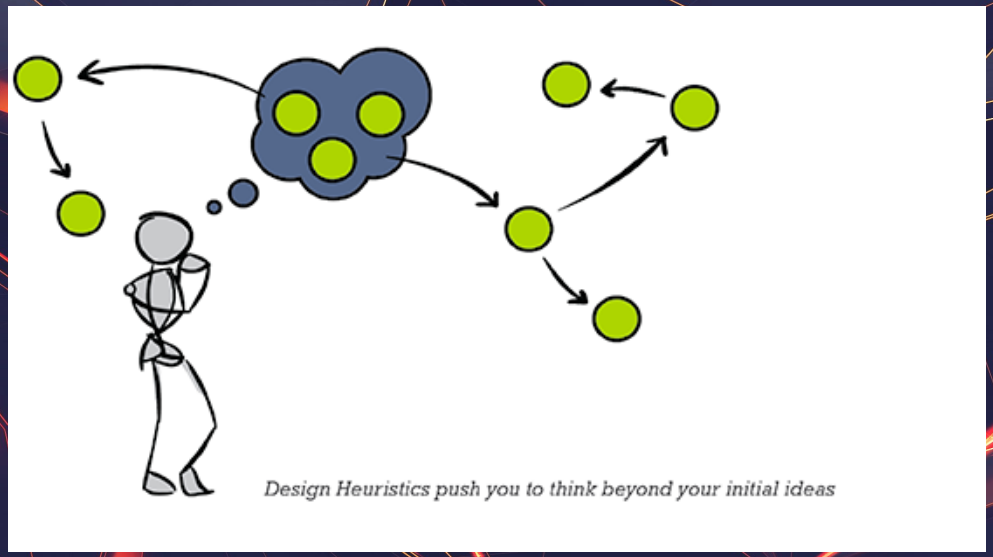


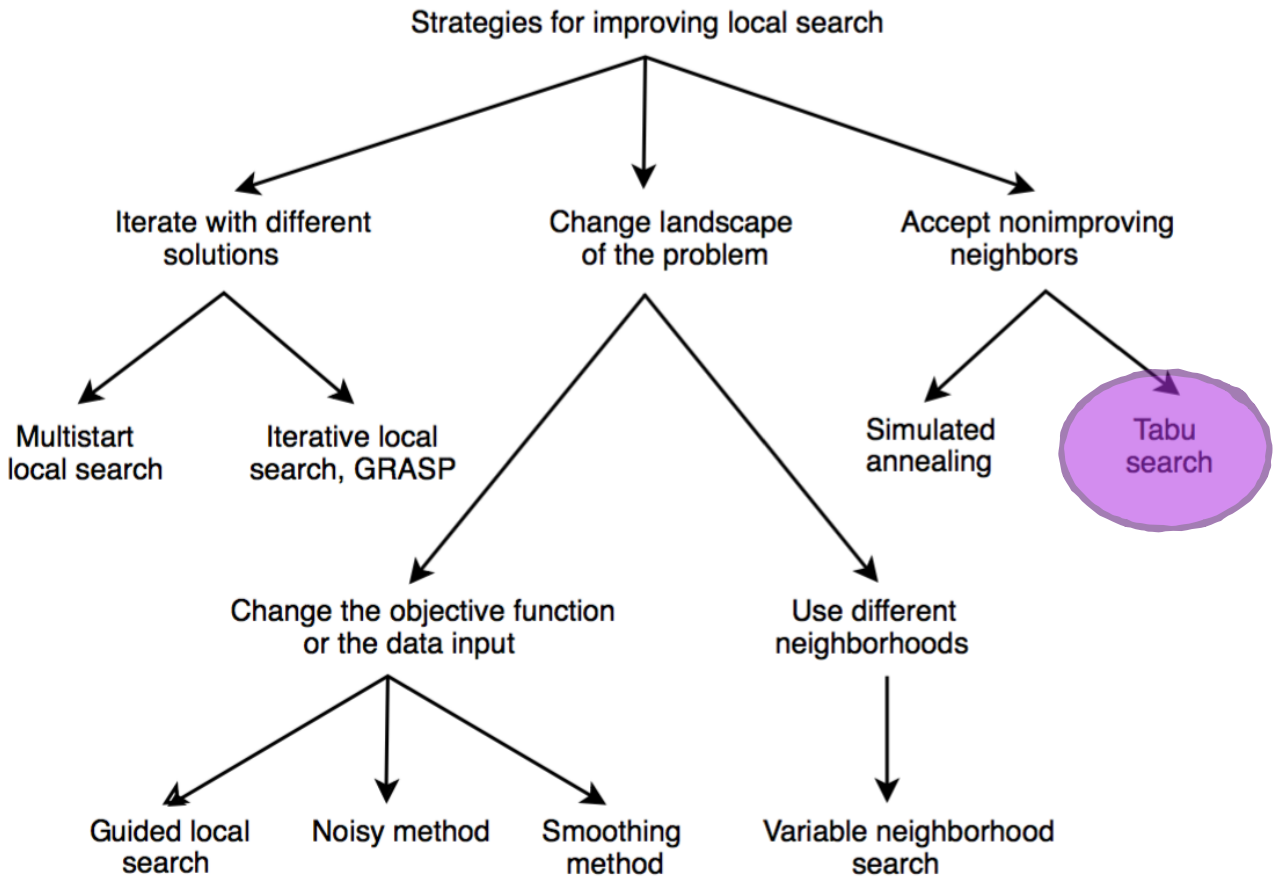
SINGLE SOLUTION METAHEURISTICS

Tabu Search

Prof. Eduardo Pécora

Thanks to Professor Manuel Laguna for providing me his slides!

ESCAPING FROM LOCAL OPTIMA



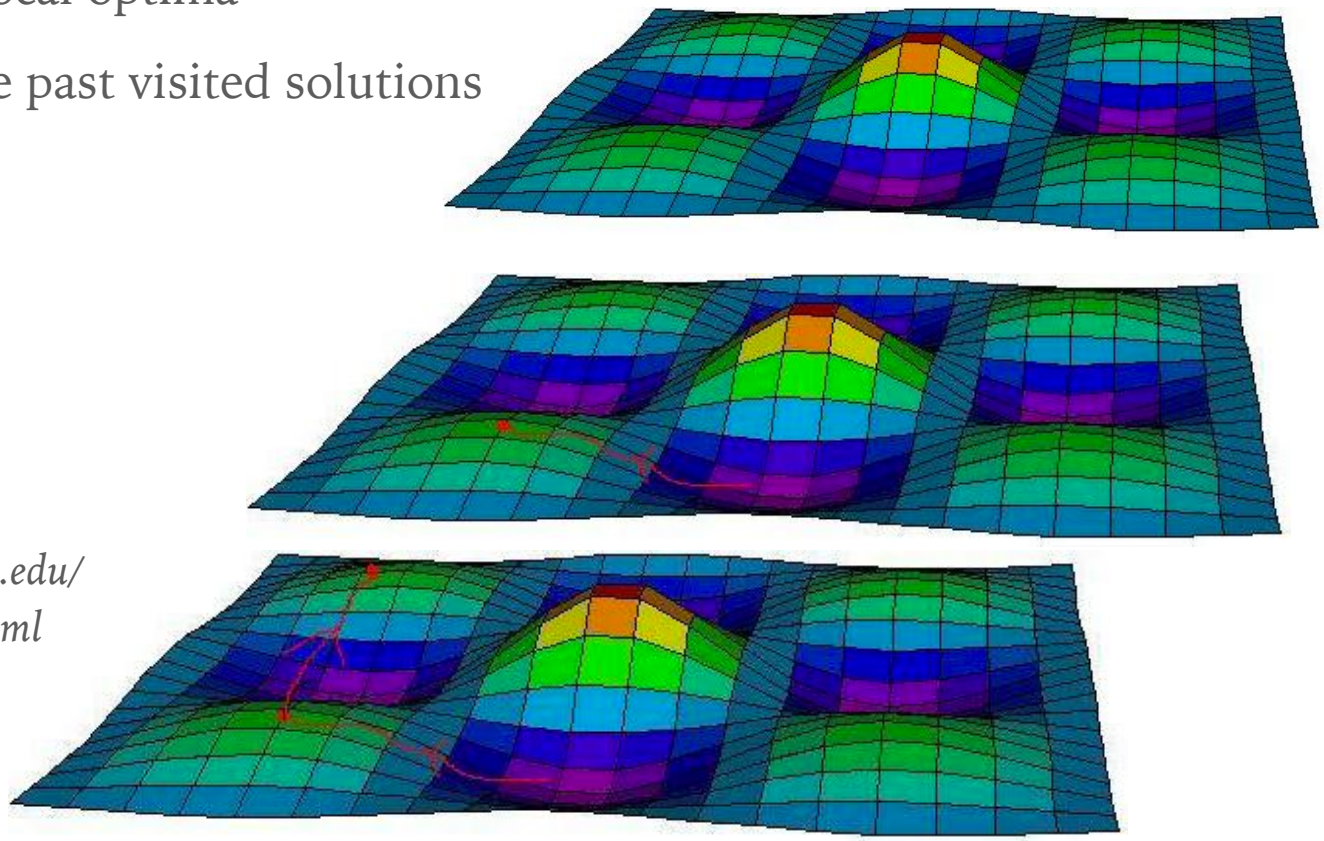
TABU SEARCH - TS

- TS - Tabu search algorithm was proposed by Fred Glover
- In the 1990s, the tabu search algorithm became very popular in solving optimization problems in an approximate manner. Nowadays, it is one of the most widespread S-metaheuristics. The use of **memory**, which stores information related to the search process, represents the particular feature of tabu search.
- 1997 Fred Glover published the seminal book of Tabu Search with Professor Manuel Laguna

F. Glover. Tabu search: Part I. ORSA Journal on Computing, 1(3):190–206, 1989

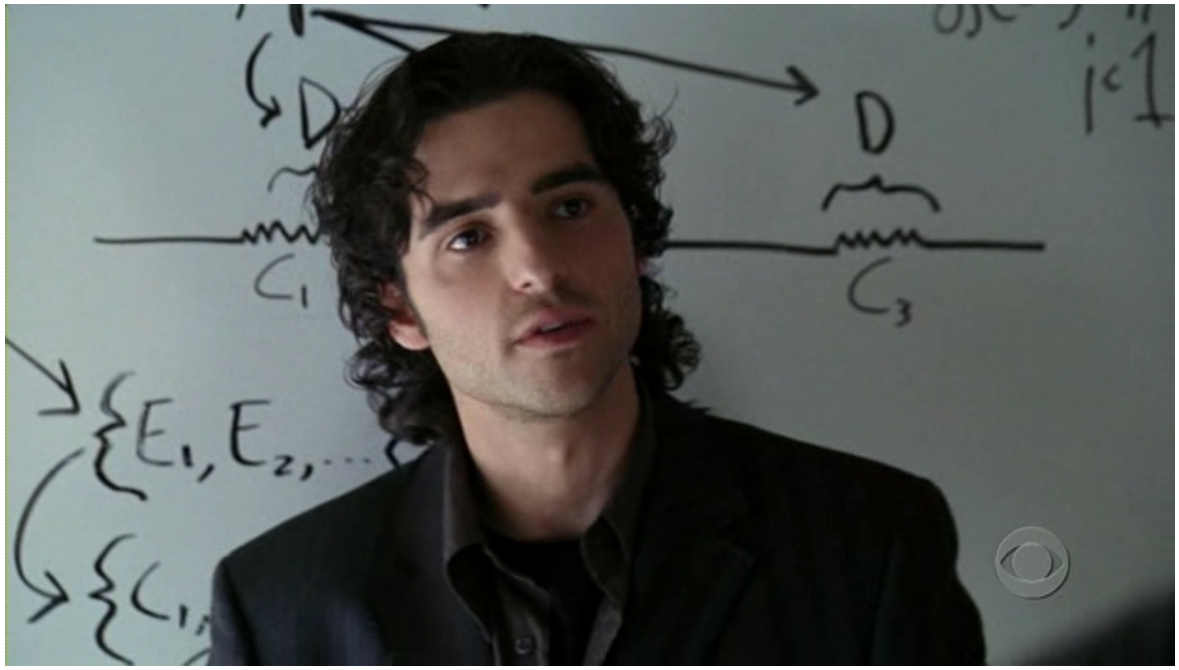
MEMORY

- Used to scape a local optima
- Keeps track of the past visited solutions



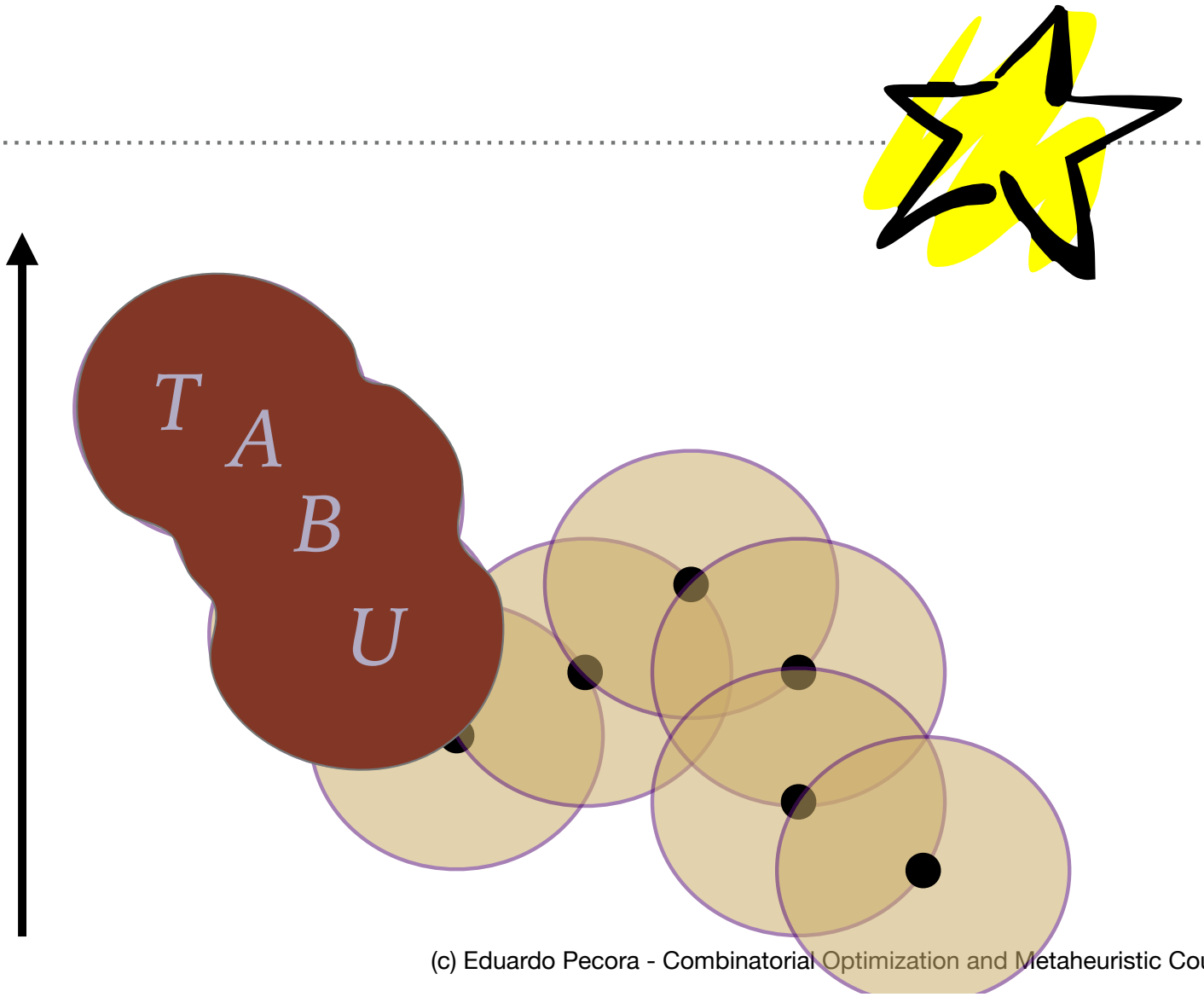
http://www.math.cornell.edu/~numb3rs/baker/408.html

NUMB3RS



Prof. Charles Edward "Charlie" Eppes, used the TABU SEARCH algorithm in Season 4 Episode 08 to find kidnappers.

TABU SEARCH



Basic Concepts

- Solution
  - Initial
  - Current
  - Best
- Move
  - Attributes
  - Value
- Neighborhood
  - Original
  - Modified (Reduced or Expanded)
- Tabu
  - Status
  - Activation rules

Short-Term Memory

» The main goal of the STM is to avoid reversal of moves and cycling

» The most common implementation of the STM is based on move attributes and the recency of the moves

Example 1

- After a move that changes the value of x_i from 0 to 1, we would like to prevent x_i from taking the value of 0 in the next *TabuTenure* iterations

  – Attribute to record: i
  – Tabu activation rule: move ($x_i \leftarrow 0$) is tabu if i is tabu-active

Example 2

- After a move that exchanges the positions of element i and j in a sequence, we would like to prevent elements i and j from exchanging positions in the next *TabuTenure* iterations

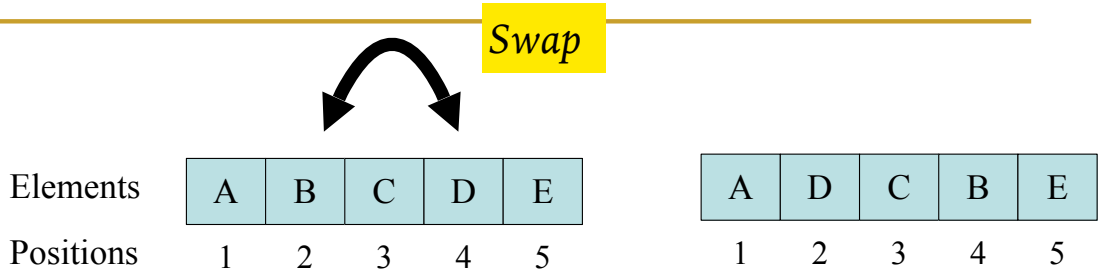
  – Attributes to record: *i* and *j*
  – Tabu activation rule: move ($i \leftrightarrow j$) is tabu if both *i* and *j* are tabu-active

Example 3

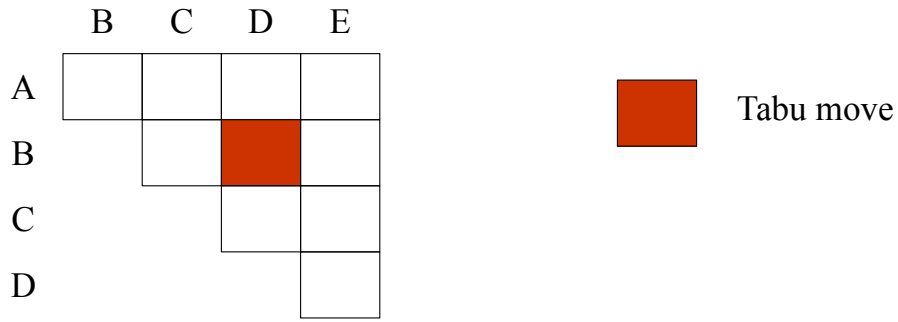
- After a move that drops element i from and adds element j to the current solution, we would like to prevent element i from being added to the solution in the next *TabuAddTenure* iterations and prevent element j from being dropped from the solution in the next *TabuDropTenure* iterations

  – Attributes to record: i and j
  – Tabu activation rules:
    - move (Add i) is tabu if i is tabu-active
    - move (Drop j) is tabu if j is tabu-active

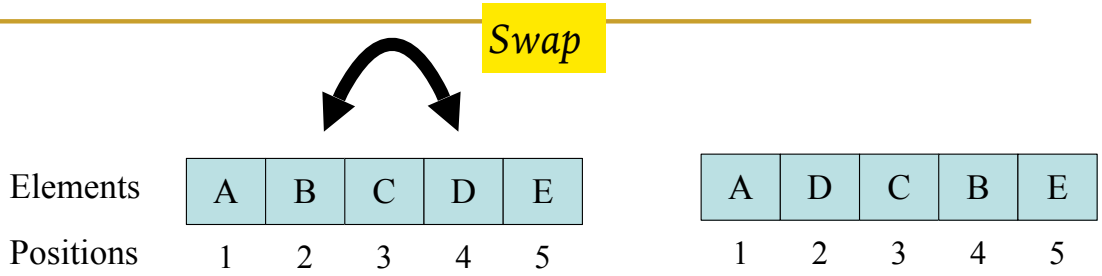
Example 4.3



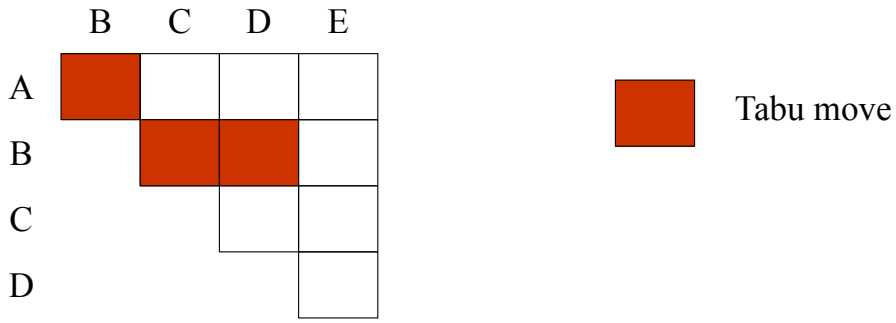
Tabu activation rule: move (B ↔ D) is tabu



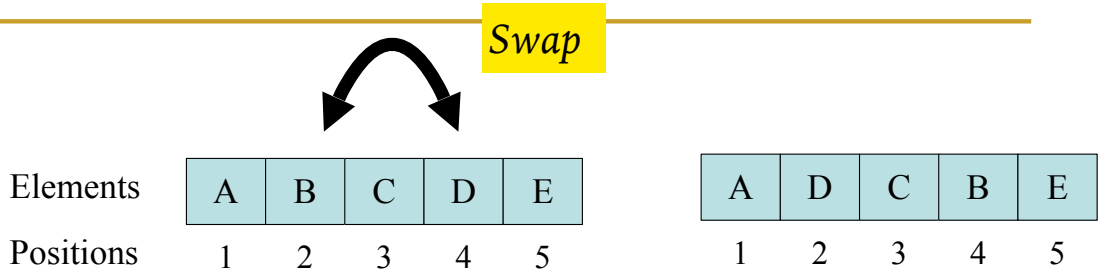
Example 4.2



Tabu activation rule: move (B ↔ *) is tabu if B moves to 3 or earlier

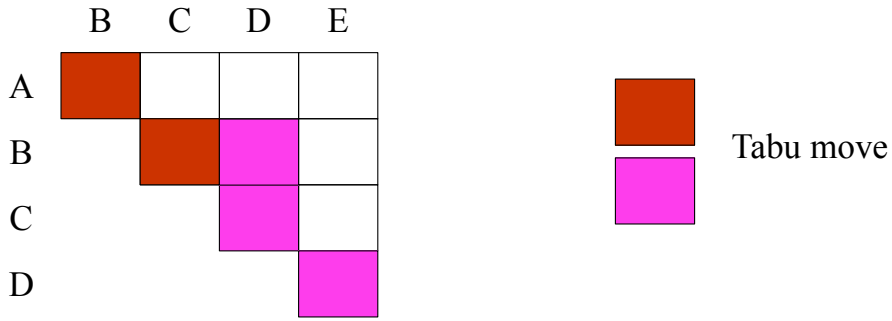


Example 4.2

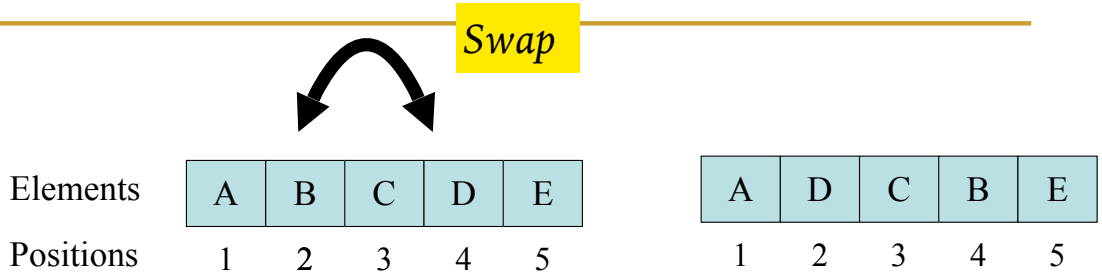


Tabu activation rule: move (B ↔ *) is tabu if B moves to 3 or earlier

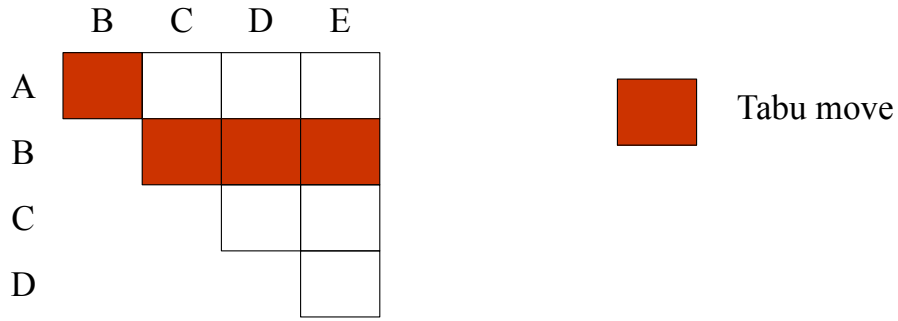
move (D ↔ *) is tabu if D moves to 3 or later



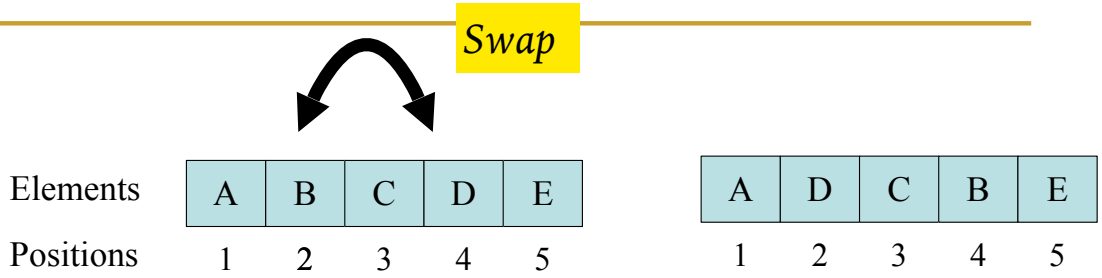
Example 4.1



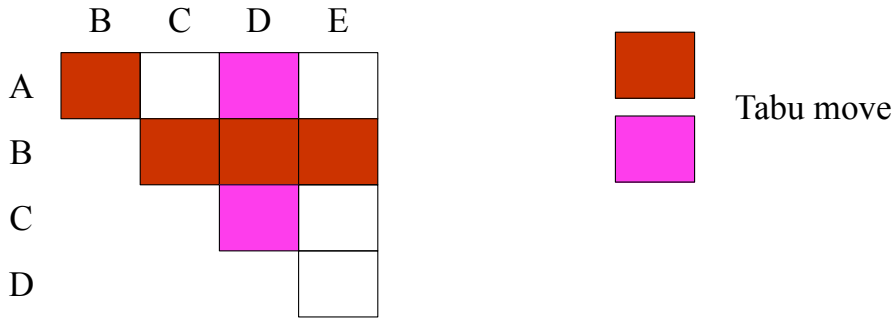
Tabu activation rule: move (B ↔ *) is tabu



Example 4.1



Tabu activation rule: move (B ↔ *) is tabu
move (D ↔ *) is tabu



EXAMPLE: KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	0	0	0	0	0	0	0	0	0		
C1	1	3	6	7	9	1	0	0	0	<=	17
C2	4	9	6	9	2	0	1	4	0	<=	30
C3	6	8	3	9	6	8	9	3	0	<=	45

Neighbourhood - Change 1

OBJ	0	0	0	0	1	0	0	0	8		
C1	1	3	6	7	9	1	0	0	9	<=	17
C2	4	9	6	9	2	0	1	4	2	<=	30
C3	6	8	3	9	6	8	9	3	6	<=	45

Tabu List - STM

VALUE 1								
VALUE 0					5			

LTM

	0	1
X1	1	
X2	1	
X3	1	
X4	1	
X5		1
X6	1	
X7	1	
X8	1	

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	0	0	0	0	1	0	0	0	8		
C1	1	3	6	7	9	1	0	0	9	<=	17
C2	4	9	6	9	2	0	1	4	2	<=	30
C3	6	8	3	9	6	8	9	3	6	<=	45

LTM

	0	1
X1	2	
X2	1	1
X3	2	
X4	2	
X5		2
X6	2	
X7	2	
X8	2	

Neighbourhood - Change 1

OBJ	0	1	0	0	1	0	0	0	13		
C1	1	3	6	7	9	1	0	0	12	<=	17
C2	4	9	6	9	2	0	1	4	11	<=	30
C3	6	8	3	9	6	8	9	3	14	<=	45

Tabu List - STM

VALUE 1								
VALUE 0		5			4			

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	0	1	0	0	1	0	0	0	13		
C1	1	3	6	7	9	1	0	0	12	<=	17
C2	4	9	6	9	2	0	1	4	11	<=	30
C3	6	8	3	9	6	8	9	3	14	<=	45

Neighbourhood - Change 1

OBJ	0	1	0	1	1	0	0	0	17		
C1	1	3	6	7	9	1	0	0	19	<=	17
C2	4	9	6	9	2	0	1	4	20	<=	30
C3	6	8	3	9	6	8	9	3	23	<=	45

Infeasible

Tabu List - STM

VALUE 1								
VALUE 0		4		5	3			

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	0	1	0	0	1	0	0	0	13		
C1	1	3	6	7	9	1	0	0	12	<=	17
C2	4	9	6	9	2	0	1	4	11	<=	30
C3	6	8	3	9	6	8	9	3	14	<=	45

Neighbourhood - Change 1

OBJ	0	1	0	0	1	1	0	0	15		
C1	1	3	6	7	9	1	0	0	13	<=	17
C2	4	9	6	9	2	0	1	4	11	<=	30
C3	6	8	3	9	6	8	9	3	22	<=	45

Tabu List - STM

VALUE 1								
VALUE 0		4			3	5		

LTM

	0	1
X1	3	
X2	1	2
X3	3	
X4	3	
X5		3
X6	2	1
X7	3	
X8	3	

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	0	1	0	0	1	1	0	0	15		
C1	1	3	6	7	9	1	0	0	13	<=	17
C2	4	9	6	9	2	0	1	4	11	<=	30
C3	6	8	3	9	6	8	9	3	22	<=	45

Neighbourhood - Change 1

OBJ	1	1	0	0	1	1	0	0	18		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	15	<=	30
C3	6	8	3	9	6	8	9	3	28	<=	45

Tabu List - STM

VALUE 1								
VALUE 0	5	3			2	4		

LTM

	0	1
X1	3	1
X2	1	3
X3	4	
X4	4	
X5		4
X6	2	2
X7	4	
X8	4	

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	1	1	0	0	1	1	0	0	18		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	15	<=	30
C3	6	8	3	9	6	8	9	3	28	<=	45

Neighbourhood - Change 1

OBJ	1	1	0	0	1	1	1	0	19		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	16	<=	30
C3	6	8	3	9	6	8	9	3	37	<=	45

Tabu List - STM

VALUE 1								
VALUE 0	4	2			1	3	5	

LTM

	0	1
X1	3	2
X2	1	4
X3	5	
X4	5	
X5		5
X6	2	3
X7	4	1
X8	5	

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	1	1	0	0	1	1	1	0	19		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	16	<=	30
C3	6	8	3	9	6	8	9	3	37	<=	45

Neighbourhood - Change 1

OBJ	1	1	0	0	1	1	1	1	16		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	20	<=	30
C3	6	8	3	9	6	8	9	3	40	<=	45

Tabu List - STM

VALUE 1								
VALUE 0	3	1			0	2	4	5

LTM

	0	1
X1	3	3
X2	1	5
X3	6	
X4	6	
X5		6
X6	2	4
X7	4	2
X8	5	1

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	1	1	0	0	1	1	1	1	16		
C1	1	3	6	7	9	1	0	0	14	<=	17
C2	4	9	6	9	2	0	1	4	20	<=	30
C3	6	8	3	9	6	8	9	3	40	<=	45

Neighbourhood - Change 1

OBJ	1	1	0	0	0	1	1	1	8		
C1	1	3	6	7	9	1	0	0	5	<=	17
C2	4	9	6	9	2	0	1	4	18	<=	30
C3	6	8	3	9	6	8	9	3	34	<=	45

Tabu List - STM

VALUE 1					5			
VALUE 0	2	0				1	3	5

LTM

	0	1
X1	3	4
X2	1	6
X3	7	
X4	7	
X5	1	6
X6	2	5
X7	4	3
X8	5	2

EXAMPLE KNAPSACK

	X1	X2	X3	X4	X5	X6	X7	X8			
OBJ	1	1	0	0	0	1	1	1	8		
C1	1	3	6	7	9	1	0	0	5	<=	17
C2	4	9	6	9	2	0	1	4	18	<=	30
C3	6	8	3	9	6	8	9	3	34	<=	45

Neighbourhood - Change 1

OBJ	1	1	0	1	0	1	1	1	12		
C1	1	3	6	7	9	1	0	0	12	<=	17
C2	4	9	6	9	2	0	1	4	27	<=	30
C3	6	8	3	9	6	8	9	3	43	<=	45

Tabu List - STM

VALUE 1					4			
VALUE 0	1			5		0	2	4

LTM

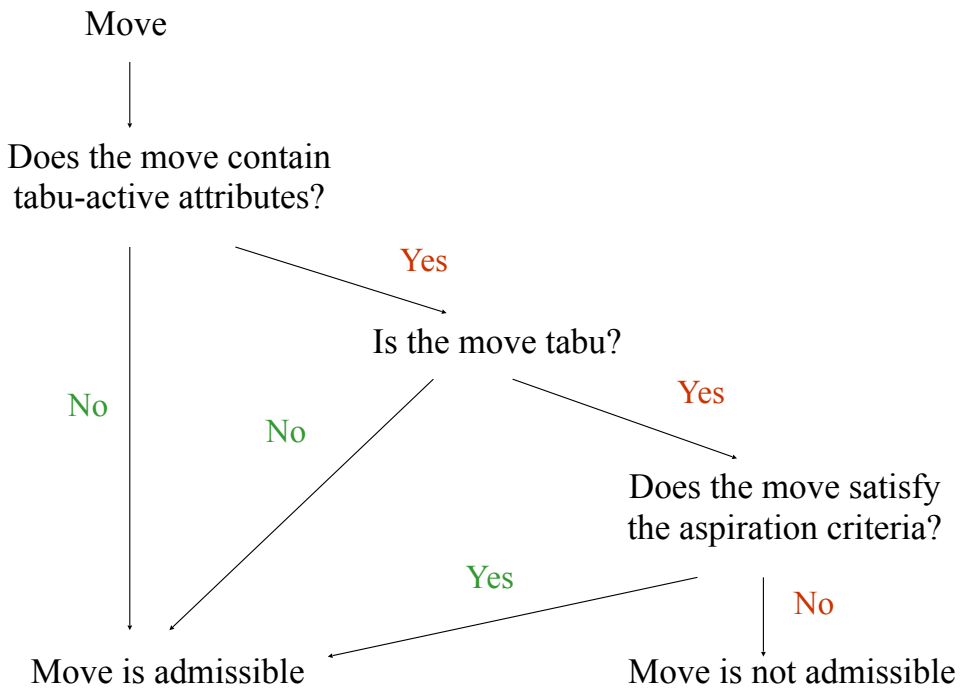
	0	1
X1	3	5
X2	1	7
X3	8	
X4		1
X5	2	6
X6	2	6
X7	4	4
X8	5	3

Frequency-based Memory

- Transition Measure
  - Number of iterations where an attribute has been changed (e.g., added or deleted from a solution)

- Residence Measure
  - Number of iterations where an attribute has stayed in a particular position (e.g., belonging to the current solution)

Tabu Decision Tree



Tabu Tenure Management

- Static Memory
  - The value of *TabuTenure* is fixed and remains fixed during the entire search
  - All attributes remain tabu-active for the same number of iterations

- Dynamic Memory
  - The value of *TabuTenure* is not constant during the search
  - The length of the tabu-active status of attributes varies during the search

Aspiration Criteria

- By Objective
  - A tabu move becomes admissible if it yields a solution that is better than an aspiration value
- By Search Direction
  - A tabu move becomes admissible if the direction of the search (improving or non-improving) does not change

Example 8

- Transition Measure
  - Number of times that element i has been moved to an earlier position in the sequence sequence

- Residence Measure
  - Number of times that element i has occupied position k

Modifying Choice Rules

- Frequency-based memory is typically used to modify rules for …
  - choosing the best move to make on a given iteration
  - choosing the next element to add to a restarting solution

- The modification is based on penalty functions

Modifying Move Values for Diversification

- Rule
  - Choose the move with the best move value if at least one admissible improving move exists
  - Otherwise, choose the admissible move with the best modified move value

Modified move value =

*Move value – Diversification parameter * F(frequency measure)*

Example 9

- The frequency of elements occupying certain positions can be used to bias a construction procedure and generate new restarting points

- For instance, due dates can be modified with frequency information (of jobs finishing on time) before reapplying the EDD rule

Strategic Oscillation

- Strategic oscillation operates by orienting moves in relation to a boundary

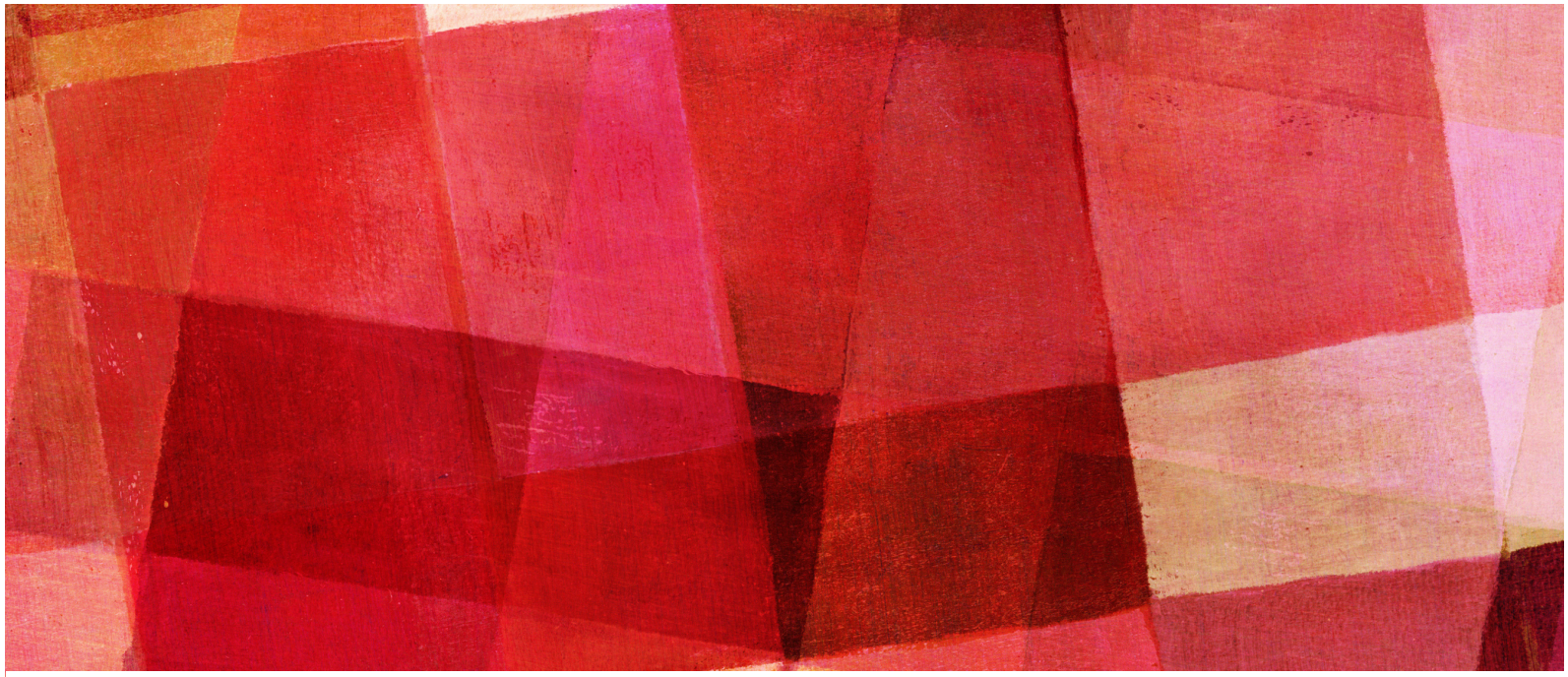
- Such an oscillation boundary often represents a point where the method would normally stop or turn around

In another words "The solution may go unfeasible"

Example 10

- In the knapsack problem, a TS may be designed to allow variables to be set to 1 even after reaching the feasibility boundary

- After a selected number of steps, the direction is reversed by choosing moves that change variables from 1 to 0



WRITE A TABU SEARCH PSEUDO CODE