

# Tratamento de problemas $\mathcal{NP}$ -difíceis: Busca Tabu

Cid C. Souza  
Eduardo C. Xavier

Instituto de Computação/Unicamp

29 de abril de 2011

## Busca Tabu

- Vimos que as heurísticas de busca local param quando atingem um ótimo local.
- Gostaríamos de evitar ótimos locais e tentar de alguma forma obter soluções melhores.
- Uma maneira de se fazer isto é deixar que a solução corrente piore na esperança de que num futuro obteremos soluções melhores.
- A busca tabu força a exploração de novas áreas do espaço de busca evitando soluções geradas previamente.
- Novas soluções são visitadas mesmo que tenham valores pior do que um ótimo local.

## Busca Tabu

```
Busca-Tabu    (* problema de minimização *)  
   $S \leftarrow \text{Gera-Solucao-Inicial};$   
   $S^* \leftarrow S;$   
  Enquanto (não atingir critério de parada) faça  
     $N(S) \leftarrow \text{Escolha-Vizinhos-Não-Tabu}(S);$   
     $S' \leftarrow \text{Sol. de menor custo em } N(S);$   
     $S \leftarrow S' ; (* \text{mesmo com custo pior que } S *)$   
    Se  $\text{custo}(S) < \text{custo}(S^*)$  então  
       $S^* \leftarrow S$   
    Atualize Lista-Tabu  
  fim-enquanto  
  Retornar  $S^*$ .
```

## Busca Tabu

- Inserir na busca local uma lista de movimentos **tabu** que impedem, por algumas iterações, que um determinado movimento seja realizado. *Objetivo*: evitar que uma solução seja revisitada.
- Repetir a busca local básica por  $\alpha$  iterações ou se nenhuma melhora foi obtida nas últimas  $\beta$  iterações. Após isso iniciar de uma nova solução.
- Os parâmetros  $\alpha$  e  $\beta$  são fixados *a priori*.
- Parâmetros a ajustar: tamanho da lista tabu  $t$ ,  $\alpha$  e  $\beta$ .

## Busca Tabu - SAT

- Vamos considerar o problema SAT na forma CNF com pesos nas cláusulas:  $n$  variáveis booleanas e  $m$  cláusulas.
- Uma solução é representada por um vetor binário  $x = (x_1, \dots, x_n)$ .
- A função de avaliação de uma solução corresponde a soma dos pesos das cláusulas satisfeitas.
- A vizinhança  $N(S)$  é definida pela inversão de um bit de  $S$ .
- Uma memória  $M$  armazena soluções tabu ou movimentos tabu.

## Busca Tabu - SAT

- Construir uma memória  $M$  que indique quantas iterações um movimento deve ficar tabu:

$$M(i) = \begin{cases} 0 & \text{não está tabu} \\ j & \text{está tabu pelas próximas } j \text{ iterações} \end{cases}$$

- A cada iteração as posições  $M(i) > 0$  são decrementadas de 1.
- Na vizinhança é considerado apenas aqueles tal que  $M(i) = 0$ .
- Vamos usar esta última memória no exemplo.

## Busca Tabu - SAT

- Suponha que temos oito variáveis em uma entrada do problema.
- Suponha uma solução em um determinado momento  $x = (1, 1, 0, 0, 0, 1, 1, 1)$  com valor 33.
- Suponha a memória  $M = (3, 0, 1, 5, 0, 4, 2, 0)$   
As únicas variáveis que podem ser invertidas são 2, 5 e 8.  
1 está tabu pelas próximas três iterações; 3 pelas próximas uma; 4 pelas próximas cinco; etc.
- Toda a vizinhança de  $x$  é  
 $x_1 = (0, 1, 0, 0, 0, 1, 1, 1)$ ;  $x_2 = (1, 0, 0, 0, 0, 1, 1, 1)$ ;  
 $x_3 = (1, 1, 1, 0, 0, 1, 1, 1)$ ;  $x_4 = (1, 1, 0, 1, 0, 1, 1, 1)$ ;  
 $x_5 = (1, 1, 0, 0, 1, 1, 1, 1)$ ;  $x_6 = (1, 1, 0, 0, 0, 0, 1, 1)$ ;  
 $x_7 = (1, 1, 0, 0, 0, 1, 0, 1)$ ;  $x_8 = (1, 1, 0, 0, 0, 1, 1, 0)$ .
- Mas somente vamos considerar em  $N(x)$  as soluções  $x_2$ ,  $x_5$  e  $x_8$ !

## Busca Tabu - SAT

- Suponha que a melhor solução é  $x_5$  com valor 32.
- Note que houve uma piora da solução já que o problema é de maximização.
- A memória deve ser atualizada então de

$$M = (3, 0, 1, 5, 0, 4, 2, 0)$$

para

$$M = (2, 0, 0, 4, 5, 3, 1, 0)$$

- O processo continua como o descrito.



## Busca Tabu - SAT

- Alguns ajustes podem ser feitos na busca tabu.
- Se uma solução  $x_j$  é muito melhor do que a melhor solução até o momento, deveríamos aceitar  $x_j$  mesmo que este seja tabu (critério de aspiração).
- Podemos aceitar soluções tabu com uma certa probabilidade que aumenta proporcionalmente a quão melhor é a solução tabu.
- Outras considerações estão ligadas ao horizonte de memorização. Em alguns momentos pode ser bom diminuir (por exemplo de 5 para 2 no exemplo) quando as soluções de uma região parecem promissoras.

## Busca Tabu - SAT

- Também pode ser usada uma outra memória  $F$  que indica a frequência de realização de inversões.
- Suponha que após um certo número de iterações temos  $F = (5, 7, 11, 3, 9, 8, 1, 6)$ .
- Se ficamos muito tempo da busca sem termos uma melhora significativa, a memória  $F$  pode ser usada.
- Ela indica por exemplo que o bit 7 foi o menos invertido nas últimas iterações.
- Podemos *diversificar* a busca!
- OBS: Após um certo número  $\gamma$  de iterações a memória  $F$  deve ser apagada.

## Busca Tabu - SAT

- Suponha  $F = (5, 7, 11, 3, 9, 8, 1, 6)$  e que a solução atual tenha valor 35 e a melhor até então tem valor 37.
- Os movimentos não-tabu nos fornecem soluções com valores  $v(x_2) = 30$ ,  $v(x_3) = 33$  e  $v(x_7) = 31$ .
- Suponha que nenhuma solução tabu tenha valor maior que 37, então não podemos aceita-las por aspiração.
- Podemos aceitar a melhor solução com uma penalização dada por  $F$ .

$$v'(x) = v(x) - 0.7 \cdot F(i)$$

- Teremos  $v'(x_2) = 30 - 0.7 \cdot 7 = 25.1$ ,  $v'(x_3) = 33 - 0.7 \cdot 11 = 25.3$  e  $v'(x_7) = 31 - 0.7 \cdot 1 = 30.3$ .
- Então vamos aceitar  $x_7$  mesmo tendo valor menor do que  $x_3$ .

## Busca Tabu - TSP

- Vamos ver mais um exemplo com o problema TSP.
- Uma solução corresponde a uma permutação dos números de 1 até  $n$ .  
Exemplo com oito cidades: (2, 4, 7, 5, 1, 8, 3, 6).
- A vizinhança  $N(t)$  de uma solução corresponde a trocas de posições de duas cidades.  
OBS: Você poderia considerar troca de arestas como na heurística 2-opt.
- Na tupla exemplo, existem  $\binom{8}{2} = 28$  vizinhos.

## Busca Tabu - TSP

- A memória  $M$  será representada por uma matriz indicando quais vértices foram trocados.
- Supondo o exemplo com oito cidades e 5 iterações como tabu, um exemplo de  $M$  é:

$$\begin{pmatrix} - & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ - & - & 0 & 0 & 0 & 5 & 0 & 0 \\ - & - & - & 0 & 0 & 0 & 4 & 0 \\ - & - & - & - & 3 & 0 & 0 & 0 \\ - & - & - & - & - & 0 & 0 & 2 \\ - & - & - & - & - & - & 0 & 0 \\ - & - & - & - & - & - & - & 0 \\ - & - & - & - & - & - & - & - \end{pmatrix}$$

- A matriz indica, por exemplo, que a última troca foi de 2 com 6. A troca mais antiga foi entre 1 e 4. Note que só 5 trocas estão tabu.

## Busca Tabu - TSP

- Da mesma forma, uma memória de frequência  $F$  pode ser construída.
- No mesmo exemplo, suponha que guardamos informações das 50 iterações anteriores ( $\gamma = 50$ ).

$$\begin{pmatrix} - & 0 & 2 & 3 & 3 & 0 & 1 & 1 \\ - & - & 2 & 1 & 3 & 1 & 1 & 0 \\ - & - & - & 2 & 3 & 3 & 4 & 0 \\ - & - & - & - & 1 & 1 & 2 & 1 \\ - & - & - & - & - & 4 & 2 & 1 \\ - & - & - & - & - & - & 3 & 1 \\ - & - & - & - & - & - & - & 6 \\ - & - & - & - & - & - & - & - \end{pmatrix}$$

- A matriz indica, por exemplo, que a troca de 7 com 8 foi a que mais ocorreu (seis vezes). Você pode usar esta informação para diversificar a busca penalizando trocas muito frequentes.

## Busca Tabu - TSP

Resumindo:

- Busca-Tabu é uma heurística local que tentar fugir de ótimos locais:
  - ▶ Aceitar soluções mesmo que sejam piores.
  - ▶ Criar tabela tabu para evitar movimentos já realizados.
  - ▶ Usar tabela de frequência para diversificar a busca.