Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

A Parameterized Runtime Analysis of Evolutionary Algorithms for the Euclidean Traveling Salesperson Problem

Andrew M. Sutton and **Frank Neumann**
School of Computer Science
University of Adelaide
Adelaide, SA 5005, Australia

Abstract

We contribute to the theoretical understanding of evolutionary algorithms and carry out a parameterized analysis of evolutionary algorithms for the Euclidean traveling salesperson problem (Euclidean TSP). We exploit structural properties related to the optimization process of evolutionary algorithms for this problem and use them to bound the runtime of evolutionary algorithms. Our analysis studies the runtime in dependence of the number of inner points k and shows that simple evolutionary algorithms solve the Euclidean TSP in expected time $O(n^{4k}(2k-1)!)$. Moreover, we show that, under reasonable geometric constraints, a locally optimal 2-opt tour can be found by randomized local search in expected time $O(n^{2k}k!)$.

Introduction

Stochastic search algorithms such as evolutionary algorithms (EAs) (Eiben and Smith 2007) and ant colony optimization (ACO) (Dorigo and Stützle 2004) have been applied to a wide range of combinatorial optimization problems. In contrast to numerous applications of these algorithms, it is hard to understand their behavior from a theoretical point of view. Our goal is to gain new insights into the working principles of these algorithms and show by rigorous analysis when and why this class of algorithm works. With this paper, we contribute to the theoretical understanding of evolutionary algorithms for combinatorial optimization problems by studying their computational complexity.

Related Work

Initial studies on the computational complexity of evolutionary algorithms have considered the behavior of these algorithms on artificial pseudo-Boolean functions (Droste, Jansen, and Wegener 2002; He and Yao 2001; 2003; Yu and Zhou 2006). The goal of these studies is to understand the impact of the different modules of an evolutionary algorithm and to develop new methods for their analysis. Furthermore, classical problems from combinatorial optimization such as minimum spanning trees (Neumann and Wegener 2007; 2006) and shortest paths (Scharnow, Tinnefeld, and Wegener 2004; Baswana et al. 2009; Doerr, Happ, and Klein 2007) have been considered. One cannot hope to beat the best problem-specific algorithms for these classical polynomial solvable problems, but these studies provide interesting insights into the search behavior of these algorithms and show that many classical problems are solved by general purpose algorithms such as evolutionary algorithms in expected polynomial time. Studies on classic NP-hard combinatorial optimization problems such as makespan scheduling (Witt 2005), covering problems (Friedrich et al. 2009; Oliveto, He, and Yao 2009; Friedrich et al. 2010), and multi-objective minimum spanning trees (Neumann 2007) show that these algorithms achieve good approximations for these problems in expected polynomial time. For a comprehensive presentation of the different results that have been achieved see (Neumann and Witt 2010).

A promising approach to gain further insights into the behavior of evolutionary algorithms is to study them in the context of parameterized complexity (Downey and Fellows 1999). This approach allows one to analyze the runtime of evolutionary algorithms in dependence of the structure of a problem instance. The parameterized analysis of evolutionary algorithms has been started only recently. Results have been obtained for the vertex cover problem (Kratsch and Neumann 2009) and the problem of computing a spanning tree with a maximal number of leaves (Kratsch et al. 2010).

Our Contribution

We consider the traveling salesperson problem (TSP) which is one of the most famous NP-hard combinatorial optimization problems and analyze the influence of TSP problem structure on solubility by evolutionary algorithms. We present a parameterized analysis of simple evolutionary algorithms on the Euclidean TSP by investigating the runtime complexity of simple EAs in dependence of the number of interior points that an instance has. The number of interior points is computed by taking the convex hull given by the points of the input instance and counting the points lying within the convex region.

We show that if the points of a Euclidean TSP instance are embedded into a grid and are in convex position, both a simple randomized local search and a simple (1+1) evolutionary algorithm can solve the problem in polynomial time. We also show that if the angles between any three points are not arbitrarily small and the number of interior points is fixed by a constant k, we show that within expected time $O(n^{f(k)})$ where f depends only on k, randomized local search has found a local optimum and the (1+1)-EA has solved the TSP. This proves that the (1+1)-EA is a (randomized) XP-algorithm (Downey and Fellows 1999) for the Euclidean TSP when the requisite constraint on angles holds.

The remainder of the paper is organized as follows. We begin by introducing the Euclidean TSP and simple evolutionary algorithms tasked to solve it. We then study structural properties that facilitate the technical analysis. We analyze the runtime of simple evolutionary algorithms on points in convex position and then bound their runtime parameterized by the number of interior points. We investigate the parameterized complexity of finding locally optimal 2-opt tours and solving the TSP to optimality with a simple (1+1) evolutionary algorithm.

Simple EAs and the Euclidean TSP

Let V be a set of n points in the plane labeled as $[n] = \{1, \ldots, n\}$ such that no three points are collinear. We consider the complete, weighted Euclidean graph $G(V, E)$ where E is the set of all 2-sets from V. The weight of an edge $\{u, v\} \in E$ is equal to $d(u, v)$: the Euclidean distance separating the points. The goal is to find a set of n edges of minimum weight that form a Hamiltonian cycle in G. A candidate solution of the TSP is a permutation x of V which we consider as a sequence of distinct elements $x = (x_1, x_2, \ldots, x_n)$, such that $x_i \in [n]$. The Hamiltonian cycle in G induced by such a permutation is the set of n edges

$$C(x) = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{n-1}, x_n\}, \{x_n, x_1\}\}.$$

The optimization problem is to find a permutation x which minimizes the fitness function

$$f(x) = \sum_{\{u,v\} \in C(x)} d(u, v). \tag{1}$$

The *inversion* operator is closely related to the well-known 2-change (or 2-opt) operation for TSP. A permutation x is transformed into a permutation $\sigma_{ij}[x]$ by inverting the subsequence in x from position i to position j where $1 \leq i < j \leq n$. The usual effect of the inversion operator is to delete the two edges $\{x_{i-1}, x_i\}$ and $\{x_j, x_{j+1}\}$ from $C(x)$ and reconnect the tour $C(\sigma_{ij}[x])$ using edges $\{x_{i-1}, x_j\}$ and $\{x_i, x_{j+1}\}$. Here and subsequently, we consider arithmetic on the indices to be modulo n, i.e., $1-1 = n$ and $n + 1 = 1$. Since the underlying graph G is undirected, when $(i, j) = (1, n)$, the operator has no effect since the current tour is only reversed. There is also no effect when $(i, j) \in \{(2, n), (1, n-1)\}$. In this case, it is straightforward to check that the edges removed from $C(x)$ are equal to the edges replaced to create $C(\sigma_{ij}[x])$.

Many randomized search heuristics such as evolutionary algorithms applied to the TSP operate by iteratively generating successive permutations using applications of the inversion operator. Such an algorithm starts from a random initial permutation $x^{(1)}$ and generates successive permutations $x^{(t+1)}$ that attempt to improve upon $x^{(t)}$. The general form of a simple evolutionary algorithm is as follows.

$x \leftarrow$ a random permutation of $[n]$.
repeat forever
    $y \leftarrow \textsc{Mutate}(x)$
    if $f(y) < f(x)$ **then** $x \leftarrow y$

Note, that in practice a stopping criteria is required. For our theoretical investigations, we consider the infinite stochastic process $(x^{(1)}, x^{(2)}, x^{(3)}, \ldots)$ where $x^{(t)}$ equals the permutation x after the t-th step of the algorithm. We are interested in the expected value of t such that $x^{(t)}$ is for the first time a candidate solution of interest (for example, an optimal solution). We call this the expected time to reach the desired goal.

In this paper, we will analyze two algorithms called *randomized local search* (RLS) and *(1+1) evolutionary algorithms* ((1+1)-EA) which are commonly studied in the computational complexity analysis of evolutionary algorithms (see e.g. (Droste, Jansen, and Wegener 2002; Neumann and Witt 2010). In the case of the TSP, a natural choice for the mutation operator is based on a *random inversion* operation. A random inversion of a permutation x is a permutation obtained from applying the inversion operator $\sigma_{ij}[x]$ where $\{i, j\}$ is selected uniformly at random from the set of $\binom{n}{2}$ distinct 2-subsets of $[n]$. RLS and the (1+1)-EA are both characterized by the above pseudocode but differ in implementation of the $\textsc{Mutate}$ procedure. In RLS, the mutation step $\textsc{Mutate}(x)$ is defined by performing a single random inversion $\sigma_{ij}[x]$. In the (1+1)-EA, the mutation step $\textsc{Mutate}(x)$ is defined by performing $r + 1$ random inversions where r is drawn from a Poisson distribution with parameter $\lambda = 1$. The motivation for this kind of operation is that the Poisson distribution with parameter 1 is the limiting case to the binomial distribution with probability $1/n$. Therefore, for n sufficiently large, the number of distinct inversion operations in each mutation is approximately distributed the same as the number of distinct bit-flip operations in traditional uniform bitstring mutation.

Structural Properties

In the following, we show some structural properties that will later be used for the runtime analysis of the algorithms. Geometrically, it will often be convenient to consider an edge $\{u, v\}$ as the unique planar line segment with end points u and v. We say a pair of edges $\{u, v\}$ and $\{s, t\}$ *intersect* if they cross at a point in the Euclidean plane. An important observation, which we state here without proof, is that any pair of intersecting edges form the diagonals of a convex quadrilateral in the plane.

Proposition 1. *If $\{u, v\}$ and $\{s, t\}$ intersect at a point p, they form the diagonals of a convex quadrilateral described by points u, s, v, and t. Hence edges $\{s, u\}$, $\{s, v\}$, $\{t, v\}$ and $\{t, u\}$ form a set of edges that mutually do not intersect.*

We say the tour $C(x)$ is *intersection-free* if it contains no pairs of edges that intersect. If a tour is not intersection-free, an intersection can always be removed by an inversion. This notion is captured by the following lemma.

Lemma 1. *Let x be a permutation such that $C(x)$ is not intersection-free. Then there exists an inversion that removes* *a pair of intersecting edges and replaces them with a pair of non-intersecting edges.*

Proof. Suppose $\{x_{i-1}, x_i\}$ and $\{x_j, x_{j+1}\}$ intersect in $C(x)$. Let $y = \sigma_{ij}[x]$. Then

$$C(x) \setminus C(y) = \{\{x_{i-1}, x_i\}, \{x_j, x_{j+1}\}\}, \quad \text{and}$$
$$C(y) \setminus C(x) = \{\{x_{i-1}, x_j\}, \{x_i, x_{j+1}\}\}.$$

By Proposition 1, since $\{x_{i-1}, x_i\}$ and $\{x_j, x_{j+1}\}$ intersect, the two new edges introduced to $C(y)$ by $\sigma_{ij}[\cdot]$ do not intersect. Note that it is still possible that the introduced edges intersect with some of the remaining edges in $C(y)$. □

Angle-bounded point sets

A challenge to the the runtime analysis of algorithms that employ edge exchange operations such as 2-opt is that, when points are allowed in arbitrary positions, the minimum change in fitness between neighboring solutions can be made arbitrarily small. Indeed, proof techniques for worst-case analysis often leverage this fact (Englert, Röglin, and Vöcking 2007). To circumvent this, we impose bounds on the angles between points, which allows us to express runtime results as a function of trigonometric expressions involving these bounds. Momentarily, we will refine this further by introducing a class of TSP instances embedded in an $m \times m$ grid. In that case, we will see that the resulting trigonometric expression is bounded by a polynomial in m.

We say V is *angle-bounded by* $\epsilon > 0$ if for any three points $u, v, w \in V$, $0 < \epsilon < \theta < \pi - \epsilon$ where θ denotes the angle formed by the line from u to v and the line from v to w. This allows us to express a bound in terms of ϵ on the change in fitness from a move that removes an inversion.

Lemma 2. *Suppose V is angle-bounded by ϵ. Let x be a permutation such that $C(x)$ is not intersection-free. Let $y = \sigma_{ij}[x]$ be the permutation constructed from an inversion on x that replaces two intersecting edges in $C(x)$ with two non-intersecting edges.*[1] *Then, if d_{min} denotes the minimum distance between any two points in V, $f(x) - f(y) > 2d_{min}\left(\frac{1-\cos(\epsilon)}{\cos(\epsilon)}\right)$.*

Proof. The inversion σ_{ij} removes intersecting edges $\{u, v\}$ and $\{s, t\}$ from $C(x)$ and replaces them with the pair $\{s, u\}$ and $\{t, v\}$ to form $C(y)$. We label the point at which the original edges intersect as p.

Denote as θ_u and θ_v the angles between the line segments that join at each point u and v, respectively. Since all angles are strictly positive, the points u, s, and p form a nondegenerate triangle with angles θ_s, θ_u, and $(\pi - (\theta_s + \theta_u))$. By the law of sines we have

$$\frac{d(s,u)}{\sin(\pi - (\theta_s + \theta_u))} = \frac{d(s,u)}{\sin(\theta_s + \theta_u)} = \frac{d(u,p)}{\sin(\theta_s)} = \frac{d(s,p)}{\sin(\theta_u)}.$$

Hence,

$$d(u,p) + d(s,p) = d(s,u)\left(\frac{\sin(\theta_s) + \sin(\theta_u)}{\sin(\theta_s + \theta_u)}\right). \quad (2)$$

[1] Lemma 1 guarantees the existence of such an inversion.

Since u, s, and p form a triangle, $0 < (\theta_s + \theta_u) < \pi$ and we have $0 < \sin(\theta_s) < 1$ (since $0 < \theta_s < \pi$), $0 < \sin(\theta_u) < 1$ (since $0 < \theta_u < \pi$), and $0 < \sin(\theta_s + \theta_u) < 1$ (since $0 < \theta_s + \theta_u < \pi$).

Furthermore, since V is angle-bounded by $0 < \epsilon < \pi - \epsilon$, by (2),

$$d(u,p) + d(s,p) > d(s,u)\left(\frac{\sin(\epsilon) + \sin(\epsilon)}{\sin(\epsilon + \epsilon)}\right) > d(s,u). \quad (3)$$

Since there is also a nondegenerate triangle formed by the points t, v, and p, a symmetric argument holds and thus

$$d(t,p) + d(v,p) > d(t,v)\left(\frac{\sin(\epsilon) + \sin(\epsilon)}{\sin(\epsilon + \epsilon)}\right) > d(t,v). \quad (4)$$

Combining Equations (3) and (4) we have

$$\begin{aligned} &f(x) - f(y) = [d(u,v) + d(s,t)] - [d(t,v) + d(s,u)] \\ &= d(u,p) + d(v,p) + d(t,p) + d(s,p) - [d(t,v) + d(s,u)] \\ &> [d(t,v) + d(s,u)]\left(\frac{2\sin(\epsilon)}{\sin(2\epsilon)}\right) - [d(t,v) + d(s,u)] > 0 \end{aligned}$$

The constraint that the difference is strictly positive follows directly from Equations (3) and (4). Hence,

$$\begin{aligned} f(x) - f(y) &> [d(t,v) + d(s,u)]\left(\frac{2\sin(\epsilon)}{\sin(2\epsilon)} - 1\right) \\ &\geq 2d_{min}\left(\frac{2\sin(\epsilon)}{\sin(2\epsilon)} - 1\right) = 2d_{min}\left(\frac{1-\cos(\epsilon)}{\cos(\epsilon)}\right). \end{aligned}$$ □

Since each inversion which removes an intersection results in a permutation whose fitness is improved by the amount bounded in Lemma 2, it is now straightforward to bound the time it takes for RLS to discover a permutation that corresponds to an intersection-free tour. This time bound is expressed as a function of the angle bound ϵ.

Lemma 3. *Let V be a set of planar points angle-bounded by $\epsilon > 0$. Furthermore, suppose d_{max} and d_{min} denote the maximum and minimum distance between any two points, respectively. Then the expected time until the RLS finds an intersection-free tour is bounded by $O\left(n^3\left(\frac{d_{max}}{d_{min}} - 1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\right)$.*

Proof. Let x be an arbitrary permutation. As long as $C(x)$ is not intersection-free, by Lemma 1, there is an inversion σ_{ij} which removes a pair of intersecting edges and replaces them with a pair of non-intersecting edges. Moreover, by Lemma 2, such an inversion results in an improvement of at least $2d_{min}(1 - \cos(\epsilon)) / (\cos(\epsilon))$.

Consider an optimal solution $x^\star$. As long as $C(x)$ is not intersection-free, $f(x) > f(x^\star)$ since Lemmas 1 and 2 guarantee an improving inversion. From x, there can be at most $\frac{f(x)-f(x^\star)}{2d_{min}}\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)$ improving inversions until an intersection-free tour is found. For any permutation x, $nd_{min} \leq f(x) \leq nd_{max}$ since $|C(x)| = n$. It follows that for all permutations x, the number of inversions required to transform x into a permutation z where $C(z)$ is intersection-free is bounded above by $\frac{n(d_{max}-d_{min})}{2d_{min}}\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)$. In each step, RLS chooses an inversion that removes a pair of intersecting edges with probability at least $\binom{n}{2}^{-1}$, thus the time until an intersecting pair is removed is geometrically distributed with expectation $O(n^2)$. It follows that the total expected waiting time to reach a solution z where $C(z)$ is intersection-free is bounded above by $O\left(n^3\left(\frac{d_{max}}{d_{min}}-1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\right)$. $\square$

We can also prove a similar result for the (1+1)-EA. In this case, we can show that, though the stochastic process is infinite, the number of iterations spent on tours which are *not* intersection-free is finite in expectation. In fact, the following lemma provides a bound on this expectation.

Lemma 4. *Suppose V is angle bounded by ϵ. Let $(x^{(1)}, x^{(2)}, \ldots, x^{(t)}, \ldots)$ denote the sequence of permutations generated by the (1+1)-EA. Let α be an indicator variable defined on permutations of $[n]$ as*

$$\alpha(x) = \begin{cases} 1 & \textit{if } C(x) \textit{ contains intersections;} \\ 0 & \textit{otherwise.} \end{cases}$$

Denoting the expectation operator as $\mathbb{E}(\cdot)$, we have $\mathbb{E}\left(\sum_{t=1}^{\infty}\alpha(x^{(t)})\right) = O\left(n^3\left(\frac{d_{max}}{d_{min}}-1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\right)$.

Proof. Since there is a nonzero probability to make an improving move at any suboptimal step, the optimal solution must be reached almost surely as $t \to \infty$. Since an optimal solution is intersection-free, the expectation in the claim exists and is finite.

Consider the stochastic process $(y^{(1)}, y^{(2)}, \ldots, y^{(i)})$ which is the restriction of $(x^{(1)}, x^{(2)}, \ldots)$ constructed by taking only permutations $y^{(t)}$ that correspond to non-intersection-free tours in the order they are visited by the (1+1)-EA. It follows that $\mathbb{E}\left(\sum_{t=1}^{\infty}\alpha(x^{(t)})\right) = \mathbb{E}(i)$.

Since, for all $t \leq i$, $C(y^{(t)})$ is not intersection-free, by Lemma 2, there exists an inversion mutation that results in an offspring that improves on the fitness of $f(y^{(t)})$ by at least $2d_{min}\left(1-\cos(\epsilon)\right)/\left(\cos(\epsilon)\right)$. Hence, from $y^{(t)}$, the probability of improving the fitness by at least the above amount is greater than or equal to $(en(n-1)/2)^{-1}$: the probability that Poisson mutation selects exactly one specific inversion (i.e., one which removes an intersection).

For any arbitrary permutation x, $nd_{max} \geq f(x) \geq nd_{min}$ so there can be at most $\frac{n(d_{max}-d_{min})}{2d_{min}(1-\cos(\epsilon))/cos(\epsilon)}$ such improvements until a global optimum is reached. By the above argument, the expected number of permutations $y^{(t)}$ before such an improvement is bounded by $O(n^2)$. This yields the claimed bound. $\square$

Quantized point sets

The lower bound on the angle between any three points in V provides a constraint on how small the change in fitness between neighboring inversions can be. This lower bound is useful in the case of a *quantized* point-set. That is, when the points can be embedded on an $m \times m$ grid. Quantization, for example, occurs when the x and y coordinates of each point in the set are rounded to the nearest value in a set of m equidistant values (e.g., integers). We point out that it is still important that the quantization preserves the constraint on collinearity since collinear points violate a nonzero angle bound. We have the following lemma.

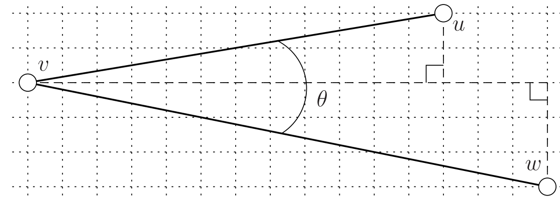

Figure 1: If the slope of the lines from v to u and u to v are of opposite sign, they form the hypotenuses of two right triangles and $\theta \geq 2\arctan((m-1)^{-1})$.

Lemma 5. *Suppose V is a set of points that lie on an $m \times m$ unit grid, no three collinear. Then V is angle-bounded by $\arctan\left(1/(2(m-2)^2)\right)$.*

Proof. The grid imposes a coordinate system on V in which the concept of line slope is well-defined. Let $u, v, w \in V$ be arbitrary points. We consider the angle θ at point v formed by the lines from v to u and v to w. Let s_1 and s_2 denote the slope of these lines, respectively. If the slopes are of opposite sign, then $\theta \geq 2\arctan((m-1)^{-1})$ since the lines form hypotenuses of two right triangles with adjacent sides of length at most $m-1$ and opposite sides with length at least 1 (see Figure 1).

We now consider the case where the slopes are nonnegative. The nonpositive case is handled identically (or by simply changing the sign of the slopes by the appropriate transformation). Without loss of generality, assume $s_1 > s_2 \geq 0$. Equality is impossible since u, v, and w cannot be collinear. Since the points lie on an $m \times m$ grid, s_1 and s_2 must be ratios of whole numbers at most $m-1$, say $s_1 = a/b$ and $s_2 = c/d$. The angle at point v is $\theta = \arctan(a/b) - \arctan(c/d) = \arctan\left(\frac{ad-cb}{bd+ac}\right)$. The minimum positive value for the expression $(ad-cb)/(bd+ac)$ over the integers from 0 to $m-1$ is $\frac{1}{2(m-2)^2}$. Since the inverse of the tangent is monotone, the minimum nonzero angle must be $\theta \geq \arctan\left(1/(2(m-2)^2)\right)$. $\square$

Lemma 5 allows us to translate the somewhat awkward trigonometric expression in the claim of Lemma 2 (and subsequent lemmas that depend on it) into a convenient polynomial that can be expressed in terms of m.

Lemma 6. *Let V be a set of n points that lie on an $m \times m$ unit grid, no three collinear. Then, V is angle-bounded by ϵ where $\cos(\epsilon)/(1-\cos(\epsilon)) = O(m^4)$.*

Proof. It follows from Lemma 5 that the angle bound on V is $\epsilon = \arctan\left(1/(2(m-2)^2)\right)$. Since $\cos(\arctan(x)) = 1/\sqrt{1+x^2}$ we have $\frac{\cos(\epsilon)}{1-\cos(\epsilon)} = \frac{2(m-2)^2}{\sqrt{1+4(m-2)^4}-2(m-2)^2}$. and since $z/(\sqrt{1+z^2}-z) = O(z^2)$, setting $z = 2(m-2)^2$ completes the proof. $\square$

Analysis for Convex Position

A finite point set V is in *convex position* when every point in V is a vertex of its convex hull. Deĭneko et al. (2006) observed that the Euclidean TSP is easy to solve when V is in convex position. In this case, the optimal permutation is any linear ordering of the points which respects the ordering of the points around the convex hull. Such an ordering can be found in time $O(n \log n)$ (de Berg et al. 2008).

In the context of evolutionary algorithms, the natural question arises, if V is in convex position, how easy is it for a simple EA? In this case, a tour is intersection-free if and only if it is globally optimal, hence finding an optimal solution is exactly as hard as finding an intersection-free tour. We can thus immediately apply the results derived in the last section to answer this question for angle-bounded point sets.

Theorem 1. *If V is in convex position and angle-bounded by ϵ, then both RLS and the (1+1)-EA solve the TSP on V in expected time $O\left(n^3\left(\frac{d_{max}}{d_{min}}-1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\right)$.*

Proof. Since V is in convex position, any intersection-free tour is globally optimal. In the case of RLS, the claim follows directly from Lemma 3.

In the case of the (1+1)-EA, let $x^{(t)}$ denote the first permutation in the stochastic process such that $C(x^{(t)})$ is intersection-free. Lemma 4 states that the expected number of permutations in $(x^{(1)}, x^{(2)}, \ldots)$ that do *not* correspond to intersection-free tours is bounded by $O\left(n^3\left(\frac{d_{max}}{d_{min}}-1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\right)$, hence the expectation of t is at most this. □

If the points are quantized in an $m \times m$ grid, we can immediately appeal to Lemmas 5 and 6 to derive a polynomial time bound on the expected runtime for both algorithms.

Theorem 2. *If V is in convex position and embedded in an $m \times m$ grid with no three collinear, then both the RLS and the (1+1)-EA solve the TSP on V in expected time $O(n^3m^5)$*

Proof. The bound follows directly from Theorem 1 since, by Lemma 5, V is angle-bounded by $\arctan\left(1/(2(m-2)^2)\right)$. In this case, $d_{max} = (m-1)\sqrt{2}$ and $d_{min} = 1$. Finally, appealing to Lemma 6, $\cos(\epsilon)/(1-\cos(\epsilon)) = O(m^4)$. Substituting these terms into the claimed bound of Theorem 1 completes the proof. □

Parameterized Analysis

Parameterized complexity theory is an extension to traditional computational complexity theory in which the analysis of hard algorithmic problems is decomposed into parameters of the problem input. This approach illuminates the relationship between hardness and different aspects of problem structure because it often isolates the source of exponential complexity in NP-hard problems.

A parameterization of a problem is a mapping of problem instances into the set of natural numbers (for a detailed description, see, e.g., (Flum and Grohe 2006)). Given a problem of size n and parameter k, a parameterized analysis is an expression of the complexity in terms of both n and k. We will express their complexity in terms of the parameterization to show their dependence on the cardinality of the interior point set.

In the case of the TSP, we are interested in expressing the runtime complexity as a function of n and k where $n = |V|$ and k is the number of vertices that lie in the interior of the convex hull of V. Note that the problem at hand is *fixed-parameter tractable* as there is a dynamic programming algorithm which solves the problem in time $f(k) \cdot p(n)$ where p is a polynomial in n (Deĭneko et al. 2006). However, it is an open question whether general evolutionary algorithms such as the ones that we consider can solve the problem within such a time bound.

We denote by $\mathfrak{H}(V) \subseteq V$ the convex hull of V. A permutation x respects hull-order if any two points in the subsequence of x induced by $\mathfrak{H}(V)$ are consecutive in x if and only if they are consecutive on the hull.

Lemma 7. *If $C(x)$ is intersection-free, then x respects hull-order.*

Proof. This follows immediately from the proof of Theorem 2 in (Quintas and Supnick 1965). □

Lemma 7 entails the following bound on the number of unique permutations that yield intersection-free tours.

Lemma 8. *Suppose $|V \setminus \mathfrak{H}(V)| = k$. Then there are at most $(n-k)^k k!$ unique intersection-free tours.*

Proof. For any set of $1 < p < n$ points, there are $p^{n-p}(n-p)!$ permutations in which the p points remain in the same order. Since $|\mathfrak{H}(V)| = (n-k)$, there are exactly $(n-k)^k k!$ permutations that respect hull-order. Since each intersection-free tour must respect hull-order, we have the claimed bound. □

We can also derive from Lemma 7 a convenient bound on the minimal number of inversions necessary to transform an intersection-free tour into a permutation that corresponds to a globally optimal tour.

Lemma 9. *Suppose $|V \setminus \mathfrak{H}(V)| = k$ and $C(x)$ is an intersection-free tour on V. Then there is a sequence of at most $2k$ inversions that transforms x into an optimal permutation.*

Proof. For any permutation x, a component in position i can be shifted into position j by at most two consecutive inversion operations. If $|i-j| = 1$, a single inversion suffices. If $|i-j| > 1$, then when $i < j$, the two operations are $\sigma_{i(j-1)}[\sigma_{ij}[x]]$. When $i > j$, the two operations are $\sigma_{(j+1)i}[\sigma_{ji}[x]]$. The relative ordering of the other components is not affected.

By Lemma 7, since $C(x)$ is intersection-free, it must be hull-respecting. Let $x^\star$ be an optimal permutation such that the elements in $\mathfrak{H}(V)$ have the same linear order in $x^\star$ as they do in x. Then x can be transformed into $x^\star$ by moving each of the k interior points into their correct position. By the above argument, each can be moved into place by at most 2 inversions and the claim is proved. □

Runtime complexity of local optima

When $\mathfrak{H}(V) \neq V$, RLS does not necessarily converge to the global optimum with probability one since it can become trapped in a local optimum. However, if the number of interior points is sparse (i.e., $O(1)$), we can use the above results to bound the complexity of finding local optima.

For RLS, a permutation x is *locally optimum* if there does not exist an inversion $y = \sigma_{ij}[x]$ such that $f(y) < f(x)$. In this case, RLS cannot make further improvements. Indeed, if no three points are collinear in V, then if x is locally optimal, $C(x)$ must be intersection-free since, if it were not, Lemma 1 guarantees an inversion exists that removes an intersection and, by the quadrangle inequality, the resulting offspring would have improving fitness. It is important to note that the converse is not necessarily true. However, we may show the following.

Lemma 10. *Suppose $C(x)$ is intersection-free. Then for any neighboring inversion $y = \sigma_{ij}[x]$ with $f(y) < f(x)$, $C(y)$ is also intersection-free.*

Proof. Suppose for contradiction that $y = \sigma_{ij}[x]$ with $f(y) < f(x)$, but $C(y)$ is not intersection-free. Since $C(x)$ was intersection-free, it follows that the pair of edges introduced by the inversion must intersect. By the quadrangle inequality, the total length of these edges must be greater than the edges they replaced, contradicting that $f(y)$ is strictly less than $f(x)$. $\square$

Theorem 3. *Suppose V is angle bounded by ϵ and that $|V \setminus \mathfrak{H}(V)| = k$. Then the expected time until RLS finds a local optimum is $O\Big(n^3 \left(\frac{d_{max}}{d_{min}} - 1\right)\left(\frac{\cos(\epsilon)}{1-\cos(\epsilon)}\right)\Big) + O\big(n^{2k}k!\big)$.*

Proof. After RLS finds an intersection-free tour, by Lemma 10, all subsequent tours will also be intersection free. The total number of intersection-free tours hence serves as a bound on the number of possible improving moves after the first intersection-free tour is encountered. By Lemma 8, this bound is $(n-k)^k k!$.

As long as RLS has not yet found a local optimum, by definition there exists an improving inversion; the expected waiting time to find such an inversion is bounded by $O(n^2)$. Thus, after the first time an intersection-free tour is encountered, the expected time until a local optimum is found is bounded by $O(n^{2k}k!)$. Adding this to the bound from Theorem 3 on the expected time to find an intersection-free tour completes the proof. $\square$

Theorem 4. *Suppose that V is quantized in an $m \times m$ grid and that $|V \setminus \mathfrak{H}(V)| = k$. Then the expected time until RLS finds a local optimum is $O\big(n^3 m^5\big) + O\big(n^{2k}k!\big)$.*

Proof. The proof is analogous to the proof of Theorem 3, substituting $d_{max} = (m-1)\sqrt{2}$ and $d_{min} = 1$, and appealing to Lemma 6 for the fact that, in this case, $\cos(\epsilon)/(1 - \cos(\epsilon)) = O(m^4)$. $\square$

Runtime complexity of the (1+1)-EA

We now turn our attention to the (1+1)-EA. In Lemma 4, we have already bounded the expected time the algorithm spends in permutations that correspond to tours that are not intersection-free. Thus it will suffice to bound the expected time the (1+1)-EA spends on non-optimal intersection-free tours and consider the total time spent in either phase. We will make use of Lemma 7 which states that every permutation that corresponds to an intersection-free tour respects hull-order.

We now use this fact to show that when there are few interior points, intersection-free tours are in somehow “close” to an optimal solution in the sense that relatively small perturbations by the (1+1)-EA suffice to solve the problem. This is easily combined with our previous stated bound on the time spent on tours with intersections to yield the following runtime result.

Theorem 5. *Let V be a set of points quantized on an $m \times m$ such that $|V \setminus \mathfrak{H}(V)| = k$. The expected time for the (1+1)-EA to solve the TSP on V is $O(n^3 m^5) + O(n^{4k}(2k-1)!)$.*

Proof. Let $(x^{(1)}, x^{(2)}, \ldots)$ denote the sequence of permutations visited by the (1+1)-EA. Let $x^\star$ denote an optimal solution to the TSP instance. Consider the following pair of indicator variables α and β defined on permutations of $[n]$ where α is defined as in Lemma 4 and

$$\beta(x) = \begin{cases} 1 & \text{if } C(x) \text{ is intersection-free and } f(x) > f(x^\star) \\ 0 & \text{if } C(x) \text{ otherwise.} \end{cases}$$

Let T be the smallest time such that $f(x^{(T)}) = f(x^\star)$. Thus T is a random variable that corresponds to the runtime of the (1+1)-EA on V. Note that $T = \sum_{t=1}^{\infty} \alpha(x^{(t)}) + \sum_{t=1}^{\infty} \beta(x^{(t)})$. By Lemmas 4 and 6, $\mathbb{E}\left(\sum_{t=1}^{\infty} \alpha(x^{(t)})\right) = O(n^3 m^5)$.

Now suppose $C(x^{(t)})$ is some intersection-free tour such that $f(x^{(t)}) > f(x^\star)$. By Lemma 9, there are at most $2k$ inversion moves that transform $x^{(t)}$ into an optimal permutation. The probability that the (1+1)-EA performs exactly ℓ inversions is Poisson distributed. Since there are $\binom{\binom{n}{2}}{\ell}\ell!$ possible unique sequences of ℓ inversions, the probability of performing a specific set of ℓ unique inversions in a particular order is at least $\frac{1}{e(\ell-1)!} \times \left(\binom{\binom{n}{2}}{\ell}\ell!\right)^{-1}$. It immediately follows that the waiting time until a specific sequence of ℓ inversions occurs is bounded by $O(n^{2\ell}(\ell-1)!)$ since $\binom{N}{\ell} \leq N^\ell/\ell!$. Setting $\ell = 2k$, the expected time to jump from an intersection-free tour to an optimal permutation is bounded above by $\mathbb{E}\left(\sum_{t=1}^{\infty} \beta(x^{(t)})\right) = O(n^{4k}(2k-1)!)$. $\square$

Conclusion

In this paper, we have studied the runtime complexity of evolutionary algorithms on the Euclidean TSP. We have carried out a parameterized analysis that studies the dependence of the hardness of a problem instance on the number of interior points in the instance. Moreover, we have shown that under reasonable geometric constraints (low angle bounds), simple evolutionary algorithms solve the convex TSP in polynomial time. Furthermore, we showed that if the instance contains k interior points, for a given, fixed angle bound, randomized local search algorithms using the inversion (2-opt) move operator find local optima in an expected $O(n^{2k}k!)$ iterations, and simple evolutionary algorithms solve the Euclidean TSP in an expected number of $O(n^{4k}(2k-1)!)$ iterations.

References

Baswana, S.; Biswas, S.; Doerr, B.; Friedrich, T.; Kurur, P. P.; and Neumann, F. 2009. Computing single source shortest paths using single-objective fitness functions. In Jansen, T.; Garibay, I.; Wiegand, R. P.; and Wu, A. S., eds., *Proceedings of the Tenth International Workshop on Foundations of Genetic Algorithms (FOGA 2009)*, 59–66. Orlando, USA: ACM Press.

de Berg, M.; Cheong, O.; van Kreveld, M.; and Overmars, M. 2008. *Computational Geometry: Algorithms and Applications*. Springer, third edition.

Deĭneko, V. G.; Hoffman, M.; Okamoto, Y.; and Woeginger, G. J. 2006. The traveling salesman problem with few inner points. *Operations Research Letters* 34:106–110.

Doerr, B.; Happ, E.; and Klein, C. 2007. A tight analysis of the (1+1)-EA for the single source shortest path problem. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, 1890–1895. IEEE Press.

Dorigo, M., and Stützle, T. 2004. *Ant Colony Optimization*. MIT Press.

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer.

Droste, S.; Jansen, T.; and Wegener, I. 2002. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science* 276:51–81.

Eiben, A., and Smith, J. 2007. *Introduction to Evolutionary Computing*. Springer.

Englert, M.; Röglin, H.; and Vöcking, B. 2007. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1295–1304. Society for Industrial and Applied Mathematics.

Flum, J., and Grohe, M. 2006. *Parameterized complexity theory*. Springer-Verlag.

Friedrich, T.; He, J.; Hebbinghaus, N.; Neumann, F.; and Witt, C. 2009. Analyses of simple hybrid evolutionary algorithms for the vertex cover problem. *Evolutionary Computation* 17(1):3–20.

Friedrich, T.; He, J.; Hebbinghaus, N.; Neumann, F.; and Witt, C. 2010. Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation* 18(4):617–633.

He, J., and Yao, X. 2001. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence* 127(1):57–85.

He, J., and Yao, X. 2003. Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence* 145(1-2):59–97.

Kratsch, S., and Neumann, F. 2009. Fixed-parameter evolutionary algorithms and the vertex cover problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '09)*, 293–300. ACM Press.

Kratsch, S.; Lehre, P. K.; Neumann, F.; and Oliveto, P. S. 2010. Fixed parameter evolutionary algorithms and maximum leaf spanning trees: A matter of mutation. In Schaefer, R.; Cotta, C.; Kolodziej, J.; and Rudolph, G., eds., *Proceedings of the Eleventh Conference on Parallel Problem Solving from Nature*, volume 6238 of *Lecture Notes in Computer Science*, 204–213. Springer.

Neumann, F., and Wegener, I. 2006. Minimum spanning trees made easier via multi-objective optimization. *Natural Computing* 5(3):305–319.

Neumann, F., and Wegener, I. 2007. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science* 378(1):32–40.

Neumann, F., and Witt, C. 2010. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer.

Neumann, F. 2007. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research* 181(3):1620–1629.

Oliveto, P. S.; He, J.; and Yao, X. 2009. Analysis of the (1+1) -ea for finding approximate solutions to vertex cover problems. *IEEE Transactions on Evolutionary Computation* 13(5):1006–1029.

Quintas, L. V., and Supnick, F. 1965. On some properties of shortest hamiltonian circuits. *The American Mathematical Monthly* 72(9):977–980.

Scharnow, J.; Tinnefeld, K.; and Wegener, I. 2004. The analysis of evolutionary algorithms on sorting and shortest paths problems. *Journal of Mathematical Modelling and Algorithms* 349–366.

Witt, C. 2005. Worst-case and average-case approximations by simple randomized search heuristics. In *Proceedings of the Twenty-Second Symposium on Theoretical Aspects of Computer Science (STACS '05)*, volume 3404 of *Lecture Notes on Computer Science*, 44–56. Springer.

Yu, Y., and Zhou, Z.-H. 2006. A new approach to estimating the expected first hitting time of evolutionary algorithms. In *Proceedings of the Twenty-First Conference on Artificial Intelligence (AAAI 2006)*. AAAI Press.