Local Search for String Problems: Brute Force is Essentially Optimal

Jiong Guo, Danny Hermelin, Christian Komusiewicz

Universität des Saarlandes, Ben-Gurion University, TU Berlin & Université de Nantes

CPM 2013
A local search problem $\Pi$ consists of

- a set $\mathcal{I}$ of **instances**,  
- for each instance $I \in \mathcal{I}$ a set $F(I)$ of **feasible solutions**,  
- an **objective function** $f : F(I) \to \mathbb{Z}$, and  
- for each solution $s \in F(I)$ a **neighborhood** $N(s, I) \subseteq F(I)$.

**Goal:** Find **locally optimal** solution $s \in F(I)$ such that for all $s' \in N(s, I)$, $f(s) \leq f(s')$.

**Generic local search algorithm:**

1. $s := \text{“some element of } F(I) \text{”}$  
2. **while** $s$ is not locally optimal :  
3. \quad find $s' \in N(s)$ with $f(s) < f(s')$  
4. \quad $s := s'$  
5. **return** $s$

**Examples:** $k$-Means, Simplex
Advantages of local search:
- fast in practice
- local optima often close to global optimum
- generic algorithm scheme $\leadsto$ easy to implement

Tuning local search algorithms:
- faster convergence to local optimum
- changing type/size of neighborhood $N$

Theoretical analysis:
- number of iterations $\leadsto$
  - PLS-completeness [Johnson, Papadimitriou & Yannakakis, JCSS 1988],
  - Smoothed Analysis [Spielman & Teng, JACM 2004]
- efficiently searching the neighborhood $N$ $\leadsto$
  - Parameterized Local Search
Example:

**LS-TSP**

**Input:** A set $V$ of $n$ cities, a distance matrix $d : V \times V \to \mathbb{Z}$, a tour $T$, and an integer $k$.

**Question:** Can we obtain a shorter tour $T'$ by replacing $\leq k$ arcs of $T$?

**Obvious:** Solvable in $n^{O(k)}$ time

**Aim:** Improve running time to $f(k) \cdot \text{poly}(n)$

**But:** **LS-TSP** is $W[1]$-hard parameterized by $k$ [Marx, ORL 2008]

$\implies$ probably no $f(k) \cdot \text{poly}(n)$-time algorithm
Can we avoid brute-force $n^{O(k)}$ algorithms for searching the neighborhood?

**General:** Parameterized local search problems usually $W[1]$-hard with respect to the $k$-change neighborhood

[Fellows et al., JCSS, 2012]

Positive results for...

... restricted inputs:

- **LS-Feedback Arc Set** in tournaments [Fomin et al., AAAI 2010]
- **LS-TSP** on planar graphs [Guo et al., Algorithmica, to appear]

... other neighborhood types:

- **LS-TSP** with $\leq k$ swaps in an $m$-bounded range  
  [Guo et al., Algorithmica, to appear]
- **LS-TSP** with maximum shift $k$ [Balas, AOR 1999]
Can we avoid brute-force $n^{O(k)}$ algorithms for searching the neighborhood?

**General:** Parameterized local search problems usually W[1]-hard with respect to the $k$-change neighborhood

[Fellows et al., JCSS, 2012]

Positive results for...

... restricted inputs:

- **LS-Feedback Arc Set** in tournaments [Fomin et al., AAAI 2010]
- **LS-TSP** on planar graphs [Guo et al., Algorithmica, to appear],

... other neighborhood types:

- **LS-TSP** with $\leq k$ swaps in an $m$-bounded range [Guo et al., Algorithmica, to appear]
- **LS-TSP** with maximum shift $k$ [Balas, AOR 1999]

How easy are string problems with respect to parameterized local search?
Can we avoid brute-force $n^{O(k)}$ algorithms for searching the neighborhood?

**General:** Parameterized local search problems usually W[1]-hard with respect to the $k$-change neighborhood

[Fellows et al., JCSS, 2012]

Positive results for...

... restricted inputs:

- **LS-Feedback Arc Set** in tournaments [Fomin et al., AAAI 2010]
- **LS-TSP** on planar graphs [Guo et al., Algorithmica, to appear],

... other neighborhood types:

- **LS-TSP** with $\leq k$ swaps in an $m$-bounded range [Guo et al., Algorithmica, to appear]
- **LS-TSP** with maximum shift $k$ [Balas, AOR 1999]

How easy are **string** problems with respect to parameterized local search for the **Hamming distance neighborhood**?
Closest String

**Input:** Strings $S_1, \ldots, S_m$ of length $n$ over alphabet $\Sigma$, an integer $d$.

**Question:** Is there a string $S$ of length $n$ with Hamming distance $\leq d$ to each $S_i \in S$?

**Known:**

- **NP-hard** [Frances & Litmann, TOCS 1999]
- $O(d^{d+1} \cdot m + nm)$-time algorithm
  
  [Gramm, Niedermeier & Rossmanith, Algorithmica 2003]
- $f(m) \cdot \text{poly}(n)$-time algorithm
  
  [Gramm, Niedermeier & Rossmanith, Algorithmica 2003]
LS-Closest String

Input: Strings $S_1, \ldots, S_m$ of length $n$ over alphabet $\Sigma$, a string $S$ with Hamming distance $\leq d$ to each $S_i \in S$, and $k \geq 0$.

Question: Is there a string $S'$ of length $n$ with Hamming distance

- $\leq k$ to $S$, and
- $< d$ to each $S_i \in S$?

$S_1 :=$ BAAAA
$S_2 :=$ CCACB
$S_3 :=$ ABBAC
**LS-Closest String**

**Input:** Strings $S_1, \ldots, S_m$ of length $n$ over alphabet $\Sigma$, a string $S$ with Hamming distance $\leq d$ to each $S_i \in S$, and $k \geq 0$.

**Question:** Is there a string $S'$ of length $n$ with Hamming distance

- $\leq k$ to $S$, and
- $< d$ to each $S_i \in S$?

\[
\begin{align*}
S_1 & := \text{BAAAAA} & d_H(S, S_1) & = 4 \\
S_2 & := \text{CCACB} & d_H(S, S_2) & = 4 \\
S_3 & := \text{ABBAC} & d_H(S, S_3) & = 3 \\
S & := \text{BBBBBB} & k & = 2
\end{align*}
\]
LS-Closest String

**Input:** Strings $S_1, \ldots, S_m$ of length $n$ over alphabet $\Sigma$, a string $S$ with Hamming distance $\leq d$ to each $S_i \in S$, and $k \geq 0$.

**Question:** Is there a string $S'$ of length $n$ with Hamming distance

- $\leq k$ to $S$, and
- $< d$ to each $S_i \in S$?

<table>
<thead>
<tr>
<th>$S_1$ := BAAAAA</th>
<th>$d_H(S, S_1) = 4$</th>
<th>$d_H(S', S_1) = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_2$ := CCACB</td>
<td>$d_H(S, S_2) = 4$</td>
<td>$d_H(S', S_2) = 3$</td>
</tr>
<tr>
<td>$S_3$ := ABBAC</td>
<td>$d_H(S, S_3) = 3$</td>
<td>$d_H(S', S_3) = 3$</td>
</tr>
<tr>
<td>$S$ := BBBBBB</td>
<td>$k = 2$</td>
<td>$d_H(S', S) = 2$</td>
</tr>
<tr>
<td>$S'$ := BBAAB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**LS-Closest String**

**Input:** Strings $S_1, \ldots, S_m$ of length $n$ over alphabet $\Sigma$, a string $S$ with Hamming distance $\leq d$ to each $S_i \in S$, and $k \geq 0$.

**Question:** Is there a string $S'$ of length $n$ with Hamming distance

- $\leq k$ to $S$, and
- $< d$ to each $S_i \in S$?

| $S_1$ := BAAAAA | $d_H(S, S_1) = 4$ | $d_H(S', S_1) = 2$ |
| $S_2$ := CCACB  | $d_H(S, S_2) = 4$ | $d_H(S', S_2) = 3$ |
| $S_3$ := ABBAC  | $d_H(S, S_3) = 3$ | $d_H(S', S_3) = 3$ |

| $S := BBBBBB$ | $k = 2$ | $d_H(S', S) = 2$ |
| $S' := BBAAB$ |

**Proposition:** **LS-Closest String** can be solved in $d^k \cdot \text{poly}(n, m)$ time.

Proof by *parameterized reduction* from **Multicolored Hitting Set**

**Input:** A hypergraph $G = (V, E)$ and vertex-coloring $c : V \rightarrow \{1, \ldots, k\}$.

**Question:** Is there a size-$k$ set $V' \subseteq V$ such that

- $V' \cap E \neq \emptyset$ for all $E \in E$, and
- $V'$ is colorful?

**Known:** **Multicolored Hitting Set** parameterized by $k$ is $W[2]$-hard
Parameterized Reduction:

\[(l, k) \xrightarrow{A} (l', k')\]

\((l, k)\) is a yes-instance \iff \((l', k')\) is a yes-instance

- \(A\) runs in \(f(k) \cdot \text{poly}(|l|)\) time
- \(k' \leq g(k)\)

A parameterized reduction from a \(W[t]\)-hard problem \(L\) to a problem \(L'\) shows \(W[t]\)-hardness of \(L'\)
\[ V = \{1, 2, 3, 4, 5, 6\} \]
\[ E = \{E_1, E_2, E_3, E_4\} \]

\[ E_1 = \{1, 2, 3\} \quad E_2 = \{4, 6\} \quad E_3 = \{1, 6\} \quad E_4 = \{3, 5\} \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_{E_1})</td>
<td>R</td>
<td>B</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(S_{E_2})</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>(S_{E_3})</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(S_{E_4})</td>
<td>X</td>
<td>X</td>
<td>R</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
</tbody>
</table>
\[ V = \{1, 2, 3, 4, 5, 6\} \]
\[ \mathcal{E} = \{E_1, E_2, E_3, E_4\} \]
\[ E_1 = \{1, 2, 3\} \quad E_2 = \{4, 6\} \quad E_3 = \{1, 6\} \quad E_4 = \{3, 5\} \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_{E_1})</td>
<td>R</td>
<td>B</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(S_{E_2})</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>(S_{E_3})</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>(S_{E_4})</td>
<td>X</td>
<td>X</td>
<td>R</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>(S_R)</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>(S_B)</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>(S_O)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
\[ V = \{1, 2, 3, 4, 5, 6\} \]
\[ \mathcal{E} = \{E_1, E_2, E_3, E_4\} \]
\[ E_1 = \{1, 2, 3\} \quad E_2 = \{4, 6\} \quad E_3 = \{1, 6\} \quad E_4 = \{3, 5\} \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_{E_1} )</td>
<td>R</td>
<td>B</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>( S_{E_2} )</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>( S_{E_3} )</td>
<td>R</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>( S_{E_4} )</td>
<td>X</td>
<td>X</td>
<td>R</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>( S_R )</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>( S_B )</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>( S_O )</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>( S )</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
\[ V = \{1, 2, 3, 4, 5, 6\} \]
\[ E = \{E_1, E_2, E_3, E_4\} \]

\[ E_1 = \{1, 2, 3\} \quad E_2 = \{4, 6\} \quad E_3 = \{1, 6\} \quad E_4 = \{3, 5\} \]

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & \mid V \\
S_{E_1} & R & B & R & X & X & X & |V| \\
S_{E_2} & X & X & X & O & X & B & |V| \\
S_{E_3} & R & X & X & X & X & B & |V| \\
S_{E_4} & X & X & R & X & O & X & |V| \\
S_R & R & R & R & R & R & R & |V| \\
S_B & B & B & B & B & B & B & |V| \\
S_O & O & O & O & O & O & O & |V| \\
S & * & * & * & * & * & * & * \\
S' & R & * & * & * & O & B & \\
\end{array}
\]
For binary alphabet:

| $S_{E_1}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| $S_{E_2}$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| $S_{E_3}$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $S_{E_4}$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $S_{\{R,B,O\}}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $S_{\{B,O\}}$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| $S_{\{R,O\}}$ | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| $S_{\{R,B\}}$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| $S_{\{R\}}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $S_{\{B\}}$ | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| $S_{\{O\}}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| $S$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $S'$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

C. Komusiewicz  (TU Berlin & Univ. Nantes)  Closest String (Binary Alphabet)
LS-Closest String:

- NP-hard
- $W[2]$-hard for radius $k$ of the Hamming neighborhood even in case $|\Sigma| = 2$
- no $n^{o(k)}$-time algorithm (assuming the Exponential Time Hypothesis)
- solvable in $d^k \cdot \text{poly}(n, m)$ time

Further results:

$W[1]$-hardness for radius $k$ of the Hamming neighborhood for local search variants of

- Longest Common Subsequence
- Shortest Common Supersequence
- Shortest Common Superstring
**LS-Longest Common Subsequence**

**Input:** Strings $T_1, \ldots, T_m$ over alphabet $\Sigma$, a string $S$ that is a subsequence of each $T_i$, and integer $k$.

**Question:** Is there a letter $\sigma \in \Sigma$ and a string $\tilde{S}$ of length $|S|$ such that $\tilde{S}\sigma$ is a subsequence of each string in $\tau$ and $d_H(\tilde{S}, S) \leq k$?

$$S_1 := \text{BCBAAACBA}$$
$$S_2 := \text{BCCACBAC}$$
$$S_3 := \text{CABBAC}$$
**LS-Longest Common Subsequence**

**Input:** Strings $T_1, \ldots, T_m$ over alphabet $\Sigma$, a string $S$ that is a subsequence of each $T_i$, and integer $k$.

**Question:** Is there a letter $\sigma \in \Sigma$ and a string $\tilde{S}$ of length $|S|$ such that $\tilde{S}\sigma$ is a subsequence of each string in $\tau$ and $d_H(\tilde{S}, S) \leq k$?

$S_1 :=$ BCBAACBA

$S_2 :=$ BCCACBAC

$S_3 :=$ CABBAC

$S :=$ ABA, $k = 1$
**LS-Longest Common Subsequence**

**Input:** Strings $T_1, \ldots, T_m$ over alphabet $\Sigma$, a string $S$ that is a subsequence of each $T_i$, and integer $k$.

**Question:** Is there a letter $\sigma \in \Sigma$ and a string $\tilde{S}$ of length $|S|$ such that $\tilde{S}\sigma$ is a subsequence of each string in $\tau$ and $d_H(\tilde{S}, S) \leq k$?

\[
S_1 := \text{BCBAAACBA} \\
S_2 := \text{BCCACBAC} \\
S_3 := \text{CABBAC}
\]

\[
S := \text{ABA, } k = 1 \\
\tilde{S}\sigma := \text{BBAC}
\]
**LS-Longest Common Subsequence**

**Input:** Strings $T_1, \ldots, T_m$ over alphabet $\Sigma$, a string $S$ that is a subsequence of each $T_i$, and integer $k$.

**Question:** Is there a letter $\sigma \in \Sigma$ and a string $\tilde{S}$ of length $|S|$ such that $\tilde{S}\sigma$ is a subsequence of each string in $\tau$ and $d_H(\tilde{S}, S) \leq k$?

\[
\begin{align*}
S_1 := & \text{BCBAAACBA} \\
S_2 := & \text{BCCACBAC} \\
S_3 := & \text{CABBAC} \\
\hline
S := & \text{ABA}, \; k = 1 \\
\tilde{S}\sigma := & \text{BBAC}
\end{align*}
\]

**Theorem:** **LS-Longest Common Subsequence** with $|\Sigma| = O(1)$ is $W[1]$-hard for the radius $k$ of the Hamming neighborhood.
So far: only negative results

Possible directions:

- other problems: **Multiple Sequence Alignment**, ... 
- other neighborhood types: “shift”-based distances, insertions/deletions at arbitrary solution positions