

Problemas de Satisfação de Restrições

(parte II)



Profa. Josiane M. Pinheiro
Prof. Sérgio R. P. da Silva

agosto/2006

Tratando restrições especiais

- Certos tipos de restrições ocorrem com frequência em problemas reais
- Podem ser tratados com algoritmos específicos, mais eficientes do que os algoritmos de uso geral
- Exemplos:
 - Restrições *TodosDiferentes*
 - No problema criptoaritmético todas as variáveis devem ter valores distintos
 - Restrições *noMáximo*
 - Restrições de recurso: realizar alguma tarefa com no máximo x de recursos

Restrições *TodosDiferentes*

- Detecção:
 - se existem m variáveis envolvidas na restrição
 - se elas tem n valores totalmente distintos possíveis
 - e se $m > n$, então a restrição não pode ser satisfeita

- Para resolver:
 - Remova da restrição qq variável que tenha o domínio unitário
 - Elimine o valor desta variável dos domínios das variáveis restantes
 - Se, em qq instante for produzido um domínio vazio, ou
 - existirem mais variáveis do que valores de domínios restantes
 - Existe uma inconsistência

Restrições *noMáximo*

□ Exemplo:

- Sejam PA_1, \dots, PA_4 os números de pessoas atribuídos a cada uma de 4 tarefas
- Temos a restrição de que não mais de 10 pessoas sejam atribuídas no total *noMáximo*(10, PA_1, PA_2, PA_3, PA_4)

□ Detecção:

- Podemos simplesmente verificar a soma dos valores mínimos dos domínios correntes
- Se esta soma for mais do que o máximo permitido, temos uma inconsistência
- Também podemos excluir o valor máximo de qq domínio se ele não for consistente com os valores mínimo dos outros domínios

Retrocesso Inteligente

- ❑ BUSCA-COM-RETROCESSO: falha -> volta para a variável precedente
 - Mas isto pode se totalmente inútil
- ❑ Ex: supõe as variáveis na ordem: Q, NGS, V, T, AM, AO, TN
- ❑ Supõe que geramos a atribuição $\{Q=R, NGS=G, V=B, T=R\}$
- ❑ Quando experimentamos AM, todo valor viola uma restrição
- ❑ Voltamos até T e experimentamos uma nova cor
- ❑ Obviamente, a mudança da cor atribuída a T não resolve o problema de AM
- ❑ Isto acontece pq T não está no **conjunto de conflito** de AM

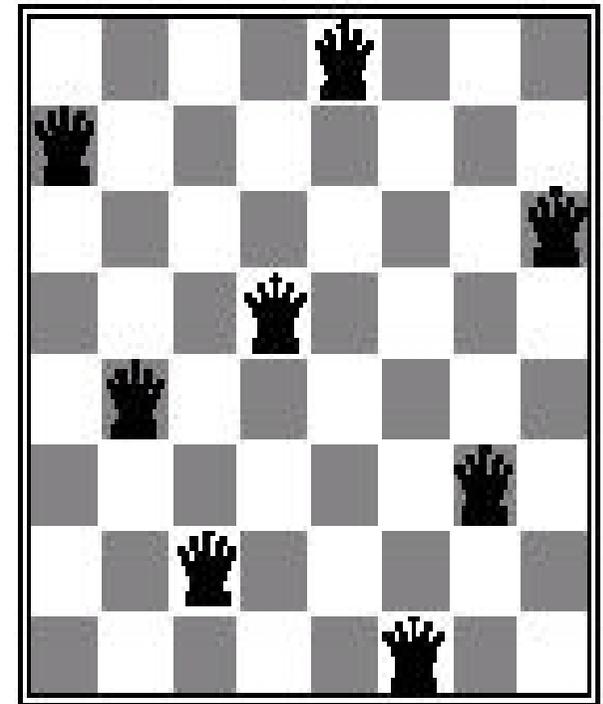
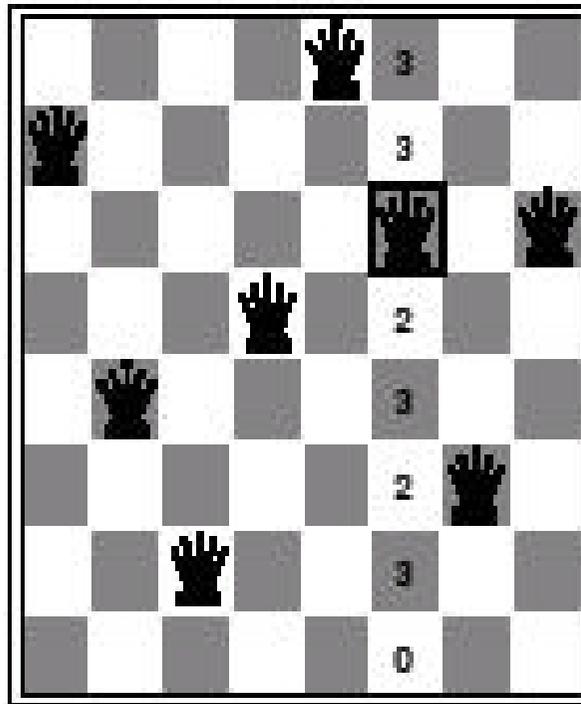
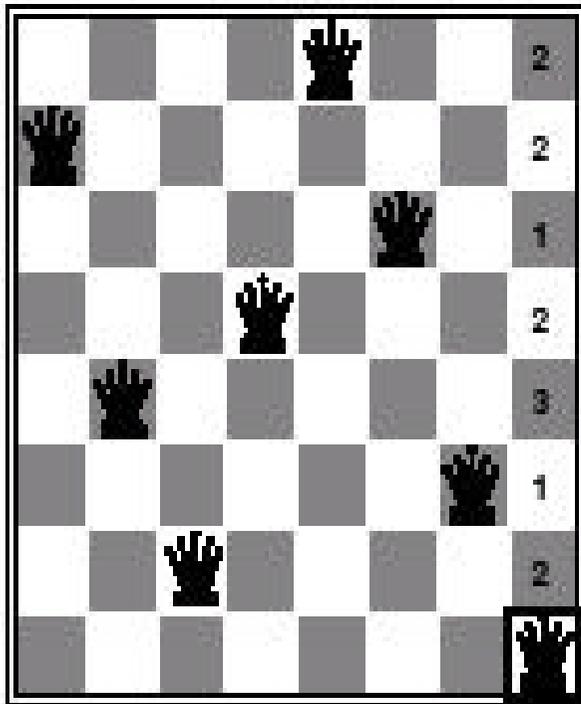
Conjunto de Conflito

- É o conjunto de variáveis que causaram a falha
 - Conjunto de conflito de $AM = \{Q, NGS, V\}$
- Conjunto de conflito de uma variável X são as variáveis previamente atribuídas que estão conectadas a X por uma restrição
- **Método de retorno orientado por conflito**
 - Volta até a variável mais **recente** do conjunto de conflito
 - Experimenta um novo valor para ela
 - Tenta resolver o problema mudando o valor das variáveis que causaram o problema

Busca Local para PSRs

- Utilizam uma formulação de estados completos
- O estado inicial atribui um valor para cada variável
- Função sucessor funcional alterando o valor de uma variável de cada vez
- Ex: Problema das n-rainhas
 - Estado inicial: configuração aleatória das n rainhas uma em cada coluna
 - Função sucessor: escolhe uma rainha e considera a possibilidade de movê-la para outro lugar na mesma coluna
- Escolha de um novo valor para uma variável: **Heurística de conflitos mínimos**

Problema das n-rainhas



função CONFLITOS-MÍNIMOS(*psr*, *max_etapas*) retorna uma solução ou falha

corrente ← uma atribuição inicial completa para *psr*

para *i*=1 até *max_etapas* faça

se *corrente* é uma solução para *psr* então retornar *corrente*

var ← uma variável em conflito escolhida ao acaso a partir de VARIÁVEIS[*psr*]

valor ← o valor *v* para *var* que minimiza CONFLITOS(*var*, *v*, *corrente*, *psr*)

definir *var* = *valor* em *corrente*

retornar falha

Heurística de conflitos mínimos para PSRs

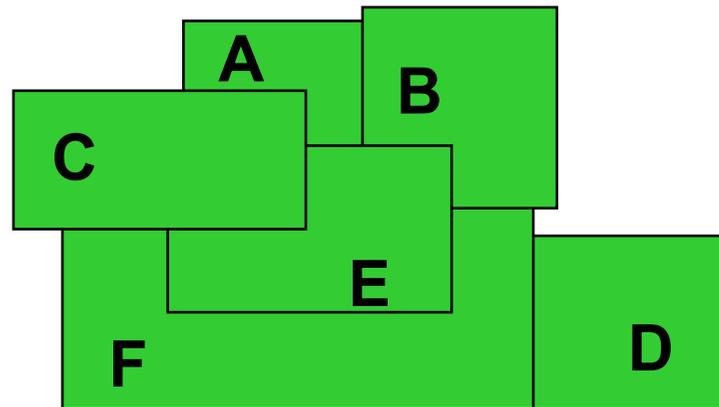
- É surpreendente quando recebe um estado inicial razoável
- O tempo de execução para problema das n-rainhas é praticamente independente do tamanho do problema
- Resolve até mesmo o problema de 1 milhão de rainhas em uma média de 50 etapas
- A heurística é utilizada na programação de observações do telescópio Hubble
 - reduzindo o tempo necessário para programar uma semana de observações de três semanas para + ou - 10 minutos

Heurística de conflitos mínimos

- Vantagem da busca local: utilizá-la em configuração on-line quando o problema se altera
- Importante para problemas de escalonamento
- Escala de linhas aéreas para uma semana
 - Pode envolver milhares de vôos
 - Dezenas de milhares de atribuições de pessoas
 - Mas um mau tempo em um aeroporto pode torná-la inviável
- Gostaríamos de reparar a escala com um número mínimo de mudanças
 - Utilizando um algoritmo de busca local a partir da escala atual

Para exercitar...

- ❑ Resolva o problema de colorir as áreas abaixo utilizando apenas as cores vermelho (R), verde (G) e azul (B).
- ❑ Áreas adjacentes devem ter cores diferentes



Para exercitar...

- Resolva o problema de criptoaritmética abaixo:
- TWO
- + TWO
- -----
- FOUR