COMBINATORIAL OPTIMIZATION

$< s , f >$: an instance

s : Solutions Set

$f : s \rightarrow \Re$ Cost function to minimize (Max)

Find $s^* \in S$ s.t.

$f(s^*) \leq f(s), \forall s \in S$ (MIN)

or

$f(s^*) \geq f(s), \forall s \in S$ (MAX)

Local Search (LS)

- Neighborhood structure :

  - i solution

  - $N : S \rightarrow 2^S$

    $i \rightarrow N(i) \subseteq S$

    $N(i)$ = ¨ near ¨ to i solutions

- ĩ is a local minimum if

  $f(i) \leq f(j), \forall j \in N(i)$

- ĩ is a local maximum if

  $f(i) \geq f(j), \forall j \in N(i)$

Local Search Algorithm

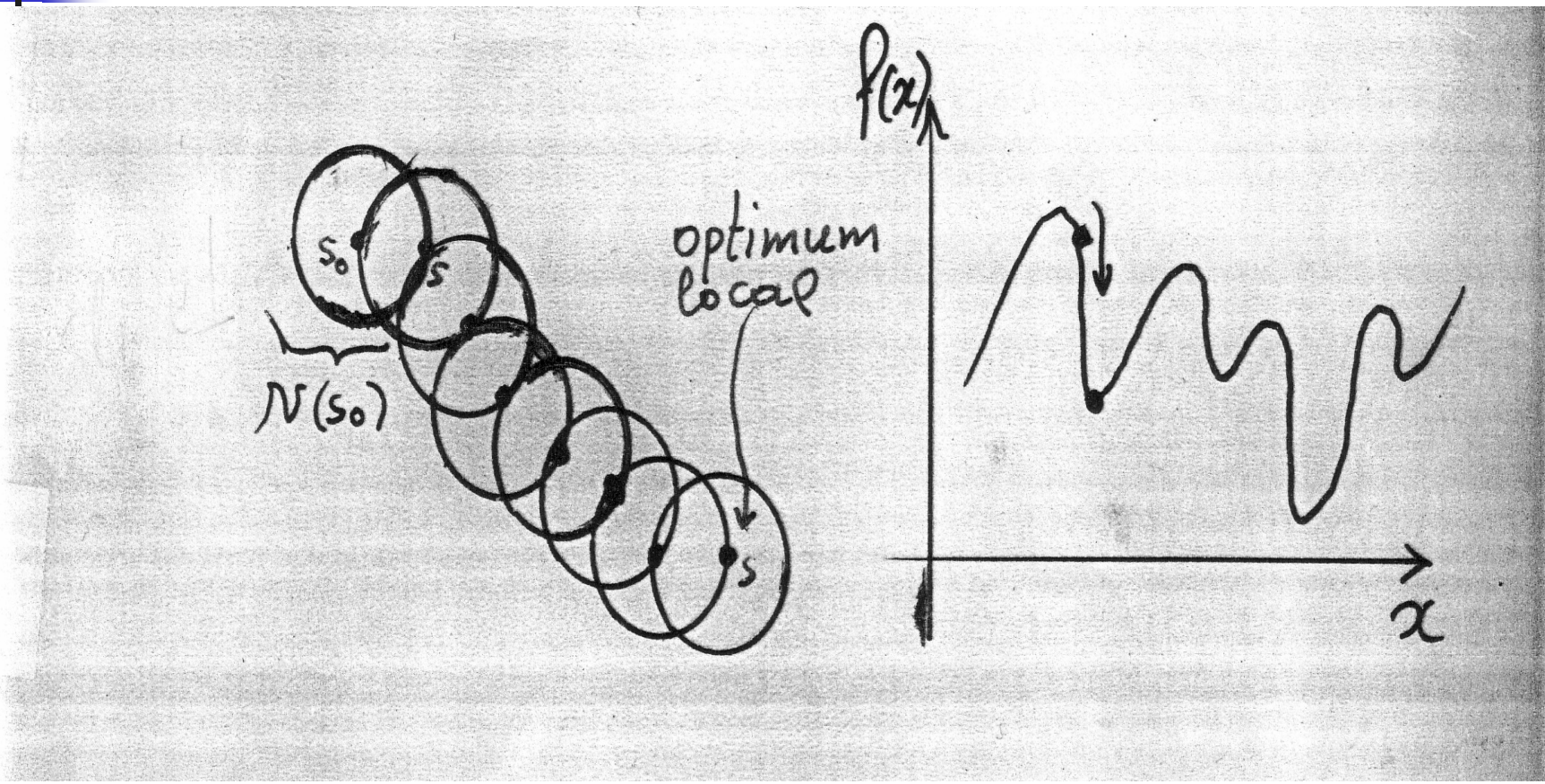
- Define a neighborhood N (S)
- Initial solution $S = S_0$
- REPEAT

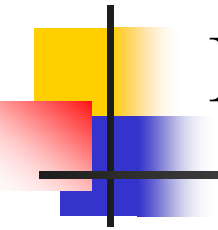
  Find a solution $S' \in N(S)$

  improving the cost $S := S'$

  If S' does not exist STOP (local optimum)

The local search algorithm





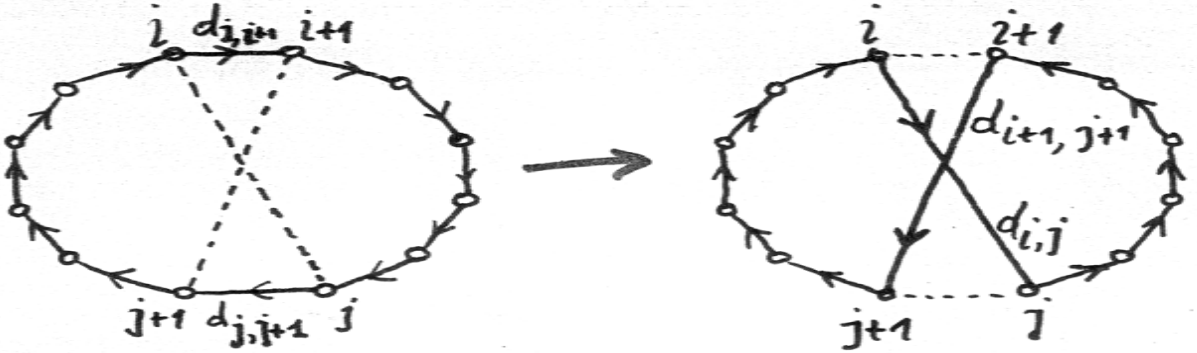
Examples

- The Traveling Salesman Problem

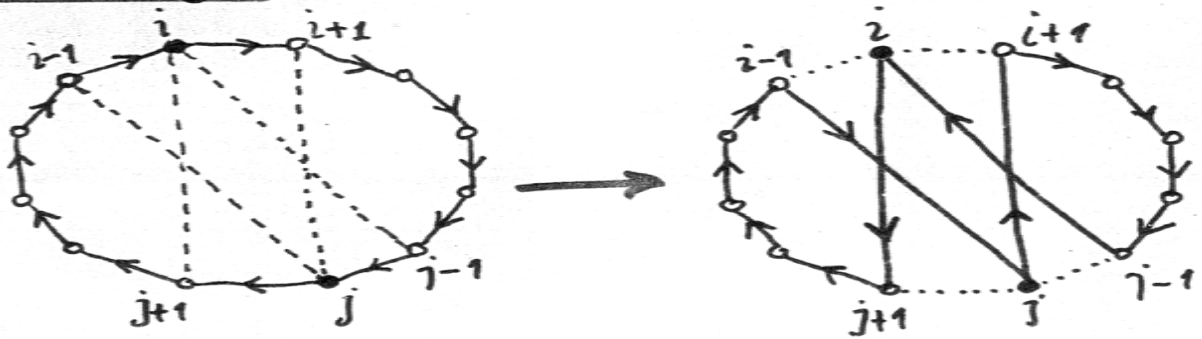
  2 - opt , 3 – opt , . . . , k – opt

  2 - exchange

2-opt



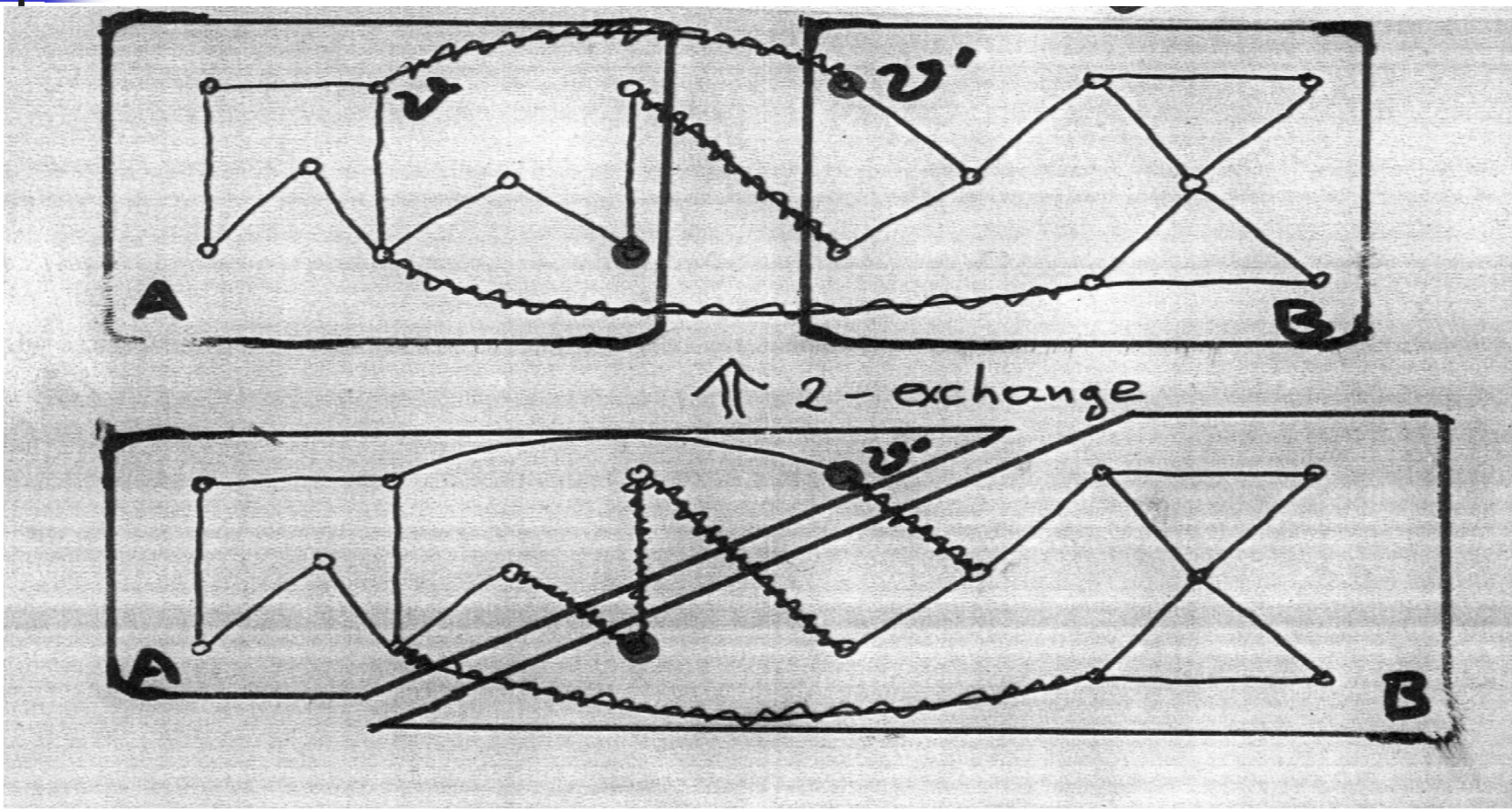
2-exchange



Examples

- The Bipartitioning of a weighted graph $G (V , E , W)$, $|V| = 2 n$.
  Find partitions A , B of V with
  $|A| = |B|$ and
  Minimizing $f (A , B) = \sum_{\substack{\vartheta \in A \\ \vartheta' \in B}} \omega_{\vartheta \vartheta'}$

Graph Bipartitioning



Search strategies in LS

- First improvement
- Best improvement
- Worst improvement