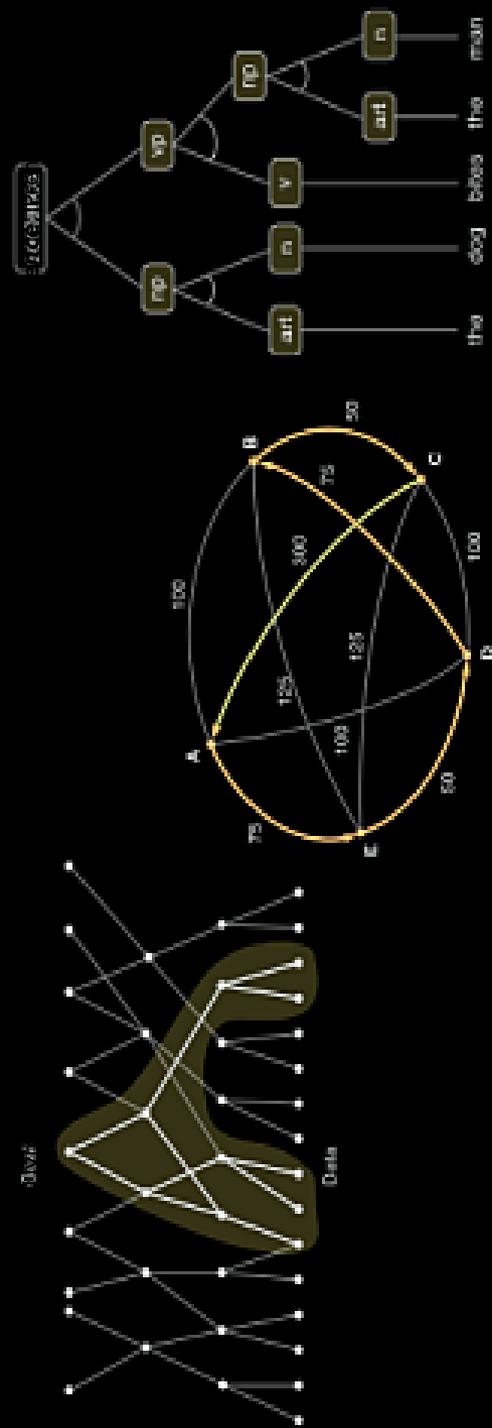


**Universidade Federal de Campina Grande**  
**Departamento de Sistemas e Computação**  
**Curso de Pós-Graduação em Ciência da Computação**

# Inteligência Artificial

## Resolução de Problemas (Parte I)

**Prof.<sup>a</sup> Joseana Macêdo Fachine Régis de Araújo**  
**[joseana@computacao.ufcg.edu.br](mailto:joseana@computacao.ufcg.edu.br)**

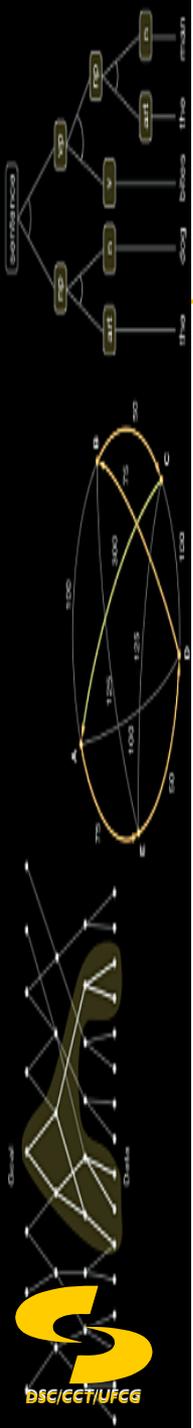


# Resolução de Problemas

---

## Tópicos

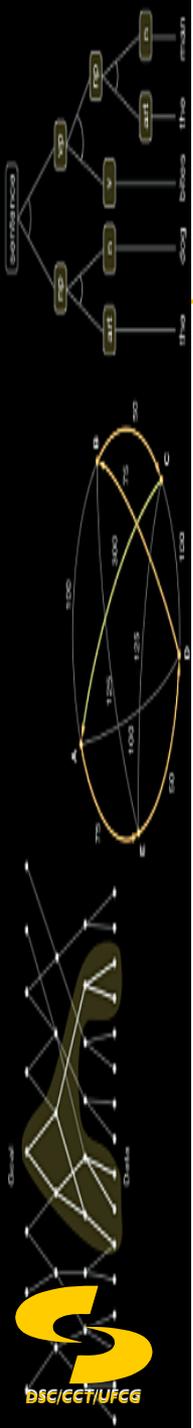
- Conceitos Básicos
- Espaço de Estados



# Resolução de Problemas

---

- Dedicar-se ao estudo e elaboração de algoritmos, capazes de resolver, por exemplo, problemas considerados intratáveis do ponto de vista da computação convencional.



# Resolução de Problemas

- ❑ Primeiros problemas por computador: prova automática de teoremas e jogos.
- ❑ Capacidade de cálculo e memória dos computadores: insuficientes perante o enorme número de caminhos de solução.
- ❑ **Exemplo:** jogo de xadrez
- ❑ **Um dos objetivos de IA:** resolver problemas que o homem não sabe resolver facilmente ou num tempo razoável, desde que sejam completamente formalizados.

# Exemplos de Problemas

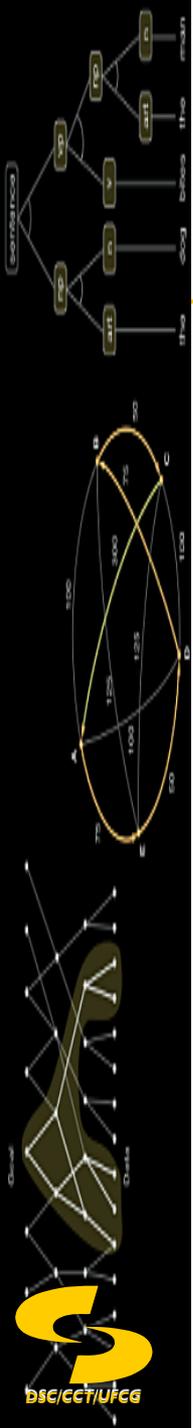
- ❑ O quebra-cabeças 3x3
- ❑ O Caixeiro Viajante
- ❑ Cálculo Integral Formal
- ❑ Empilhamento de blocos: a partir de uma configuração de blocos iniciais, qual a seqüência de movimentos para se chegar a uma configuração final?
- ❑ As Oito Rainhas
- ❑ As Torres de Hanói

# Formulação de Problemas

---

## Elementos:

- ❑ Estado Inicial
- ❑ Função Sucessor
- ❑ Teste de Objetivo
- ❑ Custo de Caminho



# Problemas e soluções bem definidos

## Quatro componentes para definir um problema:

1. O **estado inicial** em que o agente começa.
2. Uma descrição das **ações** possíveis que estão disponíveis para o agente.
  - Formulação mais comum: uso de uma **função sucessor**.
  - Estado inicial e função sucessor: definem o **espaço de estados** do problema.
  - **Caminho** no espaço de estados – seqüência de estados conectados por uma seqüência de ações.
3. O **teste de objetivo** – determina se um dado estado é um estado objetivo.
4. Função de **custo de caminho** – atribui um custo numérico a cada caminho.

# Exemplos de Problemas

## □ Miniproblemas

- Utilizado para ilustrar ou exercitar diversos métodos de resolução de problemas.
- Pode ter uma descrição concisa e exata – pode ser utilizado com facilidade por diferentes sistemas de busca, com a finalidade de comparar o desempenho de algoritmos.

## □ Problemas do mundo real

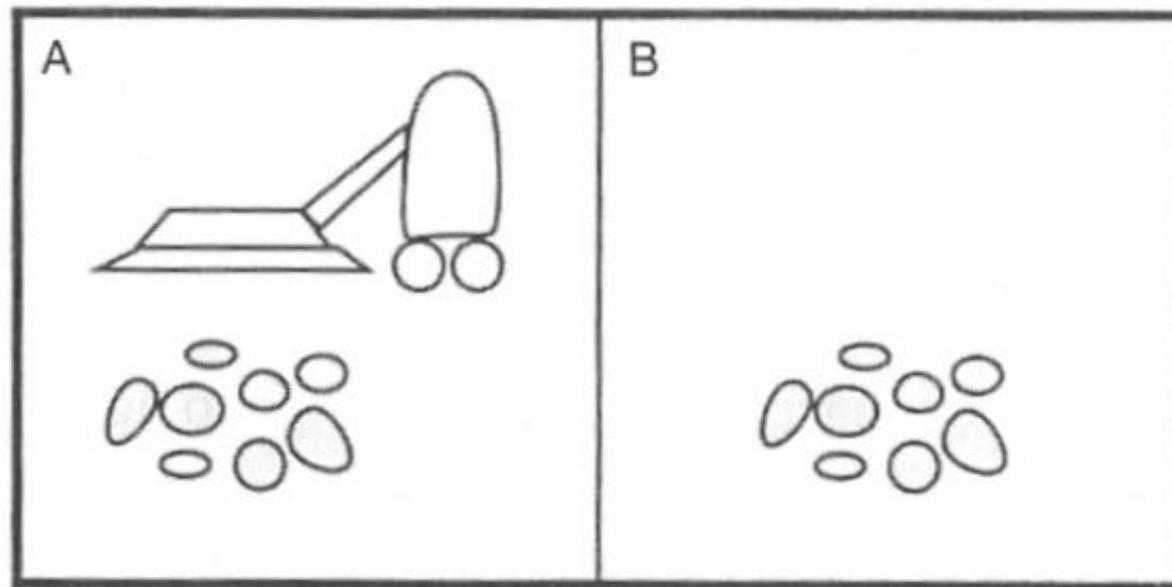
- Tendem a não apresentar uma única descrição consensual, mas é possível fornecer uma idéia geral de suas formulações.

# Exemplos de Problemas

- **Exemplos: Miniproblemas**
  - Mundo do Aspirador de Pó
  - Problema do Quebra-cabeça de 8 Peças
  - Problema das 8 Rainhas
  
- **Exemplos: Problemas do mundo real**
  - Problema de Roteamento
  - Problema de Viagens Aéreas
  - Problema de *Tour*
  - Problema do Caixeiro Viajante
  - Problema de Layout de VLSI
  - Problema de Navegação de Robôs
  - Problema da Seqüência Automática de Montagem
  - Problema de Pesquisas na Internet

# Miniproblemas

- Exemplo 1: Mundo do aspirador de pó com apenas 2 locais.



# Miniproblemas

## Problema do Mundo do Aspirador de Pó - Formulação

### □ Estados

- O agente está em uma entre duas posições, cada uma das quais pode conter sujeira ou não.
- Há  $2 \times 2^2 = 8$  estados do mundo possíveis.

### □ Estado inicial

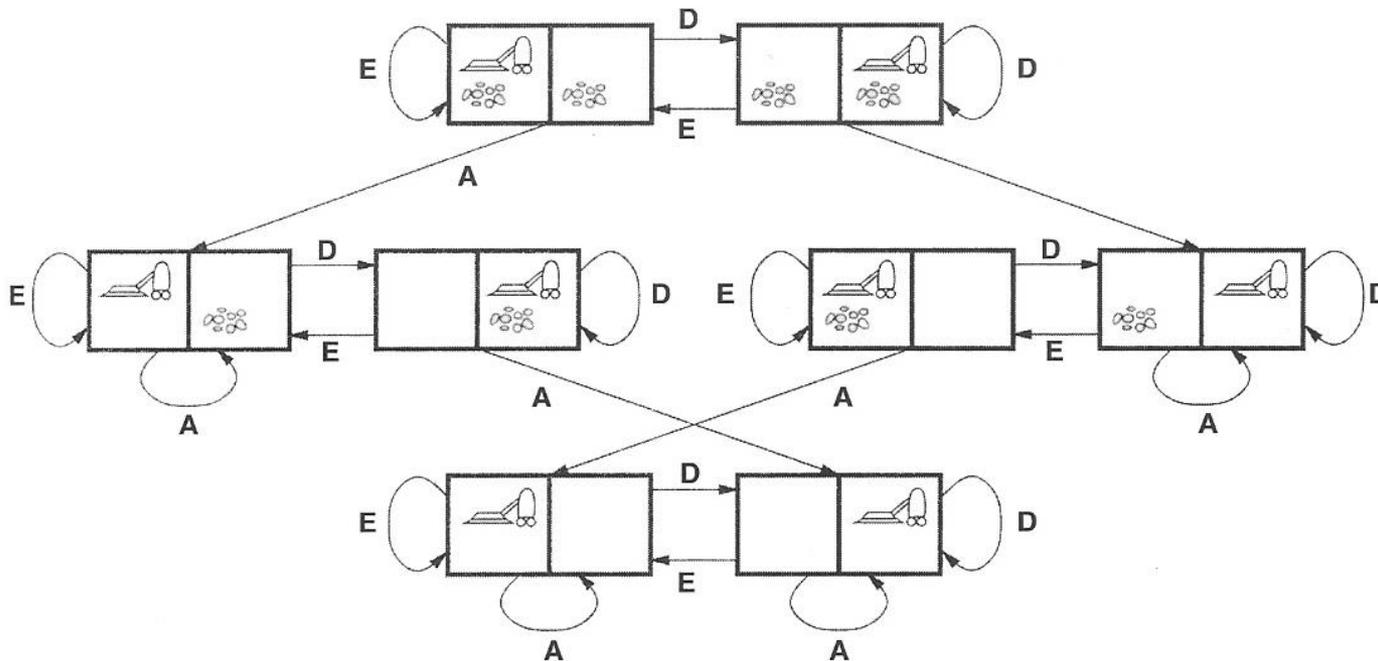
- Qualquer estado pode ser designado como estado inicial.

### □ Função Sucessor

- Gera os estados válidos que resultam da tentativa de executar as três ações (*Esquerda*, *Direita* e *Aspirar*).

# Miniproblemas

Espaço de estados para o mundo do aspirador de pó.



Os arcos denotam ações: E = Esquerda, D = Direita, A = Aspirar

# Miniproblemas

## Problema do Mundo do Aspirador de Pó - Formulação

- **Teste de objetivo**
  - Verifica se todos os quadrados estão limpos.
- **Custo de caminho**
  - Cada passo custa 1, e assim o custo do caminho é o número de passos do caminho.
- Esse miniproblema tem posições discretas, sujeira discreta, limpeza confiável e nunca é desorganizado depois de limpo.
- Ambiente com  $n$  posições:  $n \times 2^n$  estados

# Miniproblemas

**Exemplo 2:** Uma instância típica do quebra-cabeça de 8 peças

7	2	4
5		6
8	3	1

Estado inicial

	1	2
3	4	5
6	7	8

Estado objetivo

# Miniproblemas

## Problema do Quebra-cabeça de 8 Peças - Formulação

### □ Estados

- Uma descrição de estado especifica a posição de cada uma das oito peças e do espaço vazio em um dos nove quadrados.

### □ Estado inicial

- Qualquer estado pode ser designado como estado inicial.

### □ Função Sucessor

- Gera os estados válidos que resultam da tentativa de executar as três ações (o espaço vazio se desloca para a *Esquerda*, *Direita*, *Acima* ou *Abaixo*).

# Miniproblemas

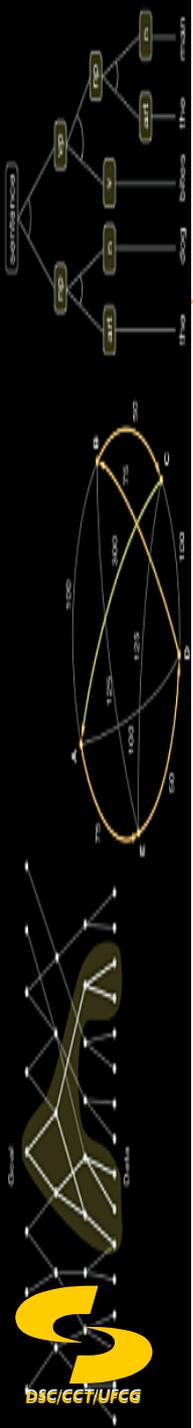
## Problema do Quebra-cabeça de 8 peças - Formulação

### □ Teste de objetivo

- Verifica se o estado corresponde à configuração de objetivo mostrada na figura (São possíveis outras configurações de objetivos)

### □ Custo de caminho

- Cada passo custa 1, e assim o custo do caminho é o número de passos do caminho.



# Miniproblemas

## Problema do Quebra-cabeça de 8 Peças - Formulação

### □ Abstrações incluídas

- As ações são reduzidas a seus estados iniciais e finais, ignorando-se as posições intermediárias por onde o bloco está deslizando.
- Foram abstraídas ações como sacudir o tabuleiro quando as peças ficam presas ou extrair as peças com uma faca e colocá-las de volta no tabuleiro.

# Miniproblemas

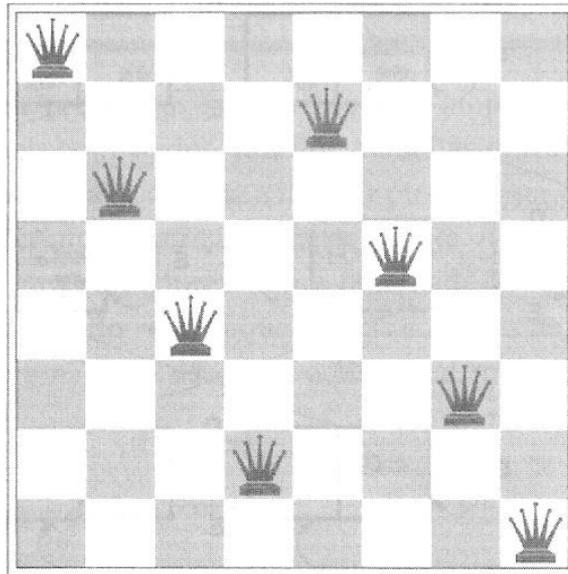
## Problema do Quebra-cabeça de 8 Peças - Formulação

- Pertence à família de **quebra-cabeças de blocos deslizantes** – usados com frequência como problemas de teste para novos algoritmos de busca em IA.
- **Número de estados acessíveis**
  - Quebra-cabeça de 8 peças:  $9!/2 = 181.440$
  - Quebra-cabeça de 15 peças (tabuleiro de 4 x 4): aproximadamente 1,3 trilhão (instâncias aleatórias podem ser resolvidas de forma ótima em alguns *ms* pelos melhores algoritmos de busca).
  - Quebra-cabeça de 24 peças (tabuleiro 5 x 5): cerca de 10 estados (difíceis de resolver de forma ótima com as máquinas e os algoritmos atuais)

# Miniproblemas

## Exemplo 3: Problema das 8 Rainhas

- **Objetivo:** posicionar 8 rainhas em um tabuleiro de xadrez de tal forma que nenhuma rainha ataque qualquer outra (uma rainha ataca qualquer peça situada na mesma linha, coluna ou diagonal).



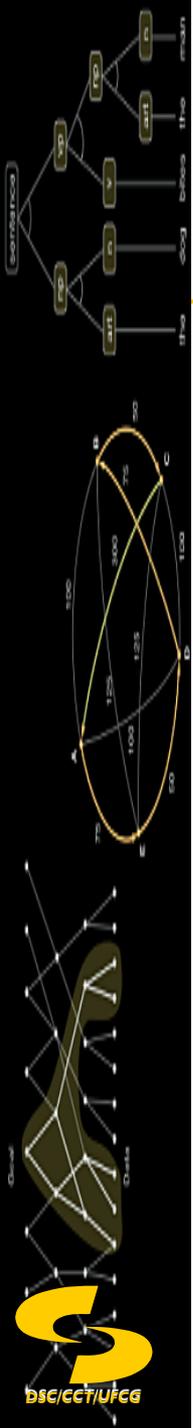
Uma quase solução para o problema das 8 rainhas.

# Miniproblemas

---

## Problema das 8 Rainhas - Formulação

- Formulação incremental
- Formulação de estados completos



# Miniproblemas

## Problema das 8 Rainhas – Formulação Incremental

- **Estados**
  - Qualquer disposição de 0 a 8 rainhas no tabuleiro é um estado.
- **Estado inicial**
  - Nenhuma rainha no tabuleiro.
- **Função Sucessor**
  - Colocar uma rainha em qualquer quadrado vazio.
- **Teste de objetivo**
  - 8 rainhas estão no tabuleiro e nenhuma é atacada.
- **Seqüências possíveis de investigação:  $64 \times 63 \times \dots \times 57 \approx 10^{14}$**

# Miniproblemas

## Problema das 8 Rainhas – Formulação Incremental Melhorada

- ❑ Proibir a colocação de uma rainha em qualquer quadrado que já estiver sob ataque.
- ❑ **Estados**
  - Os estados são disposições de  $n$  rainhas ( $0 \leq n \leq 8$ ), uma por coluna nas  $n$  colunas mais à esquerda, sem que nenhuma rainha ataque outra.
- ❑ **Função Sucessor**
  - Adicione uma rainha a qualquer quadrado na coluna vazia mais à esquerda de tal modo que ela não seja atacada por qualquer outra rainha.
- ❑ Espaço de estados: 2.057

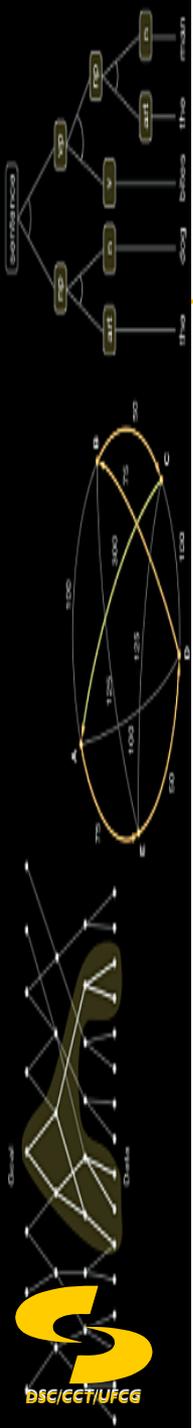
# Problemas do mundo real

---

## Problema de Roteamento: Formulação

### □ Exemplos de aplicação

- Roteamento em redes de computadores
- Planejamento de operações militares
- Sistemas de planejamento de viagens aéreas



# Problemas do mundo real

## Problema de Viagens Aéreas – Formulação

- **Estados**
  - Cada um é representado por uma posição (p.ex.: um aeroporto) e pela hora atual.
- **Estado inicial**
  - É especificado pelo problema.
- **Função Sucessor**
  - Retorna os estados resultantes de tomar qualquer voo programado que parte depois da hora atual somada ao tempo de trânsito no aeroporto, desde o aeroporto atual até outro.
- **Teste de objetivo**
  - Estamos no destino após algum tempo especificado?

# Problemas do mundo real

## Problema de Viagens Aéreas – Formulação

### □ Custo de caminho

- Depende do custo monetário, do tempo de espera, do tempo de voo, dos procedimentos alfandegários, da hora do dia, ...

- Um sistema realmente bom deve incluir planos de contingência. **Exemplo:** reservas substitutas em vôos alternativos.

# Problemas do mundo real

## Problemas de *Tour* – Formulação

- Estreitamente relacionados aos problemas de roteamento
- **Diferença:** cada estado deve incluir não apenas a posição atual, mas também o *conjunto de cidades que o agente visitou*.

# Problemas do mundo real

## Problema do Caixeiro-Viajante (PCV) – Formulação

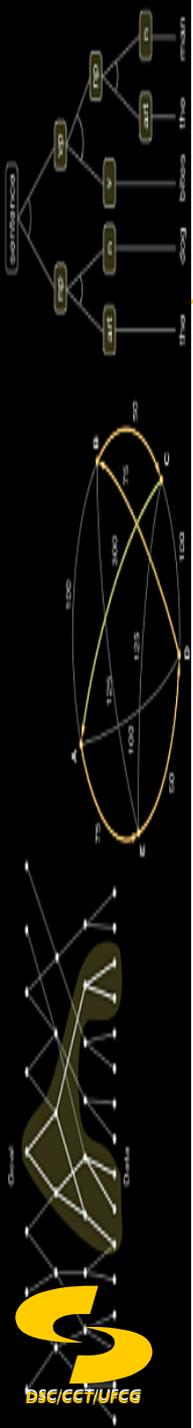
- É um **problema de *tour*** em que cada cidade deve ser visitada exatamente uma vez.
- **Objetivo:** encontrar o percurso mais curto.
- **Outras aplicações:** planejamento do movimento de máquinas automáticas para perfuração de placas de circuitos e de máquinas industriais em fábricas.

# Problemas do mundo real

---

## Problema de Layout de VLSI – Formulação

- Exige o posicionamento de milhões de componentes e conexões em um chip para minimizar a área, os retardos de circuitos e as capacitâncias de fuga e maximizar o rendimento industrial.
- Divisão usual:
  - Layout de células
  - Roteamento de canais



# Problemas do mundo real

## Problema da Navegação de Robôs – Formulação

- Generalização do problema de roteamento.
- **Característica:** um robô pode se mover em um espaço contínuo com (em princípio) um conjunto infinito de ações e estados possíveis.
- Robô com movimento circular sobre uma superfície plana: espaço bidimensional.
- Robô com braços e pernas ou rodas – espaço de busca com várias dimensões.

# Problemas do mundo real

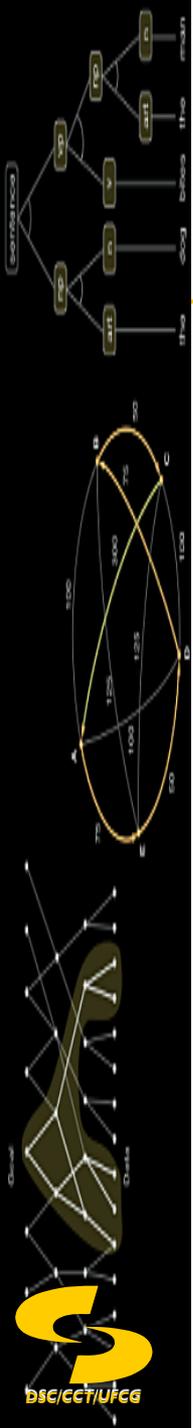
## Problema da Seqüência Automática de Montagem – Formulação

- ❑ **Objetivo:** encontrar uma ordem na qual devem ser montadas as peças de algum objeto.
- ❑ Outro problema de montagem: **projeto de proteínas**
  - **Objetivo:** encontrar uma seqüência de aminoácidos que serão incorporados em uma proteína tridimensional com as propriedades adequadas para curar alguma doença.

# Problemas do mundo real

## Problema de Pesquisas na Internet – Formulação

- ❑ **Objetivo:** procurar respostas para perguntas, informações inter-relacionadas ou oportunidades de compras



# Resolução de Problemas

---

## Próximos passos:

- Em busca de soluções

