

Compêndio de Exemplos para Heurística A* Baseada em Ciclos (Versão Otimizada)

Brandon Mejia

20 de outubro de 2025

Conteúdo

1 A Lógica Central da Heurística	2
1.1 Fase 1: O Diagnóstico (A Pergunta Chave)	2
1.2 Fase 2: O Mapeamento (A Resposta à Pergunta)	2
1.3 Fase 3: A Estratégia (O Tratamento segundo o Diagnóstico)	2
2 Exemplos para Ciclos de Tamanho 2 (k=2)	3
2.1 Exemplo 2.1: Troca Par-Par	3
2.2 Exemplo 2.2: Troca Ímpar-Ímpar	3
2.3 Exemplo 2.3: Troca Par-Ímpar	3
3 Exemplos para Ciclos Pequenos (k=3, k=4)	3
3.1 Exemplo 3.1: Ciclo de Tamanho 3 (Misto)	3
3.2 Exemplo 3.2: Ciclo de Tamanho 3 (Apenas Ímpares)	4
3.3 Exemplo 3.3: Ciclo de Tamanho 4 (Misto)	4
4 Exemplos para Ciclos Grandes (k maior que 4)	5
4.1 Caso A: Ciclos com Números Pares	5
4.1.1 Exemplo 4.1: Ciclo k=5 (2 Pares, 3 Ímpares)	5
4.1.2 Exemplo 4.2: Ciclo k=6 (4 Pares, 2 Ímpares)	5
4.1.3 Exemplo 4.3: Ciclo k=7 (1 Par, 6 Ímpares)	5
4.2 Caso B: Ciclos Apenas com Números Ímpares	5
4.2.1 Exemplo 4.4: k=5 com "Empréstimo" Possível	5
4.2.2 Exemplo 4.5: k=6 SEM Pares no Array	6
4.2.3 Exemplo 4.6: k=7 com "Empréstimo" Possível	6

1 A Lógica Central da Heurística

Esta secção resume o fluxo de "pensamento" do algoritmo, desde a identificação do problema até à aplicação de uma estratégia específica, refletindo a versão mais recente e otimizada.

1.1 Fase 1: O Diagnóstico (A Pergunta Chave)

A primeira e mais importante pergunta que a heurística procura responder é:

"Como é que as peças que estão fora do lugar se ligam umas às outras?"

1.2 Fase 2: O Mapeamento (A Resposta à Pergunta)

A resposta a essa pergunta é encontrada ao construir a **estrutura de ciclos**. Uma vez completada esta fase, o algoritmo já não sabe apenas *quantas* peças estão erradas, mas sim *como* os seus erros se relacionam entre si.

1.3 Fase 3: A Estratégia (O Tratamento segundo o Diagnóstico)

O algoritmo analisa cada ciclo de forma individual e escolhe uma estratégia segundo o **TMANHO** desse ciclo.

1. **Ciclo de Tamanho 2:** Resolve-se com um único intercâmbio direto. É o caso mais simples e o custo calculado é o custo real e exato.
2. **Ciclo de Tamanho 3 ou 4: (Lógica Otimizada)** Aplica-se uma busca inteligente. Em vez de testar todas as combinações possíveis, o algoritmo identifica um conjunto mínimo de $k - 1$ trocas que garantidamente resolve o ciclo. De seguida, testa apenas as **permutações** (as diferentes ordens) desse pequeno conjunto para encontrar a mais barata.
3. **Ciclo de Tamanho Grande (k superior a 4):** Adota-se uma estratégia "inteligente e económica":
 - A) **Se o ciclo contém pelo menos um número par:** Usa-se a estratégia do "pivô par" para evitar os dispendiosos intercâmbios entre ímpares. O custo é estimado com a fórmula de limite inferior (lower bound):

$$(N_{pares} - 1) \times 2 + (N_{ímpares}) \times 11$$

- B) **Se o ciclo só contém números ímpares:** Faz-se uma análise de custo-benefício, comparando duas opções:
 - (a) *Opção Interna:* Resolver o ciclo com $k - 1$ trocas internas entre ímpares (custo alto: $(k - 1) \times 20$).
 - (b) *Opção "Empréstimo":* "Tomar emprestado" um número par de outra parte do array e usá-lo como pivô (custo: $k \times 11$).

A heurística escolhe o custo **MÍNIMO** entre estas duas opções.

2 Exemplos para Ciclos de Tamanho 2 ($k=2$)

2.1 Exemplo 2.1: Troca Par-Par

```
data: [1, 4, 3, 2]
goal: [1, 2, 3, 4]
```

Identificação: O ciclo é ($\text{índice } 1 \leftrightarrow 3$), envolvendo os valores $\{4, 2\}$.

Análise e Custo: Ambos os valores são Pares. A troca é Par-Par.

Custo da Heurística = 2

2.2 Exemplo 2.2: Troca Ímpar-Ímpar

```
data: [5, 2, 3, 4, 1]
goal: [1, 2, 3, 4, 5]
```

Identificação: O ciclo é ($\text{índice } 0 \leftrightarrow 4$), envolvendo os valores $\{5, 1\}$.

Análise e Custo: Ambos os valores são Ímpares. A troca é Ímpar-Ímpar.

Custo da Heurística = 20

2.3 Exemplo 2.3: Troca Par-Ímpar

```
data: [1, 2, 8, 4, 3]
goal: [1, 2, 3, 4, 8]
```

Identificação: O ciclo é ($\text{índice } 2 \leftrightarrow 4$), envolvendo os valores $\{8, 3\}$.

Análise e Custo: Um valor é Par (8) e o outro Ímpar (3). A troca é Par-Ímpar.

Custo da Heurística = 11

3 Exemplos para Ciclos Pequenos ($k=3, k=4$)

(*Esta lógica foi radicalmente otimizada. A explicação reflete o novo método.*)

3.1 Exemplo 3.1: Ciclo de Tamanho 3 (Misto)

```
data: [2, 3, 1]
goal: [1, 2, 3]
```

Identificação: O ciclo é ($0 \rightarrow 1 \rightarrow 2 \rightarrow 0$), envolvendo $\{2, 3, 1\}$.

Análise (Método Otimizado): O algoritmo identifica um conjunto de $k - 1 = 2$ trocas que resolvem o ciclo (ex: usando o ‘2’ como pivô: ‘(2,3)’ e ‘(2,1)’). De seguida, testa as $2! = 2$ permutações:

- *Ordem 1:* Trocar ‘(2,3)’ (custo 11), depois ‘(3,1)’ (custo 11). Total: 22.
- *Ordem 2:* Trocar ‘(2,1)’ (custo 11), depois ‘(1,3)’ (custo 20). Total: 31.

Custo da Heurística = min(22, 31) = 22

3.2 Exemplo 3.2: Ciclo de Tamanho 3 (Apenas Ímpares)

```
data: [3, 5, 1]
goal: [1, 3, 5]
```

Identificação: O ciclo é $(0 \rightarrow 1 \rightarrow 2 \rightarrow 0)$, envolvendo $\{3, 5, 1\}$.

Análise (Método Otimizado): O conjunto de trocas resolutivas é, por exemplo, ' $(3,5)$ ' e ' $(3,1)$ '. Ambas são Ímpar-Ímpar (custo 20). Qualquer permutação destas duas trocas resultará no mesmo custo total.

Custo da Heurística = $20 + 20 = 40$

3.3 Exemplo 3.3: Ciclo de Tamanho 4 (Misto)

```
data: [2, 3, 4, 1]
goal: [1, 2, 3, 4]
```

Identificação: O ciclo é $(0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0)$, envolvendo $\{2, 3, 4, 1\}$.

Análise (Método Otimizado): O algoritmo antigo testaria $6^3 = 216$ sequências. O novo identifica um conjunto de 3 trocas resolutivas (ex: ' $(2,3)$ ', ' $(2,4)$ ', ' $(2,1)$ '). Depois, testa apenas as $3! = 6$ permutações dessas 3 trocas.

- Uma dessas permutações é: ' $(2,1)$ ' (custo 11), ' $(1,3)$ ' (custo 20), ' $(3,4)$ ' (custo 11). Total: 42.
- Outra é: ' $(2,4)$ ' (custo 2), ' $(4,3)$ ' (custo 11), ' $(3,1)$ ' (custo 20). Total: 33.

A busca recursiva explora as 6 permutações e encontra a mais barata.

Custo da Heurística = 24

4 Exemplos para Ciclos Grandes (k maior que 4)

4.1 Caso A: Ciclos com Números Pares

4.1.1 Exemplo 4.1: Ciclo k=5 (2 Pares, 3 Ímpares)

```
data: [12, 13, 14, 15, 11]
goal: [11, 12, 13, 14, 15]
```

Identificação: Ciclo de tamanho 5 envolvendo {12, 13, 14, 15, 11}.

Análise: $N_{pares} = 2$ (12, 14), $N_{ímpares}=3$ (11, 13, 15).

Custo: $(2 - 1) \times 2 + 3 \times 11 = 1 \times 2 + 33 = 35$.

Custo da Heurística = 35

4.1.2 Exemplo 4.2: Ciclo k=6 (4 Pares, 2 Ímpares)

```
data: [2, 4, 3, 6, 5, 8]
goal: [8, 2, 3, 4, 5, 6]
```

Identificação: Ciclo de tamanho 6 envolvendo {2, 4, 3, 6, 5, 8}.

Análise: $N_{pares} = 4$ (2, 4, 6, 8), $N_{ímpares}=2$ (3, 5).

Custo: $(4 - 1) \times 2 + 2 \times 11 = 3 \times 2 + 22 = 6 + 22 = 28$.

Custo da Heurística = 28

4.1.3 Exemplo 4.3: Ciclo k=7 (1 Par, 6 Ímpares)

```
data: [2, 3, 5, 7, 9, 11, 1]
goal: [1, 2, 3, 5, 7, 9, 11]
```

Identificação: Ciclo de tamanho 7 envolvendo {2, 3, 5, 7, 9, 11, 1}.

Análise: $N_{pares} = 1$ (só o 2), $N_{ímpares}=6$ (os restantes).

Custo: $(1 - 1) \times 2 + 6 \times 11 = 0 \times 2 + 66 = 66$.

Custo da Heurística = 66

4.2 Caso B: Ciclos Apenas com Números Ímpares

4.2.1 Exemplo 4.4: k=5 com "Empréstimo" Possível

```
data: [3, 5, 7, 9, 1, 100]
goal: [1, 3, 5, 7, 9, 100]
```

Identificação: Ciclo de tamanho 5 com {3, 5, 7, 9, 1}. O array contém um par (100).

Análise e Custo:

- Opção Interna: $(5 - 1) \times 20 = 80$.
- Opção "Empréstimo": $5 \times 11 = 55$.

Custo da Heurística = min(80, 55) = 55

4.2.2 Exemplo 4.5: k=6 SEM Pares no Array

```
data: [3, 5, 7, 9, 11, 1]
goal: [1, 3, 5, 7, 9, 11]
```

Identificação: Ciclo de tamanho 6 com $\{3, 5, 7, 9, 11, 1\}$. O array não tem pares.

Análise e Custo: A opção de "empréstimo" é impossível. Apenas a estratégia interna é válida.
 $Custo = (6 - 1) \times 20 = 5 \times 20 = 100$.

Custo da Heurística = 100

4.2.3 Exemplo 4.6: k=7 com "Empréstimo" Possível

```
data: [3, 5, 7, 9, 11, 13, 1, 2]
goal: [1, 3, 5, 7, 9, 11, 13, 2]
```

Identificação: Ciclo de tamanho 7 com $\{3, 5, 7, 9, 11, 13, 1\}$. O array contém um par (2).

Análise e Custo:

- Opção Interna: $(7 - 1) \times 20 = 120$.
- Opção "Empréstimo": $7 \times 11 = 77$.

Custo da Heurística = min(120, 77) = 77