Sorting by Swaps with Noisy Comparisons*

Tomás Gavenciak
Department of Computer Science,
ETH Zurich, Zurich, Switzerland
tomas.gavenciak@inf.ethz.ch

Barbara Geissmann
Department of Computer Science,
ETH Zurich, Zurich, Switzerland
barbara.geissmann@inf.ethz.ch

Johannes Lengler
Department of Computer Science,
ETH Zurich, Zurich, Switzerland
johanes.lengler@inf.ethz.ch

ABSTRACT

We study sorting of permutations by random swaps if the comparison operator is noisy. The noise is not associated with the underlying fitness, but is inherent to the comparison operator. This type of fitness-independent noise has not been studied before in the community, but is prototypical for comparison-based evolutionary algorithms, which often do not need to compute or approximate explicit fitness values. As quality measure, we compute the average fitness of the stationary distribution. To measure runtime, we compute the minimal number of steps after which the expected fitness approximates the average fitness of the stationary distribution.

As mutations we allow swaps of any two elements which have distance at most r. We give theoretical results for the extreme cases $r = 1$ and $r = n$, and experimental results for intermediate cases. We find a tradeoff between faster convergence (for large r) and better average quality of the solution after convergence (for small r).

ACM Reference format:
Tomás Gavenciak, Barbara Geissmann, and Johannes Lengler. 2017. Sorting by Swaps with Noisy Comparisons. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17),* 9 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

An important and classical aspect of evolutionary algorithms (EAs) is how robust their performance is in the presence of noise [6]. This theme has gained increased attention in the last few years, [1–4, 7, 11, 14, 15], see [18] for a comprehensive review. Mostly, noise is modelled by imperfect fitness function evaluations that – instead of the exact fitness value – return a perturbed value (e.g., by a Gaussian additive term). This model is very accurate for algorithms which explicitly use the fitness. However, it is less useful for *comparison-based* EAs which do not compute (or approximate) the underlying fitness function. For example, if genetic algorithms are used to optimize chess engines, then the selection process will not rely on fitness values, but rather on comparisons (e.g., by tournaments) between different engines.

*Research supported by SNF (project number 200021_165524)

GECCO '17, Berlin, Germany

DOI: 10.1145/nnnnnnn.nnnnnnn

In this paper we investigate how to evaluate comparison-based EAs in the presence of noise. Crucially, the noise in our model does not perturb the fitness, but rather we assume that the process of comparing is error-prone. More precisely, we assume that every comparison gives a false output with some probability $p < 1/2$.

We use sorting by swaps as a prototypical example, since a permutation x can naturally be compared with any offspring y that is created from x by a single swap, without reference to a fitness function. Moreover, we will restrict to the case of comparison-based $(1 + 1)$ evolutionary algorithms on some search space $\mathcal{S}_n$, as described by Algorithm 1. The general approach can be extended to $(\mu + \lambda)$ algorithms, but there are some additional subtleties which we want to avoid – for example, which of the μ individuals of the last generation the algorithm actually chooses as output.

There are some problems that make it more complicate to define a theoretical evaluation of comparison-based EAs in the presence of noise. For example, the standard measure for the runtime of an EA in theoretical studies is the number of fitness evaluations until an optimal solution is hit for the first time. This is arguably unsuitable for noisy comparison-based algorithms, because even if they do find an optimum, due to the noisy measurements the algorithm might not even be able to recognise it as a best-so-far solution. Moreover, in a noisy environment the global optimum may not have a practical advantage over some other search points. Thus if we expect the algorithm to find the global optimum, we force it to spend possibly a lot of time for searching through the set of all solutions which are practically indistinguishable. An alternative which avoids the aforementioned problems is the fixed-budget approach, in which we ask for the best solution that the algorithm obtains within a fixed budget B of function evaluations. However, this approach has the disadvantage that it needs an additional parameter, the budget B. Instead we aim for parameter-free alternatives.

Algorithm 1: The class of noisy (1+1) evolutionary algorithms for optimizing an unknown function $f : \mathcal{S}_n \to \mathbb{R}$. The (possibly randomized) mutation operator Mut creates a new individual $\textsc{Mut}(x)$ from x. The IsFitter operator tries to decide whether its first argument is fitter than its second argument, but gives an erroneous answer with probability p.

Initialize: Sample $x^{(0)}$ uniformly at random from $\mathcal{S}_n$.
Optimize: for $t = 1, 2, 3, \ldots$ **do**
  $y^{(t)} \leftarrow \textsc{Mut}(x^{(t)})$.
  if $\textsc{IsFitter}(y^{(t)}, x^{(t-1)})$ **then**
    $x^{(t)} \leftarrow y^{(t)}$,
  else
    $x^{(t)} \leftarrow x^{(t-1)}$.

As a solution, we regard the algorithm as a Markov chain on its state space. As such, the algorithm converges to some stationary distribution of states, the *stationary distribution*, and the fitness of the output can naturally be defined as the expected fitness of the stationary distribution. Moreover, there is a well-established notion of mixing time, which is a natural measure for the time after which the solution does not change further. However, this notion is not necessarily a good measure for the time until good solutions are found. The mixing time measures the time until the *genotypical* distribution of solutions becomes stable. But if there is, for example, a large plateau of almost-optimal solutions, then the time to reach this plateau may be significantly shorter than the mixing time. Thus we rather measure the runtime of the algorithm by the minimal time until the expected quality of the solution is close to the expected quality of a solution of the stationary distribution. Intuitively (but not formally), this corresponds to convergence of the *phenotypical* distribution instead of the *genotypical* distribution.

As mentioned before, we study the notions discussed above for the *sorting* problem, i.e., the individuals are permutations of the set $\{1, \dots, n\}$. This problem has been introduced in the seminal paper of Scharnow, Tinnefeld and Wegener [20], and it has been studied in different encodings [10]. Several mutation operators are discussed in [20], one of which is the *swap operator* $\textsc{Swap}(i, j)$, which exchanges the elements at position i and j. In this paper we will only consider Swap operations since they are the only ones for which the algorithm can decide in constant time whether the operation is advantageous or not (i.e., if the two elements were in the wrong order before swapping or not). Thus Swap operations allow naturally to compare parent and offspring without explicitly accessing the fitness of either search point. For $0 < p < 1/2$ and $1 \le r \le n$ we will study the algorithm $\textsc{SwapSort}_{p,r}$, which produces a random offspring by choosing uniformly at random two elements in distance at most r and swapping them. The comparison between offspring and parent yields the correct answer with probability $1-p$, and the wrong answer with probability p. We will see that there is a trade-off between speed of convergence (for large r) and quality of the final solution (for small r). A similar trade-off for $r = 1$ and $r = n$ has been first observed in [12] in a slightly different noise model.

To measure the quality of the solution, we need to assume a *ground truth*, i.e., an unknown underlying fitness function. We consider the following options for a permutations π, see also Section 2 for formal definitions.

- $I(\pi)$ is the number of inversions, i.e., the number of pairs (i, j), $i < j$ such that $\pi(i) > \pi(j)$ [20].
- $W(\pi)$ is the weighted number of inversions, where each inversion (i, j) is weighted by $\pi(i) - \pi(j)$ [12].
- $D(\pi)$ is the total dislocation, i.e., the sum of all distances of elements i from their positions $\pi(i)$ [8].

The noisy process of sorting by random swaps and its variants has been studied in other contexts as well: Similar Markov chains are studied as *(biased) card shuffling processes*[1] with a focus on the mixing time as a measure of efficiency of the shuffling [5]. The $\{0, 1\}$−sequence sorting studied in Section 3.2 is related to the *asymmetric exclusion process*, studied in statistical physics to model the dynamics of continuous particle diffusion in an infinite space [21, 22]. We can model the process $\textsc{SwapSort}_{p,1}$ as a canonical ensemble using $I(\pi)$ as the energy function and temperature $1/\log(1/p - 1)$. We omit the details in this paper. To our best knowledge, the results of this paper have not been known in the above contexts unless explicitly stated.

[1] In biased card shuffling, card values influence swap probability, so the resulting permutation distribution is not uniform.

1.1 Our Results

1.1.1 Adjacent swaps. We first consider $\textsc{SwapSort}_{p,1}$, which only swaps adjacent elements. Our first result shows that this algorithm has high fitness in the stationary distribution.

Theorem 1.1. *Fix a constant $0 < p < 1/3$, and let π be a random permutation from the stationary distribution of $\textsc{SwapSort}_{p,1}$. Then*

$$p(n-1) \le \mathbb{E}(I(\pi)) \le \mathbb{E}(W(\pi)) \le n\frac{2p(1-p)}{(3p-1)^2} + 2^{-\Omega(n)}.$$

Moreover with high probability[2] *$I(\pi) \le W(\pi) = O(n \log n)$ and the maximum dislocation is $\max_{1 \le i \le n} |i - \pi(i)| = O(\log n)$.*

Note that the bounds on the expectations are asymptotically tight. For sufficiently small p and large n, the ratio of the upper and lower bounds is close to 2 for both I and W. This is illustrated on Figure 6 together with experimental results. Experiments of Section 4.3 suggest that $\mathbb{E}(I(\pi)) \simeq f_1(p)n$ and $\mathbb{E}(W(\pi)) \simeq f_2(p)n$ for some (unknown) functions f_1, f_2.

Our next result shows that the convergence time (in terms of fitness) of the algorithm is $\Theta(n^2)$ for both I and W. More precisely, let π be a random permutation from the stationary distribution of $\textsc{SwapSort}_{p,1}$, and let

$$T^{(I)}_{\text{conv}} := \min\left\{ t > 0 \mid \left| \frac{\mathbb{E}(I(x^{(t)}))}{\mathbb{E}(I(\pi))} - 1 \right| < \varepsilon \right\}$$

and

$$T^{(W)}_{\text{conv}} := \min\left\{ t > 0 \mid \left| \frac{\mathbb{E}(W(x^{(t)}))}{\mathbb{E}(W(\pi))} - 1 \right| < \varepsilon \right\}$$

the times until $\textsc{SwapSort}_{p,1}$ has approached the quality of its stationary distribution up to an error of ε. Then:

Theorem 1.2. *$T^I_{conv} = \Theta(n^2)$ and $T^W_{conv} = \Theta(n^2)$.*

We also run experiments on the convergence times in Section 4.2. For $p \le 0.2$, the measured convergence times are between n^2 and $2n^2$ (within 95% confidence).

1.1.2 Arbitrary swaps. Now we turn to $\textsc{SwapSort}_{p,n}$, which may swap any pair of elements. We do not provide theoretical result on the convergence time in this case. Experiments of Section 4.2 indicate that the convergence time is almost n times faster than for $r = 1$, and in particular suggest convergence time to be bounded by $O(n \log n)$. This would be also consistent with the mixing time of a random card-swapping process shown to be $O(n \log n)$ by Diaconis [9, chapter 3D]. However, this increase in speed comes at a cost, since the quality of the solution is dramatically worse than for $\textsc{SwapSort}_{p,1}$.

[2] That is with probability at least $1 - 1/n$.

Theorem 1.3. *Fix a constant $0 < p < 1/2$, and let π be a random permutation from the stationary distribution of $\textsc{SwapSort}_{p,n}$. Then*

$$\mathbb{E}(I(\pi)) = \Theta(n^2), \quad \mathbb{E}(D(\pi)) = \Theta(n^2), \quad \mathbb{E}(W(\pi)) = \Theta(n^3).$$

More precisely,

$$2 \cdot \mathbb{E}(I(\pi)) \geq \mathbb{E}(D(\pi)) \geq \frac{p(n^2-1)}{6}, \text{ and } \mathbb{E}(W(\pi)) \geq \frac{pn^3}{648}.$$

We remark that the upper bounds in Theorem 1.3 are trivial and hold for any permutation. In particular, the algorithm achieves only a constant improvement over a random permutation. Similarly to $\textsc{SwapSort}_{p,1}$, experiments of Section 4.3 indicate that $\mathbb{E}(I(\pi)) \simeq f_3(p)n^2$ and $\mathbb{E}(W(\pi) \simeq f_4(p)n^3$ for some (unknown) functions f_3, f_4. For $p \to 0.5$, both $\mathbb{E}(I)$ and $\mathbb{E}(W)$ smoothly converge to the expected fitness functions of a random permutation.

1.1.3 Bounded-range swaps. For the intermediate range, $1 < r < n$, experiments also suggest a smooth transition between the above results, namely $\mathbb{E}(I(\pi)) = \Theta(nr)$ and $\mathbb{E}(W(\pi)) = \Theta(nr^2)$ for any fixed $p \leq 0.3$. We show lower bounds on the fitness functions:

Theorem 1.4. *For any constant $0 < p < \frac{1}{2}$ and any $r \in \{1, \dots, n\}$, if π is a random permutation from the stationary distribution of $\textsc{SwapSort}_{p,r}$,*

$$\mathbb{E}(I(\pi)) = \Omega(rn), \quad \mathbb{E}(D(\pi)) = \Omega(rn), \quad \mathbb{E}(W(\pi)) = \Omega(r^2 n).$$

2 NOTATION AND FORMAL DEFINITIONS

2.1 Algorithm

Let $\mathcal{S}_n$ be the set of all permutations of $\{1, \dots, n\}$, i.e., the set of all bijective functions $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$. For a parameter $r \in \{1, \dots, n\}$ and a parameter $p < 1/2$, we define $\textsc{SwapSort}_{p,r}$ to be the $(1+1)$ evolutionary algorithm as given by Algorithm 1, with search space $\mathcal{S}_n$ and the following $\textsc{Mut}_r$ and $\textsc{IsFitter}_p$ operations. $\textsc{Mut}_r(x)$ selects two indices i, j with $1 \leq |i-j| \leq r$ uniformly at random and returns the permutation $y = \textsc{Swap}(x, i, j)$, where

$$y(k) := \begin{cases} x(j) & \text{if } k = i\,, \\ x(i) & \text{if } k = j\,, \\ x(k) & \text{otherwise}\,. \end{cases}$$

We say that a swap $\textsc{Swap}(x, i, j)$ is *successful* if x_i and x_j are correctly sorted after swapping, i.e., if $i < j$ and $x_i > x_j$, or if $i > j$ and $x_i < x_j$. The $\textsc{IsFitter}_p(x, y)$ operator is only defined on permutations x and $y = \textsc{Swap}(x, i, j)$ which differ by a single swap. If the swap was successful, then $\textsc{IsFitter}_p(x, y)$ returns y with probability $1-p$, and x with probability p. Otherwise, it returns x with probability $1-p$, and y with probability p.

We use $S_{p,r}$ to denote the Markov chain associated with the algorithm $\textsc{SwapSort}_{p,r}$ extended to all n-element sequences (the actual values do not matter and do not change in the process, and values may repeat). However the reachable states are determined by the (multi-)set of values, and for starting with permutations the state space is $\mathcal{S}_n$. Each combined mutation/selection operation corresponds to one transition step of the Markov chain [17]. Let $S_{p,r}(\pi)$ the process $S_{p,r}$ starting with permutation π, $S_{p,r,n}$ starting from a random n-element permutation and let $S^{\infty}_{p,r,n}$ be the unique stationary distribution (which exists since $S_{p,r}$ is irreducible and positive recurrent). Finally, we let $x^{(t)}$ be a random permutation after t steps when $S_{p,r}(\pi)$ is clear from the context.

2.2 Fitness Functions

We consider the following three functions as fitness function to be minimized: the total dislocation in a permutation π is the sum of displacement of all elements, where the displacement of an element is the absolute difference between its positions in π and in the sorted n-element sequence $\pi_0 = (1, \dots, n)$; the number of inversions is the number of pairs of elements in π that are placed in a different order than in π_0; the weighted number of inversions is the sum of all absolute differences in value of inverse pairs. Let $\pi(i)$ denote the element in π at position i. The formal definitions of the three fitness functions are as follows:

Definition 2.1. Let $\pi \in \mathcal{S}_n$ be a permutation of the set $\{1, \dots, n\}$. The *number of inversions* of π is

$$I(\pi) := \sum_{i<j\colon \pi(i)>\pi(j)} 1\,.$$

The *weighted number of inversions* of π is

$$W(\pi) := \sum_{i<j\colon \pi(i)>\pi(j)} |\pi(i) - \pi(j)|\,.$$

The *total dislocation* (or *Spearman footrule*) of π is

$$D(\pi) := \sum_{i\in[n]} |i - \pi(i)|\,.$$

Note that any of these fitness functions decreases with a successful swap. Therefore, the operator $\textsc{IsFitter}(x, y)$ can equivalently be defined by comparing the fitness of x and y, and returning the fitter with probability $1-p$, and the less fit with probability p. We used the previous definition to emphasize that $\textsc{IsFitter}$ can be defined without reference to any explicit fitness function.

Observe that for every permutation of the set $\{1 \dots, n\}$, the weighted number of inversions is at least the number of inversions. This follows immediately from their definitions, since every two elements in such a permutation differ by at most one:

Lemma 2.2. *$W(\pi) \geq I(\pi)$ for any permutation $\pi \in \mathcal{S}_n$.*

There is an upper bound for each fitness function. The number of different pairs in a set of n elements is $\binom{n}{2}$. Therefore, the maximum number of inversions in a permutation is $\binom{n}{2}$. The total sum of the absolute differences of all pairs in the set $\{1, \dots, n\}$ is $\sum_{k=1}^{n-1}(n-k)k = \binom{n+1}{3}$. The maximum total dislocation of a permutation on n elements is $\lfloor \frac{n^2}{2} \rfloor$, see [16]. These three upper bounds are all tight for the reversed sorted permutation $(n, \dots, 1)$.

Lemma 2.3. *For $\pi \in \mathcal{S}_n$ a permutation of the set $\{1, \dots, n\}$,*

$$I(\pi) \leq \binom{n}{2}, \qquad W(\pi) \leq \binom{n+1}{3}, \qquad D(\pi) \leq \left\lfloor \frac{n^2}{2} \right\rfloor.$$

3 THEORETICAL ANALYSIS

3.1 Sorting with adjacent swaps

Here we theoretically analyze $\textsc{SwapSort}_{p,1}$, the sorting algorithm with adjacent swaps, and its Markov chain $S_{p,1}$. A main tool will be

a sorting process of 0-1 sequences, which is analyzed in Section 3.1.3, and coupled to $S_{p,1}$ in Section 3.1.4.

3.1.1 Convergence speed.

Proof of Theorem 1.2. Fix $\varepsilon > 0$. Benjamini et al. [5] show that for any constant $p < 1/2$, the mixing time of the Markov chain $S_{p,1,n}$ is $T_{\text{mix}} = O(n^2)$. After mixing, the relative error of probabilities of resulting permutations compared to the stationary distribution are below ε, and so are the relative errors for the distributions of the marginals I and W. This gives us $(1-\varepsilon)\mathbb{E}(I(S^\infty_{p,1,n})) < \mathbb{E}(I(x^{(t)})) < (1+\varepsilon)\mathbb{E}(I(S^\infty_{p,1,n}))$ for any $t \geq T_{\text{mix}}$, and similarly for W.

On the other hand, every swap of adjacent elements reduces the number of inversion by 1. So in expectation, $\Omega(n^2)$ swaps are needed to go from a random permutation (with $\Theta(n^2)$ expected inversions) to a permutation with $O(n)$ inversions. By Theorem 1.1, a permutation in the stationary distribution has $\mathbb{E}(I) = O(n)$ and $\mathbb{E}(W) = O(n)$, and since by Lemma 2.2 we have $W(\pi) \geq I(\pi)$, the lower bounds follow. □

3.1.2 Properties of the Markov chain.

For any $0 < p < 1/2$ and any $n > 1$, the process $S_{p,1}(s)$ on any sequence $s = (s_1, \ldots, s_n)$ is aperiodic, irreducible, and all the states are ergodic. Moreover we show the following.

Lemma 3.1. *For any $0 < p < 1/2$ and any n-element sequence s, the Markov chain $S_{p,1}(s)$ is reversible and has a unique stationary distribution P_0 with*

$$P_0(s') = \left(\frac{p}{1-p}\right)^{I(s')} / Z\,,$$

for any s' with the same value composition as s, where Z is a normalizing constant for the distribution (only depending on p and the composition of s).

Proof. Let $c = p/(1-p)$. Note that for any sequence s, $P_0(s)$ only depends on $I(s)$. After one step from the distribution P_0, for any s we have $P_0'(s) = p_a P_0(s) + p_b P_0(s^+)c^{-1} + p_c P_0(s^-)c^1$, where p_a is the probability that s was generated from s, p_b the probability that s was generated by a "correct" swap, decreasing the number of inversions, $P_0(s^+)$ is the probability of any single state with $I(s^+) = I(s) + 1$, and symmetrically p_c is the probability that s was generated by an "incorrect" swap and $P_0(s^-)$ is the probability of any single state with $I(s^-) = I(s) - 1$. By expanding the definitions of $P_0(s^+)$ and $P_0(s^-)$, we get $P_0'(s) = (p_a + p_b + p_c)P_0(s) = P_0(s)$. Therefore P_0 is a stationary distribution.

Verifying reversibility is a straightforward computation since the swaps are chosen uniformly and all states with the same I are equally likely. The uniqueness of P_0 follows from irreducibility and finiteness. □

3.1.3 Sorting 0-1 sequences.

Let $B_{p,n,k} = S_{p,1}(0^k 1^{n-k})$, e.g. the sorting process of k zeroes and $n-k$ ones, and let $B^\infty_{p,n,k}$ denote its stationary distribution. Since the stationary distribution is unique by Lemma 3.1, the starting state choice is just a convenience.

Lemma 3.2. *For any constant $0 < p < 1/3$, any n and $0 < k < n$ we have*

$$\mathbb{E}(I(B^\infty_{p,n,k})) \leq \frac{2p(1-p)}{(3p-1)^2} + 2^{-\Omega(n)}\,.$$

Moreover, for some $\lambda(p)$ depending only on p, and any $l \geq 0$,

$$\mathbb{P}[I(B^\infty_{p,n,k}) > \lambda(p) + l] < \bar{c}^{\,l}\,,$$

where $\bar{c} = 2p/(1-p)$.

Proof. For a 0-1 sequence $s = (s_1, \ldots, s_n)$, let $u = \{s_i < s_{i+1} | 1 \leq i \leq n-1\}$ and $d = \{s_i > s_{i+1} | 1 \leq i \leq n-1\}$ be the number of *up*-transitions and *down*-transitions of s, respectively. Obviously, $d - 1 \leq u \leq d + 1$.

In the state with sequence s, the probability that $I(s)$ increases in one step by 1 is $p^\uparrow = \frac{up}{n-1}$, since this happens only when a up-transition pair is selected and the elements are sorted as descending. Analogously, the probability $I(s)$ decreases by 1 is $p^\downarrow = \frac{d(1-p)}{n-1}$. Note that $I(s)$ may not change by more than 1 in a single step.

For the sequence of values of I, we can observe that whenever $d > 0$ and $u > 0$ it holds that

$$\frac{p^\uparrow}{p^\downarrow} = \frac{up}{d(1-p)} \leq \frac{(d+1)p}{d(1-p)} \leq \frac{2p}{1-p}\,.$$

The only state with $d = 0$ is the sorted state with $I = 0$ and the only state with $u = 0$ is the reversed sorted state with $I = I_{max} = k(n-k)$. In all other states, the ratio $p^\uparrow/p^\downarrow$ depends only on u and d and is upper bounded by $2p/(1-p)$.

To upper bound $\mathbb{E}(I)$, consider the random walk $\bar{I}$ on $0, \ldots, I_{max}$ with $\bar{p}^\uparrow = 2p/(p+1)$ and $\bar{p}^\downarrow = (1-p)/(p+1)$, so $\bar{p}^\uparrow + \bar{p}^\downarrow = 1$. In the stationary distribution of $\bar{I}$, $\mathbb{P}[\bar{I} = i] = c^i/Z$ where $\bar{c} = \bar{p}^\uparrow/\bar{p}^\downarrow = 2p/(1-p)$ and $Z = 1 - \bar{c}^{I_{max}}$ is a normalizing constant. This follows directly by analyzing the ratios $\mathbb{P}[\bar{I} = i]/\mathbb{P}[\bar{I} = i+1]$.

By a direct series summation and using $I_{max} \geq n$ and $\bar{c} < 1$ we get

$$\mathbb{E}(\bar{I}) = \sum_{i=0}^{I_{max}} i\bar{c}^{\,i} Z = \frac{\bar{c}}{(1-\bar{c})^2} + O(\bar{c}^{\,n}) = \frac{2p(1-p)}{(3p-1)^2} \pm 2^{-\Omega(n)}\,.$$

Since $p^\uparrow/p^\downarrow \leq \bar{p}^\uparrow/\bar{p}^\downarrow$ we can monotonically couple I and $\bar{I}$, getting $\mathbb{E}(I) \leq \mathbb{E}(\bar{I})$ in the stationary distributions. For the second part of the lemma, note that $\mathbb{P}[\bar{I} > l] = \sum_{i=l+1}^\infty \bar{c}^i/Z = \bar{c}^{\,l+1}/((1-\bar{c})Z) = \bar{c}^{\,l}\bar{c}^{\,1-\log_{\bar{c}}((1-\bar{c})Z)}$. With $\lambda(p) = \log_{\bar{c}}((1-\bar{c})Z)$ we get $\mathbb{P}[\bar{I} > \lambda(p) + l] < \bar{c}^l$, and the bound for I follows since the coupling is monotone. □

3.1.4 Sorting permutations.

Proof of Theorem 1.1. Let $\pi_0 = (1, 2, \ldots, n)$ be the sorted n-element sequence. By Lemma 3.1, the stationary distribution $S^\infty_{p,1,n}$ is the same as $S^\infty_{p,1}(\pi_0)$, so we analyze the latter. For a permutation π, let $T_k(\pi) \in \{0,1\}^n$ be the *k-th threshold 0-1 sequence* where $T_k(\pi)_i = 1$ if $\pi_i > k$ and $T_k(\pi)_i = 0$ if $\pi_i \leq k$. So $T_k(\pi)$ contains exactly k zeros and $n-k$ ones.

We decouple the process $S_{p,1}(\pi_0)$ into $n-1$ processes denoted $(B_{p,n,1}, \ldots B_{p,n,n-1})$: The state π of the process $S_{p,1}(\pi_0)$ corresponds in the states $(T_k(\pi), \ldots, T_k(\pi))$ of the 0-1 process $B_{p,n,k}$. The coupled processes share the event space: In every step we randomly choose two adjacent positions $(i, i+1)$ to be compared and

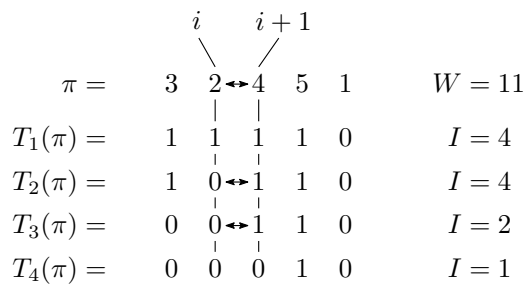

Figure 1: A single step of the coupled processes.

with probability p choose to order their values in the descending (wrong) order, and ascending (right) order otherwise. The same event then decides the change in all the coupled processes. It is straightforward to see that starting from corresponding states of $S_{p,1}$ and $(B_{p,n,1}, \dots B_{p,n,n-1})$, the resulting states after one event are again corresponding. See Figure 1 for an illustration.

For any permutation π we have $W(\pi) = \sum_{k=1}^{n-1} I(T_k(\pi))$ by re-ordering the summation:

$$W(\pi) = \sum_{\substack{i<j \\ \pi_i > \pi_j}} \pi_i - \pi_j = \sum_{k=1}^{n-1} \sum_{\substack{i<j \\ \pi_i > k \\ \pi_j \le k}} 1 = \sum_{k=1}^{n-1} I(T_k(\pi)) \,. \tag{1}$$

In the stationary distribution of the coupled processes, we have $\mathbb{E}(W(S^{\infty}_{p,1,n})) = \sum_{k=1}^{n-1} \mathbb{E}(I(B^{\infty}_{p,1,k}))$ and the first part of the theorem follows from Lemma 3.2 and Lemma 2.2.

To see the bound on $W(S^{\infty}_{p,1,n})$ with high probability, we use the tail bounds of Lemma 3.2: For any k and $\beta > 0$, $\mathbb{P}[I(B^{\infty}_{p,n,k}) > \lambda(p)+\beta \log n] < \bar{c}^{\,\beta \log n}$. For β big enough we get $\bar{c}^{\,\beta \log n} = O(n^{-3})$ and by a union bound over $n-1$ events $I(B^{\infty}_{p,n,k}) > \lambda(p) + \beta \log n$ for $k = 1, \dots, n-1$. Together with 1, this implies the bound on W. The bound for I follows from Lemma 2.2.

To show the maximum dislocation bound, consider a state π, any permutation element $1 \le k \le n$ and let $j = \pi^{-1}(k)$. The dislocation of k is then $|k-j|$. If $j < k$, $T_k(\pi)_j = 1$ and necessarily $I(T_k(\pi)) \ge k - j$. By Lemma 3.2, $\mathbb{P}[k - j > \beta \log n] < n^{-3}$ in the stationary distribution for some β only depending on p. A symmetric argument shows that $\mathbb{P}[j - k > \beta \log n] < n^{-3}$ for $j > k$. By union bound of the events "element k has dislocation at least $\beta \log n$" for all $1 \le k \le n$, we obtain the maximum dislocation claim.

We show the lower bound on $\mathbb{E}(I(\pi))$ to hold after any number of steps: consider any finite length realization of the random process starting from a random permutation π_0 with the last permutation being π. Divide the $n-1$ pairs of adjacent positions into three groups: A have never had a swap, the last swap of B was wrong, and the last swap of C was right. It is easy to see that $\mathbb{E}(|C|) \le (1-p)(n-1)$ and so $\mathbb{E}(|A| + |B|) \ge pn$. In π we find one inverted pair for each B: the pair that was last swapped on B has to be also inverted in π. For each position in A, the expected number of inversions of π_0 between the elements left of A and right of A is at least 1 whenever $n \ge 3$, and all these inversions are also present in π. The claim follows from linearity of expectation and the observations that all the inverted pairs counted for A and B are distinct. □

3.2 Sorting with any swaps

We now analyze sorting with arbitrary swaps, i.e., we consider the Markov chain $S_{p,\infty}$. Observe that the Markov chain for any such process that allows non-adjacent swaps is not reversible by the Kolmogorov Criterion [13].

Proof of Theorem 1.3. The upper bounds follow immediately from Lemma 2.3.

For the lower bounds, consider the following equivalent way to describe one step in $S_{p,\infty}$: With probability $(1-2p)$, arrange the two randomly chosen elements correctly, and with probability $2p$, arrange them randomly (swap or do not swap them each with probability p).

Let permutation π be a state after $S_{p,\infty,n}$ has converged. Let R^+ be the set of all elements in π whose last step during the process was a random swap. Let R be a subset of R^+, such that if the last step for two elements in R^+ was the same random swap, let only one of them be in R and decide which one at random, otherwise, flip a coin to decide whether to include an element in R. Clearly, for each element, the probability to be in R is $\frac{p}{2}$, (p to be in R^+ and $\frac{1}{2}$ to be in R if being in R^+), and $\mathbb{E}(|R|) = \frac{p}{2}n$.

Each element of R got placed to a random position in π. Thus, the expected dislocation of an element in R is

$$\frac{1}{n} \cdot \frac{1}{n} \cdot 2 \cdot \sum_{i=1}^{n} \binom{i}{2} = \frac{n}{3} - \frac{1}{3n} \,.$$

Therefore, by linearity of expectation, the expected total dislocation is at least $\mathbb{E}(D(S^{\infty}_{p,n,n})) \ge |R| \left(\frac{n}{3} - \frac{1}{3n}\right) = \frac{p(n^2-1)}{6}$.

By the well-known inequalities of Diaconis and Graham [8], it holds that $I \ge \frac{1}{2}D$, thus $\mathbb{E}(I(S^{\infty}_{p,n,n})) \ge \frac{p(n^2-1)}{12}$.

The order in which the elements of R appear in π is random. Moreover, the ranks of the elements in R are uniformly distributed between 1 and n, as are their positions in which they appear in π. Consider the first $\frac{1}{3}n$ positions in π: The expected number of elements from R that are larger than $\frac{2}{3}n$ and appear in one of these positions is $\frac{1}{9}|R|$. By pigeon hole principle, the number of elements that are smaller than $\frac{1}{2}n$ and appear in the middle or last third of π is at least $\frac{1}{6}n$. All elements of this set are inverse to all elements in the first set and differ by at least $\frac{1}{6}n$. Therefore, $\mathbb{E}(W(S^{\infty}_{p,n,n})) \ge \frac{1}{6}n \cdot \frac{1}{9}|R| \cdot \frac{1}{6}n \ge \frac{1}{648}pn^3$. □

3.3 Sorting with bounded-range swaps

Finally, we turn to the general process where we allow swaps between elements whose that lie at most r positions apart, i.e., we consider the Markov chain of $S_{p,r}$.

Proof of Theorem 1.4. The prove is similar to the one of Theorem 1.3: We consider again the equivalent process description and the sets R^+ (the set of all elements whose last step was a random swap) and R (the subset of R^+ that includes each element with probability $\frac{1}{2}$), with $\mathbb{E}(|R|) = \frac{pn}{2}$.

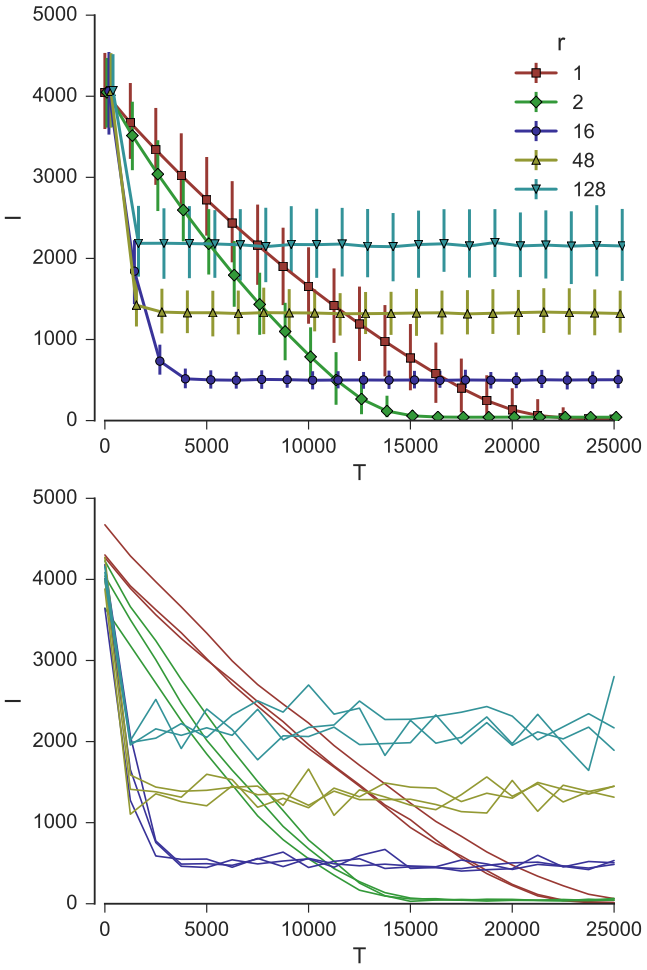

Figure 2: *Top:* Development of I in time for different values of r with $n = 512$ and $p = 0.1$ (mean and 95% confidence intervals over 300 runs). *Bottom:* Example developments of I for 3 independent process runs.

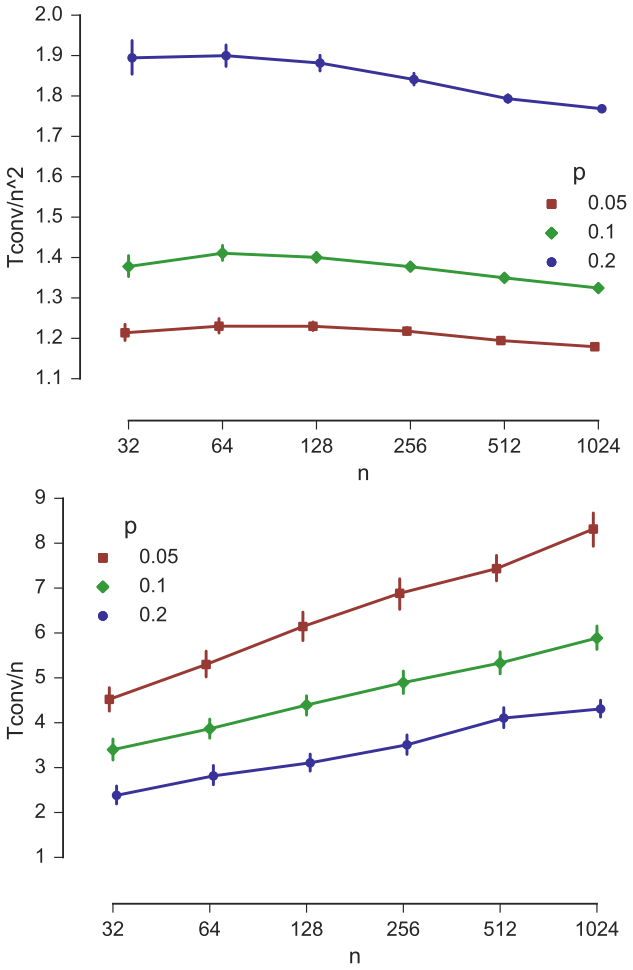

Figure 3: *Top:* T_{conv} normalized by n^2 for $r = 1$ and various n and p. *Bottom:* T_{conv} normalized by n for $r = n$ and various n and p. *In both:* mean and 95% confidence intervals over 300 runs.

For each element in R, its new expected dislocation will be larger than $\Omega(r)$, since the element is placed to a random position inside a radius r compared to its old position. Therefore, $\mathbb{E}(D(\pi)) \geq |R| \cdot \Omega(r) = \Omega(prn)$, and by [8], also $\mathbb{E}(I(\pi)) \geq \Omega(prn)$.

For the weighted number of inversions, we observe the following: If an element i has dislocation d_i, then it is inverse to at least d_i pairwise different larger or smaller elements (no such element can have the same difference to i), and its "contribution" to W is at least $\frac{1}{2}\binom{d_i+1}{2}$, where the factor $\frac{1}{2}$ is to prevent double counting. The expected value of $\binom{d_i+1}{2}$ is in $\Omega(r^2)$. Thus, $\mathbb{E}(W(\pi)) \geq |R| \cdot \Omega(r^2) = \Omega(pr^2n)$. □

We complement the theoretical analysis of Section 3 with experimental observations.

4 EXPERIMENTAL RESULTS

4.1 Methodology

The experimental results were obtained by simulating the process for a given number of steps or until the fitness function converged (for convergence criterion, see Section 4.2). The simulation is implemented in a combination of C++ and Python, the sources are freely available at GitHub. *(Note: To maintain double-blindness of the review process, we postpone publishing of the sources and the link.)* For reproducibility, the simulation uses a pseudo-random generator with a deterministic seed.

Every plot is based on 300 independent runs. We generally use n around 512 as the experimental results are consistent already for $n \geq 128$. All the plots include 95% confidence interval error bars even where these are actually not visible. (Bottom of Figure 2 is an exception to the above.)

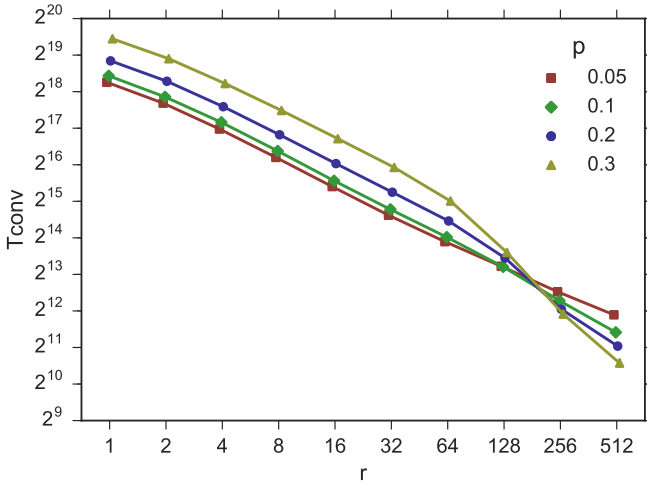

Figure 4: T_{conv} for $n = 512$ and various p and r (mean and 95% confidence intervals over 300 runs).

To illustrate the evolution of the solution quality in time, see Figure 2.

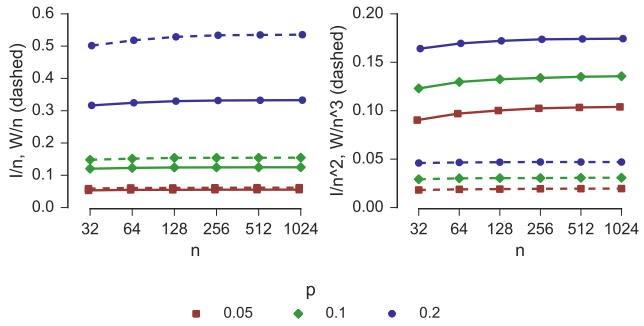

Figure 5: *Left:* I/n and W/n for $r = 1$ and various n and p. *Right:* I/n^2 and W/n^3 for $r = n$ and various n and p. *In both:* converged phase means and 95% confidence intervals over 300 runs.

4.2 Speed of convergence

An all the experiments, we use the convergence of $I(\pi)$ as the main stopping criterion[3]. Based on experiments, we assume the sequence $I(x^{(t)})$ for $t = 0, 1, \ldots$ is a monotonically decreasing function with an additive noise term. We define the convergence time T_{conv} as the smallest time t when

$$\mathbb{E}(I(x^{(t)})) \leq (1+\varepsilon)\mathbb{E}(I(S^{\infty}_{p,r,n})) \,.$$

For an overview and discussion on various stopping criteria, see [19].

To estimate $\mathbb{E}(I(x^{(t)}))$ we average over a sliding window starting at time t. To choose an appropriate size of the window proportional to the convergence time, we estimate convergence time as

$$T_{\text{est}} = \frac{n^2}{\bar{r}(1-2p)} \,,$$

[3]Experiments show that the convergence behavior of $W(\pi)$ is very similar and the convergence would differ by less than 5%.

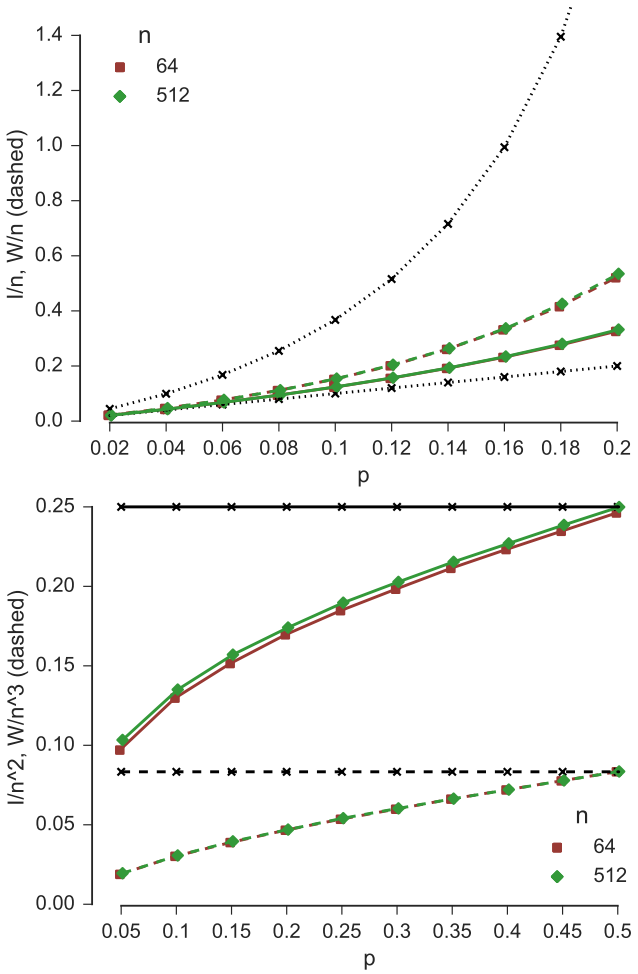

Figure 6: *Top:* Converged I/n and W/n for $r = 1$ and various n and p. Note that $n = 64$ and $n = 512$ overlap. The dotted lines are the upper and lower bounds on both $\mathbb{E}(I/n)$ and $\mathbb{E}(W/n)$ from Theorem 1.1. *Bottom:* Converged I/n^2 and W/n^3 for $r = n$ and various n and p. The horizontal lines indicate a random permutation expectation. *Both:* Converged phase mean and 95% confidence intervals over 300 runs.

where $\bar{r}$ is the average swap length, e.g. $\bar{r} = (\sum_{i=1}^{r} i(n-i))/(\sum_{i=1}^{r}(n-i))$. The experiments show that this estimate is within $0.2 - 2.3$ multiplicative error of the measured mean T_{conv} on our data. Note that the asymptotic behavior may be different and we need only a rough estimate.

We choose $\varepsilon = 0.05$ and window size $w = \lceil 0.05 T_{\text{est}} \rceil = \Omega(n)$. We also set a sampling rate $s = \lceil T_{\text{est}}/1000 \rceil$ to speed up the computations.

In step t divisible by s and with $t > 3w$, we compute mean I of windows starting at t and $\frac{3}{2}t$:

$$\overline{I(x^{(t)})} = \frac{1}{w}\sum_{i=t}^{t+w-1} I(i) \quad \text{and} \quad \overline{I(x^{(3/2\,t)})} = \frac{2}{t}\sum_{i=3/2\,t+1}^{2t} I(x^{(i)}) \,.$$

If $\overline{I(x^{(t)})} \leq (1+\varepsilon)\overline{I(x^{(3/2\,t)})}$ then we estimate $T_{\text{conv}} = t$.

If $t \geq \frac{2}{3}T_{\text{conv}}$ then $\overline{I(x^{(3/2)})}$ is an estimate for $\mathbb{E}(I(S^{\infty}_{p,r,n}))$. On the other hand if $t < \frac{2}{3}T_{\text{conv}}$ then t and $\frac{3}{2}t$ are both in the not-converged phase and seeing a false-positive would imply a very small average descent of I between the windows.

See Figure 3 for the plots of mean T_{conv} for the extreme cases $r = 1$ and $r = n$, and Figure 4 for a dependency on r. The experimental results indicate that for a fixed p and $r = 1$, $T_{\text{conv}} = \Theta(n^2)$ (in accordance with Theorem 1.2).

For $r \simeq n$, the time measurements are less accurate but consistent. The convergence times are only hundreds of steps with large fitness changes. Also, $\mathbb{E}(I)/n^2$ of the stationary distribution is close to that of a random permutation (see Figure 6). However, such imprecision of convergence time has no effect on the converged fitness measurements.

4.3 Converged state quality

We estimate the qualitative properties I and W of the stationary distribution as the distribution of the values in the range $[\frac{3}{2}T_{\text{conv}}, 2T_{\text{conv}}]$ over 300 independent process runs. In all the data, note that the 95% confidence error bars are very small and generally not visible.

For values of $p \leq 0.3$ and $n \leq 1024$, we observe that estimating $I(S^{\infty}_{p,1,n}) \simeq f_1(p)n$, $W(S^{\infty}_{p,1,n}) \simeq f_2(p)n$, $I(S^{\infty}_{p,n,n}) \simeq f_3(p)n^2$ and $W(S^{\infty}_{p,n,n}) \simeq f_4(p)n^3$ (for some fixed unspecified functions $f_1, \ldots, f_4$) are surprisingly accurate. See Figure 5.

To estimate the dependency on p (e.g. the functions $f_1, \ldots f_4$), see Figures 6 and 6. Finally, see Figure 7 for the experimental dependency of I and W on r. This trade-off corresponds to the lower bounds of Theorem 1.4. Note the non-linearity at $r \geq 128$ is likely to be caused by the average swap-distance $\bar{r}$ being lower than $r/2$ (while $\bar{r} \simeq r/2$ when $r \ll n$).

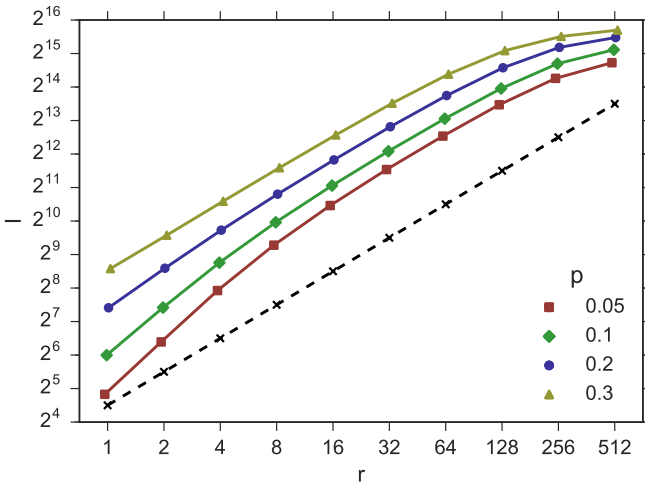

Figure 7: Converged I for $n = 512$ with various r and p (converged phase mean and 95% confidence intervals over 300 runs). The dashed line is the *direction* of the lower-bound from Theorem 1.4 (ignoring a mult. factor depending on p).

5 CONCLUSION

We have studied sorting by random swaps with a noisy comparison operator. We considered swaps of elements in distance at most r, and we found a trade-off between fast convergence (for large r) and high quality of the solution (for small r). We have only proven the results for the extreme cases $r = 1$ and $r = n$ formally, so a natural next step is to verify theoretically the experimental results for arbitrary r.

Since all parameter choices have strengths and weaknesses, an important question is whether an adaptive algorithm that decreases r over time (similar to Simulated Annealing) can profit from both cases. I.e., can such an algorithm achieve a linear expected (weighted) number of inversions in the stationary distribution with a sub-quadratic convergence time? We leave this question open for future research.

REFERENCES

[1] Youhei Akimoto, Sandra Astete-Morales, and Olivier Teytaud. 2015. Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. *Theoretical Computer Science* 605 (2015), 42–50.

[2] Sandra Astete-Morales, Marie-Liesse Cauwet, Jialin Liu, and Olivier Teytaud. 2016. Simple and cumulative regret for continuous noisy optimization. *Theoretical Computer Science* 617 (2016), 12–27.

[3] Sandra Astete-Morales, Marie-Liesse Cauwet, and Olivier Teytaud. 2015. Evolution strategies with additive noise: A convergence rate lower bound. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*. ACM, 76–84.

[4] Sandra Astete-Morales, Marie-Liesse Cauwet, and Olivier Teytaud. 2016. Analysis of Different Types of Regret in Continuous Noisy Optimization. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM, 205–212.

[5] Itai Benjamini, Noam Berger, Christopher Hoffman, and Elchanan Mossel. 2005. Mixing Times of the Biased Card Shuffling and the Asymmetric Exclusion Process. *Trans. Amer. Math. Soc.* 357, 8 (2005), 3013–3029.

[6] Hans-Georg Beyer. 2000. Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer methods in applied mechanics and engineering* 186, 2 (2000), 239–267.

[7] Marie-Liesse Cauwet, Jialin Liu, Baptiste Rozière, and Olivier Teytaud. 2016. Algorithm portfolios for noisy optimization. *Annals of Mathematics and Artificial Intelligence* 76, 1-2 (2016), 143–172.

[8] Persi Diaconis and R. L. Graham. 1977. Spearman's Footrule as a Measure of Disarray. *Journal of the Royal Statistical Society. Series B (Methodological)* 39, 2 (1977), 262–268. http://www.jstor.org/stable/2984804

[9] Persi W. Diaconis. 1988. *Group representations in probability and statistics*. Hayward, Calif. Institute of Mathematical Statistics. http://opac.inria.fr/record=b1087294

[10] Benjamin Doerr and Edda Happ. 2008. Directed trees: A powerful representation for sorting and ordering problems. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 3606–3613.

[11] Tobias Friedrich, Timo Kötzing, Martin S Krejca, and Andrew M Sutton. 2016. The Compact Genetic Algorithm is Efficient under Extreme Gaussian Noise. *IEEE Transactions on Evolutionary Computation* (2016).

[12] B. Geissmann and P. Penna. 2016. Sort well with energy-constrained comparisons. *ArXiv e-prints* (Oct. 2016). arXiv:cs.DS/1610.09223

[13] F.P. Kelly. 1979. *Reversibility and stochastic networks*. J. Wiley. https://books.google.ch/books?id=HhfvAAAAMAAJ

[14] Jialin Liu, David L St-Pierre, and Olivier Teytaud. 2014. A mathematically derived number of resamplings for noisy optimization. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 61–62.

[15] JJ Merelo, Zeineb Chelly, Antonio Mora, Antonio Fernández-Ares, Anna I Esparcia-Alcázar, Carlos Cotta, P de las Cuevas, and Nuria Rico. 2016. A statistical approach to dealing with noisy fitness in evolutionary algorithms. In *Computational Intelligence*. Springer, 79–95.

[16] Lon H. Mitchell. 2004. Maximal total absolute displacement of a permutation. *Discrete Mathematics* 274, 1-3 (2004), 319–321. DOI:http://dx.doi.org/10.1016/S0012-365X(03)00205-X

[17] Allen E Nix and Michael D Vose. 1992. Modeling genetic algorithms with Markov chains. *Annals of mathematics and artificial intelligence* 5, 1 (1992), 79–88.

[18] Pratyusha Rakshit, Amit Konar, and Swagatam Das. 2016. Noisy Evolutionary Optimization Algorithms-A Comprehensive Survey. *Swarm and Evolutionary*

Computation (2016).
[19] Martín Safe, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. 2004. *On Stopping Criteria for Genetic Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, 405–413. DOI: http://dx.doi.org/10.1007/978-3-540-28645-5_41
[20] Jens Scharnow, Karsten Tinnefeld, and Ingo Wegener. 2004. The analysis of evolutionary algorithms on sorting and shortest paths problems. *Journal of Mathematical Modelling and Algorithms* 3, 4 (2004), 349–366.
[21] Frank Spitzer. 1970. Interaction of Markov processes. *Advances in Mathematics* 5, 2 (1970), 246 – 290. DOI: http://dx.doi.org/10.1016/0001-8708(70)90034-4
[22] Craig A. Tracy and Harold Widom. 2009. Asymptotics in ASEP with Step Initial Condition. *Communications in Mathematical Physics* 290, 1 (2009), 129–154. DOI: http://dx.doi.org/10.1007/s00220-009-0761-0