Very Large Scale Neighborhood Search

By Jim Orlin

Collaborators include:

Ravi Ahuja, Ozlem Ergun,

Abraham Punnen, Dushyant Sharma

Neighborhood Search

***Combinatorial Optimization*: minimize (f(S) : S $\in$ F)**

- **f is typically linear,**
- **F is finite**

***Neighborhood Function*:**

- **For each S $\in$ F, there is a *neighborhood* N(S);**
- **We say that S is a *local optimum* if f(S) $\leq$ f(T) for all T $\in$ N(S);**

Neighborhood Search

Neighborhood Search
(local improvement algorithm)

begin

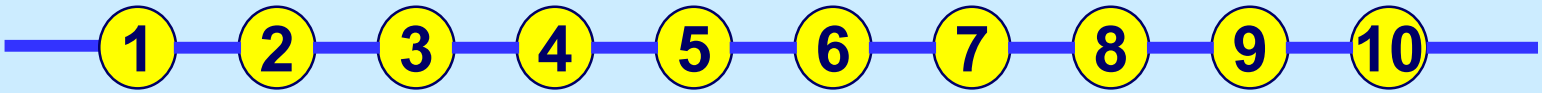
initialize with some S $\in$ F;

while **S is not a local optimum** ***do***

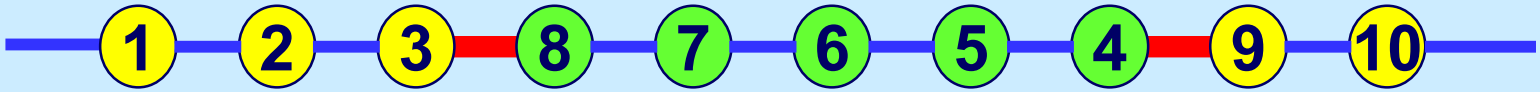
replace S by some T $\in$ N(S) such that
f(T) < f(S);

end

TSP and 2-exchanges



The original tour



A 2-neighbor of the original tour

We say that a tour T' is a ***2-neighbor*** of a tour T if it is possible to obtain T' from T by adding two edges and deleting two edges. The operation is called a **2-exchange.**

When we say city i, we really mean the city that is in position i of the current tour.
(Or you may assume that the current tour is 1, 2, 3, ..., n)

T' = T + (3,8) + (4,9) - (3,4) - (8,9).
Obtained by the operation **Flip[4,8]**

A Neighborhood Search Technique has 3 Parts
Neighborhood/Search/Select

1. *A neighborhood structure*, e.g., 2-exchange neighborhood
2. *A method for searching the neighborhood*
   - Start searching from the *current solution*
   - Simulated annealing: a neighbor is selected at random.
   - Tabu search: the entire neighborhood is searched.
3. *A method for selecting the next current solution*
   - Simulated annealing: selection depends on temperature
   - Tabu search: selection depends on the tabu list and more.

Very Large Scale Nbhd (VLSN) search

Rule of Thumb for Larger Neighborhoods:

improved local optima

greater search time

This talk:

Focuses on VERY LARGE neighborhoods that can be searched very efficiently (preferably in polynomial time) or are searched heuristically.

often exponentially large neighborhoods

I'm planning on using the 2-opt neighborhood. It has n^2 neighbors.

Pretty good. But please check out larger neighborhoods.

Two survey papers

"A Survey of Very Large Scale Neighborhood Search Techniques",
Ahuja, Ergun, Orlin, and Punnen [1999]
http://web.mit.edu/jorlin/www/

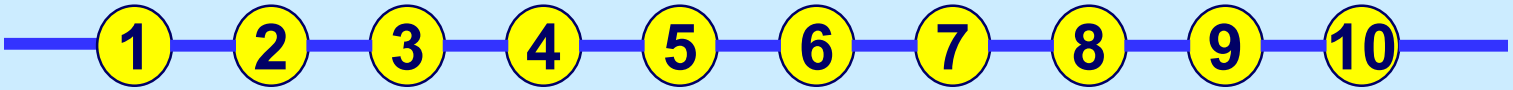
“A study of exponential neighborhoods....” Deineko and Woeginger [2000]

Personal view of VLSN search

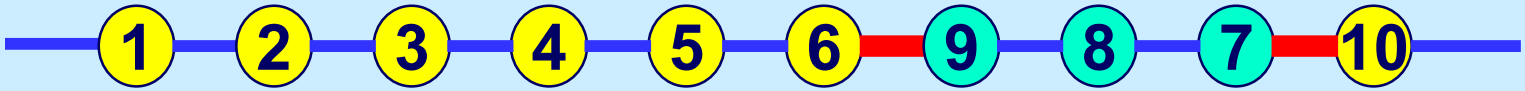
- I view it as liberating
  - it does not restrict neighborhood search to exhaustively searching a neighborhood
  - permits many alternatives to standard neighborhood search
- I view it as a very practical method for addressing problems that are large
- For decades researchers have argued that solving special cases of problems is potentially of use because it can help us solve more general cases.
  - This is one way of realizing the potential

An example of VLSN: Independent 2-exchanges

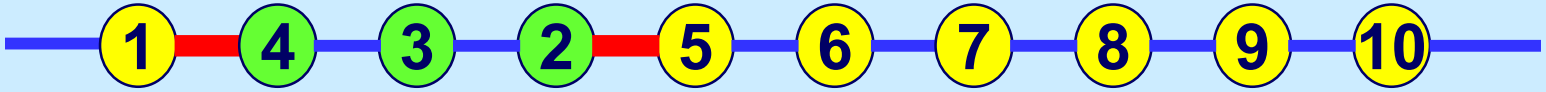
The original tour T



Flip[7,9] and obtain T_1

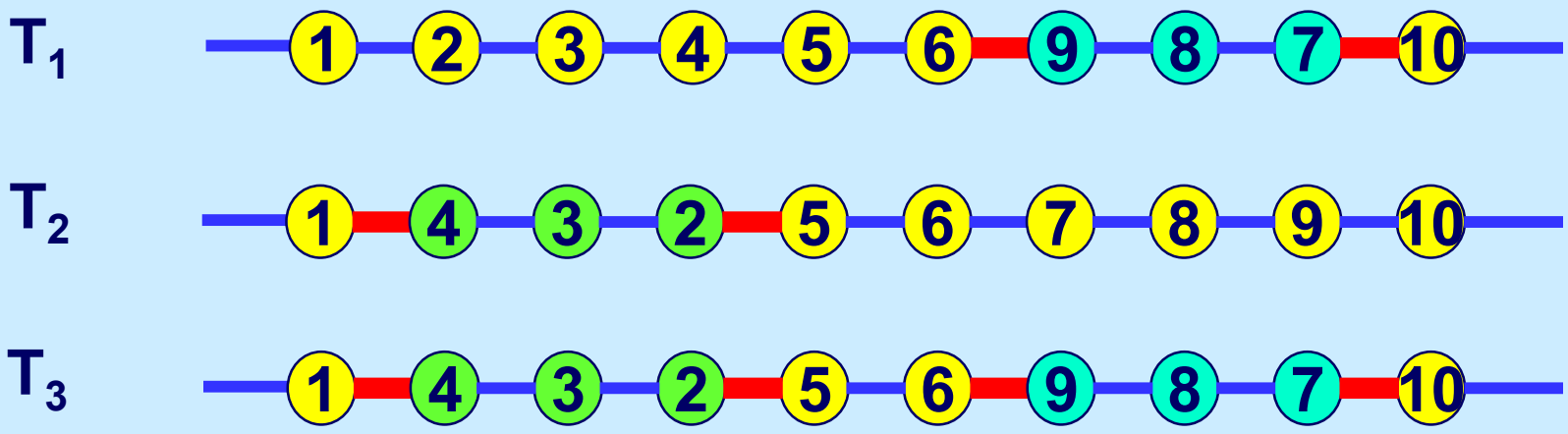


Flip[2,4] and obtain T_2



Two exchanges, Flip[i,j] and Flip[i',j'] are independent if i' > j+1 (or i > j'+1).

Independent 2-exchanges

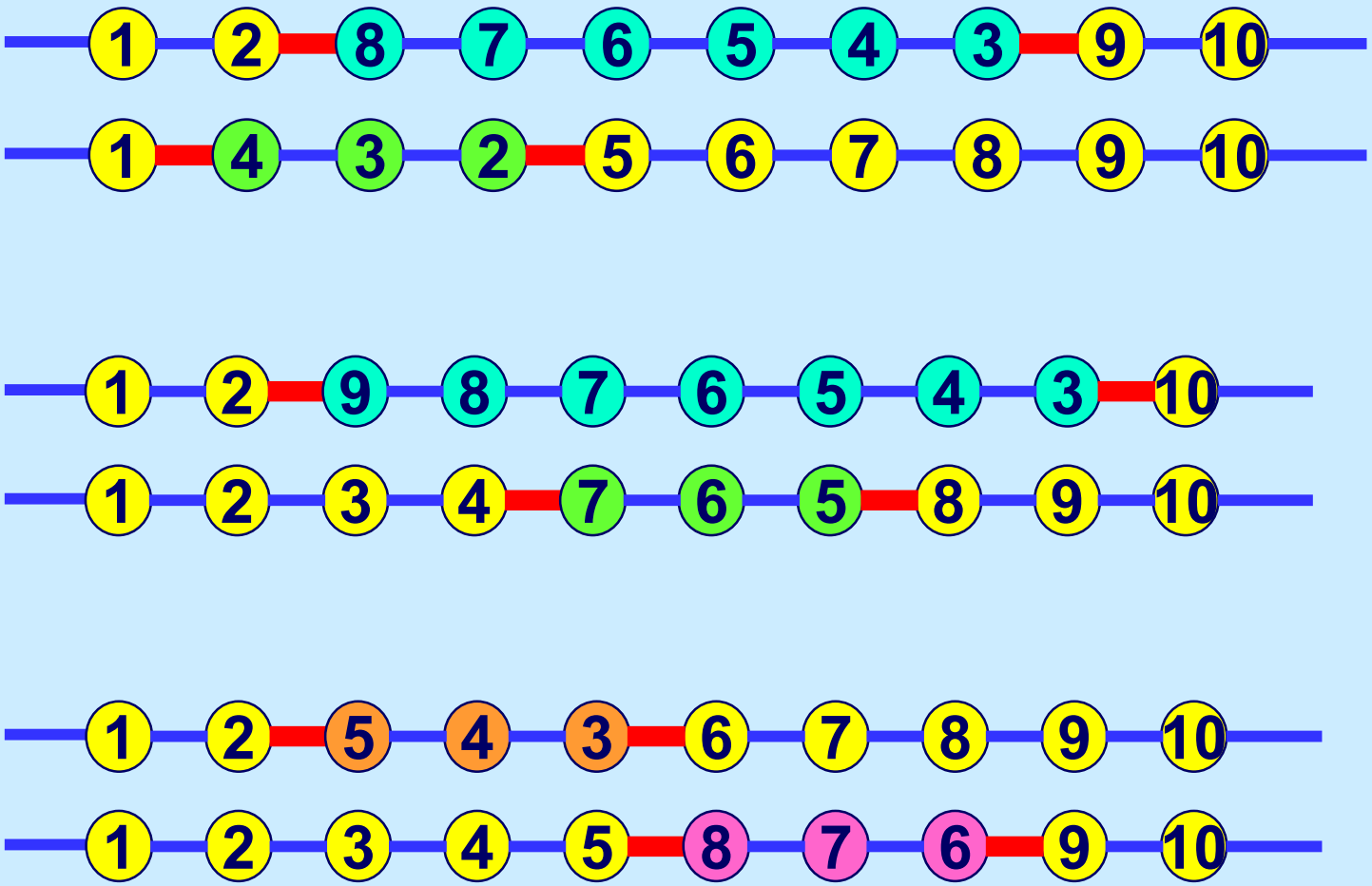


Flips [2,4] and [7,9] are independent

- **costs can be calculated independently**
- **one interval is to the left of the other (used for efficient searching)**

$$c(T_3) - c(T_0) = \underbrace{[c(T_1) - c(T_0)]}_{\text{cost of [7,9]}} + \underbrace{[c(T_2) - c(T_0)]}_{\text{cost of [2,4]}}.$$

Pairs that are not independent



Dynasearch/ Ejection Chains, and more

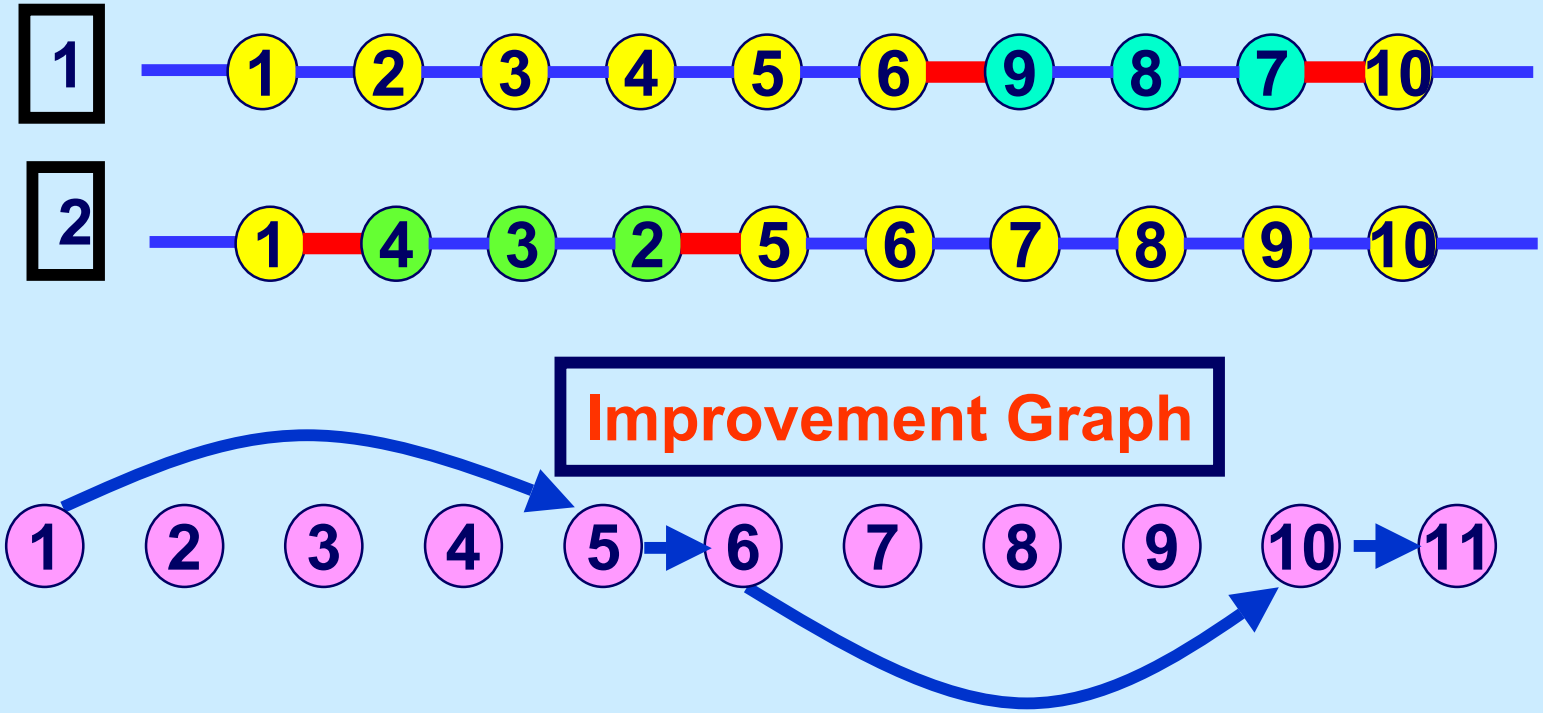
We say that T' is an independent 2-exchange neighbor of T if T' can be obtained via k independent 2-exchanges for some k.

Size of neighborhood = $\Theta(n^{1.815})$

Dynasearch: Potts and Van de Velde [1996]
$O(n^2)$ DP algorithm.

Ejection chains: Glover [1992].

The Improvement Graph

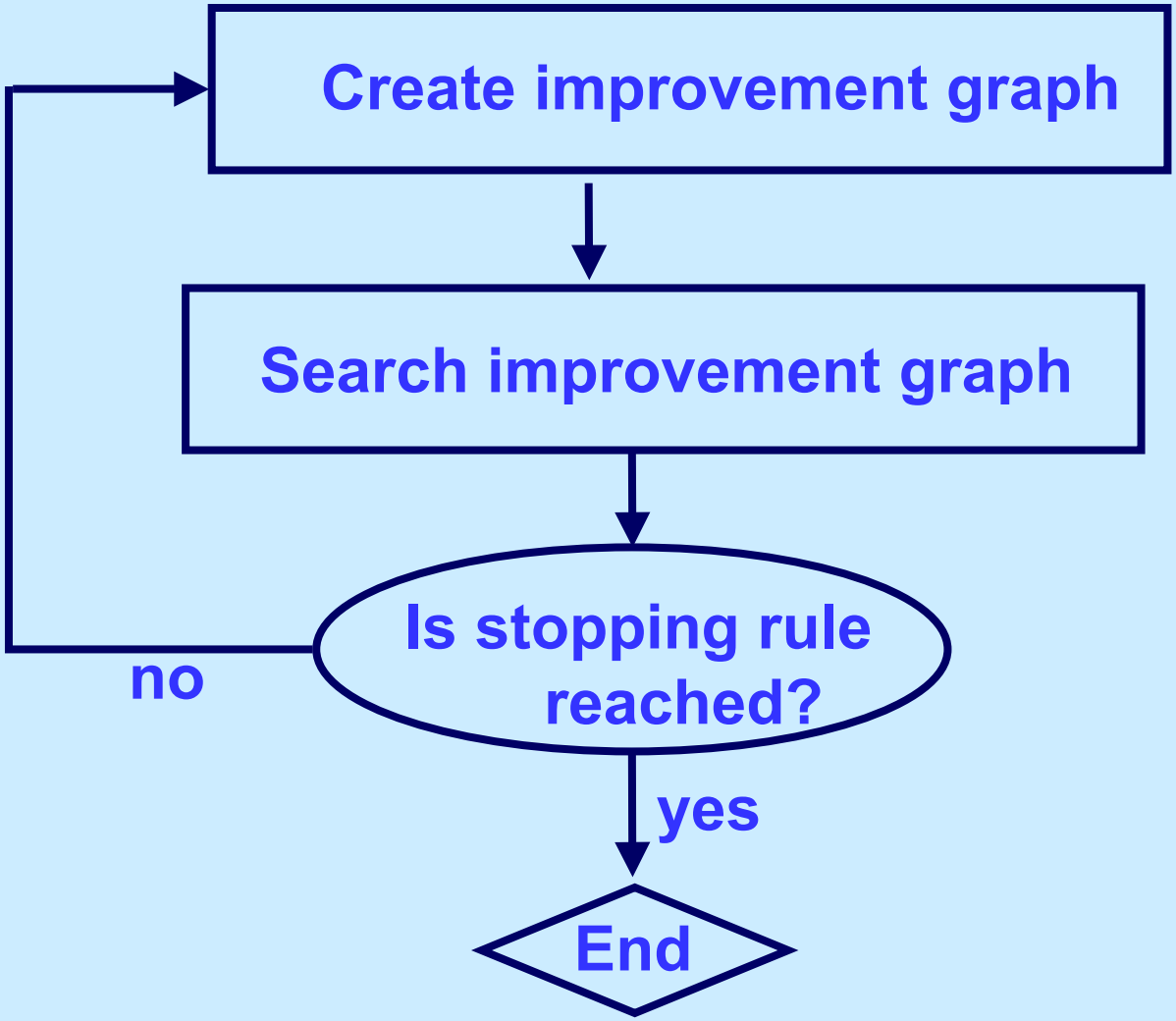


c_{ij}: cost of keeping city i fixed and city j fixed and flip cities i+1, ..., j-1

Min cost collection of independent 2-exchanges:
shortest path from node 1 to node n+1.

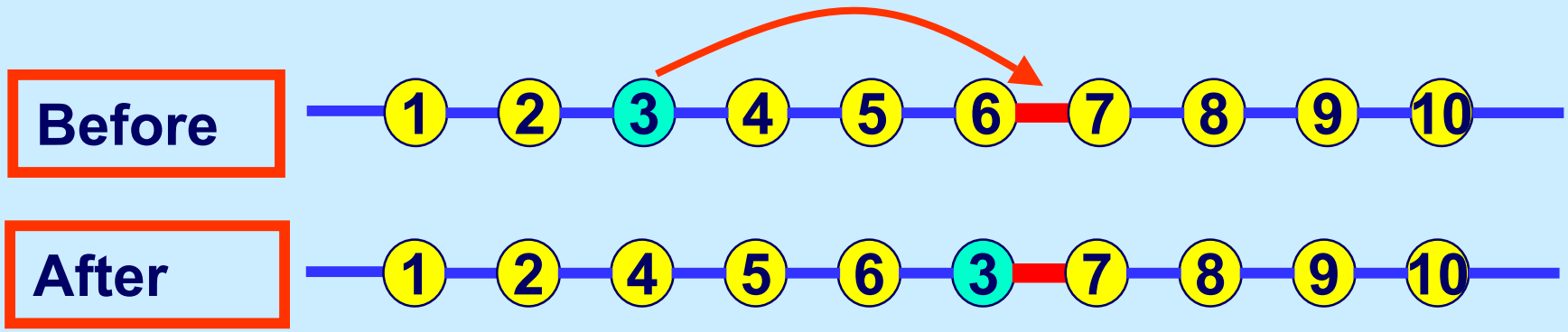
$O(n^2)$ time. Note: there are $O(n^2)$ 2-exchanges

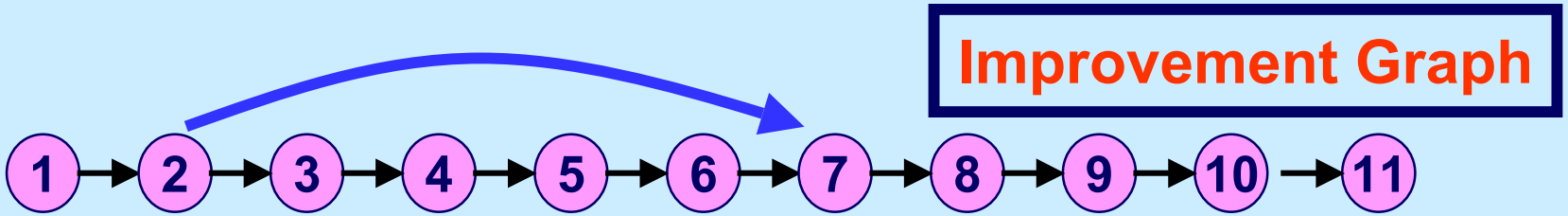
Summary of the Method



One can incorporate tabu search by adjusting the improvement graph

Insertion based neighborhoods





c_{ij}: cost of keeping city i fixed and city j fixed and inserting city i+1 after city j-1.

Min cost collection of independent 2-exchanges:
shortest path from node 1 to node n+1.
$O(n^2)$ time.

Some advantages for VLSN search

- They can provide a limited form of look-ahead
- They can provide a limited form of parallelism
- They can be very effective in practice.
- They are appealing.

Cyclic Exchange for Partitioning Problems

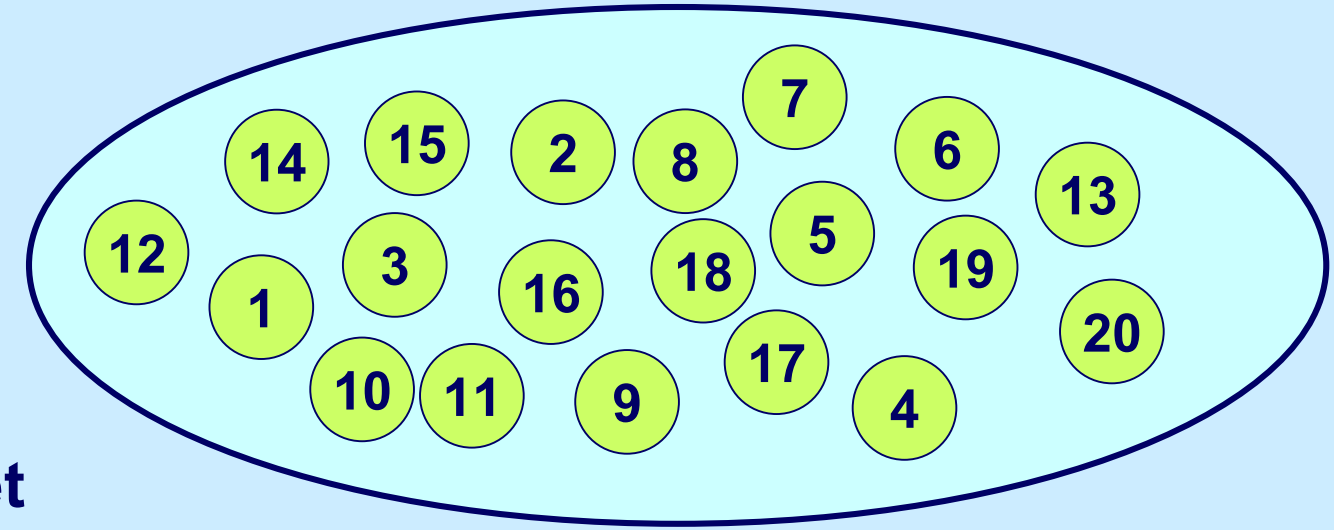
- S : set of objects
- Partition S into subsets (S_1, S_2, S_3, .. , S_p)
- Define a (possibly nonlinear) cost function $f(S_i)$ for each subset S_i
- Find the partition S_1, S_2, … , S_p such that

$$\sum_{p=1,K} f(S_p)$$

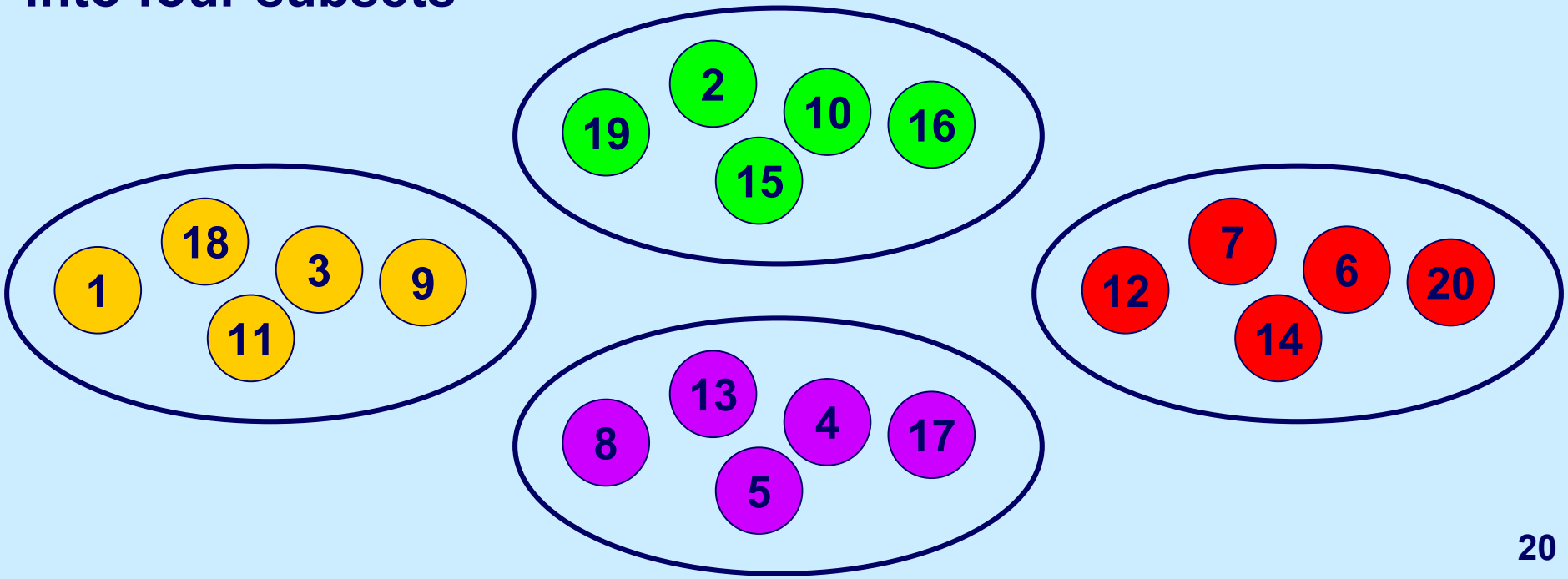
is minimum.

Partitioning Problems

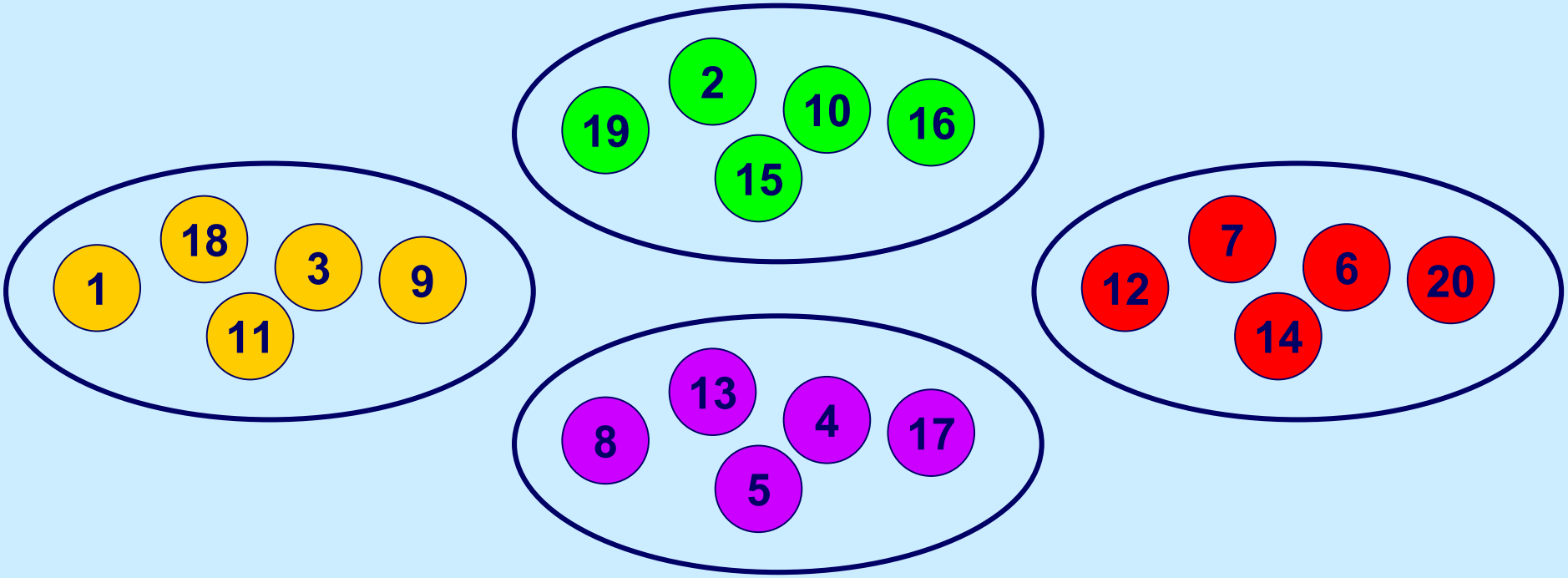
Given a set S of 20 objects



Partition the set into four subsets



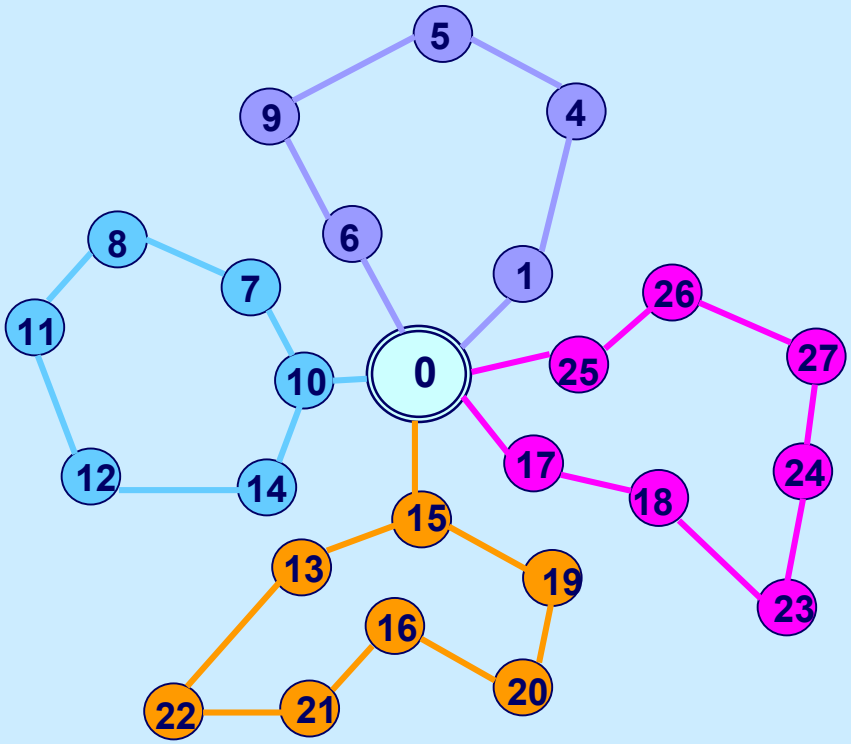
The Cost Structure



The cost for part S_i is $f(S_i)$.

Find the partition with minimum total cost.

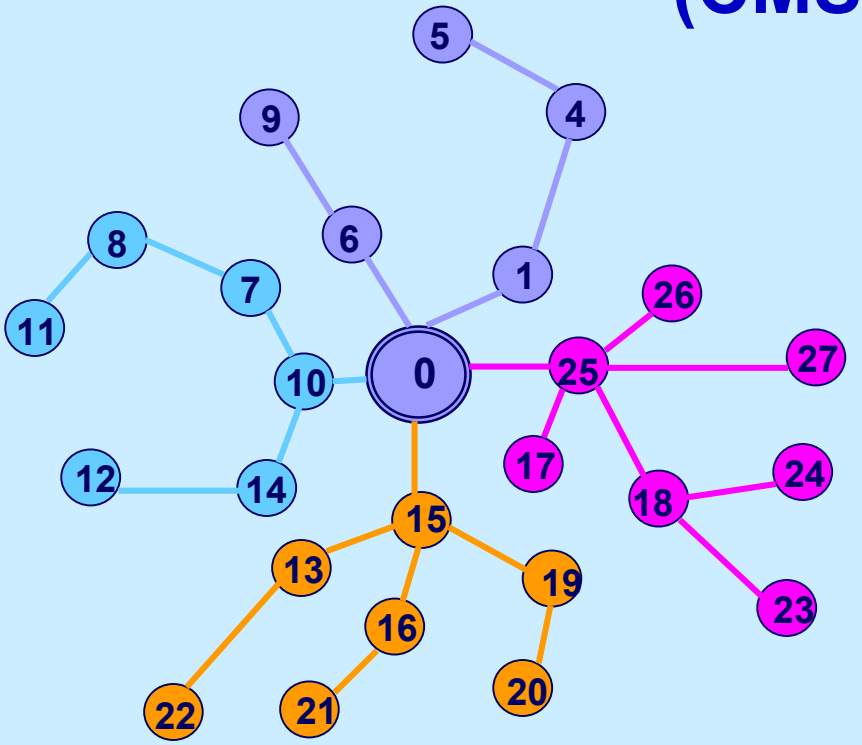
Vehicle Routing Problems, Scheduling Problems, Clustering Problems, and more



Vehicle Routing: Assign trucks to pick up cargoes so as to minimize total travel time and meet service requirements.

Note: $f(S_i)$ may be difficult to compute for a subset S_i.

Capacitated Minimum Spanning Tree Problem (CMST)



No link (arc) carries more than K units of flow (at most K nodes per subtree).

Minimize the total cost of connection.

Features:

A central computer to be connected to a number of terminals

Each terminals has a demand of 1 (homogeneous case)

Some references on CMST

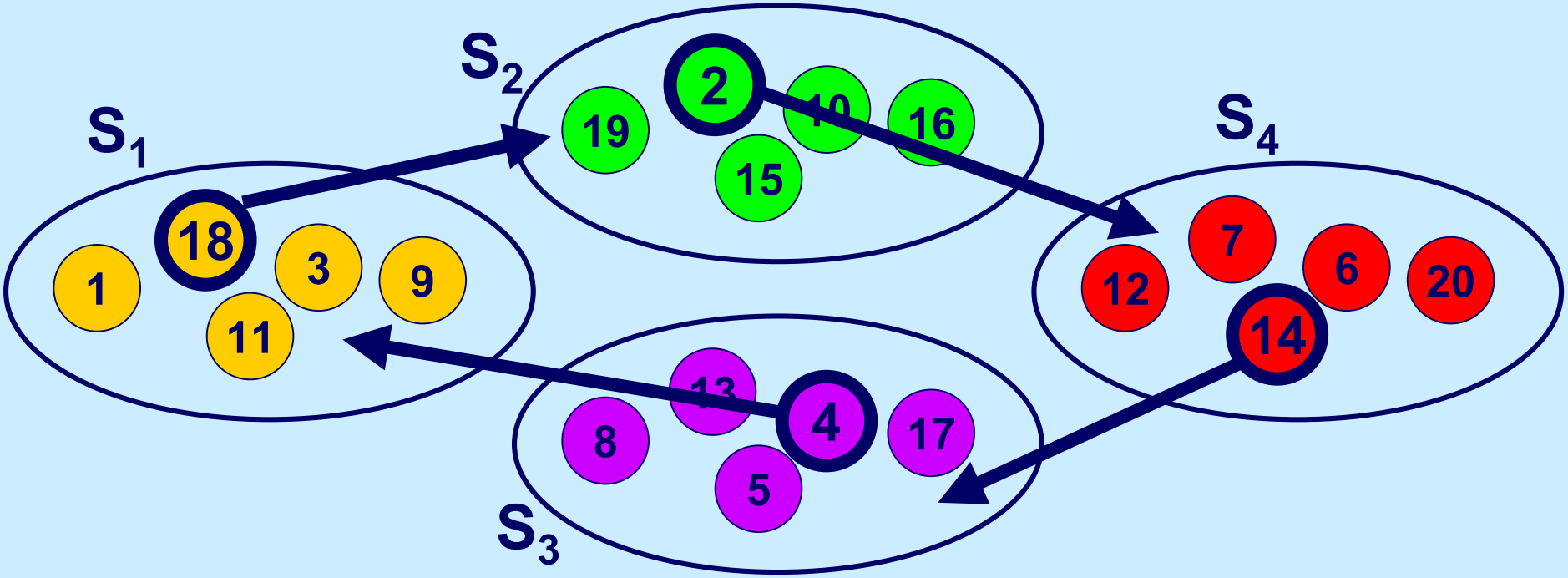
Exact Algorithms

Chandy and Russell [1972], Chandy and Lo [1973], Elias and Fergusan [1974], Kershenbaum and Boorstyn [1983], Gavish [1982, 1983, 1985], Gouveia and Paixao [1991], Malik and Wu [1993], Gouveia [1993, 1995], Hall [1995], Gouveia and Martins [1996]

Heuristic Algorithms

Esau and Williams [1966], Martin [1967], Sharma and El-Bardai [1970], Whitney [1970], Frank et al. [1971], Chandy and Russell [1972], Elias and Ferguson [1974], Kershenbaum [1974], Kershenbaum and Chou [1974], Karnaugh [1976], Gavish and Altinkemer [1986, 1988], Gavish [1991], Sharaiah et al. [1995], and Amberg et al. [1996]

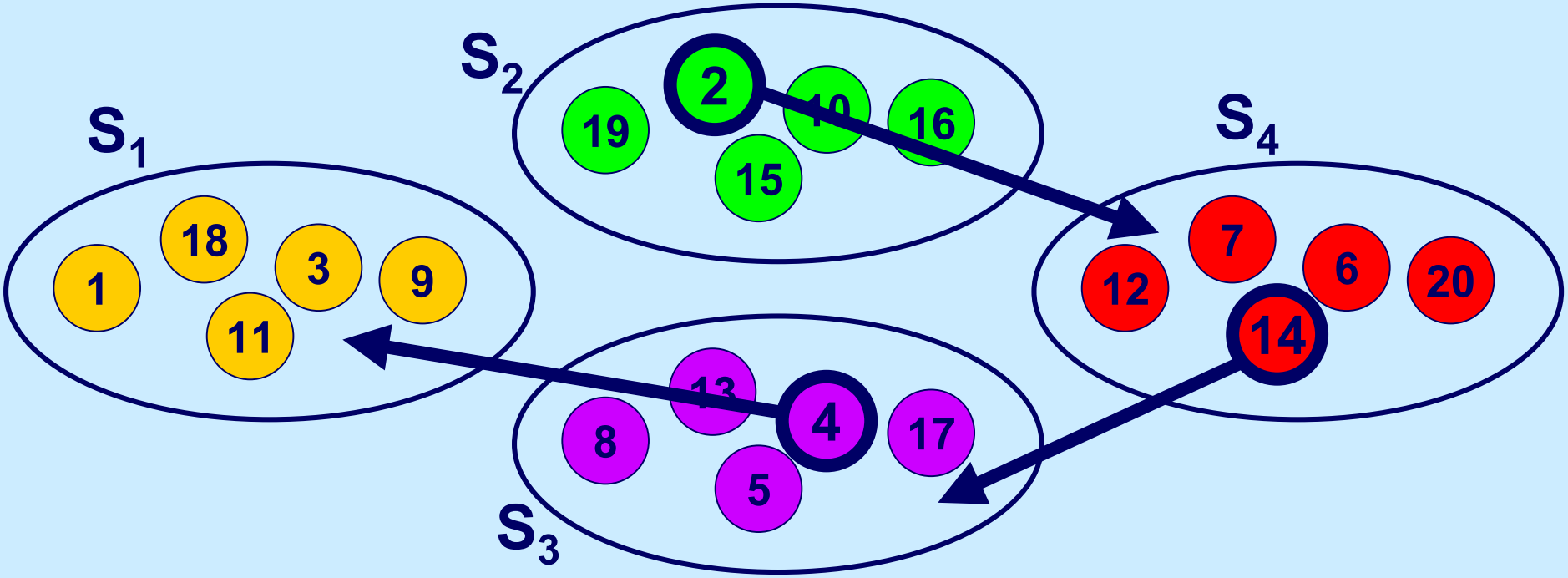
The Cyclic Exchange Neighborhood (Multi-Swap)



Each subset either (i) is unchanged or (ii) has one item inserted and another item deleted

The number of neighbors can grow exponentially in the number of items.

A Path Exchange



After a path exchanges one subset increases by 1 and another decreases by 1

A Simplification for this talk

We consider only cyclic exchanges

Improvement graph: used to identify profitable cyclic exchanges

Suggested originally by

Thompson and Orlin [1989] and

Thompson and Psaraftis [1993]

Thompson [1989]

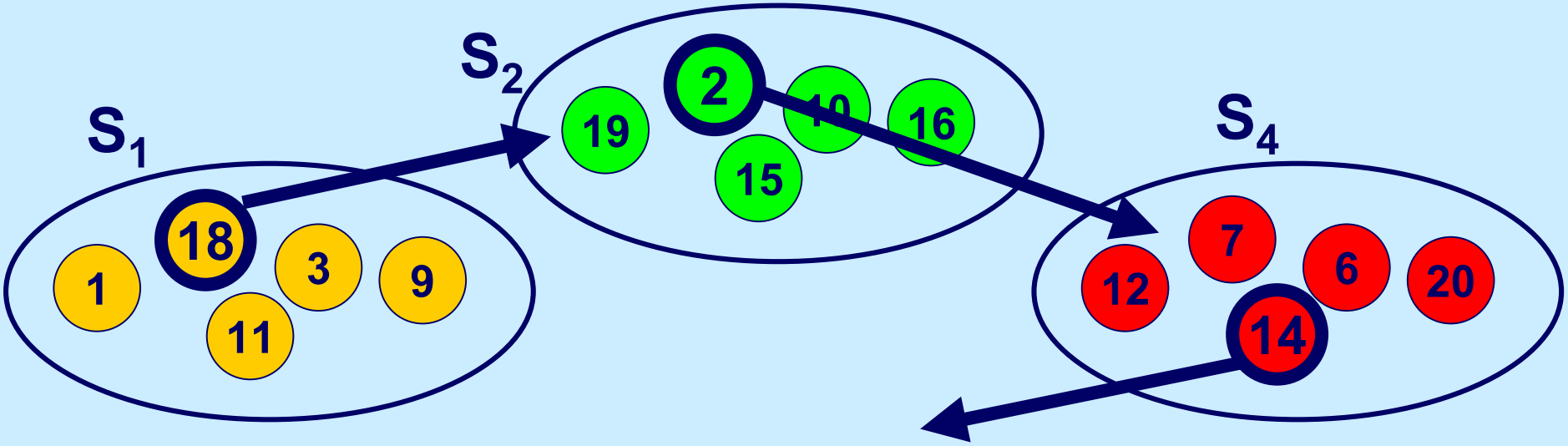
Other References

Ahuja, Orlin, and Sharma [1998, and 2003]

Gendereau, Guertin, Potvin, and Sequin [1998]

Maria G. Scutella, Antonio Frangioni, Emiliano Necciari [2000]

The Effect of a Cyclic Exchange, Multi-Swap



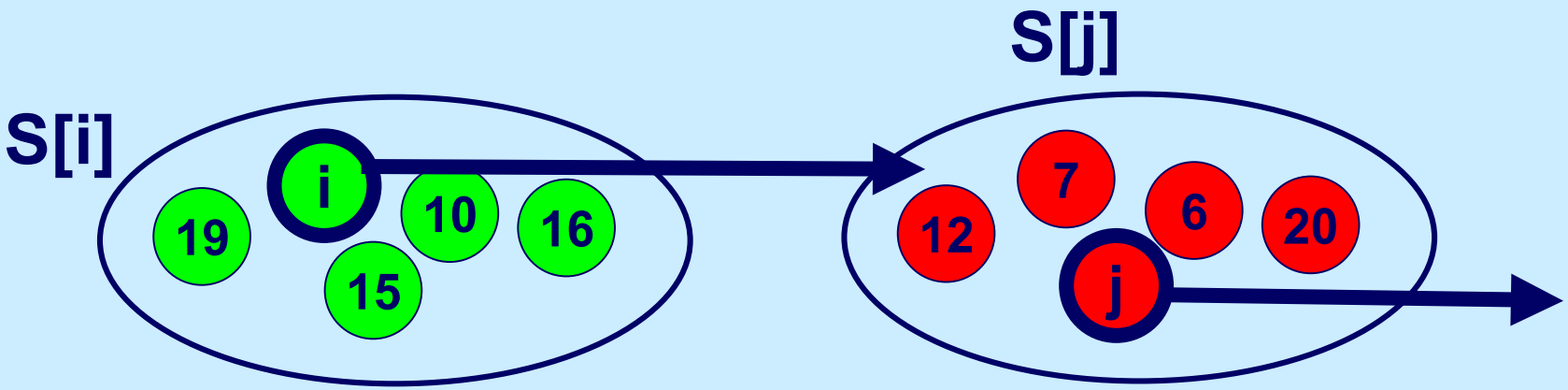
Let S_2 be the 2nd subset prior to the exchange.

Let S'_2 be the subset after. Then $S'_2 = S_2 + \{18\} - \{2\}$.

Let $c_{18,2} = f(S_2) - f(S_2 + \{18\} - \{2\})$ = increase in cost of 2nd part after the exchange.

Let $c_{2,14} = f(S_4) - f(S_4 + \{2\} - \{14\})$ = increase in cost of 4th part after the exchange.

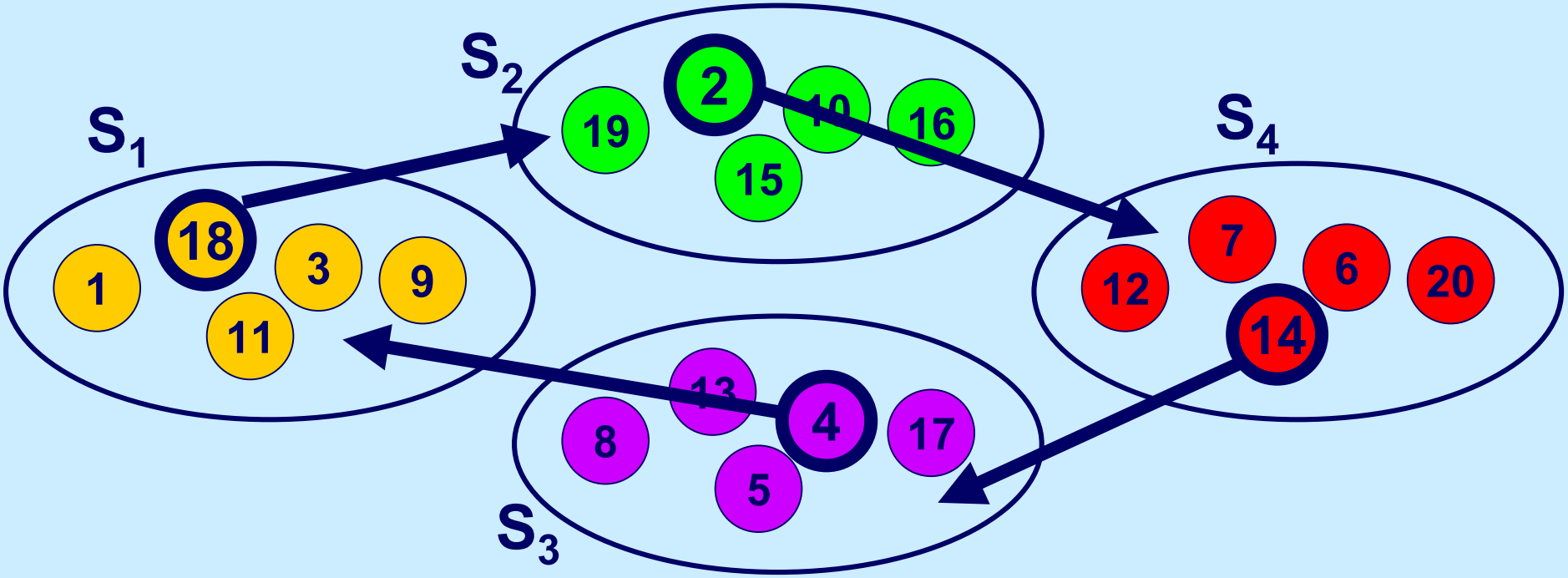
The improvement Graph G



Create a node in G for each element in the partition

Let $c_{i,j}$ = f(S[j]) – f(S[j] + {i} – {j}) = increase in cost of part S[j] after an exchange that moves i to S[j] and moves j out.

About the improvement graph



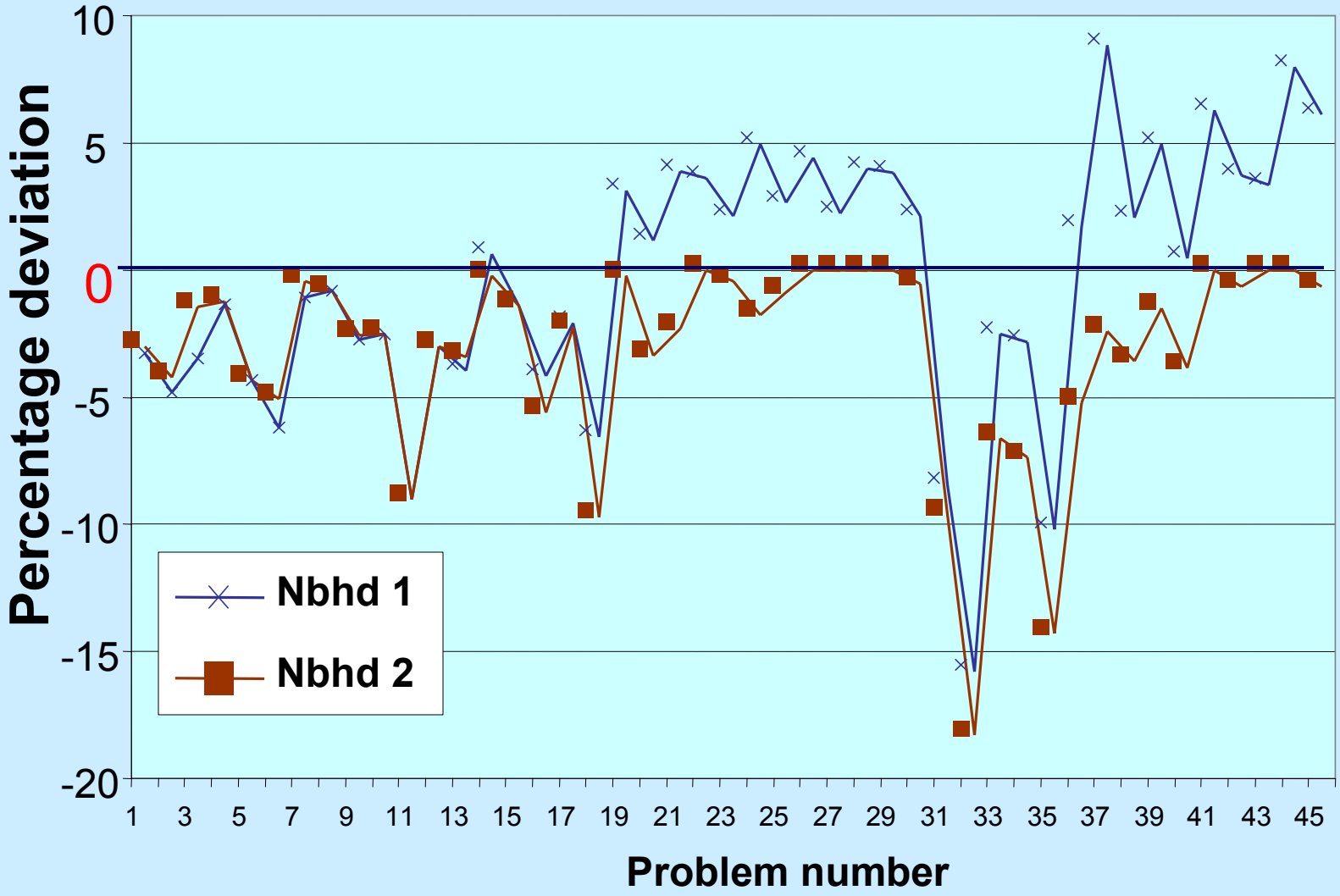
Let "increase in the cost" from the exchange is the cost of the corresponding cycle in the improvement graph. A negative cost cycle leads to an improving exchange.

"Increase" is $c_{4,18} + c_{18,2} + c_{2,14} + c_{14,4}$

The Improvement Graph

- There is a one-to-one correspondence between cyclic exchanges and "<u>subset-disjoint</u>" directed cycles in G. (At most one node enters or leaves per subset).
- A negative cost subset-disjoint cycle in G defines an improving cyclic exchange.
- Identifying a negative cost subset-disjoint cycle is NP-hard.
- We can find negative cost subset-disjoint cycles effectively in practice using either heuristics or implicit enumeration.

Data for the CMST Problem. Ahuja, Orlin, Sharma [1998]



Comparison on heterogeneous demand problems.

Airline Scheduling

Schedule Generation

Fleet Assignment

Through Assignment

Maintenance Routing

Crew Scheduling

Airline Fleet Assignment Model

Assign planes of different types to different flight legs so as to minimize the cost of assignment

- Flight coverage and aircraft integrality
- Aircraft balance
- Fleet size for each of K different fleets
- Possibly: connections, maintenance, and more

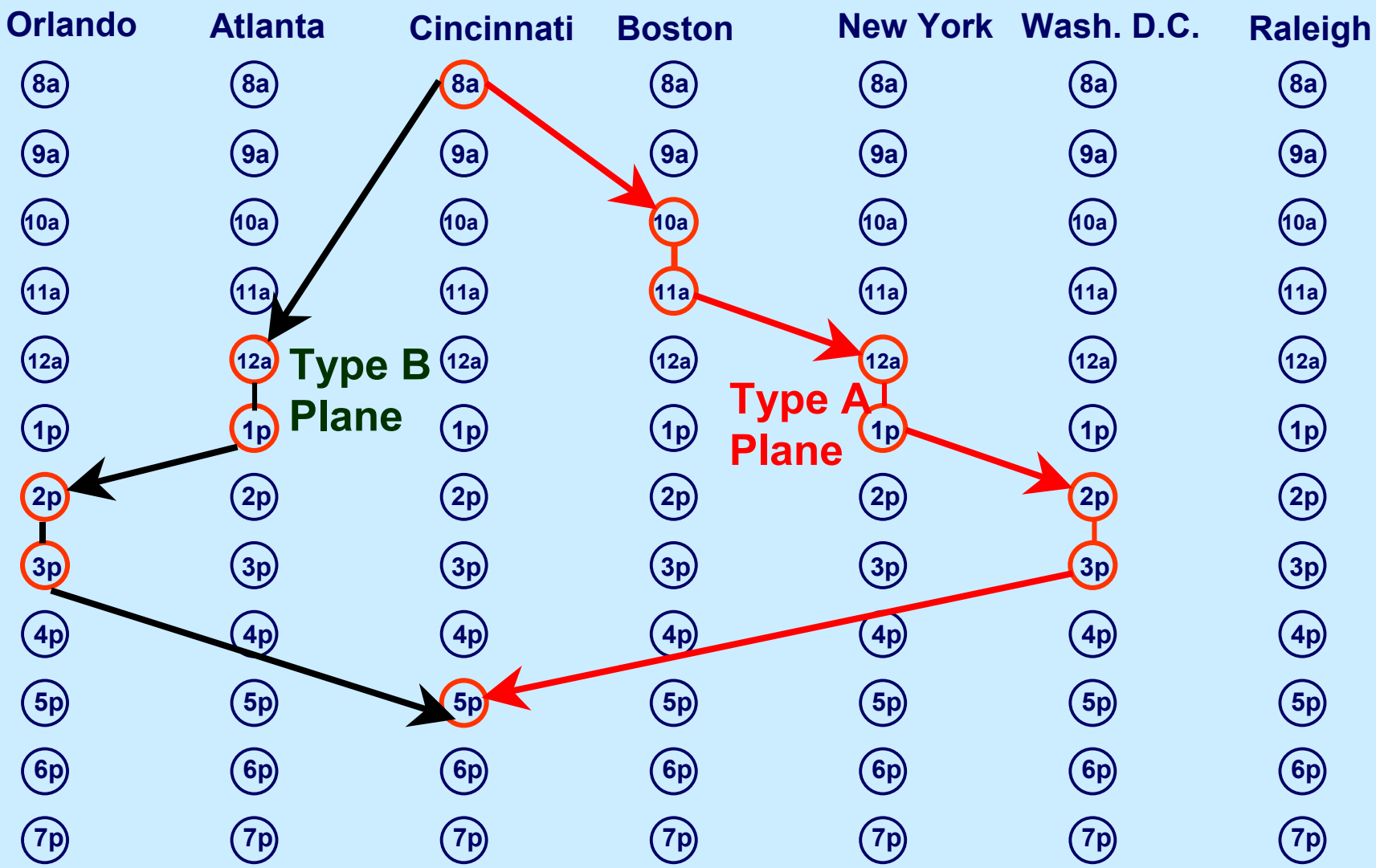
On Neighborhood Search

Usual neighborhood search does not seem appropriate

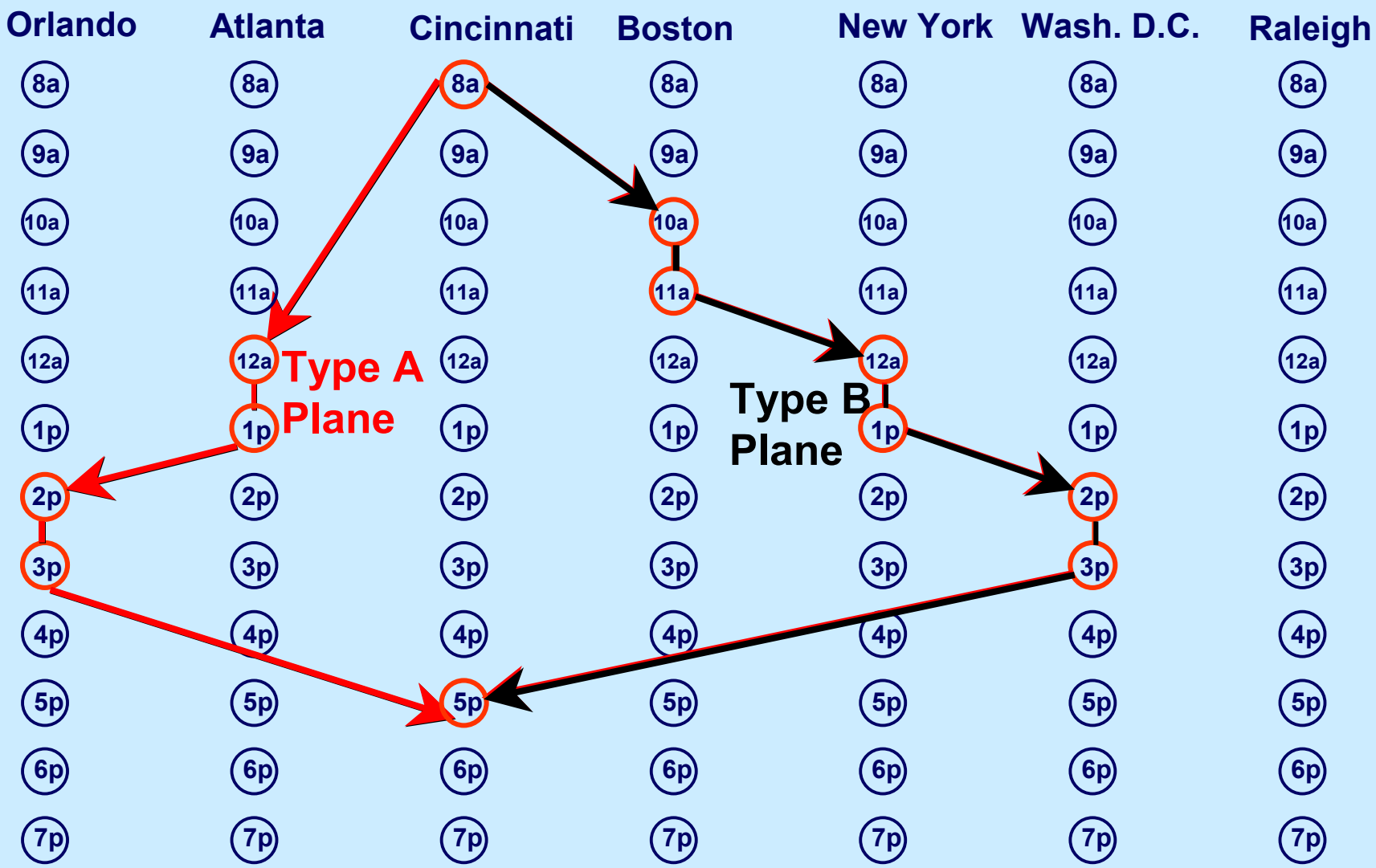
One cannot swap a fleet type on one leg for a fleet type on another leg.

But we can swap a fleet type on a sequence of legs for a fleet type on other legs.

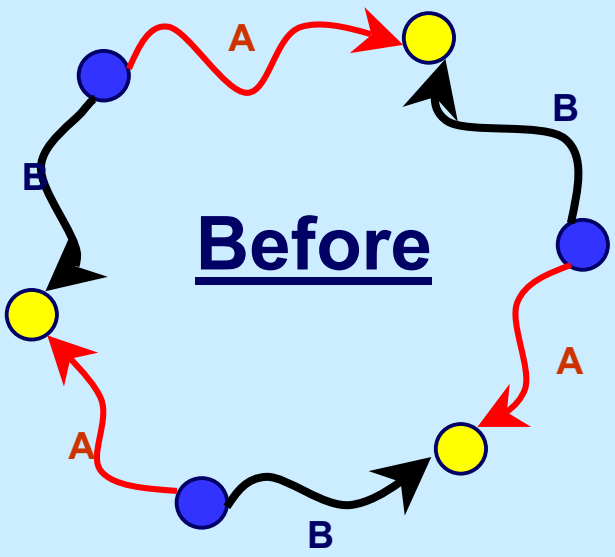
Single A-B Swaps (before)

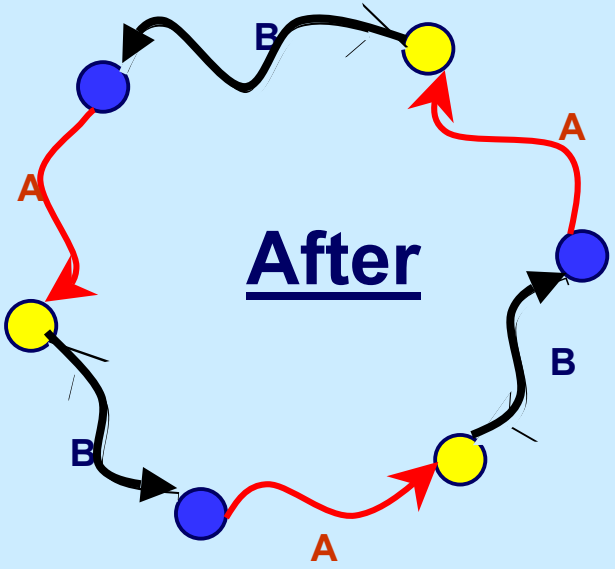


Single A-B Swaps (after)



Multi A-B Swaps

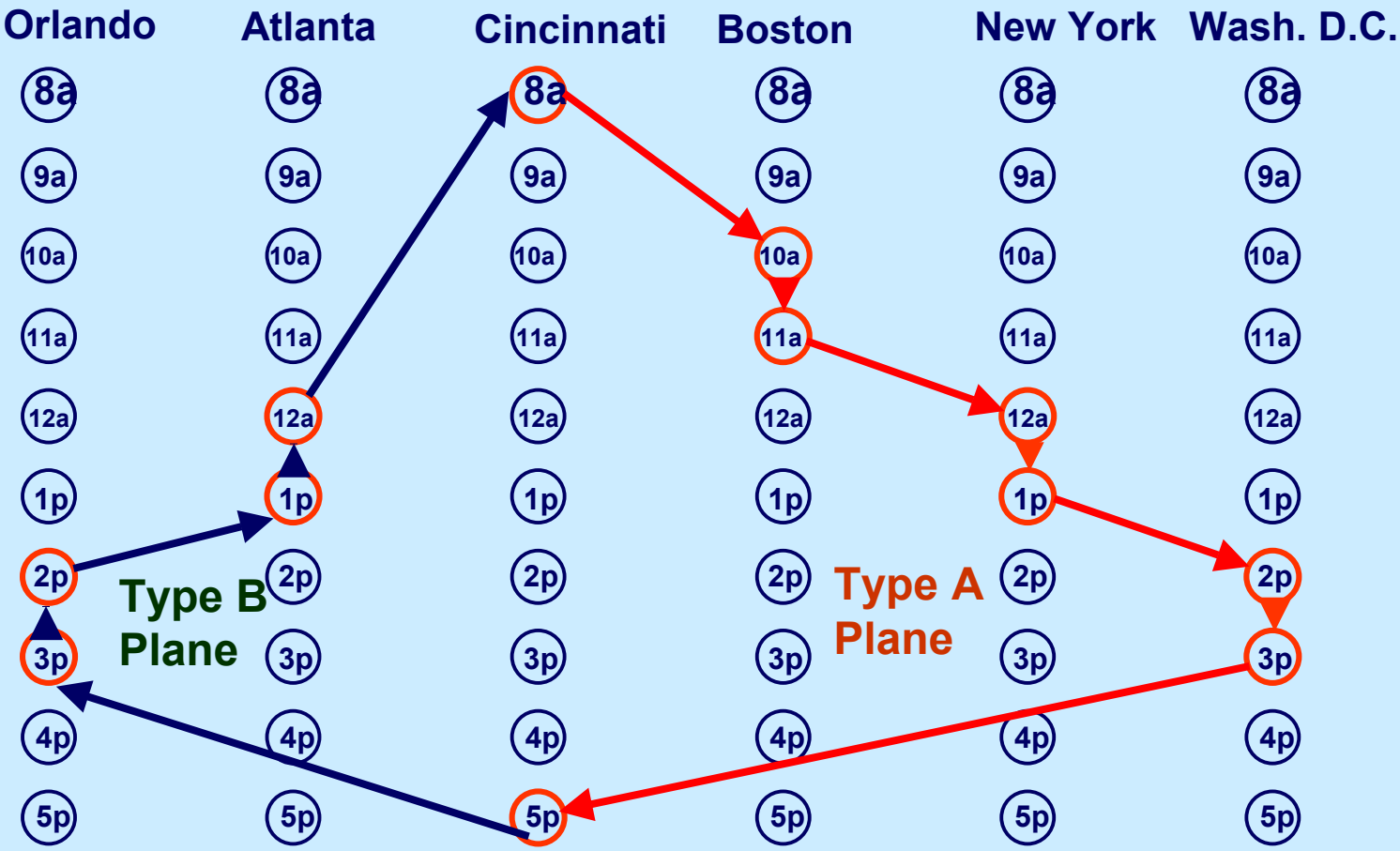




- 1 red path directed out and 1 black path directed out
- 1 red path directed in and 1 black path directed in

reversing B arcs leads to a directed cycle.

A-B Graph (Talluri [1996])



Construct the A-B Graph, consisting of legs flown by planes of types A and B. Reverse directions for arcs of type B. Cost of (i,j) is the cost of moving i from one type to another.

Computational Results: Research with United Airlines

Ahuja, Orlin, Sharma [2000]

1609 Flights Legs, 13 Fleet types.

Initial solution: guaranteed to be within .1% of optimal.

	Increase in Fleeting Profits per year	Increase in Through Profits per year	Increase in Total Profits per year	Running Time
Local Search	-$0.7 million $3.2 million if sole objective	$27.8 million	$27.1 million	20-30 secs
Tabu Search	-$1.7 million $3.2 million if sole objective	$30.3 million	$28.6 million	15-20 min

***Through flight*: direct connecting flight with the same flight number**

Neighborhoods Based on Polynomial Time Algorithms for Special Cases

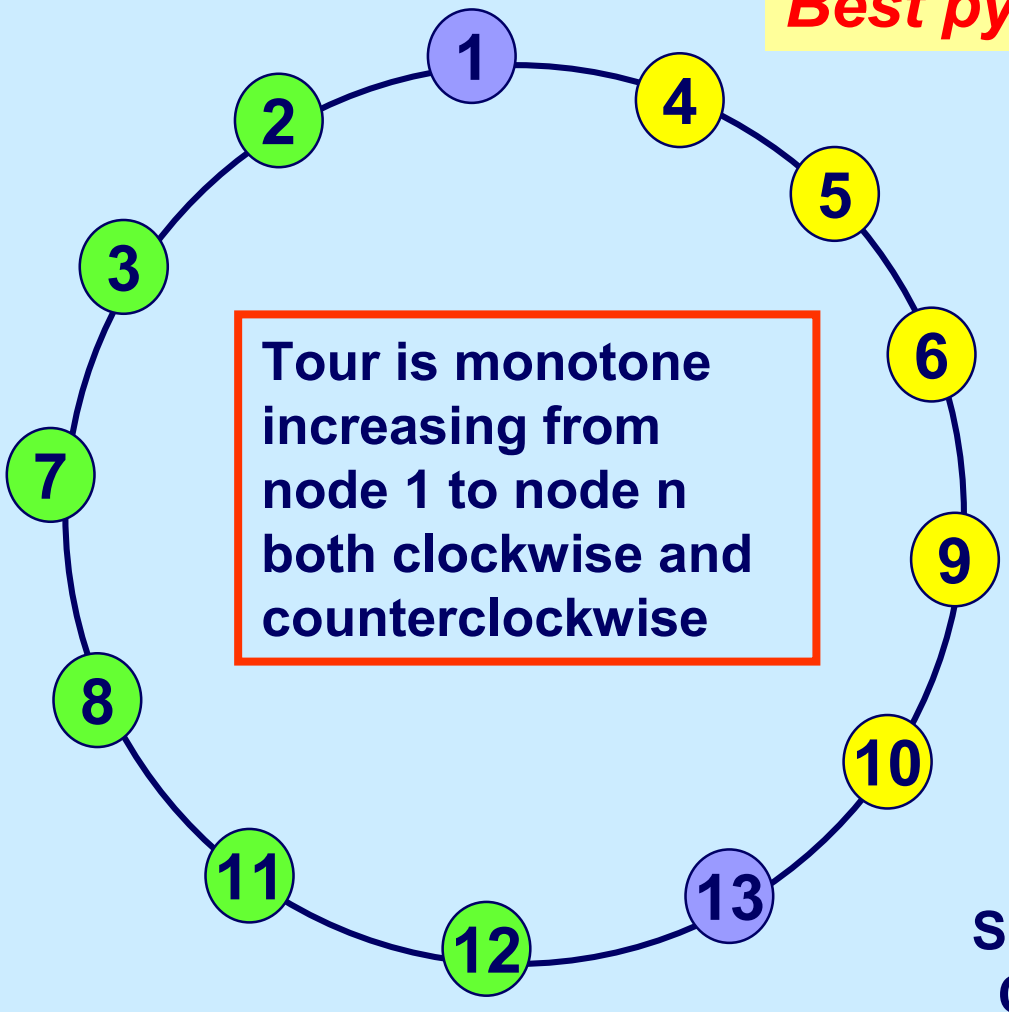
Vast literature based on polynomially solvable special cases.

Methodology: turn a special case into a neighborhood.

Illustrations:

- **Pyramidal Tours**
- **Halin Graphs**

Pyramidal Tours and Neighbors



Tour is monotone increasing from node 1 to node n both clockwise and counterclockwise

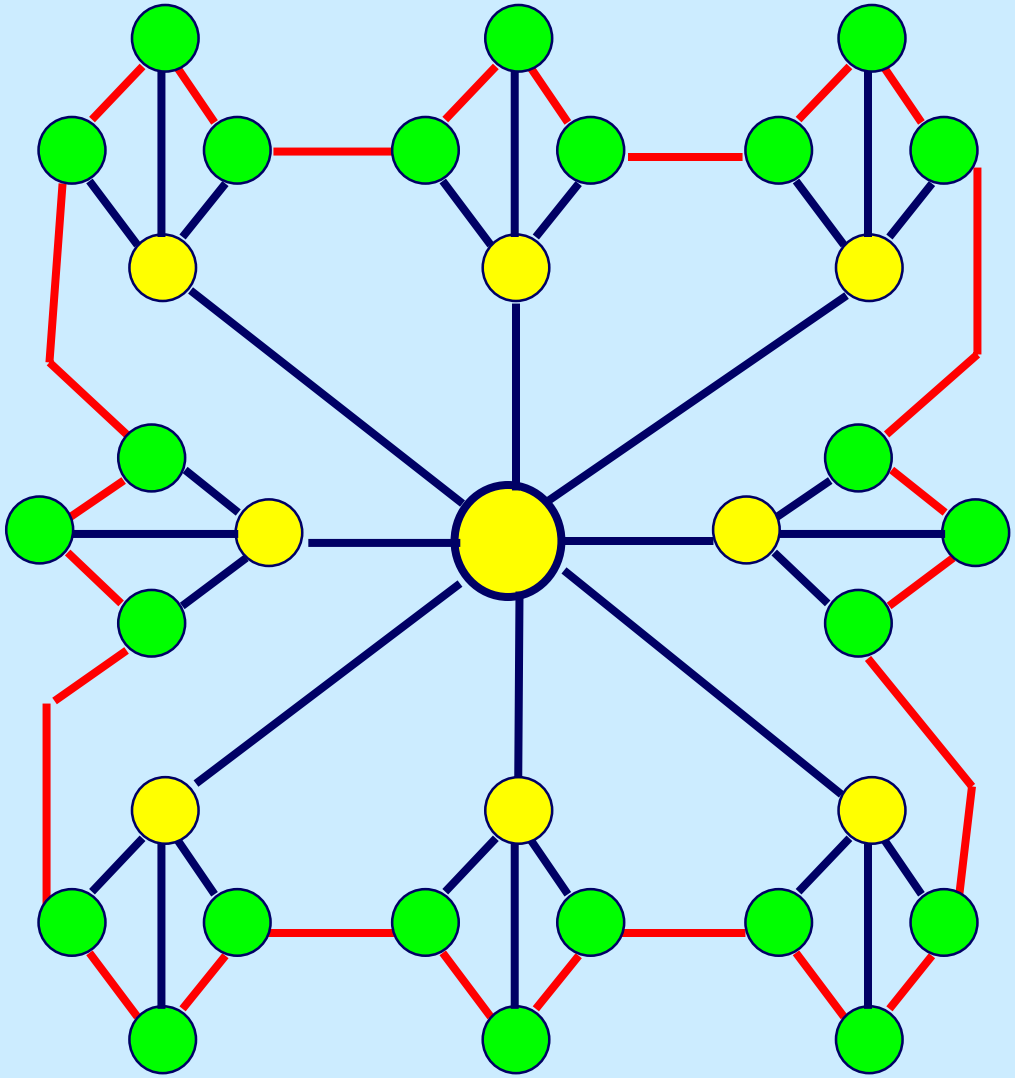
Best pyramidal tour : $O(n^2)$ time.

Pyramidal neighbor: a tour T' that is pyramidal if the initial tour is 1, 2, ..., n

Best pyramidal neighbor : $O(n^2)$ time.

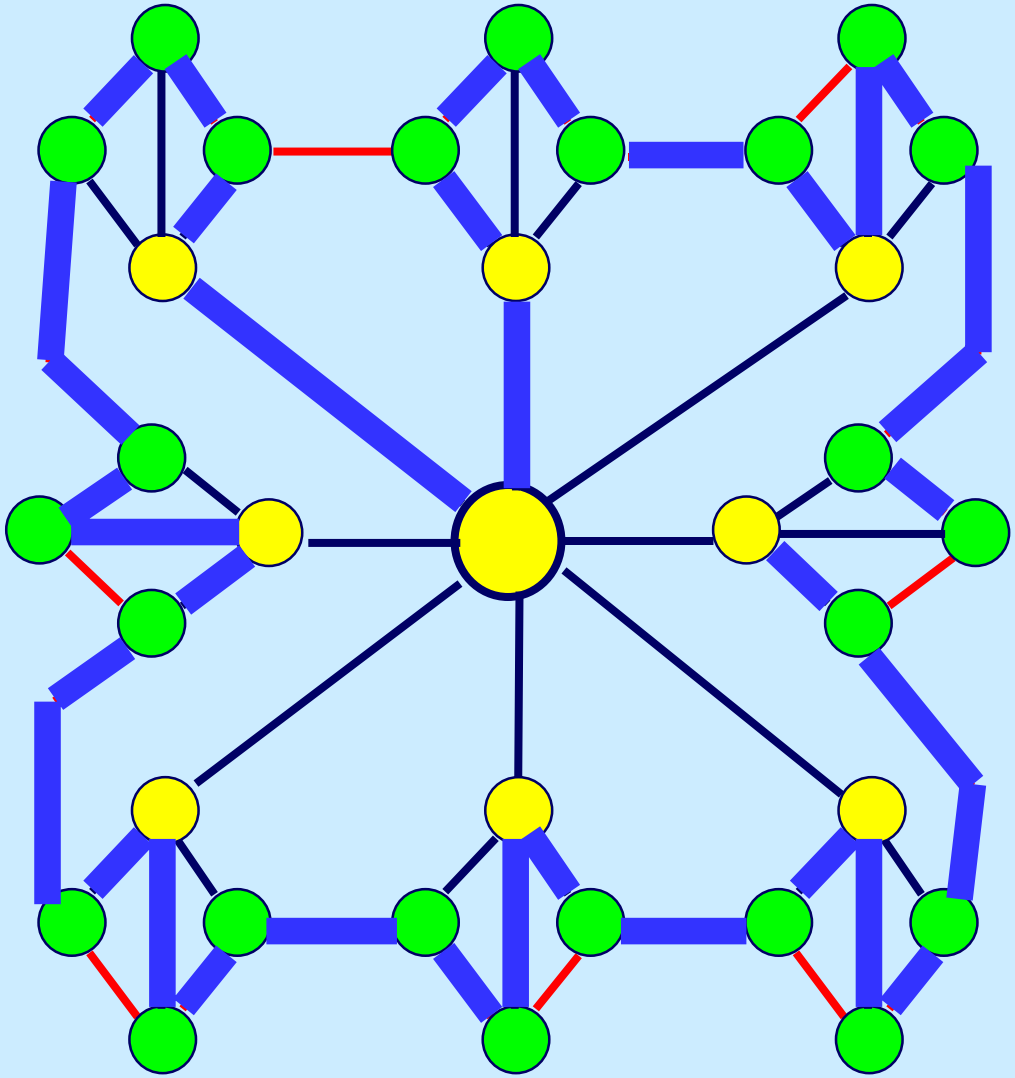
Pyramidal neighborhood and extensions:
Sarvanov and Doroshko [1981]
Carlier and Villon [1990], and
Burkard, Deineko, and Woeginger [1998].

Halin Graphs



A *Halin Graph* is a tree whose nodes do not have degree 2, and the leaves are connected in the plane to form a cycle.

Halin Graphs



One can find min cost tours in *Halin Graph* in O(n) time.

Cornuejols, Naddef, and Pulleyblank [1983]

Halin Neighbors

Approach: embed a Tour inside a Halin graph, and optimize over the neighborhood.

Suggested by Punnen

Requires: a polynomial time algorithm for embedding T in a Halin Graph

A Generic Approach

Let X be an *NP-hard* problem.

Let X' be a restriction of X, where X $\in$ *P*.

Develop a subroutine CreateNeighborhood(S)

INPUT: feasible subset S, for (F, f) $\in$ X

OUTPUT: Instance (F', f) $\in$ X', with F' $\subseteq$ F, and S $\in$ F'

A Generic Approach

INPUT: feasible subset S, for $(F, f) \in X$

OUTPUT: Instance $(F', f) \in X'$, with $F' \subseteq F$, and $S \in F'$

S is feasible for the special case F' that is solvable in polynomial time.

We refer to F' as the ***X'-induced neighborhood*** of S.

The X'-induced neighborhood can be searched in polynomial time.

More on the generic approach

Essential aspect: for a given solution x*, solve a restriction of the original problem for which x* is feasible.

Any solution in the restriction is a neighbor.

Conclusions

- VLSN Search is another tool in our heuristic toolkit
- Key is defining a neighborhood that can be searched efficiently and that leads to good local optima
- Search techniques
  - Network Flow approaches and DP
  - Polynomial algorithms for subproblems
  - IP and DP and … approaches