Very Large Scale Neighborhood Search

By Jim Orlin

Presented at ISMP 2000

Collaborators:
Ravi Ahuja, Ozlem Ergun,
Abraham Punnen, Dushyant Sharma

Neighborhood search

***Combinatorial Optimization*: minimize (f(S) : S ? F)**

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

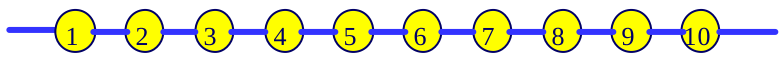
***Neighborhood Function*:**

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

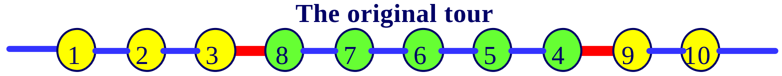
Neighborhood Search (local improvement algorithm)

begin
    initialize with some S ? F;
    while* S is not a local optimum *do
        replace S by some T ? 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.

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

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)

A Neighborhood Search Technique has Three 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.**

What is very large scale neighborhood (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.

- **exponentially large neighborhoods**
- **polynomial time search algorithm**
- **select the optimal solution in the neighborhood**

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

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

Overview of this talk

Examples of VLSN Search?

- **Neighborhoods searched using Network Flow and/or DP-based methods.**
  - dynasearch, assignment based neighborhoods, cyclic exchange neighborhoods, fleet scheduling neighborhoods
- **Neighborhoods based on polynomially solvable special cases of a hard optimization problem**

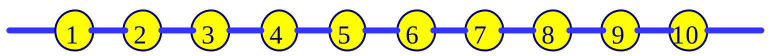
Algorithmic Analysis of VLSN Search

- **Computational complexity of the search algorithms**
- **Size of the neighborhoods**

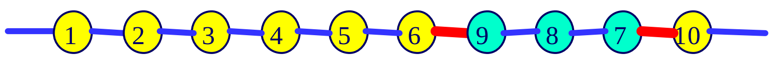
Why is VLSN search of interest to math programmers?

- **Opportunities for interesting algorithmic analysis**
- **potential for improved problem solving**
- **Use in interactive scheduling**

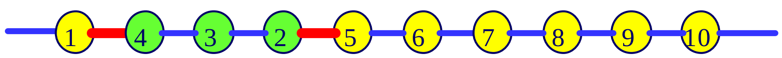
Independent 2-exchanges



The original tour T



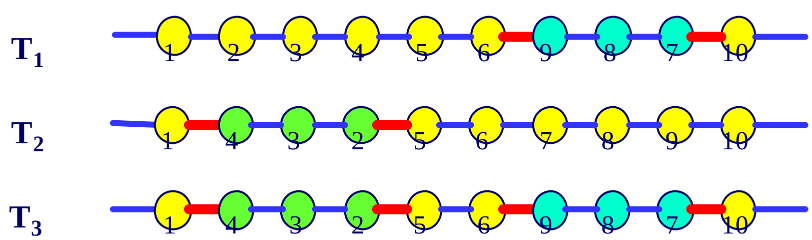
Flip[7,9] and obtain T_1



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



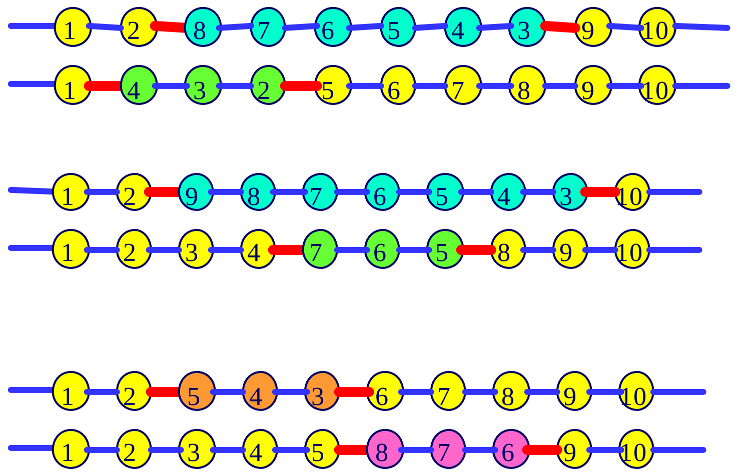
Flips [2,4] and [7,9] are independent

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

$$c(T_3) - c(T_0) = [c(T_1) - c(T_0)] + [c(T_2) - c(T_0)].$$

cost of [7,9] **cost of [2,4]**

Pairs that are not independent



Dynasearch/ Ejection Chains, and more

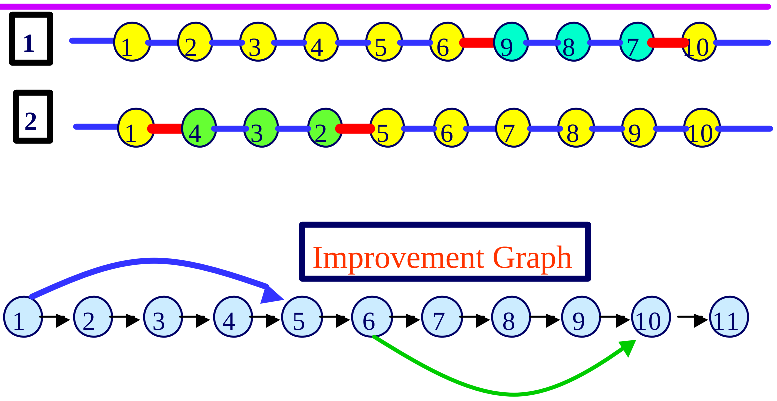
We say that T’ is an independent 2-exchange neighbor of T if T’ can be obtained via k independent 2-exchanges for some k.

Size of neighborhood = ? **($n^{1.7548...}$)**

Dynasearch: **Potts and van de Velde [1995]**
O(n^2) DP algorithm.

Ejection chains: **Glover and [1992, 1996]**
O(n^2) shortest path algorithm.

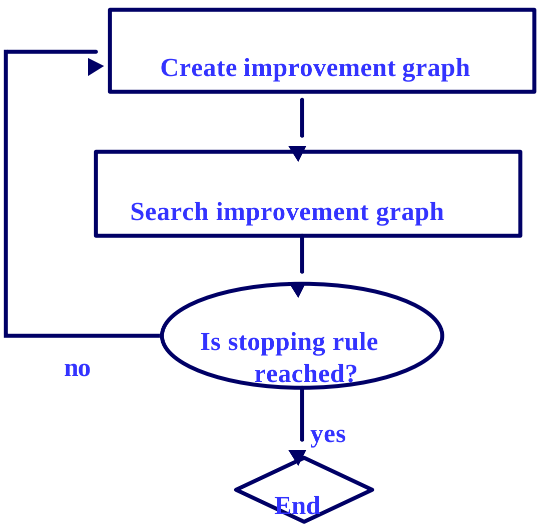
The Improvement Graph



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

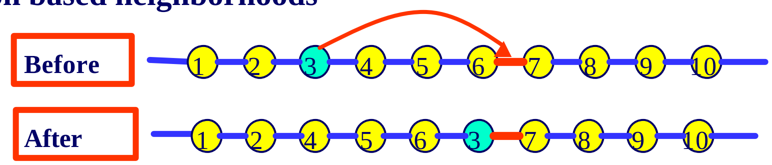
Summary of the Method

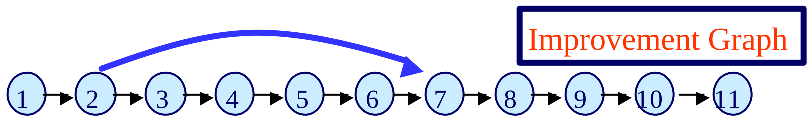


It is easy to incorporate tabu search by adjusting the improvement graph

Other Neighborhoods and Improvement Graphs

Insertion based neighborhoods

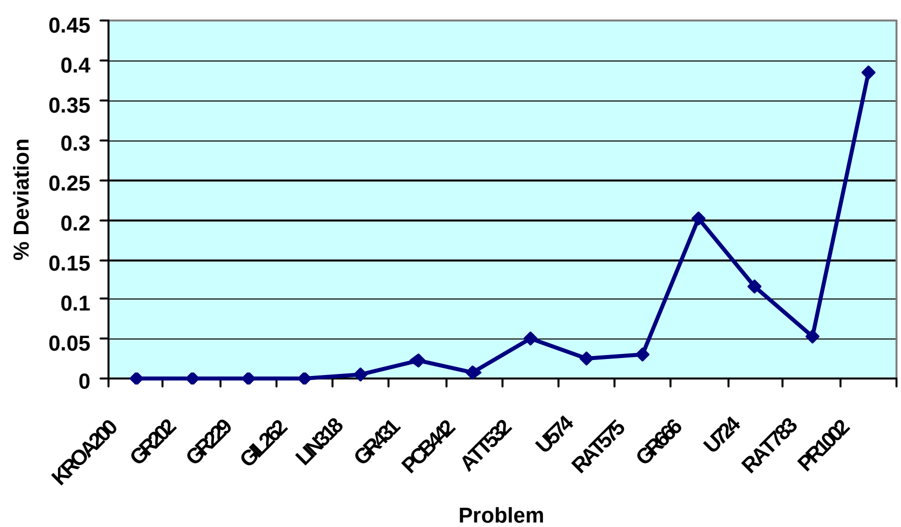




c_{ij}: **cost of keeping city i fixed and city j fixed and inserting city i+1 after city j-1.**

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

Dynasearch with compounded 2-opt neighborhoods



Advantages for VLSN, based on compounded neighborhoods

- **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.**

There are more advantages to follow for other VLSN search algorithms

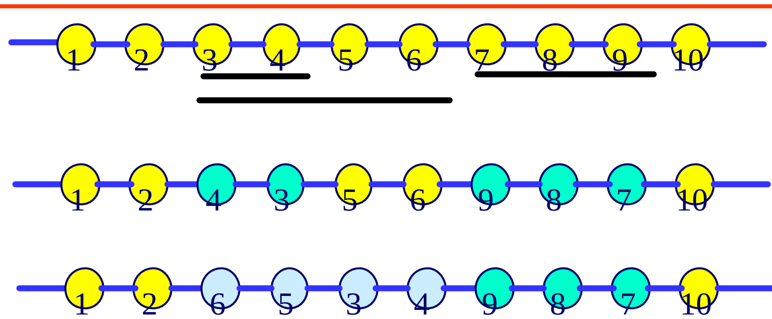
Independence not required: Twisted Neighborhood

Suppose that S is a collection of intervals

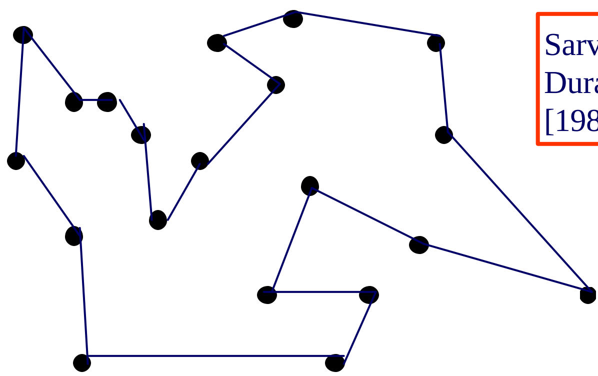
- **If I, J ? S, then I and J do not "cross", that is,**
  I ? J or J ? I or I ? J = ? .

Then from S we obtain a tour by flipping cities in interval, starting with the smallest intervals

Aurenhammer [1988], O(n^7) algorithm



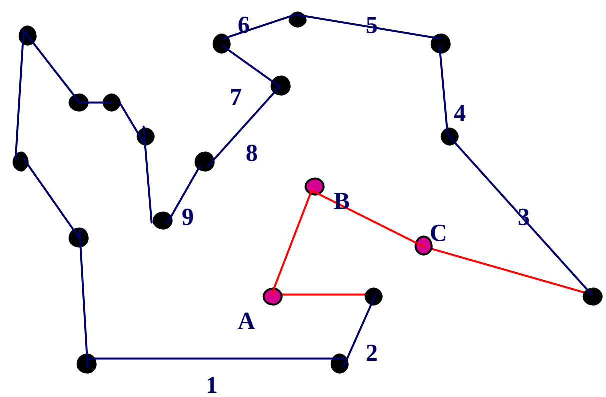
An Assignment Based Neighborhood



Sarvanov and Durashko [1981]

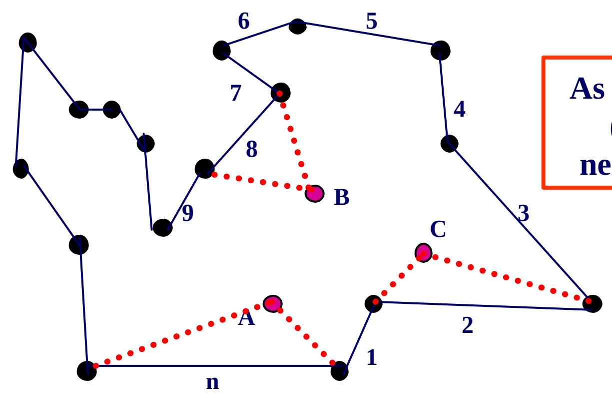
Start with a tour. Select a subset S of cities (possibly cities on a path, possibly chosen some other way).

The Assignment based Neighborhood



Delete the nodes A, B and C from the tour and reinsert so that no two cities of A, B, and C are adjacent.

The Assignment-Based Neighborhood



As many as (n/2)! neighbors

To reinsert the nodes, solve an assignment problem. The cost of (A,n) is the incremental cost of replacing the edge n by edges passing through A.

Cyclic Exchange for Partitioning Problems

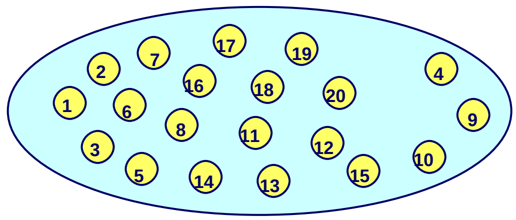
- **S : set of objects**
- **Partition S into subsets (S_1, S_2, S_3, .. , S_p)**
- **Define a (possibly nonlinear) cost function f(S_i) for each subset S_i**
- **Find the partition S_1, S_2, … , S_p such that**

$$? _{p=1,K} f(S_p)$$

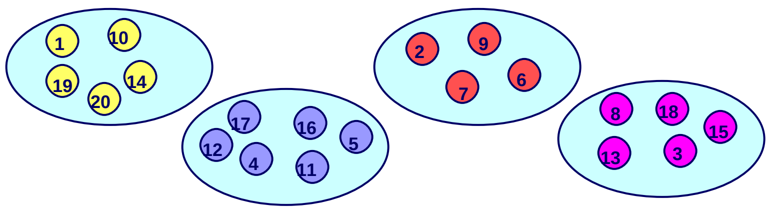
is minimum.

Partitioning Problems

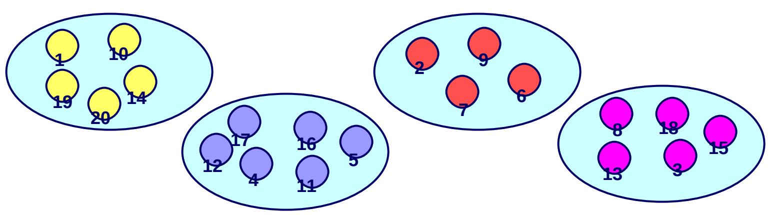
Given a set S of 20 objects



Partition the set into four subsets



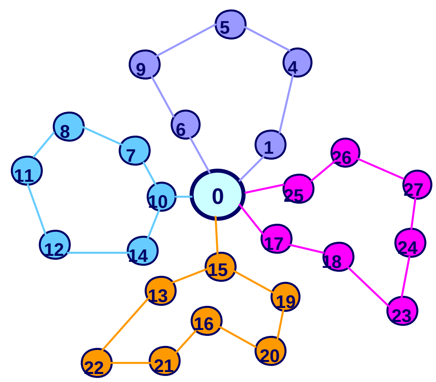
The Cost Structure



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

Find the partition with minimum total cost.

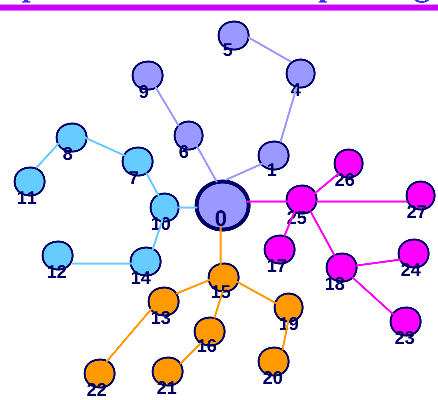
Vehicle Routing Problems, Scheduling Problems, Clustering Problems, and more



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)



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]

Features:

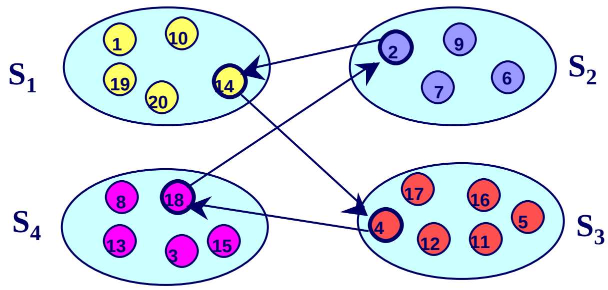
A central computer to be connected to a number of terminals

Each terminals has a demand of 1 (homogeneous case)

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

Minimize the total cost of connection

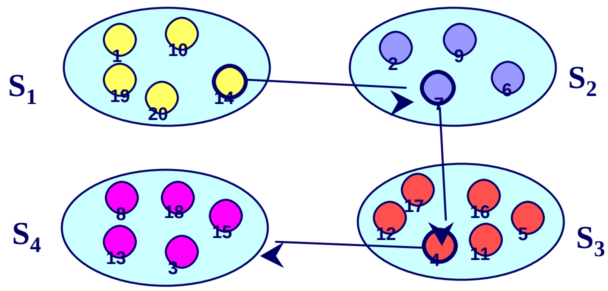
The Cyclic Exchange Neighborhood (Multi-Swap)



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 Mathematical Simplication

Path exchanges can be transformed to cyclic exchanges

- We consider only cyclic exchanges

Improvement graph: used to identify profitable cyclic exchanges

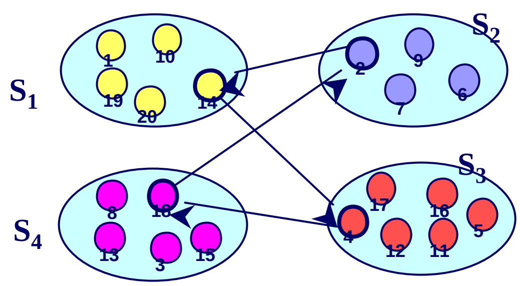
Suggested originally by

Thompson and Orlin [1989] and
Thompson and Psaraftis [1993]
Thompson [1989]

Other References

- Ahuja, Orlin, and Sharma [1998]
- Gendereau, Guertin, Potvin, and Sequin [1998]
- Maria G. Scutella, Antonio Frangioni, Emiliano Necciari [2000]

The Effect of a Cyclic Exchange (Multi-Swap)



Improvement graph.

$c_{2,14} = f(S_1 + 2 - 14) - f(S_1)$
$c_{18,2} = f(S_2 + 18 - 2) - f(S_2)$
$c_{14,4} = f(S_3 + 14 - 4) - f(S_3)$
$c_{4,18} = f(S_4 + 4 - 18) - f(S_4)$

$c_{2,14}$: effect on subset S_1 of adding 2 and deleting 14

$c_{14,4}$: effect on subset S_3 . . . .

$c_{4,18}$: effect on subset S_4 . . . .

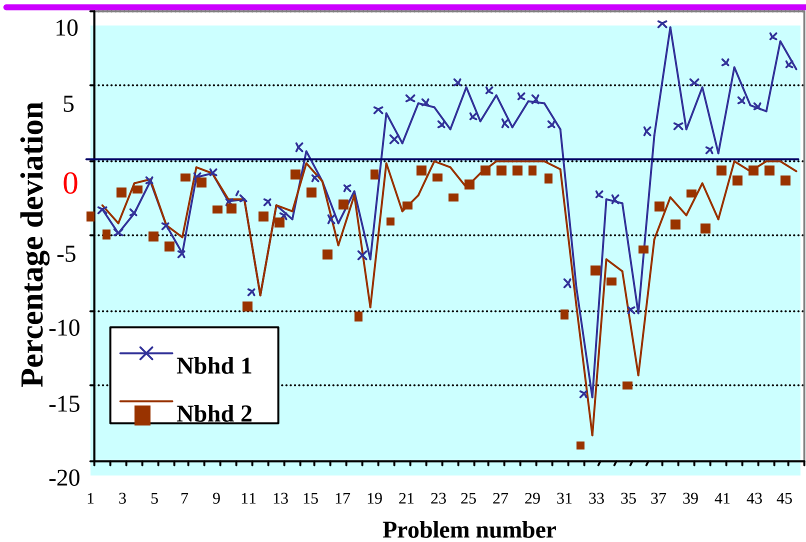
$c_{18,2}$: effect on subset S_2 . . . .

The cost of the cycle is the effect of the cyclic exchange.

The Improvement Graph

- There is a one-to-one correspondence between cyclic exchanges and "subset-disjoint" directed cycles in G.
- 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.

Review of the first few techniques

Can be implemented using improvement graphs

Based on "parallelizing" a simple neighborhood search technique

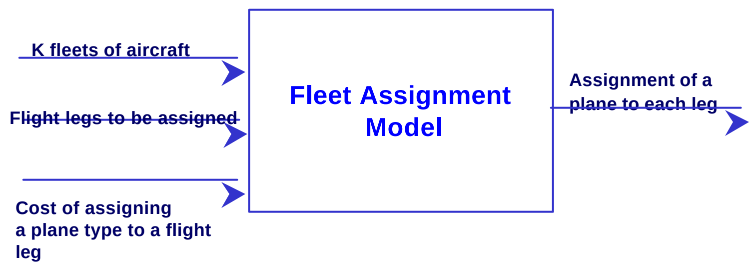
Next two result:

Talluri's neighborhood used in situations involving multi-commodity flows (also uses improvement graphs)

Algorithms based on polynomially solvable special cases of hard problems.

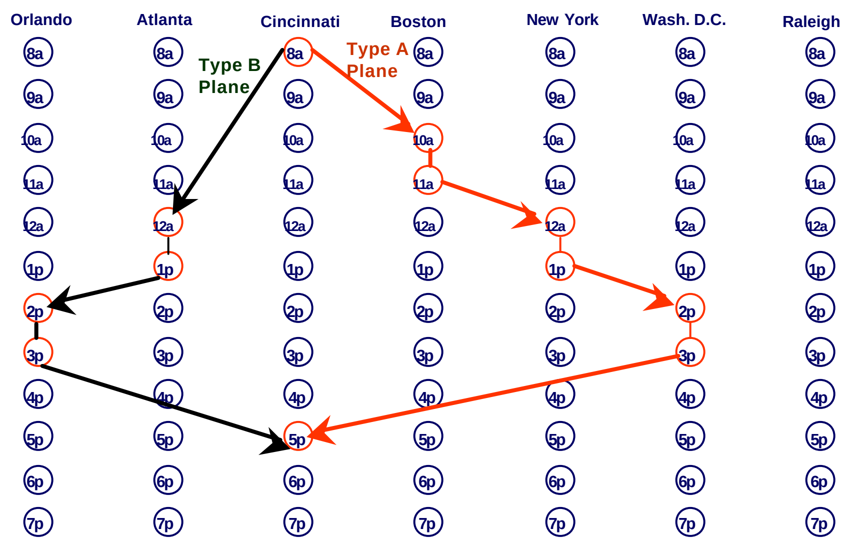
Airline Fleet Assignment Model

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

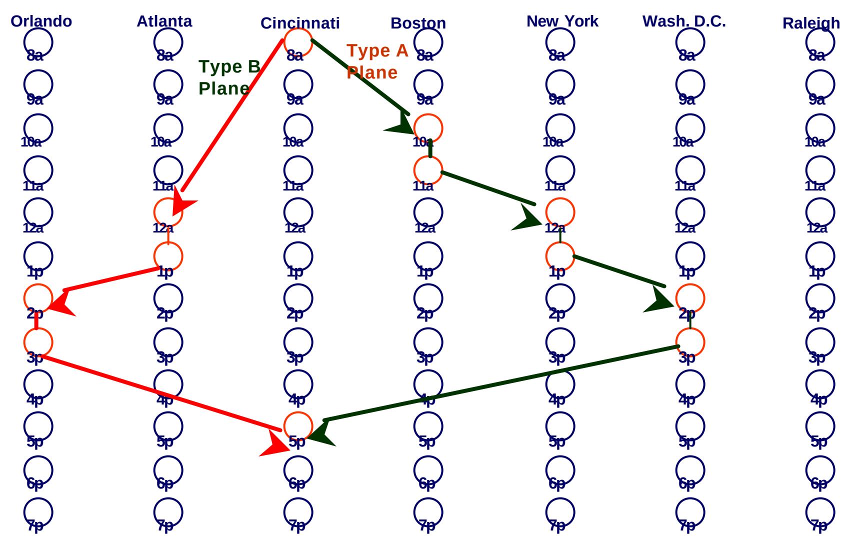


- Flight coverage and aircraft integrality
- Aircraft balance
- Fleet size for each of K different fleets
- Possibly: connections, maintenance, and more

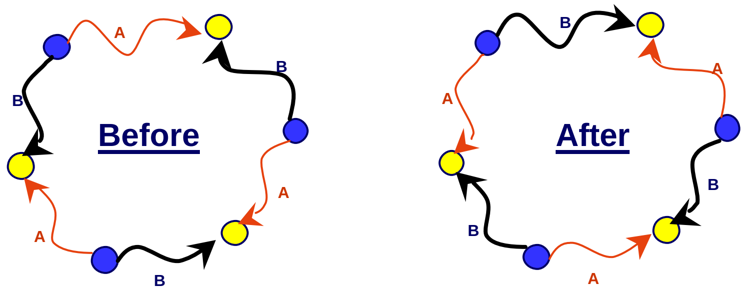
Single A-B Swaps (before)



Single A-B Swaps (after)



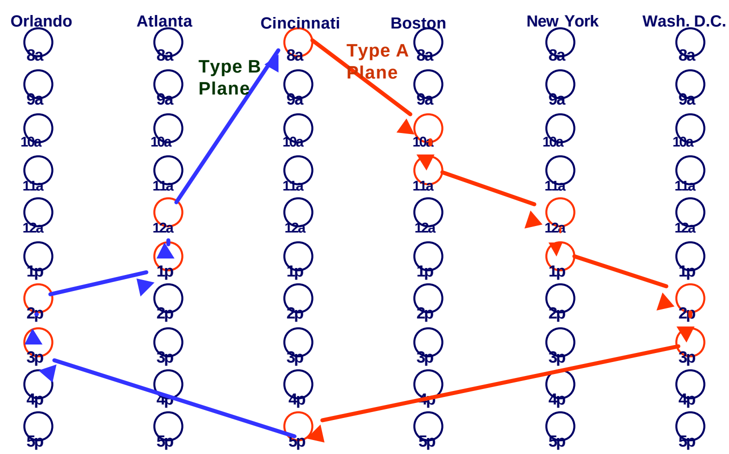
Multi A-B Swaps



- 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.

The use of VLSN in Interactive Scheduling

- Talluri proposed this technique for use in operational scheduling, for adjusting schedules on a real time basis.
- Suppose a scheduler wants to assign a type A plane to leg L, but wants to keep almost all of the current schedule fixed.
- **Idea**: look for an improvement on the A-B network.

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 0.038%	20-30 secs
Tabu Search	-\$1.7 million \$3.2 million if sole objective	\$30.3 million	\$28.6 million 0.041%	15-20 min

Through flight: direct connecting flight with the same flight number

Neighborhoods Based on Polynomial Time Algorithms for Special Cases

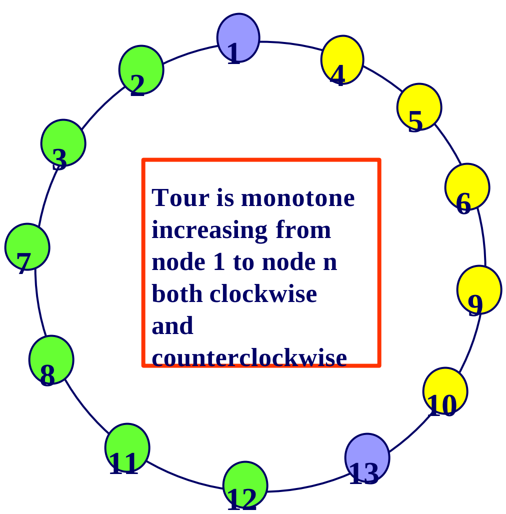
Vast literature based on polynomially solvable special cases.

Idea: turn a special case into a neighborhood.

Illustrations:

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

Pyramidal Tours and Neighbors



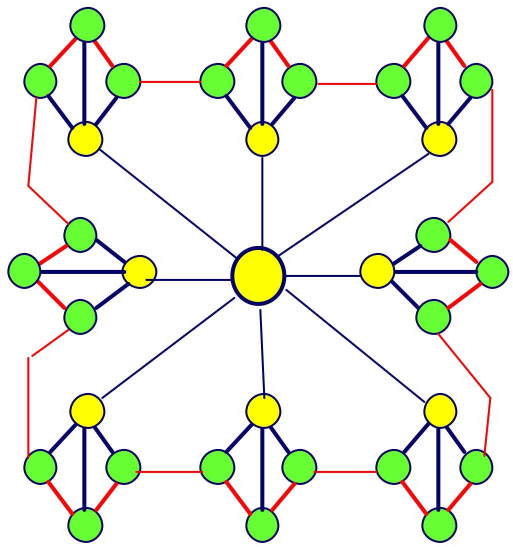
Best pyramidal tour : $O(n^2)$ time.

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

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

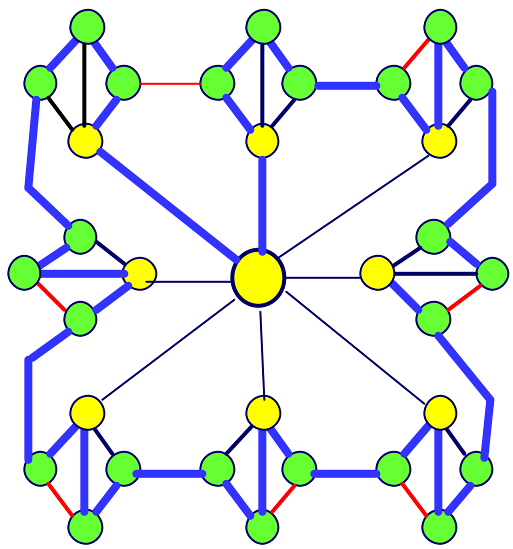
Pyramidal neighborhood and extensions: Sarvanov and Doroshko [1981] Carlier and Villon [1990], and Burkard, Deineko, and Woeginger [1998].

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

Idea: 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

General Approach

Let X be an *NP-hard* problem.
Let X' be a restriction of X, where X ? *P*.

Suppose there is a subroutine CreateNeighborhood(S)

INPUT: feasible subset S, for (F,f) ? X
OUTPUT: Instance (F',f) ? X', with F' ? F, and S ? 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.
Allows researchers to use the literature on polynomial time special cases.

Summary of Neighborhood Search Results

Network Flow and DP based Algorithms

- Dynasearch and variants
- Assignment based neighborhoods
- Cyclic exchange neighborhood
- Fleet assignment neighborhood

Neighborhoods based on efficiently solvable subproblems

- Pyramidal Neighborhoods
- Halin Graph Neighborhoods

- Final Topic: restricted dynamic programming, an intriguing connection to dynasearch

Dynasearch and Restricted DP

Dynamic Programming for TSP

Let C(S,k) be the minimum cost of a tour starting at node 1, terminating at node k, and passing through subset S. (1,k ? S)

$C(S,k) = \min (C(S/k, j) + c_{jk} : j ? S)$

Let ***Q**** be a collection of pairs (S,k) where S ? {1, ..., n}, and k is a city.

Restricted DP is a heuristic. (State-space truncations)

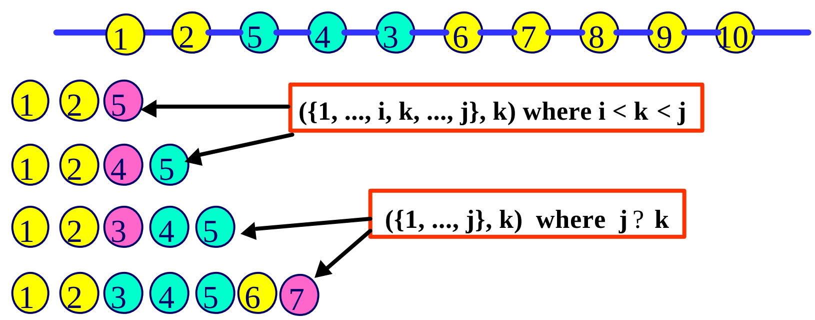
$C(S,k) = \min (C(S/j, j) + c_{jk} : j ? S$, and $(S, k) ? Q^*$, and $(S/j, j) ? Q^*)$

Restricted DP Based on 2-exchanges.

Ergun and Orlin [2000]

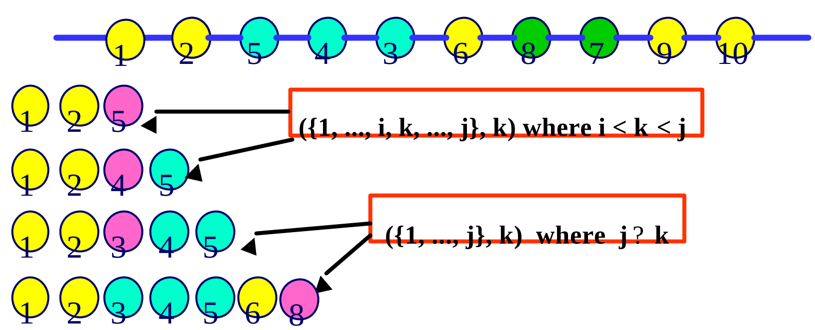
Let Q* = { (S, k) : S are the first cities visited if a 2-exchange is made, and k is the last city visited of S}.

***neighborhood induced restrictions*.**



Restricted DP

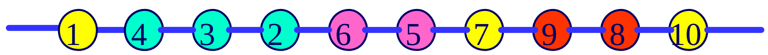
- The same set Q* is obtained if one considers independent 2-exchanges



Restricted DP again

Algorithm. Solve the DP restricted to states in Q*

- **equivalent to optimizing over a neighborhood that includes independent 2-exchanges.**
  - **2-exchanges that can be ordered from left to right.**
- **running time: $O(n^3)$**



This approach can be extended. It works for insertion based neighborhoods, swap based neighborhoods, and more.

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 approaches**
  - **DP with neighborhood induced state space restriction**
- **Appealing to algorithmic researchers**
- **Useful in interactive scheduling**
- **Can be very effective in practice**