Can a permutation be sorted by best short swaps?

Shu Zhang
Department of Computer Science and Technology, Shandong University
Jinan, China
zhangshu365@163.com

Daming Zhu
Department of Computer Science and Technology, Shandong University
Jinan, China
dmzhu@sdu.edu.cn

Haitao Jiang
Department of Computer Science and Technology, Shandong University
Jinan, China
htjiang@sdu.edu.cn

Jingjing Ma
Department of Computer Science and Technology, Shandong University
Jinan, China
majingjing.sdu@gmail.com

Jiong Guo
Department of Computer Science and Technology, Shandong University
Jinan, China

Haodi Feng
Department of Computer Science and Technology, Shandong University
Jinan, China

Abstract

A short swap switches two elements with at most one element caught between them. Sorting permutation by short swaps asks to find a shortest short swap sequence to transform a permutation into another. A short swap can eliminate at most three inversions. It is still open for whether a permutation can be sorted by short swaps each of which can eliminate three inversions. In this paper, we present a polynomial time algorithm to solve the problem, which can decide whether a permutation can be sorted by short swaps each of which can eliminate 3 inversions in $O(n)$ time, and if so, sort the permutation by such short swaps in $O(n^2)$ time, where n is the number of elements in the permutation.

A short swap can cause the total length of two element vectors to decrease by at most 4. We further propose an algorithm to recognize a permutation which can be sorted by short swaps each of which can cause the element vector length sum to decrease by 4 in $O(n)$ time, and if so, sort the permutation by such short swaps in $O(n^2)$ time. This improves upon the $O(n^2)$ algorithm proposed by Heath and Vergara to decide whether a permutation is so called lucky.

2012 ACM Subject Classification Mathematics of computing

Keywords and phrases Algorithm, Complexity, Short Swap, Permutation, Reversal

Digital Object Identifier 10.4230/LIPIcs.CPM.2018.14

Acknowledgements This paper is supported by national natural science foundation of China, No. 61472222, 61732009, 61761136017, 61672325.

29th Annual Symposium on Combinatorial Pattern Matching (CPM 2018).
Editors: Gonzalo Navarro, David Sankoff, and Binhai Zhu; Article No. 14; pp. 14:1–14:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A short swap on a permutation represents an operation which switches two elements with at most one element caught between them in the permutation. Sorting by short swaps asks to find a shortest sequence of short swaps which can transform a given permutation into another. This problem was first proposed by Heath and Vergara, who also proposed an approximation algorithm which can achieve a performance ratio 2 for this problem [9].

Short swap can be thought of as a kind of rearrangement operations on permutations, where a rearrangement has been being used to account for the gene order variations in a genome [3], and can be formalized as some basic operations such as reversal, translocation, and transposition [15]. Sorting permutation by rearrangements can be used to trace the evolutionary path between genomes [14], and plays important roles in computational biology and bioinformatics [13][8].

A short swap can be thought of as a two or three element consecutive subsequence reversal on a permutation [9]. Sorting a signed permutation by reversals was introduced by Bafna and Pevzner[1]. Hannenhalli and Pevzner proposed a polynomial time algorithm for this problem [8]. Other algorithmic progresses can be looked up in [11][6][7]. Sorting unsigned permutation by reversals turns to be NP-hard [4]. Thus people have been engaging in designing approximation algorithms for this problem [16][12][2].

Moreover, a short swap can be thought of as a swap of length 2 to 3 on a permutation. Jerrum has shown that minimum sorting by swaps can be solved in polynomial time [10]. The complexity of sorting by short swaps remains open up to now. Heath and Vergara proposed an upper bound $(\frac{n^2}{4})$ $+O(n \log n)$ for the minimum number of short swaps to sort an n-element permutation [9]. Feng *et. al.* improved the bound to $(\frac{3}{16})n^2 + O(n \log n)$ later [5].

In fact, the time complexity of deciding whether a permutation can be sorted by short swaps which eliminate three inversions, is still open. In this paper, we present a sufficient and necessary condition for a permutation to be sorted by short swaps which eliminate three inversions, based on which, we can propose an algorithm to recognize a permutation which can be sorted by short swaps which eliminate three inversions in $O(n)$ time, and if so, sort the permutation by short swaps to eliminate three inversions, in $O(n^2)$ time.

In the 2-approximation algorithm for sorting by short swaps [9], Heath and Vergara proposed to use an element vector to indicate how long a distance the element is from that element position it aims to be moved to, and showed that a short swap can cause two element vector's length sum to decrease by at most 4. Thus a so-called best cancellation refers to a short swap which can cause two element vector's length sum to decrease by 4. Heath and Vergara also presented an $O(n^2)$ algorithm to decide whether a permutation can be sorted by best cancellations. In this paper, we further propose a sufficient and necessary condition for a permutation to be sorted by best cancellations. Based on this observation, we propose an algorithm to recognize a permutation which can be sorted by best cancellations in $O(n)$ time, and if so, sort the permutation by best cancellations, in $O(n^2)$ time.

2 Preliminaries

Let $\pi = [\pi_1, \pi_2, ..., \pi_n]$ be a permutation of $\{1, 2, ..., n\}$. A *swap* on π switches π_i with π_j, where π_i and π_j are two elements in π. The swap is short, if there is at most one element between π_i and π_j in π. Let ρ be an arbitrary swap on π. We denote by $\pi \cdot \rho$ the permutation ρ transforms π into. For example, let ρ be a swap which switches 7 with 4 in $\pi = [5, 3, 1, 7, 6, 4, 2]$. Then $\pi \cdot \rho = [5, 3, 1, 4, 6, 7, 2]$. The problem of sorting a permutation by short swaps can be formulated as follows.

Instance: A permutation π
Solution: A sequence of short swaps ρ_1, ρ_2, ..., ρ_k, such that $\pi \cdot \rho_1 \cdot \rho_2, ..., \cdot \rho_k = [1, 2, ..., n]$ and k is minimized.

As usually used, let ι denote the identity permutation $[1, 2, ..., n]$. The minimum number of short swaps which transform π into ι is referred to as the short swap distance of π, and denoted by $sw^3(\pi)$.

2.1 Happy permutation

An *inversion* in π refers to a pair of elements that are not in their correct relative order. Formally, the pair composed of π_i and π_j is an inversion of π_i and π_j in π, if $i < j$ and $\pi_i > \pi_j$. Let inv_π be the set of inversions in π. A short swap ρ is said to *eliminate* $|inv_\pi| - |inv_{\pi\cdot\rho}|$ inversions (of π), if $|inv_\pi| \geq |inv_{\pi\cdot\rho}|$, and *add* $|inv_{\pi\cdot\rho}| - |inv_\pi|$ inversions (of π) otherwise.

A short swap can eliminate at most 3 inversions of π. If $\pi \neq \iota$, at least 1 inversion of two adjacent elements occurs in π, which can be eliminated by a short swap. Thus the short swap distance of π can be bounded by,

▶ **Lemma 1.** $\lceil \frac{|inv_\pi|}{3} \rceil \leq sw^3(\pi) \leq |inv_\pi|$

Proof. See Theorem 3 in [9]. ◀

Due to Lemma 1, a short swap is referred to as *best* (resp. *worst*), if it can eliminate (resp. add) 3 inversions of π. A permutation, say π is referred to as *happy*, if $sw^3(\pi) = \frac{|inv_\pi|}{3}$. A permutation is happy, if and only if it can be transformed into ι by none other than best short swaps.

A consecutive sub sequence $\pi[x \to y] \equiv [\pi_x, ..., \pi_y]$ of π is referred to as an *independent sub-permutation* (abbr. ISP) in π, if for $1 \leq l < x \leq i \leq y < h \leq n$, $\pi_l < \pi_i < \pi_h$. An ISP is referred to as *minimal*, if none of its sub sequence, other than itself, is an ISP. A minimal ISP in π is abbreviated as an MISP. Since no inversion happens between two distinct ISPs, it suffices to pay attention to sorting an MISP by best short swaps.

For an element π_i in π, we refer to the integer interval $[i, \pi_i]$ as the vector of π_i in π and denote it as $v_\pi(\pi_i)$, where $|v_\pi(\pi_i)| = |\pi_i - i|$ is referred to as the *length* of $v_\pi(\pi_i)$. The element vector length indicates the difference between the element index and its correct index. The element π_i is referred to as *vector-right*, if $\pi_i - i > 0$; *vector-left*, if $\pi_i - i < 0$; and *vector-zero*, if $\pi_i - i = 0$. An MISP is *isolated*, if it contains just one element. An isolated MISP must admit one and only one vector-zero element. Let $\pi[x \to y]$ be an arbitrary MISP. If $\pi[x \to y]$ is not isolated, then π_x must be vector-right, and π_y vector-left.

2.2 Lucky permutation

Let $V_\pi = \{v_\pi(\pi_i) \mid 1 \leq i \leq n\}$. We denote by $L(V_\pi)$ the length sum of all those vectors in V_π. A short swap always involves two element vectors. An element can be caused by one short swap to change its vector's length by at most 2. Thus a short swap can cause $L(V_\pi)$ to decrease by at most 4. If $\pi \neq \iota$, Heath *et. al.* have shown in [9] that it can always find two elements in π and a sequence of short swaps to switch them, such that if switching the two elements uses m short swaps which transform π into π', then $L(V_\pi) - L(V_{\pi'}) \geq 2m$. This leads to another short swap distance bound of π, which can be described as,

▶ **Lemma 2.** $\frac{L(V_\pi)}{4} \leq sw^3(\pi) \leq \frac{L(V_\pi)}{2}$

Proof. See Theorem 10 in [9]. ◀

A permutation π is referred to as *lucky*, if $sw^3(\pi) = \frac{L(V_\pi)}{4}$.

3 How to recognize a happy permutation

We denote by $\rho\langle i, j\rangle$ $(i < j)$ a swap on π, which switches π_i with π_j. If $\rho\langle i, j\rangle$ is short, then $i+1 \leq j \leq i+2$. The short swap $\rho\langle i,\, j\rangle$ *affects* an ISP in π, if at least one of π_i, π_{i+1}, π_j occurs in the ISP. The short swap $\rho\langle i,\, j\rangle$ *acts on* an ISP, if all of π_i, π_{i+1} π_j occur in the ISP. To check if a permutation is happy, we present a sufficient and necessary condition for a short swap to be worst. A best or worst short swap must switch two elements with another element caught between them. Thus $\rho\langle i,\, i+2\rangle$ will usually be used to represent a best or worst short swap.

▶ **Lemma 3.** *A short swap, say $\rho\langle i, i+2\rangle$ on π is worst, if and only if $\pi_i < \pi_{i+1} < \pi_{i+2}$.*

Let $\pi[x \to y]$ be an ISP in π. If a short swap $\rho\langle i, j\rangle$ which acts on $\pi[x \to y]$ transforms π into π', then $\pi'[x \to y]$ must be an ISP in π'.

▶ **Lemma 4.** *If a worst short swap acts on an MISP, it must transform the MISP into an ISP which remains an MISP.*

For an arbitrary ISP $\pi[x \to y]$ in π, an element π_j in $\pi[x \to y]$ is referred to as *position-odd*, if $j - x$ is zero or even; *position-even*, otherwise. An ISP is referred to as *sorted* if no inversion occurs in the ISP; *unsorted*, otherwise. An ISP $\pi[x \to y]$ in π is referred to as *happy*, if it can be transformed into $\iota[x \to y]$ by none other than best short swaps. By the following theorem, we present a sufficient and necessary condition for an MISP to be happy.

▶ **Theorem 5.** *An unsorted MISP is happy if and only if,* (1) *an element in the MISP is vector-zero if it is position-even; not vector-zero otherwise; and* (2) *for any two vector-left* (*resp. vector-right*) *elements, say π_i, π_j in the MISP, if $i > j$, then $\pi_i > \pi_j$.*

To prove Theorem 5, let's start with a couple of lemmas. Although in Theorem 5, those two properties are mentioned for an MISP to meet, it cannot refuse an ISP in π to meet those two properties. Thus an ISP is said to meet the Theorem-5 property (1), if all position-even elements are vector-zero, while all position-odd elements in the ISP are not; and said to meet the Theorem-5 property (2), if all those vector-left as well as vector-right elements increase monotonously. To show Theorem 5, we insist to show that a worst short swap can always transform a sorted ISP or an ISP which meets those two Theorem-5 properties into an ISP which meets those two Theorem-5 properties. This asks to observe on if a worst short swap acts on an ISP which meets those two Theorem-5 properties, and transform it into an MISP, whether this MISP meets those two Theorem-5 properties. No matter how many MISPs a short swap affects, we always treat those MISPs a short swap affects as an ISP.

▶ **Lemma 6.** *If a worst short swap acts on an ISP which meets those two Theorem-*5 *properties, it must transform the ISP into an ISP which meets those two Theorem-*5 *properties.*

If the ISP the worst short swap acts on is an MISP, Lemma 6 can be redescribed as:

▶ **Corollary 7.** *If a worst short swap acts on an MISP with those two Theorem-*5 *properties, it must transform the MISP into an MISP with those two Theorem-*5 *properties.*

▶ **Lemma 8.** *A short swap cannot be worst, if it affects just two MISPs each of which is isolated or meets those two Theorem-*5 *properties.*

▶ **Lemma 9.** *If a worst short swap affects three MISPs, each of which is isolated or meets those two Theorem-*5 *properties, it must transform the ISP which consists only these three MISPs into an MISP with those two Theorem-*5 *properties.*

Proof. Let $\rho\langle i,i+2\rangle$ be a worst short swap which affects three MISPs in π, each of which is isolated or meets those two Theorem-5 properties. Then $\pi_i < \pi_{i+1} < \pi_{i+2}$. That MISP caught between the other two MISPs in π must be isolated. Thus without loss of generality, let $\pi[x \to i]$, $[i+1]$ and $\pi[i+2 \to y]$ be those three MISPs $\rho\langle i,i+2\rangle$ affects. Let $\pi' = \pi \cdot \rho\langle i,i+2\rangle$.

Proof for $\pi'[x \to y]$ to be an MISP. Note that $\pi'_i = \pi_{i+2}$, $\pi'_{i+2} = \pi_i$ and $\pi'_j = \pi_j$ for $j \neq i$ and $j \neq i+2$. We show that if $\pi'[x_1 \to y_1]$ is an MISP with $x \leq x_1 \leq y_1 \leq y$, then $x = x_1$ and $y = y_1$.

Otherwise, let on one hand, $x \neq x_1$. (1) If $x < x_1 < i+1$, then in $\pi'[x \to x_1 - 1]$, an arbitrary element is less than an arbitrary element in $\pi'[x_1 \to y]$. Since $\pi[x \to x_1 - 1] = \pi'[x \to x_1 - 1]$, $\pi[x \to x_1 - 1]$ must be an ISP. The assumption for $\pi[x \to i]$ to be an MISP is contracted. (2) If $i+2 < x_1 \leq y$, it can follow (1) to show that $\pi[i+2 \to x_1 - 1]$ must be an ISP. The assumption for $\pi[i+2 \to y]$ to be an MISP is contracted. (3) If $x_1 = i+1$ or $x_1 = i+2$, then $\pi'[x_1 \to y_1]$ cannot be an MISP because $\pi'_i > \pi'_{i+1} > \pi'_{i+2}$. That is the proof for $x = x_1$. For the same reason, $y = y_1$.

Proof for $\pi'[x \to y]$ to meet those two Theorem-5 properties. Since $[i+1]$ is isolated, $\pi_{i+1} = i+1$, and for $x \leq l \leq i$ and $i+2 \leq h \leq y$, $\pi_l < \pi_{i+1} < \pi_h$.

(1) If $\pi[x \to i]$ and $\pi[i+2 \to y]$ are both isolated, then $i = x$ and $y = i+2$, and $\pi'[x \to y] = [i+2,\ i+1,\ i]$ meets those two Theorem-5 properties trivially.

(2) If one of $\pi[x \to i]$ and $\pi[i+2 \to y]$ is isolated, then $i = x$ and $y \neq i+2$ or $i \neq x$ and $y = i+2$. We only focus on the former subcase, where $i = x$ and $y \neq i+2$, to present the proof. In this subcase, $\pi_i = \pi'_{i+2} = i < i+2$, $\pi_{i+1} = \pi'_{i+1} = i+1$, which means π'_{i+1} is vector-zero and π'_{i+2} vector-left. Since $\pi[i+2 \to y]$ is not isolated, π'_i and π_{i+2} are vector-right. All position-odd (resp. position-even) elements in $\pi[i+2 \to y]$ remain position-odd and not vector-zero (resp. position-even and vector-zero) in $\pi'[x \to y]$. The proof for $\pi'[x \to y]$ to meet Theorem-5 property (1), is done.

The vector-zero element π_i in $\pi[x \to y]$ turns into the vector-left element π'_{i+2} in $\pi'[x \to y]$, and all elements in $\pi[i+2 \to y]$ turn into elements in $\pi'[x \to y]$ in the the same relative order as they are in $\pi[i+2 \to y]$. Thus to show that $\pi'[x \to y]$ meets Theorem-5, it suffices to show that π'_{i+2} is the leftmost vector-left element in $\pi'[x \to y]$, and less than any other vector-left element in $\pi'[x \to y]$. Of course this is true, because π'_i is vector-right, π'_{i+1} is vector-zero and $\pi'_{i+2} = \pi_i < \pi_{i+1} < \pi_h$ for $h > i+1$. The proof for $\pi'[x \to y]$ to meet Theorem-5 property (2), is done.

(3) If none of $\pi[x \to i]$ and $\pi[i+2 \to y]$ is isolated, then $i \neq x$ and $y \neq i+2$. By Lemma 6, to make sure for $\pi'[x \to y]$ to meet those two Theorem-5 properties, it suffices to show that $\pi[x \to y]$ meets those two Theorem-5 properties.

Since $\pi[x \to i]$ and $\pi[i+2 \to y]$ meet Theorem-5 property (2), and $\pi_l < \pi_{i+1} < \pi_h$ for $x \leq l \leq i$ and $i+2 \leq h \leq y$, $\pi[x \to y]$ meets the Theorem-5 property (2).

Since $\pi[x \to i]$ meets the Theorem-5 property (1), $i - x$ is even. Then, (1)the vector-zero element π_{i+1} is position-even in $\pi[x \to y]$; (2)each position-odd (resp. position-even) element in $\pi[x \to i]$ and $\pi[i+2 \to y]$, remains position-odd (resp. position-even) in $\pi[x \to y]$. This implies that $\pi[x \to y]$ meets the Theorem-5 property (1). ◀

The proof of Theorem 5 can be given by Corollary 7 and Lemma 8, 9.

Proof. Only if: Let $\pi[x \to y]$ be an unsorted and happy MISP, which can be transformed into $\iota[x \to y]$ by m best short swaps, say ρ_1, ρ_2, ..., ρ_m. Then $(\pi \cdot \rho_1 \cdot \rho_2 \dots \rho_{m-1} \cdot \rho_m)[x \to y] = \iota[x \to y]$. Let $\pi^k[x \to y] = (\iota \cdot \rho_m \cdot \rho_{m-1} \dots \rho_{m+2-k} \cdot \rho_{m+1-k})[x \to y]$ for

$1 \le k \le m$. Then $\pi^m[x \to y] = \pi[x \to y]$. By induction for k, we show every unsorted MISP in $\pi^k[x \to y]$ meets those two Theorem-5 properties.

(1) Without loss of generality, let $\rho_m = \rho\langle i, i+2\rangle (1 \le i \le n-2)$. Then $\rho\langle i, i+2\rangle$ must be a worst short swap which acts on ι. It follows that $\pi^1[x \to y] = (\iota \cdot \rho_m)[x \to y] = [x,\ x+1,\ ...,\ i-1,\ i+2,\ i+1,\ i,\ i+3,\ ...,\ y]$, where $[x]$, ..., $[i-1]$, $[i+3]$, ..., $[y]$ are isolated MISPs and $[i+2,\ i+1,\ i]$ is an unsorted MISP, which meets those two Theorem-5 properties trivially.

(2) By inductive assumption, let all unsorted MISPs in $\pi^{k-1}[x \to y]$ meet those two Theorem-5 properties. Assume again $\rho_{m+1-k} = \rho\langle i, i+2\rangle (x \le i \le y-2)$ with $\pi^k[x \to y] = (\pi^{k-1} \cdot \rho\ \langle i, i+2\rangle)[x \to y]$. Note that $\rho\langle i, i+2\rangle$ must be a worst short swap which acts on $\pi^{k-1}[x \to y]$. By Lemma 8, $\rho\langle i, i+2\rangle$ cannot affect two MISPs. By Corollary 7 and Lemma 9, all unsorted MISPs in $\pi^k[x \to y]$ must meet those two Theorem-5 properties.

If: Let $\pi[x \to y]$ be an MISP in π which meets those two Theorem-5 properties. The proof for $\pi[x \to y]$ to be happy, is to show that one can find a best short swap which can act on $\pi[x \to y]$ and transform it into an ISP in which each MISP either is isolated or meets those two Theorem-5 properties.

Identify a best short swap: Let π_i be the biggest element in $\pi[x \to y]$. Then $\rho\langle i, i+2\rangle$ can be shown to be a best short swap which acts on $\pi[x \to y]$. The proof can be stated as:

(1) Since $\pi[x \to y]$ meets those two Theorem-5 properties and π_i is the biggest in $\pi[x \to y]$, π_i must be vector-right and position-odd in $\pi[x \to y]$ and no vector-right element can occur on the right side of π_i, which implies π_{i+1} is position-even and equal to $i+1$.

(2) Then $\pi_i \ge i+2$ follows from that π_i is vector-right, $\pi_{i+2} \le i$ follows from that no vector-right element can occur on the right side of π_i. Thus $\pi_i > \pi_{i+1} > \pi_{i+2}$.

Let $\pi'[x \to y] = (\pi \cdot \rho\langle i, i+2\rangle)[x \to y]$. We devote to show that all unsorted MISPs in $\pi'[x \to y]$ must meet those two Theorem-5 properties.

The proof to meet the Theorem-5 property (2): Since $\pi_i \ge i+2$ is vector-right, $\pi_{i+2} \le i$ is vector-left, $\pi'_i = \pi_{i+2} \le i$ is either vector-zero or vector-left, $\pi'_{i+2} = \pi_i \ge i+2$ is either vector-zero or vector-right. This indicates that no vector-left (resp. vector-right) element in $\pi[x \to y]$ can turn into vector-right (resp. vector-left) in $\pi'[x \to y]$. Moreover, no two vector-left (resp. vector-right) elements in $\pi[x \to y]$ can occur in $\pi'[x \to y]$ in the other order than they are in $\pi[x \to y]$. It follows that all unsorted MISPs in $\pi'[x \to y]$ meet the Theorem-5 property (2).

The proof to meet the Theorem-5 property (1): All position-even elements in $\pi'[x \to y]$ are vector-zero because $\rho\langle i, i+2\rangle$ switches only π_i with π_{i+2}. The first element in an unsorted MISP in $\pi'[x \to y]$ must be vector-right, then must be position-odd in $\pi'[x \to y]$. Thus to make sure for all unsorted MISPs in $\pi'[x \to y]$ to meet the Theorem-5 property (1), it suffices to show that for all π'_j in $\pi'[x \to y]$, if π'_j is position-odd and vector-zero, then $[\pi'_j]$ is an isolated MISP. Since $\pi[x \to y]$ meets the Theorem-5 property (1), only π'_i and π'_{i+2} can be position-odd and vector-zero in $\pi'[x \to y]$.

If π'_{i+2} is vector-zero, $[\pi'_{i+2}]$ must be an isolated MISP, because π'_{i+2} is the biggest element in $\pi'[x \to y]$.

If π'_i is vector-zero, it must be the smallest in $\pi'[i \to y]$. The reason is, (1)since $\pi[x \to y]$ meets the Theorem-5 property (1) and $\pi_{i+2} = i$, an element in $\pi[i \to y]$ is bigger than $\pi_{i+2} = \pi'_i$, if it is position-even in $\pi[x \to y]$; (2)since $\pi[x \to y]$ meets the Theorem-5 property (2) and π_{i+2} is vector-left, an element in $\pi[i+3 \to y]$ is bigger than $\pi_{i+2} = \pi'_i$, if it is vector-left in $\pi[x \to y]$; (3)π_i is the unique vector-right element in $\pi[i \to y]$ and bigger than $\pi_{i+2} = \pi'_i$. It follows that $[\pi'_i]$ is an isolated MISP. ◀

Algorithm 1: How to recognize a happy permutation.

Algorithm *Happy permutation*
Input: A permutation π.
Output: The best short swap sequence ρ if π is happy; *no*, otherwise.
1 $lb \leftarrow 0$; $rb \leftarrow 0$; $x \leftarrow 1$; $b \leftarrow 0$;
2 For i from 1 to n do
3   if $(i > b)$ then $x \leftarrow i$; (an MISP starts with π_x)
4   if $(i - x \; mod \; 2 = 1$ and $\pi_i = i)$ then $i \leftarrow i + 1$; (π_i is position-even, vector-zero.)
5   if $(i - x \; mod \; 2 = 0$ and $\pi_i < i$ and $\pi_i > lb)$
6   then $lb \leftarrow \pi_i$; $i \leftarrow i + 1$; (π_i is position-odd, vector-left.)
7   if $(i - x \; mod \; 2 = 0$ and $\pi_i > i$ and $\pi_i > rb)$
8   then $rb \leftarrow \pi_i$; $i \leftarrow i + 1$; $b \leftarrow \pi_i$; (π_i is position-odd, vector-right.)
9   if $(i = x$ and $\pi_i = i)$ then $b \leftarrow \pi_i$, $i \leftarrow i + 1$; ($[\pi_i]$ is isolated.)
10   else return *no*;
11 end for
12 Return Sort(π);

In fact, an MISP in π can be recognized by,

▶ **Lemma 10.** *An MISP in π starts with π_i, if and only if $i = 1$ or for $1 \leq j \leq i - 1$, $i > \pi_j$.*

To decide if π is happy, it suffices to check if all MISPs in π, if unsorted, meet those two Theorem-5 properties.

An element in an MISP can be decided to be position-odd or position-even by the first element index of the MISP and its index. Then an MISP can be decided to meet the Theorem-5 property (1) by the value of $|\pi_i - i|$ for all π_i in this MISP.

An element in π can be decided to be vector-right, vector-left or vector-zero by the value of $\pi_i - i$. To check if all unsorted MISPs in π meet the Theorem-5 property (2), it suffices to check if π meets the Theorem-5 property (2). Fortunately, π can be decided to meet the Theorem-5 property (2) by checking if all those vector-left (resp. vector-right) elements increase monotonously in the order from π_1 to π_n.

We present an algorithm to recognize and sort a happy permutation π in Algorithm 1. If π is happy, the algorithm returns a best short swap sequence which can transform π into ι by invoking a subroutine named as Sort(π); returns *no*, otherwise. In the algorithm description, we use the integer parameter lb (resp. rb) to maintain the biggest vector-left (resp. vector-right) element in $\pi[1 \rightarrow i - 1]$, b the biggest element in $\pi[1 \rightarrow i - 1]$, x the starting index of the MISP in which π_i is an element.

Running the algorithm from Step 1 to Step 11 can decide if π is happy or not. This can take $O(n)$ time, where n is the number of elements in π. Later, let π be happy. We present on how to find a sequence of best short swaps to transform π into ι. To identify a best short swap which switches π_i with π_{i+2}, it suffices to record the integer i. Thus in Sort(π), we will employ a linear integer array $\rho[1 \sim X]$ to maintain the best short swap sequence to sort π, where $X \leq \frac{n(n-1)}{6}$, $\rho[j]$ indicates to switch $\pi_{\rho[j]}$ with $\pi_{\rho[j]+2}$.

The rightmost vector-right element in π must be the rightmost vector-right element in an MISP in π. Let π_i be the rightmost vector-right element in π. Then it follows the proof of the Theorem 5 sufficient condition that the short swap which switches π_i with π_{i+2} is best. By Theorem 5 again, this operation must transform π into a happy permutation. Thus the trick for finding the rightmost vector-right element in π to identify a best short swap can be done repeatedly until π is transformed into ι. The algorithm Sort(π) is depicted in Figure 2.

Algorithm 2: How to sort a happy permutation.

Algorithm *Sort*(π)
$x \leftarrow 0$;
while $\pi \neq \iota$
  find the rightmost vector-right element π_i;
  while $\pi_i > i$
    $\rho[x] \leftarrow i$; $\pi \leftarrow \pi \cdot \rho[x]$; $x \leftarrow x + 1$;
    $i \leftarrow i + 2$;
  end while
end while
Return ρ.

A rightmost vector-right element, say π_i, remains rightmost and vector-right in the permutation the short swap which switches π_i with π_{i+2} transforms π into, until it turns into vector-zero. So it takes $O(n)$ time to find all the rightmost vector-right elements. On the other hand, each best short swap can eliminate 3 inversions, the total inversion number is $O(n^2)$. Thus the time complexity of Sort(π) is $O(n^2)$. It follows that the time complexity of recognizing a happy permutation is $O(n^2)$.

4 How to recognize a lucky permutation

A short swap on π is referred to as a *best cancellation*, if it cause $L(V_\pi)$ to decrease by 4 [9]. The permutation π is referred to as *lucky*, if it can be transformed into ι by none other than best cancellations. A short swap is referred to as a *promising cancellation* (resp. *promising addition*), if it switches two adjacent elements in π and causes $L(V_\pi)$ to decrease (resp. increase) by 2.

An ISP $\pi[x \rightarrow y]$ is referred to as *sub-lucky*, if it can be transformed into $\iota[x \rightarrow y]$ by none other than promising cancellations. To check if a permutation is lucky, we set about to check if an ISP is sub-lucky. This asks us to observe what kind of a short swap is a promising addition or cancellation.

▶ **Lemma 11.** *The short swap $\rho\langle i, i+1\rangle$ on π is a promising addition, if and only if $\pi_i \leq i$ and $\pi_{i+1} \geq i+1$.*

Following Lemma 11, a promising cancellation can be identified by,

▶ **Corollary 12.** *The short swap $\rho\langle i, i+1\rangle$ on π is a promising cancellation, if and only if $\pi_i \geq i+1$ and $\pi_{i+1} \leq i$.*

By the following theorem, we state for what an MISP is sub-lucky.

▶ **Theorem 13.** *An unsorted MISP is sub-lucky if and only if,* (1) *all elements in the MISP are not vector-zero; and* (2) *for any two vector-left* (*resp. vector-right*) *elements, say π_i, π_j in the MISP, if $i > j$, then $\pi_i > \pi_j$.*

The second property of the theorem implies that those vector-left as well as vector-right elements increase monotonously. In fact, we can use the same way as used to show Theorem 5 to show the theorem. Although in Theorem 13, those two properties are mentioned for an MISP to meet, it cannot refuse an ISP in π to meet those two properties. Thus an ISP is said to meet the Theorem-13 property (1), if all elements in the ISP are not vector-zero; and said to meet the Theorem-13 property (2), if all those vector-left as well as vector-right elements increase monotonously. The following lemma, although seems trivial, deserves to be stated.

▶ **Lemma 14.** *If an ISP meets those two Theorem-*13 *properties, then all MISPs in the ISP meet those two Theorem-*13 *properties.*

To show Theorem 13, we show that an ISP, if meets those two Theorem-13 properties, cannot be transformed by a promising addition into one out of those two Theorem-13 properties. That is,

▶ **Lemma 15.** *If a promising addition acts on an ISP which meets those two Theorem-*13 *properties, it must transform the ISP into one which meets those two Theorem-*13 *properties.*

An ISP with two or more MISPs does not always meet those two Theorem-13 properties. However, Lemma 15 can be extended to fit for some situation where a promising addition affects two MISPs.

▶ **Lemma 16.** *If a promising addition affects two MISPs, each of which is isolated or meets those two Theorem-*13 *properties, it must transform the two MISPs into an ISP which meets those two Theorem-*13 *properties.*

To show Theorem 13, we need to observe on what kind of an ISP a promising cancellation can transform an MISP with those two Theorem-13 properties into.

▶ **Lemma 17.** *If a promising cancellation acts on an MISP with those two Theorem-*13 *properties, it must transform the MISP into an ISP in which all unsorted MISPs meets those two Theorem-*13 *properties.*

Similar to Theorem 5, Theorem 13 can be proved with Lemma 14, 15, 16 and 17.

A best cancellation must switch two elements between which another element has been caught. Thus we will usually denote by $\rho\langle i, i+2\rangle$ a best cancellation on π. A best cancellation can be identified by,

▶ **Lemma 18.** *A short swap, say $\rho\langle i, i+2\rangle$ on π is a best cancellation, if and only if $\pi_i \geq i+2$ and $\pi_{i+2} \leq i$.*

In π, there exist $\lfloor \frac{n}{2} \rfloor$ even elements and $\lceil \frac{n}{2} \rceil$ odd elements. Thus those even elements in π can be extracted into a subsequence of π as $[\pi_{x[1]}, \pi_{x[2]}, \ldots, \pi_{x[\lfloor \frac{n}{2} \rfloor]}]$ where, (1) $x[i] < x[i+1]$ for $1 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1$; (2) $\pi_{x[i]}$ is even in π, $1 \leq x[i] \leq n$. Likewise, those odd elements in π can be extracted into $[\pi_{y[1]}, \ldots, \pi_{y[\lceil \frac{n}{2} \rceil]}]$ where, (1) $y[i] < y[i+1]$ for $1 \leq i \leq \lceil \frac{n}{2} \rceil - 1$; (2) $\pi_{y[i]}$ is odd in π, $1 \leq y[i] \leq n$. Moreover, let $Even[\pi] \equiv [e_1, e_2 \ldots e_{\lfloor \frac{n}{2} \rfloor}]$ with $e_i = \frac{\pi_{x[i]}}{2}$, $1 \leq i \leq \lceil \frac{n}{2} \rceil$, $Odd[\pi] \equiv [o_1, o_2 \ldots o_{\lfloor \frac{n}{2} \rfloor}]$ with $o_i = \frac{\pi_{y[i]}+1}{2}$, $1 \leq i \leq \lceil \frac{n}{2} \rceil$. Then $Even[\pi]$ must be a permutation of $\{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$, $Odd[\pi]$ a permutation of $\{1, 2, \ldots, \lceil \frac{n}{2} \rceil\}$. A sufficient and necessary condition for a permutation to be lucky can be announced by,

▶ **Theorem 19.** *The permutation π is lucky if and only if,* (1) *each of its elements admits a vector with zero or even absolute value;* (2) *each unsorted MISP in $Even[\pi]$ and $Odd[\pi]$ is sub-lucky.*

Proof. Only if: Let π be lucky and unsorted, $\rho\langle i, i+2\rangle$ a best cancellation on π. Then $\rho\langle i, i+2\rangle$ must cause $|v_\pi(\pi_i)|$ as well as $|v_\pi(\pi_{i+2})|$ to decrease by 2. Since π can be transformed into ι by none other than best cancellations, $|\pi_j - j| \bmod 2 = 0$ for $1 \leq j \leq n$. The proof for π to meet the Theorem-19 property (1), is done.

A position-even (resp. position-odd) element in π remains position-even (resp. position-odd) in $\pi \cdot \rho\langle i, i+2\rangle$. Since π meets the Theorem-19 property (1), an even (resp. odd) element in π must be position-even (resp. position-odd). This implies $Even[\pi] = [\frac{\pi_2}{2}, \frac{\pi_4}{2}, \ldots, \frac{\pi_{2\lfloor \frac{n}{2} \rfloor}}{2}]$, $Odd[\pi] = [\frac{\pi_1}{2}, \frac{\pi_3}{2}, \ldots, \frac{\pi_{2\lceil \frac{n}{2} \rceil - 1}}{2}]$.

Let i be even. By Lemma 18, $\pi_i \geq i + 2$ and $\pi_{i+2} \leq i$. Thus $\frac{\pi_i}{2} \geq \frac{i}{2} + 1$ and $\frac{\pi_{i+2}}{2} \leq \frac{i}{2}$. By Corollary 12, $\rho\langle\frac{i}{2},\frac{i}{2}+1\rangle$ can be viewed as a promising cancellation which acts on an MISP in $Even[\pi]$. Thus, if one can use best cancellations to transform π into a permutation, say π' with $Even[\pi'] = Even[\iota]$, then all unsorted MISPs in $Even[\pi]$ are sub-lucky. The same argument can be employed to show that all unsorted MISPs in $Odd[\pi]$ are sub-lucky. The proof for π to meet the Theorem-19 property (2), is done.

If: Let π be unsorted and meet those two Theorem-19 properties. The proof for π to be lucky, is to show that one can find a best cancellation ρ on π which transforms π into a permutation which meets those two Theorem-19 properties. Firstly, the Theorem-19 property (1) implies that $Even[\pi] = [\frac{\pi_2}{2}, \frac{\pi_4}{2}, ..., \frac{\pi_{2\lfloor \frac{n}{2} \rfloor}}{2}]$, $Odd[\pi] = [\frac{\pi_1}{2}, \frac{\pi_3}{2}, ..., \frac{\pi_{2\lceil \frac{n}{2} \rceil - 1}}{2}]$.

Let π_i be the rightmost vector-right element in π. Then $\pi_{i+2} \leq i+2$ because π_{i+2} is either vector-zero or vector-left. We argue that if i is even, $\rho\langle i,i+2\rangle$ must be a best cancellation on π.

(1) Since i is even, $\pi_i \geq i + 2$, and $\frac{\pi_i}{2}$ and $\frac{\pi_{i+2}}{2}$ must occur in $Even[\pi]$.

(2) To get to $\pi_{i+2} \leq i$, we argue that $\frac{\pi_i}{2}$ and $\frac{\pi_{i+2}}{2}$ must occur in one unsorted MISP in $Even[\pi]$.

It follows $\pi_{i+2} \leq i+2$ and $\pi_i \geq i+2$ that $\frac{\pi_i}{2} \geq \frac{i}{2} + 1$ and $\frac{\pi_{i+2}}{2} \leq \frac{i}{2} + 1$. Thus $\frac{\pi_i}{2} > \frac{\pi_{i+2}}{2}$. Thus an inversion of $\frac{\pi_i}{2}$ and $\frac{\pi_{i+2}}{2}$ occurs in $Even[\pi]$, which means $\frac{\pi_i}{2}$ and $\frac{\pi_{i+2}}{2}$ occur in one MISP. By the Theorem-19 property (2), the MISP in $Even[\pi]$ with $\frac{\pi_i}{2}$ and $\frac{\pi_{i+2}}{2}$ must be sub-lucky. Thus by the Theorem-13 property (1), $\frac{\pi_{i+2}}{2}$ in $Even[\pi]$ is not vector-zero. It follows that $\frac{\pi_{i+2}}{2} \leq \frac{i}{2}$, and equivalently, $\pi_{i+2} \leq i$.

The same argument can be employed to show that if i is odd, $\rho\langle i,i + 2\rangle$ is a best cancellation.

Let $\pi' = \pi \cdot \rho\langle i, i + 2\rangle$. It remains to show that π', if unsorted, must meet those two Theorem-19 properties.

Since $\rho\langle i,i+2\rangle$ is a best cancellation, it must cause $|v_\pi(\pi_i)|$ and $|v_\pi(\pi_{i+2})|$ each to decrease by 2. Since π meet the Theorem-19 property (1), π' must meet the Theorem-19 property (1).

If i is even, since π meets the Theorem-19 property (1), then $\frac{\pi_{i+2}}{2}$ must occur on the right side next to $\frac{\pi_i}{2}$ in $Even[\pi]$. Since $\rho\langle i, i + 2\rangle$ is a best cancellation, $\rho\langle \frac{i}{2}, \frac{i}{2} + 1\rangle$ must be a promising cancellation which acts on an MISP in $Even[\pi]$. By Lemma 17, all unsorted MISPs in $Even[\pi']$ meet those two Theorem-13 properties. That is, all unsorted MISPs in $Even[\pi']$ are sub-lucky by Theorem 13. Moreover, it follows $Odd[\pi'] = Odd[\pi]$ that all MISPs in $Odd[\pi']$ are sub-lucky. Thus, π' meets Theorem-19 property (2)

If i is odd, π' can be shown to meet the Theorem-19 property (2) in the same way as for i to be even. ◀

To decide if π meets the Theorem 19 property (1), it suffices to check for all i in [1, n], if i and π_i are both even, or both odd.

Let π_i be an arbitrary element in π. We refer to $\frac{\pi_i}{2}$ (resp. $\frac{\pi_i+1}{2}$) as the *image* of π_i in $Even[\pi]$ (resp. $Odd[\pi]$). Then for a lucky permutation π, π_i is vector-right (resp. vector-left, vector-zero) in π, if and only if its image in $Even[\pi]$ or $Odd[\pi]$ is vector-right (resp. vector-left, vector-zero). Thus, to decide if π meets the Theorem-19 property (2), it suffices to check for, (1) if the image of a vector-zero element occurs in an isolated MISP in $Odd[\pi]$ or $Even[\pi]$; and (2) if those vector-left and even (resp. odd) elements in π, as well as those vector-right and even (resp. odd) elements, always increase monotonously in the order from π_1 to π_n.

The image in $Even[\pi]$ (resp. $Odd[\pi]$) of a vector-zero element, say π_i, can be decided to occur in an isolated MISP in $Even[\pi]$ (resp. $Odd[\pi]$) by checking if all even (resp. odd) elements in $\pi[1 \rightarrow i - 1]$ are smaller than π_i. Those vector-left (resp. vector-right) elements

Algorithm 3: How to recognize a lucky permutation.

Algorithm *lucky permutation*
Input: A permutation π.
Output: The best short swap sequence ρ if π is lucky; *no*, otherwise.
1 $lo \leftarrow 0$; $ro \leftarrow 0$; $le \leftarrow 0$; $re \leftarrow 0$;
2 For $i \rightarrow 1$ to n do
3   If (i and π_i are both even) then
4   If ($\pi_i \geq i$ and $\pi_i > re$)
5   then $re \leftarrow \pi_i$; $i \leftarrow i + 1$; (π_i is vector-right even or $[\pi_i]$ is isolated)
6   If ($\pi_i < i$ and $\pi_i > le$)
7   then $le \leftarrow \pi_i$; $i \leftarrow i + 1$; (π_i is vector-left even)
8   If (i and π_i are both odd) then
9   If ($\pi_i \geq i$ and $\pi_i > ro$)
10   then $ro \leftarrow \pi_i$; $i \leftarrow i + 1$; (π_i is vector-right odd or $[\pi_i]$ is isolated)
11   If ($\pi_i < i$ and $\pi_i > lo$)
12   then $lo \leftarrow \pi_i$; $i \leftarrow i + 1$; (π_i is vector-left odd)
13   Else return *no*;
14 End for
15 Return Sort(π);

can be decided to be monotonous increasing by checking for each vector-left (resp. vector-right) even (resp. odd) element, say π_i, if π_i is bigger than the biggest vector-left (resp. vector-right) even (resp. odd) element in $\pi[1 \rightarrow i - 1]$. In fact, it is not necessary to pay special attention to check if a vector-zero element occurs in an isolated MISP. This benefits from

▶ **Lemma 20.** *In $\pi[1 \rightarrow k]$ for $k \geq 2$, the biggest vector-right element must be bigger than the biggest vector-left element.*

We present in Figure 3 the algorithm to decide if π is lucky, and if so, to find a best cancellation sequence to sort π. If π is lucky, the algorithm will return a best cancellation sequence which can transform π into ι by invoking the Sort(π); return *no*, otherwise. Since by the sufficiency proof of Theorem 19, one can employ the same way as to find a best short swap in Theorem 5 to find a best cancellation, the subroutine Sort(π) is just so as it has been depicted in Algorithm 2.

In the algorithm description, we use the integer parameter le (resp. lo) to maintain the biggest vector-left even (resp. odd) element in $\pi[1 \rightarrow i - 1]$, re (resp. ro) the biggest even (odd) element in $\pi[1 \rightarrow i - 1]$. It follows Lemma 20 that $le < re$, $lo < ro$.

Running the algorithm from Step 1 to Step 14 can inform us if π is lucky or not. This takes $O(n)$ time, where n is the number of elements in π. Let π_i be the rightmost vector-right element in a lucky permutation π, by the proof of Theorem 19, the short swap which switches π_i with π_{i+2} is a best cancellation. By Theorem 19 again, this operation must transform π into a lucky permutation. By the complexity analysis for Sort(π) in Section 3, it has been known Sort(π) can run in $O(n^2)$ time. Thus the time complexity of sorting a lucky permutation is $O(n^2)$.

5 Conclusion

Sort a happy permutation or a lucky permutation by short swaps is a special case of minimum sorting by short swaps problem. In this paper, we proposed a polynomial-time algorithm to recognize a happy permutation and sort it with the fewest short swaps. We also gave a new algorithm to recognize a lucky permutation with $O(n)$ steps, which improves the time complexity of $O(n^2)$ [9]. The complexity of minimum sorting by short swaps problem remains open. The best known approximation ratio of this problem is 2, which was given by Heath and Vergara [9]. It is interesting that if we can get a smaller approximation ratio for this problem.

References

1. V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *Siam Journal on Computing*, 25(2):272–289, 1993.
2. Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-approximation algorithm for sorting by reversals. *Lecture Notes in Computer Science*, pages 200–210, 2002.
3. Guillaume Bourque and Pavel A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Research*, 12(1):26–36, 2002.
4. Alberto Caprara. Sorting permutations by reversals and eulerian cycle decompositions. *Siam Journal on Discrete Mathematics*, 12(1):91–110, 1999.
5. Xuerong Feng, Ivan Hal Sudborough, and Enyue Lu. A fast algorithm for sorting by short swap. In *Proceeding of the 10th IASTED International Conference on Computational and Systems Biology*, pages 62–67, 2006.
6. Gustavo Rodrigues Galvão and Zanoni Dias. Approximation algorithms for sorting by signed short reversals. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '14, Newport Beach, California, USA, September 20-23, 2014*, pages 360–369. ACM, 2014. doi:10.1145/2649387.2649413.
7. Gustavo Rodrigues Galvão, Orlando Lee, and Zanoni Dias. Sorting signed permutations by short operations. *Algorithms for Molecular Biology*, 10(1):1–17, 2015.
8. Sridhar Hannenhalli. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the Acm*, 46(1):1–27, 1999.
9. L. S. Heath and J. P. Vergara. Sorting by short swaps. *Journal of Computational Biology A Journal of Computational Molecular Cell Biology*, 10(5):775–89, 2003.
10. Mark R. Jerrum. The complexity of finding minimum-length generator sequences. In *Colloquium on Automata, Languages and Programming*, pages 270–280, 1984.
11. Haim Kaplan, Ron Shamir, and Robert E. Tarjan. *A Faster and Simpler Algorithm for Sorting Signed Permutations by Reversals.* Society for Industrial and Applied Mathematics, 1999.
12. J. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1-2):180–210, 1995.
13. P Pevzner and G Tesler. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Research*, 13(1):37–45, 2003.
14. G. P. Pradhan and P. V. Prasad. Evaluation of wheat chromosome translocation lines for high temperature stress tolerance at grain filling stage. *Plos One*, 10(2):1–20, 2015.
15. D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B F Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences of the United States of America*, 89(14):6575–6579, 1992.
16. G. A. Watterson, W. J. Ewens, T. E. Hall, and A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99(1):1–7, 1982.