An Algorithm for the Factorization of Permutations on a Tree

Theresa P. Vaughan
Department of Mathematics
University of North Carolina at Greensboro
Greensboro, NC 27412

Frederick J. Portier
Department of Mathematics and Computer Science
Mount Saint Mary's College
Emmitsburg, MD 21727

ABSTRACT. In this paper we consider a permutation $\sigma \in S_n$ as acting on an arbitrary tree with n vertices (labeled $1, 2, 3, \ldots, n$). Each edge $[a, b]$ of T corresponds to a transposition $(a, b) \in S_n$, and such a "tree of transpositions" forms a minimal generating set for S_n. If $\sigma \in S_n$, then σ may be written as a product of transpositions from $T, \sigma = t_k t_{k-1} \ldots t_2 t_1$. We will refer such a product as a T-factorization of σ of length k. The primary purpose of this paper is to describe an algorithm for producing T-factorizations of σ. Although the algorithm does not guarantee minimal factorizations, both empirical and theoretical results indicate that the factorizations produced are "nearly minimal". In particular, the algorithm produces factorizations that never exceed the known upper bounds.

1. Introduction.

In this paper we consider a permutation $\sigma \in S_n$ as acting on an arbitrary tree with n vertices (labeled $1, 2, 3, \ldots, n$). Each edge $[a, b]$ of T corresponds to a transposition $(a, b) \in S_n$, and such a "tree of transpositions" forms a minimal generating set for S_n [1] (this result goes back as far as Cayley). It is convenient to abuse notation somewhat and use the letter T to represent both the set of transpositions generating S_n and the set of edges of the corresponding tree. To introduce the topic, we make some brief definitions here; the full details may be found in Section 2.

JCMCC 18 (1995), pp. 11-31

If $\sigma \in S_n$, then σ may be written as a product of transpositions from $T, \sigma = t_k t_{k-1} \dots t_2 t_1$. We will refer such a product as a *T-factorization* of σ of length k. The minimum value for k is called the *T-rank* of σ. The primary purpose of this paper is to describe an algorithm for producing T-factorizations of σ. Although the algorithm does not guarantee minimal factorizations, both empirical and theoretical results indicate that the factorizations produced are "nearly minimal". In particular, the algorithm produces factorizations that never exceed the known upper bounds (due to Vaughan [7]). In two special cases, there are known methods for finding T-factorizations of minimal length; the case when T is a path is probably the better known, and serves as a good illustration of the general problem.

Special Case: The Path

A path, T, has edges $\{[i, i+1] \mid i = 1 \dots n-1\}$. This special case has been extensively studied and a great deal is known about it. The T-rank of a permutation σ is the number of "inversions" in σ, and a minimal T-factorization may be found by successively applying transpositions which reduce the number of inversions. Riordan [6] and Knuth [4] are standard references. Edelman [3] recently studied the relationship between the T-rank of σ and its cycle structure.

In this case, the geometry of T is that of a straight line. An inversion pair of σ is a pair $\{i, j\}$ such that $\{\sigma(i), \sigma(j)\}$ is "out of order", i.e., $i < j$ and $\sigma(i) > \sigma(j)$. We may imagine the tree, T, with n labels attached where a factorization causes each label i to "travel" from the vertex originally labeled by i, to the vertex labeled by $\sigma(i)$. Each label can move only one edge at a time and this is done by trading places with the label on the opposite end of the edge. The geometry of T insures that each label i trades places with every label j for which $\{i, j\}$ is an inversion pair. For example, for $n = 5$, if $\sigma(1) = 3$, and $\sigma(3) = 2$, we have that $\{3, 2\}$ is an inversion pair (see Figure 1). After the transposition $(2, 3)$ is applied, the labels 2 and 3 change places. This can all be made precise and leads to the characterization of the T-rank of σ as the number of inversion pairs.

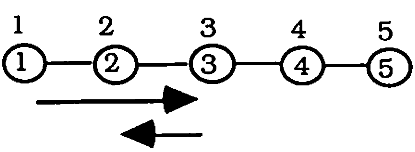


Figure 1: Path with labels attached.

Special Case: The Star.

A star represents the other extreme case among the set of all trees on n vertices. This tree has one central vertex of degree $n-1$ and a collection of $n-1$ vertices of degree 1 (see Figure 2). In Portier and Vaughan [5], this case is studied in detail and, among other things, a simple algorithm for minimal T-factorizations is described. In some ways, this case is the simplest of all since no label has to travel over more than two edges. In particular, the T-rank of σ is the sum of the T-ranks of the disjoint cycles of σ. Furthermore, a cycle, C, of length m has T-rank $m-1$ if the center vertex is moved by C and $m+1$ if it is not. There is, seemingly, no good way to define a notion analogous to inversion pair when T is a star. The difficulty stems from the fact that, for any choice of labels i and j, there are factorizations in which i and j trade places over a given edge and others where they do not.

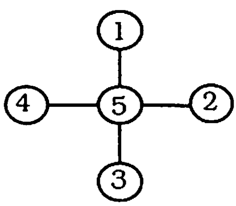


Figure 2: A star.

The General Case.

For a general tree T, some upper and lower bounds for the T-rank of σ are found in [7], but the problem of finding the exact T-rank of σ (other than by an exhaustive search) remains open, as does the related problem of finding useful necessary and sufficient conditions for a T-factorization of σ to have minimal length.

As stated earlier, the main purpose of this paper is to present an algorithm which produces T-factorizations for a general tree T. At each step of the algorithm a transposition from T is applied until the composition results in the identity. The algorithm has a number of appealing properties:

- an easily computed number related to the rank of σ is reduced by 2 at every step
- it satisfies a provable necessary condition for minimality
- geometric intuition and all of our data so far, suggests strongly that every minimal T-factorization could be produced by it

- the T-factorizations produced by it are (in all our data so far) nearly minimal
- the algorithm always produces a minimal T-factorization if T is a path or a star

At any given step, the algorithm may identify several possible transpositions, only one of which may be applied. At present, there is no theoretical basis for identifying which, if any, of the possible transpositions will insure a reduction in rank. Geometrical considerations suggest several heuristics.

After a few preliminaries, we define the VPA algorithm in Section 3, and prove that it can be applied to every permutation in S_n, and that it always terminates in the identity. As a consequence, we find rough upper and lower bounds for the T-rank of σ. In Section 4, we discuss some of the detailed geometric aspects of σ acting on T, and prove a necessary condition for a T-factorization of σ to be of minimal length. In Section 5, we prove that all the T-factorizations produced by the algorithm satisfy this necessary condition. We also prove that they satisfy another condition which seems to be desirable for minimality, though we have not been able to prove that it is necessary. Finally, in Section 6, several heuristics are defined and their performance on small permutations is described.

We shall be using the elementary theory and terminology of finite graphs and trees (see [1,2]) and finite permutation groups (see [1,6,8]).

2. Preliminaries.

If n is a positive integer, S_n denotes the symmetric group on the set $\{1, 2, \ldots, n\}$. It is well-known that a set of transpositions of S_n, say

$$T = \{(a_i, b_i) \mid i = 1, 2, \ldots, k\}$$

is a minimal generating set for the group S_n if and only if $k = n - 1$, and the graph with vertex set $\{1, 2, \ldots, n\}$ and edge set T is a tree.

We assume throughout that $T = \{(a_i, b_i) \mid i = 1, 2, \ldots, n-1\}$ is a minimal generating set of transpositions for S_n. The *degree* of a vertex v of T is the number of edges containing v, and an *outer vertex* of T is a vertex of degree one. If $\sigma \in S_n$, then the least number m such that σ is equal to a product of m transpositions from T, is called the *rank of σ with respect to T*, or the *T-rank of σ*, or just the rank of σ if T is understood. If we have

$$\sigma = t_k t_{k-1} \ldots t_2 t_1 \qquad (t_i \in T \text{ for } i = 1, \ldots, k)$$

we say that the right-hand side is a *T-factorization* of σ, of *length k*. If the length of a T-factorization of σ is the rank of σ, then we will say that the T-factorization is *minimal.*

Since T is a tree, given any two vertices x, y of T there is a unique path in T connecting x and y, denoted $[x \to y]$. If the vertices on this path are, in order from x to y: $x = a_1, a_2, \ldots, a_k = y$, we may also write this path as

$$[x \to y] = [a_1, a_2, \ldots, a_k].$$

The number of edges in $[x \to y]$ is called the *T-distance* from x to y, or just the distance from x to y if T is understood. If $\sigma \in S_n$, and if $i \in \{1, \ldots, n\}$, the *σ-path of i* is $P(i, \sigma) = [i \to \sigma(i)]$, and the length of this path is denoted by $L(i, \sigma)$. The sum of all the lengths of these paths is denoted by $\mathrm{PL}(\sigma)$, and called the *path-length* of σ:

$$\mathrm{PL}(\sigma) = \sum_{i=1}^{n} L(i, \sigma).$$

The minimal disjoint connected components of T generated by the σ-paths, are called the *σ-components of T*, and the subtree of T spanned by the union of all these components is called the *span of σ*, denoted $\mathrm{Sp}(\sigma)$.

Lemma 2.1. *Let $\sigma \in S_n$. If $(a, b) \in T$, then the number of σ-paths containing the edge (a, b) is even. Furthermore, $PL(\sigma)$ is even.*

Proof: If the edge (a, b) is removed from T, the resulting configuration consists of two disjoint trees X and Y. Define the sets A and B by:

$$A = \{x \in X \mid \sigma(x) \in Y\}, B = \{y \in Y \mid \sigma(y) \in X\}.$$

Since σ is a permutation, these sets must have the same cardinality. Clearly, the edge $[a, b]$ is contained in the σ-path $P(i, \sigma)$ if and only if $i \in A$ or $i \in B$, so the number $N(a, b)$ of σ-paths containing $[a, b]$ is $|A \cup B|$, which is an even number. Since $\mathrm{PL}(\sigma)$ is the total number of edges contained in the σ-paths (an edge is counted once for each path containing it) it is clear that

$$\mathrm{PL}(\sigma) = \sum_{(a,b) \in T} N(a, b)$$

and so $\mathrm{PL}(\sigma)$ must be an even number also. □

Lemma 2.2. *Let $\sigma \in S_n$ and $t \in T$. Define $\tau = \sigma t$. Then $PL(\sigma) - PL(\tau)$ is either* 2, 0 *or* -2.

Proof: Write $t = (a, b)$. Then if $i \neq a, b$, $P(i, \sigma) = P(i, \tau)$. We consider three cases.

Case 1: $P(a, \sigma) = [a, b \ldots, \sigma(a)]$ and $P(b, \sigma) = [b, a, \ldots, \sigma(b)]$. Then since $\tau(a) = \sigma(b)$, and $\tau(b) = \sigma(a)$, we have $P(b, \tau) = [b, \ldots, \sigma(a)]$ and $P(a, tau) = [a, \ldots, \sigma(b)]$, and then $\mathrm{PL}(\sigma) - \mathrm{PL}(\tau) = 2$.

Case 2: The edge $[a,b]$ is contained in one of $P(a,\sigma), P(b,\sigma)$, and is not contained in the other; say $P(a,\sigma) = [a,b,\ldots,\sigma(a)]$ and $P(b,\sigma) = [b,c,\ldots,\sigma(b)]$ where $c \neq a$. Then

$$P(a,\tau) = [a,b,c,\ldots,\sigma(b)] \text{ and } P(b,\tau) = [b,\ldots,\sigma(a)]$$

and $\mathrm{PL}(\sigma) - \mathrm{PL}(\tau) = 0$.

Case 3: $P(a,\sigma) = [a,x,\ldots,\sigma(a)]$ and $P(b,\sigma) = [b,y,\ldots,\sigma(b)]$, and (a,b) is not contained in either $P(a,\sigma)$ or in $P(b,\sigma)$. Then we have

$$P(a,\tau) = [a,b,y,\ldots,\sigma(b)] \text{ and } P(b,\sigma) = [b,a,x,\ldots,\sigma(a)],$$

and $\mathrm{PL}(\sigma) - \mathrm{PL}(\tau) = -2$.

These three cases are mutually exclusive and exhaustive. □

Corollary 2.3. *If $\sigma \in S_n$, then rank $\sigma \geq PL(\sigma)/2$.*

3. The algorithm

In this section we describe an algorithm, VPA, that results in a T-factorization of any given permutation σ. The underlying idea is that of a "greedy algorithm"; we attempt to reduce the total path-length as much as possible. In its basic form, VPA allows a certain amount of choice of what to do at any stage. We have some conjectures about certain restrictions on the choice of steps which might produce shorter T-factorizations. We describe these briefly at the end of this section, and more fully in Section 6.

VPA consists of a finite sequence of "steps", of three different types: each step adds to the result of the preceding step by one or two specially chosen transpositions of T. The algorithm terminates when the result of some step is the identity permutation, which we denote by e. Upon termination, we have an equation of the form $\sigma = \beta\alpha$, where α and β are products of T-transpositions, and this gives the corresponding T-factorization of σ.

We first describe the algorithm, and then prove the results necessary to show that it works: i.e., at any stage, at least one of the steps is possible, and the algorithm must always terminate in the identity, denoted by e.

Definition 3.1: Let (x,y) be an edge of T, and suppose $P(x,\sigma) = [x,y, z\ldots w]$ and $P(y,\sigma) = [y,x,t\ldots r]$. Then (x,y) is an *A-transposition for σ*. Note that if (x,y) is an A-transposition, then $\sigma(xy) = \tau$ satisfies $P(y,\tau) = [y,z\ldots w]$ and $P(x,\tau) = [x,\tau\ldots,r]$. Also, if u is not x or y, then $P(u,\tau) = P(u,\sigma)$.

Definition 3.2: Let (x,y) be an edge of T, and suppose $P(x,\sigma) = [x,\ldots z, t,w]$ and $P(y,\sigma) = [y,\ldots,r,w,t]$. Then (t,w) is a *B-transposition for σ*. Note that if (t,w) is a B-transposition, then $(t,w)\sigma = \tau$ satisfies

$P(x,\tau) = [x,\ldots,z,t]$ and $P(y,\tau) = [y,\ldots,r,w]$. Also, if u is not x or y, then $P(u,\tau) = P(u,\sigma)$.

Definition 3.3: Let (x,y) be an edge of T, and suppose the following hold:

- x, y are adjacent vertices of T,
- $\sigma(y) = y$
- $\sigma(z) = x \neq z$
- $P(x,\sigma) = [x,y,\ldots,w]$
- $P(z,\sigma) = [z,\ldots,y,x]$

Then (x,y) is a *C-transposition for* σ. Note that $\tau = (x,y)\sigma(x,y)$ satisfies the following: $\tau(x) = x$, $P(y,\tau) = [y,\ldots,w]$, and $P(z,\tau) = [z,\ldots,y]$. Also, if u is not x or y, then $P(u,\tau) = P(u,\sigma)$.

Lemma 3.4 is an obvious consequence of the above definitions.

Lemma 3.4. *Let $\sigma \in S_n$, where $\sigma \neq e$.*

- *If t is an A-transposition for σ, then $PL(\sigma t) = PL(\sigma) - 2$.*
- *If t is a B-transposition for σ, then $PL(t\sigma) = PL(\sigma) - 2$.*
- *If t is a C-transposition for σ, then $PL(t\sigma t) = PL(\sigma) - 2$.*

Algorithm 3.5. (VPA): Given $\sigma \in S_n$. We define sequences of permutations in S_n, $\{\sigma_i\}$, $\{\alpha_i\}$, $\{\beta_i\}$ as follows: $\sigma_0 = \sigma$, $\alpha_0 = e$, $\beta_0 = e$. If, for some value of i, $\sigma_i = e$, the algorithm terminates; otherwise perform the first of the following three steps that applies:

A-**step:** If there exists an A-transposition (x,y) for σ_i, put

$$\sigma_{i+1} = \sigma_i(x,y), \alpha_{i+1} = (x,y)\alpha_i, \beta_{i+1} = \beta_i.$$

B-**Step:** If there is no A-transposition for σ_i, and there exists some B-transposition (x,y) for σ_i, then put

$$\sigma_{i+1}\sigma_i = \sigma_i(x,y), \alpha_{i+1} = \alpha_i, \beta_{i+1} = \beta_i(x,y).$$

C-**Step:** If there is no A-transposition for σ_i, and no B-transposition (x,y) for σ_i, then there must be a C-transposition (x,y) for σ_i (we prove this later) and then put

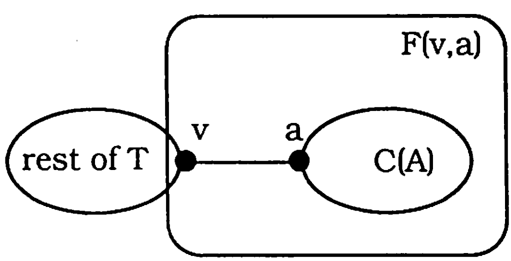
$$\sigma_{i+1}\sigma_i = (x,y)\sigma_i(x,y), \alpha_{i+1} = (x,y)\alpha_i, \beta_{i+1} = \beta_i(x,y).$$

Note that at the ith step, we have $\sigma = \beta_i \sigma_i \alpha_i$, and if the algorithm terminates at the mth step, then $\sigma = \beta_m \alpha_m$ is a T-factorization of σ, which will be called a VPA-factorization of σ.

It should be observed that an A-transposition for σ is a B-transposition for σ^{-1} and vice versa; and that a C-transposition for σ is also a C-transposition for σ^{-1}, and vice versa. Thus, if $\sigma = t_m \ldots t_1$ is a VPA factorization, then $\sigma^{-1} = t_1 \ldots t_m$ is also a VPA factorization. Several of the following proofs make use of this duality. If the rank of σ is 1, 2, or 3, it is easy to check that the VPA-factorization is minimal. If σ has rank 1, then σ "is" an A-step for σ; if s has rank 2, there are two A-steps. If σ has rank 3 an examination of the small number of possibilities for $\mathrm{Sp}(\sigma)$ yields the desired result. For example, if σ is a 4-cycle of rank 3, $\mathrm{Sp}(\sigma)$ is either a path on 4 vertices or a star on 4 vertices, and σ will be minimally factored with 3 A-steps.

We must first prove that for any $\sigma \neq e$, there is always an A-step or a B-step or a C-step. We begin with the following definition.

Definition 3.6: Let v be a vertex of T and a be any vertex adjacent to v. Removal of the edge (a, v) disconnects T into two components, one containing a (denoted $C(a)$), and one containing v. Let $F(v, a)$ be the subtree of T consisting of $C(a)$ together with v and the edge (a, v). Then $F(v, a)$ is called the *fan* of T determined by v and a. If every path in $F(v, a)$ from v to an outer vertex has length $\leq k$, where k is some positive integer, then we say that $F(v, a)$ is a *k-fan*.



Theorem 3.7. *Let $\sigma \in S_n$, and suppose that if x is an outer vertex of T then $\sigma(x) \neq x$. Now suppose that there are no possible A- or B- or C-steps for σ. Then for every positive integer k, every k-fan $F(v, a)$ in T has the following properties:*

(i) if $x \in F(v, a)$, then $\sigma(x) \neq v$

(ii) $\sigma(v) \notin F(v, a)$.

Proof: By induction on k. When $k = 1$, a 1-fan $F(v, a)$ is an interval of length 1; $F(v, a) = [a, v]$, where a is an outer vertex of T. Since $\sigma(a) \neq a$, then $[a, v]$ is contained in $P(a, \sigma)$. Since there are no A-transpositions, then $[v, a]$ cannot be contained in $P(v, \sigma)$, that is, $\sigma(v) \neq a$. Suppose that $\sigma(z) = a$; then $P(z, \sigma) = [z, \ldots, v, a]$ and has length at least 2. Since there are no B-transpositions, then $P(a, \sigma)$ cannot be $[a, v]$, that is, $\sigma(a) \neq v$. Then we have $P(a, \sigma) = [a, v, \ldots, x]$ and $P(z, \sigma) = [z, \ldots, v, a]$ (where $x, z \neq v$). Since there are no C-transpositions, then it must be the case that $\sigma(v) \neq v$. Thus, (i) and (ii) are true when $k = 1$.

Now fix $k > 1$, and suppose that (i) and (ii) are true for all j-fans, where $j < k$. Let $F(v, a)$ be a k-fan which is not a $(k-1)$-fan. Then there exists some $x \in F(v, a)$ such that x is adjacent to a and $x \neq v$, and for this x, $F(a, x)$ is a j-fan for some $j < k$. For each such x, since (i) and (ii) are true for $F(a, x)$, then $\sigma(a)$ is not in $F(a, x)$, and if $y \in F(a, x)$, then $\sigma(y) \neq a$; and if $\sigma(z) = a$, then z is not in $F(a, x)$. It follows that $P(a) = [a, v, \ldots, \sigma(a)]$ and $P(z) = [z, \ldots, v, a]$ $(z \neq a)$.

Since there are no A-transpositions, then $P(v)$ cannot contain the edge (a, v). Since every path between v and an element of $F(a, x)$ must contain the edge (a, v), then $P(v) = [v, \ldots, \sigma(v)]$ cannot contain elements of $F(a, x)$, and in particular, $\sigma(v)$ is not in $F(a, x)$, and $z \neq v$. This holds for every $x \in F(v, a)$ which is adjacent to a and with $x \neq v$, and so $\sigma(v)$ is not in $F(v, a) - \{v\}$.

Similarly, since there are no B-transpositions, then $\sigma(y) \neq v$ for any y in $F(v, a) - \{v\}$. Finally, comparing $P(a)$ and $P(z)$ above, it must be the case that $\sigma(v) \neq v$, since there are no C-transpositions. Thus $\sigma(v)$, $\sigma^{-1}(v)$ are not in $F(v, a)$. Now the conditions (i) and (ii) are true for $F(v, a)$. This completes the induction. □

Corollary 3.8. *Let $\sigma \in S_n$, and suppose that $\sigma \neq e$. Then σ has either an A-transposition, a B-transposition, or a C-transposition.*

Proof: Let T^* be the subtree of T spanned by the set $\{x \mid \sigma(x) \neq x\}$. Then T^* satisfies the hypothesis of Theorem 3.7. Since T^* has finite diameter, we can pick any outer vertex v, and regard T^* as a k-fan $F(v, a)$ where a is the unique vertex adjacent to v, and k is some positive integer not exceeding the diameter of T^*. If σ had no A- or B- or C-transpositions, then by Theorem 3.7, $\sigma(v)$ would not be in T^*, contradicting the fact that T^* contains all the vertices x such that $\sigma(x) \neq x$. The result follows. □

Theorem 3.9. *Let $\sigma \in S_n$, and suppose that σ has no fixed points. Then σ has an A-transposition.*

Proof: Let x_0 be an outer vertex of T and define inductively a path in T, $S = [x_0, x_1, x_2, \ldots]$, by: $P(x_0, \sigma) = [x_0, x_1, \ldots]$, $P(x_1, \sigma) = [x_1, x_2, \ldots]$, and in general, $P(x_i, \sigma) = [x_i, x_{i+1}, \ldots]$. Since σ has no fixed points, then $x_i \neq x_{i+1}$, for $i = 0, 1, \ldots$. Since S is a path in T, then any two successive members of S are adjacent in T. Finally, since T is finite, there must be some least index i such that $x_{i+1} = x_k$, where $0 \leq k \leq i$, and from the preceding observations x_{i+1} can only be x_{i-1}. That is, we have $P(x_{i-1}, \sigma) = [x_{i-1}, x_i, \dots]$ and $P(x_i, \sigma) = [x_i, x_{i-1}, \dots]$ Then (x_i, x_{i-1}) is an A-transposition for σ. □

Corollary 3.10. *Let $\sigma \in S_n$, and suppose that some σ-component of T contains no fixed points of σ. Then σ has an A-transposition.*

Theorem 3.11. *Let $\sigma \in S_n$, where $\sigma \neq e$. (a) Then Algorithm 3.5 applied to σ, terminates in finitely many steps. (b) Let r, s, t (respectively) be the numbers of A- , B- , C-steps used in the algorithm. Then the number of transpositions in the resulting T-factorization of σ is given by $m = r+s+2t$, and we have*

$$\mathrm{PL}(\sigma)/2 \leq m \leq \mathrm{PL}(\sigma) - 1.$$

Proof: (a) The identity, e, is the only permutation with path-length 0. By Lemma 3.4, each step of the algorithm reduces path-length by 2, and when path-length has been reduced to 0, the algorithm terminates. (b) An A- or B- step contributes one transposition to the resulting T-factorization of σ, while a C-step contributes two transpositions. So the total number is $m = r + s + 2t$. Since each step reduces path-length by 2, we have $r + s + t = \mathrm{PL}(\sigma)/2$, and so $m \leq \mathrm{PL}(\sigma)/2$. Consider the next-to-last step of the algorithm, σ_{m-1}. Since σ_{m-1} must have path-length 2, the only possibility is that $\sigma_{m-1} = (x, y)$ where (x, y) is an edge of T. Then the last step of the algorithm must be an A-step, since (x, y) is an A-transposition for σ_{m-1}. Then $r \geq 1$, and it follows that

$$m = r + s + 2t \leq 2(r + s + t) - 1 = \mathrm{PL}(\sigma) - 1.$$

This completes the proof. □

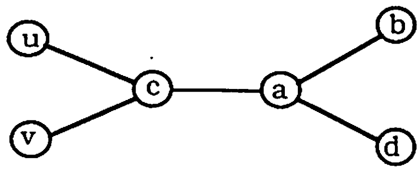
Corollary 3.12. *Let $\sigma \in S_n$. Then $PL(\sigma)/2 \leq$ rank $\sigma \leq PL(\sigma) - 1$.*

We are interested in reducing the path-length as quickly as possible (to produce a T-factorization of σ which is reasonably close to minimal). That is, the sequence of steps used should ideally have as many A-steps and B-steps as possible. It is easy to find examples (see below) to show, that different choices of C-steps can lead to differences in the total number of A-steps and B-steps used.

Example 3.13: Let $T = \{(u, c), (v, c), (c, a), (a, d), (a, b)\}$ and let σ be the permutation (with $\mathrm{PL}(\sigma) = 12$):

$$\sigma=\begin{pmatrix} a & b & c & d & u & v \\ a & v & d & c & b & u \end{pmatrix}$$

T:



Then σ has no A-transpositions or B-transpositions, and it has two C-transpositions: (a,b) and (a,d). If we use (a,b) first, we get the factorization

$$\sigma = (a,b)(a,c)(a,d)(c,u)(c,v)(a,c)(a,b)$$

of length 7 (the T-rank of σ). However, if we use (a,d) first, we get

$$\sigma = (a,d)(v,c)(a,b)(a,c)(a,b)(u,c)(v,c)(a,c)(a,d)$$

of length 9. Using (a,b) first, then all the remaining "steps" are A-steps; using (a,d) first, this is not the case.

In general, in seeking to maximize the total number of A- and B-steps used, our examples suggest that it is a good idea, when choosing C-steps, (i) to choose a C-step (a,b) (with $\sigma(b)=b$) such that (if possible) $\sigma(a)$ is an outer vertex of T and (ii) it is also a good idea to try to reduce path-length in a "balanced" way, i.e. choosing always to reduce the length of some longest path among all the paths of s which are affected by C-steps. We discuss some other possibilities in Section 6.

Interestingly enough, no special arrangements are necessary in the case when the tree T is a path or a star. □

Theorem 3.14. *VPA produces minimal factorizations for the cases where T is a path or a star.*

Proof: For the case of the path, it is easy to prove that an A or B step reduces the inversion number by one, and a C-step reduces it by two. The result follows from the fact that the inversion number of σ is the rank of σ. If T is a star, the steps of the VPA are (up to rearrangement of commuting factors) precisely the steps of the algorithm given in [3], which gives a minimal factorization. □

4. Trajectories and minimality

In this section, T and σ are fixed. We first prove a theorem and corollary which give necessary conditions for a T-factorization of σ to be minimal.

Definition 4.1: Given a factorization $\sigma = t_m t_{m-1} \dots t_2 t_1$ and x a vertex of T, we define the *trajectory of x determined by this factorization* to be the (ordered) sequence $R(x, \sigma) = \{x = x_0, x_1, x_2, \dots, x_m\}$ where $x_i = (t_i t_{i-1} \dots t_2 t_1)(x)$ for $i = 1, 2, \dots, m$. We may refer to these as "trajectories of σ" when the factorization is understood. Two trajectories are *disjoint* if they are disjoint as sets. For a fixed T-factorization, it is clear that in a trajectory $R(x, \sigma)$, for each $i = 1, 2, \dots, m$, x_i and x_{i-1} are either equal or adjacent, and that for each i, precisely two of the trajectories have unequal entries in the $i, i+1$ positions. The trajectory $R(x, \sigma)$ is said to be *monotone* provided that for all $t = 1, 2, \dots, m$ we have:

(i) $x_t \in R(x, \sigma) \Rightarrow x_t \in P(x, \sigma) = [x = a_0, a_1, a_2, \dots, a_k = \sigma(x)]$

(ii) If $x_t = a_r$ then either $x_{t+1} = a_r$ or $x_{t+1} = a_{r+1}$.

(A monotone trajectory for x traces out the path of x, $P(x, \sigma)$, in order from x to $\sigma(x)$, without either "going off" the path, or "backing up" on the path.)

We can write $t_i = (u_i, v_i)$ (where $(u_i, v_i) \in T$). If x and y are vertices of T with the following property:

$$x_{i-1} = t_{i-1} \dots t_1(x) = u_i \text{ and } y_{i-1} = t_{i-1} \dots t_1(y) = v_i,$$

then we will say that *x and y cross at t_i*.

Lemma 4.2. *Given a T-factorization $\sigma = t_m t_{m-1} \dots t_2 t_1$. If x and y cross at t_i, then*

$$\sigma = t_m \dots t_1 = t_m \dots t_{i+1} t_{i-1} \dots t_1 (x, y) = (\sigma(x), \sigma(y)) t_m \dots t_{i+1} t_{i-1} \dots t_1$$

Proof: For any permutation a and transposition (r, σ), it is true that $a(r, \sigma) = (a(r), a(\sigma))a$. The result follows from this. □

In the above lemma, $[x, y]$ need not be an edge of T. If it is, of course, then the lemma gives an alternative T-factorization of σ.

Theorem 4.3. *Let $\sigma = t_m \dots t_1$. Suppose that $\{1 \le i_1, i_2, \dots, i_k \le m\}$ is an increasing sequence of integers. (We will use the notation $t(i) = t_i$.) Let $\sigma(i_1, i_2, \dots, i_k)$ denote the result of removing the transpositions $t(i_1), t(i_2), \dots, t(i_k)$ from the factorization $t_m \dots t_1$. For each $i = i_1, i_2, \dots, i_k$ write $t_i = (u_i, v_i)$, and let x_i, y_i, a_i, b_i be defined by*

$$u_i = t_{i-1} t_{i-2} \dots t_1(x_i)$$
$$v_i = t_{i-1} t_{i-2} \dots t_1(y_i)$$
$$a_i = \sigma(x_i), b_i = \sigma(y_i).$$

Then

$$\sigma = t_m \dots t_1 = \sigma(i_1, i_2, \dots, i_k)(x_1, y_1)(x_2, y_2) \dots (x_k, y_k)$$
$$= (a_1, b_1)(a_2, b_2) \dots (a_k, b_k)\sigma(i_1, i_2, \dots, i_k).$$

Proof: The proof is by induction on k. The case $k = 1$ is just Lemma 4.3; for simplicity we work out only the case $k = 2$. For each $i = 1, \dots, m$, define $t_i = t_i t_{i-1} \dots t_2 t_1$.

Put $i = i_1$ and $j = i_2$. Since $1 \le i < j \le m$, then $\tau_j = t_j t_{j-1} \dots t_{i+1}\tau_i$, and by Lemma 4.3, we have $\tau_j = t_j t_{j-1} \dots t_{i+1}\tau_i = t_{j-1} \dots t_{i+1}\tau_i(x_2, y_2)$. Since $\tau_i = t_i t_{i-1} \dots t_1 = t_{i-1} \dots t_1(x_1, y_1)$, it follows that $\sigma = \sigma(i, j)(x_1, y_1)\cdot(x_2, y_2)$. Since $a_2 = \sigma(x_2)$, $b_2 = \sigma(y_2)$, and $a_1 = \sigma(x_1)$, $b_1 = \sigma(y_1)$, then we have $(a_2, b_2)\sigma(x_2, y_2) = \sigma = (a_1, b_1)\sigma(x_1, y_1)$ so that $(a_2, b_2)(a_1, b_1)\sigma = \sigma(x_2, y_2)(x_1, y_1) = \sigma(i, j)$, and $(a_1, b_1)(a_2, b_2)\sigma(i, j) = \sigma$ as required. □

Obviously, if the product (or any subproduct of) $(x_1, y_1)(x_2, y_2) \dots (x_k, y_k)$ is the identity then the initial factorization of σ is not minimal. If any of the (x_i, y_i) (or (a_i, b_i)) is an edge of T, then one can get a different T-factorization of σ. Unfortunately, a non-minimal factorization of σ need not have any product of this type equal to the identity, and need not allow any rearrangement along these lines. The example in Section 3 is such a one. Nevertheless, this theorem supplies a reasonable necessary criterion for minimality, which we shall see is satisfied by the factorizations produced by the VPA.

Corollary 4.4. *Suppose that $\sigma = t_m \dots t_1$ is a minimal T-factorization. If x, y are two distinct vertices of T, then x and y cross at most once, that is, if x and y cross at some t_i $(1 \le i \le m)$, then x and y do not cross at t_j for any $j \ne i$.*

Lemma 4.5. *Suppose that x is an outer vertex of T, adjacent to y, and that $\sigma(x) = x$. If $\sigma = t_m \dots t_1$ is a VPA factorization, then $t_i(x) = x$ for all $i = 1, 2, \dots, m$.*

Proof: The result is certainly true if $m = 1$, and then it follows easily by induction for all $m \ge 1$. □

In the next lemma, recall the notation of the VPA: at the ith step of the VPA, we have permutations σ_i, α_i and β_i such that $\sigma = \beta_i \sigma_i \alpha_i$. The permutations α_i and β_i are products of the transpositions associated with the various A, B, and C-steps employed up to that point.

Lemma 4.6. *Let $i > 1$, and suppose that the first i steps of a VPA for σ are all A or B-steps. Then (a) for every x in T, the trajectories $R(x, \alpha_i)$ and $R(x, \beta_i)$ are monotone, and (b) the α_i components of T, and the β_i components of T, do not contain any fixed points of σ.*

Proof: Statement (a) follows by induction from the definitions of A and B-steps. To see (b), suppose that $\sigma(c) = c$, and that, for example, the first step is an A-step. We have $\sigma = t_m \dots t_1$, where $t_1 = (u, v)$, and neither u nor v is fixed by σ (from the definition of an A-step). Then $t_1(c) = c$, and $\sigma_1(c) = c$; then if t is an A or B-transposition for σ_1 we must have $t(c) = c$ also. Proceeding inductively, none of the first i steps of the VPA can move c, that is, if $x \neq c$, the trajectories $R(x, \alpha_i)$ and $R(x, \beta_i)$ do not contain c. Since all the trajectories $R(x, \alpha_i)$ are monotone, then each of the α_i-components of T consists precisely of a union of all the vertices contained in some collection of the trajectories $R(x, \alpha_i)$; similarly for β_i. Since none of these trajectories contains any fixed points of σ, then (b) follows. □

Corollary 4.7. *Let $\sigma(x) \in x$, and suppose that $P(x, \sigma)$ contains a fixed point of σ. Then every VPA factorization of σ requires at least one C-step.*

Theorem 4.8. *Let $\sigma \in S_n$. The following are equivalent:*

1. *m = rank σ =$PL(\sigma)/2$*
2. *σ has a VPA factorization with no C-steps*
3. *σ has a VPA factorization with no C-steps or B-steps*
4. *For each $i = 0, 1, \dots, m-1$, if $\sigma_i(x) \neq x$, then $P(x, \sigma_i)$ contains no fixed points of σ_i.*

Proof: The implications (1) $\Leftrightarrow$ (2) and (3) $\Leftrightarrow$ (1) follow from Theorem 3.11, and (4) $\Leftrightarrow$ (3) follows from Theorem 3.9; while (3) $\Leftrightarrow$ (4) and (2) $\Leftrightarrow$ (4) come from Corollary 4.7. □

5. Conditions satisfied by the VPA

In this section, we first show that the VPA has a property which seems to be desirable for minimality, that is, if two elements x and y of T have disjoint σ-paths, and if $\sigma = t_m \dots t_1$ is a VPA factorization, then the trajectories $R(x, \sigma)$ and $R(y, \sigma)$ are disjoint. (We have not been able to prove that a minimal factorization of σ must have this property, but it seems highly probable.) We then prove that any VPA factorization satisfies the necessary condition for minimality given in Corollary 4.4. We will show that if $\sigma = t_m \dots t_1$ is a VPA factorization, then two distinct elements of T do not cross more than once.

Definition 5.1: If X and Y are subsets of the set of vertices of T, then the distance between X and Y is

$$d(X, Y) = d_T(X, Y) = \min\{d|T(x, y) \mid x \in X, y \in Y\}.$$

We may regard a path or trajectory of σ as a set of vertices of T, and so it makes sense to speak of the distance between two σ-paths, for instance.

Theorem 5.2. *Suppose that $\sigma = t_m \dots t_1$ is a VPA factorization, and $x, y \in T$. Suppose that $d(P(x,\sigma), P(y,\sigma)) \geq 2$. Then the trajectories $R(x,\sigma)$ and $R(y,\sigma)$ are disjoint.*

Proof: The proof is by induction on m. The result is certainly true for all σ which have a VPA with no C-steps, by Theorem 4.8 , since in that case, the VPA would consist entirely of A-steps, and all the trajectories would be monotone. In particular, it is true for all σ of rank 1 or 2, which are precisely those with a VPA factorization of length 1 or 2.

Assume the result is true for all permutations τ with a VPA factorization of length less than m. Suppose first that $\tau_1 = (a, b)$, is an A-step for σ. Then $\sigma_1 = t_m \dots t_2$ is a VPA factorization of length $m-1$, $P(a,\sigma_1) = P(b,\sigma) - (a, b]$, $P(b,\sigma_1) = P(a,\sigma) - [a, b)$, and for all $x \in a, b$ we have $P(x,\sigma_1) = P(x,\sigma)$. The situation is similar for a B-step. For a C-step, two of the paths have a point removed and a new vertex is fixed, while the others remain the same. Thus, in all cases, the distance between two paths cannot decrease, that is, for all $x, y \in T$, we have

$$d(P(x,\sigma_1), P(y,\sigma_1)) \geq d(P(x,\sigma), P(y,\sigma))$$

Now assume that $d(P(x,\sigma), P(y,\sigma)) \geq 2$. Then $d(P(x,\sigma_1), P(y,\sigma_1)) \geq 2$ also, and by the induction assumption, since $d(P(x,\sigma_1), P(y,\sigma_1)) \geq 2$, then the trajectories $R(x,\sigma_1)$ and $R(y,\sigma_1)$ are disjoint. Since $d(P(x,\sigma), P(y,\sigma)) \geq 2$, then (x, y) cannot be an A or B or C-transposition for σ, and so $t_1 \neq (x, y)$, and the trajectories $R(x, t_1)$ and $R(y, t_1)$ are also disjoint. Then $R(x,\sigma)$ and $R(y,\sigma)$ are disjoint, as required. □

Theorem 5.3. *Suppose that $\sigma = t_m \dots t_1$ is a VPA factorization where $d(P(x,\sigma), P(y,\sigma)) = 1$. Then the trajectories $R(x,\sigma)$ and $R(y,\sigma)$ are disjoint.*

Proof: The proof is by induction on m. The result is easily checked for $m = 1$ or 2.

The assumption implies that $P(x,\sigma)$ and $P(y,\sigma)$ are disjoint, and so (x, y) cannot be an A or B or C-transposition for σ. Then for σ_1 (the result of the first step of the VPA) we have $d(P(x,\sigma_1), P(y,\sigma_1)) \geq 1$. If the equality holds, then $R(x,\sigma_1)$ and $R(y,\sigma_1)$ are disjoint by the induction assumption; otherwise they are disjoint by Theorem 5.2. Since the first A or B or C-step is not (x, y), then $t_1 \in (x, y)$, and as in the proof of Theorem 5.2, we have that $R(x,\sigma)$ and $R(y,\sigma)$ are disjoint. □

Lemma 5.4. *Suppose that $\sigma = t_m \dots t_1$ is a VPA factorization, and that $t_1 = (a, b)$ is an A-transposition for σ (respectively, that t_m is a B-*

transposition for σ). Then a and b cross at t_1 (respectively, t_m), and do not cross at t_i for any $i = 2, \ldots, m$ (respectively, $i = 1, \ldots, m-1$).

Proof: We have $\sigma_1 = t_m \ldots t_2$, and by the definition of an A-step,

$$d(P(a,\sigma_1), P(b,\sigma_1)) = 1$$

Thus $R(a,\sigma_1)$ and $R(b,\sigma_1)$ are disjoint. It follows that a and b do not cross at any t_i with $2 \leq i \leq m$. The proof is similar when t_m is a B-transposition for σ. □

Lemma 5.5. *Let $\sigma = t_m \ldots t_1$ be a VPA factorization, and suppose that $t_1 = t_m = (a,b)$ is a C-transposition for σ (i.e., $\sigma(a) = a$, and $\sigma(b) \neq b$, $\sigma(z) = b$, and $a \in P(b,\sigma)$, $a \in P(z,\sigma)$). Then a and b cross at t_1 and do not cross at t_i for any $i = 2, \ldots, m$, and a and z cross at t_m, and do not cross at t_i for any $i = 1, \ldots, m-1$.*

Proof: Since $t_1 = (a,b)$ is a C-transposition it is easily verified that a and b cross at t_1, and a and z cross at t_m. We need only show that a and b and a and z do not cross at any of $t_{m-1}, \ldots, t_2$.

Since the first step is a C-step, we have $\sigma_1 = t_{m-1} \ldots t_2$, and $P(b,\sigma_1) = \{b\}$, $P(a,\sigma_1) = P(b,\sigma) - (a,b]$, $P(z,\sigma_1) = P(z,\sigma) - (a,b]$. Thus

$$d(P(a,\sigma_1), P(b,\sigma_1)) = 1 \text{ and } d(P(z,\sigma_1), P(b,\sigma_1)) = 1.$$

We can then conclude by Theorem 5.3 that $R(a,\sigma_1)$ and $R(b,\sigma_1)$ are disjoint, and so a and b do not cross at any t_i with $2 \leq i \leq m$, as required, and also $R(z,\sigma_1)$ and $R(b,\sigma_1)$ are disjoint, and so b and z do not cross at any t_i with $2 \leq i < m$. Since $z \neq a$, and a and b do cross at t_1, then a and z do not cross at t_1, and so a and z do not cross at any t_i with $1 \leq i < m$. □

Theorem 5.6. *Let $\sigma = t_m \ldots t_1$ be a VPA factorization. If $x, y \in T$, and if x and y cross at t_i for some $i = 1, 2, \ldots, m$, then x and y do not cross at t_j for any $j \neq i$.*

Proof: The proof is by induction; the result is easily seen for $m = 1$ or 2. Suppose $m > 2$, and suppose that x and y cross at t_i, where i is minimal. If $i = 1$ or m, the result follows from Lemmas 5.4 and 5.5, so suppose $2 \leq i < m$. Then $t_1(x)$ and $t_1(y)$ cross at t_i in σ_1, and by the induction assumption, they do not cross at any other t_j in σ_1. Then x and y do not cross at any other t_j in σ_1. If σ has an A-step, then x and y do not cross at t_1, and $\sigma_1 = t_m \ldots t_2$, and the result follows; similarly if σ has a B-step. If σ has a C-step, then x and y do not cross at either t_1 or t_m (by Lemma 5.5), and $\sigma_1 = t_{m-1} \ldots t_2$, and the result follows. □

6. Heuristics

Each step of VPA allows a number of permissible steps. At this point, we have no theoretical basis for preferring one A (or B or C)-step over another. However we do know that the final result does depend on these choices. In an attempt to determine the effectiveness of VPA and to investigate its behavior under various decision criteria (heuristics), we have programmed six versions of the VPA, which we will call VPA_1, VPA_2, and so on. We have run these for all permutations in S_n, for $n = 1, 2, \ldots, 7$, for all non-isomorphic trees with n vertices. These trees are enumerated in the Table 1.

The rank of each permutation for each tree was computed separately by constructing the appropriate Cayley graph. Each heuristic was then applied to each permutation for each tree and the actual T-rank was compared to the length of the resulting T-factorization. Due to the fact that a given permutation is either even or odd, the difference between T-rank and the length of any T-factorization must be an even number. Let us denote the length of the T-factorization of σ by VPA_i as $\text{LF}_i(\sigma)$ and $\text{DF}_i(\sigma)$ as the difference $\text{LF}_i(\sigma)$-T-rank of σ.

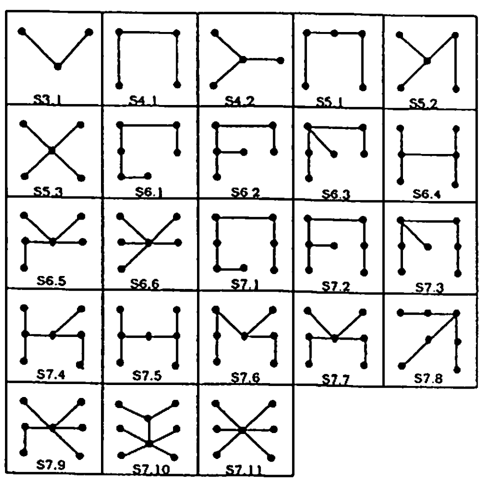


Table 1. Non-isomorphic trees with ≤ 7 vertices

The six heuristics are described below. In all cases, A-steps are preferred to B-steps and B-steps are chosen over C-steps. If there are several transpositions that satisfy the selection criteria for a heuristic, the one chosen is the first one encountered. The symbols x, y, z, used in a way consistent with Definitions 3.1, 3.2, 3.3

VPA_1	*A*-**step.** Select the first one encountered.
	B-**step.** Select the first one encountered.
	C-**step.** Select the first one encountered.
VPA_2	*A*-**step.** If possible, select (x, y) where either x or y will be removed from the span of the tree. Otherwise, select the first one encountered.
	B-**step.** Same as *A*-Step.
	C-**step.** If possible, select (x, y) where x will be removed from the span of the tree. Otherwise, select the first one encountered.
VPA_3	*A*-**step.** Select (x, y) so that either $L(x, \sigma)$ or $L(y, \sigma)$ is maximal
	B-**step.** Same as *A*-Step.
	C-**step.** Select (x, y) so that either $L(x, \sigma)$ or $L(z, \sigma)$ is maximal
VPA_4	*A*-**step.** Select (x, y) so that $L(x, \sigma) + L(y, \sigma)$ is maximal
	B-**step.** Same as *A*-Step.
	C-**step.** Select (x, y) so that either $L(x, \sigma) + L(z, \sigma)$ is maximal
VPA_5	*A*-**step.** Select (x, y) so that the distance from x or y to an outer vertex is maximal
	B-**step.** Same as *A*-Step.
	D-**step.** Select (x, y) so that the distance from x to an outer vertex is maximal
VPA_6	*A*-**step.** Select (x, y) so that the crossover number on $[x \to y]$ is maximal
	B-**step.** Same as *A*-Step.
	D-**step.** Same as *A*-Step.

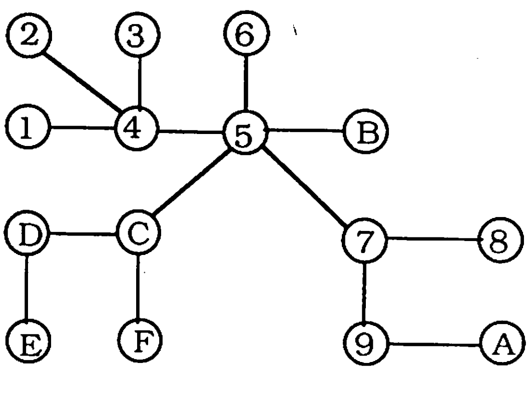
The crossover number of an edge $[x \to y]$ is the number of σ-paths, $[a \to \sigma(a)]$, in which $[x \to y]$ is found. The rationale for VPA_6 is to reduce the total crossover number as quickly as possible.

The trees S3.1, S4.1, S5.1, S5.3, S6.1, S6.6, S7.1, and S7.11 are either star graphs or paths and therefore all heuristics produced exact factorizations for all σ. In all cases studied and for all heuristics, $DF_i(\sigma) \leq 4$. In particular, $DF_3(\sigma) \leq 2$ and $DF_4(\sigma) \leq 2$ for all σ in all cases and these seemed to be the best overall heuristics for the cases studied. In addition, VPA_3 and VPA_4 gave exact factorizations for S5.2, thus being exact for all permutations of 5 letters. Table 2 shows the worst and best heuristics studied, VPA_1 and VPA_4 respectively. The entries in a row labeled *Sn.m* are the number of permutations in S_n for which the equation at the top of the column was satisfied. For example, of the 120 permutations generated by way of tree S5.2, 119 had exact factorizations using the heuristic VPA_1 (i.e., $DF_1(\sigma) = 0$).

Tree	$DF_1(\sigma)=0$	$DF_1(\sigma)=2$	$DF_1(\sigma)=4$	$DF_4(\sigma)=0$	$DF_4(\sigma)=2$
S5.2	119	1		120	
S6.2	702	18		714	6
S6.3	696	24		716	4
S6.4	686	34		720	
S6.5	706	14		720	
S7.2	4813	226	1	4938	102
S7.3	4739	297	4	4924	116
S7.4	4617	422	1	4941	99
S7.5	4526	514		4886	154
S7.6	4752	282	6	4946	94
S7.7	4732	306	2	4990	50
S7.8	4752	288		4949	91
S7.9	4890	150		5040	
S7.10	4638	402		5040	

Table 2: The best case and worst case results

As an additional example, we consider several permutations on $\{1, 2, 3, \ldots, 9, A, \ldots, F\}$ with T given below:



Since it was not possible to compute the actual rank of the all 15! permutations, 10 permutations were arbitrarily chosen and all 6 heuristics applied. A lower bound was computed using a theorem from [4], reproduced below without proof.

Theorem. *Let $\sigma = t_m \ldots t_1$ be a T-factorization. Let K be the set of all fixed points of σ such that for some $y \neq x$, we have $x \in P(y, \sigma)$. Let J be the set of all x such that $\sigma(x) \neq x$, and for some $y \neq x$, $P(x, \sigma) \in P(y, \sigma)$,* *and the two paths are in the same direction. Then*

$$m \geq |K| + |J| + PL(\sigma)/2.$$

Table 3 summarizes the results of this second experiment. If the parity of a permutation is not that of the lower bound, an adjustment can then be made. This is reflected in the computed lower bound (LB).

Image of σ	$\frac{PL}{2}$	$\|K\|$	$\|J\|$	LB	LF_1	LF_2	LF_3	LF_4	LF_5	LF_6
A5B6F1ED23C7984	25	0	1	26	34	30	28	30	34	34
B87AD2F1C4593E6	22	0	0	23	25	25	25	25	25	25
386B21C47AFD95E	18	0	1	19	23	21	23	23	23	23
AFDC3472B65E981	23	1	1	26	32	30	28	28	32	32
9BEFDC6817A2534	22	0	2	25	29	29	29	27	29	29
CDF1AB9E3428765	23	0	1	24	30	30	28	28	30	28
E8FB3C4D1A56972	22	0	1	24	28	30	28	28	28	30
7ACFED2649B3158	22	0	0	23	25	25	25	25	25	25
E512DAC3B469F87	22	0	1	23	29	27	29	27	29	29
9D8436FB1C5A7E2	20	1	0	22	24	24	24	24	24	26

Table 3: Results of the second experiment

Conclusion

In this paper we have described an algorithm that factors permutations from S_n into a product of transpositions. The set of allowable transpositions forms a tree on n vertices. The algorithm guarantees factorizations that are bounded above by PL(σ) − 1 and bounded below by PL(σ)/2. In addition, the algorithm produces minimal factorizations in the case of the path and the star. Necessary, but not sufficient, conditions for minimality are satisfied by the resulting factorizations. The paper concludes with a computer investigation of VPA under various decision criteria and small values of n.

References

[1] Berge, C, "Principles of Combinatorics", Academic Press (1959).

[2] Behzad, Chartrand, Lesniak-Foster, "Graphs and Digraphs", Prindle, Weber and Schmidt, 1979

[3] Edelman, P. H., *On Inversions and Cycles in Permutations*, European Journal of Combinatorics, 8 (1987).

[4] Knuth, D. E., "The Art of Computer Programming, Vol 3, Sorting and Searching", Addison-Wesley, 1973

[5] Portier, F. J., Vaughan, T. P., *Whitney Numbers of the Second Kind for the Star Poset*, European Journal of Combinatorics, 11 (1990).

[6] Riordan, J. "An Introduction to Combinatorial Analysis", John Wiley, New York and London, 1958.

[7] Vaughan, T. P. *Bounds for the Rank of a Permutation on a Tree*, The Journal of Combinatorial Mathematics and Combinatorial Computing, **10** (1991).

[8] Wielandt, H., "Finite Permutation Groups". Academic Press, New York, 1964.