Heuristic Optimization Methods

Tabu Search: Introductory Topics

Agenda

- Introduction to Tabu Search
  - Tabu criterion
    - Tabu Tenure
  - Aspiration
  - Examples of use
    - Knapsack
    - TSP
  - Intensification
  - Diversification

Terminology: Neighborhoods (1)

- To make sure we understand this correctly:
- A **neighborhood** is a set of solutions that are close to one given solution
  - A set of solutions that can be reached from another by making only one move
  - $N(x)$ usually denotes the neighborhood of x
- A **neighborhood operator** is a function that takes a solution and returns its neighborhood
  - $N: S \rightarrow 2^S$
  - So, while a neighborhood is one specific collection of neighbors, a neighborhood operator is the blue-print for how to make those neighbors if we are given an initial solution around which to build the neighborhood

Terminology: Neighborhoods (2)

- Let us assume we have a knapsack problem, with n binary variables
- Furthermore, let us use flip-neighborhoods
- That is, $N(x)$ = {all y in S, such that the Hamming distance between y and x is 1}
    - N is the neighborhood operator
    - $N(x)$ is the neighborhood of solution x

Tabu

- The word tabu (or taboo) comes from Tongan
  - a language of Polynesia
  - used by the aborigines of Tonga island to indicate things that cannot be touched because they are sacred

- *”Loaded with a dangerous, unnatural force”*
- *”Banned due to moral, taste or risk”*

Tabu Search

- Tabu Search:
  - Cut off the search from parts of the search space (temporarily)
  - Guide the search towards other parts of the search by using penalties and bonuses
- Uses principles for intelligent problem solving
- Uses structures that are exploring the search history, without remembering everything
  - Branch&Bound, A*: have complete memory
  - Simulated Annealing: have no memory

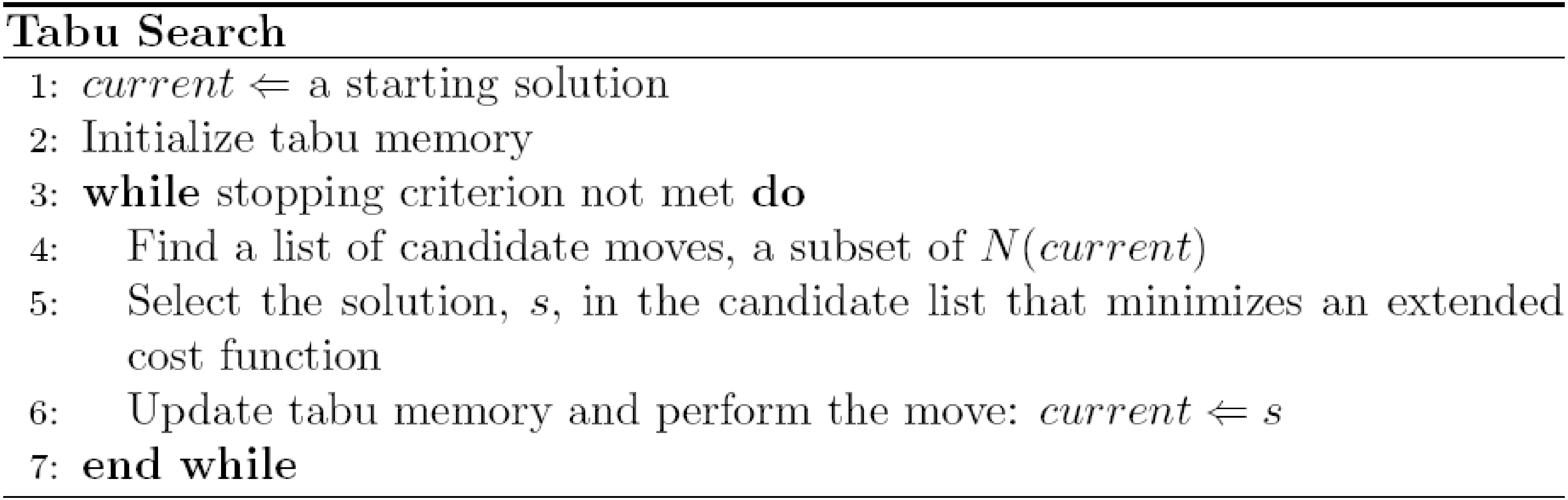
Origin of Tabu Search

- Fred Glover 1986: ”Future paths for integer programming and links to artificial intelligence”
- Pierre Hansen 1986: ”The Steepest Ascent/Mildest Descent Heuristic for Combinatorial Optimization”
- *Tabu* coined by Glover

Main Ideas of Tabu Search

- Based on Local Search – LS
- Allows non-improving moves
  - can exit local optima
- Uses extra memory to avoid looping, and to diversify the search
- General strategy for controlling a LS, or other “inner” heuristic
- *Meta-Heuristic* (Glover)

General Formulation



Tabu Search

```
$current \Leftarrow$ a starting solution
Initialize tabu memory
while stopping criterion not met do
    Find a list of candidate moves, a subset of $N(current)$
    Select the solution, $s$, in the candidate list that minimizes an extended cost function
    Update tabu memory and perform the move: $current \Leftarrow s$
end while
```

Some Critical Choices

- Choice of neighborhood, *N*
- Definition of the tabu memory
- How to select the candidate list
- The definition of the evaluation function
  - – Improvement in solution values
  - – Tabu criteria
  - – Aspiration criteria
  - – Long term strategies
    - Diversification, intensification, …

Basic Tabu Search

- Local Search with "Best Improvement" strategy
  - Always select the best move
- But: Some neighbors are *tabu*, and cannot be selected
  - Defined by the *tabu criterion*
  - Tabu neighbors might be selected anyway if they are deemed to be good enough
    - *Aspiration criterion*
- Memory – tabu list

The Tabu Criterion (1)

- In Tabu Search, we allow moving to a worse solution
- Since we (in basic TS) always select the ”Best Improvement”, how can we avoid cycling between solutions?
- The answer is the tabu criterion:
- We are not allowed to move to solutions that we have visited before
  - They are tabu!

The Tabu Criterion (2)

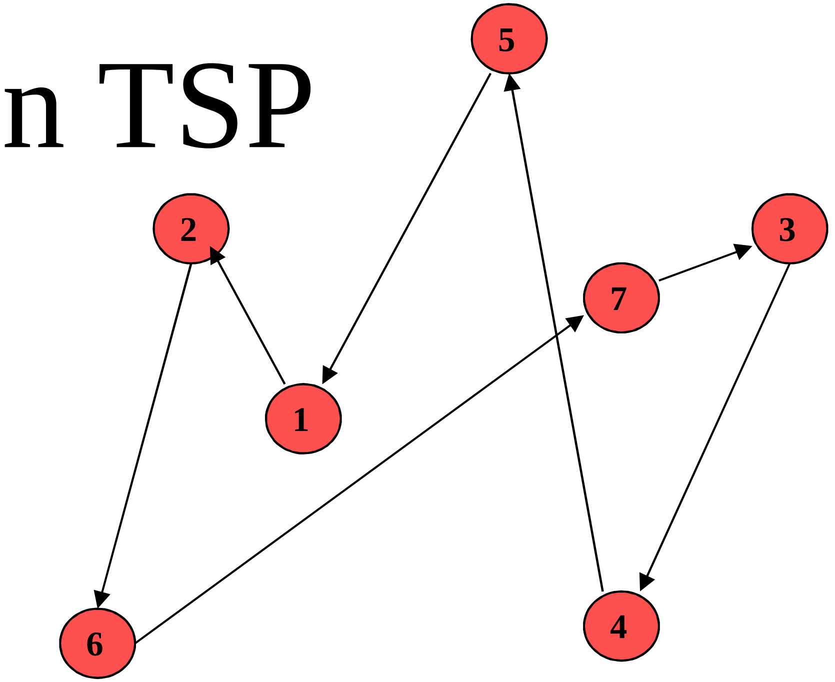
- The basic job of the tabu criterion is thus to avoid visiting the same solution more than once
- How to accomplish this?
  - Store all the solutions visited during the search, and check that the new solution is not among those previously visited
    - Too time consuming!
  - Find some way of (approximately) represent those solutions that we have seen most recently, and avoid returning immediately to those (or similar) solutions

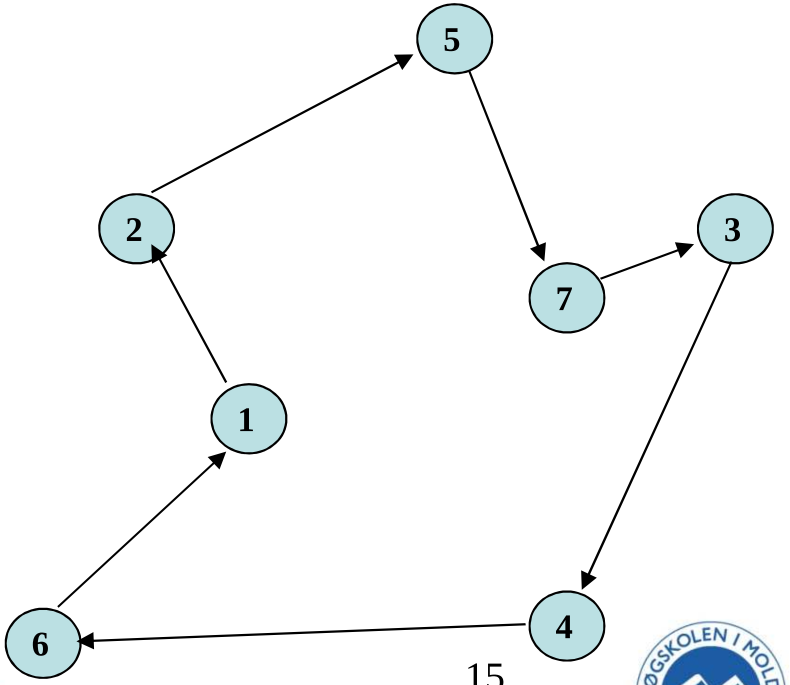
Tabu Attribute Selection

- Attribute
  - A property of a solution or a move
- Can be based on any aspect of the solution that are changed by a move
- Attributes are the basis for tabu restrictions
  - We use them to represent the solutions visited recently
- A move can change more than one attribute
  - e.g. a 2-opt move in TSP involves 4 cities and 4 edges
- Similar to the “features” in GLS, but we don’t require the attributes to have costs

Example – Attributes in TSP

- Attributes based on the edges
  - A1: Edges added to the tour
  - A2: Edges removed from the tour
- Move of type exchange
  - Exchanges two cities
  - 4 edges removed
  - 4 edges added
  - Exchange(5,6)
    - A1:(2,5),(5,7),(4,6),(6,1)
    - A2:(2,6),(6,7),(4,5),(5,1)





TS – Tabu Criterion

- The tabu criterion is defined on selected attributes of a move, (or the resulting solution if the move is selected)
- It is very often a component of the solution
- The attribute is tabu for a certain amount of time (i.e. iterations)
  - This is called the *Tabu Tenure* (TT)
- The tabu criterion usually avoids the immediate move reversal (or repetition)
- It also avoids the other (later) moves containing the tabu attribute. This cuts off a much larger part of the search space

TS – Attributes and Tabu Criteria

- Can have several tabu criteria on different attributes, each with its own tabu tenure
  - These can be disjunct
- If a move is to exchange a component (e.g. *edge*) *in* the solution with a component *not in* the solution, we can have the following tabu attributes and criteria
  - Edge added
  - Edge dropped
  - Edge added or edge dropped
  - Edge added and edge dropped

Use of Attributes in Tabu Restrictions

- Assume that the move from $s_k \longrightarrow s_{k+1}$ involves the attribute A
- The usual tabu restriction:
  - – Do not allow moves that reverse the status for A
- The TSP example:
  - – Move: exchange cities 2 and 5: $x_{2,5}$
  - – The tabu criterion could disallow:
    - Moves involving 2 and 5
    - Moves involving 2 or 5
    - Moves involving 2
    - Moves involving 5

Tabu Tenure (1)

- The tabu criterion will disallow moves that change back the value of some attribute(s)
- For how long do we need to enforce this rule?
  - For ever: the search stops because no changes are allowed
  - For too long: the search might become too limited (too much of the search space is cut off due to the tabu criterion)
  - For too short: the search will still cycle, but the length of the cycle can be more than 2
- The number of iterations for which the value of the attribute remains tabu is called the *Tabu Tenure*

Tabu Tenure (2)

- Earlier: The magical number 7, plus or minus 2
- Sometimes: in relation to problem size: $n^{1/2}$
- Static (fixed) tabu tenure is not recommended
  - The search gets more easily stuck in loops
- Dynamic tabu tenure is highly recommended
  - Change the tabu tenure at certain intervals
  - Can use uniform random selection in $[tt_1, tt_2]$
    - This is usually called dynamic, even though it is not
- Reactive Tabu Search
  - Detect stagnation → increase TT
  - When escaped → reduce TT

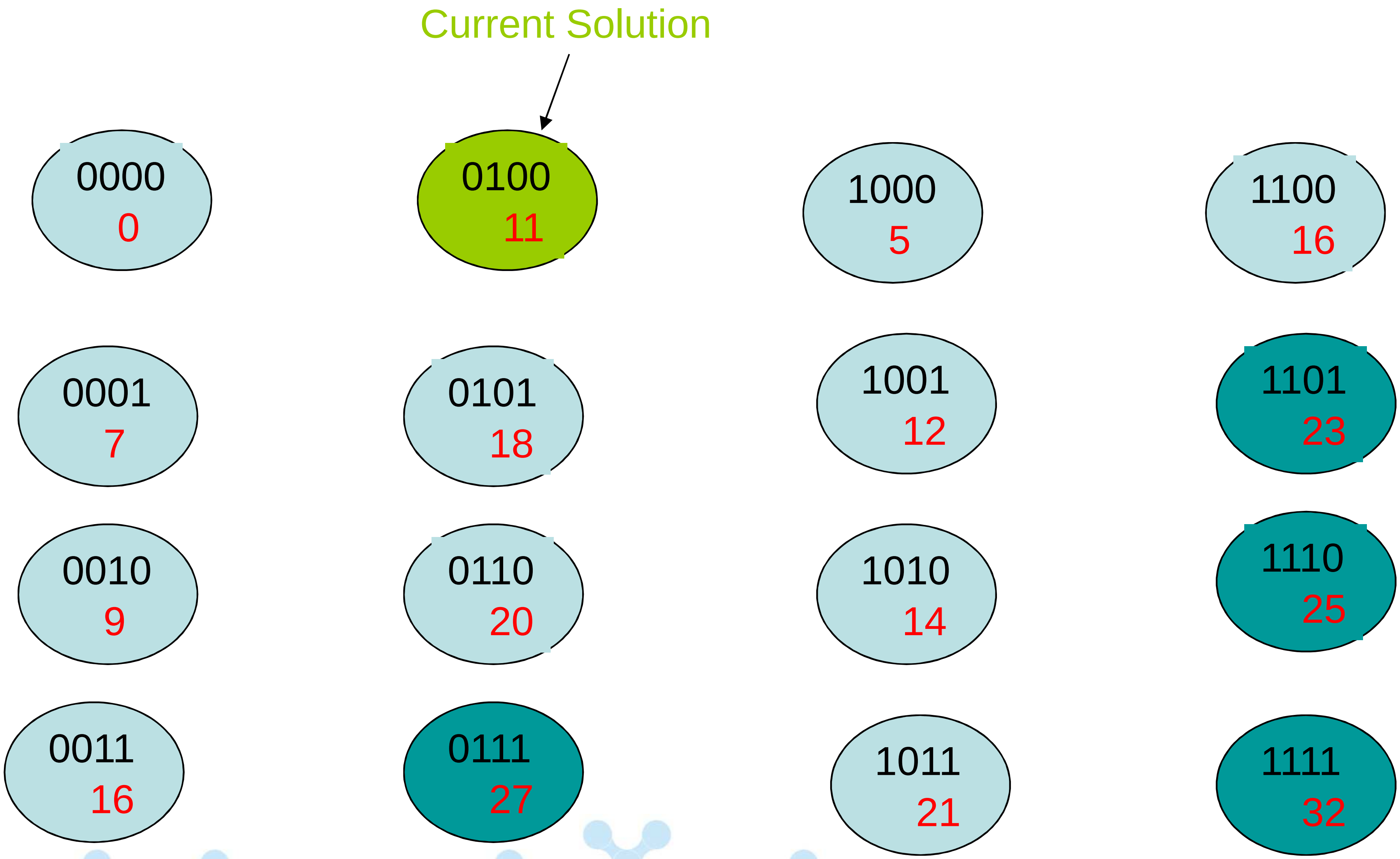
Tabu Tenure (3)

- Dependent on the tabu attributes
- Example: TSP – n cities – 2-opt
  - Use *edges-added* and *edges-dropped* as tabu attributes
  - $|n^2|$ edges in the problem instance
  - $|n|$ edges in the solution
  - Many more edges outside the solution than in the solution
  - Using the same TT would be unbalanced

Example: 0/1 Knapsack

- Flip-Neighborhood
- If the move is selecting an item to include in the solution, then any move trying to remove the same item is *tabu* for the duration of the *tabu tenure*
- Similarly, an item thrown out is not allowed in for the duration of the tabu tenure iterations
- Here the attribute is the same as the whole move

Flip Neighborhood



Current Solution

0000
0

0001
7

0010
9

0011
16

0100
11

0101
18

0110
20

0111
27

1000
5

1001
12

1010
14

1011
21

1100
16

1101
23

1110
25

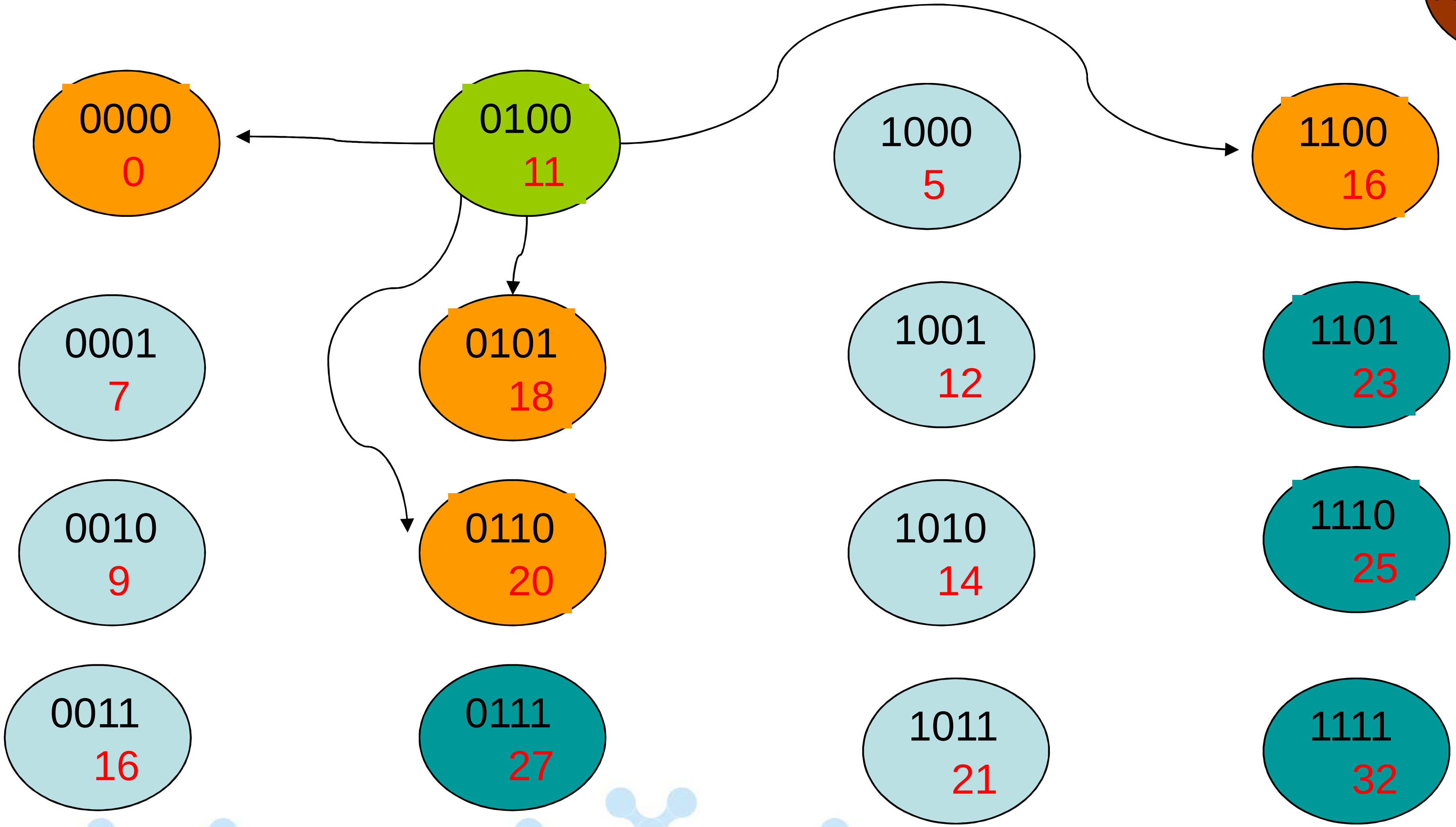
1111
32

Flip Neighborhood

Neighbor

Infeas. Neighbor

Variables flipped so far: none



0000 0

0100 11

1000 5

1100 16

0001 7

0101 18

1001 12

1101 23

0010 9

0110 20

1010 14

1110 25

0011 16

0111 27

1011 21

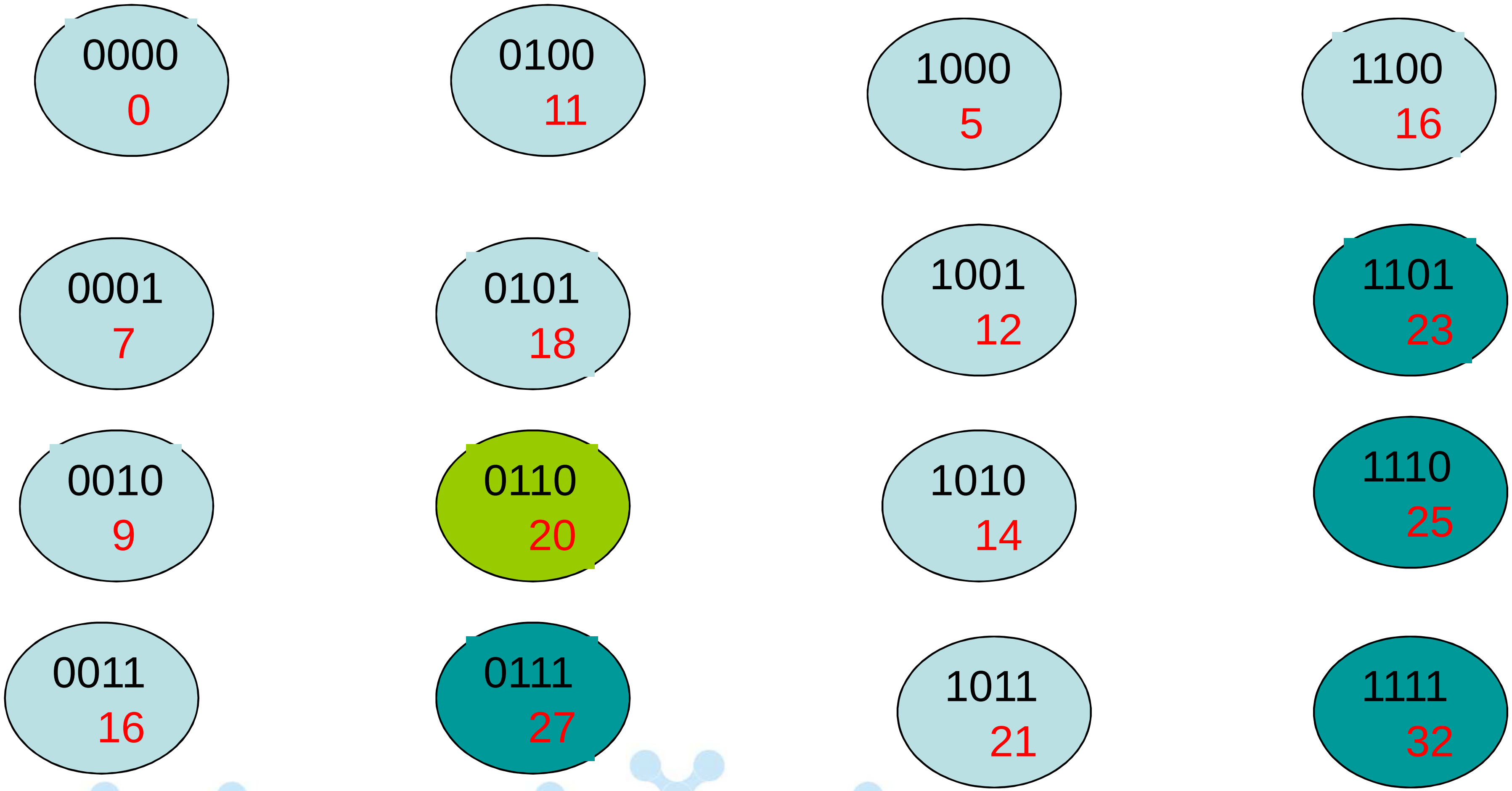
1111 32

Flip Neighborhood

Neighbor

Infeas. Neighbor

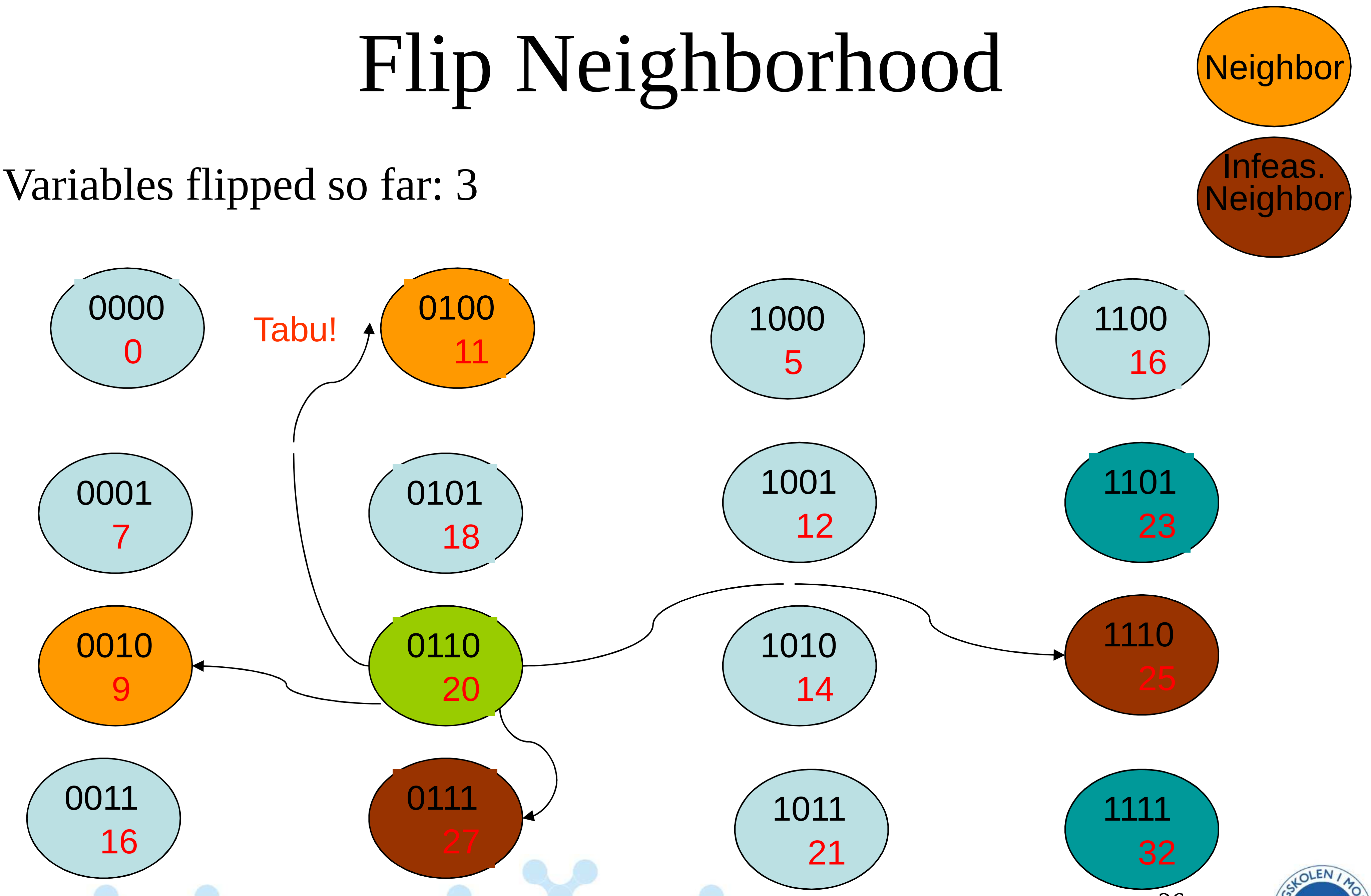
Variables flipped so far: 3



0000 0	0100 11	1000 5	1100 16
0001 7	0101 18	1001 12	1101 23
0010 9	0110 20	1010 14	1110 25
0011 16	0111 27	1011 21	1111 32

Flip Neighborhood

Variables flipped so far: 3

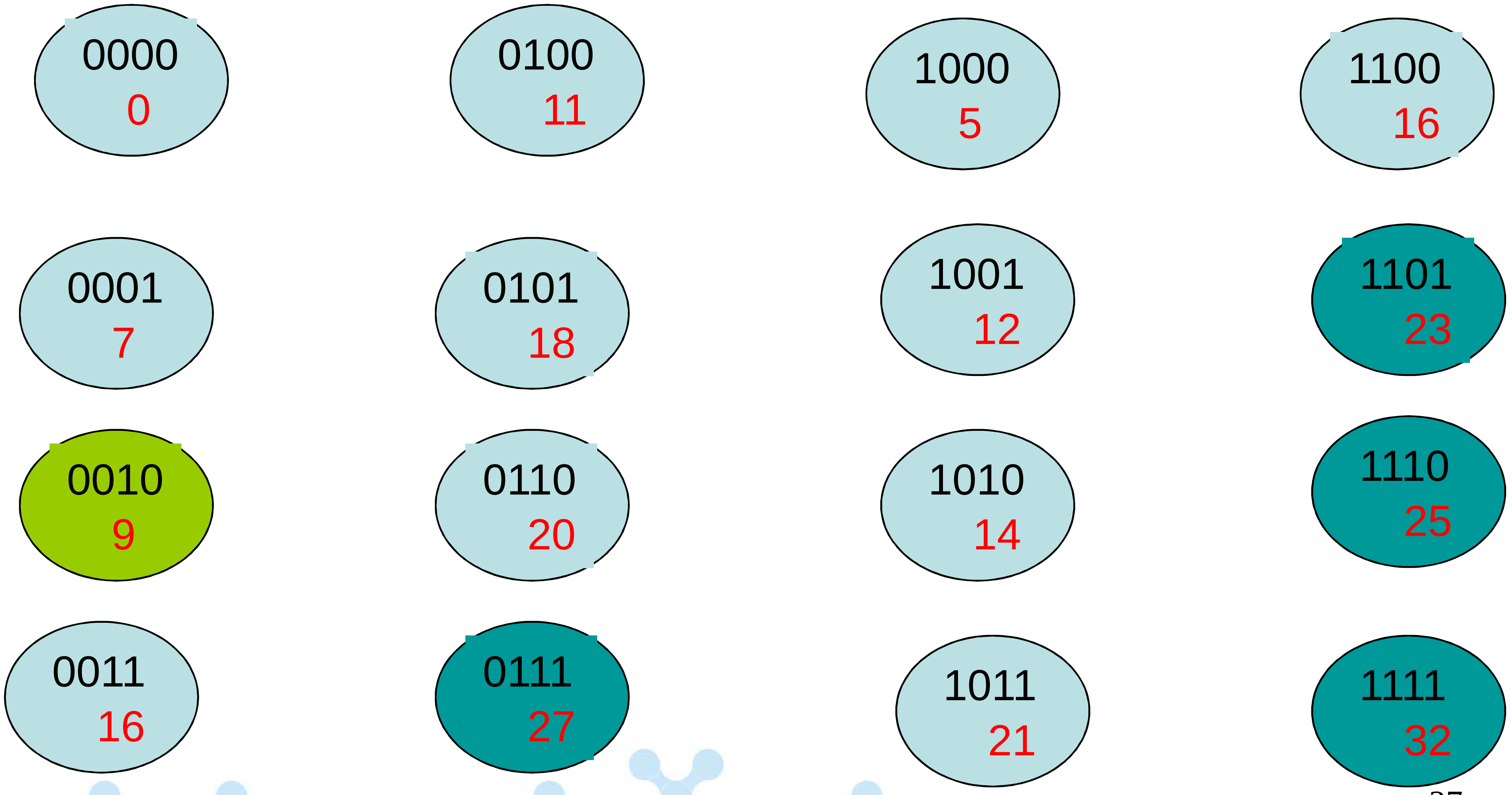


Flip Neighborhood

Neighbor

Infeas. Neighbor

Variables flipped so far: 3, 2



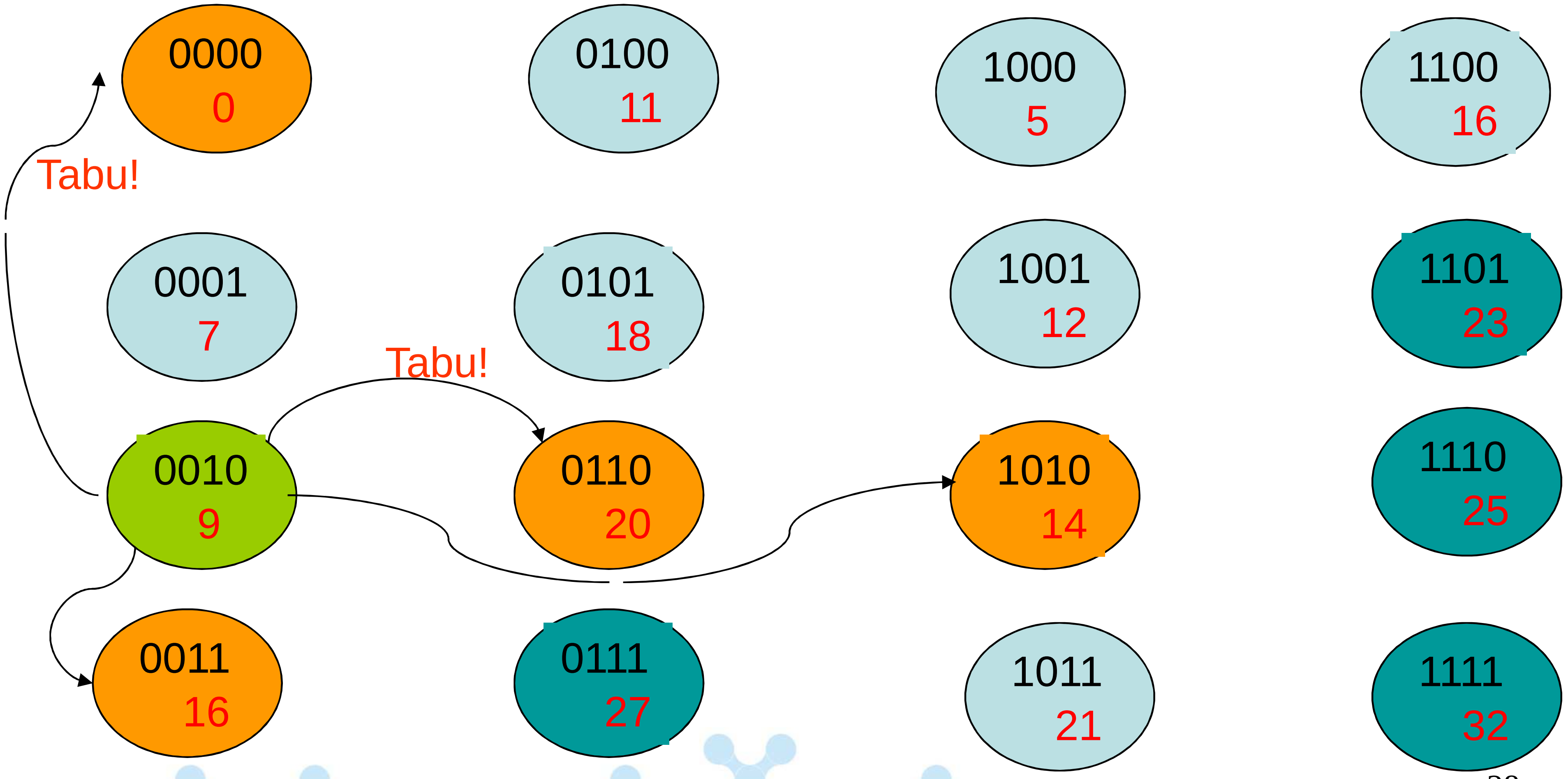
0000 0	0100 11	1000 5	1100 16
0001 7	0101 18	1001 12	1101 23
0010 9	0110 20	1010 14	1110 25
0011 16	0111 27	1011 21	1111 32

Flip Neighborhood

Neighbor

Infeas. Neighbor

Variables flipped so far: 3, 2



0000
0

Tabu!

0001
7

0010
9

0011
16

0100
11

0101
18

Tabu!

0110
20

0111
27

1000
5

1001
12

1010
14

1011
21

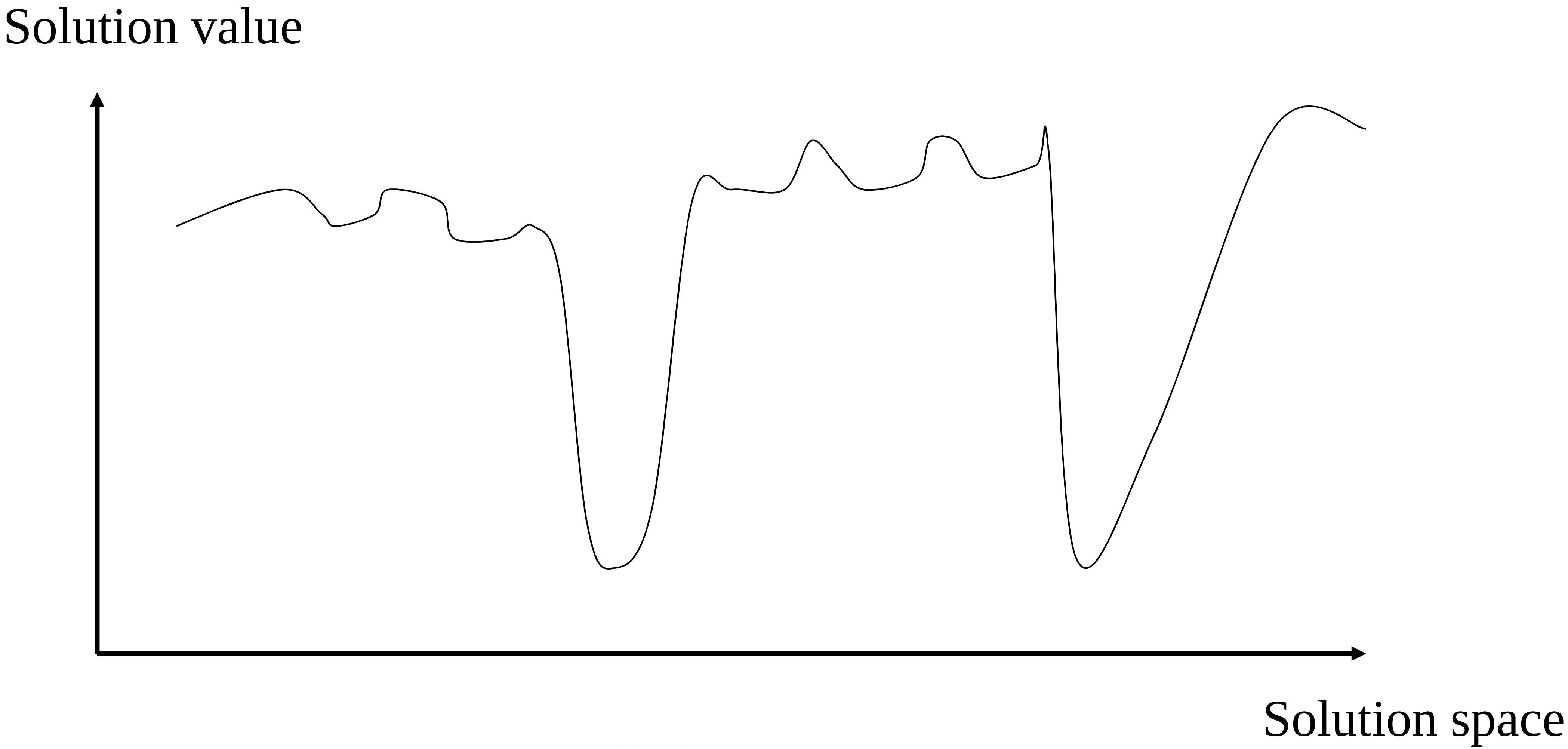
1100
16

1101
23

1110
25

1111
32

Local and Global optima



Aspiration Criterion (1)

- The tabu criterion is usually not exact
  - Some solutions that are not visited are nevertheless tabu for some time
- Possible problem: one of the neighbors is very good, but we cannot go there because some attribute is tabu
- Solution: if we somehow know that the solution is not visited before, we can allow ourselves to move there anyway
  - i.e., the solution is a new best solution: obviously we have not visited it before!

Aspiration Criterion (2)

- Simplest: Allow new best solutions, otherwise keep tabu status
- Criteria based on
  - Degree of feasibility
  - Degree of change
  - Feasibility level vs. Objective function value
  - Objective function value vs. Feasibility level
  - Distance between solutions
    - E.g. hamming distance
  - Influence of a move
    - The level of structural change in a solution
- If all moves are tabu:
  - Choose the best move, or choose randomly (in the candidate list)

Frequency Based Memory

- Complementary to the short term memory (tabu status)
- Used for long term strategies in the search
- Frequency counters
  - *residency*-based
  - *transition*-based
- TSP-example
  - how often has an edge been in the solution? (*residency*)
  - how often has the edge status been changed? (*transition*)

TS - Diversification

- Basic Tabu Search often gets stuck in one area of the search space
- Diversification is trying to get to somewhere else
- Historically random restarts have been very popular
- Frequency-based diversification tries to be more clever
  - penalize elements of the solution that have appeared in many other solutions visited

TS - Intensification

- To aggressively prioritize good solution attributes in a new solution
- Usually based on frequency
- Can be based on elite solutions, or part of them (vocabularies)

Intensification and Diversification

- Intensification
  - Aggressively prioritize attributes of good solutions in a new solution
    - Short term: based directly on the attributes
    - Longer term: use of elite solutions, or parts of elite solutions (vocabulary building)
- Diversification
  - The active spreading of the search, by actively prioritizing moves that gives solutions with new composition of attributes

Intensification and Diversification - simple mechanisms

- Use of frequency-based memory
- Based on a subset S_f of all the solutions visited (or moves executed)
- Diversification:
  - Choose S_f to contain a large part of the generated solutions (e.g. all the local optima)
- Intensification:
  - Choose S_f to be a small subset of *elite* solutions
    - E.g., that have overlapping attributes
  - Can have several such subset
    - Partitioning, clustering-analysis

Whips and Carrots

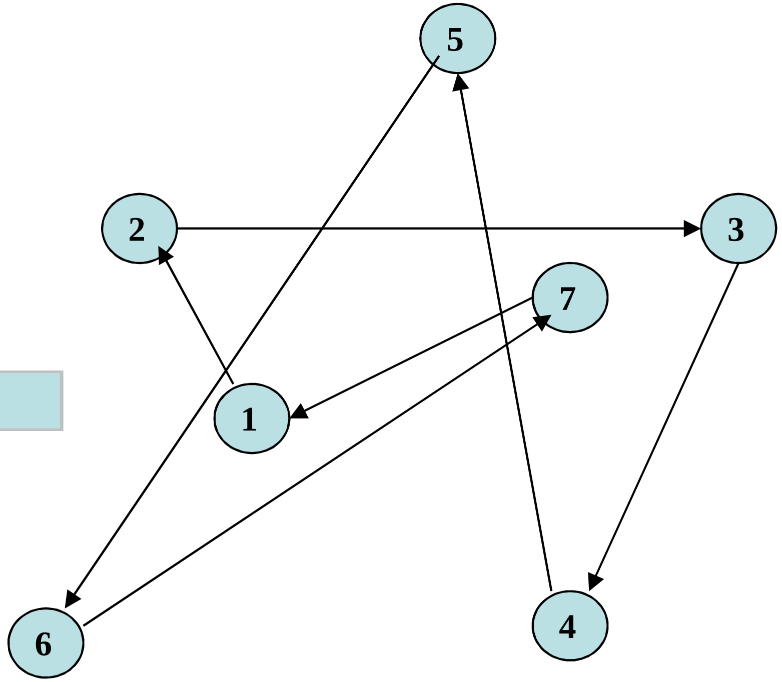
- Used in the move evaluation function, in addition to the change in the objective function value and tabu status
- A carrot for intensification will be a whip for diversification
- Diversification:
  - Moves containing attributes with a high frequency count are penalized
  - TSP-example: $g(x)=f(x)+w_1\Sigma\omega_{ij}$
- Intensification:
  - Moves to solutions containing attributes with a high frequency among the elite solutions are encouraged
  - TSP-example: $g(x)=f(x)-w_2\Sigma\gamma_{ij}$

TS Example: TSP

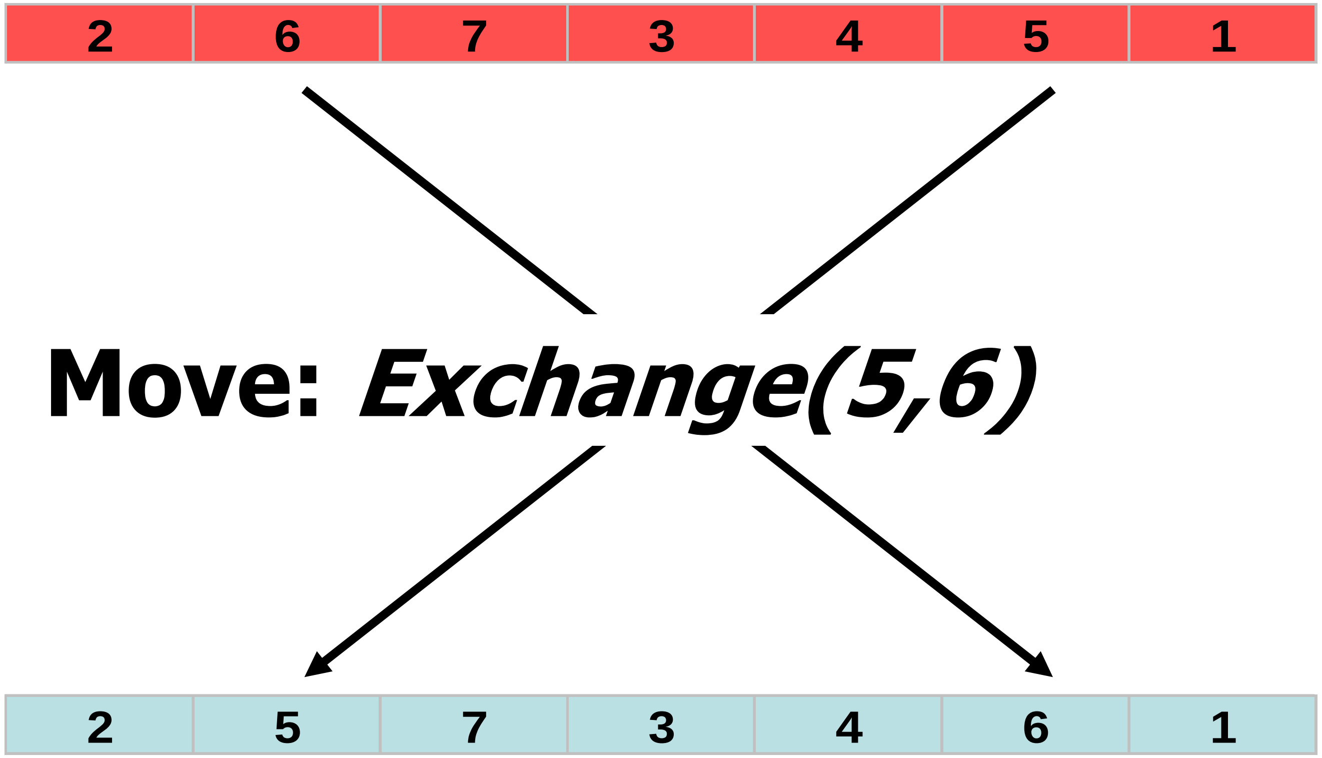
- Representation: permutation vector
- Move: pairwise exchange

$$(i, j) \quad i < j \quad i, j \in [1, n]$$

1	2	3	4	5	6	7



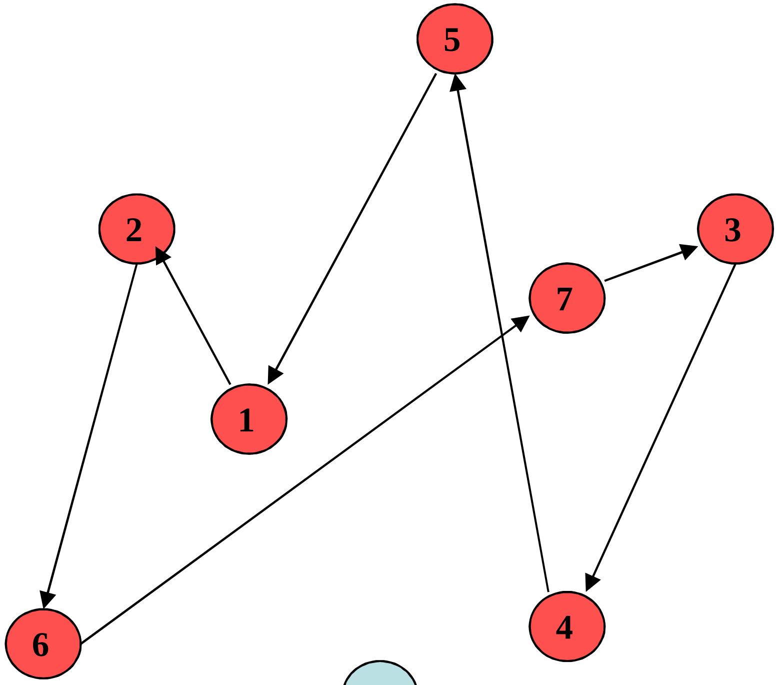
Move: Exchange in permutation vector

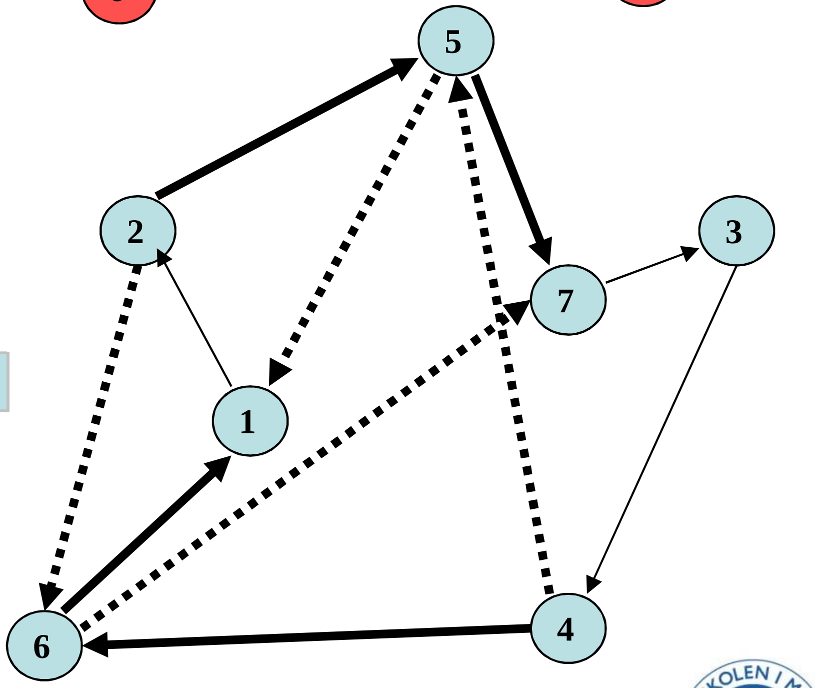


2	6	7	3	4	5	1

Move: ***Exchange(5,6)***

2	5	7	3	4	6	1





TSP Example

- Number of neighbors: $\binom{n}{2}$

- For every neighbor: *Move value*

$$\Delta_{k+1} = f(i_{k+1}) - f(i_k), \qquad i_{k+1} \in N(i_k)$$

- Choice of tabu criterion
  - Attribute: cities involved in a move
  - Moves involving the same cities are tabu
  - Tabu tenure = 3 (fixed)
- Aspiration criterion
  - new best solution

TSP Example: Data structure

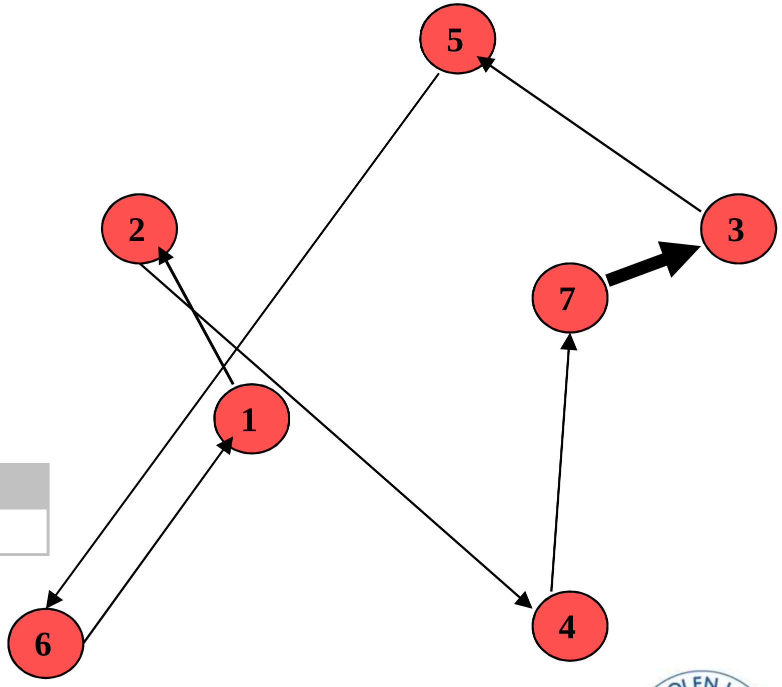
- Data structure: triangular table, storing the number of iterations until moves are legal
- Updated for every move

	2	3	4	5	6	7
1	0	2	0	0	0	0
	2	0	3	0	0	0
		3	0	0	0	0
			4	1	0	0
				5	0	0
					6	0

TSP Example: Tabu Criteria/Attributes

- Illegal to operate on given cities
- Illegal to change the city in position *k* in the vector
- Criteria on edges
  - Links often present in good solutions
  - Length of links w.r.t. the average
- For permutation problems
  - Attributes related to previous/next often work well

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



TSP Example: Iteration 0

Starting solution: Value = 234

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

Tabu list:

	2	3	4	5	6	7
1	0	0	0	0	0	0
	2	0	0	0	0	0
		3	0	0	0	0
			4	0	0	0
				5	0	0
					6	0

TSP Example: Iteration 1

Current solution: Value = 234

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

Candidate list:

Exchange	Value
5.4	-34
7.4	-4
3.6	-2
2.3	0
4.1	4

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

After move: Value = 200

Tabu list:

	2	3	4	5	6	7
1	0	0	0	0	0	0
	2	0	0	0	0	0
		3	0	0	0	0
			4	3	0	0
				5	0	0
					6	0

TSP Example: Iteration 2

Current solution: Value = 200

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

Candidate list:

Exchange	Value
3.1	-2
2.3	-1
3.6	1
7.1	2
6.1	4

← Choose move (3,1)

Tabu list:

	2	3	4	5	6	7
1	0	0	0	0	0	0
	2	0	0	0	0	0
		3	0	0	0	0
			4	3	0	0
				5	0	0
					6	0

TSP Example: Iteration 2

Current solution: Value = 200

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

Candidate list:

Exchange	Value
3.1	-2
2.3	-1
3.6	1
7.1	2
6.1	4

Choose move (3,1)

Update tabu list

Tabu list:

	2	3	4	5	6	7
1	0	3	0	0	0	0
	2	0	0	0	0	0
		3	0	0	0	0
			4	2	0	0
				5	0	0
					6	0

TSP Example: Iteration 3

Current solution: Value = 198

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

Candidate list:

Exchange	Value
1.3	2
2.4	4
7.6	6
4.5	7
5.3	9

Tabu!

Choose move (2,4)

NB: Worsening move!

Tabu list:

	2	3	4	5	6	7
1	0	3	0	0	0	0
	2	0	0	0	0	0
		3	0	0	0	0
			4	2	0	0
				5	0	0
					6	0

TSP Example: Iteration 3

Current solution: Value = 198

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

Candidate list:

Exchange	Value
1.3	2
2.4	4
7.6	6
4.5	7
5.3	9

Tabu!

Choose move (2,4)

NB: Worsening move!

Tabu list:

	2	3	4	5	6	7
1	0	2	0	0	0	0
	2	0	3	0	0	0
		3	0	0	0	0
			4	1	0	0
				5	0	0
					6	0

Update tabu list

TSP Example: Iteration 4

Current solution: Value = 202

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

Candidate list:

Exchange	Value
4.5	-6
5.3	-2
7.1	0
1.3	3
2.6	6

Tabu!

Choose move (4,5)

Aspiration!

Tabu list:

	2	3	4	5	6	7
1	0	2	0	0	0	0
	2	0	3	0	0	0
		3	0	0	0	0
			4	1	0	0
				5	0	0
					6	0

Observations

- In the example 3 out of 21 moves are prohibited
- More restrictive tabu effect can be achieved by
  - Increasing the tabu tenure
  - Using stronger tabu-restrictions
    - Using OR instead of AND for the 2 cities in a move

TSP Example: Frequency Based Long Term Memory

- Typically used to diversify the search
- Can be activated after a period with no improvement
- Often penalize attributes of moves that have been selected often

Tabu-status (closeness in time)

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

Frequency of moves

Summary

- Introduction to Tabu Search
  - Tabu criterion
    - Tabu Tenure
  - Aspiration
  - Examples of use
    - Knapsack
    - TSP
  - Intensification
  - Diversification