Outline

1. The Quadratic Assignment Problem

2. Fundamental Search Space Properties
   - Distances
   - Fitness-Distance Correlation
   - Ruggedness
   - Plateaux
   - Barriers and Basins

3. Bin Packing

Quadratic Assignment Problem

- **Given**: \( n \) locations with a matrix \( D = [d_{ij}] \in \mathbb{R}^{n \times n} \) of distances and \( n \) units with a matrix \( F = [f_{kl}] \in \mathbb{R}^{n \times n} \) of flows between them

- **Task**: Find the assignment \( \sigma \) of units to locations that minimize the sum of product between flows and distances, ie,

\[
\min_{\sigma \in \Sigma} \sum_{i,j} f_{ij} d_{\sigma(i)\sigma(j)}
\]

Applications: hospital layout; keyboard layout
Example: QAP

\[
D = \begin{pmatrix}
0 & 4 & 3 & 2 & 1 \\
4 & 0 & 3 & 2 & 1 \\
3 & 3 & 0 & 2 & 1 \\
2 & 2 & 2 & 0 & 1 \\
1 & 1 & 1 & 1 & 0
\end{pmatrix}, \quad F = \begin{pmatrix}
0 & 1 & 2 & 3 & 4 \\
1 & 0 & 2 & 3 & 4 \\
2 & 2 & 0 & 3 & 4 \\
3 & 3 & 3 & 0 & 4 \\
4 & 4 & 4 & 4 & 0
\end{pmatrix}
\]

The optimal solution is \( \sigma = (1, 2, 3, 4, 5) \), that is, facility 1 is assigned to location 1, facility 2 is assigned to location 2, etc.

The value of \( f(\sigma) \) is 100.

Delta evaluation

Evaluation of 2-exchange \( \{r, s\} \) can be done in \( O(n) \)

\[
\Delta(\psi, r, s) = b_{rr} \cdot (a_{\psi_r\psi_s} - a_{\psi_s\psi_r}) + b_{rs} \cdot (a_{\psi_r\psi_r} - a_{\psi_r\psi_s}) + b_{ss} \cdot (a_{\psi_s\psi_r} - a_{\psi_s\psi_s}) + \\
\sum_{k=1, k \neq r, s}^{n} (b_{kr} \cdot (a_{\psi_k\psi_s} - a_{\psi_s\psi_r}) + b_{ks} \cdot (a_{\psi_k\psi_r} - a_{\psi_k\psi_s}) + b_{kk} \cdot (a_{\psi_k\psi_k} - a_{\psi_k\psi_k}))
\]

Concepts that Define the Search Landscape

- Search space \( S \)
- Neighborhood structure \( N \subseteq S \times S \)
- Evaluation Function \( g(\pi) : S \mapsto \mathbb{R} \)

Definition:
The Search Landscape \( L(\pi) \) of \( \pi \) is the triple

\[
L(\pi) := \langle S(\pi), N(\pi), g(\pi) \rangle
\]
Distances

Set of paths in $G_N$ with $s, s' \in S$:

$$\Phi(s, s') = \{ (s_1, \ldots, s_h) | s_1 = s, s_h = s' \forall i : 1 \leq i \leq h - 1, (s_i, s_{i+1}) \in E_N \}$$

If $\phi = (s_1, \ldots, s_h) \in \Phi(s, s')$ let $|\phi| = h$ be the length of the path; then the distance between any two solutions $s, s'$ is the length of shortest path between $s$ and $s'$ in $G_N$:

$$d_N(s, s') = \min_{\phi \in \Phi(s, s')} |\phi|$$

Note: with permutations it is easy to see that:

$$d_N(\pi, \pi') = d_N(\pi^{-1} \cdot \pi', i)$$

Distances for Linear Permutation Representations

Swap neighborhood operator

Computable in $O(n^2)$ by the precedence distance metric:

$$\#\{(i, j) | 1 \leq i < j \leq n, pos_{\pi'}(\pi_j) < pos_{\pi'}(\pi_i)\}$$

$$diam(G_N) = n(n - 1)/2$$

Interchange neighborhood operator

Computable in $O(n) + O(n)$ since

$$d_N(\pi, \pi') = d_N(\pi^{-1} \cdot \pi', i) = n - c(\pi^{-1} \cdot \pi')$$

where $c(\pi)$ is the number of disjoint cycles that decompose a permutation.

$$diam(G_{N^X}) = n - 1$$

Insert neighborhood operator

Computable in $O(n) + O(n \log(n))$ since

$$d_N(\pi, \pi') = d_N(\pi^{-1} \cdot \pi', i) = n - |lis(\pi^{-1} \cdot \pi')|$$

where $lis(\pi)$ denotes the length of the longest increasing subsequence.

$$diam(G_{N_I}) = n - 1$$

Fundamental Search Space Properties

The behavior and performance of an SLS algorithm on a given problem instance crucially depends on properties of the respective search space.

Simple properties of search space $S$:

- search space size $|S|$
- search space diameter $diam(G_N)$
  ($= $ maximal distance between any two candidate solutions)

Note: it depends on the neighborhood size $|N|$

- number of (optimal) solutions $|S'|$, solution density $|S'|/|S|$
- distribution of solutions within the neighborhood graph

Solution densities and distributions can generally be determined by:

- exhaustive enumeration;
- sampling methods;
- counting algorithms (often variants of complete algorithms).
Distances for Circular Permutation Representations

Reversal neighborhood operator

sorting by reversal is known to be NP-hard

Block moves neighborhood operator

unknown whether it is NP-hard but there does not exist a proved polynomial-time algorithm

Distances for Assignment Representations

▶ Hamming Distance

▶ An assignment can be seen as a partition of \( n \) in \( k \) mutually exclusive non-empty subsets

One-exchange neighborhood operator

The partition-distance \( d_{1E}(P, P') \) between two partitions \( P \) and \( P' \) is the minimum number of elements that must be moved between subsets in \( P \) so that the resulting partition equals \( P' \).

The partition-distance can be computed in polynomial time by solving an assignment problem. Given the assignment matrix \( M \) where in each cell \((i, j)\) it is \( |S_i \cap S_j'| \) with \( S_i \in P \) and \( S_j' \in P' \) and defined \( A(P, P') \) the assignment of maximal sum then it is \( d_{1E}(P, P') = n - A(P, P') \)

Example: Search space size and diameter for the TSP

▶ Given: Symmetric TSP instance with \( n \) vertices
▶ Candidate solutions = permutations of vertices
▶ Search space size = \((n-1)!/2\)
▶ Size of 2-exchange neighborhood
  \( = \binom{n}{2} = n \cdot (n - 1)/2 \)
▶ Size of 3-exchange neighborhood
  \( = \binom{n}{3} = n \cdot (n - 1) \cdot (n - 2)/6 \)
▶ Diameter of neighborhood graphs: Exact values unknown.
  ▶ Bounds for 2-exchange neighborhood = \([n/2, n - 1]\)
  ▶ Bounds for 3-exchange neighborhood = \([n/3, n - 1]\)

Example: Correlation between solution density and search cost for GWSAT over set of hard Random-3-SAT instances:
Phase Transition for 3-SAT

Random instances ⇒ $m$ clauses of $n$ uniformly chosen variables

Example: Complete distribution of position types for hard Random-3-SAT instances

<table>
<thead>
<tr>
<th>instance</th>
<th>avg sc</th>
<th>SLMIN</th>
<th>LMIN</th>
<th>IPLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>uf20-91/easy</td>
<td>13.05</td>
<td>0%</td>
<td>0.11%</td>
<td>0%</td>
</tr>
<tr>
<td>uf20-91/medium</td>
<td>83.25</td>
<td>&lt;0.01%</td>
<td>0.13%</td>
<td>0%</td>
</tr>
<tr>
<td>uf20-91/hard</td>
<td>563.94</td>
<td>&lt;0.01%</td>
<td>0.16%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Example: Sampled distribution of position types for hard Random-3-SAT instances

<table>
<thead>
<tr>
<th>instance</th>
<th>avg sc</th>
<th>SLMIN</th>
<th>LMIN</th>
<th>IPLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>uf50-218/medium</td>
<td>615.25</td>
<td>0%</td>
<td>47.29%</td>
<td>0%</td>
</tr>
<tr>
<td>uf100-430/medium</td>
<td>3410.45</td>
<td>0%</td>
<td>43.89%</td>
<td>0%</td>
</tr>
<tr>
<td>uf150-645/medium</td>
<td>10231.89</td>
<td>0%</td>
<td>41.95%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Classification of search positions

<table>
<thead>
<tr>
<th>position type</th>
<th>&gt;</th>
<th>=</th>
<th>&lt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLMIN (strict local min)</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LMIN (local min)</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>IPLAT (interior plateau)</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>SLOPE</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>LEDGE</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>LMAX (local max)</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>SLMAX (strict local max)</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

"+" = present, "−" absent; table entries refer to neighbors with larger (">"), equal ("="), and smaller ("<") evaluation function values

Example:

Sampled distribution of position types for hard Random-3-SAT instances

<table>
<thead>
<tr>
<th>instance</th>
<th>avg sc</th>
<th>SLOPE</th>
<th>LEDGE</th>
<th>LMAX</th>
<th>SLMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>uf20-91/easy</td>
<td>0.59%</td>
<td>99.27%</td>
<td>0.04%</td>
<td>&lt;0.01%</td>
<td></td>
</tr>
<tr>
<td>uf20-91/medium</td>
<td>0.31%</td>
<td>99.40%</td>
<td>0.06%</td>
<td>&lt;0.01%</td>
<td></td>
</tr>
<tr>
<td>uf20-91/hard</td>
<td>0.56%</td>
<td>99.23%</td>
<td>0.05%</td>
<td>&lt;0.01%</td>
<td></td>
</tr>
</tbody>
</table>

(based on exhaustive enumeration of search space; sc refers to search cost for GWSAT)

Example:

Sampled distribution of position types for hard Random-3-SAT instances

<table>
<thead>
<tr>
<th>instance</th>
<th>avg sc</th>
<th>SLOPE</th>
<th>LEDGE</th>
<th>LMAX</th>
<th>SLMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>uf50-218/medium</td>
<td>&lt;0.01%</td>
<td>52.71%</td>
<td>0%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>uf100-430/medium</td>
<td>0%</td>
<td>56.11%</td>
<td>0%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>uf150-645/medium</td>
<td>0%</td>
<td>58.05%</td>
<td>0%</td>
<td>0%</td>
<td></td>
</tr>
</tbody>
</table>

(based on sampling along GWSAT trajectories; sc refers to search cost for GWSAT)
Local Minima

**Note:** Local minima impede local search progress.

**Simple properties of local minima:**
- *number of local minima:* $|l_{\text{min}}|$, *local minima density* $|l_{\text{min}}|/|S|$
- *localization of local minima:* distribution of local minima within the neighborhood graph

**Problem:** Determining these measures typically requires exhaustive enumeration of search space.

⇒ Approximation based on sampling or estimation from other measures (such as autocorrelation measures, see below).

---

**Example:** Distribution of local minima for the TSP

**Goal:** Empirical analysis of distribution of local minima for Euclidean TSP instances.

**Experimental approach:**
- Sample sets of local optima of three TSPLIB instances using multiple independent runs of two TSP algorithms (3-opt, ILS).
- Measure pairwise distances between local minima (using bond distance $= \text{number of edges in which two given tours differ}$).
- Sample set of purportedly globally optimal tours using multiple independent runs of high-performance TSP algorithm.
- Measure minimal pairwise distances between local minima and respective closest optimal tour (using bond distance).

---

**Empirical results:**

<table>
<thead>
<tr>
<th>Instance</th>
<th>avg sq [%]</th>
<th>$d_{l_{\text{min}}}$</th>
<th>$d_{\text{opt}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>rat783</td>
<td>3.45</td>
<td>197.8</td>
<td>185.9</td>
</tr>
<tr>
<td>pr1002</td>
<td>3.58</td>
<td>242.0</td>
<td>208.6</td>
</tr>
<tr>
<td>pcb1173</td>
<td>4.81</td>
<td>274.6</td>
<td>246.0</td>
</tr>
</tbody>
</table>

(based on local minima collected from 1 000/200 runs of 3-opt/ILS)

avg sq [%]: average solution quality expressed in percentage deviation from optimal solution

**Interpretation:**
- Average distance between local minima is small compared to maximal possible bond distance, $n$.
  ⇒ *Local minima are concentrated in a relatively small region of the search space.*
- Average distance between local minima is slightly larger than distance to closest global optimum.
  ⇒ *Optimal solutions are located centrally in region of high local minima density.*
- Higher-quality local minima found by ILS tend to be closer to each other and the closest global optima compared to those determined by 3-opt.
  ⇒ *Higher-quality local minima tend to be concentrated in smaller regions of the search space.*

Note: These results are fairly typical for many types of TSP instances and instances of other combinatorial problems. In many cases, local optima tend to be clustered; this is reflected in multi-modal distributions of pairwise distances between local minima.
Fitness-Distance Correlation (FDC)

Idea: Analyze correlation between solution quality (fitness) \( g \) of candidate solutions and distance \( d \) to (closest) optimal solution.

Measure for FDC: empirical correlation coefficient \( r_{fdc} \).

Fitness-distance plots, i.e., scatter plots of the \((g_i, d_i)\) pairs underlying an estimate of \( r_{fdc} \), are often useful to graphically illustrate fitness distance correlations.

- The FDC coefficient, \( r_{fdc} \), depends on the given neighborhood relation.
- \( r_{fdc} \) is calculated based on a sample of \( m \) candidate solutions (typically: set of local optima found over multiple runs of an iterative improvement algorithm).

High FDC \( (r_{fdc} \) close to one):

- ‘Big valley’ structure of landscape provides guidance for local search;
- search initialization: high-quality candidate solutions provide good starting points;
- search diversification: (weak) perturbation is better than restart;
- typical, e.g., for TSP.

Low FDC \( (r_{fdc} \) close to zero):

- global structure of landscape does not provide guidance for local search;
- typical for very hard combinatorial problems, such as certain types of QAP (Quadratic Assignment Problem) instances.

Example: FDC plot for TSPLIB instance rat783, based on 2500 local optima obtained from a 3-opt algorithm

Applications of fitness-distance analysis:

- algorithm design: use of strong intensification (including initialization) and relatively weak diversification mechanisms;
- comparison of effectiveness of neighborhood relations;
- analysis of problem and problem instance difficulty.

Limitations and short-comings:

- \textit{a posteriori} method, requires set of (optimal) solutions, \textbf{but}: results often generalize to larger instance classes;
- optimal solutions are often not known, using best known solutions can lead to erroneous results;
- can give misleading results when used as the sole basis for assessing problem or instance difficulty.
Ruggedness

Idea: Rugged search landscapes, i.e., landscapes with high variability in evaluation function value between neighboring search positions, are hard to search.

Example: Smooth vs rugged search landscape

Note: Landscape ruggedness is closely related to local minima density: rugged landscapes tend to have many local minima.

The ruggedness of a landscape $L$ can be measured by means of the empirical autocorrelation function $r(i)$:

$$r(i) := \frac{1}{m} \cdot \frac{\sum_{k=1}^{m} (g_k - \bar{g}) \cdot (g_{k+i} - \bar{g})}{\sum_{k=1}^{m} (g_k - \bar{g})^2}$$

where $g_1, \ldots, g_m$ are evaluation function values sampled along an uninformed random walk in $L$.

Note: $r(i)$ depends on the given neighborhood relation.

High AC (close to one):

- “smooth” landscape;
- evaluation function values for neighboring candidate solutions are close on average;
- low local minima density;
- problem typically relatively easy for local search.

Low AC (close to zero):

- very rugged landscape;
- evaluation function values for neighboring candidate solutions are almost uncorrelated;
- high local minima density;
- problem typically relatively hard for local search.

Note:

- Measures of ruggedness, such as AC, are often insufficient for distinguishing between the hardness of individual problem instances;
- but they can be useful for
  - analyzing differences between neighborhood relations for a given problem,
  - studying the impact of parameter settings of a given SLS algorithm on its behavior,
  - classifying the difficulty of combinatorial problems.
Plateaux

Plateaux, i.e., ‘flat’ regions in the search landscape

**Intuition:** Plateaux can impede search progress due to lack of guidance by the evaluation function.

Definitions

- **Region:** connected set of search positions.
- **Border of region** $R$: set of search positions with at least one direct neighbor outside of $R$ (border positions).
- **Plateau region:** region in which all positions have the same level, i.e., evaluation function value, $l$.
- **Plateau:** maximally extended plateau region, i.e., plateau region in which no border position has any direct neighbors at the plateau level $l$.
- **Solution plateau:** Plateau that consists entirely of solutions of the given problem instance.
- **Exit of plateau region** $R$: direct neighbor $s$ of a border position of $R$ with lower level than plateau level $l$.
- **Open / closed plateau:** plateau with / without exits.

Measures of plateau structure:

- **plateau diameter** = diameter of corresponding subgraph of $G_N$
- **plateau width** = maximal distance of any plateau position to the respective closest border position
- **number of exits, exit density**
- **distribution of exits within a plateau, exit distance distribution** (in particular: avg./max. distance to closest exit)

Some plateau structure results for SAT:

- Plateaux typically don’t have an interior, i.e., almost every position is on the border.
- The diameter of plateaux, particularly at higher levels, is comparable to the diameter of search space. (In particular: plateaux tend to span large parts of the search space, but are quite well connected internally.)
- For open plateaux, exits tend to be clustered, but the average exit distance is typically relatively small.
Observation:

The difficulty of escaping from closed plateaux or strict local minima is related to the height of the barrier, i.e., the difference in evaluation function, that needs to be overcome in order to reach better search positions:

Higher barriers are typically more difficult to overcome (this holds, e.g., for Probabilistic Iterative Improvement or Simulated Annealing).

Definitions:

- Positions $s, s'$ are mutually accessible at level $l$ iff there is a path connecting $s'$ and $s$ in the neighborhood graph that visits only positions $t$ with $g(t) \leq l$.

- The barrier level between positions $s, s'$, $bl(s, s')$ is the lowest level $l$ at which $s'$ and $s'$ are mutually accessible; the difference between the level of $s$ and $bl(s, s')$ is called the barrier height between $s$ and $s'$.

- Basins, i.e., maximal (connected) regions of search positions below a given level, form an important basis for characterizing search space structure.

Example: Basins in a simple search landscape and corresponding basin tree

Note: The basin tree only represents basins just below the critical levels at which neighboring basins are joined (by a saddle).

Outline

1. The Quadratic Assignment Problem
2. Fundamental Search Space Properties
   - Distances
   - Fitness-Distance Correlation
   - Ruggedness
   - Plateaux
   - Barriers and Basins
3. Bin Packing
Two-Dimensional Packing Problems

Two dimensional bin packing
Given: A set \( L = (a_1, a_2, \ldots, a_n) \) of \( n \) rectangular items, each with a width \( w_j \) and a height \( h_j \) and an unlimited number of identical rectangular bins of width \( W \) and height \( H \).
Task: Allocate all the items into a minimum number of bins, such that the original orientation is respected (no rotation of the items is allowed).

Two dimensional strip packing
Given: A set \( L = (a_1, a_2, \ldots, a_n) \) of \( n \) rectangular items, each with a width \( w_j \) and a height \( h_j \) and a bin of width \( W \) and infinite height (a strip).
Task: Allocate all the items into the strip by minimizing the used height and such that the original orientation is respected (no rotation of the items is allowed).

Two dimensional cutting stock
Each item has a profit \( p_j > 0 \) and the task is to select a subset of items to be packed in a single finite bin that maximize the total selected profit.