METAHEURISTICS

Constructive Heuristics

Neighborhood

Constructive Heuristics

- A type of heuristic method that refer to the process of constructing an initial feasible solution from scratch.

It differs from local search heuristics which start with a complete solution and then try to improve the current solution further via local moves

- Starts with a scratch (empty solution).
- Used local heuristic to select the new element to be included in the solution.
- Repeatedly extends the current solution until a complete solution is obtained.

Simple to design

Easy to implement

Executes quickly

But, usually, no optimality

Greedy Algorithms

Creating an empty solution ***S; S*** ={};
Repeat
e_i = Local-Heuristic(E\{e/e ∈ s}) ; /* E is set of elements */
/∗ next element selected from the set E minus already selected elements ∗/
If S∪ e_i ∈ F **Then** /∗ test the feasibility of the solution , F∈ ***search space*** ∗/
S = ***S*** ∪ e_i ;
Until Complete solution found

Local-Heuristic used to choose the best element

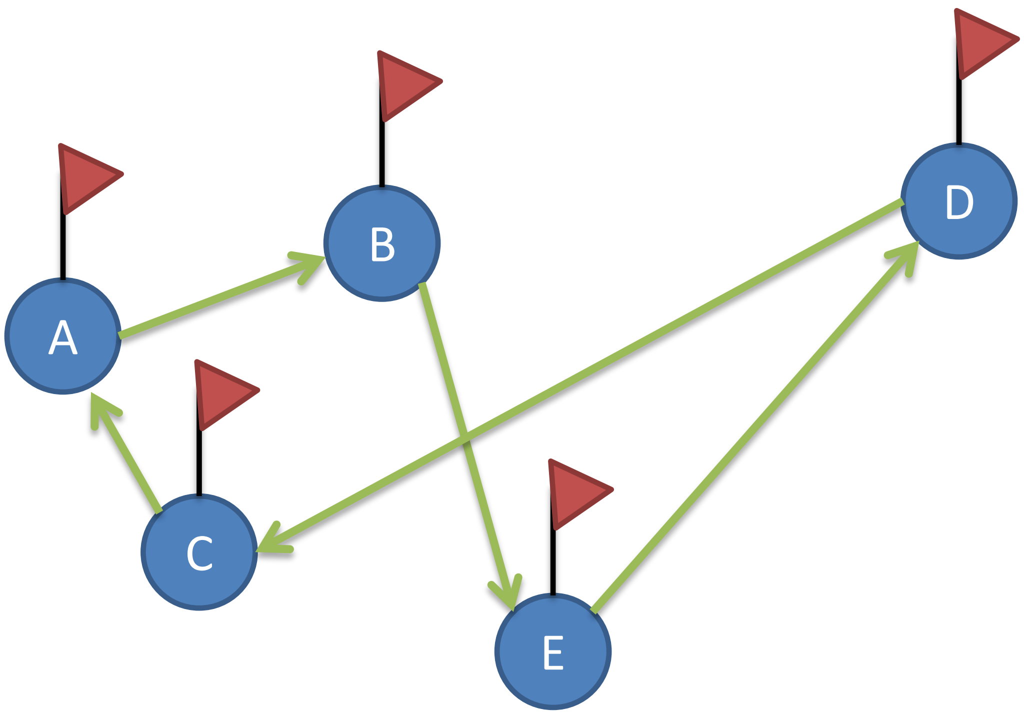
Nearest Neighbour algorithm

Creating an empty solution ***S; S*** ={};
Randomly select e
S = ***S*** ∪ e ;
Repeat
Next nearest element selected from the set E (e_i)
If S∪ e_i ∈ F **Then** /$*$ test the feasibility of the solution , F∈ ***search space*** $*$/
S = ***S*** ∪ e_i ;
Until Complete solution found

Random algorithm

Creating an empty solution ***S; S*** ={};
Randomly select e
S = ***S*** ∪ e ;
Repeat
Next element is randomly selected from the set E (e_i)
If S∪ e_i ∈ F **Then** /∗ test the feasibility of the solution , F∈ ***search space*** ∗/
S = ***S*** ∪ e_i ;
Until Complete solution found

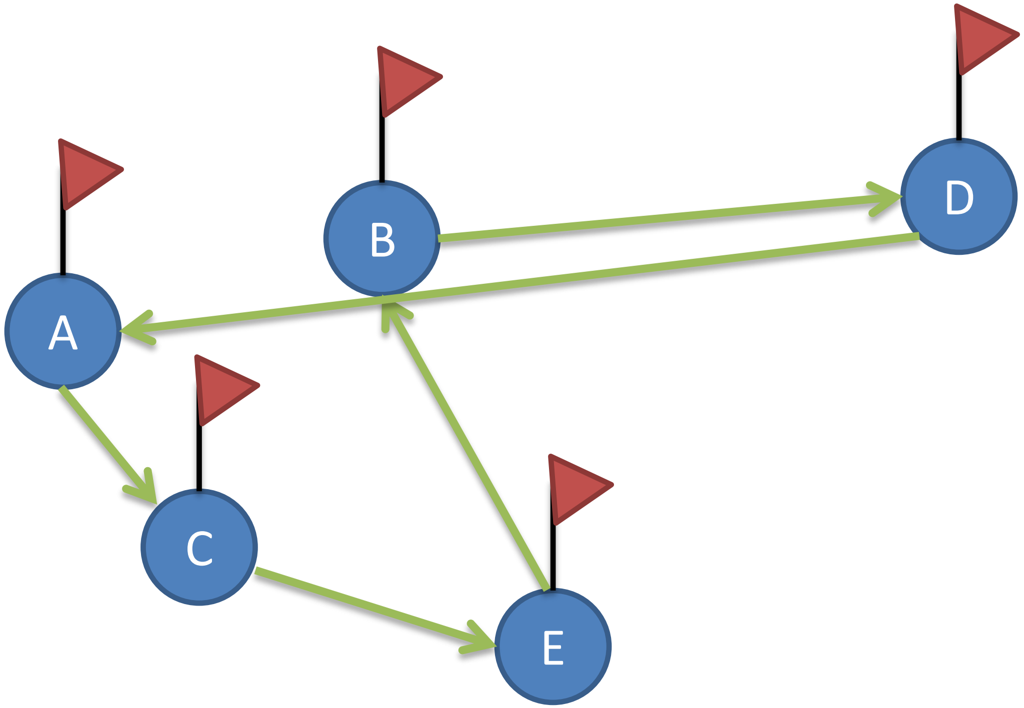
Example: Random Algorithm - TSP



DISTANCE TABLE					
	A	**B**	**C**	**D**	**E**
A	0	2	1	7	5
B		0	2	4	3
C			0	6	2
D				0	4
E					0

Solution: A-B-E-D-C-A Cost=2+3+4+6+1 Cost=16

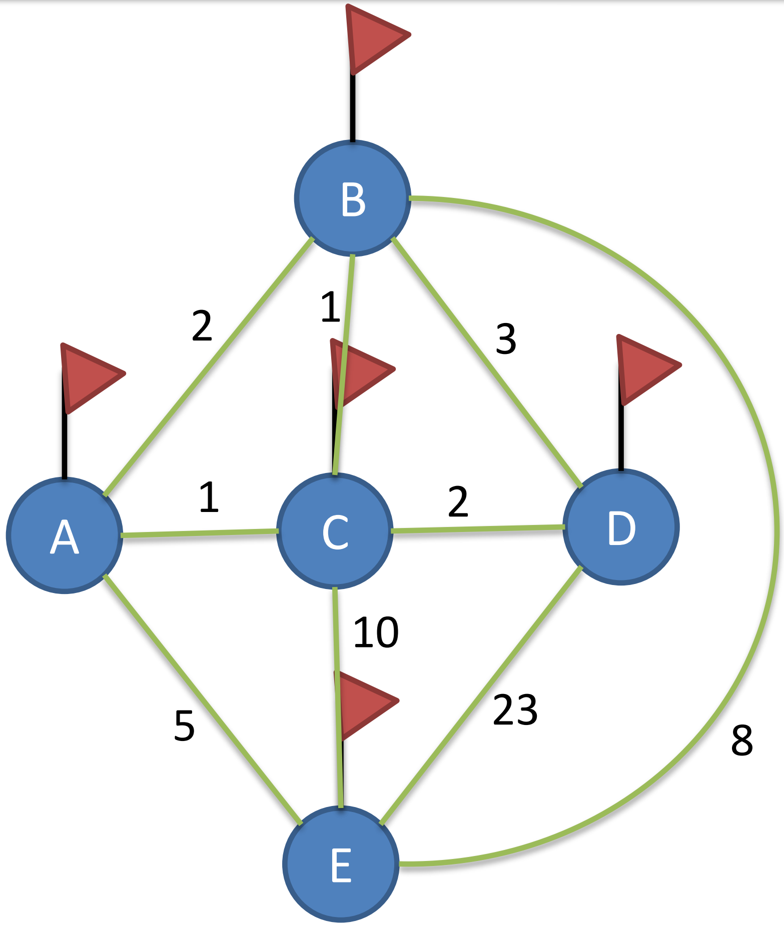
Example: Nearest Neighbour Algorithm - TSP



DISTANCE TABLE					
	A	**B**	**C**	**D**	**E**
A	0	2	1	7	5
B		0	2	4	3
C			0	6	2
D				0	4
E					0

Solution: A-C-E-B-D-A Cost=1+2+3+4+5 Cost=15

Example2: Random Algorithm



Solution: A-C-E-D-B-A

Cost=1+10+23+3+2

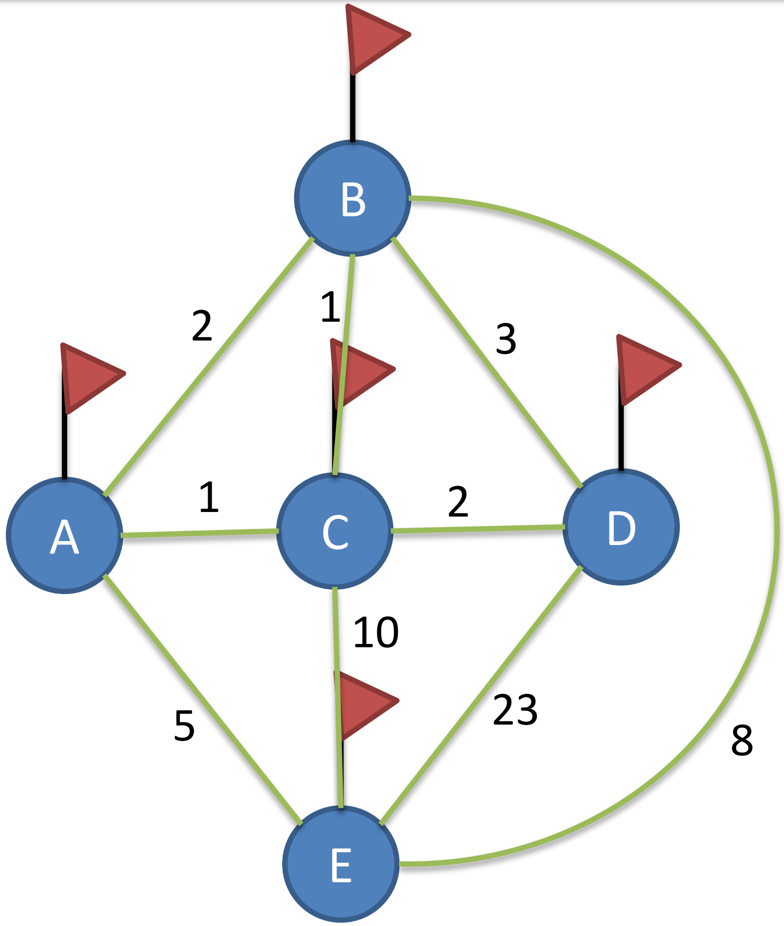
Cost=39

Optimal Solution: A-C-D-B-E-A

Cost=1+2+3+8+5

Cost=19

Example2: Nearest Neighbour Algorithm



Solution: A-C-B-D-E-A

Cost=1+1+3+23+5

Cost=23

Optimal Solution: A-C-D-B-E-A

Cost=1+2+3+8+5

Cost=19

Neighborhood

A neighborhood function ***N*** is a mapping ***N*** : $S \rightarrow 2^s$ that assigns to each solution ***s*** of ***S*** a set of solutions ***N(s)*** ⊂ S

- Plays a crucial role in the performance of metaheuristic.
- If the neighborhood structure is not adequate to the problem, any metaheuristic will fail to solve the problem

A solution **s'** in the neighborhood of ***s*** (s' ∈ N(**s**)) is called a neighbor of ***s***.

A **neighbor** is generated by the application of a move operator m that performs a small perturbation to the solution ***s***.

Neighborhood

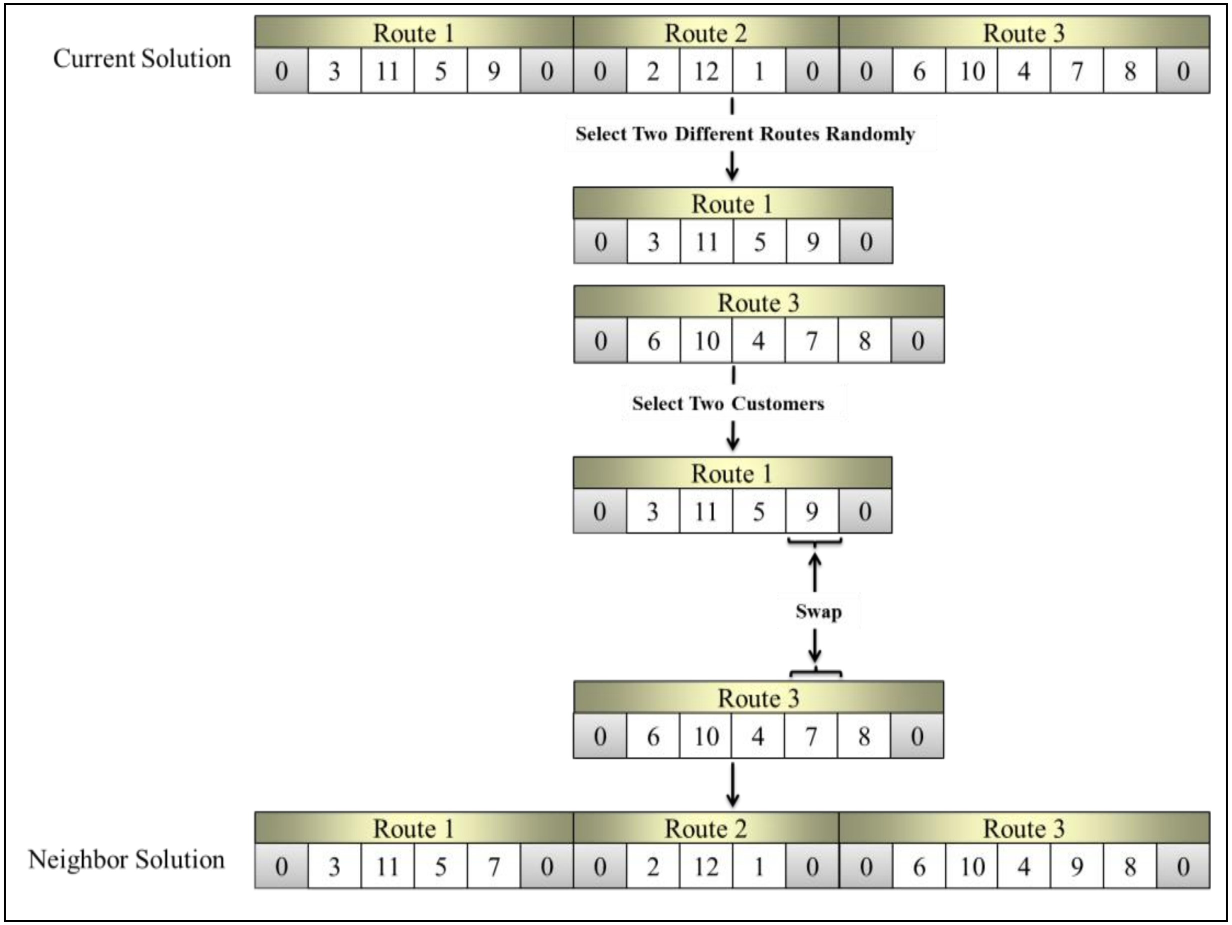
The main property that must characterize a neighborhood is **locality**.

Locality is the effect on the solution when performing the move (perturbation) in the representation.

When small changes are made in the representation (small effect on the solution, the neighborhood is said to have a **strong locality**.

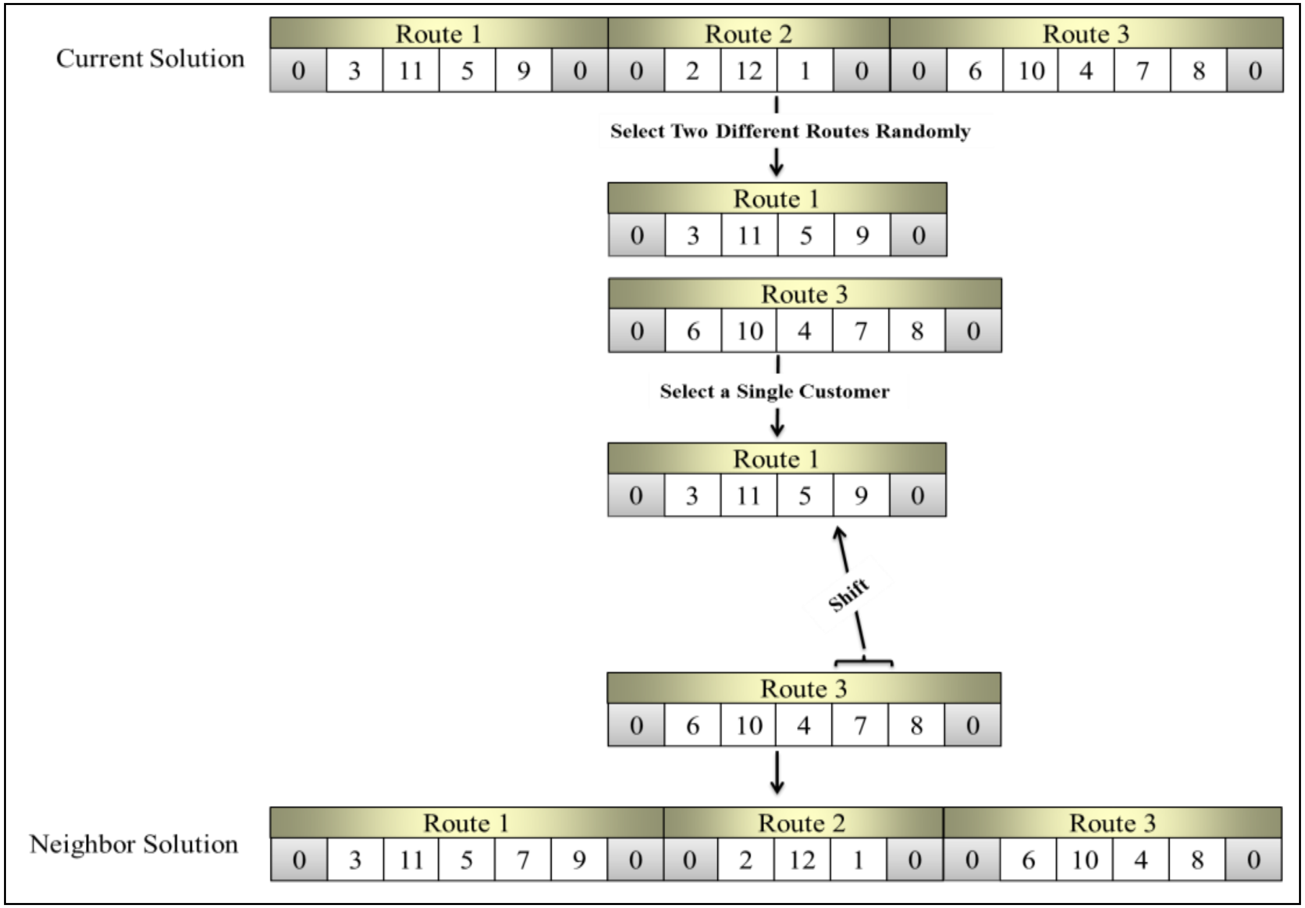
Weak locality is characterized by a large effect on the solution when a small change is made in the representation.

Example: Swap neighborhood operator (VRP)



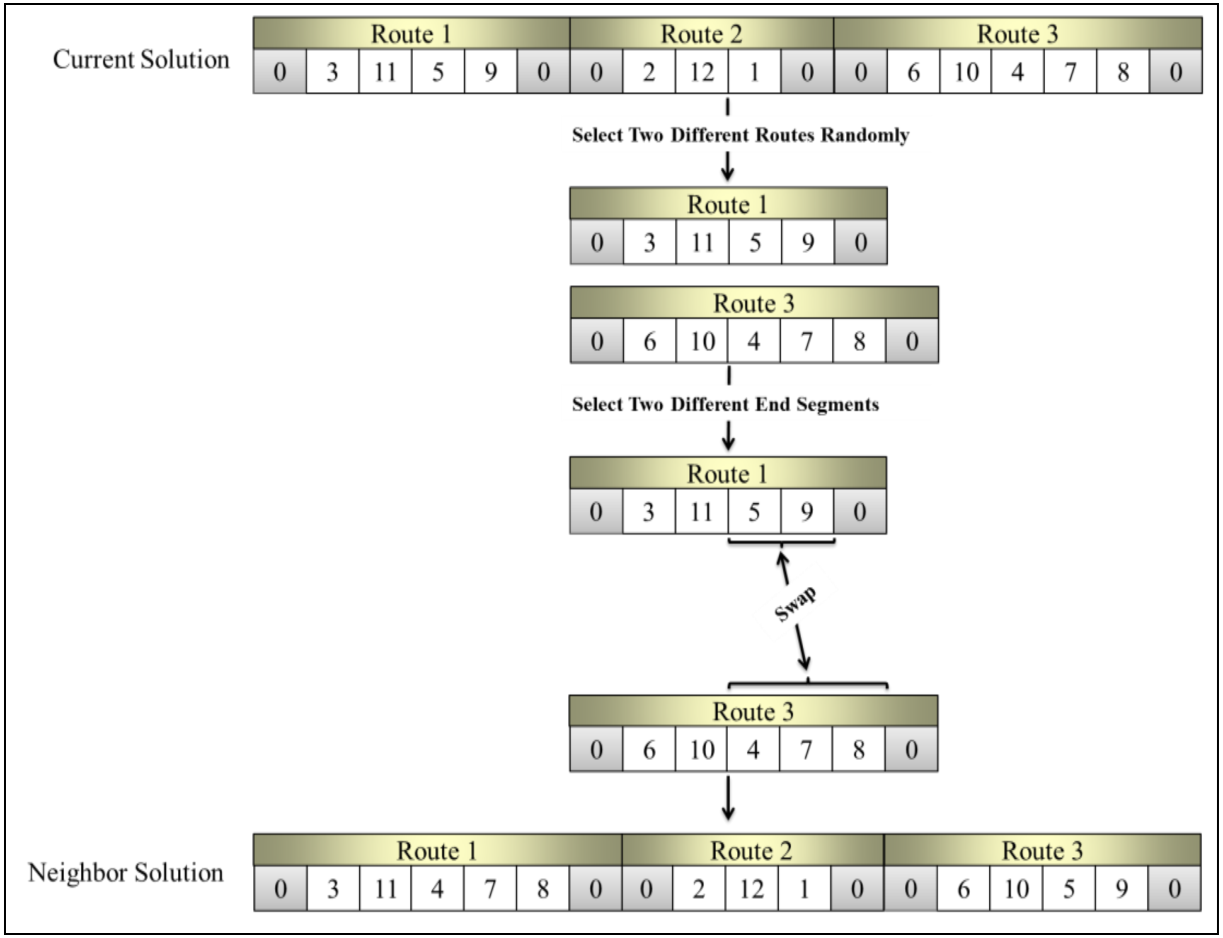
replace a single customer by another in different route.

Example: Move (Shift) neighborhood operator (VRP)



transfer one customer to the other route

Example: 2-opt star neighborhood operator (VRP)



randomly selects two routes from the current solution and swap the customers located at the end sections of selected routes

Single-base Meta-heuristics