Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

MSE Algorithms

L04: Decomposition Methods, Learning Methods: Tabu Search

Samuel Beer

Course Overview

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

1. Introduction: Basic problems and algorithms
2. Constructive methods: Random building, Greedy
3. Local searches
4. Randomized methods
5. Threshold accepting, Simulated Annealing
6. **Decomposition methods: Large neighborhood search**
7. **Learning methods for solution improvement: Tabu search**
8. Learning methods for solution building: Artificial ant systems
9. Methods with a population of solutions: Genetic algorithms

Lecture 4:
Decomposition and Learning Methods

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Goal:

Know sufficiently many methods to start tackeling Santa's Challenge.

- Decomposition Methods
- Learning Methods: Tabu Search
- Inbetween: Quadratic Assignment Problem (QAP)
- Santa's Challenge

Recap: Solving Optimization Problems with Meta Heuristics

Recap: Summary

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Meta-Heuristics for Optimization Problem:

1. Use a greedy **Constructive Method** to create an initial solution (e.g. Nearest Neighbour for TSP)

2. Use a **Local Improvement Method** to improve the solution (e.g. 2-opt for TSP)

3. Apply a **Selection Rule** to decide which improving move is taken (e.g. Best Improving Move)

4. Use **Randomized Methods** (e.g. Simulated Annealing) to explore larger search spaces than purely greedy methods

Decomposition Methods

Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Sizes of Problem Instances:

Class	Typical technique	Size (order)
Toy	Complete enumeration	10^1
Small	Exact method	10^1 - 10^2
Medium	Meta-heuristics	10^2 - 10^4 (memory limit $O(n^2)$)
Large	Decomposition techniques	10^3 - 10^7
Very Large	Distributed database	above

Decomposition techniques/algorithms:

- **Very Large Neighbourhood Search (VLNS)**
- **Popmusic: A generic decomposition technique**

Very Large Neighbourhood Search (VLNS)

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Basic Ideas:

- Start with a reasonable solution
- Neighbourhood size is too large to fully explore (e.g. n^k, 2^n or variable size)
- Partially explore the large neighbourhood to find improvements

Examples:

- **Integer Linear Programming**: Fix the value of a subset (a majority) of variables, solve optimally the sub-problem on the remaining variables. Then repeat with other subsets of fixed variables.
- **Iterated Local Search**: Randomly perturb the best solution so far, and apply an improving method to the perturbed solution
- **Lin-Kernighan algorithm for TSP:** Exchange a variable number (of up to all *n)* edges of a given solution (see next slide).

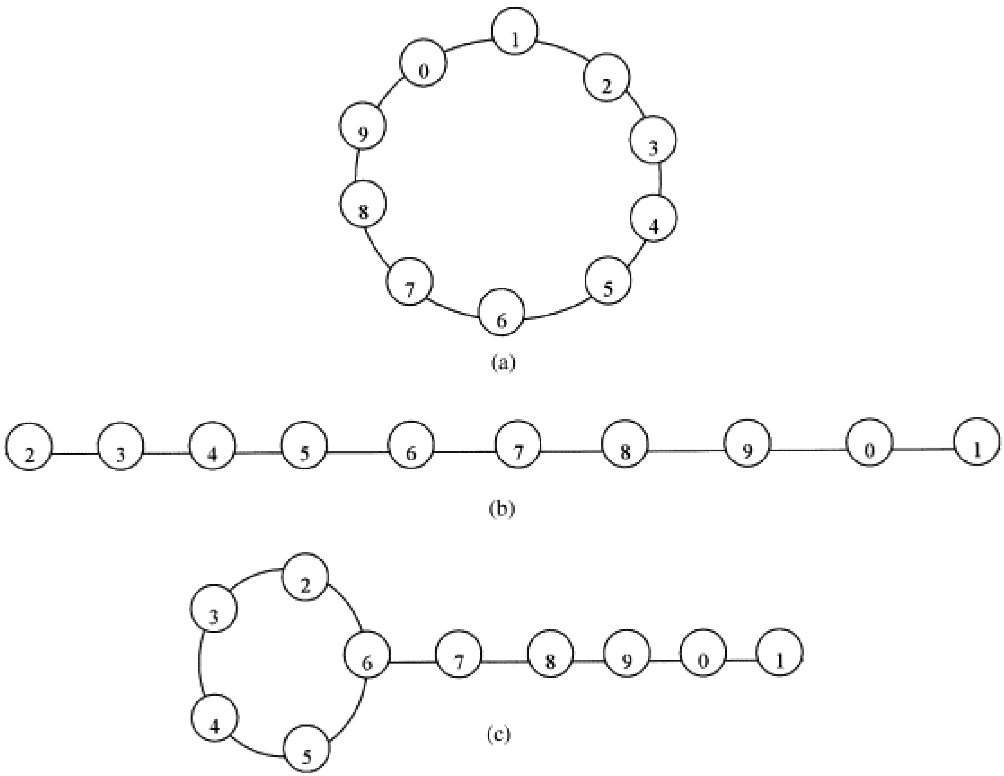
Overview:

- Ravindra K. Ahuja et al, 2002: "A survey of very large-scale neighborhood search techniques", at https://dx.doi.org/10.1016/S0166-218X(01)00338-9

Lin–Kernighan algorithm for TSP

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

- **Basic idea:** Apply a series of pairs of moves to an existing tour as shown in example to the right:
  (a) initial tour
  (b) a Hamiltonian path
  (c) a stem and cycle
- From (b) to (c) the edge (2, 6) is inserted
- From (c) to (b') the edge (5, 6) is removed (and so on)
- Node 1 (right end) stays fixed for the entire series of moves
- Edge insertion moves ((b) to (c)) are guided by a cumulative cost criterion
- Series of moves terminates if no further cost reduction is achievable



Source: https://dx.doi.org/10.1016/S0166-218X(01)00338-9

Meta-Heuristic: Partial Optimization Metaheuristic Under Special Intensication Conditions (POPMUSIC)

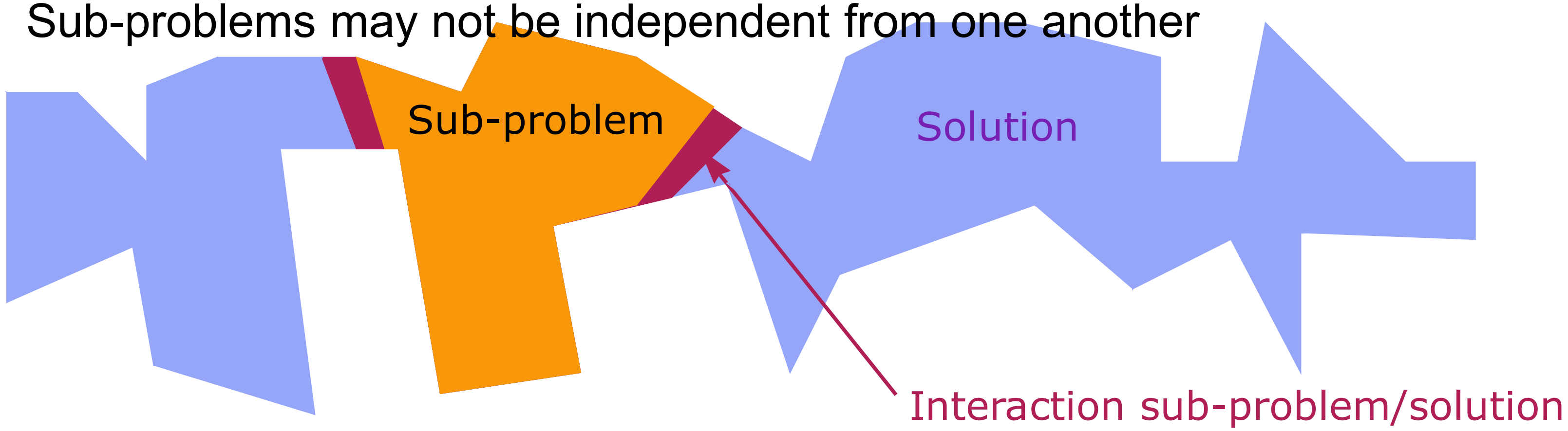
Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Idea

- Start from an initial solution
- Decompose solution into parts
- Optimize a sub-problem, i.e. several parts of the solution, which are close together (this presumes that there exist a proximity measure between parts and an optimization algorithm for smaller problems of the same kind)
- Repeat, until the optimized portions cover the entire solution

Difficulty

- Sub-problems may not be independent from one another



POPMUSIC Algorithm

Zürcher Hochschule für Angewandte Wissenschaften
zhaw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Solution $S = s_1 \cup s_2 \cup \ldots \cup s_p$ // p "disjoint" parts

$O = \varnothing$ // Set of "already optimized" seed parts

Parameter r // Number of "nearest" parts constituting a sub-problem

While $O \neq S$, **repeat**

1. Choose a seed part $s_i \notin O$
2. Create a sub-problem R composed of the r parts in S "nearest" of s_i
3. Optimize sub-problem R
4. **If** R has been improved **then**
   a) Update S by the parts of R
   b) Set $O = O \setminus R$ // parts in R are not yet completely optimized

   Else

   a') Set $O = O \cup s_i$ // s_i is already optimized

Choices in POPMUSIC

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

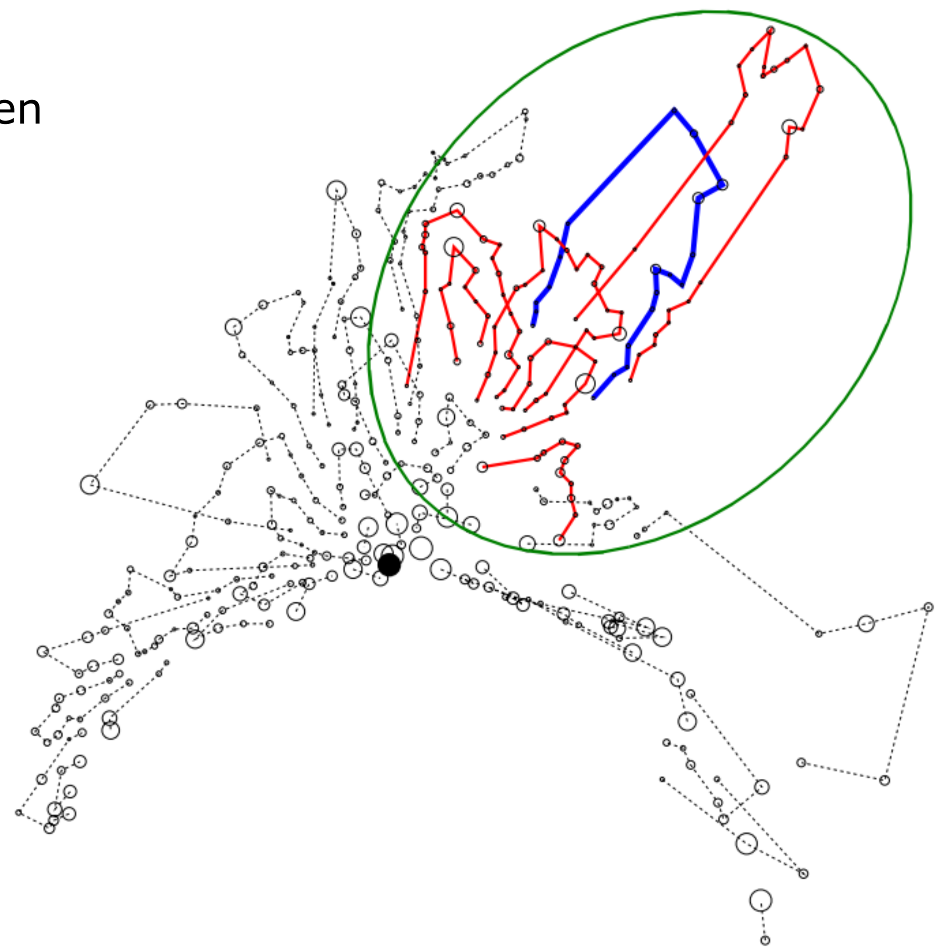
- Definition of a "part"
- Proximity measure between two parts
- Parameter *r,* the number of “nearest" parts to optimize as a sub-problem *R*
- Optimization procedure for sub-problems *R*

Variants:

- Line b): Slower: Set $O = \emptyset$ instead of $O = O \setminus R$ → *dump all already optimized parts*
- Line a’): Faster: Set $O = O \cup R$ instead of $O = O \cup s_i$ → *consider all parts in R as optimized*

POPMUSIC for CVRP

- **Part**: Tour of one vehicle
- **Distance between parts**: Distance between centres of gravity
- **Sub-Problem**: A subset of the parts, i.e. a smaller CVRP
- **Optimization process**: E.g. Tabu Search



Exercise

POPMUSIC for TSP:

How can a "part" and a "sub-problem" be defined for the TSP?

SOLUTION: POPMUSIC for TSP

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

POPMUSIC for TSP:

- Part: A single city *c* of the tour
- Distance: The difference of the indices of the two cities in the tour
- Sub-Problem: The *r* adjacent cities in the tour, immediately before/after the seed city *c*
- Solve TSP for the *r* cities (e.g. with Exhaustive Search), keeping the first and the last one fixed
- Re-Combination: Re-insert subtour in the existing tour

Quadratic Assignment Problem
QAP

Quatratic Assignment Problem QAP

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Given sets of *n facilities* (also called *activities*) and *n locations,* the *flows* between facilities and the *distances* between locations, the objective of the **Quadratic Assignment Problem** is to *assign each facility to a location* in such a way as to *minimize the total transportation cost.*

Applications:

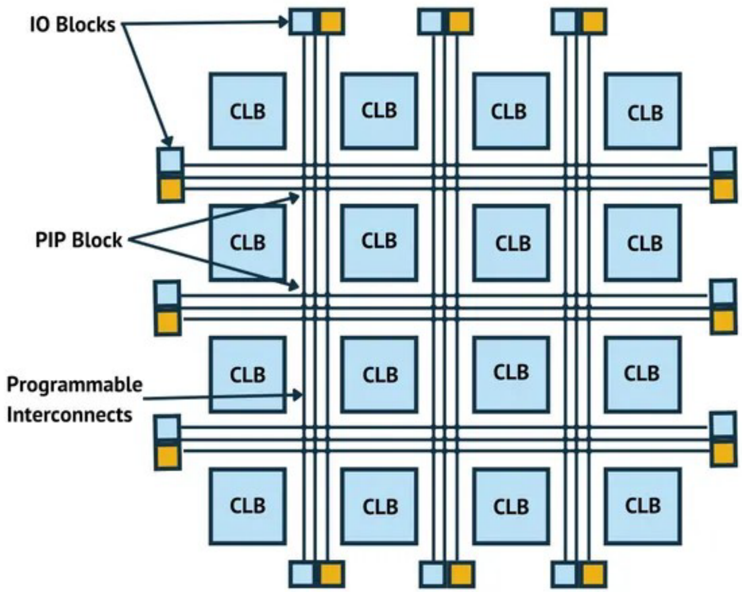
- Number of connections between electronic modules + Possible positions for the placement of the modules in a rack
- Number of connections between logic blocks + Possible positions for the placement of the blocks on a programmable chip (FPGA, see next slide)
- Number of passengers that must change flights + Assignment of flights to the gates at an airport
- Frequency of consecutive appearance of two letters in a language + Time between typing of two successive keys on a keyboard
- Meeting frequency of two employees + Time for walking from one desk to the other

Data Resources: https://www.opt.math.tugraz.at/qaplib/

Quatratic Assignment Problem QAP

FPGA as an application of QAP:

- The digital circuit to be realized with an FPGA is decomposed into parts (activities), which each fit into a configurable logic block (CLB) of the FPGA (locations).
- The flows between any two activities in this context are the number of interconnects between the corresponding parts used to synthesize the desired digital circuit.
- Since there are only limited interconnect resources on the FPGA, it is crucial to place activities with high interconnect requirements in locations close to each other.



Formal Definition of QAP

Identifying a permutation matrix $\mathbf{X}$ of dimension $n \times n$ (whose elements X_{ij} are 1 if activity j is assigned to location i and 0 in the other cases) such that:

$$\min z = \sum_{i,j=1}^{n} \sum_{h,k=1}^{n} d_{ih} f_{jk} X_{ij} X_{hk}$$

Subject to

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad \text{for } j = 1, 2, 3, \ldots, n$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad \text{for } i = 1, 2, 3, \ldots, n$$

$$X_{ij} \in (0, 1) \quad \text{for } i, j = 1, 2, 3, \ldots, n \qquad (1)$$

Where $d_{ih} \in D$ distance matrix and $f_{jk} \in F$ flow matrix.

Source: http://image.slidesharecdn.com/solutiontoquadraticassignmentproblem2-160331120135/95/solving-quadratic-assignment-problems-qap-using-ant-colony-system-4-638.jpg?cb=1459425710

Exercise

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

What is a potential neighborhood for the QAP?

What size does the neighborhood have?

SOLUTION: Neighborhoods for QAP

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Neighborhoods for QAP:

- Swap locations of 2 facilities ("2-exchange neighborhood")
  Size: $O(n^2)$

- Exchange locations of k = 3, 4, … facilities
  Size: $O\left(\binom{n}{k} \cdot k!\right) = O\left(n^k\right)$ (for $k \ll n$)

Tabu Search

Basic Idea of Tabu Search

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Basic Algorithm: E.g. Local Search with Best Improving Move strategy

Memory: Solutions visited and/or moves performed

Use Memory to…

- Forbid to come back to a solution already visited
- Forbid to perform the reverse of a move
- Penalize moves frequently used
- Force moves never used otherwise

"Do not revert a move too soon!"

Wipe Memory: Remove forbidden (tabu) status after a given number of iterations

Improvements: Even a forbidden move can be accepted if it improves best solution so far

Resources: Glover & Laguna, "*Tabu Search*" *or* https://www.uv.es/rmarti/paper/docs/ts1.pdf

Meta-Heuristic: Tabu Search

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

```
Data             Initial solution s
                 Utility function to minimize f
                 Set M of moves that can be applied to any feasible solution
                 Parameters t, max_iter : Tabu duration, number of iterations

Initializations
                 T = ∅                          // Set of forbidden moves (at most t)
                 s* := s                               // Best solution found

For max_iter iterations repeat
      Value_best_neighbour = ∞
      For all m ∈ M, m ∉ T repeat
         If Value_best_neighbour > f(s ⊕ m) then
            Value_best_neighbour = f(s ⊕ m)
            Best_move = m

      s = s ⊕ Best_move

      Replace the oldest move in T by Best_move⁻¹ (or add Best_move⁻¹, if |T| < t)

      If f(s*) > f(s) then
         s* = s

Return s*
```

Find best move **m**

Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering
IAMP Institut für Angewandte
Mathematik und Physik

Tabu Search for the Knapsack Problem:

Propose a setting/strategy for a Tabu Search for the Knapsack Problem

SOLUTION: Tabu Search for Knapsack

Example of a Tabu Search for Knapsack Problem

- Represent a solution as a bitvector of length n
- Each bit indicates whether the corresponding item is taken or not
- Neighborhood: Flip one item from 0 to 1 or from 1 to 0
- Tabu: Don't flip the same item for the next k steps again

Tuning Tabu Search

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

If Tabu duration is too low:

- A limited set of solutions is cyclically visited
- Algorithm gets stuck in the neighbourhood of a local minimum

If Tabu duration is too high:

- Too many solutions forbidden, especially those that can be interesting
- The search remains uphill instead of going down into valleys

Solution Strategies

- Reactive search:
  - Learn dynamically a good tabu value
  - Increase tabu value if the same solution is visited twice
  - Decrease tabu value if never returned to the same solution for many iterations

- Random tabu duration:
  - After each move, choose tabu duration randomly, e.g. $t = \lfloor |M| \cdot U(0,1)^4 + 1 \rfloor$ where U(0, 1) is uniformly distributed between 0 and 1, M is set of possible moves.
  - Effect: Few moves are tabu for a long time, forbidding cycles; most moves are forbidden for very few iterations

Santa's Challenge