Miles, K., & Bowly, S. (2015). Generating new test instances by evolving in instance space. *Computers & Operations Research*, 63, 102-113.” e <https://matilda.unimelb.edu.au/matilda/> são referências nesse contexto. Os grupos que realizarem uma proposta satisfatória de *instance space analysis* (ISA), e apresentá-la no relatório final, podem ter um acréscimo de até 5 pontos na média final da disciplina. Como referência de trabalho que utiliza a ISA no contexto de linhas de produção, segue o link: <https://drive.google.com/file/d/10-OpLWCIPuPV0TA3XcQtH2002nf1PFm1/view?usp=sharing>.