



Planejamento de Trajetória para Soldagem por Pontos com Braços Robóticos em Portas de Veículos

Regina Araújo

June 6, 2025

Descrição Geral

O problema modelado é baseado em um processo real de soldagem de portas em uma fábrica de carrocerias automotivas. O objetivo principal é **minimizar o tempo de ciclo** respeitando as restrições operacionais.

Contexto Operacional

- 2 plataformas de soldagem
- Cada plataforma possui 4 braços robóticos

Características do Processo

- 110 pontos de solda por porta
- 4 tipos distintos de soldagem

Observação Importante

- Devido à confidencialidade, não foram utilizados dados reais

Motivação para a Simplificação

Devido à complexidade envolvida na modelagem e na obtenção de soluções aplicáveis ao problema real, optou-se por trabalhar com uma **versão simplificada** do processo.

Abordagem Adotada

- Considera-se uma célula de soldagem com **número reduzido de pontos de solda**;
- Essa simplificação visa **facilitar a análise** e o **desenvolvimento de métodos de solução**.

Objetivo

- Criar um modelo representativo que mantenha a essência do problema original, mas com menor complexidade computacional.

Conjuntos

- M : Conjunto de máquinas

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Parâmetros

- dist_{ij} : Distância euclidiana entre i e j , para $i, j \in V$.

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Parâmetros

- dist_{ij} : Distância euclidiana entre i e j , para $i, j \in V$.
- v : Velocidade do robô.

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Parâmetros

- dist_{ij} : Distância euclidiana entre i e j , para $i, j \in V$.
- v : Velocidade do robô.
- t_r : Tempo de resfriamento da solda.

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Parâmetros

- dist_{ij} : Distância euclidiana entre i e j , para $i, j \in V$.
- v : Velocidade do robô.
- t_r : Tempo de resfriamento da solda.
- t_s : Tempo de setup (troca de tipo).

Conjuntos

- M : Conjunto de máquinas
- P : Conjunto de pontos de solda
- $V = M \cup P$: Conjunto de todos os vértices

Parâmetros

- dist_{ij} : Distância euclidiana entre i e j , para $i, j \in V$.
- v : Velocidade do robô.
- t_r : Tempo de resfriamento da solda.
- t_s : Tempo de setup (troca de tipo).
- t_{ij} : Tempo total de i a j :

$$t_{ij} = \frac{\text{dist}_{ij}}{v} + t_r + \begin{cases} t_s & \text{se } i, j \in P \text{ e tipos diferentes,} \\ 0 & \text{caso contrário.} \end{cases}$$

Variáveis de Decisão

- $x_{kij} \in \{0, 1\}$:

$$\begin{cases} 1, & \text{se a máquina } k \text{ vai de } i \text{ a } j, \forall i, j \in P, k \in M, \\ 0, & \text{caso contrário.} \end{cases}$$

Variáveis de Decisão

- $x_{kij} \in \{0, 1\}$:

$$\begin{cases} 1, & \text{se a máquina } k \text{ vai de } i \text{ a } j, \forall i, j \in P, k \in M, \\ 0, & \text{caso contrário.} \end{cases}$$

- $u_{ki} \geq 0$: Variável auxiliar para eliminar sub-rotas (MTZ), $\forall i \in P, k \in M$.

Variáveis de Decisão

- $x_{kij} \in \{0, 1\}$:

$$\begin{cases} 1, & \text{se a máquina } k \text{ vai de } i \text{ a } j, \forall i, j \in P, k \in M, \\ 0, & \text{caso contrário.} \end{cases}$$

- $u_{ki} \geq 0$: Variável auxiliar para eliminar sub-rotas (MTZ), $\forall i \in P, k \in M$.
- $Z \geq 0$: Tempo de ciclo total (*makespan*).

$$\min Z$$

Sujeito a:

$$\sum_{k \in M} \sum_{i \in V} x_{kij} = 1 \quad \forall j \in P \quad (1)$$

(3)

- (1): Cada ponto $i, j \in P$ é visitado uma única vez.

$$\min Z$$

Sujeito a:

$$\sum_{k \in M} \sum_{i \in V} x_{kij} = 1 \quad \forall j \in P \quad (1)$$

$$\sum_{j \in P} x_{kkj} = 1 \quad \forall k \in M \quad (2)$$

$$(3)$$

- (1): Cada ponto $i, j \in P$ é visitado uma única vez.
- (2): Cada máquina k inicia no seu vértice.

$$\min Z$$

Sujeito a:

$$\sum_{k \in M} \sum_{i \in V} x_{kij} = 1 \quad \forall j \in P \quad (1)$$

$$\sum_{j \in P} x_{kkj} = 1 \quad \forall k \in M \quad (2)$$

$$\sum_{i \in P} x_{kik} = 1 \quad \forall k \in M \quad (3)$$

- (1): Cada ponto $i, j \in P$ é visitado uma única vez.
- (2): Cada máquina k inicia no seu vértice.
- (3): Cada máquina k termina no seu vértice.

$$x_{kij} = 0 \quad \forall k \in M, i \in M, j \in V \quad (4)$$

(7)

- (4): Máquinas não visitam outras máquinas.

$$x_{kij} = 0 \quad \forall k \in M, i \in M, j \in V \quad (4)$$

$$\sum_{i \in V} x_{kij} = \sum_{i \in V} x_{kji} \quad \forall k \in M, j \in V \quad (5)$$

(7)

- (4): Máquinas não visitam outras máquinas.
- (5): Conservação de fluxo em cada ponto j .

$$x_{kij} = 0 \quad \forall k \in M, i \in M, j \in V \quad (4)$$

$$\sum_{i \in V} x_{kij} = \sum_{i \in V} x_{kji} \quad \forall k \in M, j \in V \quad (5)$$

$$u_{ki} - u_{kj} + |V| \cdot x_{kij} \leq |V| - 1 \quad \forall k \in M, i, j \in P, i \neq j \quad (6)$$

$$(7)$$

- (4): Máquinas não visitam outras máquinas.
- (5): Conservação de fluxo em cada ponto j .
- (6): Restrição MTZ para evitar sub-rotas.

$$x_{kij} = 0 \quad \forall k \in M, i \in M, j \in V \quad (4)$$

$$\sum_{i \in V} x_{kij} = \sum_{i \in V} x_{kji} \quad \forall k \in M, j \in V \quad (5)$$

$$u_{ki} - u_{kj} + |V| \cdot x_{kij} \leq |V| - 1 \quad \forall k \in M, i, j \in P, i \neq j \quad (6)$$

$$Z \geq \sum_{i \in V} \sum_{j \in V} t_{ij} \cdot x_{kij} \quad \forall k \in M \quad (7)$$

- (4): Máquinas não visitam outras máquinas.
- (5): Conservação de fluxo em cada ponto j .
- (6): Restrição MTZ para evitar sub-rotas.
- (7): Limitação do tempo de ciclo por máquina.

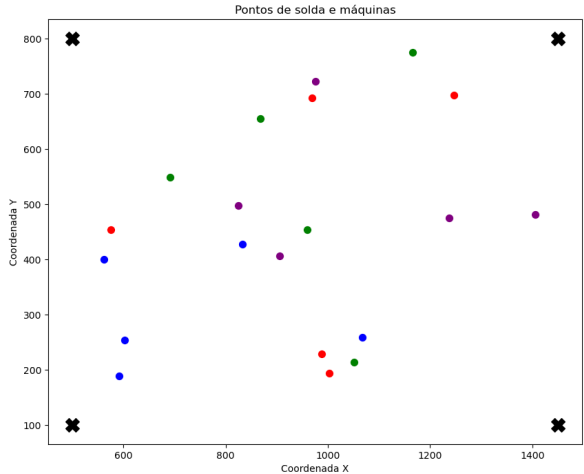
O modelo foi implementado utilizando:

- **Jupyter Notebook:** Ambiente interativo para desenvolvimento e testes
- **Python:** Linguagem de programação principal
- **Bibliotecas:**
 - ▶ `gurobipy`: Interface Python para o solver Gurobi
 - ▶ `numpy`: Manipulação numérica eficiente
 - ▶ `matplotlib`: Visualização dos resultados
- **Solver:** Gurobi Optimizer versão 12.0.2 (solver comercial usando licença acadêmica)

1. Configuração inicial do ambiente
2. Geração dos dados sintéticos com NumPy (usando como base os dados reais)
3. Cálculo das matrizes de distância e tempo
4. Construção passo a passo do modelo:
 - 4.1 Definição das variáveis
 - 4.2 Adição das restrições
 - 4.3 Especificação da função objetivo
5. Otimização com Gurobi
6. Visualização com Matplotlib

Geração de Dados - Cenário

- **20 pontos aleatórios** distribuídos em **4 tipos**
- **Velocidade do robô:** 40 mm/s
- **Tempo de resfriamento:** 2 s
- **Setup inicial:** 10 s

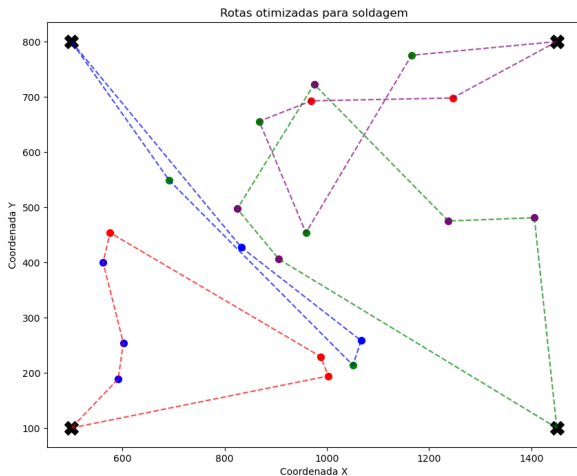


Resultado da Otimização

- **Tempo de resolução com Gurobi: 610 segundos**
- **Makespan (tempo máximo):**
61.08 unidades de tempo

Rotas das máquinas e seus tempos:

- **M1 (59.47):** M1 → 1 (A) → 2 (A) → 5 (A) → 6 (B) → 9 (B) → 7 (B) → M1
- **M2 (61.08):** M2 → 10 (B) → 8 (B) → 14 (C) → 11 (C) → M2
- **M3 (60.08):** M3 → 18 (D) → 17 (D) → 16 (D) → 19 (D) → 20 (D) → M3
- **M4 (59.56):** M4 → 15 (C) → 12 (C) → 13 (C) → 4 (A) → 3 (A) → M4



Comparação entre Cenários

Cenário	Pontos	Velocidade (mm/s)	Setup (s)	Makespan	Tempo Solver (s)
1	20	40	10	61.08	610
2	20	30	10	74.51	1675
3	15	40	10	50.50	5
4	15	30	10	60.66	2
5	15	40	5	45.50	1
6*	20	40	5	<i>timeout</i>	<i>timeout</i>

- O tempo de resolução aumenta com o número de pontos e com menor velocidade.
- A redução no tempo de setup e o aumento da velocidade tendem a melhorar o makespan e o desempenho do solver.
- Em alguns casos, cenários podem apresentar falhas de resolução por limitações do solver (ex: tempo limite).

- O modelo de otimização proposto mostrou-se eficiente para pequenos conjuntos de pontos.
- Variações de velocidade, tempo de setup e distribuição influenciam significativamente o makespan.
- A elevação no tempo de setup pode acarretar problemas, como possíveis colisões.
- O tempo de resolução cresce com o número de pontos e a complexidade do problema.

Obrigada!

Perguntas?

Apresentado por: Regina Araújo
UFPB