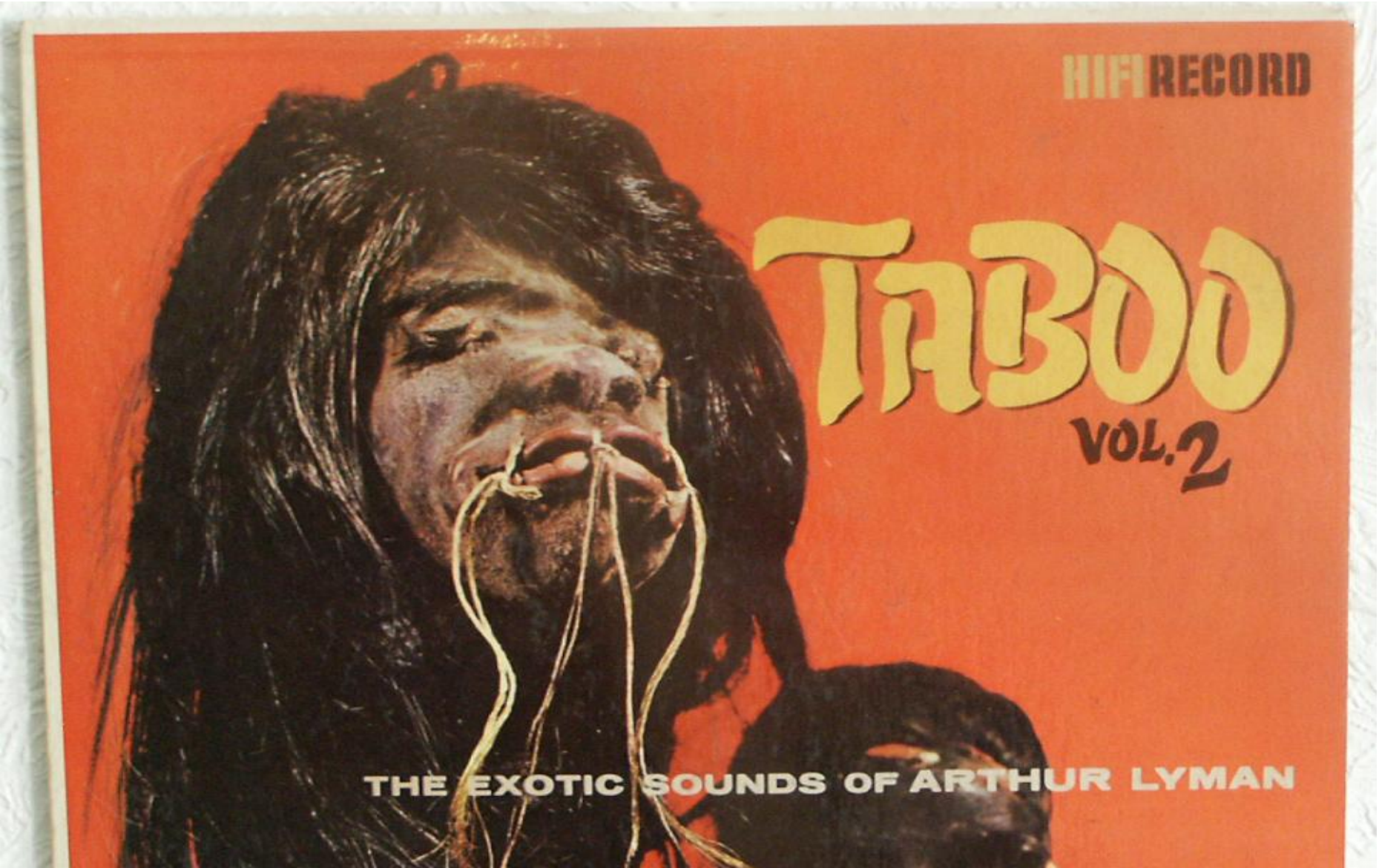
Prof's Joke: Tabu Search is not this



Tabu Search

Tabu Search (TS) algorithm is based on the idea that you want to prevent the search from going back to some regions.

- Tabu means “forbidden”. In TS some *moves* (going from s0 to snew) are **tabu**. ⇒ disallowed

- Why would you want to prohibit some moves? **

⇒ don't repeat work, for example

Problem: Find best ordering of materials in a composite

- To explain the Tabu Search method we will first do simple example, which is a permutation problem.

- This problem is to find the ordering of layers of material that **maximize** the insulating property of the composite material.

- If there are 7 layers of material, then any permutation of the integers from 1 to 7 represents a possible layering of the seven materials (e.g. 1234567 is a different solution from 1234576

different permutations

swap: exchange 2 items in a permutation

Problem: Find best ordering of materials in a composite

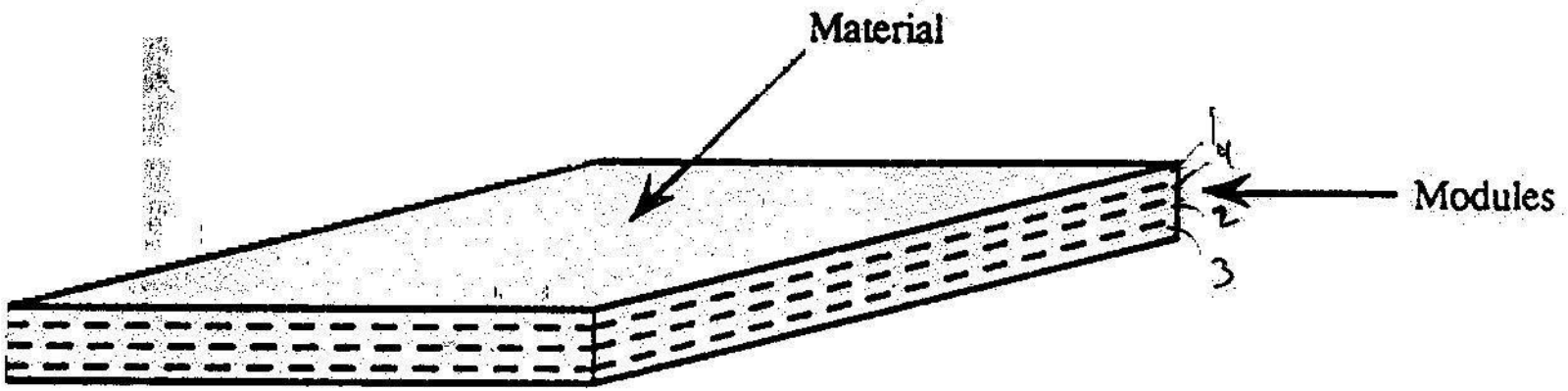


Figure 3.1: Modules in an insulating material

Cost function in Composite Materials Problem

no knowledge of function internals.

- The objective function **Cost(S)** is the insulating value of the material. Hence if S=(1567342), then Cost(S) is the insulating value of the composite material consisting of material 1 followed by material 5, followed by material 6, etc.
- In this example the objective function **Cost(S)** is a "black box"—we don't know its functional form. **Cost(S)** is computed in a separate computer code (like the *modules* in our course projects).
- We will two example problems for Tabu Search—we will call this the "First Tabu Search Problem".

Neighborhood

- The neighborhood **N(S)** is defined to be a swap of any two elements in the permutation S.

- We will define as **tabu** any swap that has taken place in the last **m** iterations. Hence you cannot repeat the same swap in m iterations. (Later we will discuss that alternative moves can be defined to be tabu.)

- Hence if 1234567 has changed to 1634527, which kinds of swaps are prevented for m iterations?
-
- How do we keep track of this? (Figures on Iteration 0 to Iteration 26 from next page)

cannot swap(2,3) or swap(3,4)

swap(3,4) swap(2,3)

eg. 1 3 4 2 → 1 4 3 2 → 1 4 2 3 →

in this example, swaps are value-based

NOT position-based

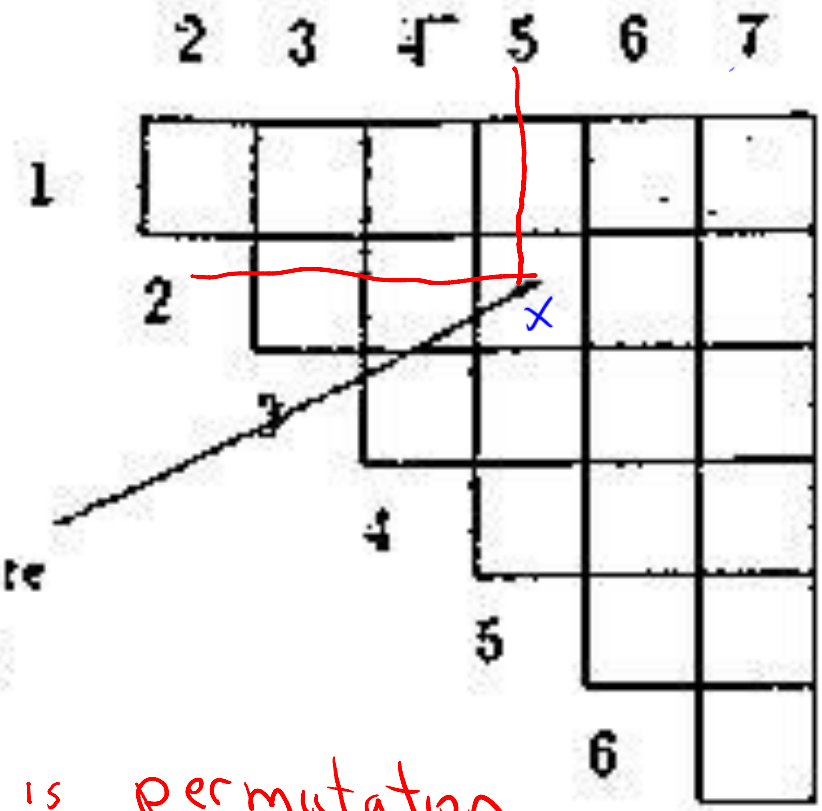
The steps that follow are based on two rules determining which steps for changing the perturbations are **tabu**:

- ***Paired Swap Tabu Restriction***—if a swap between k and j occurs, then a swap between k and j cannot occur for m more iterations unless the "best solution aspiration criterion" is satisfied

- ***Best Solution Aspiration Criterion—***the tabu on swapping k and j can be overridden if Cost (NewS) is better than the best solution (globalmax) in all previous iterations.

⇒ if the swap is <u>actually</u> good, do it! ⇒ Cost (newS) > best-cost

Table used to keep track of previous decisions



if $(x_{2,5} > 0) \Rightarrow$ cannot swap!

Remaining tabu tenure for module pair (2,5)

$-- x_{2,5}$ each iteration

decision is permutation

"module pair" = swap that occured

Figure 3.5: Tabu data structure for attributes consisting of module pairs exchanged → "sparse" matrix

Goal is to **Maximize** F(S) the Insulation Value (" First Tabu Search Problem")_

F(Snew) - F(Scurrent)

Iteration 0 (*Starting point*)

Current solution

best swap

S_{curr} =

2	5	7	3	4	6	1

permutation

F(Scurr) **Insulation Value=10**

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

All entries zero

Out of all possible swaps!

Top 5 candidates

Swap	Value	
5,4	6	*
7,4	4	
3,6	2	
2,3	0	
4,1	-1	

Snew = Scurr + swap

You define how many swaps you will consider out of a maximum of n(n-1)/2. Only the best 5 are shown in this plot.

Iteration 1

Iteration 1

Current solution

best swap

2	4	7	3	5	6	1

Insulation Value=16

10+6

denotes best swap

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

Top 5 candidates

Swap	Value
3,1	2
2,3	1
3.6	-1
7.1	-2
6.1	-4

we could make non-optimal moves if desired

Swapping 5 and 4 was the decision in previous iteration so we put m=3 in the box for (5,4) swap. The tenure length is m.

Iteration 2

Iteration 2

Current solution

best non-tabu move

2	4	7	1	5	6	3

Tabu moves!

Insulation Value=18

(10+6)+2

Tabu structure

	2	3	4	5	6	7
1		1				
2						
3						
4				2		
5						
6						

decremented

Top 5 candidates

Swap	Value	
1,3	-2	T
2,4	-4	*
7,6	-6	
4,5	-7	T
5,3	-9	

all moving away from best value

The new current solution becomes the best solution found so far with an insulating value of 18. At this iteration, two exchanges are classified tabu, as indicated by the nonzero entries in the tabu structure.

T

Iteration 3 : What is F(Snew) with 4-5 Swap? Why is there an *and a T next to 4,5?

Iteration 3

Current solution best move..

14+6=20 >18 ← previous global best ⟹ DO IT ⟹ override tabu

4	2	7	1	5	6	3

1

Tabu moves!

Insulation Value=14

((10+6)+2) −4

Previous Best S has objectiv = 18

Tabu structure

	2	3	4	5	6	7
		2				
2			3			
3						
4				1		
5						
6						

decremented

Top 5 candidates

Swap	Value	
4,5	6	T*
5,3	2	
7,1	0	
1,3	-3	T
2,6	-6	

Overriding TABU In Iteration 3

- We overrode the tabu to do a 4,5 swap in Iteration 3 because the objective function of the solution obtained would be 14+6= 20, which is better than any of the previous solutions found.
- This is an "aspiration by objective", meaning that you override the tabu if by doing so you obtain a better answer than has been previously found in all previous iterations.

Iteration 4

Iteration 4

Current solution

best non tabu move

5	2	7	1	4	6	3

tabu move

Insulation Value=20

Tabu structure

	2	3	4	5	6	7
1		1				
2			2			
3						
4				3		
5						
6						

decremented

reset to m=3

Top 5 candidates

Swap	Value	
7,1	0	*
4,3	-3	
6,3	-5	
5,4	-6	T
2,6	-8	

Additional Variations on Tabu Search: Adding a Frequency-based Memory

- You can add “frequency-based memory to measure the number of times a given pair has swapped over the entire search
- Then you can add a frequency element to your criterion for choice of the move to select as your next solution.
- For example you can penalize the value by the frequency (see next slide)

cute trick:
use 1 matrix to
store both frequency
and recency:

upper right of matrix
is same as before

Iteration 26
Current solution

1	3	6	2	7	5	4

Insulation Value=12

Tabu structure
(Recency)

	1	2	3	4	5	6	7
1				3			
2							
3	3					2	
4	1	5					1
5		4		4			
6			1		2		
7	2			3			

(Frequency)

frequency →

Top 5 candidates

Swap	Value	Penalized Value
1,4	3	3
2,4	-1	-6
3,7	-3	-3
1,6	-5	-5
6,5	-4	-6

frequency

$-6 = -1 - (5)$

for (2,4)

Tabu Search Applied to Problem Solved Earlier by SA and GA
"Second Tabu Search Problem"

- Next Slide shows the application we did before for Simulated Annealing, which is to find the best integer solution for a cubic polynomial.
- The neighborhood is defined as binary strings created by one flip of a bit in the Scurrent string.
- The neighborhood is picked by "cycling," which we will discuss.

A simple example. Maximize $f(x) = x^3 - 60x^2 + 900x + 100$.

starting value

S0 = 10011

m = 3

A bit change (flip) is TABU here if the same bit was flipped in a previous iteration during the tenure period.

cycling = circular shift

Changed

Table 1.7 Maximizing $f(x)$

Iteration	bit	string	f	new string
1	1	0 0 0 1 1	2287	
	2	1 1 0 1 1	343	
	3	1 0 1 1 1	1227	
	4	1 0 0 0 1	2692	**
(bit 4 tabu until iteration 5)				1 0 0 0 1
2	5	1 0 0 0 0	3236	**
	1	0 0 0 0 1	941	
	2	1 1 0 0 1	725	
	3	1 0 1 0 1	1801	
(bit 5 tabu until iteration 6)				1 0 0 0 0
3	4	tabu		
	5	tabu		
	1	0 0 0 0 0	100	
	2	1 1 0 0 0	964	**
(bit 2 tabu until iteration 7)				1 1 0 0 0
4	3	1 1 1 0 0	212	
	4	tabu		
	5	tabu		
	1	0 1 0 0 0	3972	**
(bit 1 tabu until iteration 8)				0 1 0 0 0
5	2	tabu		
	3	0 1 1 0 0	3988	
	4	0 1 0 1 0	4100	**
	5	tabu		
(bit 4 tabu until iteration 9)				0 1 0 1 0
6	1	tabu		
	2	tabu		
	3	0 1 1 1 0	3684	**
	4	tabu		
(bit 3 tabu until iteration 10)				0 1 1 1 0

no longer tabu

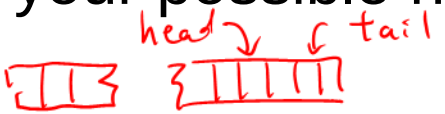
"Second Tabu Search Problem)

- In the previous example, there are five bits in the decision vector S. Conventionally the neighborhood would be all possible flips of one digit giving __5__ elements of the neighborhood.
- In the example are all the possible flips of one bit being considered? No If No, then what would be a reason ? reduce calls of COST function
- How are the elements being considered selected? Is it just random selection? No! ?

  ⇒ cycling ⇒ circular/rotary shift

Cyling

- Cyling is a way to deterministically reduce the size of your neighborhood.
- Cyling sets up a procedure by which you systematically select only a portion of your possible neighborhood for cost evaluation. ⇒



⇒ rotary shift head & tail

- Usually cyling is implemented so that within K iterations you have done each possible type of "sub-neighborhood" once. K is the cycle length. sub neighbourhood = head to tail
- What is the sub-neighborhood and what is the cycle length in the polynomial example. sn = 4, k = 5 ?

Note in "First Tabu Problem" the Neighborhood is ALL Possible Swaps

Iteration 1

Current solution

2	4	7	3	5	6	1

Insulation Value=16

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

Top 5 candidates

Swap	Value
3,1	2
2,3	1
3,6	-1
7,1	-2
6,1	-4

$\frac{n(n-1)}{2}$ = # of pairs to swap for permutation of length n

How many possible swaps are there for n=7? Only top 5 are shown here.

Repeated Slide of FIRST Tabu Search Example

What is the TOTAL number of swaps that were computed? (Only the top 5 were plotted but the computations were done for the entire neighborhood.

Iteration 0 (*Starting point*)

Current solution

S_{curr} =

2	5	7	3	4	6	1

permutation

F(scurr) **Insulation Value=10**

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

All entries zero

Top 5 candidates

Swap	Value	
5,4	6	*
7,4	4	
3,6	2	
2,3	0	
4,1	-1	

Snew = Scurr + swap

You define how many swaps you will consider out of a maximum of _____.Only the best 5 are shown in this plot.

End of handout 9-22-10

How could you create a cycle for this problem?

What would be tabu with this cycling Scheme for First Tabu Problem?

-

Range of ways to define "tabu"

Illustrative Tabu Restrictions

A move is tabu if:

(R1) x_j changes from 1 to 0 (where x_j previously changed from 0 to 1).

(R2) x_k changes from 0 to 1 (where x_k previously changed from 1 to 0).

(R3) at least one of (R1) and (R2) occur. (This condition is more restrictive than either (R1) or (R2) separately—i.e. it makes more moves tabu.)

(R4) both (R1) and (R2) occur. (This condition is less restrictive than either (R1) or (R2) separately—i.e. it makes fewer moves tabu.)

(R5) both (R1) and (R2) occur, and in addition the reverse of these moves occurred simultaneously on the same iteration in the past. (This condition is less restrictive than (R4).)

(R6) $g(x)$ receives a value v' that it received on a previous iteration (i.e. $v' = g(x')$ for some previously visited solution x').

(R7) $g(x)$ changes from v'' to v', where $g(x)$ changed from v' to v'' on a previous iteration (i.e. $v' = g(x')$ and $v'' = g(x'')$ for some pair of solutions x' and x'' previously visited in sequence.)

Other Aspiration Criteria

- The exisiting algoirthm uses **Global aspiration by objective**, i.e. the tabu can be overridden if the move results in an Objective Funciton value that is better than any value found previously.
- Alternative is Regional Aspiration by Objective:
  - Divide the search space into regions. An example of region r is

$$\text{Region}_r = \{S \text{ given } \text{Min}_r \leq S_j \leq \text{Max}_r \text{ for } S = (S_1, S_2, \ldots, S_n)\}$$

Then for the rth region, you compute BestS_r, and the aspiration criterion is based on this regional best solution

Illustrative Aspiration Criteria

Aspiration by Default: If all available moves are classified tabu, and are not rendered admissible by some other aspiration criteria, then a 'least tabu' move is selected. (For example, select a move that loses its tabu classification by the least increase in the value of *current_iteration*, or by an approximation to this condition.)

Aspiration by Objective: Global form (customarily used): A move aspiration is satisfied, permitting x^{trial} to be a candidate for selection, if $c(x^{trial}) < best\_cost$.

Regional form: Subdivide the search space into regions $R \in \mathbf{R}$, identified by bounds on values of functions $g(x)$ (or by time intervals of search). Let $best\_cost(R)$ denote the minimum $c(x)$ for x found in R. Then, for $x^{trial} \in R$, a move aspiration is satisfied (for moving to x^{trial}) if $c(x^{trial}) < best\_cost(R)$.

Aspiration by Search Direction: Let $direction(e) = improving$ if the most recent move containing $\bar{e}$ was an improving move, and $direction(e) = nonimproving$, otherwise. ($direction(e)$ and $tabu\_end(e)$ are set to their current values on the same iteration.) An attribute aspiration for e is satisfied (making e tabu-inactive) if $direction(e) = improving$ and the current trial move is an improving move, i.e. if $c(x^{trial}) < c(x^{now})$.

Aspiration by Influence: Let $influence(e) = 0$ or 1 according to whether the move that establishes the value of $tabu\_start(e)$ is a low influence move or a high influence move ($influence(e)$ is set at the same time as setting $tabu\_start(e)$). Also, let $latest(L)$, for $L = 0$ or 1, equal the most recent iteration that a move of influence level L was made. Then an attribute aspiration for e is satisfied if $influence(e) = 0$ and $tabu\_start(e) < latest(1)$. ($e$ is associated with a low influence move, and a high influence move has been performed since establishing the tabu status for e.) For multiple influence levels, $L = 0, 1, 2, \ldots$, the aspiration for e is satisfied if there is an $L > influence(e)$ such that $tabu\_start(e) < latest(L)$.