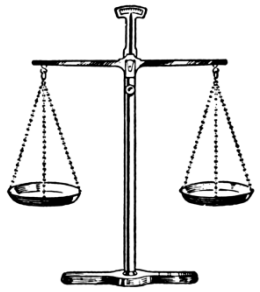


Tabu Search

Introduction

The word tabu (or taboo) originates from Tongan

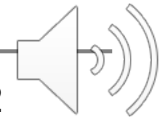
– a language of Polynesia

– used by the aborigines of Tonga island to indicate things that cannot be touched because they are sacred

Meaning of Tabu:

– *”Loaded with a dangerous, unnatural force”*

– *”Banned due to moral, taste or risk”*



Introduction

TabuSearch:

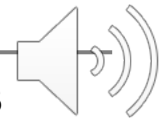
Cut off some parts of the search space (temporarily)

– Guide the search towards other parts of the search space by using penalties and bonuses

Uses structures that are exploring the search history, without remembering everything

– Branch&Bound: have complete memory

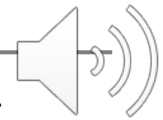
– Simulated Annealing: have no memory



Introduction

- Glover, F. 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*. Vol. 13, pp. 533-549.

- Hansen, P. 1986. The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming. *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy.



Introduction

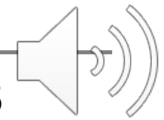
Based on Local Search – LS

It allows non-improving moves – can exit local optima

Uses extra memory to avoid looping, and to diversify the search

Tabu search exploits memory to classify a subset of the moves in a neighborhood as forbidden (or ***tabu***)

A ***tabu list*** records forbidden moves, which are referred to as ***tabu moves***



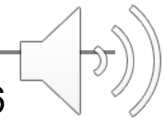
Introduction

Tabu restrictions are subject to an important exception.

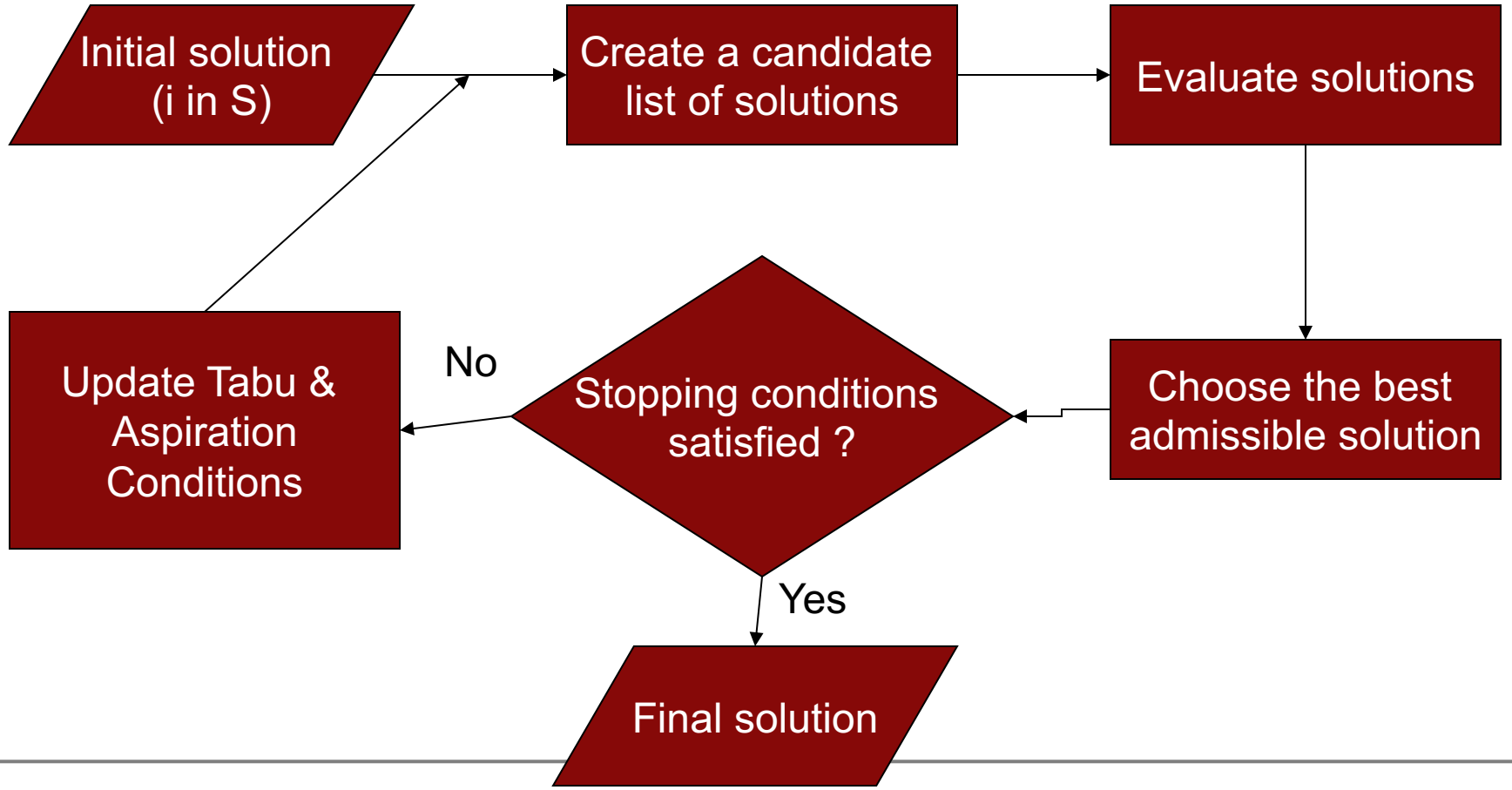
When a tabu move has a sufficiently attractive evaluation because it would result in a solution better than any visited so far, then its tabu classification may be overridden.

A condition that allows such an override to occur is called an ***aspiration criterion***

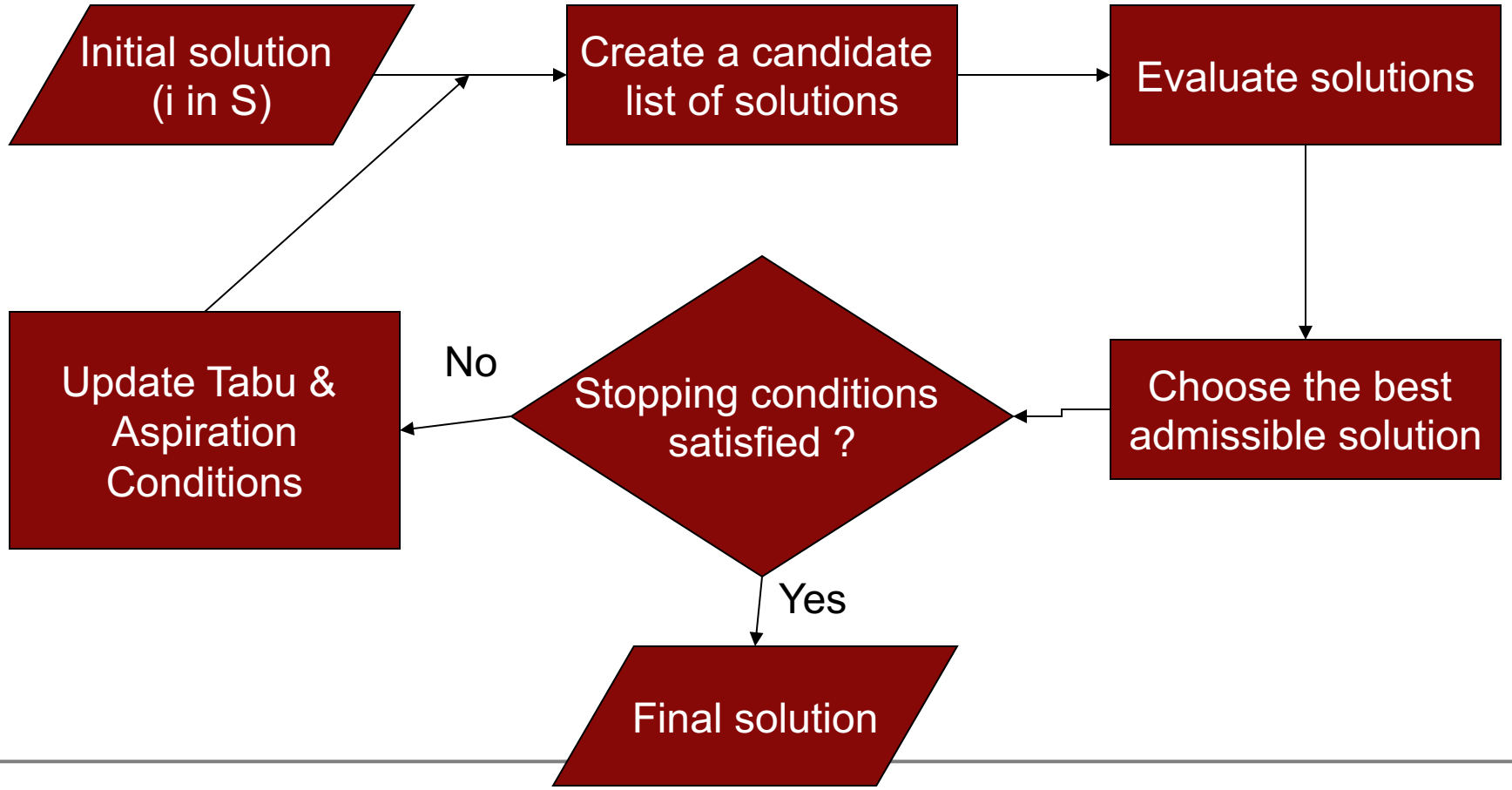
The ***neighborhood*** includes solutions that can be reached from current solution



Flowchart of a Standard Tabu Search Algorithm



Flowchart of a Standard Tabu Search Algorithm

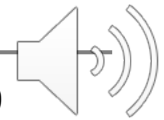


General Formulation

- Solution Representation and 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 and Aspiration criteria affect the search strategies:

Exploration vs Exploitation, ...



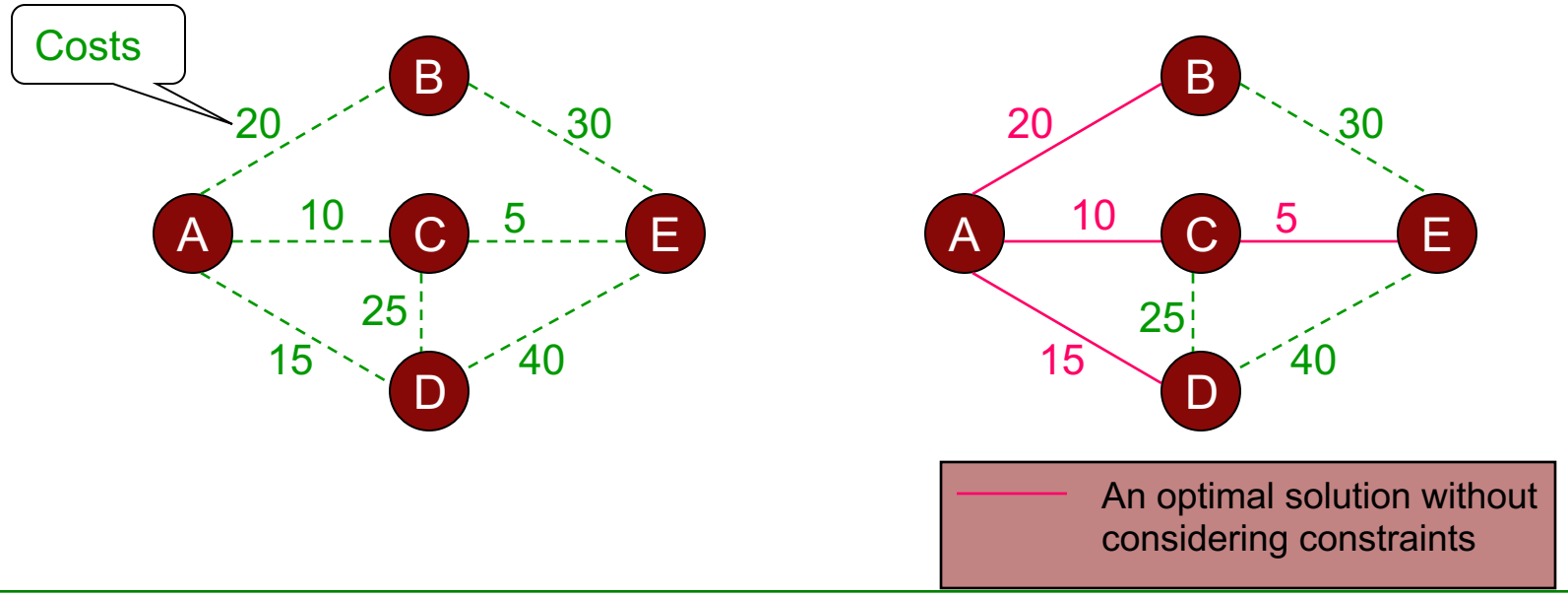
Tabu Search Stopping Conditions

Some immediate stopping conditions could be the following:

1. N(i, K+1) = 0. (no feasible solution in the neighborhood of solution i)
2. K is larger than the maximum number of iterations allowed.
3. The number of iterations since the last improvement of i* is larger than a specified number.
4. Evidence can be given than an optimum solution has been obtained.

Example [5]

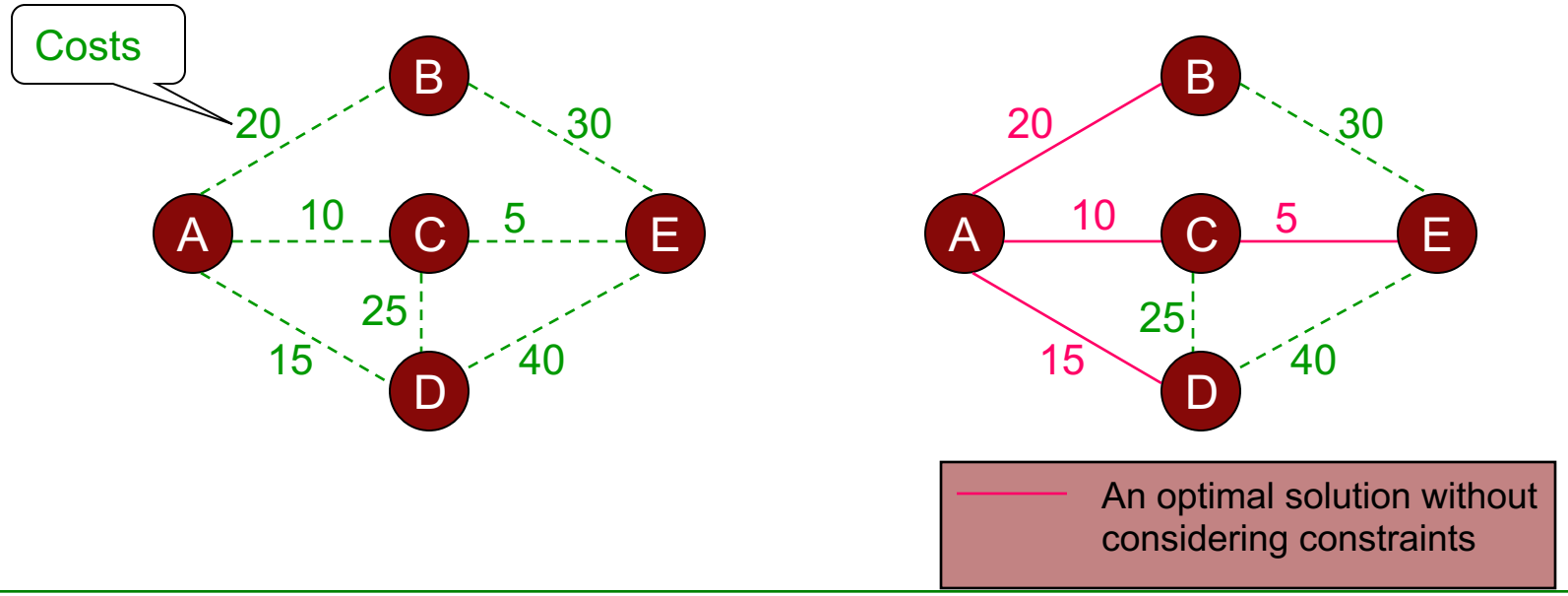
- Minimum spanning tree problem with constraints.
- *Objective: Connects all nodes with minimum costs*



Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example [5]

- Minimum spanning tree problem with constraints.
- *Objective: Connects all nodes with minimum costs*

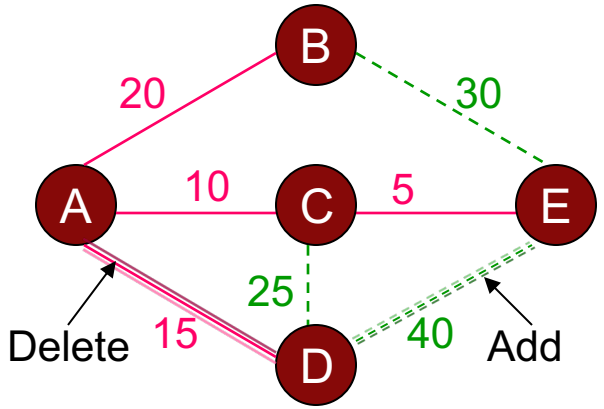


Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Iteration 1

Cost=50+200 (constraint penalties)



Add	Delete	Cost
BE	CE	75+200=275
BE	AC	70+200=270
BE	AB	60+100=160
CD	AD	60+100=160
CD	AC	65+300=365
DE	CE	85+100=185
DE	AC	80+100=180
DE	AD	**75+0=75**

New cost = 75 (iteration 2)

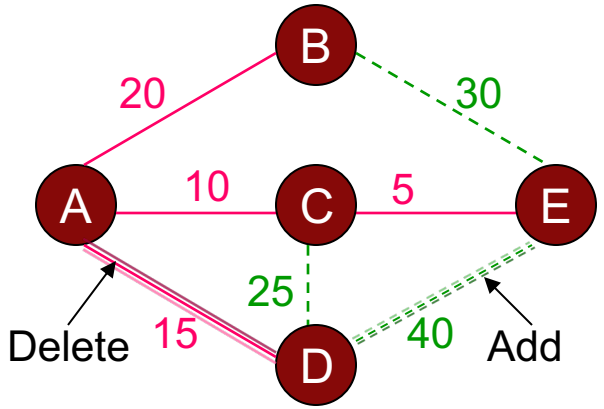
(local optimum)

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Iteration 1

Cost=50+200 (constraint penalties)



Add	Delete	Cost
BE	CE	75+200=275
BE	AC	70+200=270
BE	AB	60+100=160
CD	AD	60+100=160
CD	AC	65+300=365
DE	CE	85+100=185
DE	AC	80+100=180
DE	AD	**75+0=75**

New cost = 75 (iteration 2)

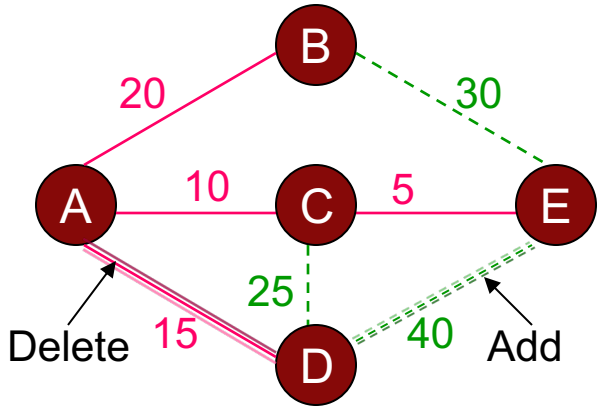
(local optimum)

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Iteration 1

Cost=50+200 (constraint penalties)



Add	Delete	Cost
BE	CE	75+200=275
BE	AC	70+200=270
BE	AB	60+100=160
CD	AD	60+100=160
CD	AC	65+300=365
DE	CE	85+100=185
DE	AC	80+100=180
DE	AD	**75+0=75**

New cost = 75 (iteration 2)

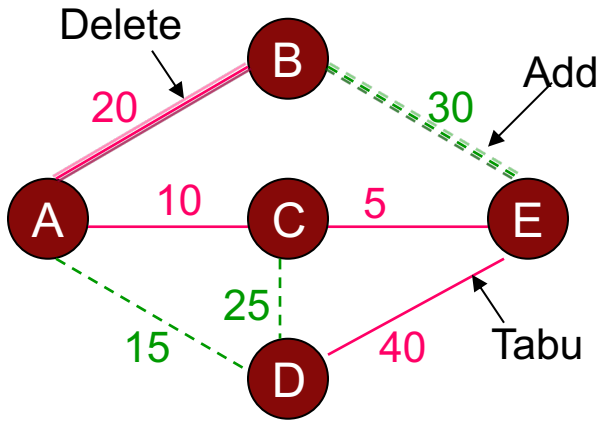
(local optimum)

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Tabu list: DE

Iteration 2 Cost=75



Add	Delete	Cost
AD	DE*	Tabu move
AD	CE	85+100=185
AD	AC	80+100=180
BE	CE	100+0=100
BE	AC	95+0=95
BE	AB	**85+0=85**
CD	DE*	60+100=160
CD	CE	95+100=195

**** A tabu move will be considered only if it would result in a better solution than the best trial solution found previously (Aspiration Condition)***

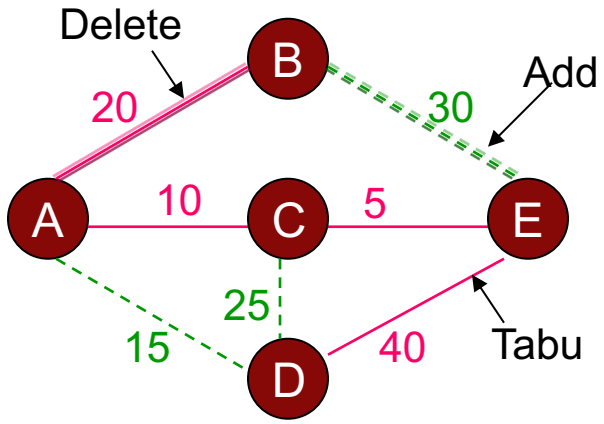
Iteration 3 new cost = 85 Escape local optimum

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Tabu list: DE

Iteration 2 Cost=75



Add	Delete	Cost
AD	DE*	Tabu move
AD	CE	85+100=185
AD	AC	80+100=180
BE	CE	100+0=100
BE	AC	95+0=95
BE	AB	**85+0=85**
CD	DE*	60+100=160
CD	CE	95+100=195

**** A tabu move will be considered only if it would result in a better solution than the best trial solution found previously (Aspiration Condition)***

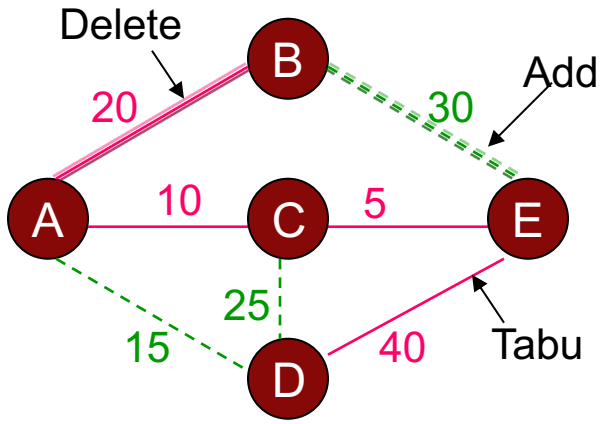
Iteration 3 new cost = 85 Escape local optimum

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Tabu list: DE

Iteration 2 Cost=75



Add	Delete	Cost
AD	DE*	Tabu move
AD	CE	85+100=185
AD	AC	80+100=180
BE	CE	100+0=100
BE	AC	95+0=95
BE	AB	**85+0=85**
CD	DE*	60+100=160
CD	CE	95+100=195

**** A tabu move will be considered only if it would result in a better solution than the best trial solution found previously (Aspiration Condition)***

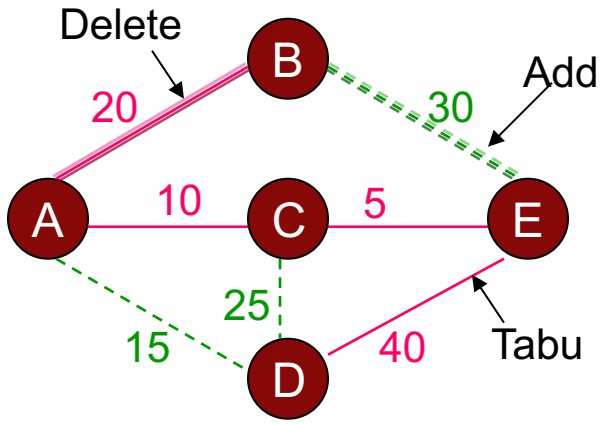
Iteration 3 new cost = 85 Escape local optimum

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Tabu list: DE

Iteration 2 Cost=75



Add	Delete	Cost
AD	DE*	Tabu move
AD	CE	85+100=185
AD	AC	80+100=180
BE	CE	100+0=100
BE	AC	95+0=95
BE	AB	**85+0=85**
CD	DE*	60+100=160
CD	CE	95+100=195

**** A tabu move will be considered only if it would result in a better solution than the best trial solution found previously (Aspiration Condition)***

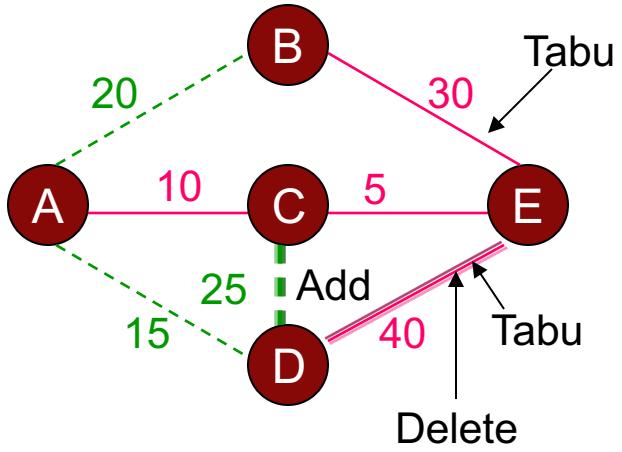
Iteration 3 new cost = 85 Escape local optimum

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example

Tabu list: DE & BE

Iteration 3 Cost=85



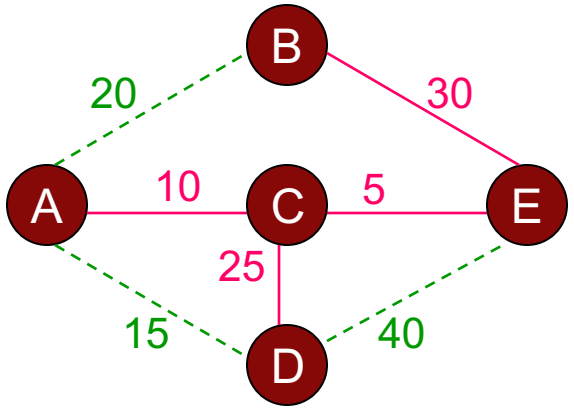
Add	Delete	Cost
AB	BE*	Tabu move
AB	CE	100+0=100
AB	AC	95+0=95
AD	DE*	60+100=160
AD	CE	95+0=95
AD	AC	90+0=90
CD	DE*	**70+0=70**
CD	CE	105+0=105

**** A tabu move will be considered only if it would result in a better solution than the best trial solution found previously (Aspiration Condition)***

Iteration 4 new cost = **70** Override tabu status

Constraints 1: Link AD can be included only if link DE also is included. (penalty:100)
Constraints 2: At most one of the three links – AD, CD, and AB – can be included. (Penalty of 100 if selected two of the three, 200 if all three are selected.)

Example



Optimal Solution

Cost = 70

Additional iterations only find inferior solutions