Permutation distance measures for memetic algorithms with population management

Marc Sevaux$^1$   Kenneth Sörensen$^2$

$^1$University of Valenciennes, CNRS, UMR 8530, LAMIH-SP
Le Mont Houy - Bat Jonas 2, F–59313 Valenciennes cedex 9, France
marc.sevaux@univ-valenciennes.fr

$^2$University of Antwerp, Faculty of Applied Economics
Prinsstraat 13, B–2000 Antwerp, Belgium
kenneth.sorensen@ua.ac.be

MIC’2005
Sixth Metaheuristics International Conference
Vienna 22–26 August 2005
MA|PM: memetic algorithms with population management
MA|PM: memetic algorithms with population management

Permutation distance measures for MA|PM

Marc Sevaux,
Kenneth Sørensen

MA|PM

$1 | r_j | \sum w_j T_j$

This talk

Permutation distance measures

Exact match
Deviation
Squared deviation
A
R
Edit
LCS
Reversal
Kendall's $\tau$

Comparison
Results
Conclusion
MA|PM: memetic algorithms
with population management
MA|PM: memetic algorithms with population management

This talk

Permutation distance measures

Exact match
Deviation
Squared deviation
A
R
Edit
LCS
Reversal
Kendall’s τ

Comparison
Results
Conclusion
MA|PM: memetic algorithms with population management
MA|PM: memetic algorithms with population management
MA|PM: memetic algorithms with population management

This talk

Permutation distance measures

Exact match
Deviation
Squared deviation
A
R
Edit
LCS
Reversal
Kendall’s τ

Comparison
Results
Conclusion
MA|PM: memetic algorithms with population management

Permutation distance measures for MA|PM

Marc Sevaux, Kenneth Sørensen

MA|PM

1|\sum w_j T_j

This talk

Permutation distance measures
Exact match
Deviation
Squared deviation
A
R
Edit
LCS
Reversal
Kendall’s ρ

Comparison
Results
Conclusion
MA|PM: memetic algorithms with population management

Permutation distance measures for MA|PM

Marc Sevaux, Kenneth Sörensen

MA|PM

$\sum_{j} w_{j} T_{j}$

This talk

Permutation distance measures

Exact match
Deviation
Squared deviation
A
R
Edit
LCS
Reversal
Kendall's $\tau$

Comparison
Results
Conclusion
The total weighted tardiness single machine scheduling problem

- $n$ jobs
- One machine
- No preemption
- Release date $r_j$
- Processing times $p_j$
- Due dates $d_j$
- Weights $w_j$
- Completion time $C_j$
- Tardiness $T_j = \max(0, C_j - d_j)$
- **Objective**: minimize total weighted tardiness $(1|r_j| \sum w_j T_j)$
- $\mathcal{NP}$-Hard in a strong sense
This talk

- Simple MA|PM for total weighted tardiness single machine scheduling problem (1|r_j| \sum w_j T_j)
  - Linear ordered crossover
  - Pairwise adjacent interchange local search
  - General pairwise interchange mutation
  - “Sawtooth” population management parameter

- Develop a normalized version of each distance measure (d \in [0, 1])

- Examine each of the distance measures and compare performance
Permutation distance measures

- Exact match
- Deviation
- Squared deviation
- $A$ (absolute)
- $R$ (relative)
- Edit
- Longest Common Subsequence
- Reversal
- Kendall’s Tau

- Remark: only permutations of $[1 \cdots n]$
The Exact Match Distance


“Hamming distance for permutations”

\[ d_{em}(S, T) = \sum_{i=1}^{n} x_i \text{ where } x_i = \begin{cases} 0 & \text{if } S(i) = T(i) \\ 1 & \text{otherwise} \end{cases} \]

Normalized:

\[ \hat{d}_{em}(S, T) = \frac{1}{n} d_{em} \]

Example:

\[
\begin{align*}
S & \quad [1,2,3,4,5] \quad d_{em} = 5 \\
T & \quad [2,4,5,1,3] \quad \hat{d}_{em} = 1
\end{align*}
\]
The Deviation Distance


- “Spearman’s footrule”
- Total deviation of all items
  \[ d_{\text{dev}}(S, T) = \sum_{k=1}^{n} |i - j| \text{ where } T(j) = S(i) = k \]

- Normalized:
  \[ \hat{d}_{\text{dev}}(S, T) = \begin{cases} 
    2/n^2 \times d_{\text{dev}}(S, T) & \text{if } n \text{ is even} \\
    2/(n^2 - 1) \times d_{\text{dev}}(S, T) & \text{if } n \text{ is odd} 
\end{cases} \]

- Example:
  \[ S = [1,2,3,4,5] \quad d_{\text{dev}} = 10 \]
  \[ T = [2,4,5,1,3] \quad \hat{d}_{\text{dev}} = 10 \times 2/24 = .833 \]
The Squared Deviation Distance

- “Spearman’s rank correlation coefficient”
- Sum of squares of deviations

\[ d_{sdev}(S, T) = \sum_{k=1}^{n} (i - j)^2 \quad \text{where} \quad T(j) = S(i) = k \]

- Normalized:

\[ \hat{d}_{sdev}(S, T) = \frac{3}{n^3 - n} \, d_{sdev}(S, T) \]

- Example:

\[
\begin{align*}
S & \quad [1,2,3,4,5] \quad d_{dev} = 22 \\
T & \quad [2,4,5,1,3] \quad \hat{d}_{dev} = 22 \times 3/120 = .55
\end{align*}
\]
The A Distance


- Absolute position of the items is important
- Differences between indices of items

\[ d_A(S, T) = \sum_{i=1}^{n} |S(i) - T(i)|. \]

- Normalized:

\[ \hat{d}_A(S, T) = \begin{cases} 
2/n^2 \times d_A(S, T) & \text{if } n \text{ is even} \\
2/(n^2 - 1) \times d_A(S, T) & \text{if } n \text{ is odd}
\end{cases} \]

- Example:

<table>
<thead>
<tr>
<th>S</th>
<th>[1,2,3,4,5]</th>
<th>( d_A = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>[2,4,5,1,3]</td>
<td>( \hat{d}_A = 10 \times 2/24 = .833 )</td>
</tr>
</tbody>
</table>
The R Distance


- Relative position of the items is important
- Number of times \( S(i + 1) \) does not follow \( S(i) \) in \( T \)
- Normalized:

\[
\hat{d}_R(S, T) = \frac{1}{n - 1} d_R(S, T)
\]

- Example:
  \[
  S = [1,2,3,4,5] \quad d_A = 3 \\
  T = [2,4,5,1,3] \quad \hat{d}_A = 3/4 = .75
  \]
The Edit Distance


- Three edit operations (add, remove, substitute)
- Minimum number of edit operations required to transform $S$ into $T$
- Or: minimum cost of edit transformation of $S$ into $T$
- Normalized: divide by $n$

Example:

$S = [1,2,3,4,5]$ \( d_{\text{edit}} = 4 \) (Remove and add 1 and 3)

$T = [2,4,5,1,3]$ \( \hat{d}_A = 4/5 = .8 \)
The Longest Common Subsequence Distance

- Subsequence: sequence of characters that appears in the same order in the string
  Example: [3, 5, 7] is a subsequence of [1, 2, 3, 4, 5, 6, 7]
- LCS: longest subsequence that appears in both strings
- LCS distance: \( n - \text{LCS} \)
- Normalize:
  \[
  \hat{d}_{\text{LCS}} = \frac{d_{\text{LCS}}}{n - 1}
  \]
- Example:
  \( S = [1, 2, 3, 4, 5] \)
  \( d_{\text{LCS}} = 2 \) (LCS = [2, 4, 5])
  \( T = [2, 4, 5, 1, 3] \)
  \( \hat{d}_{\text{LCS}} = \frac{2}{4} = .5 \)
- Alternative: longest common substring
The Reversal Distance


- Minimum number of substring reversals required to transform $S$ into $T$
- Reversal 123456789 $\rightarrow$ 123876549
- Important in molecular biology
- But: $\mathcal{NP}$-hard:
- Too bad: 2–opt
Kendall’s Tau


- Number of pairwise adjacent permutations required to transform $S$ into $T$

$$d_\tau(S, T) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} z_{ij}$$

where $z_{ij} = \begin{cases} 1 & \text{if } S(i) < S(j) \text{ and } T(i) > T(j) \\ 0 & \text{otherwise} \end{cases}$

- Normalized:

$$\hat{d}_\tau(S, T) = 2 \frac{d_\tau(S, T)}{n^2 - n}$$

- Example:

$S = [1,2,3,4,5]$ \hspace{1 cm} $d_\tau = 6$

$T = [2,4,5,1,3]$ \hspace{1 cm} $\hat{d}_\tau = 6 \times 2/20 = .6$
Comparison of distance measures

- 20 instances, 100 jobs
- 5 minutes per run

<table>
<thead>
<tr>
<th></th>
<th>Complexity</th>
<th>Avg. it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>$O(n)$</td>
<td>6719.3</td>
</tr>
<tr>
<td>A</td>
<td>$O(n)$</td>
<td>3924.4</td>
</tr>
<tr>
<td>R</td>
<td>$O(n)$</td>
<td>2617.9</td>
</tr>
<tr>
<td>Deviation</td>
<td>$O(n^2)$</td>
<td>437.9</td>
</tr>
<tr>
<td>Squared deviation</td>
<td>$O(n^2)$</td>
<td>232.5</td>
</tr>
<tr>
<td>Edit</td>
<td>$O(n^2)$</td>
<td>362.7</td>
</tr>
<tr>
<td>Longest common subsequence</td>
<td>$O(n^2)$</td>
<td>418.8</td>
</tr>
<tr>
<td>Kendall’s tau</td>
<td>$O(n^2)$</td>
<td>329.3</td>
</tr>
</tbody>
</table>
Results

The figure shows the fraction over best-found for various data file distances. The x-axis represents the data file numbers from 1 to 20, and the y-axis represents the fraction over best-found. The different lines correspond to different distance metrics:

- **Edit distance** (diamonds)
- **A distance** (crosses)
- **R distance** (squares)
- **Exact match distance** (clear circles)
- **Deviation distance** (open circles)
- **Squared deviation distance** (crosses)
- **Longest common subsequence** (open circles)
- **Kendall’s tau** (black diamonds)

Each line visualizes the performance of its respective distance metric across the data files, indicating how well each metric performs in capturing the relationship between the best-found and the data files.
Conclusion

- Simple distance measures work better
- Tentative explanation
  - Local search is very fast but not very powerful
  - Simple distance measures allow for a lot more iterations
  - Better distance measures cannot compensate
- Perspectives
  - Better local search mechanisms
  - Different problems
  - Interplay between population management strategy and distance measure
  - . . .
Permutation distance measures for memetic algorithms with population management

Marc Sevaux\textsuperscript{1}  Kenneth Sørensen\textsuperscript{2}

\textsuperscript{1}University of Valenciennes, CNRS, UMR 8530, LAMIH-SP  
Le Mont Houy - Bat Jonas 2, F–59313 Valenciennes cedex 9, France  
marc.sevaux@univ-valenciennes.fr

\textsuperscript{2}University of Antwerp, Faculty of Applied Economics  
Prinsstraat 13, B–2000 Antwerp, Belgium  
kenneth.sorensen@ua.ac.be

MIC'2005  
Sixth Metaheuristics International Conference  
Vienna 22–26 August 2005