Métodos de Busca Informada Part II

Problema de Satisfação de Restrições

Constraint Satisfaction Problem - CSP:

um conjunto de variáveis que podem assumir valores de um dado domínio

+

um conjunto de restrições que especificam propriedades da solução

Solução:
especificar valores para todas as variáveis tal que as restrições são satisfeitas

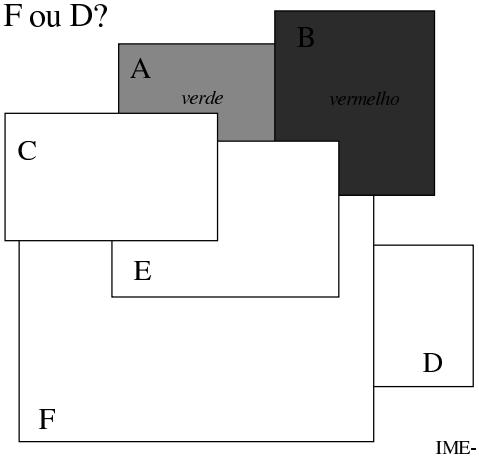
Problema de Satisfação de Restrições

Constraint Satisfaction Problem - CSP:

- cada variável V_i em CSP possue um **domínio** D_i
- **domínio:** conjunto de todos os valores possíveis que a variável pode assumir
- **restrições:** especificam sub-conjuntos do domínio de uma variável ou do produto cartesiano de mais variáveis.

Exemplo: colorir um mapa

Evitar colorir com a mesma cor os países adjacentes.
Cores possíveis: verde, vermelho e azul. Qual é o próximo país a ser colorido? C, E, F ou D?



5

CSP

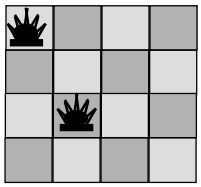
- Problemas de escalonamento em geral são problemas de CSP (jobshop, escala de aviões, gerenciamento de alarmes).
- Pode ser resolvido por algoritmos de busca gerais porém, algoritmos específicos são melhores

6

Problema das n-rainhas visto como um CSP

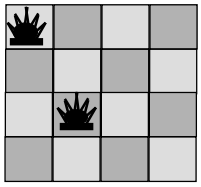
- **Variáveis**: localidade de cada uma das n rainhas (possíveis valores: posições do tabuleiro)
- **Restrições**: estado onde não existam duas rainhas se atacando (mesma linha, coluna ou diagonal)

4-rainhas



- Variáveis: ??
- Domínios: ??
- Restrições: ??

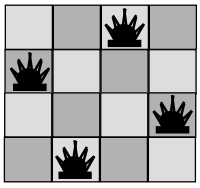
4-rainhas



- **Variáveis**: Q_i , *especifica a linha ocupada pela rainha da coluna i*
- **Domínios de cada variável**: *{1, 2, 3, 4} (linhas do tabuleiro)*
- **Restrições**: *conjunto de valores permitidos para pares de variáveis {(1,3), (1,4), (1,5), …, (2,4), (2,5), …}*

4-rainhas

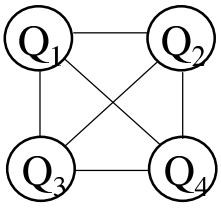
- Variáveis: Q_1, Q_2, Q_3, Q_4
- Domínios: $D_i = \{1,2,3,4\}$
- Restrições:
  - $Q_i \neq Q_j$ (não pode estar na mesma linha)
  - $|Q_i - Q_j| \neq |i - j|$ (ou mesma diagonal)
- Restrição também podem representadas pelo conjunto de valores permitidos, por exemplo, para (Q_1, Q_2), temos: (1,3) (1,4) (2,4) (3,1) (4,1) (4,2)



Q_1=2
Q_2=4
Q_3=1
Q_4=3

Gráfico de restrições

- CSP binário: cada restrição relaciona no máximo 2 variáveis
- Gráfico de restrições: nós são variáveis, arcos mostram as restrições



CSP

- Tipo de restrições:
  - **Unárias**. Especifica um subconjunto permitido do domínio para cada variável.
  - **Binárias** (relaciona 2 variáveis) (ex.: 8-rainhas). Especifica um subconjunto permitido do produto entre dois domínios
  - **Absolutas** (devem ser satisfeitas obrigatoriamento)
  - **Preferênciais** (podem ser relaxadas)
  - Restrições sobre domínios **discretos** ou **contínuos**

CSP

- Domínio discreto: podemos enumerar valores permitidos
- Domínio contínuo: envolve algebra sofisticada.
- Problemas de CSP menos complexos:
  - restrições absolutas, binárias ou unárias, com domínios discretos
- Ainda assim, existem problemas bem interessantes que satisfazem essas suposições

CSP: paradigma de linguagem de programação
(Trilogy, Charm/Bull)

Questão

Como aplicar um algoritmo geral de busca a um CSP?

- **Estado?**
- **Estado Inicial?**
- **Operadores?**
- **goal?**

8-rainhas: restrições binárias, domínio discreto e finito

- Seja
  - V_1 a linha ocupada pela rainha na primeira coluna
  - V_2 a linha ocupada pela rainha na segunda coluna.
- Os domínios para V_1 e V_2 são {1, 2, 3, 4, 5, 6, 7, 8}.
- A restrição de não ataque pode ser representada pelo conjunto de pares de valores permitidos para V_1 e V_2: {(1,3), (1,4), (1,5), … (2,4),(2,5), …}

15

CSP visto como Busca Geral

- **Estado**: conjunto de *variáveis* com seus valores associados
- **Estado Inicial**: todas as variáveis sem nenhum valor associado
- **Operadores**: atribuição de valores a uma variável a partir de um conjunto de seus valores possíveis (domínio da variável)
- **Teste do goal:** verifica se todas as variáveis possuem valores e se todas as restrições são satisfeitas

16

CSP visto como Busca Geral

- Profundidade máxima da árvore: n número de variáveis. Todas as soluções devem estar numa profundidade n.
- Número finito de variáveis: pode-se usar **DFS**
- Como a ordem em que valores são atribuidos as variáveis não importa, muitos algoritmos de CSP, atribuem valores possíveis a UMA variável de cada vez a partir de um nó.
- Fator de ramificação: máximo de $\{D_i\}$

Backtracking Search

- Em CSP, o teste de goal é decomposto em um conjunto de restrições sobre variáveis ao invés de ser uma caixa-preta ⟶ problema estruturado
- DFS não precisa fazer busca quando restrições já foram violadas
- Antes de gerar filhos um teste de violação de restrições deve ser aplicado ao nó fazendo com que a busca *retorne* para tentar outro caminho abandonando aquele: **backtracking search**

Forward checking

- **BS** pode falhar: restrições podem não ser violadas no nó local mas pode não existir combinações possíveis para o restante da busca
- **Forward checking** : olha para frente para detectar a possibilidade de não existir solução para aquele caminho:
  - para cada atribuição, **FC** elimina dos domínios de cada variável ainda não instanciada todos aqueles valores que conflitam com as variáveis já instanciadas.
  - se algum domínio se tornar vazio, a busca retorna para tentar uma outra escolha (backtracking).
- Questão: existem heurísticas para CSP?

Heurísticas para n-rainhas

- ***variável mais restrita:*** aumenta de n=30 para n=100 o tamanho de problema que pode ser tratado
- ***valor menos restritiva:*** aumenta de n=100 para n=1000 o tamanho de problema que pode ser tratado

Métodos de busca não informada

- Até aqui vimos variações de DFS:
  - Backtracking Search (BS)
  - Forward Checking (FC)
- Em que pontos da estratégia de busca é possível utilizar heurísticas?

Heurísticas para CSPs

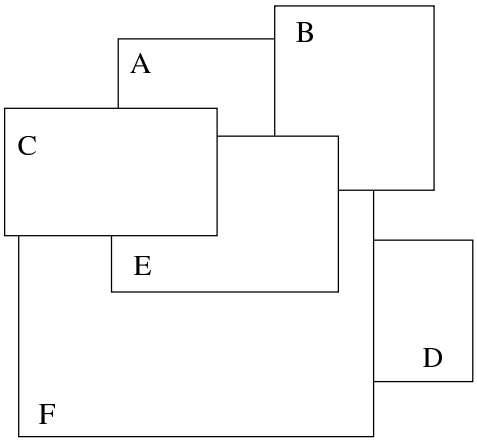
- É possível utilizar heurísticas:
  1) ?
  2) ?

Heurísticas para CSPs

- É possível utilizar heurísticas:
  1) na seleção de uma variável para fazer uma atribuição
  2) na escolha de um valor para ser atribuido à variável

Problema de colorir um mapa

- Variáveis:
- Domínios:
- Restrições:

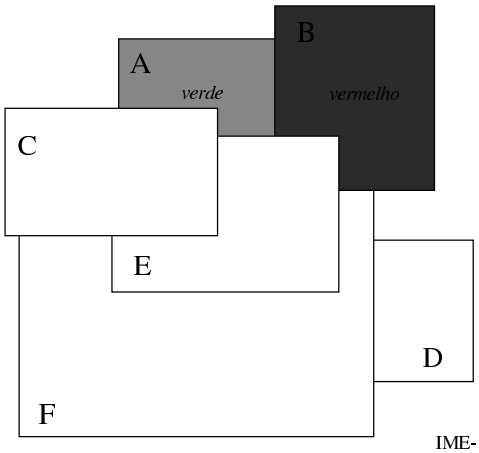


Problema de colorir um mapa

- Variáveis: países {A, B, C, D, E, F}
- Domínios: cores {verde, vermelho, azul}
- Restrições: pares com cores diferentes

Heurísticas para CSP

Depois de colorir A e B qual o país que devemos escolher?



Heurísticas para CSP

- Parece intuitivo escolher E para colorir com azul que corresponde à variável com o menor conjunto de valores possíveis:
  - E não pode ser verde e nem vermelho
  - E é a ***variável mais restrita***
- uso de **forward checking**

> Heurística da ***variável mais restrita***

Seleção da variável mais restrita com relação aos compromissos anteriores

- Deve ser usada com *FC*: verifica-se quais valores são ainda permitidos para cada variável, dadas as escolhas anteriores
- A cada ponto na busca, a variável com o *menor* conjunto de valores possíveis é selecionada (essa heurística tende a reduzir *b*)

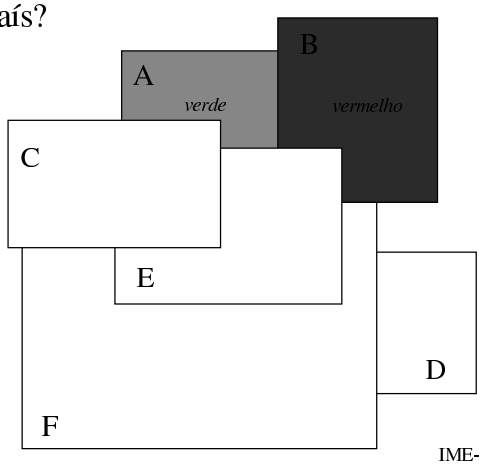
> Heurística da ***variável mais restritiva***

Seleção da variável mais restritiva com relação às variáveis sem valor

- isto é, seleciona-se a variável que está envolvida no maior número de restrições sobre as variáveis sem valor
- diminui o conjunto dos possíveis valores de um número maior de variáveis

Heurística para CSP

Se depois de atribuirmos valores para A (verde) e B (vermelho) escolhermos C, que cor deveremos escolher para este país?



> Heurística do ***valor menos restritivo***

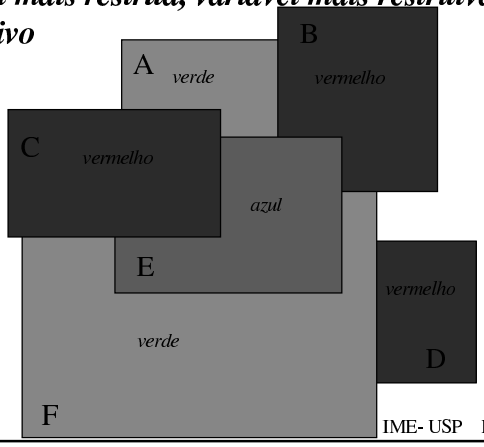
- Se depois de atribuirmos valores para A (verde) e B (vermelho) escolhermos C, parece que escolher vermelho é melhor que azul pois deixa mais liberdade para futuras escolhas

Heurística do ***valor menos restritivo***

- Uma vez selecionada a variável, essa heurística seleciona um valor que elimina o menor número de valores para serem atribuídos às variáveis sem valor e que estão conectadas com essa variável por alguma restrição
  - sobram mais valores possíveis para as próximas escolhas de variáveis
  - permite mais liberdade para futuras atribuições

Coloração de um mapa

Este mapa pode ser colorido sem se fazer busca. Método de busca em profundidade (DFS) com backward chaining e forward chaning e as heurísticas: ***variável mais restrita, variável mais restritiva*** e ***valor menos restritivo***



Heurística para CSP

- A aplicação dessas heurísticas no problema de colorir mapas encontra soluções pra problemas de profundidade 1000.
- Mapas com muita vizinhança => como determinar o número de cores necessárias?
- Problemas muito de complexidade exponencial podem causar falta de memória para o uso da estratégia FC