A ‘Hot Potato’ Gray Code for Permutations

Xi Sisi Shen\textsuperscript{1,3}

\textit{Department of Mathematics and Statistics}\n\textit{McGill University}\n\textit{Montreal, Canada}

Aaron Williams\textsuperscript{2}

\textit{Department of Mathematics and Statistics}\n\textit{McGill University}\n\textit{Montreal, Canada}

\textbf{Abstract}

We give the $n!$ permutations of $[n] = \{1, 2, \ldots, n\}$ a cyclic order inspired by the children’s game of Hot Potato. Our order is a transposition Gray code, meaning that consecutive permutations differ by a single transposition. Furthermore, each transposition is restricted in two ways: (1) It must transpose value $n$ (the “hot potato”); (2) It must transpose positions that are circularly adjacent or semi-adjacent. In other words, if each permutation is written circularly, then our order repeatedly transposes the value $n$ with a value that is one or two positions to its left or right.

\textit{Keywords:} Gray code, permutation, star transposition, adjacent transposition, vertex transitive graph, Hamilton cycle, Lovász conjecture

\section{Introduction}

Let $\Pi(n)$ be the set of permutations of $[n] = \{1, 2, \ldots, n\}$ written as strings in one-line notation. For example, $\Pi(3) = \{123, 132, 213, 231, 312, 321\}$. A permutation \textit{Gray code} is an order of $\Pi(n)$ in which successive permutations
differ by a fixed type of string operation, such as a transposition, rotation, or reversal. A Gray code is cyclic if the chosen operation can also transform the last permutation into the first permutation. Permutation Gray codes have been surveyed in Sedgewick [7], Savage [6], and Section 7.2.1.2 of Knuth [2].

We describe a new Gray code in terms of the Hot Potato game [4]. Suppose \( n \) children are positioned around a circle, with the \( i \)-th child initially holding a potato of value \( i \). The potato with value \( n \) is ‘hot’ and so the \( n \)-th child trades potatoes with another child, who then trades the hot potato with another child, and so on. Since children have short arms, each trade can move the hot potato at most \( d \) positions to the left or right around the circle. For example, Figure 1 shows the transpositions that are initially allowable for \( n = 8 \) and \( d = 2 \). After each trade, a permutation \( p_1p_2\ldots p_n \in \Pi(n) \) is obtained by setting each \( p_i \) to the value of the potato held by player \( i \). A hot potato Gray code with maximum distance \( d \) is a Gray code using the allowable transpositions.

![Figure 1](image)

Fig. 1. When the maximum distance is \( d = 2 \), the hot potato \( n \) can only be traded with potatoes that are one or two positions to its left or right. Thus, in a hot potato Gray code with \( d = 2 \), the permutation in a) must be followed by b), c), d), or e).

What is the minimum possible value for \( d \)? Consider the hot potato graph with vertices \( \Pi(n) \) and edges for allowable transpositions. The graph is disconnected for \( d = 1 \) and \( n \geq 4 \), as seen below for \( n = 4 \).

\[
\begin{array}{cccccccc}
1234 & 1243 & 1423 & 4123 & 3124 & 3142 & 2134 & 2143 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4231 & 2431 & 2341 & 2314 & 4312 & 3412 & 4132 & 1342 \\
\end{array}
\]

In this article, we give a hot potato Gray code with the minimum \( d = 2 \).

## 2 Transposition Gray Codes

In this section we review three transposition Gray codes that contribute to our result in Section 3. A transposition is a pair of distinct positive integers \((i, j)\), where \( i, j \in [n] \) in the context of \( \Pi(n) \). If \( p = p_1p_2\ldots p_n \in \Pi(n) \), then applying \((i, j)\) to \( p \) creates a new permutation in \( \Pi(n) \) by interchanging \( p_i \) and \( p_j \). A transposition \((i, j)\) with \( i < j \) has distance \( j - i \) and circular distance...
min\((j - i, n + i - j)\). Adjacent and semi-adjacent transpositions have distance 1 and 2, respectively. Similarly, circularly adjacent and circularly semi-adjacent transpositions have circular distance 1 and 2, respectively. Transpositions of the form \((i \, n)\) are star transpositions (this term comes from the star graph).

<table>
<thead>
<tr>
<th>Type</th>
<th>Restriction</th>
<th>Transpositions</th>
</tr>
</thead>
<tbody>
<tr>
<td>star</td>
<td>position (n)</td>
<td>((1 , n), (2 , n), \ldots, (n-1 , n))</td>
</tr>
<tr>
<td>adjacent</td>
<td>distance 1</td>
<td>((1 , 2), (2 , 3), \ldots, (n-1 , n))</td>
</tr>
<tr>
<td>semi-adjacent</td>
<td>distance 2</td>
<td>((1 , 3), (2 , 4), \ldots, (n-2 , n))</td>
</tr>
<tr>
<td>circularly adjacent</td>
<td>circular distance 1</td>
<td>((1 , 2), (2 , 3), \ldots, (n , 1))</td>
</tr>
<tr>
<td>circularly semi-adjacent</td>
<td>circular distance 2</td>
<td>((1 , 3), (2 , 4), \ldots, (n , 2))</td>
</tr>
</tbody>
</table>

A transposition Gray code orders \(\Pi(n)\) so that consecutive strings differ by a transposition. A partial Gray code is such an order for a subset of \(\Pi(n)\).

An adjacent transposition Gray code was known to bell ringers in the 17th century, and was rediscovered by Steinhaus, Johnson, and Trotter [2]. In this plain changes order of \(\Pi(n)\), adjacent transpositions move \(n\) through each string in \(\Pi(n - 1)\), from left-to-right or right-to-left. When \(n\) reaches the leftmost or rightmost position, the string in \(\Pi(n - 1)\) is changed according to the plain change order for \(\Pi(n - 1)\). For example, the order for \(n = 3\) is 321, 231, 213, 123, 132, 312 as in the \(n = 4\) order below. Algorithmically, the order’s repetitive pattern allows for efficient permutation generation [2].

\[
\begin{align*}
4321, & 3412, 3214, 3241, 2314, 2341, 2134, 2143, 2132, 2142, 2131, 1234, 1243, 1423, 1432, 1432, 1342, 1324, 3124, 3142, 3142, 4312,
\end{align*}
\]  \( (1) \)

Adjacent transpositions \((i \, i+1)\) and \((j \, j+1)\) are doubly-adjacent if \(|i - j| = 1\). In a doubly-adjacent Gray code consecutive adjacent transpositions are doubly-adjacent. We use the following theorem by Compton and Williamson [1].

**Theorem 2.1** ([1]) There exists a cyclic doubly-adjacent Gray code for \(\Pi(n)\).

We associate the two sequences with each cyclic doubly-adjacent partial Gray code \(q_1, q_2, \ldots, q_k\): (1) The transposition sequence is \(t_1, t_2, \ldots, t_k\) such that \(q_i\) and \(q_{i+1}\) differ by the adjacent transposition \((t_i \, t_{i+1})\); (2) The ± sequence is \(s_1, s_2, \ldots, s_k\) such that \(t_{i+1} = t_i + 1 \implies s_i = +\), and \(t_{i+1} = t_i - 1 \implies s_i = -\). (The subscripts are treated circularly, so \(k+1\) represents 1 in both cases.) These sequences are illustrated for a partial Gray code of \(\Pi(4)\) below.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_i)</td>
<td>4321</td>
<td>4312</td>
<td>4132</td>
<td>1432</td>
<td>1342</td>
<td>3142</td>
<td>3412</td>
<td>3421</td>
<td>3241</td>
<td>2341</td>
<td>2431</td>
<td>4231</td>
</tr>
<tr>
<td>(t_i)</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>(s_i)</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Star transposition Gray codes were constructed cyclically by Kompel’makher and Liskovets [3] and Ruskey and Savage [5], and efficiently generated non-cyclically by Ehrlich and later Knuth [2]. A cyclic order for \( n = 4 \) is below

\[
\begin{align*}
4321, & \ 1324, \ 1423, \ 3124, \ 3412, \ 4132, \ 4231, \ 1234, \ 1432, \ 2431, \ 2134, \ 2143, \ 3142, \ 3241, \ 3214, \ 2341, \ 2314, \ 3412, \ 3214, \ 4312, \ 4321.
\end{align*}
\]  

(2)

The inverse of \( p_1p_2\cdots p_n \in \Pi(n) \) is \( q_1q_2\cdots q_n \in \Pi(n) \) such that \( p_j = i \iff q_i = j \). In other words, the inverse is an involution that interchanges positions and values. By taking inverses the adjacent transposition order in (1) becomes: 4321, 4312, 4213, 3214, 2314, . . . . Observe that adjacent values are always transposed. Similarly, inverses of the star transposition order in (2) cause the value \( n = 4 \) to always be transposed: 4321, 1324, 1423, 3412, 3241, 2341, 2314, 3412, 3214, 4312, 4321, . . . . More generally, Gray codes using position-restricted transpositions are inverse to Gray codes using value-restricted transpositions, and vice versa. In Section 3, we give a Gray code that is simultaneously restricted by position and value.

3 A Hot Potato Gray Code with \( d = 2 \)

We create a cyclic hot potato Gray code with maximum distance \( d = 2 \). Thus, successive permutations in our order differ by a circularly adjacent or semi-adjacent transposition involving value \( n \). We refer to these transpositions as trades, and denote them as \( \rightarrow \) for right, \( \leftarrow \) for left, \( \rightarrow \) for 2-right, and \( \iff \) for 2-left, depending on value \( n \)'s movement. For example, \( \rightarrow (654321) = 456321 \) is a 2-right trade since \( n \) is transposed two positions to the right. Similarly, \( \rightarrow (123456) = 623451 \) is a right trade since \( n \) is transposed one (circular) position rightward. Exponents denote repetition, so \( \rightarrow^2 (654321) = \rightarrow^1 (564321) = 546321 \). Right trades applied to \( p_1p_2\cdots p_n \in \Pi(n) \) are summarized below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbol</th>
<th>Transpositions</th>
<th>Transposition Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>right</td>
<td>( \rightarrow )</td>
<td>(1 2), (2 3), . . . , (n 1)</td>
<td>( p_i = n ) for ( i = 1, 2, \ldots, n )</td>
</tr>
<tr>
<td>2-right</td>
<td>( \rightarrow )</td>
<td>(1 3), (2 4), . . . , (n 2)</td>
<td>( p_i = n ) for ( i = 1, 2, \ldots, n )</td>
</tr>
</tbody>
</table>

Using this terminology, the plain change order from Section 2 repeatedly applies \( n - 1 \) consecutive right (or left) trades. By taking advantage of the trade \( (1 \ n) \), we can instead apply \( n \cdot (n-1) - 1 \) consecutive right (or left) trades. We denote this process as follows, where \( p \in \Pi(n) \) and \( s \in \{+, -\} \) is the direction

\[
\text{trades}(p, s) = \begin{cases} 
  p, \rightarrow (p), \rightarrow^2 (p), \ldots, \rightarrow^{n-(n-1)-1} (p), & \text{if } s = + \\
  p, \leftarrow (p), \leftarrow^2 (p), \ldots, \leftarrow^{n-(n-1)-1} (p), & \text{if } s = - 
\end{cases}
\]
For example, trades(54321, +) moves 5 through 4321, 3214, 2143, 1432 as below,

\[
\begin{array}{cccccccccccc}
5 & 4 & 3 & 2 & 1
\end{array}
\]

\[
\begin{array}{cccccccccccc}
4 & 3 & 2 & 1 & 5
\end{array}
\]

\[
\begin{array}{cccccccccccc}
3 & 2 & 1 & 5 & 4
\end{array}
\]

\[
\begin{array}{cccccccccccc}
2 & 1 & 5 & 4 & 3
\end{array}
\]

\[
\begin{array}{cccccccccccc}
1 & 5 & 4 & 3 & 2
\end{array}
\]

\[
\begin{array}{cccccccccccc}
5 & 4 & 3 & 2 & 1
\end{array}
\]

Notice that \( n \) is inserted into successive rotations of a given string in \( \Pi(n-1) \), as illustrated above. The process is also cyclic. For example, one more right trade creates \( 14325 = 54321 \) above. Thus, \( \Pi(n) \) partitions into \((n-2)!\) trade cycles of length \( n \cdot (n-1) \). (Trade cycles are simply the disjoint cycles in the hot potato graph with \( d = 1 \).) Theorem 3.1 proves that trade cycles can be spliced together using doubly-adjacent partial Gray codes for \( \Pi(n-2) \). Figure 2 illustrates the technical proof.

**Proof:** Given a cyclic doubly-adjacent partial Gray code \( \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_2 \) for \( \Pi(4) \) from the table in Section 2 is transformed into a cyclic hot potato partial Gray code with distance \( d = 2 \) for \( 6 \cdot 5 \cdot 12 = 360 \) strings in \( \Pi(6) \). The \( i \)th column is a trade cycle from \( \mathbf{a}_i \) to \( \mathbf{b}_i \), and is read down for \( s_i = + \) and up for \( s_i = - \).

**Theorem 3.1** There exists a cyclic hot potato Gray code with \( d = 2 \) for \( \Pi(n) \).

**Proof:** Given a cyclic doubly-adjacent partial Gray code \( \mathbf{Q} = \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k \) of \( \Pi(n-2) \), we prove there is a \( d = 2 \) hot potato partial Gray code for \( n \cdot (n-1) \cdot k \) strings in \( \Pi(n) \). (Theorem 3.1 then follows from Theorem 2.1 and \( k = (n-2)! \).)

Let \( t_1, t_2, \ldots, t_k \) and \( s_1, s_2, \ldots, s_k \) be the sequences associated with \( \mathbf{Q} \). Let \( \mathbf{p}_i = n - 1 \mathbf{q}_i \) and \( \mathbf{a}_i \rightarrow t_{i+1} (\mathbf{p}_i) \) for all \( 1 \leq i \leq k \). Now consider

\[
\mathcal{G} = \text{trades}(\mathbf{a}_1, s_1), \text{trades}(\mathbf{a}_2, s_2), \ldots, \text{trades}(\mathbf{a}_k, s_k).
\]

By the choices of \( \mathbf{p}_i \) and \( \mathbf{a}_i \), \( \mathcal{G} \) contains \( n \cdot (n-1) \cdot k \) distinct elements of \( \Pi(n) \) and consecutive permutations in each trade cycle differ by a circularly-adjacent trade. Let \( \mathbf{b}_1 \) be the last permutation in \( \text{trades}(\mathbf{a}_i, s_i) \). We complete the proof by showing \( \mathbf{b}_1 \) and \( \mathbf{a}_{i+1} \) differ by a trade. Let \( \mathbf{q}_i = x_1, x_2, \ldots, x_{n-2} \) and note that \( \mathbf{q}_{i-1} = x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_i, x_{i+2}, x_i, x_{i+3}, \ldots, x_{n-2} \). Since \( \mathbf{q}_i \) and \( \mathbf{q}_{i-1} \) differ by \((t_i, t_{i+1})\).

The cases \( s_i = + \) and \( s_i = - \) appear below on the left and right, respectively.
\[
\begin{align*}
\rightarrow (b_i) &= \rightarrow \left( \rightarrow (a_i) \right) \\
&= \rightarrow \left( \rightarrow \left( \rightarrow^{t_i+1} (p_i) \right) \right) \\
&= \rightarrow^{t_i} (p_i) \\
&= \rightarrow^{t_i} (n n \rightarrow 1 q_i) \\
&= \rightarrow^{t_i} (n n \rightarrow 1 x_1 \cdots x_{n-2}) \\
&= \rightarrow^{t_i+2} (n n \rightarrow 1 q_{i+1}) \\
&\rightarrow^{t_i+2} (p_{i+1}) \\
&= a_{i+1}.
\end{align*}
\]

References


