THE MINIMUM LENGTH PERMUTATION
SEQUENCE IS NP-HARD

by

S. Even\ and O. Goldreich

Technical Report #194

January 1981

* The research was supported by the Fund for the Promotion of
Research at the Technion
ABSTRACT

Given a set of permitted permutations and a target permutation, we consider the problem of finding a shortest sequence of permitted permutations which generates the target permutation.

This problem is shown to be NP-hard even if the set of permitted permutations is closed under inversion.
1. INTRODUCTION

A natural problem which arises in connection with the isomorphism problem for graphs of bounded degree [1], and in connection to the Rubik-color-cube puzzle [2] is the following. A set of permitted permutations is given, together with a target permutation. It is necessary to determine whether the target permutation can be realized by some finite sequence of permitted permutations. This problem was solved, in polynomial time by Furst, Hopcroft and Luks [1].

Our purpose is to show that the problem of finding the length of a shortest sequence of the permitted permutations which generates the target permutation, is NP-hard; we actually show that the problem of determining whether a given length sequence exists, is NP-complete.

We also show that the problem remains NP-complete even if the set of permitted permutations is restricted to include for each permutation its inverse.

2. MINIMUM LENGTH PERMUTATION SEQUENCE WHICH GENERATES A GIVEN PERMUTATION IN NP-COMPLETE

Let \( Z = \{1, 2, \ldots, z\} \). Given a set of permitted permutations \( \{P_i\}_{i=1}^{k} \) of \( Z \), a target permutation \( P \) (of \( Z \)) and an integer \( \ell \), determine whether there exists a sequence \( i_1, i_2, \ldots, i_\ell \) (\( i_j \in \{1, 2, \ldots, k\} \)) such that

\[
P_{i_1}P_{i_2}\cdots P_{i_\ell} = P,
\]

where the left hand side is the permutation resulting from successive application of \( P_{i_1}, P_{i_2}, \ldots \text{ etc. on } Z \).
Let us call this the MLPS problem (Minimum Length Permutation Sequence).

**Theorem 1:** MLPS is NP-complete.

**Proof:** It is easily seen that MLPS ∈ NP. We complete the proof by showing that 3SAT ≤ MLPS.

### 2.1 The Reduction:

Let the set of literals of 3SAT be $L = \{v_1, v_2, \ldots, v_n; \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ and the set of clauses be $C = \{C_1, C_2, \ldots, C_m\}$, each containing at most 3 literals.

First we add $n$ clauses $C' = \{C_{m+1}, C_{m+2}, \ldots, C_{m+n}\}$, where

- $C_{m+i} = \{v_i, \bar{v}_i\}$.

Let $z = (n+m) \cdot 2 + 2n \cdot (n+m+1)$, (and $Z = \{1, 2, \ldots, z\}$). We partition $Z$ into fields as follows:

- For $0 < j < m+n$, $(2j-1, 2j)$ is $C_j$-field.
- For $1 < i < n$, $((n+m) \cdot 2 + (i-1) \cdot 2(n+m+1) + k \cdot (n+m+1))$ is $v_i$-field and
  
- $((n+m) \cdot 2 + (i-1) \cdot 2(n+m+1) + k \cdot 2(n+m+1))$ is $\bar{v}_i$-field.

For every occurrence of literal $v_i [\bar{v}_i]$ in clause $C_j$ define $SP(v_i, C_j)$ [$SP(\bar{v}_i, C_j)$] to be the permutation which permutes the two elements of $C_j$-field and shifts the elements of $v_i$-field [$\bar{v}_i$-field] cyclicly, one step to the right. These permutations are called the satisfying permutations.

For every literal $v_i [\bar{v}_i]$ define $n+m$ correcting permutations $\{CP(v_i, k)\}_{k=1}^{n+m}$ [$\{CP(\bar{v}_i, k)\}_{k=1}^{n+m}$]. $CP(v_i, k)$ [$CP(\bar{v}_i, k)$] shifts the elements of $v_i$-field [$\bar{v}_i$-field] cyclicly, $k$ steps to the right.
Let the target permutation $P$ be the permutation in which every $C_j$-field is permuted while all other fields stay put.

Define $\lambda = 2n+m$.

## 2.2 Validity of the Reduction

The following facts are of interest.

**Fact 1:** Every permutation (of the SP or CP type) operates on one literal field only.

**Fact 2:** Every SP permutation operates on one clause field, and every CP permutation leaves all clause fields invariant.

**Fact 3:** Every two permitted permutations (of either type) commute.

**Fact 4:** If one applies $n+m$ satisfying permutations, one for each clause, then for every $i$, either $v_i$-field or $\bar{v}_i$-field or both, are disturbed. [This follows from the fact that $v_i$-field contains $n+m+1$ elements.]

**Fact 5:** The target permutation $P$ cannot be generated by a sequence of (SP and CP) permutations shorter than $2n+m$. [It takes $n+m$ satisfying permutations to permute all the clause fields (Fact 2). Since this disturbs at least $n$ literal fields (Fact 4), one needs at least $n$ correcting permutations to restore the literal fields (Fact 1).]

Now, let us prove that if the 3SAT instance is satisfiable then the target permutation of the MLPS instance is generated by a sequence of length $\lambda$.

Choose a satisfying assignment for the 3SAT instance. Clearly, every assignment satisfies all the clauses of $C'$ automatically. For each clause $C_j$, $1 \leq j \leq n+m$, choose one literal, $u_j$ (either $v_j$ or $\bar{v}_j$)
which satisfies it. Apply \( SP(u_i, C_j) \).

This way, \( n+m \) satisfying permutations have been chosen, and their application to \( Z \) permutes all the clause fields. Also, by the assignment's consistency, for every \( i \) at most one of \( v_i \)-field or \( \bar{v}_i \)-field are disturbed. Thus, at most \( n \) correcting permutations are necessary to restore all the literal fields. By Fact 5, exactly \( 2n+m \) permutations will be applied.

Next, let us show that if the target permutation of the MLPS instance is achievable by a \( 2n+m \) sequence then the source 3SAT instance is satisfiable.

Since each clause field is permuted in \( P \), there are, for every \( 1 \leq j \leq n+m \), an odd number of satisfying permutations of type \( SP(\cdot, C_j) \) in the sequence (Fact 2). For each \( j \), choose one such \( SP(\cdot, C_j) \) of the sequence. We may assume that these \( n+m \) satisfying permutations are applied first (Fact 3). By Fact 4, for every \( 1 \leq i \leq n \), either \( v_i \)-field and \( \bar{v}_i \)-field or both are disturbed. However, only \( n \) permutations remain and, by Fact 1, only \( n \) literal fields can be restored. Thus, for every \( i \), not both \( v_i \)-field and \( \bar{v}_i \)-field have been disturbed.

Let us define an assignment for the 3SAT instance as follows:

If \( SP(u_i, C_j) \) has been applied above, (among the first \( n+m \) permutations) then assign 'true' to \( u_i \). This assignment is satisfying since for each clause we apply a satisfying permutation among the first \( n+m \) permutations. It is consistent, since it was shown above that not both \( v_i \)-field and \( \bar{v}_i \)-field were disturbed by the first \( n+m \) permutations.

Q.E.D.
3. THE PROBLEM REMAINS \textsc{np}-complete even if the set of permitted permutations is closed under inversion.

The problem of finding a shortest solution to the Rubik cube puzzle is a special case of the \textsc{mlps} problem. However, in this puzzle, the set of permitted permutations includes for every permutation its inverse as well. One may suspect that this constraint may simplify the problem. Our purpose is to show that the \textsc{mlps} problem remains \textsc{np}-complete even if the set of permitted permutations is closed under inversion.

Let us call this restricted \textsc{mlps} problem, RMLPS.

**Theorem 2**: RMLPS is \textsc{np}-complete.

**Proof**: Again, it is easy to see that RMLPS $\in \textsc{np}$. Also, we show that 3SAT $\leq$ RLMPS. The definition of 3SAT remains as above. $Z$ is defined similarly, except that the $C_j$-fields are extended to include 7 integers (instead of 2) and therefore
\[
z = (n+m) \cdot 7 + 2n \cdot (n+m+1).
\]

SP($u_i, C_j$) is defined to be the permutation on $Z$ which shifts cyclicly, one place to the right, the $C_j$-field and the $u_i$-field (where $u_i$ is the literal $v_i$ or $\bar{v}_i$).

DP($u_i, C_j$), called a \textit{dummy permutation}, is the inverse of SP($u_i, C_j$).

CP($u_i, k$) is defined exactly as before. Notice that the CP($u_i$) set of permutations is closed under inversion.

Clearly, the set of permitted permutations consisting of the three types above, is closed under inversion.

The target permutations, $P$, is defined to be the permutation which shifts each of the $C_j$-fields, cyclicly, one place to the right, and leaves the $u_i$-fields invariant. The value of $i$ remains as before $(2n+m)$.
The proof that the satisfiability of the 3SAT instance implies the existence of a solution to the corresponding instance of RMLPS, is similar to the one in the previous proof; dummy permutations are not used.

Next, we show that if the RMLPS instance has a solution then the corresponding 3SAT instance is satisfiable.

First, let us assume that the sequence of permitted permutations which generates $P$ is free of dummy permutations. In this case the proof that the 3SAT instance is satisfiable remains as in the previous section.

Let us show that, without loss of generality, this assumption can be made.

Let $\sigma$ be a sequence of permitted permutations, of length $\lambda$, which generates $P$. Denote by $\#SP_\sigma(j)$ the number of satisfying permutations in $\sigma$, of type $SP(\cdot, C_j)$. Similarly, $\#DP_\sigma(j)$ denotes the number of dummy permutations in $\sigma$, of type $DP(\cdot, C_j)$.

**Fact 1:** For every $\sigma$ and every $j$

$$\#SP_\sigma(j) - \#DP_\sigma(j) \equiv 1 \pmod{7}.$$  

**Fact 2:** If $\#SP_\sigma(j) > 0$ and $\#DP_\sigma(j) > 0$ then $\sigma'$, which results from the replacement in $\sigma$, of a pair $SP(u_1, C_j)$, $DP(u_2, C_j)$ by the pair $CP(u_1, 1)$, $CP(u_2, n+m)$, is also a solution to the RMLPS instance.

**Fact 3:** If there exists a solution to the RMLPS instance then there exists one, $\sigma$, in which for every $j$ one and only one of the following conditions holds:

(a) $\#SP_\sigma(j) \equiv 1 \pmod{7}$ and $\#DP_\sigma(j) = 0$.

(b) $\#SP_\sigma(j) = 0$ and $\#DP_\sigma(j) \equiv 6 \pmod{7}$.

This can be proved by Fact 1 and repeated use of Fact 2.
Let \( h = \lfloor \{ j \mid |D_{D_4}(j) > 0 \} \rfloor \).

It remains to be shown that \( h = 0 \).

By Fact 3, \( n + m - h \) clauses are satisfied in \( \sigma \) by satisfying permutation. Let us choose, from \( \sigma \), one such satisfying permutation for each of these clauses, and denote the set of permutation thus chosen by \( A \). Clearly \( |A| = n + m - h \).

The number of literal-fields disturbed by \( A \) is at least \( n - h \), since of the \( n \) clause-fields which belong to \( C' \) at most \( h \) are shifted by dummy permutations.

Let us choose, now, 6 dummy permutations per clause-field which is satisfied by dummy permutations; by Fact 3, there are at least 6 such permutations for each of the \( h \) clause-fields which are satisfied by dummy permutations. Denote by \( B \) the set of dummy permutations thus chosen. Thus, \( |B| = 6h \).

For each of the \( h \) clause-fields shifted by elements of \( B \), at most 3 literals-fields are shifted; since each clause contains at most 3 literals. Thus, after the application of \( A \) and \( B \) at least \( n - h - 3h = n - 4h \) literal-fields remain disturbed.

The number of permutations which remain in \( \sigma \) after the application of \( A \) and \( B \) is

\[
2n + m - (n + m - h) - 6h = n - 5h.
\]

But the number of literal-fields which remain disturbed should not exceed the number of remaining permutations (since each permitted permutation can restore at most one literal field). Thus,

\[
 n - 5h \geq n - 4h,
\]

and \( h = 0 \).

Q.E.D.
REFERENCES
