

Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search

Ruey-Maw Chen * and Hsiao-Fang Shih

Department of Computer Science and Information Engineering, National Chinyi University of Technology, Taichung, Taiwan; E-Mail:
iris@ncut.edu.tw

Integrantes do Grupo:

Gabriel Nicholas Pires de Moraes - 12111BCC052

Rafael Borges Morais - 11911BCC040

Victor Hugo Nunes - 12021BCC023

Estrutura do problema

20 cursos, ex da estrutura de um curso:

```
{'course_id': 'COURSE1', 'teacher': 'T7', 'class': 'C4', 'room': 'R8', 'duration': 2}
```

Temos: 16 professores, 10 turmas, 10 salas e 20 horários (4 por dia)

Objetivo: Associar cada curso a um horário, sem conflitos de professor, turma e horário.

Particula

(ID, Professor, Turma, Sala, Horário, Duração)

```
[('COURSE1', 'T1', 'C4', 'R3', 5, 2)
('COURSE2', 'T2', 'C4', 'R10', 12, 3)
('COURSE3', 'T6', 'C9', 'R4', 6, 3)
('COURSE4', 'T3', 'C7', 'R1', 13, 2)
('COURSE5', 'T7', 'C10', 'R4', 14, 2)
...
('COURSE20', 'T3', 'C6', 'R9', 0, 2)]
```

Restrições

- Professor leciona para apenas uma turma em um horário
- Turma participa de apenas um curso em um horário
- Apenas um curso por sala em um mesmo horário
- Cursos de 3 horas seguidas devem ser no mesmo dia, sem interrupções de almoço
- Professores devem lecionar pelo menos duas vezes na semana
- Deve ser levado em consideração a preferência dos alunos e professores para associação de horários.

Calculo da Velocidade

Standard PSO (SPSO) proposto por Bratton e Kennedy, usa um fator de constrição (K)

$$V_{id}^{t+1} = K[\omega V_{id}^t + c_1 \times Rand1 \times (P_{id} - X_{id}) + c_2 \times Rand2 \times (P_{gd} - X_{id})]$$

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\kappa}}, \quad \varphi = c_1 + c_2, \quad \varphi > 4$$

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1}$$

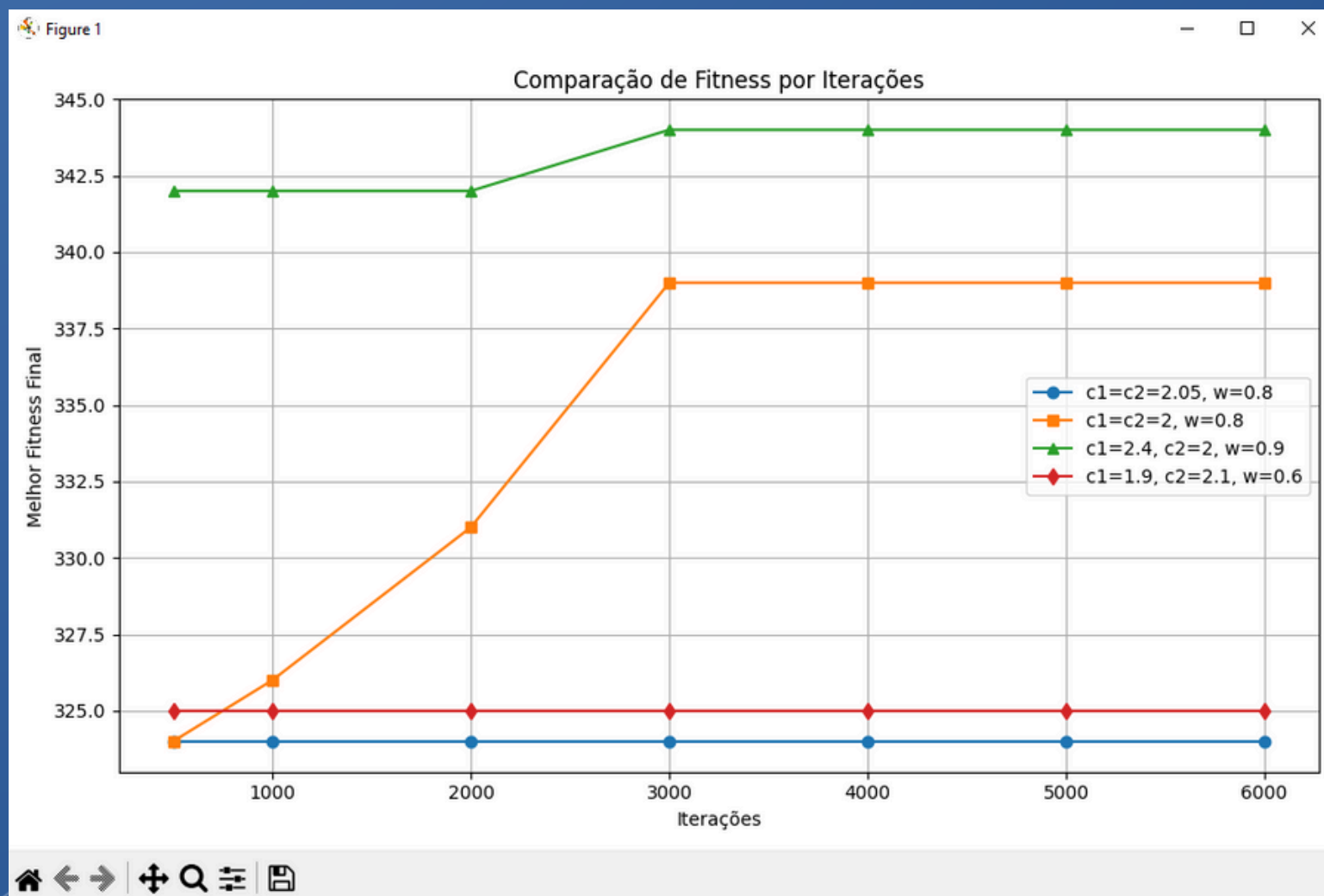
O “X” final será o novo horário alocado para o curso

Hibridização (Busca Local)

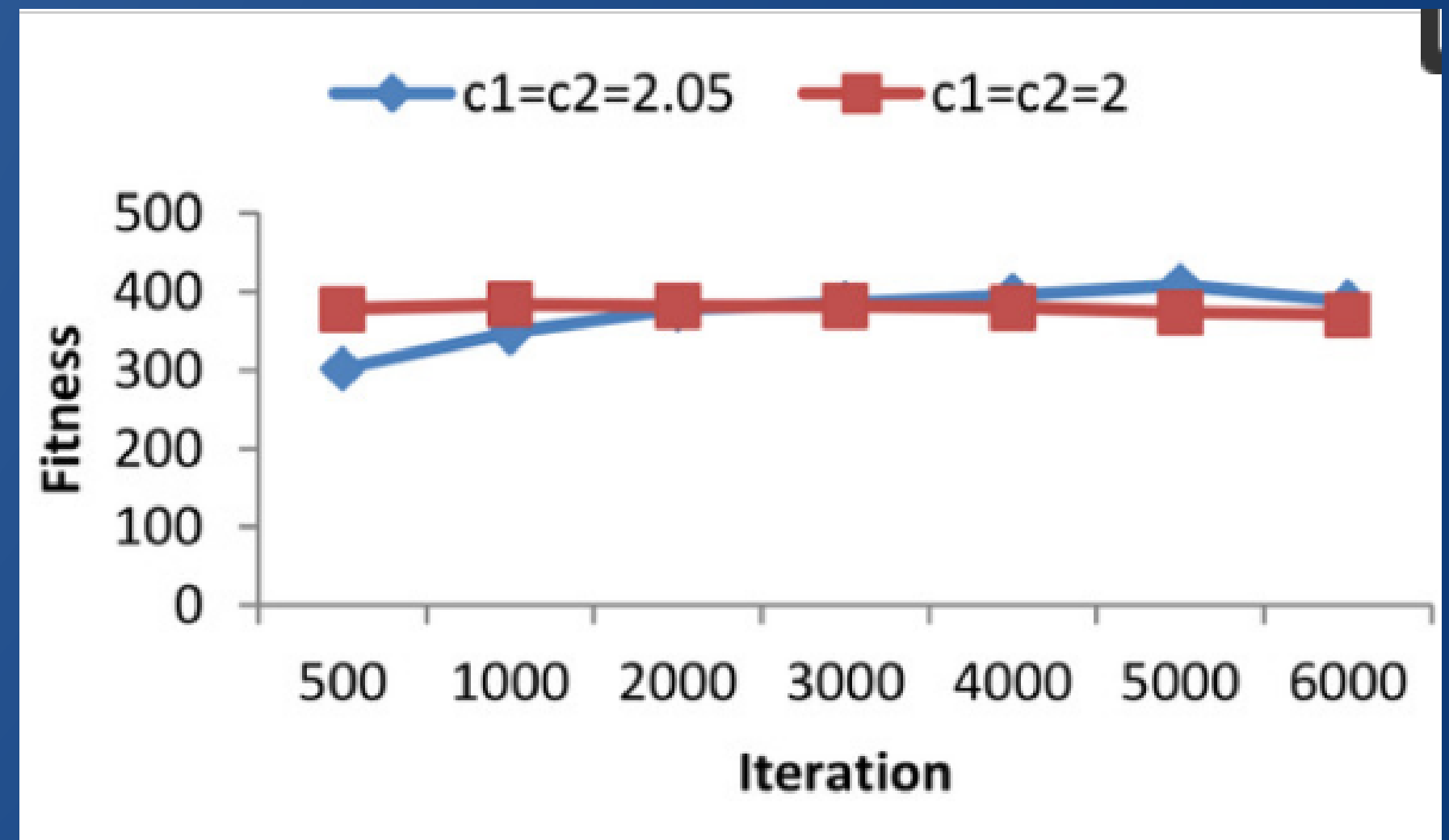
- Após a mudança da posição a partir da velocidade
- Um “swap” é feito em duas partículas aleatórias
- Se essa troca aumentar o fitness, ela é mantida
- Aumentar a diversidade
- Evitar convergência para ótimos locais

Resultados

Nosso trabalho:



Artigo:



$W=0.8$

Artigo do Trabalho 2 Adaptado para PSO

APLICAÇÃO DE ALGORITMOS GENÉTICOS E MÉTODOS EVOLUCIONÁRIOS NA RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE TURMAS: CASO UFPR.

Para fins de comparação, adaptamos o trabalho 2, para usar PSO com as técnicas do trabalho 3.

São gerados um número de indivíduos baseado no tamanho da população, cada indivíduo é representado por um vetor

Indivíduo: onde estão representados as alocações de cada turma



n = nro de turmas

Cada posição contém informações que mostram quais turmas estão alocadas em quais salas:

Ex:

"turma1_seg_08:00": "sala101_seg_08:00"

nome_da_turma e dia/horário_da_turma + nome_da_sala e dia/horário_da_sala

Prioridade: vetor onde são armazenados valores de 0 a 1, que indicarão qual sala deve ser alocada primeiro

0,132	0,543	0,765	0,021	...	0,453
0	1	2	3	...	n

n = nro de turmas

Neste caso por exemplo a turma que esta na posição 2 do vetor de turmas será alocada primeiro em sequencia a segunda maior, até que todas turmas tenham sido alocadas

Além disso também temos o vetor das velocidades, inicialmente com 0's em todas posições

Velocidade: Atributo que será utilizado para atualizar os valores da partícula

0,3	0,5	1,4	4	...	1
0	1	2	3	...	n

n = nro de turmas

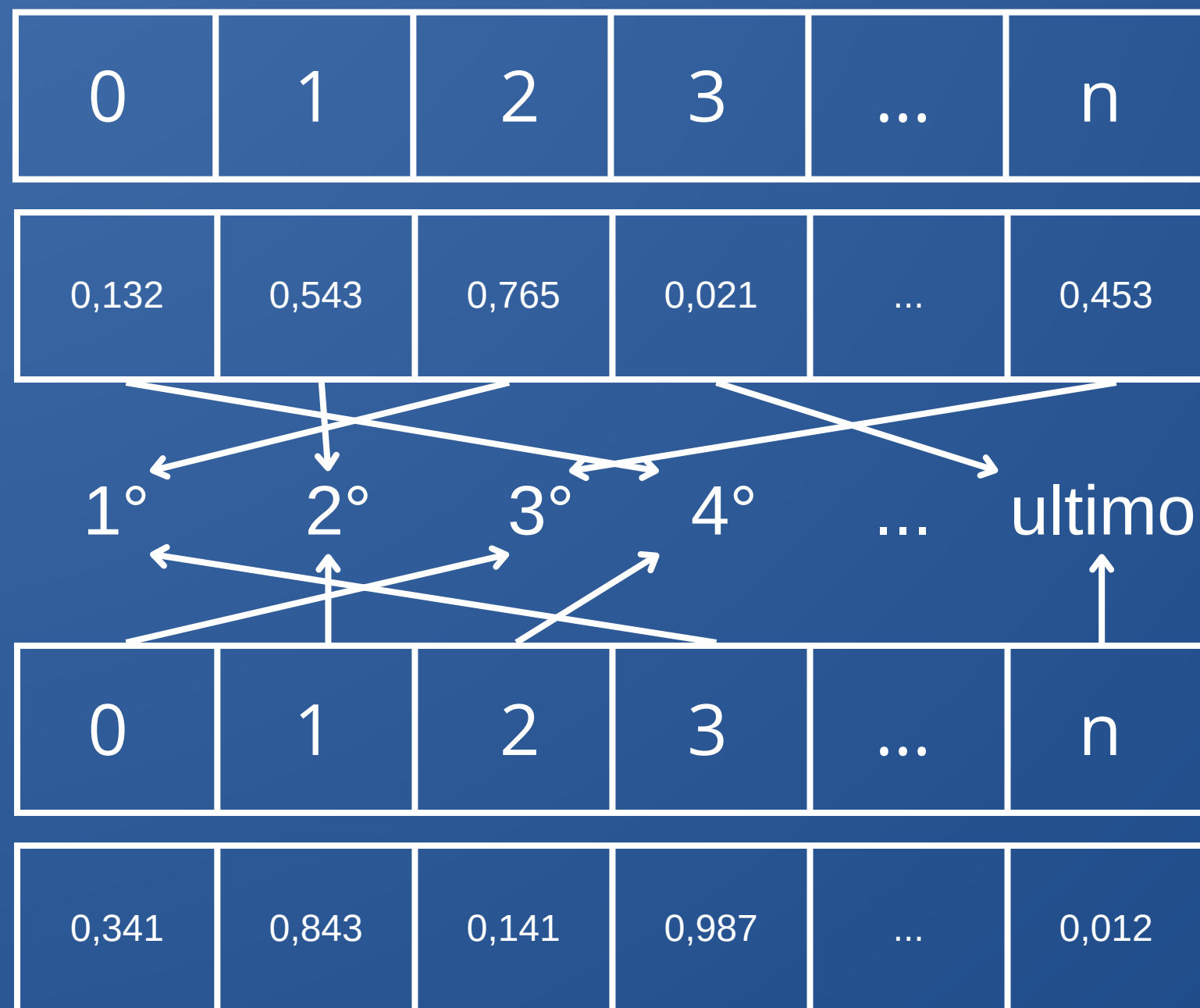
Neste caso por exemplo a turma que esta na posição 0 do vetor de turmas tem velocidade 0,3

População inicial

As turmas são alocadas seguindo a restrição de salas especiais e de horários compatíveis, então cada turma vai sendo alocada em uma sala que não tenha sido ocupada, até que não seja mais possível alocar turmas em salas ou todas as turmas tenham sido alocadas. As turmas são preferencialmente alocadas em salas que tenham mais afinidade, primeiro selecionando o bloco preferencial (se houver), então são alocadas na sala que tenha uma capacidade parecida com a quantidade de alunos da turma, ou seja ($\min(\text{capacidade} - \text{alunos})$).

Dessa forma salas que tenham números maiores em suas partículas são alocadas primeiro em suas salas preferenciais.

População inicial



Desta forma alteramos como são atribuídas as salas

Por exemplo: suponhamos que a uma turma alocada na posição 2 (chamaremos de T2H1) como é a primeira tenha se alocado na sala 1, porém a turma da posição 1 (T1H1) ficou sem sala, pois esta era a única que servia para ela, quando atualizarmos a velocidade, a T1H1 pode ter prioridade e pegar aquela sala primeiro, isso pode gerar uma T2H1 tendo que escolher uma sala pior ou ficando sem sala.

Atualização

Após isso, é calculado variáveis como pBest para cada partícula, fitness de cada indivíduo desta população e o gBest

Depois é calculado a nova velocidade que será atribuída a cada partícula.

Com base na nova velocidade é calculado os novos valores do vetor de partículas

Velocidade

$$V_{id}^{t+1} = K[\omega V_{id}^t + c_1 \times Rand1 \times (P_{id} - X_{id}) + c_2 \times Rand2 \times (P_{gd} - X_{id})]$$

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\kappa}}, \quad \varphi = c_1 + c_2, \quad \varphi > 4$$

V_{id} é o componente de velocidade atual.

X_{id} é o componente de posição do elemento

c_1 é o fator de aprendizado cognitivo (constante 1)

c_2 é o fator de aprendizado social (constante 2)

P_{id} é o componente de posição do Pbest (melhor posição individual)

P_{gd} é o componente de posição do Gbest (melhor posição global)

$Rand()$ é um número aleatório entre $[0, 1]$

ω = Inércia

K = Fator de Construção

Após o cálculo da velocidade de cada partícula é aplicado uma normalização para que os valores fiquem entre 0 e 1

Atualização

Perturbação: ocorre com uma probabilidade de 5% por partícula, Antes de atualizar a velocidade caso ocorra a perturbação é adicionado um ruído a velocidade de cada elemento da partícula.

Heurística de intercâmbio: 100 elementos têm seus valores trocados com outros elementos da partícula para a verificação da busca local.

Busca Local: Após cada movimento das partículas (após atualizar posição e velocidade), realiza-se uma busca local ao redor da nova posição, essa busca visa encontrar soluções melhores nas redondezas para atualizar pbest e possivelmente o gbest, utilizando o intercâmbio geramos uma nova solução vizinha então calcula-se o fitness, se for melhor a partícula se torna essa nova solução.

Informações passadas para o AG

- Listas de turmas
- Lista de salas
- Informações sobre blocos
- Parâmetros do PSO

Parâmetros:

- Tamanho do enxame
- Número gerações
- Inércia
- C1 e C2
-

Informações dos blocos

Guarda o nome do bloco e sua posição

Exemplos de Sala e Turma

Turma

```
'id': "turma_25",  
'num_alunos': 40,  
'bloco_preferencial': "PB",  
'tipo': "regular",  
'horarios': ["seg_09:30", "qua_11:30"]
```

Sala

```
'id': "sala_12",  
'bloco': "PB",  
'capacidade': 50,  
'tipo': "regular",  
'horarios_disponiveis': ["seg_07:30",  
"seg_09:30", "seg_11:30", "seg_13:30", ... ,  
"sab_23:30"]
```

Cálculo do Fitness

Foi utilizada esta fórmula no artigo:

$$Fitness = 10^{(T-C)} + (C/30)^{\left[\frac{C-T}{15}\right]} + \left| \frac{Bf - Bs}{5} \right| + Ph$$

Onde: T = Número de alunos da turma; C = Capacidade da sala; Bf = posição do bloco preferencial referente a turma; Bs = posição do bloco em que a turma foi alocada; Ph = incremento relacionado ao problema de turmas com alocação em salas distintas em diferentes horários da semana.

Avalia a qualidade de uma solução com base em:

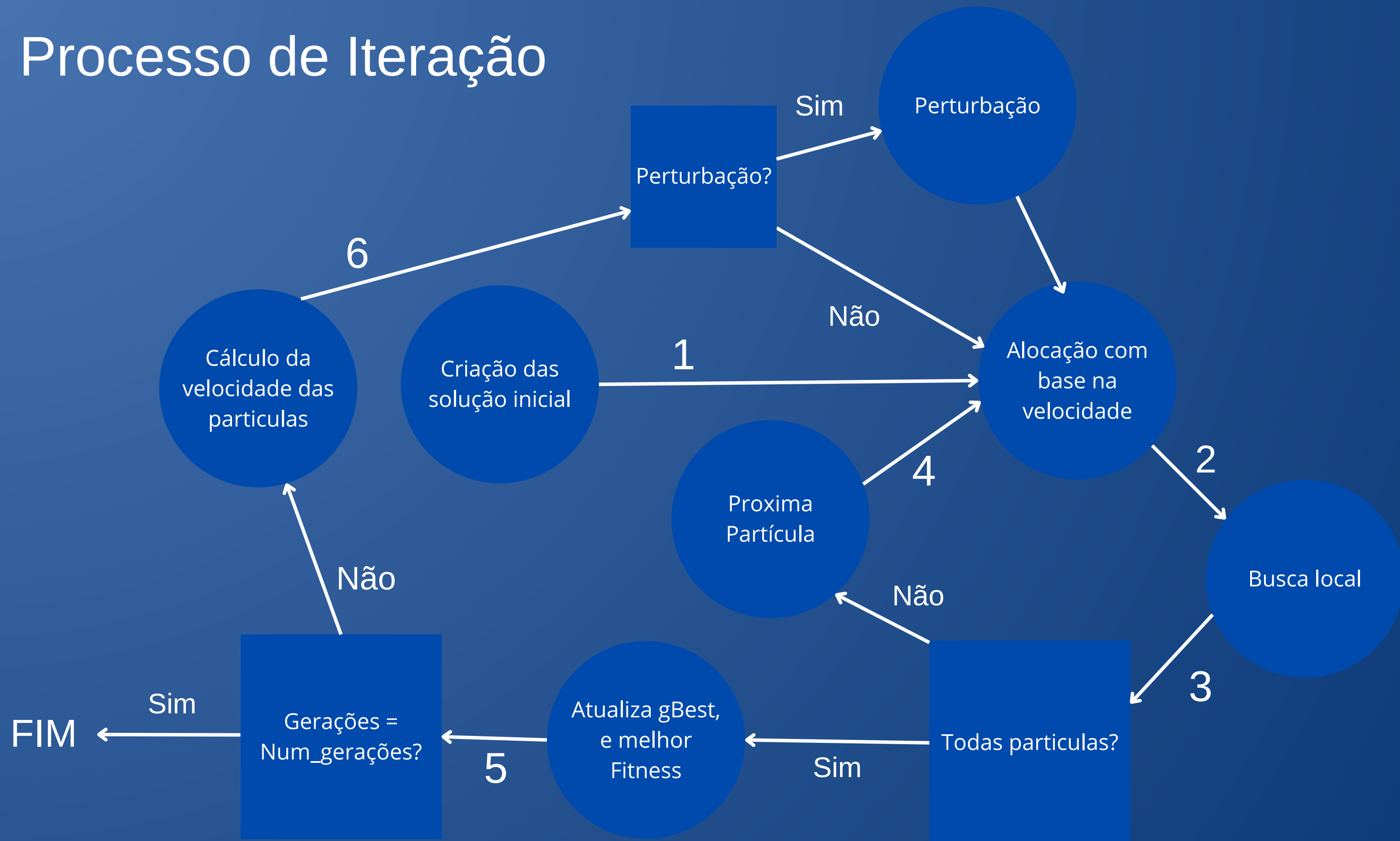
Capacidade: penaliza turmas alocadas em salas muito pequenas (muito penalizado) ou muito grandes

Preferência de bloco: penaliza alocações fora do bloco preferencial da turma

Consistência: penaliza turmas alocadas em salas diferentes em horários diferentes

Alocações inválidas: penaliza fortemente turmas não alocadas

Processo de Iteração



Testes de AG do Grupo

Tamanho da população: 25

Num de gerações: 150

Novos indivíduos por geração: 6

Períodos sem evolução: 6

=== RESULTADOS FINAIS ===

Melhor fitness: 202302.26

Problemas de capacidade: 95

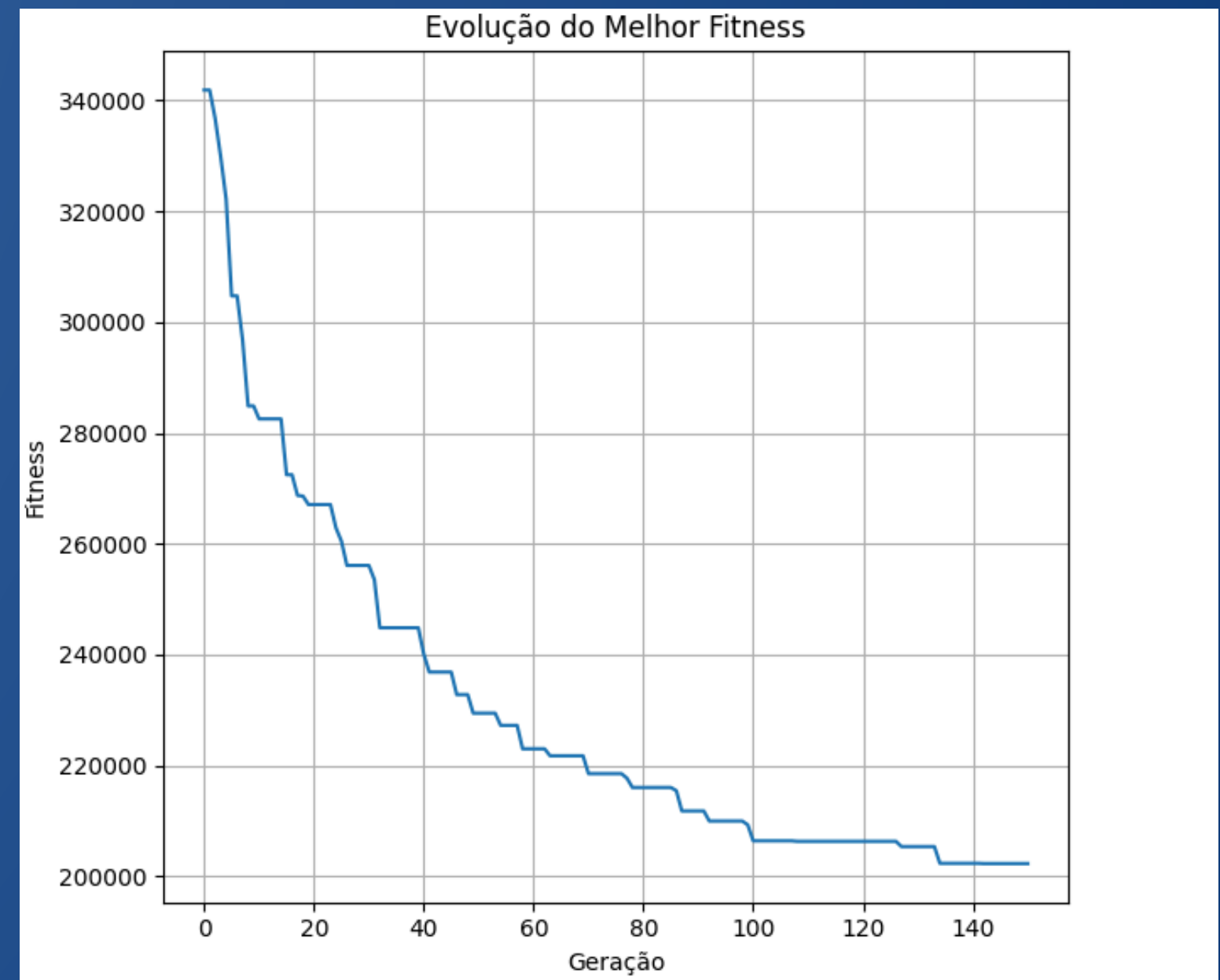
Problemas de bloco preferencial: 1467

Taxa de evolução final: 84.10%

Tempo de execução: 11.74 minutos

Resultado deles: TE - 93%

Fit - 83118

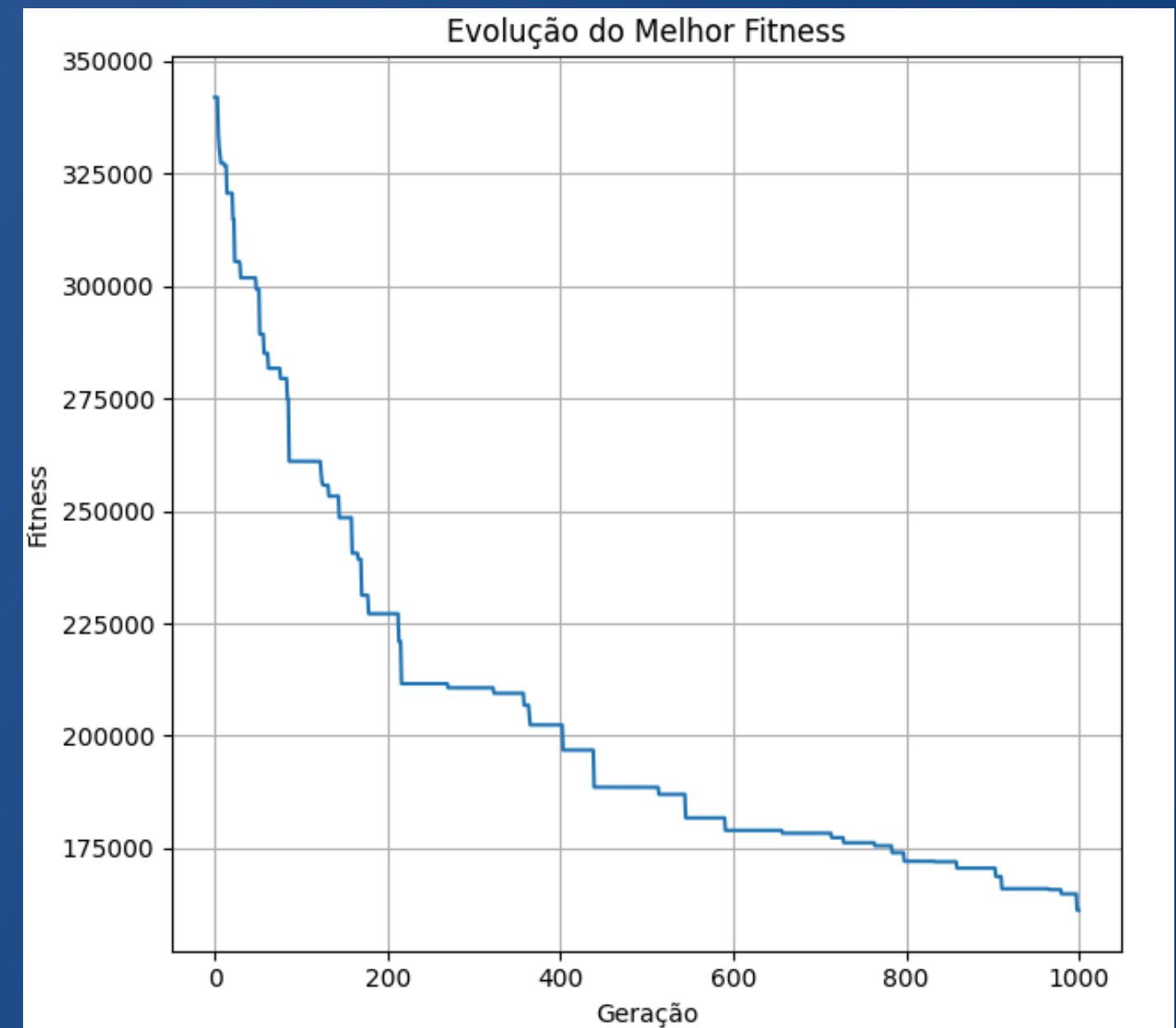


Testes do Grupo sem as Modificações

Tamanho da população: 100
Num de gerações: 1000
Novos individuos por geração: 3
Períodos sem evolução: 6

```
=== RESULTADOS FINAIS ===  
Melhor fitness: 161276.12  
Problemas de capacidade: 65  
Problemas de bloco preferencial: 1466  
Taxa de evolução final: 96.31%  
Tempo de execução: 38.92 minutos
```

Com nossos parâmetros



Testes do Grupo

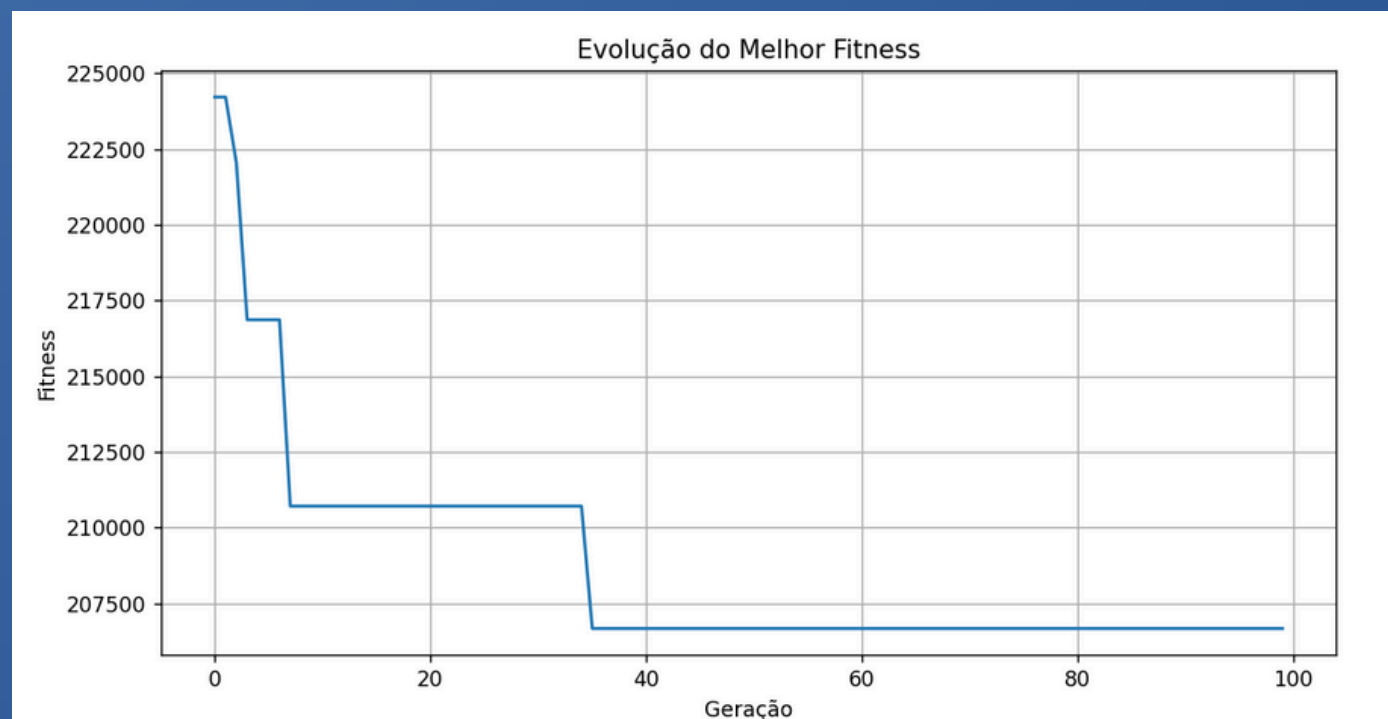
Tamanho da população: 5

Num de gerações: 100

C1: 2.05 , C2: 2.05

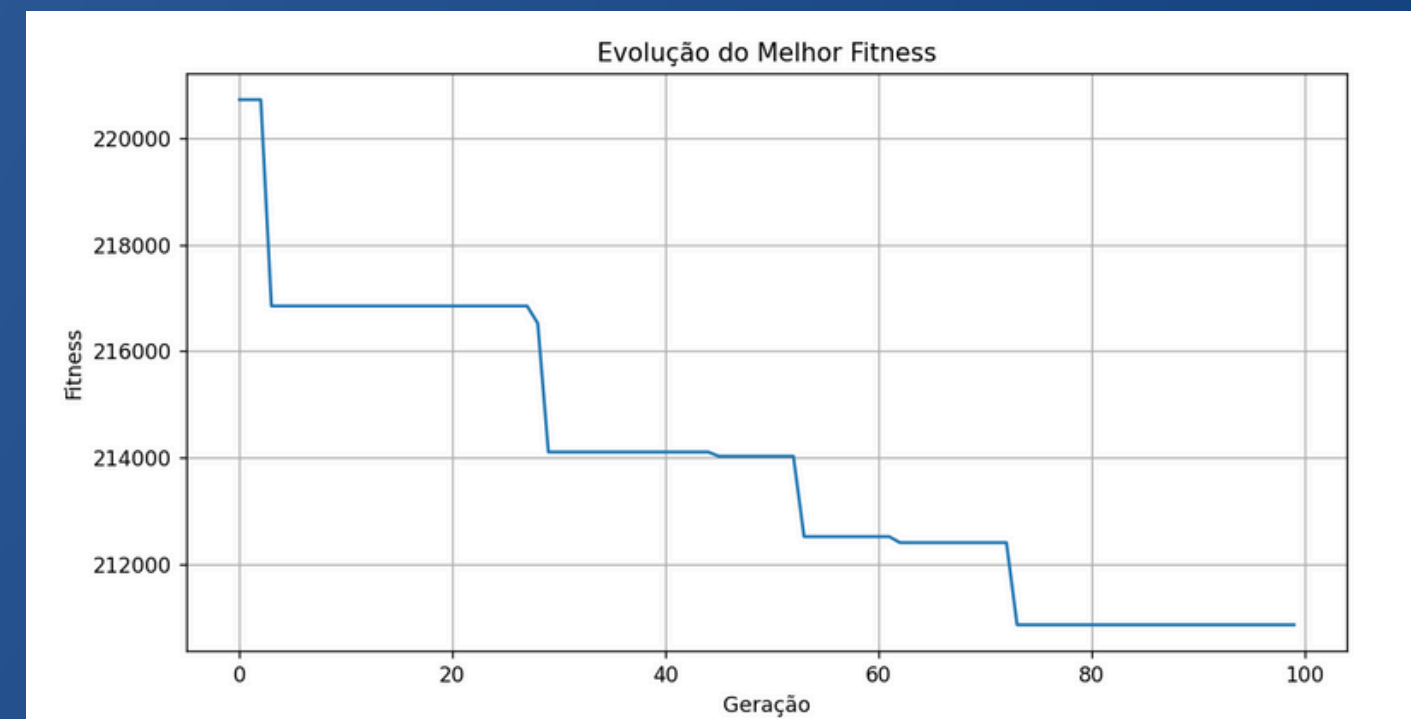
Inércia: 0.8

PSOLS



```
=== RESULTADOS FINAIS ===  
Melhor fitness: 210864.42  
Tempo de execucao: 6.99 minutos  
- Problemas de capacidade: 0  
- Problemas de bloco: 910
```

PSO



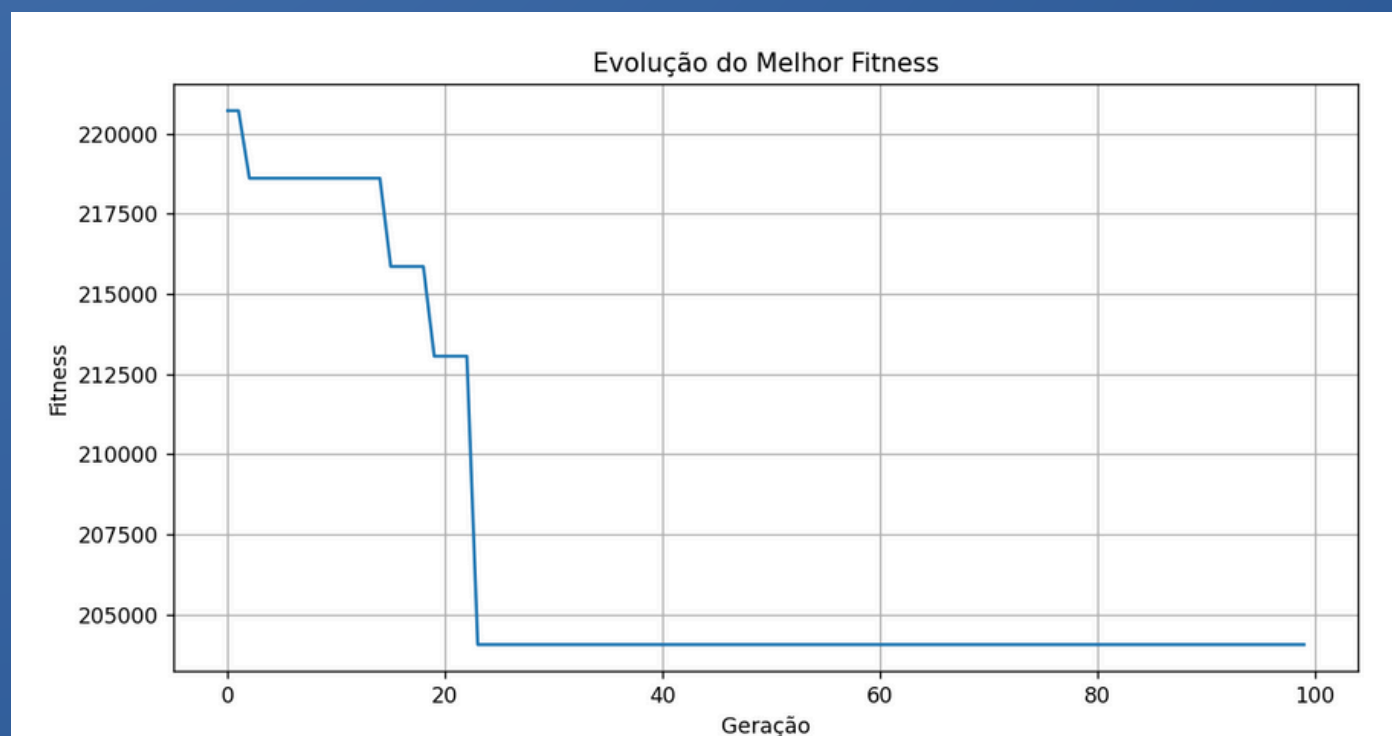
```
=== RESULTADOS FINAIS ===  
Melhor fitness: 206674.41  
Tempo de execucao: 3.50 minutos  
- Problemas de capacidade: 0  
- Problemas de bloco: 904
```

Melhor resultado do AG implementado no trabalho 2 anterior: 161276

Testes do Grupo

Tamanho da população: 5
Num de gerações: 100
C1: 2.05 , C2: 2.05
Inércia: 0.9

PSOLS



=== RESULTADOS FINAIS ===

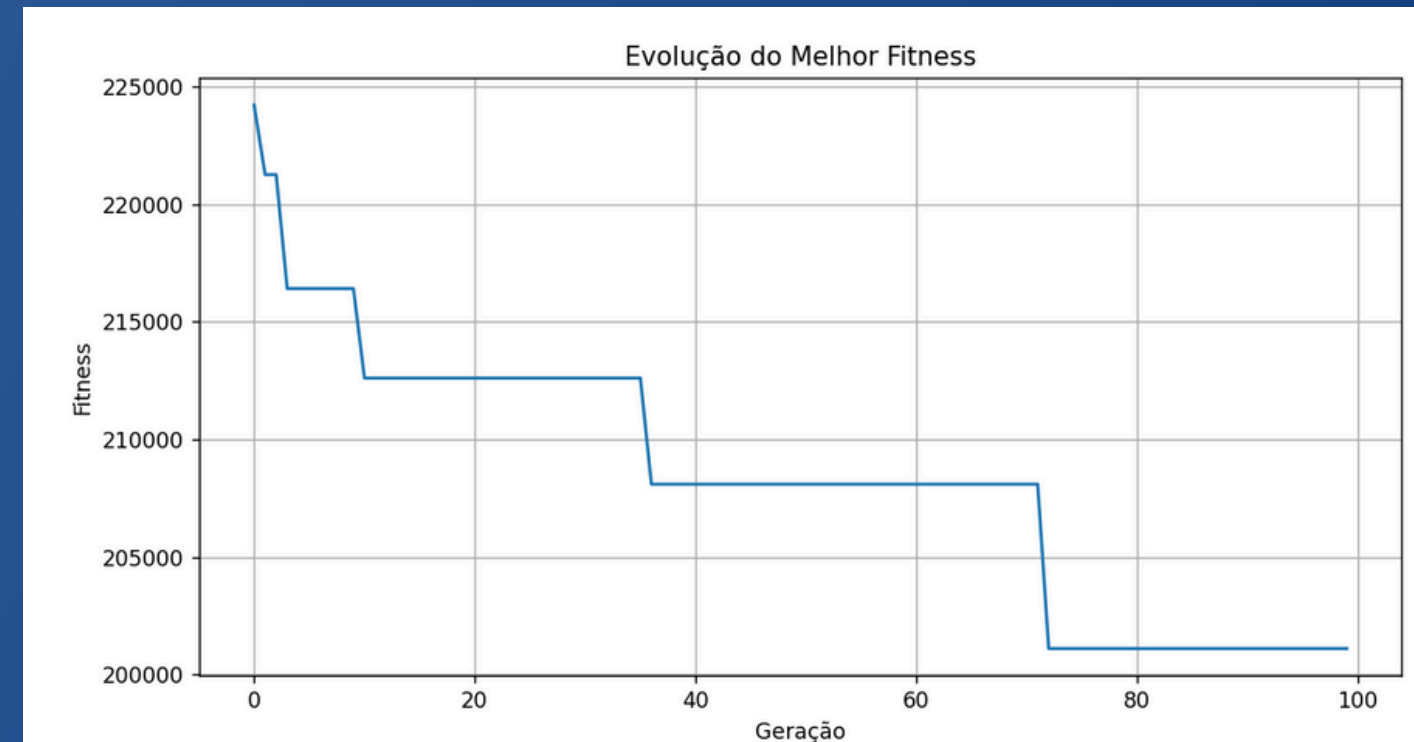
Melhor fitness: 204057.04

Tempo de execucao: 6.95 minutos

- Problemas de capacidade: 0

- Problemas de bloco: 899

PSO



=== RESULTADOS FINAIS ===

Melhor fitness: 201114.67

Tempo de execucao: 3.37 minutos

- Problemas de capacidade: 0

- Problemas de bloco: 917

Melhor resultado do AG implementado no trabalho 2 anterior: 161276

Testes do Grupo

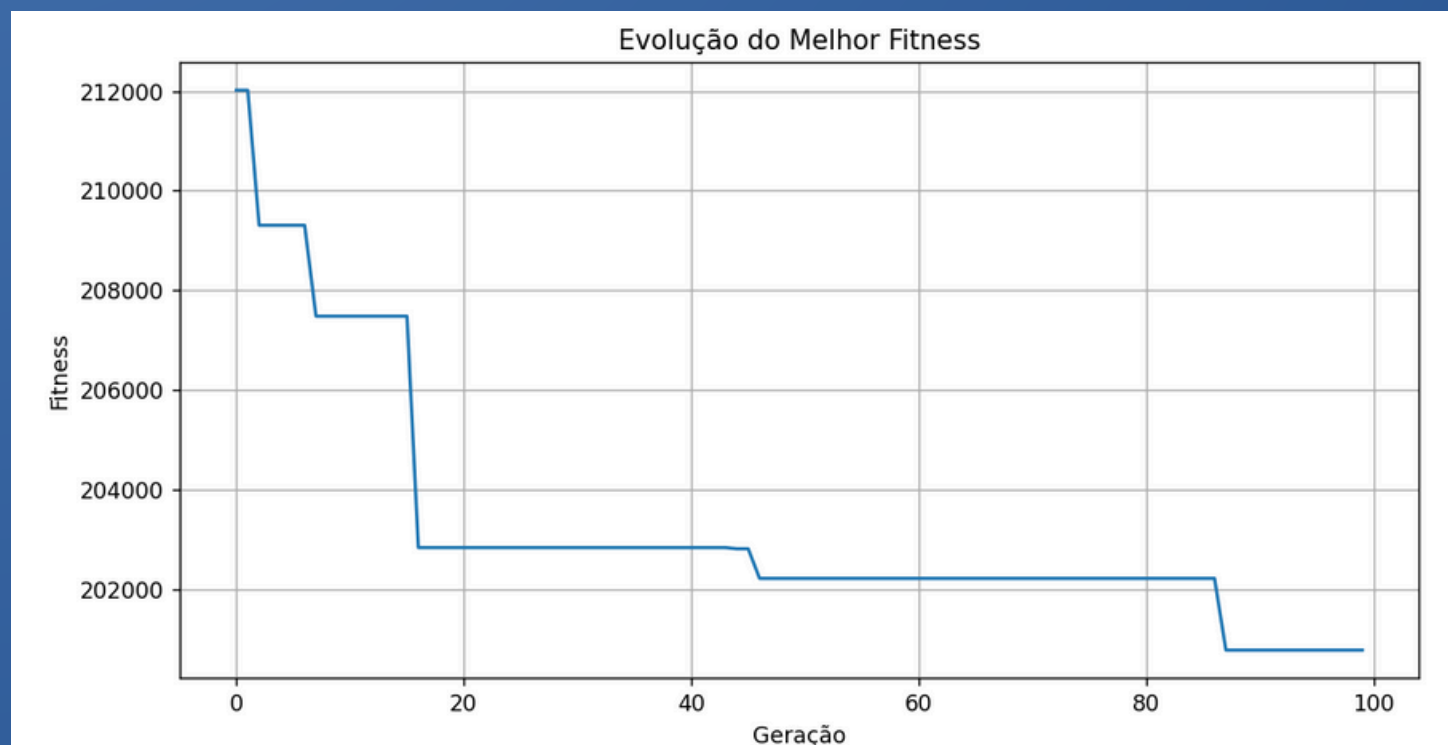
Tamanho da população: 10

Num de gerações: 100

C1: 3 , C2: 3

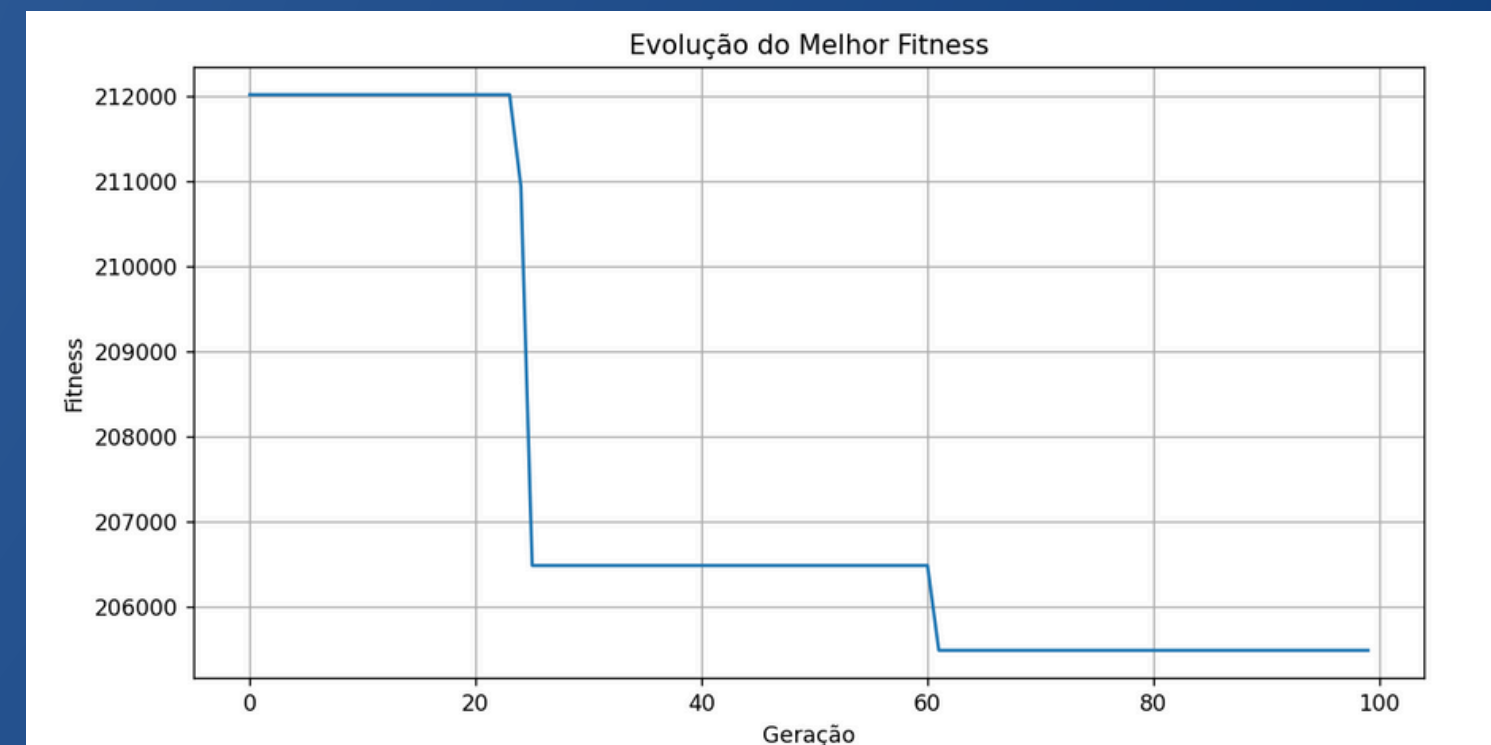
Inércia: 0.9

PSOLS



```
=== RESULTADOS FINAIS ===  
Melhor fitness: 200779.55  
Tempo de execucao: 13.27 minutos  
- Problemas de capacidade: 0  
- Problemas de bloco: 888
```

PSO



```
=== RESULTADOS FINAIS ===  
Melhor fitness: 205494.24  
Tempo de execucao: 6.62 minutos  
- Problemas de capacidade: 0  
- Problemas de bloco: 889
```

Melhor resultado do AG implementado no trabalho 2 anterior: 161276

Testes do Grupo

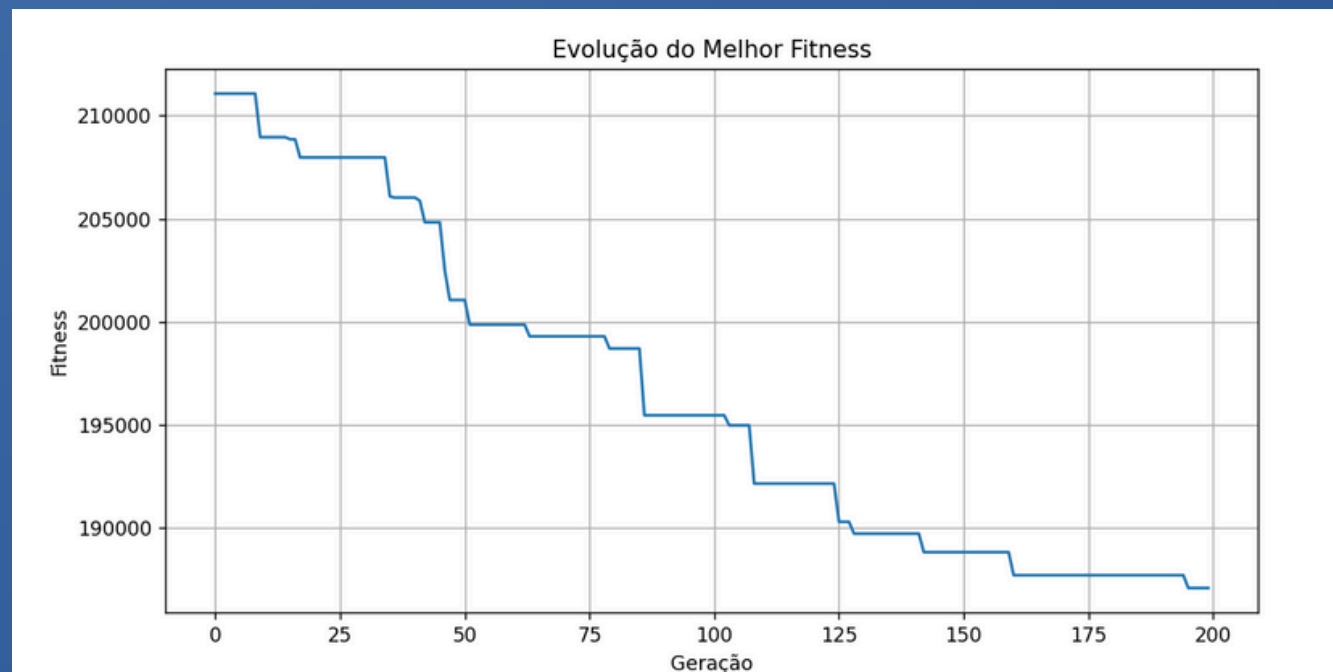
Tamanho da população: 30

Num de gerações: 200

C1: 3 , C2: 3

Inércia: 0.9

PSOLS



=== RESULTADOS FINAIS ===

Melhor fitness: 187098.44

Tempo de execucao: 81.67 minutos

- Problemas de capacidade: 0

- Problemas de bloco: 913

PSO



=== RESULTADOS FINAIS ===

Melhor fitness: 199973.88

Tempo de execucao: 41.15 minutos

- Problemas de capacidade: 0

- Problemas de bloco: 915

Melhor resultado do AG implementado no trabalho 2 anterior: 161276