

Disciplina : IA II

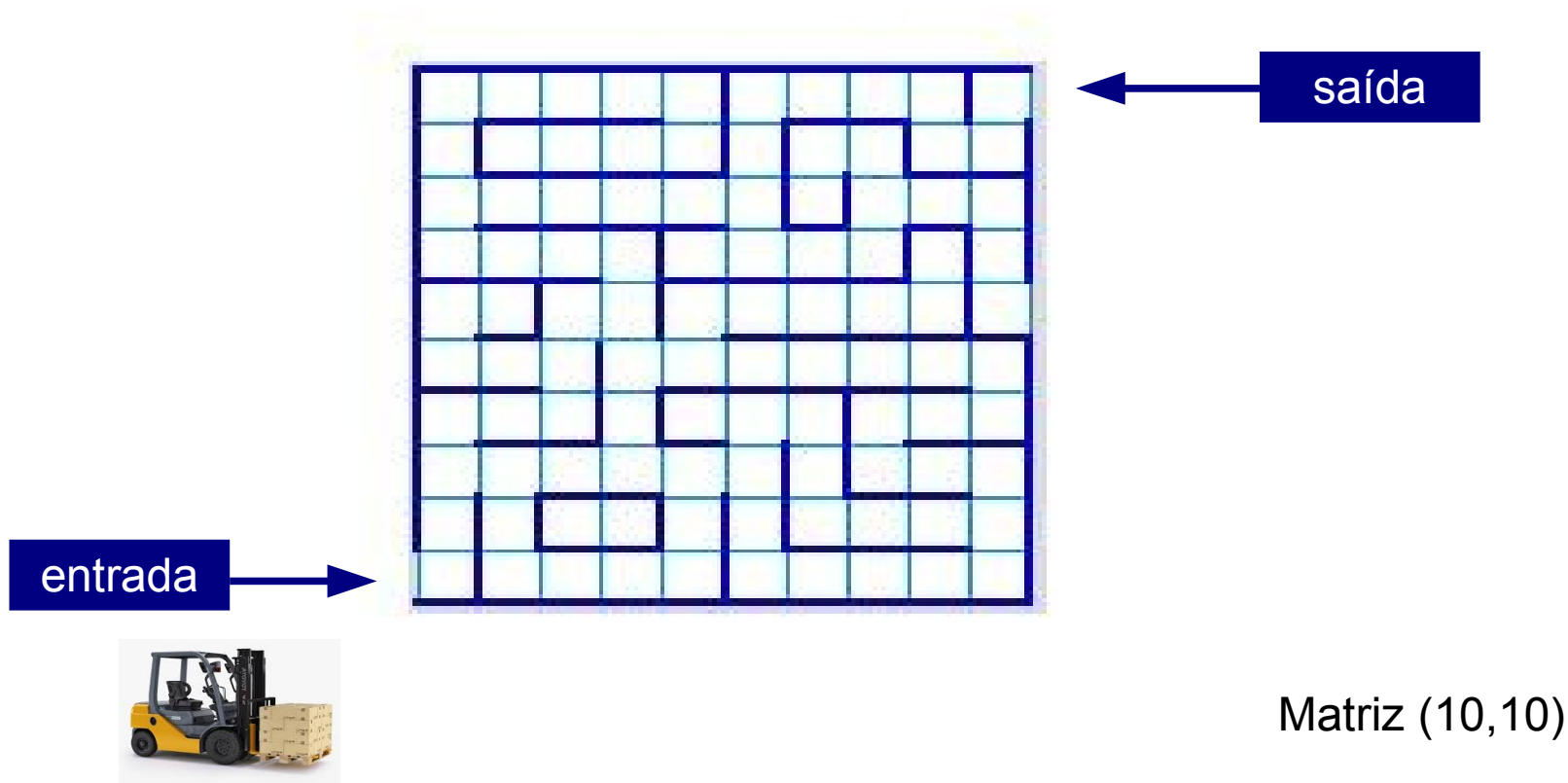
Projeto I – Empilhadeira Autoguiada

Busca de caminho via Algoritmo Genético

Prof^a Carine Webber



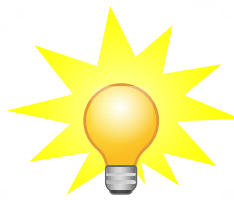
Caminho em um Labirinto



Características do Problema

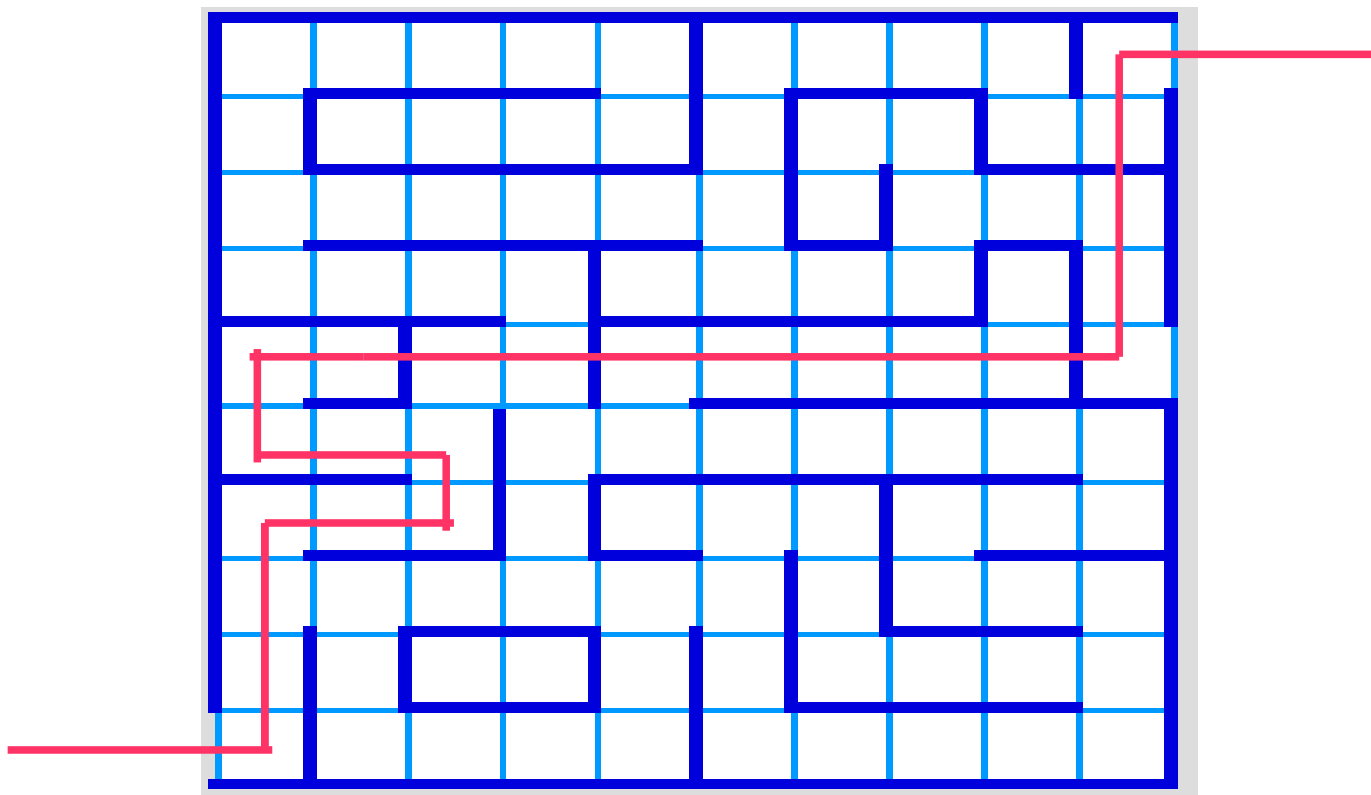
- Todo caminho começa na entrada e termina na saída do labirinto.
- Cada caminho é um indivíduo da população de soluções.
- Um caminho aleatório tem poucas chances de conduzir corretamente até a saída do labirinto.
- Os caminhos evoluem a partir de sucessivas gerações, se recombinao geneticamente e sofrendo mutaões.
- A evoluão continua até que uma soluão seja produzida.

Modelagem das soluções



- Utilizar 2 bits para representar direções:
 - ♦ 00 – leste
 - ♦ 01 – norte
 - ♦ 10 – oeste
 - ♦ 11 – sul
- Para uma matriz (10x10) definir um valor limite de bits para representar o caminho:
`0001010011 = leste,norte,norte,leste,sul`
- Avaliação: Considerar se o caminho é cortado por paredes, o total de células percorridas e a distância entre a posição alcançada no labirinto e a sua saída.

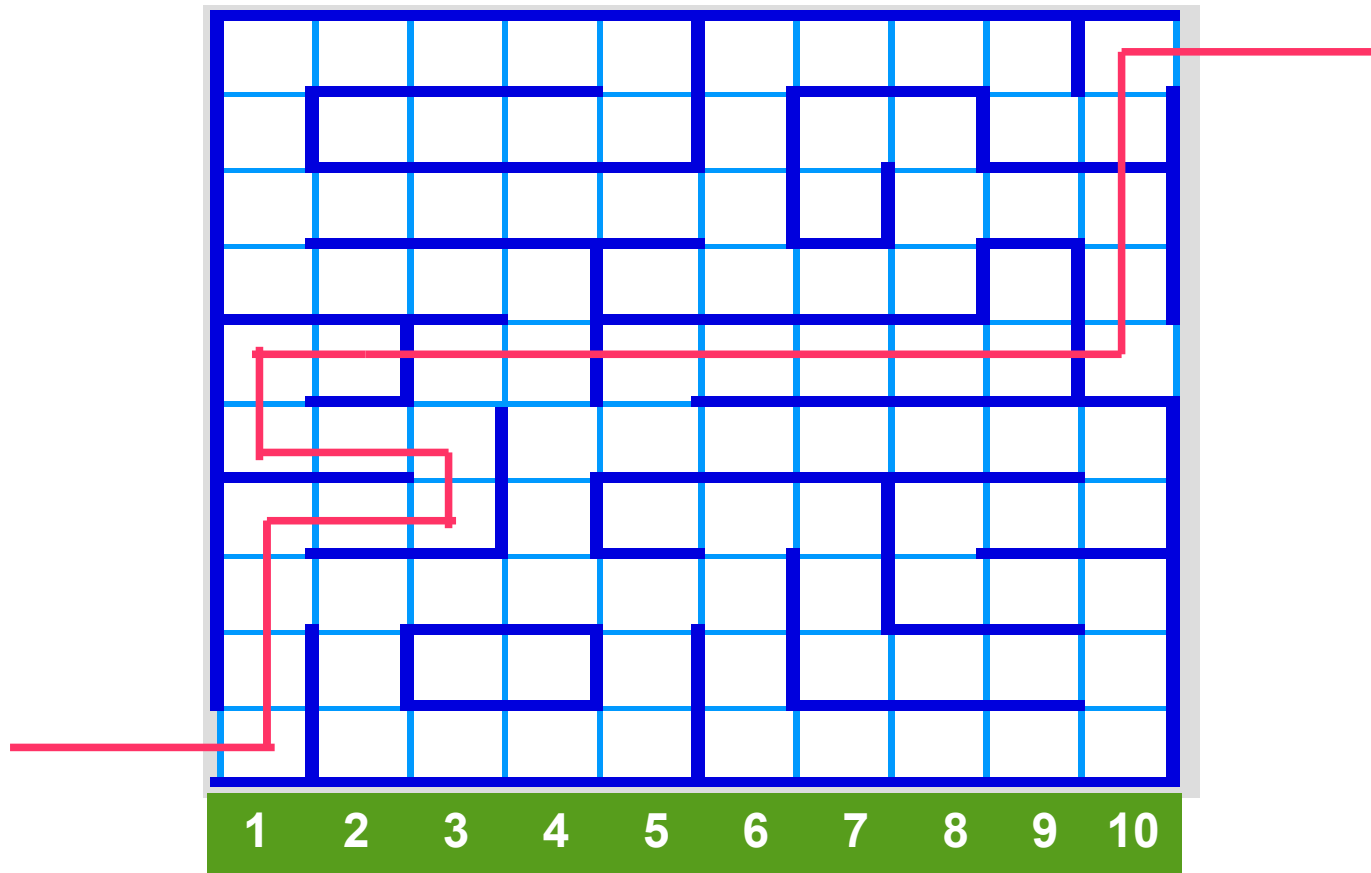
Exemplo de Caminho (solução não satisfatória)



000101011010011010010100000000000000000000101010100

Exemplo de avaliação do caminho: somar 1 ponto por célula ocupada, somar 10 pontos por parede atravessada, somar a distância entre a última célula do caminho e a saída, ... (você pode definir outras formas de avaliação)

Exemplo de caminho



Avaliação 1 da solução = passos para chegar à saída (posição (11,10)) = 23

Avaliação 2 da solução = punição por atravessar paredes (10) = $4 * 10 = 40$

Descrição do Problema



- Escreva um programa que a partir de uma população inicial de caminhos aleatoriamente gerados em um labirinto, encontre o melhor caminho fazendo uso de operadores genéticos de *crossing-over*, mutação e elitismo.
- O programa deve apresentar a melhor solução encontrada, seu valor de *fitness* e o número de gerações que foram necessárias para produzi-la.

Implementação



- O programa deve ser implementado como um sistema de simulação, onde as taxas genéticas devem ser parâmetros de configuração disponíveis na interface do sistema.
- O programa deve ser totalmente implementado em uma linguagem de programação.
- O trabalho pode ser desenvolvido individualmente ou em duplas.
- Os trabalhos serão apresentados na aula do dia 08/09 através de slides e execução do código.
- Trabalhos em atraso sofrerão redução na nota.



Cronograma

- 22/08 – fase 1 modelagem : definição de interface, estrutura de dados, operadores genéticos e parâmetros.
- 29/08 – fase 2 implementação e testes. A implementação pode partir do código previamente fornecido. Preparação de slides.
- 05/09 – entrega e apresentação dos trabalhos oralmente para a turma.

Avaliação



- A avaliação seguirá os seguintes critérios:

a) Correção do código: 5 pontos

- o código corresponde a uma implementação de AG contendo crossover e mutação de uma população?
- o código foi desenvolvido pelo grupo sem cópias?
- o código é legível e as classes compreensíveis?

b) Interface: 2 pontos

- as entradas e parametrizações estão disponíveis na interface?
- as saídas são compreensíveis?

c) Solução: 2 pontos

- a implementação encontra uma solução viável para o problema?

d) Apresentação: 1 ponto