On Multiple Occurrences
Shortest Common Superstring Problem

Anna Gorbenko
Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov
Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

In this paper, we consider multiple occurrences shortest common superstring problem. In particular, we consider an approach to solve the problem. This approach is based on an explicit reduction from the problem to the satisfiability problem.

Keywords: multiple occurrences shortest common superstring problem, satisfiability problem, NP-complete

Usage of different regularities (see e.g. [1] – [8]) allows us to create systems of robot self-awareness (see e.g. [9] – [17]). In particular, the multiple occurrences shortest common superstring problem (MOSCS) was proposed in [18]. The problem allows us to find regularities with elements of novelty.

Let \( S = \{S_1, S_2, \ldots, S_k\} \) be the set of input words, \( S_i \in \Sigma^*, \ 1 \leq i \leq k \). Let \( |S| \) denote the total length of all words in \( S \). Let \( \#occ(U, V) \) be the number of occurrences (as a factor) of the word \( U \) in the word \( V \). The decision version of the MOSCS problem (MOSCS-D) can be formulated as following:

**Instance:** A fixed alphabet \( \Sigma \), a positive integers \( k \) and \( r \), a set of input words \( S = \{S_1, S_2, \ldots, S_k\} \), nonnegative integers \( m_1, m_2, \ldots, m_k, n_1, n_2, \ldots, n_k \) and positive integer \( m \).
**Question:** Is there a word $U$ such that $|U| \leq r$, $m_i \leq \#occ(S_i, U) \leq n_i$ for all $1 \leq i \leq k$ and $\sum_{i=1}^{k} \#occ(S_i, U) \geq m$?

Since in practice we are not interested in exponential output, we can assume that $m_i$ and $n_i$ are given as sequences of units for all $i$ such that $1 \leq i \leq k$. In this case, MOSCS-D is NP-complete [18]. In this paper, we consider reductions from MOSCS-D to SAT and 3SAT assuming that $m_1, m_2, \ldots, m_k, n_1, n_2, \ldots, n_k$ and $m$ are constants. Let $\Sigma = \{a_1, a_2, \ldots, a_p\}$; $S = \{S_1, S_2, \ldots, S_k\}$;

$$\varphi = \bigwedge_{1 \leq i \leq r}((\forall 1 \leq j \leq p)(\exists[i,j] \land (\forall 1 \leq j_1 < j_2 \leq p)(\neg x[i,j[1]] \lor \neg x[i,j[2]]));$$

$$\psi[1,i] = \bigvee_j \bigwedge_{1 \leq l \leq m_i} y[i,j[l]],$$

$$J = \{j[l] \mid 1 \leq l \leq m_i, 1 \leq j[1] < j[2] < \ldots j[m_i] \leq r + 1 - |S_i|\};$$

$$\psi[2,i] = \bigwedge_{j} \bigvee_{1 \leq l \leq m_i} \neg y[i,j[l]],$$

$$J = \{j[l] \mid 1 \leq l \leq n_i + 1, 1 \leq j[1] < j[2] < \ldots j[n_i + 1] \leq r + 1 - |S_i|\};$$

$$\psi = \bigwedge_{1 \leq i \leq k} (\psi[1,i] \land \psi[2,i]);$$

$$\tau[i,j] = \bigwedge_{0 \leq l \leq |S_i| - 1, |S_i[l+1]| = a_p} (\neg y[i,j] \lor x[j + l, q]);$$

$$\tau = \bigwedge_{1 \leq i \leq k} \bigwedge_{1 \leq j \leq r + 1 - |S_i|} \tau[i,j];$$

$$\rho[i,j] = \bigvee_{0 \leq l \leq |S_i| - 1, |S_i[l+1]| = a_p} (y[i,j] \lor \neg x[j + l, q]);$$

$$\rho = \bigwedge_{1 \leq i \leq k} \bigwedge_{1 \leq j \leq r + 1 - |S_i|} \rho[i,j];$$

$$\eta = \bigvee_{1 \leq l \leq m} y[i[l], j[l]];$$

$$J = \{(i[l][s], j[l][s]) \mid (i[l][s], j[l][s]) \in \{(i, j) \mid 1 \leq i \leq k, 1 \leq j \leq r + 1 - |S_i|\},$$

$$s \neq t \rightarrow (i[l][s], j[l][s]) \neq (i[l][t], j[l][t]); 1 \leq s \leq m\};$$

$$\xi = \varphi \land \psi \land \tau \land \rho \land \eta.$$

In $\xi$ formula $\varphi$ can be considered as a choice of a common superstring, for all $i$ formula $\psi[1,i] \land \psi[2,i]$ can be considered as a choice of positions of word $S_i$ in the common superstring. It is easy to check that there is a word $U$ such that $|U| \leq r$, $m_i \leq \#occ(S_i, U) \leq n_i$ for all $1 \leq i \leq k$ and $\sum_{i=1}^{k} \#occ(S_i, U) \geq m$ if and only if $\xi$ is satisfiable. Since $m_1, m_2, \ldots, m_k, n_1, n_2, \ldots, n_k$ and $m$ are constants, it is possible to apply laws of distributivity to $\psi$ and $\eta$. Using this transformation $\xi$ can be represented in conjunctive normal form $\xi'$. This gives us a reduction from MOSCS-D to SAT.

Using standard transformations (see e.g. [19]) we can easily obtain an explicit transformation $\xi'$ into $\xi$ such that $\xi' \Leftrightarrow \xi$ and $\xi$ is a 3-CNF. It is clear that $\xi$ gives us an explicit reduction from MOSCS-D to 3SAT.

There is a well known site on which posted solvers for SAT [20]. They are divided into two main classes: stochastic local search algorithms and algorithms improved exhaustive search. All solvers allow the conventional format.
for recording DIMACS boolean function in conjunctive normal form and solve the corresponding problem. In addition to the solvers the site also represented a large set of test problems in the format of DIMACS. This set includes a randomly generated problems of 3SAT. We create a generator of natural instances for MOSCS-D. Also we use test problems from [20]. We use algorithms from [20]: fgrasp and posit. Also we design our own genetic algorithm (OA) for SAT which based on algorithms from [20]. Each test was run on a cluster of at least 100 nodes. Selected experimental results are given in Table 1.

<table>
<thead>
<tr>
<th>time</th>
<th>brute force</th>
<th>fgrasp</th>
<th>posit</th>
<th>OA</th>
</tr>
</thead>
<tbody>
<tr>
<td>average</td>
<td>27.2 h</td>
<td>32 min</td>
<td>36 min</td>
<td>2.8 min</td>
</tr>
<tr>
<td>max</td>
<td>29.1 h</td>
<td>15.16 h</td>
<td>11.37 h</td>
<td>1.26 h</td>
</tr>
<tr>
<td>best</td>
<td>26.4 h</td>
<td>16.1 min</td>
<td>11.12 min</td>
<td>80 sec</td>
</tr>
</tbody>
</table>

Table 1: Experimental results for 3SAT

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program “Developing the scientific potential of high school” 8.1616.2011.

References


Received: November 1, 2012