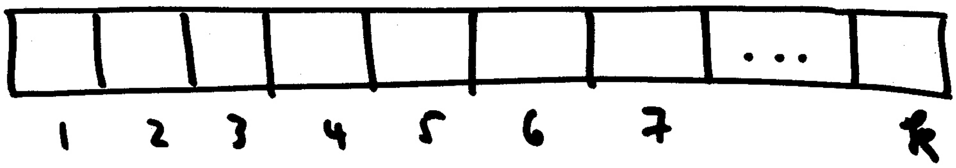
"TABU SEARCH" (BUSCA TABU)

- Meta heurística (heurística que pilota outras heurísticas)
- Método extremamente geral para encontrar um mínimo global (esperado) de uma função f definida em um conjunto viável X.
- Descrição geral: (processo de busca)

  ∴
  - sol. viável $s \in X$
  - uma vizinhança $N(s)$
  - amostra de soluções $s' \in N(s)$
  - escolher melhor s' gerada
  - mover de s para s'
    $m(s)$ movimento de s para s' (modificação)
  - para evitar mínimo local, aceita-se o movimento mesmo se $f(s') > f(s)$
  - mas, assim o procedimento pode entrar em ciclo .... então:
    - usar lista T de soluções tabu, contendo as últimas k soluções visitadas (para dado k)

- Se $s' \in T$ então o movimento $m(s)$ **não será permitido**

- Quando um movimento é realizado, a solução encontrada é **introduzida** em T e (caso $k = |T|$ seja fixo) a solução **mais antiga** é **removida** de T.



OBS.: Armazenar as k últimas soluções pode consumir muito espaço (Ex.: cada solução um vetor ...), Pode-se então armazenar os movimentos reversos (de uma dada solução chega-se à anterior da lista).

Esta mudança poderá proibir algumas soluções que não foram visitadas anteriormente, mas que poderiam ser geradas da solução corrente por um movimento tabu...

- **Nível de aspiração**:

  cancelar um status tabu de movimento se s' obtida de s é "suficientemente boa.

  $A(z)$ para cada valor z de f, ~~...~~ inicialm. $A(z) = z - 1$ Assim: de $s \longrightarrow s'$ ~~(par~~ aceita-se se $f(s') < A(f(s))$. Neste caso faz-se $A(f(s)) = f(s') - 1$ (valores inteiros de f

ALGORITMO : BUSCA TABU (GERAL)

ENTRADA:

X : CONJUNTO DE SOLUÇÕES VIÁVEIS;

f : FUNÇÃO OBJETIVO (VALORES INTEIROS);

NBMAX : MÁXIMO NÚMERO DE ITERAÇÕES ENTRE DUAS MELHORAS;

$|T_i|$: TAMANHO DE CADA LISTA TABU T_i;

INÍCIO:

PARA CADA VALOR POSSÍVEL x DE f FAÇA
    NÍVEL DE ASPIRAÇÃO $A(x)$ IGUAL A $x-1$;

GERAR UMA SOLUÇÃO VIÁVEL ALEATÓRIA $s \in X$

ESCOLHER ARBITRARIAMENTE LISTAS TABU T_i;

NBITER := 0; (* ITERAÇÃO CORRENTE *)

BESTSOL := s; (* SOLUÇÃO MELHOR *)

BESTVALUE := $f(s)$; (* VALOR DA MELHOR SOLUÇÃO *)

BESTIT := 0; (* ITERAÇÃO EM QUE BESTSOL FOI ENCONTRADA *)

ENQUANTO $f(s)$ < estimativa_do_otimo **E** NBITER − BESTIT < NBMAX **FAÇA**

NBITER := NBITER + 1;

DETERMINAR UMA MELHOR SOLUÇÃO VIÁVEL s' EM $N(s)$ PARA A QUAL O MOVIMENTO $[s \to s']$ NÃO SATISFAZ QUALQUER CONDIÇÃO TABU OU $f(s') \leq A(f(s))$;

SE $f(s')$ < BESTVALUE **ENTÃO**

  BESTVALUE := $f(s')$; BESTIT := NBITER; BESTSOL := s';

ATUALIZAR LISTAS TABU;

SE $f(s') \leq A(f(s))$ **ENTÃO** $A(f(s)) := f(s') - 1$;

SENÃO **SE** $f(s) \leq A(f(s'))$ **ENTÃO** $A(f(s')) := f(s) - 1$;

$s := s'$;

SAÍDA: BESTSOL (MELHOR SOL. VIÁVEL ENCONTRADA) E BESTVALUE (SEU VALOR)

Ex.: MINIMUM-COST TREES

Regra de escolha: Selecionar o arco admissível de menor custo

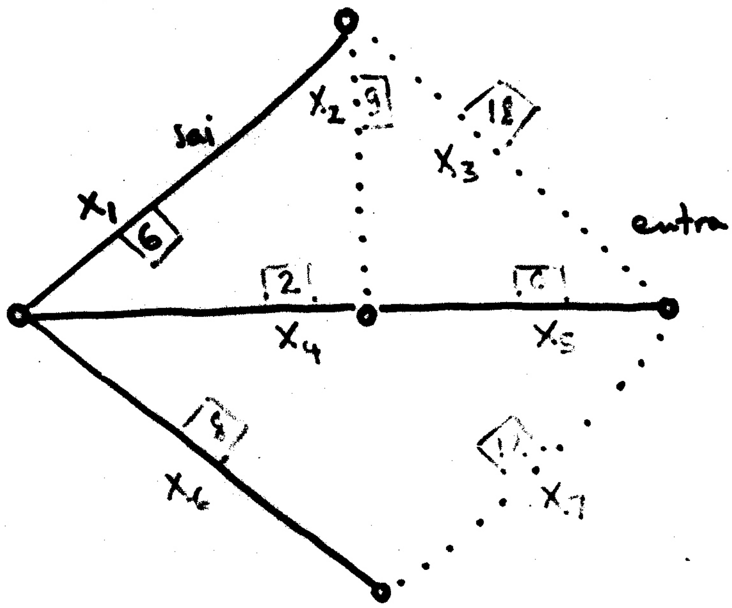
Restrição Tabu: Proibir saída de um dos dois arcos mais recentemente adicionados

Critério de aspiração: Violar a restrição tabu se a troca produz uma nova "melhor solução corrente"

Restrições: $x_1 + x_2 + x_6 \le 1$, $x_1 \le x_3$ (Penalidade de violação = 50)

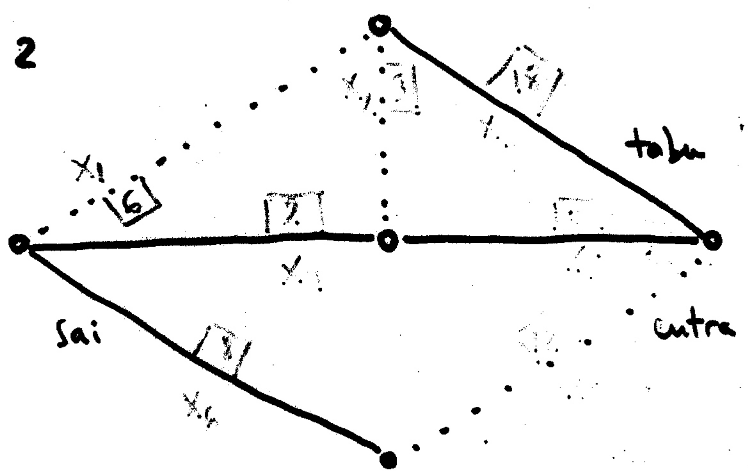
$$x_j = \begin{cases} 1 & \text{se arco } x_j \in \text{árvore} \\ 0 & \text{se arco } x_j \notin \text{árvore} \end{cases}$$

Iteração 1



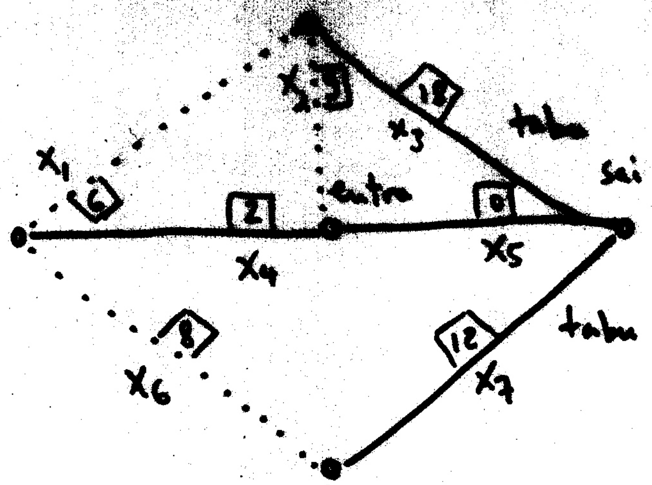
Custo: 16 + 100 (melhor sol. corrente ... inviável)

Iteração 2



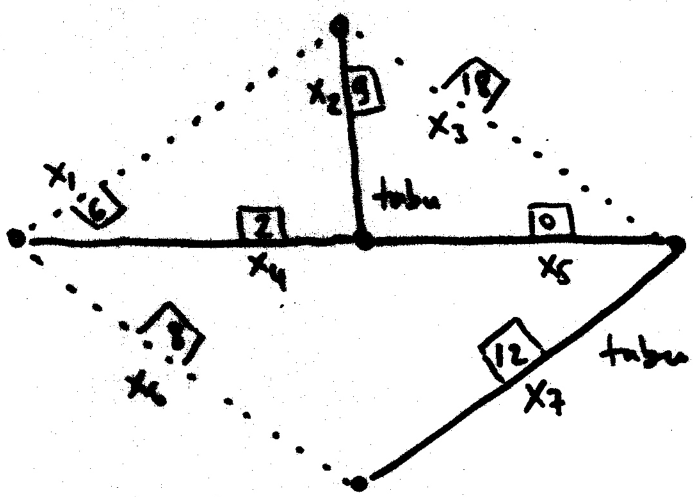
Custo: 28 (nova melhor sol. corrente ... ótimo local)

Iteração 3



custo : 32 (aspiração quebra "status tabu")

Iteração 4



Ex.: conj. indep. vértices (TS)

Dado grafo $G=(V,E)$
encontrar se possível um conj. indep. vért. S de tamanho t.

- conj. viável : partições de V em
  $s=(S,V-S)$
  em dois conjuntos S, $V-S$ onde $|S|=t$.

- vizinhança : $N(s)$ todas soluções geradas de s permutando um nó $x\in S$ (adjacente com pelo menos outro nó de S) com um nó y de $V-S$.

- função objetivo : $f(s)=|E(S)|$
  conta o nº de arestas de G que possui ambos vértices em S.

∴ Uma solução s será um conj. indep. vert. se e só se $\underline{f(s)=0}$.

- Usa-se TS para minimizar f.

Ex.: <u>Coloração</u> (TS)

Encontrar uma k-coloração de um grafo $G=(V,E)$ para dado k.

- <u>conj. viável</u> : partição de V

  $s = (S_1, S_2, \ldots, S_k)$ em k subconj.

  Esta partição será uma coloração se cada S_i for um conj. indep. de vértices, isto é, se cada conj. $E(S_i)$ de arestas de G com ambos os vértices corresp. em S_i é vazio.

- <u>função objetivo</u> : $f(s) = \sum_{i=1}^{k} |E(S_i)|$.

  <u>sol. ótima</u> : $f(s)=0$ (mínimo $f(s)$)

- <u>vizinhança</u> : de uma solução $s=(S_1, S_2, \ldots, S_k)$ será toda solução s' que podemos obter de s movendo algum nó $x \in S_i$ (onde x é adjacente a pelo menos outro nó de S_i) para algum outro subconj. S_j da partição.