UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

Statistical Mechanics and Optimization in Complex Scenarios

Mecânica Estatística e Otimização em Cenários Complexos

Eliseu Venites Filho

Dissertation presented in partial fulfillment of the requirements for the degree of Master of Physics

Advisor: Prof. Dr. Roberto da Silva

Porto Alegre
May, 2021

Abstract

Combinatorial optimization problems, such as the process of searching for extrema of a function over a discrete domain, are ubiquitous in science and engineering. Despite its ubiquity, some of these problems are notably difficult, requiring a computational cost which exponentially scales with the number of inputs. Among the vast collection of combinatorial optimization problems, the traveling salesman problem (TSP) has been of particular importance because of its huge number of applications. A common approach to such hard problems is the use of heuristics such as simulated annealing. Many versions of this heuristic are explored in the literature, but so far the effects of the statistical distribution of the coordinates of the cities on the performance of the heuristic has been neglected. We propose a simple way to explore this aspect by analyzing the performance of a standard version of simulated annealing (one using the geometrical cooling schedule) in correlated systems with a simple and useful method based on a linear combination of independent random variables. Our results suggest that performance depends on the shape of the statistical distribution of the coordinates but not necessarily on its variance corroborated by the cases of uniform and normal distributions. On the other hand, a study with different power laws (different decay exponents) for the coordinates always produces different performances. We show that the performance of the simulated annealing, even in its best version, is not improved when the distribution's first moment diverges. Surprisingly, however, we still obtain improvements when the first moment exists but the second moment diverges. Finite size scaling, fits, and universal laws support all of our results. In addition our study shows when the cost must be scaled.

Keywords: combinatorial optimization; traveling salesman problem;

heuristics; simulated annealing; monte carlo;

Resumo

Problemas de otimização combinatorial, com problemas envolvendo encontrar pontos extremos de uma função sobre um domínio contínuo, são onipresentes em ciência e engenharia. Apesar de sua onipresença, alguns desses problemas são particularmente difíceis, exigindo um custo computacional que aumenta exponencialmente com o número de entradas. Dentre a vasta coleção de problemas de otimização combinatória, o problema do caixeiro viajante (TSP) tem sido de especial importância devido a seu grande número de aplicações. Uma abordagem comum para tais problemas é a utilização de heurísticas, como o recozimento simulado. Muitas versões dessa heurística são exploradas na literatura, mas até então o efeito da distribuição das coordenadas no desempenho da heurística tem sido preterido. Neste trabalho propomos uma maneira simples de explorar esse aspecto analisando o desempenho de uma versão padrão do recozimento simulado (utilizando o cronograma de resfriamento geométrico) em sistemas correlacionados com um método simples baseado em combinações lineares de variáveis aleatórias independentes. Nossos resultados sugerem que o desempenho depende fortemente do formato da distribuição e independe de sua variância, o que foi verificado utilizando distribuições uniformes e normais. Entretanto, um estudo considerando diferentes leis de potência (diferentes expoentes de decaimento) para as coordenadas resulta em desempenhos diferentes. Mostramos que mesmo para a melhor versão do recozimento simulado estudada, o recozimento simulado não é capaz de encontrar um ciclo satisfatório quando a distribuição de coordenadas não têm o primeiro momento definido. Porém, surpreendentemente, observamos melhoras mesmo quando a distribuição tem seu segundo momento não definido. Análises de tamanho finito, ajustes e leis universais corroboram nossos resultados. Ademais, nossa análise mostra quando o custo deve ser escalado.

Palavras chave: otimização combinatorial; problema do caixeiro viajante; heurísticas; recozimento simulado; monte carlo;

Acknowledgments

This work was financed by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq-Brazil), process number 130910/2019-8.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

When reasoning about computational problems it is important to consider not only if a given problem is solvable but also how much computational resources need to be employed. In order to answer this question the field of computational complexity classifies algorithms based on how fast their time and space requirements increase with input size for the worst case scenario. The time complexity of an algorithm refers to the number of elementary steps taken to solve the problem. For our purposes the most important complexity classes are P and NP, they can be informally defined as follows: [1, Chapter 1]

- Polynomial time (P): A problem belongs to the class P if there exists an algorithm for finding the solution in a computational time that grows at most as a polynomial function of the input size.
- Non-deterministic polynomial time (NP): A problem belongs to the class NP if given a candidate solution there is an algorithm in P for checking if the candidate solution really solves the problem.

Algorithms that solve problems from P in polynomial time are referred to as efficient algorithms. A problem outside P is a problem that is theoretically solvable given a long computation time, although it can potentially become impractical as the instance size grows. It is clear that P is a subset of NP, that is P $\subseteq$ NP, since if we have a way of efficiently solving a problem, then the checking of a candidate solution can be done by simply solving the problem and comparing the computed solution to the candidate one. The question of whether every problem in NP is also in P remains open.

Moreover, another important classification is that of NP-hardness and NP-completeness. A problem belongs to the complexity class NP-Hard if it is at least as hard as any NP problem, i.e. any problem in NP can be reduced to it in polynomial time. Furthermore a NP problem that is also NP-Hard is classified as NP-Complete. Figure 1.1 illustrates how the complexity classes relate to each other.

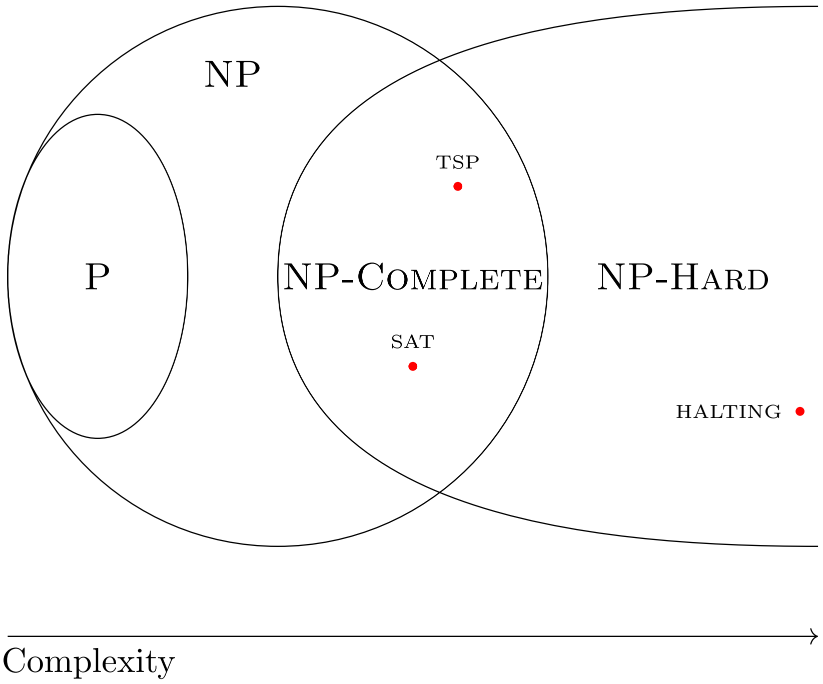


Figure 1.1: Euler diagram of the complexity classes assuming that P $\neq$ NP. Note that NP-Complete $\subset$ NP-Hard.

In the early 1970's it was shown through the so-called Cook-Levin theorem [2] that the problem of *boolean satisfiability* (SAT) is NP-Complete, therefore one could assert that a given problem is also NP-Complete by reducing it to SAT. Later, Karp did just that for several well-known problems, among them the *Traveling Salesman Problem* (TSP), the subject of this work.

For the sake of completeness, we note that the complexity classes described above are used to classify decision problems; nevertheless, there is a relationship between a decision problem and its optimization counterpart allowing for the latter to be solved in polynomial time given that the former can be shown to be decided in polynomial time.

In order to show the importance of an efficient algorithm, consider the problem of sorting a list of numbers, a very routine problem for which a number of efficient algorithms are known, among them the *quicksort*. This particular algorithm has an average running time of $\mathcal{O}(n \log n)$ although it may take as many as $\mathcal{O}(n^2)$ for the worst case scenario, where n is the list size. [3, Chapter 7]

If no such algorithm existed, the only available option would be a brute-force search over the whole set of possible permutations and because there are $n!$ of them for a list of size n the running time of the search would be $\mathcal{O}(n!)$.

Assuming that a typical personal computer has a CPU processor clock of the order of GHz and assuming one operation per clock cycle, in Table 1.1 we estimate how long it would take to sort a list of a given size n using the efficient quicksort algorithm versus searching through the whole set of permutations.

Table 1.1: Estimated running time comparison of an efficient $\mathcal{O}(n^2)$ algorithm versus an inefficient $\mathcal{O}(n!)$ one as function of input size n.

Complexity	$n = 4$	$n = 16$	$n = 64$
$\mathcal{O}(n^2)$	16 ns	256 ns	4.096 µs
$\mathcal{O}(n!)$	24 ns	5 h 48 min 42 s	2.9×10^{62} the age of the universe

1.1 Nature inspired heuristics

Traditionally, computational complexity considers the worst case scenario when analyzing a proposed algorithm for a given problem. If the proposed algorithm solves any instance of a given problem at most in polynomial time, then the problem is classified as belonging to P. Sometimes, however, it is enough to realize that for some problems even if we cannot guarantee a solution for the worst instances in polynomial time, we nonetheless may be able to solve some useful instances efficiently.

The classification of problems according to complexity classes and the identification of problems which seem impossible to be efficiently solved for all instances motivated the search for alternative methods with more relaxed mathematical constraints than the usual definition of an algorithm. In this context, the concept of an heuristic come into play.

It is important to clearly differentiate between an algorithm and a heuristic in the context of combinatorial optimization [4]. An algorithm is a well defined procedure that solves a given problem exactly identifying all solutions, including the optimal one, with the certainty of a mathematical proof. The algorithm always works for the whole class of problems for which it was designed.

On the other hand, a heuristic is an approach that yields good solutions for certain instances of the problem, but does not guarantee that the solution is optimal, neither specifies how close to optimal the solution found is. Arguments for its performance are usually justified empirically or by analogies but, crucially, no mathematical proofs are provided. Furthermore, the evaluation of heuristics is also done empirically.

A large number of heuristics are conceived through careful examination of physical systems. When dealing with combinatorial optimization problems, it is natural to draw connections to the field of statistical mechanics. The parallels are plenty and are summarized in table 1.2.

Each problem can be thought of as a system in statistical mechanical sense while the cost function we wish to minimize is described by the total energy of the system, its Hamiltonian. Moreover, an instance of a given problem corresponds to a sample of a given system and the specification of a candidate configuration for the instance is equivalent to the specification of a microstate for the system.

Table 1.2: Similarities between the frameworks of statistical physics and combinatorial optimization.

Statistical Physics	Combinatorial Optimization
System	Problem
Hamiltonian	Cost Function
Sample	Instance
System Size	Instance Size
Microstate	Configuration

In the mid 80's a deep connection between the statistical mechanics of systems with many degrees of freedom, which can reach equilibrium by means of the Metropolis prescription, and combinatorial optimization problems considering functions with complex landscapes was exposed independently by [5] and [6]. The former applied the method first in the context of circuit design, revealing further analogies between this specific problem and the physics of spin glasses, and then to the TSP. Since then, the *simulated annealing* heuristic has been applied to several combinatorial problems including the study of the thermodynamics of protein folding [7] and the evaluation of periodic orbits of multi-electron atomic systems [8], among others.

About being inspired by natural mechanisms to solve computational problems, Černy noted at the very end of his paper [6]:

"We believe that this is caused by the fact that our algorithm simulates what Nature does in looking for the equilibrium of complex systems. And Nature often does its job quite efficiently."

The goal of this work is not to test different simulated annealing heuristics or even consider several cooling schedules, but rather to analyze the performance of a given variation of simulated annealing when applied to purposely crafted artificial instances of the notoriously difficult and ubiquitous Traveling Salesman Problem (TSP), which concerns itself in finding the best Hamiltonian cycle in a complete graph, i.e., the cycle that includes all cities without repetition, with minimal cost. We focus first on investigating the performance of this heuristic for artificial instances consisting of two-dimensional points whose city coordinates are correlated and long-tailed distributed. Our work explores the transition between the TSP for points randomly distributed in a two-dimensional domain (NP-Complete problem), when the coordinates are uncorrelated, and sorting a list of numbers (P problem) when coordinates are perfectly correlated.

We begin in chapter 2 by presenting the problem at hand, the Traveling Salesman Problem. The simulated annealing heuristic is introduced in chapter 3. Chapter 4 describes the random environments considered in our analysis along with the methods used in order to generate them. The results of the simulations carried out are compiled in chapter 5, beginning with a calibration of the method and then our main contributions. Finally, in chapter 6, general conclusions are drawn.

Chapter 2

The traveling salesman problem

Simply stated, the *Traveling Salesman Problem* (TSP) asks what is the shortest tour that visits every city once and then returns to the point of departure, given a set of cities and the pairwise distance between them. The namesake of the problem are 18th and early 19th century salesmen that traveled the country stopping at several cities to gather orders for which an efficient itinerary was important for minimizing the time en route.

Despite its simplicity, the TSP is of immense theoretical importance, belonging to the NP-Complete complexity class. It has been shown to be really challenging and as a result it is regarded as one of the most intensively investigated problems in combinatorial optimization. From a more practical standpoint, apart from the obvious logistics applications, the problem has piqued the interest of researchers from a myriad of areas ranging from biology to microchip fabrication.

2.1 Brief history of the problem

The precise origin of the TSP is somewhat unclear, given its ubiquity it is thought that similar problems have been informally discussed by some mathematicians for many years before its modern formulation [9]. It is believed that the earliest promotion of the problem inside the mainstream mathematical community is due to the Austrian mathematician and economist Karl Menger, who as early as 1930 presented the problem in its modern formulation. It was also Menger that first noted that a brute-force search would be impractical for relatively small instances. A few years later, in 1934, a seminar talk by Hassler Whitney at Princeton University is credited as being responsible for spreading the problem even further [10, Chapters 1 and 2].

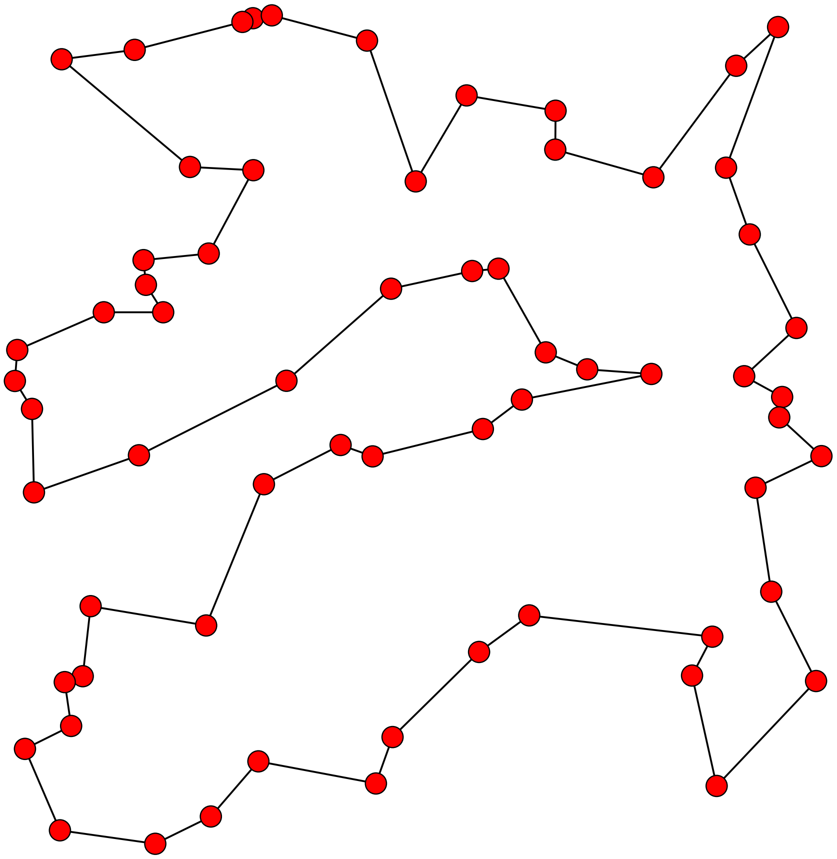


Figure 2.1: Illustrative example of relatively small instance of the TSP containing $N = 64$ points.

2.2 Mathematical formulation

A more mathematically precise formulation of the Symmetric Traveling Salesman Problem, henceforth referred to simply as TSP is as follows: a set of N points $P = \{p_i\}_{i=0}^{N-1}$ and the pairwise distance between them described by the symmetric matrix D where its element $d(i, j)$ is the distance between the points p_i and p_j are given. The cost of a cycle $\sigma = (\sigma(0), \sigma(1), \cdots, \sigma(N-1))$ is defined to be

$$\mathcal{H}(\sigma) = \sum_{k=0}^{N-1} d\left(\sigma(k), \sigma(k+1)\right) \tag{2.1}$$

with the condition for a cycle being that $\sigma(N) = \sigma(0)$.

Since we have the choice of starting point and the orientation in which the tour is traversed, the set of states of the system Γ is smaller than the set of all possible permutations S_N. This can be thought of as an equivalence class between the cycles of same cost. For example, considering a $N = 4$ point instance, the state $\sigma = (0, 1, 2, 3)$ is equivalent to $2N = 8$ permutations:

$$(0, 1, 2, 3) \sim \left\{ \begin{array}{c} (0, 1, 2, 3), (1, 2, 3, 0), (2, 3, 0, 1), (3, 0, 1, 2), \\ (3, 2, 1, 0), (2, 1, 0, 3), (1, 0, 3, 2), (0, 3, 2, 1) \end{array} \right\} \tag{2.2}$$

Hence, the total number of states to be considered is:

$$|\Gamma| = \frac{|S_N|}{2N} = \frac{N!}{2N} = \frac{(N-1)!}{2} \tag{2.3}$$

The goal is therefore to determine the best cycle passing through all points in P, i.e. the cycle of minimal cost σ^* among the set Γ of all possible cycles.

$$\sigma^* = \arg\min_{\sigma \in \Gamma} \mathcal{H}(\sigma) \tag{2.4}$$

This is still an extremely large number even for a modest number of points N. The running time of a brute-force search algorithm through the whole space of all the $\frac{(N-1)!}{2}$ possible cycles follows closely that of the second row of Table 1.1.

2.3 Overview of exact methods

In a landmark paper [9], a relatively large instance (at the time) of the TSP was solved exactly and the optimality of a tour was shown exactly using ideas from linear programming, what culminated in a method know today as the *cutting plane* algorithm.

Nowadays, one of the best known methods for solving the TSP exactly is the *branch & cut* algorithm which is a combination of the cutting plane algorithm and the branch & bound algorithm [11]. The branch & bound algorithm, widely used for combinatorial optimization problems consists in recursively dividing the search space in domains and ignoring the *branches* containing only domains that the algorithm can determine to not contain the optimal solution [12].

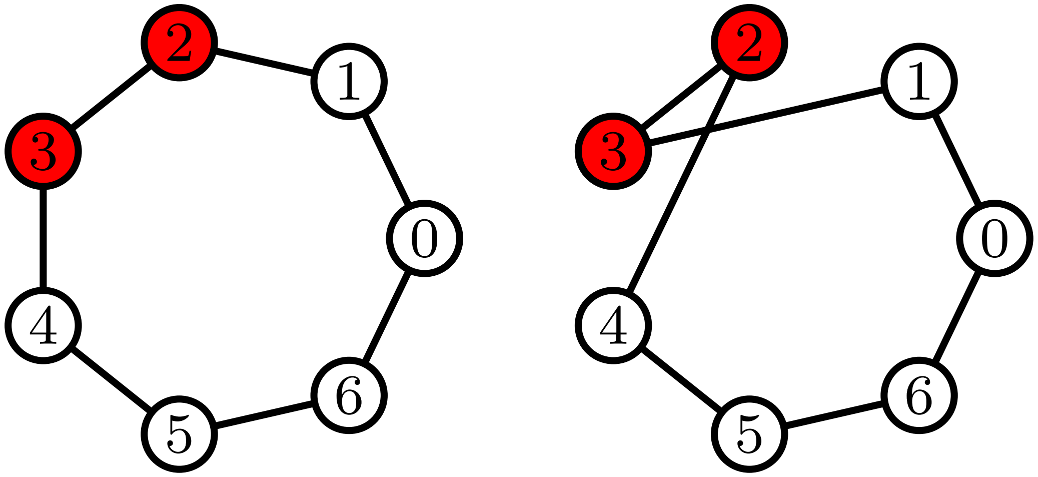
For our purposes, however, it is worth noting the following developments since they provide insights that will be useful for us later. The *nearest neighbor* (NN) [13] constructive heuristic corresponding to a particular case of the tourist random walk [14, 15] is a very intuitive process, which consists in selecting a starting point and then measuring the distance from all other points to the selected one and finally selecting the closest one. This process is repeated until all points are visited and the cycle is closed. This approach, nonetheless, in general does not yield the optimal tour: because of frustration, it is not possible to minimize all local energies at same time.

Apart from constructive methods, there are also *local search* methods, that start with a given cycle and modify it iteratively in some way looking for cost improvements. In a local search, move heuristics are employed in order to generate a new cycle from a given one. Each move heuristic defines a notion of adjacency between the set of cycles [16].

A very basic move for iterative improvement is that of a *swap*, in which two neighboring nodes have their positions within the cycle swapped. One of the advantages of this move is that the cost difference between the resulting cycle and the previous one is easily calculated by subtracting the sum of the weights of the removed edges from the sum of the weights of the new edges. It is independent of the size of the cycle.

In the example of figure 2.2, the neighboring nodes 2 and 3 are selected and have their positions swapped generating the candidate state $\sigma' = (0, 1, 3, 2, 4, 5, 6)$ from the initial state $\sigma = (0, 1, 2, 3, 4, 5, 6)$. The cost difference between the states can be easily calculated by subtracting the cost of the removed edges ($(1, 2)$ and $(2, 3)$) from the cost of the newly added edges ($(1, 3)$ and $(2, 4)$), yielding $\Delta\mathcal{H} = \mathcal{H}(\sigma') - \mathcal{H}(\sigma) = d(1, 3) + d(2, 4) - d(1, 2) - d(2, 3)$

There exists a more interesting choice of move that is commonly employed in iterative improvement. This mechanism makes use of the concept of n-optimality [17]: a given cycle is said to be n-optimal (n-opt) if it cannot be improved by replacing a set of n edges by another set of n edges. A given cycle through N points is optimal if it is an N-opt cycle, meaning that no other configuration of edges can improve its cost. The simplest of the n-opt moves is the 2-opt, it was first employed by [18] albeit without naming it. It consists in selecting two edges of the cycle and substituting them with two others.



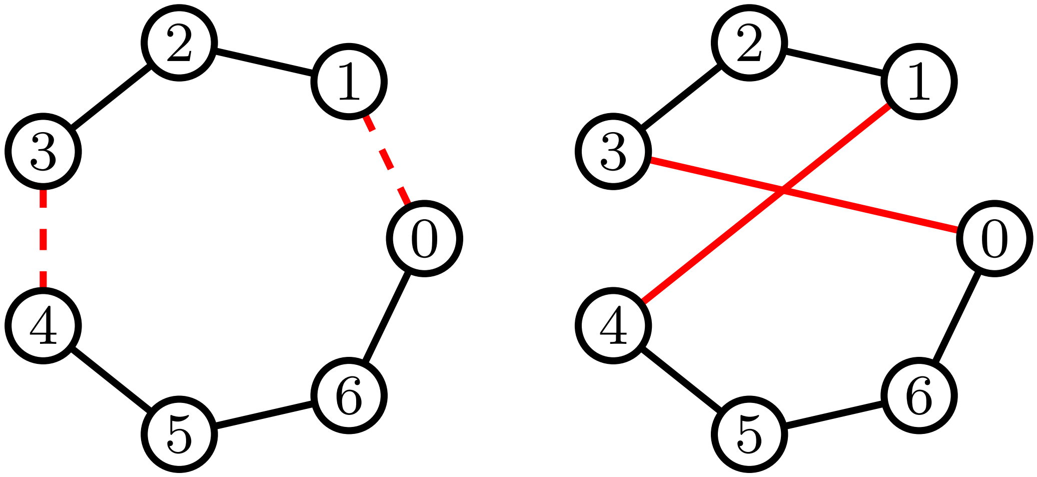
(a) Initial state with selected nodes highlighted.

(b) Resulting state. Highlighted nodes had their orders swapped.

Figure 2.2: Example of a swap move performed on a small instance of size $N = 7$.

Once the two edges are selected there is only one possible resulting cycle. This operation is equivalent to removing the selected edges, reversing the subsequence in between them and then reconnecting the cycle back together. Once again, the cost difference calculation is highly efficient and independent of the size of the system and consists in subtracting the cost of the removed edges from the newly added ones.

Starting from the initial state $\sigma = (0, 1, 2, 3, 4, 5, 6)$, in figure 2.3, the edges $(0, 1)$ and $(3, 4)$ are removed and the edges $(0, 3)$ and $(1, 4)$ are added, resulting in the candidate state $\sigma' = (0, 3, 2, 1, 4, 5, 6)$. The cost difference between these configurations is $\Delta\mathcal{H} = d(1, 4) + d(0, 3) - d(0, 1) - d(3, 4)$.



(a) Initial state with selected edges highlighted.

(b) Resulting cycle with new edges highlighted.

Figure 2.3: Example of a 2-opt move performed on a small instance of size $N = 7$.

Chapter 3

Simulated annealing heuristic

Given the classification of the Traveling Salesman Problem as a hard problem, the utilization of heuristics for its solution is alluring. One such physically motivated heuristic is simulated annealing [5, 6, 19], in which a random walk in the configuration state of the system is performed until it reaches global equilibrium or, at worst, an interesting local minimum.

This heuristics employs the Metropolis algorithm [20, 21, 22] to sample the Gibbs ensemble of the system at decreasing temperatures. The Metropolis algorithm allows us to simulate a random variable under a probability distribution using an important concept known as *Markov Chains*.

3.1 Markov chains

In the study of stochastic processes, Markov chains are of the utmost importance providing the basis for several stochastic models of physical systems. In this work we concern ourselves only with discrete time stochastic processes over a discrete state space and the definitions follow [23].

A discrete time Markov chain (from now on simply referred to as Markov chain) is a memoryless discrete time stochastic process, meaning the probabilities for the next event do not depend on the entire history of the process but rather only on the last state. More precisely, let the process $\{X_t\}_{t\in\mathbb{N}}$ be a discrete time stochastic process over the discrete state space Γ. Then this stochastic process is a Markov chain if

$$\mathbb{P}(X_{t+1} = y | X_0 = x_0, X_1 = x_1, ..., X_t = x_t) = \mathbb{P}(X_{t+1} = y | X_t = x_t)$$
$$\forall x_0, x_1, ..., x_t, y \in \Gamma$$

Additionally, if this probability is dependent only on the departing and arrival state and not on the time step of the process then the Markov chain is classified as homogeneous and is completely defined by its *stochastic matrix*. The stochastic matrix is a square matrix P whose elements are the transition probabilities from one state to another and is given by

$$[P]_{x,y} = P(x \to y) = \mathbb{P}(X_{t+1} = y | X_t = x), \quad \forall t \in \mathbb{N} \tag{3.1}$$

Naturally, it has the property that its rows each sum to 1:

$$\sum_{y \in \Gamma} P(x \to y) = 1, \quad \forall x \in \Gamma \tag{3.2}$$

The product of a stochastic matrix P by a distribution μ over the state space Γ results in another distribution ν over Γ:

$$\nu(y) = \mu P(y) = \sum_{x \in \Gamma} \mu(x) P(x \to y), \quad \forall y \in \Gamma \tag{3.3}$$

This way, the transition matrix completely defines the dynamics of the stochastic process:

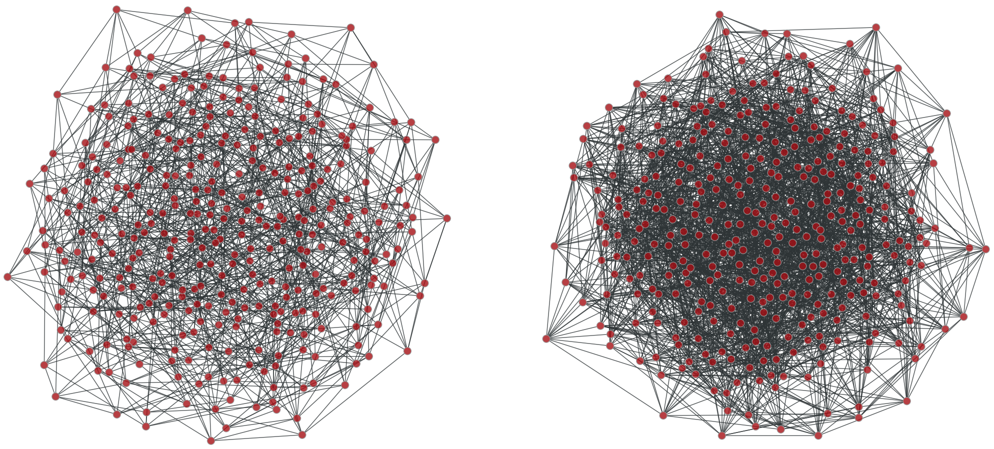
$$\mu_t(y) = \mu_{t-1} P(y) = \sum_{x \in \Gamma} \mu_{t-1}(x) P(x \to y) \quad \forall y \in \Gamma, \tag{3.4}$$

where μ_t is the distribution followed by the system at time t.

Hence, if the initial state of a homogeneous Markov chain X_0 is distributed according to the distribution μ_0 over Γ, then after t time steps, X_t will be distributed according to the distribution μ_t given by

$$\mu_t(x) = \mu_0 P^t(x) \quad \forall x \in \Gamma. \tag{3.5}$$

The transition matrix defines a *transition graph* over the state space Γ. Figure 3.1 illustrates the Markov chain transition graph defined by both the swap and 2-opt moves over the state space of an example TSP over 7 points. We note that the degree of each node (cycle state) for the 2-opt move is much



(a) Swap move: Each state is adjacent to 7 others.

(b) 2-opt move: Each state is adjacent to 14 others.

Figure 3.1: Example of a Markov chain transition graph defined by each move for a small scale example containing 7 points. Each of the $|\Gamma| = \frac{(7-1)!}{2} = 360$ nodes in this state graph represents a possible state for the cycle. Plots made using the *graph-tool* library [24].

higher allowing for a better exploration of the state space, as we will see in our results.

Considering a single time step of the process, we can split the sum in equation (3.4) into a term representing the probability flow coming from a state x different from y plus the probability that the system was in state y and remained there:

$$\begin{aligned} \mu_t(y) &= \sum_{x\in\Gamma} \mu_{t-1}(x)P(x\to y) \\ &= \sum_{\substack{x\in\Gamma \\ x\neq y}} \mu_{t-1}(x)P(x\to y) + \underbrace{\mu_{t-1}(y)P(y\to y)}_{x=y} \end{aligned} \qquad \forall y\in\Gamma. \tag{3.6}$$

Taking the normalization condition (3.2), the probability for the system to remain in a given state $y\in\Gamma$ in a time step is:

$$P(y\to y) = 1 - \sum_{\substack{x\in\Gamma \\ x\neq y}} P(y\to x), \quad \forall y\in\Gamma \tag{3.7}$$

Plugging in equation (3.7) into equation (3.6), we obtain

$$\begin{aligned} \mu_t(y) &= \sum_{\substack{x\in\Gamma \\ x\neq y}} \mu_{t-1}(x)P(x\to y) + \mu_{t-1}(y)\left[1 - \sum_{\substack{x\in\Gamma \\ x\neq y}} P(y\to x)\right] \\ &= \mu_{t-1}(y) + \sum_{\substack{x\in\Gamma \\ x\neq y}} \left[\mu_{t-1}(x)P(x\to y) - \mu_{t-1}(y)P(y\to x)\right] \end{aligned} \tag{3.8}$$

Defining the finite difference $\nabla\mu_t \equiv \mu_t - \mu_{t-1}$ we arrive at the discrete *master equation* [25]:

$$\nabla\mu_t(y) = \sum_{\substack{x\in\Gamma \\ x\neq y}} \left[\mu_{t-1}(x)P(x\to y) - \mu_{t-1}(y)P(y\to x)\right], \quad \forall y\in\Gamma, \tag{3.9}$$

which states that the probability variation for a given state is the incoming probability flux minus the outgoing one, implying that the probability flow is conserved.

By definition, a distribution π over Γ is said to be stationary if it satisfies

$$\nabla\pi = 0, \tag{3.10}$$

From equation (3.9), this constraint implies the *global balance* condition:

$$\sum_{\substack{x\in\Gamma \\ x\neq y}} \pi(x)P(x \to y) = \sum_{\substack{x\in\Gamma \\ x\neq y}} \pi(y)P(y \to x), \quad \forall y \in \Gamma. \tag{3.11}$$

The stationary state gives us a notion of equilibrium, since once the chain arrives at this regime it remains there indefinitely. This can be seen by summing $\pi(y)P(y \to y)$ on both sides of equation (3.11) and applying the normalization condition (3.2) on the right-hand side, yielding

$$\pi(y) = \sum_{x\in\Gamma} \pi(x)P(x \to y), \quad \forall y \in \Gamma, \tag{3.12}$$

or, simply:

$$\pi = \pi P. \tag{3.13}$$

A possible way to satisfy equation (3.11) is to require that the terms in the sums on both sides of the equation be equal one by one:

$$\pi(x)P(x \to y) = \pi(y)P(y \to x) \quad \forall x, y \in \Gamma. \tag{3.14}$$

This is known as the *detailed balance* condition and characterizes, in particular, systems whose dynamics are the same forwards and backwards in time. Such *reversible Markov chains* have a simple equilibrium regime, i.e., without cycles.

3.2 Metropolis prescription

In physics applications, an important steady state distribution is one that considers the equilibrium energy distribution of a system in contact with a thermal reservoir at temperature θ, which is given by:

$$\pi_\theta(x) = \frac{1}{Z_\theta} \exp\left[-\frac{1}{k_B\theta}\mathcal{H}(x)\right], \qquad \forall x \in \Gamma \tag{3.15}$$

with Z_θ being the partition function

$$Z_\theta = \sum_{x\in\Gamma} \exp\left[-\frac{1}{k_B\theta}\mathcal{H}(x)\right] \tag{3.16}$$

where k_B is the Boltzmann constant and $\mathcal{H}(x)$ is the energy associated with the state $x \in \Gamma$ of the system.

For simplicity's sake, we will use a normalized temperature $T = k_b\theta$ given in units of cost. Hence, we employ the following distribution:

$$\pi_T(x) = \frac{1}{Z_T} \exp\left[-\frac{1}{T}\mathcal{H}(x)\right], \qquad \forall x \in \Gamma \tag{3.17}$$

$$Z_T = \sum_{x\in\Gamma} \exp\left[-\frac{1}{T}\mathcal{H}(x)\right]. \tag{3.18}$$

Starting with the detailed balance condition in the equation (3.14) we apply the Gibbs distribution as the chosen stationary distribution

$$\frac{P(x \to y)}{P(y \to x)} = \frac{\pi_T(y)}{\pi_T(x)} = \exp\left[-\frac{1}{T}(\mathcal{H}(y) - \mathcal{H}(x))\right]. \tag{3.19}$$

It is worth noting that this completely removes the necessity of calculating the partition function Z_T which is in general not known.

There are several ways to satisfy this condition, one way is to split the transition process into two stages: first each state is assigned a given probability to be selected as a candidate next state and then once a candidate next state is selected it has a probability to be accepted:

$$\frac{P(x \to y)}{P(y \to x)} = \frac{g(x \to y)A(x \to y)}{g(y \to x)A(y \to x)} = \exp\left[-\frac{1}{T}(\mathcal{H}(y) - \mathcal{H}(x))\right] \tag{3.20}$$

where g represents the probability that a given state is selected and A represents the probability that the selected state is accepted.

The Metropolis prescription [20] assigns a uniform selection probability, meaning that every state has the same probability of being selected, and the following acceptance probability

$$A(x \to y) = \begin{cases} \exp\left[-\frac{1}{T}(\mathcal{H}(y) - \mathcal{H}(x))\right], & \text{if } \mathcal{H}(y) > \mathcal{H}(x) \\ 1, & \text{otherwise} \end{cases} \tag{3.21}$$

which satisfies the detailed balance condition, as can be verified.

3.3 Outline of the heuristic

Finally, simulated annealing brings all these ideas together using Markov chains to sample the Gibbs distribution of a system using the Metropolis algorithm at progressively lower temperatures so that low energy states are favored and the probability of visiting such low energy states increases greatly. Once the processes terminates at a sufficiently low temperature the state of the system will likely be one of low energy.

The pseudocode in Heuristic 1 summarizes how simulated annealing using the Metropolis algorithm is carried out.

The system begins at a given initial state σ_0 and the temperature is set at a sufficiently high T_0 (lines 2 and 3). The program consists of two nested loops. The outer loop (lines 4 to 17) is the annealing loop and is responsible for cooling the system according to the given annealing schedule, it runs for n_{steps} annealing steps until the system is at equilibrium at a sufficiently low temperature T_f. The inner loop (lines 5 to 15) is the sampling loop, in which the random walk takes place for a suitable number of iterations n_{iter}.

At each iteration of the sampling loop a new configuration is generated (line 6) using the choice of move, in our case either the swap move or the 2-opt move, and then the Metropolis prescription of equation (3.21) is applied (lines 7 to 14) in order to determine if this new configuration is accepted. As an important feature of both our move choices, they provide us with a remarkably cheap way of calculating the cost difference between two adjacent configurations, which is very important for an efficient sampling.

At the end of the outer loop (line 16), the temperature is updated to a lower one `cool`(T) according to the chosen cooling schedule. There are several possible choices for the cooling schedule [26, 27, 28, 29, 30] including one that has been shown to make the system arrive at the global minimum with probability of one after an infinite number of annealing steps [31], whose temperature should decrease logarithmically:

$$T_n = \frac{a}{b + \log(n+1)}, \quad a, b \in \mathbb{R}_+, \tag{3.22}$$

where T_n denotes the temperature at the n-th annealing step.

This choice, however, has been observed to be prohibitively slow in our experiments. On the other hand, what is called a *geometrical cooling schedule* attains very good results much faster but has no mathematical proof of performance. This schedule specifies that the temperature should be updated

Heuristic 1: Simulated Annealing with Geometrical Cooling Schedule

```
Parameters: $\sigma_0$ : initial configuration
            $T_0$ : initial temperature
            $T_f$ : final temperature
            $n_{iter}$ : number of iterations of the internal loop
            $\alpha$ : coefficient of the geometrical cooling
Result: low energy state $\sigma$

begin
    $\sigma \leftarrow \sigma_0$;
    $T \leftarrow T_0$;
    while $T > T_f$ do
        for $i \leftarrow 1$ to $n_{iter}$ do
            $\sigma' \leftarrow$ apply_move($\sigma$);
            if $\mathcal{H}(\sigma') < \mathcal{H}(\sigma)$ then
                $\sigma \leftarrow \sigma'$;
            else
                $x \leftarrow$ rand($[0, 1]$);
                if $x < \exp\left[-\frac{1}{T}(\mathcal{H}(\sigma') - \mathcal{H}(\sigma))\right]$ then
                    $\sigma \leftarrow \sigma'$;
                end if
            end if
        end for
        $T \leftarrow$ cool($T$);
    end while
    return $\sigma$
end
```

by multiplying the current temperature by a real constant $\alpha \in (0, 1)$:

$$T_{n+1} = \alpha T_n. \tag{3.23}$$

More details regarding this annealing schedule will be presented in section 5.1.

Finally, at the very end of the procedure (line 18) the heuristic returns the low energy state at which the system arrives.

This process offers a distinct advantage over deterministic local search algorithms: the Metropolis prescription allows for an increase of the system energy making it less likely to get stuck at local minima [32]. This is extremely useful for combinatorial optimization problems with complex energy landscapes such as the traveling salesman problem.

In this work we will apply simulated annealing as described here using both the simple swap and the 2-opt moves and the geometrical annealing schedule to the traveling salesman problem over complex scenarios and analyze their performance. The following chapter contains a detailed description of the artificial environments in which we intend to apply this heuristic.

Chapter 4

Artificial domains

As a consequence of the nature of heuristics, a number of empirical analyses can be carried out. For example, these analyses can explore the performance of the simulated annealing as an optimization heuristic for the TSP taking into account artificial random environments with interesting tailored characteristics. The environments are two-dimensional domains which are populated randomly by cities that follow a given probability distribution. Then we apply the simulated annealing heuristic to find low energy states. The random environments considered in this work are generated using correlated random variables and long-tailed distributions. Such scenarios will allow us to study the performance of the simulated annealing heuristic during the transition from an NP-Complete problem to one belonging to class P. They also allow to explore situations where the coordinates of the cities have no defined second moment.

4.1 Correlated coordinates

The first random environment explored is the two-dimensional domain populated by points whose x and y coordinates are correlated with a given correlation ρ_{xy}. The correlated pairs of coordinates are generated from uncorrelated pairs of random variables using the same procedure outlined by [33] in the context of emerging of rogue waves in the superposition of electrical waves with correlated phases.

Let us consider a random point $(x, y) \in \mathbb{R}^2$:

$$\begin{cases} x = \alpha_1 z_1 + \alpha_2 z_2 \\ y = \beta_1 z_1 + \beta_2 z_2, \end{cases} \tag{4.1}$$

where z_1 and z_2 are independent and identically distributed random variables, with means $\langle z_1 \rangle = \langle z_2 \rangle = \langle z \rangle$ and $\langle z_1 z_2 \rangle = \langle z_1 \rangle \langle z_2 \rangle = \langle z \rangle^2$.

As a result, the variance of the x coordinate is $\langle (\Delta x)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 = (\alpha_1^2 + \alpha_2^2) \langle (\Delta z)^2 \rangle$ where $\langle (\Delta z)^2 \rangle = \langle z^2 \rangle - \langle z \rangle^2$. Similarly, for the y coordinate, we have $\langle (\Delta y)^2 \rangle = (\beta_1^2 + \beta_2^2) \langle (\Delta z)^2 \rangle$.

Now, we impose the condition

$$\langle (\Delta x)^2 \rangle = \langle (\Delta y)^2 \rangle = \langle (\Delta z)^2 \rangle, \tag{4.2}$$

which implies that $\alpha_1^2 + \alpha_2^2 = \beta_1^2 + \beta_2^2 = 1$.

It is worth noting that although z_1 and z_2 are not correlated, the new variables x and y are correlated, and their correlation is

$$\rho_{xy} = \frac{\langle (x - \langle x \rangle)(y - \langle y \rangle) \rangle}{\sqrt{\langle (\Delta x)^2 \rangle \langle (\Delta y)^2 \rangle}} = \alpha_1 \beta_1 + \alpha_2 \beta_2. \tag{4.3}$$

After some algebra we arrive at an explicit formula for the correlated coordinates (x, y)

$$\begin{cases} x & = z_1 \sin\phi + z_2 \cos\phi \\ y & = z_1 \cos\phi + z_2 \sin\phi, \end{cases} \tag{4.4}$$

where the angle ϕ is obtained from the desired correlation ρ_{xy}:

$$\phi = \frac{1}{2} \arcsin \rho_{xy}. \tag{4.5}$$

The average for the coordinates is given by

$$\langle x \rangle = \langle y \rangle = \frac{1}{\sqrt{2}} \left[\left(1 - \sqrt{1 - \rho_{xy}^2}\right)^{1/2} + \left(1 + \sqrt{1 - \rho_{xy}^2}\right)^{1/2} \right] \langle z \rangle \tag{4.6}$$

From this we can draw two important conclusions:

1. As required by condition (4.2), the random variables x and y have the same variance of z_1 and z_2 that are themselves identically distributed and therefore their variance does not depend on the value chosen for ρ_{xy}.

2. According to equation (4.6), for $\langle z_1 \rangle = \langle z_2 \rangle = \langle z \rangle = 0$ we have $\langle x \rangle = \langle y \rangle = 0$.

Therefore, if one considers x and y as ρ_{xy}-correlated random variables generated from two independent and identically distributed random variables z_1 and z_2 with average $\langle z \rangle = 0$ and variance $\langle (\Delta z)^2 \rangle = \sigma^2$ then x and y will also have average zero and the same variance σ^2. This is important because, in our explorations of simulated annealing in this dissertation, by keeping the average and variance unchanged the only parameter accountable for the observed effects in our future analyses will be the correlation ρ_{xy}.

For the domains analyzed we choose both random variables z_1 and z_2 to be uniform random variables assuming values in $[-\frac{1}{2}, \frac{1}{2}]$. Then, in order to determine the contribution of the shape of the original distribution, we choose both z_1 and z_2 to be normal random variables with $\sigma = \frac{1}{\sqrt{12}}$ which has the same variance as the uniform distribution.

Figures 4.1 and 4.2 show examples of domains generated using both choices of z_1 and z_2 for different values of ρ_{xy}.

4.2 Long-tailed distributions

Another interesting set of instances investigated in this work considers random environments with points whose coordinates follow long-tailed distributions. We employ the following power law probability density function to generate the coordinates of the N points in two dimensions.

$$p(x; x_0, \gamma) = \begin{cases} \dfrac{\gamma - 1}{2x_0^{1-\gamma}} |x|^{-\gamma}, & \text{if } |x| \geq x_0 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

Notably, the power law distribution given by equation (4.7) has a gap of size $\Delta = 2x_0$ centered at the origin in order to be normalizable. However this gap can be made arbitrarily small by tuning the parameter x_0. Figures 4.3 and 4.4 illustrate the density function for this distribution.

Using the transformation method [34, Section 7.3], it is possible to draw samples from the distribution of equation (4.7) by making use of only two uniformly distributed random variables ξ_1 and ξ_2 assuming values in $[0, 1]$. Firstly, we decide if the value is on the negative or positive branch by checking the sign of $2\xi_1 - 1$. Then we make ξ_2 equal to the cumulative distribution

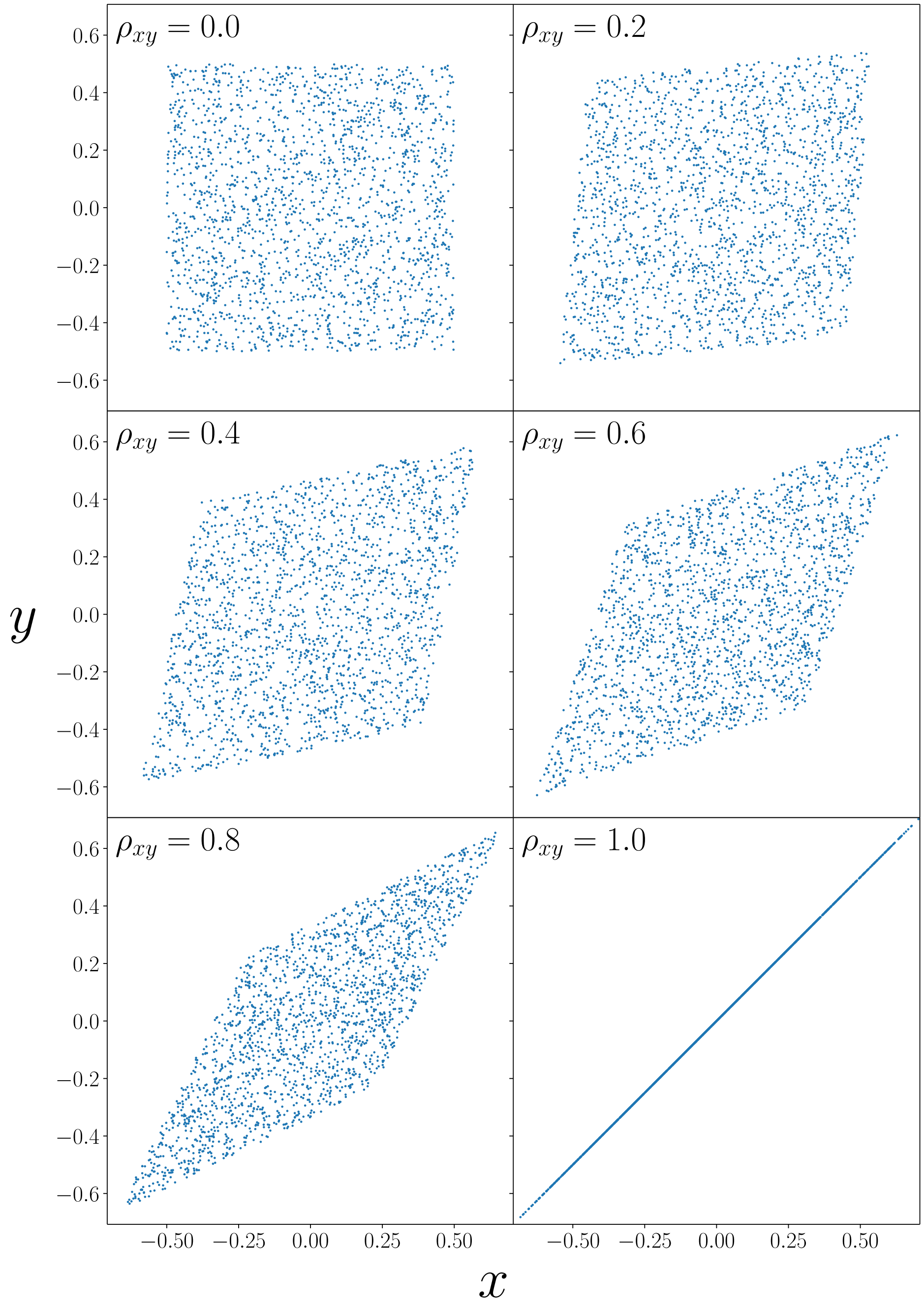


Figure 4.1: Effects of correlation between coordinated on the points distribution using a starting pair of uniform random variables over $[-\frac{1}{2}, \frac{1}{2}]$.

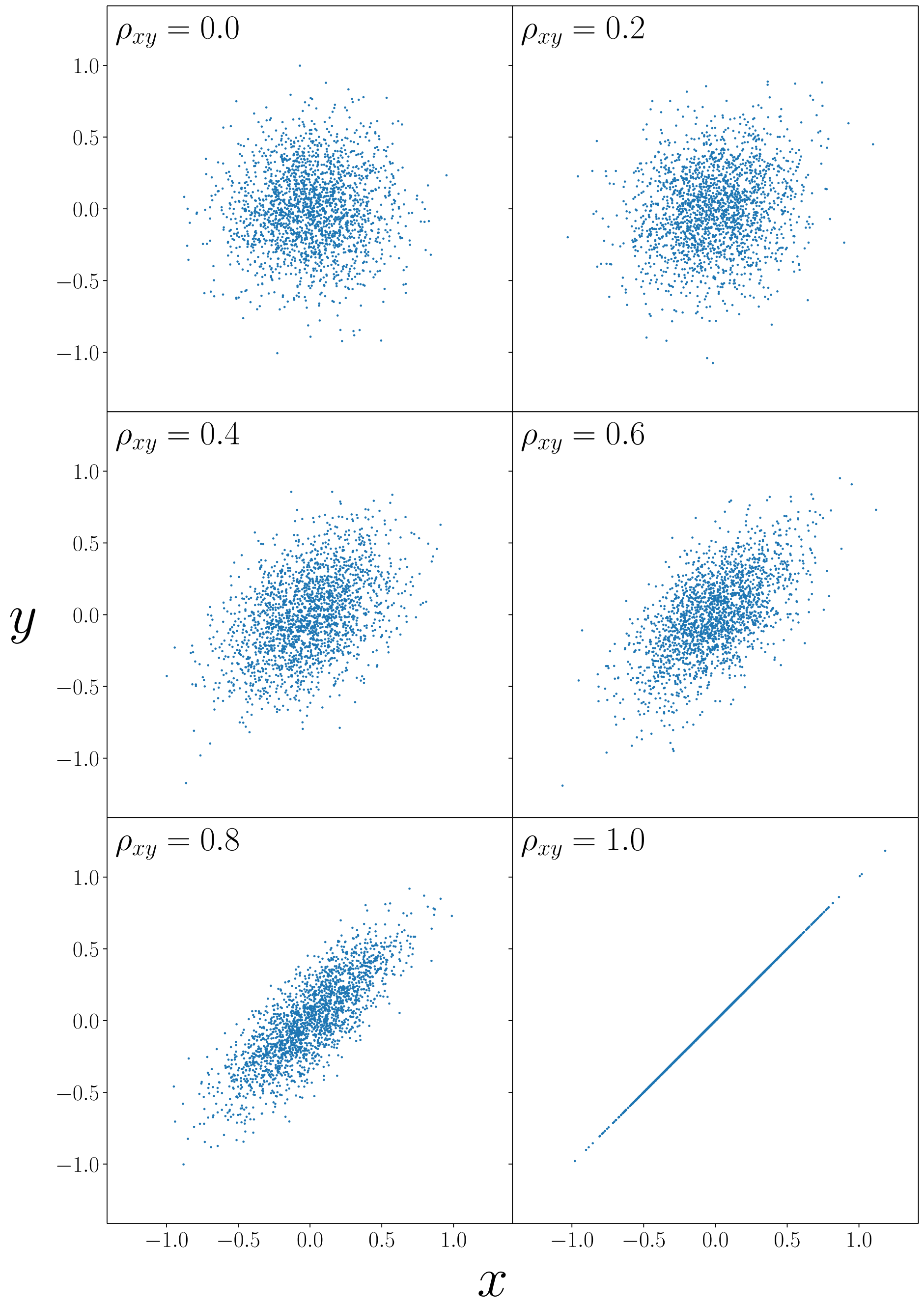


Figure 4.2: Effects of correlation between coordinated on the points distribution using a starting pair of normal random variables with $\sigma = \frac{1}{\sqrt{12}}$.

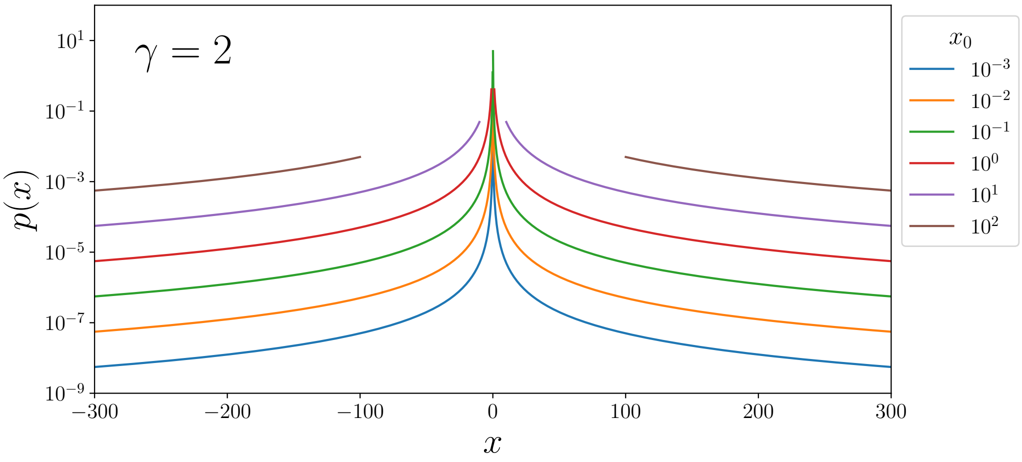


Figure 4.3: Two-tailed power law probability density function $p(x; x_0, \gamma)$ for $\gamma = 2$ and different values of x_0.

function $\dfrac{\gamma - 1}{x_0^{1-\gamma}} \displaystyle\int_{x_0}^{w} x^{-\gamma} dx$ of the power law. This results in $w = x_0(1-\xi_2)^{\frac{1}{1-\gamma}}$ and therefore the random variable following the probability distribution in equation (4.7) constructed from ξ_1 and ξ_2 is:

$$x = \frac{2\xi_1 - 1}{|2\xi_1 - 1|} x_0 (1 - \xi_2)^{\frac{1}{1-\gamma}} \tag{4.8}$$

Since ξ_2 is uniformly distributed over the interval $[0, 1]$, so is $1-\xi_2$. Hence, renaming the variable $\xi_2 \leftarrow 1 - \xi_2$ and using the sign function sgn which is equal to $+1$ for positive values and -1 for negative values, the simplified version becomes:

$$x = \operatorname{sgn}(2\xi_1 - 1) x_0 \xi_2^{\frac{1}{1-\gamma}} \tag{4.9}$$

Using the method above we can generate two dimensional points whose (x, y) coordinates follow the power law distribution of equation (4.7), which we shall label *Power law I*:

$$\begin{cases} x = \operatorname{sgn}(2\xi_1 - 1) x_0 \xi_2^{\frac{1}{1-\gamma}} \\ y = \operatorname{sgn}(2\xi_3 - 1) x_0 \xi_4^{\frac{1}{1-\gamma}} \end{cases} \tag{4.10}$$

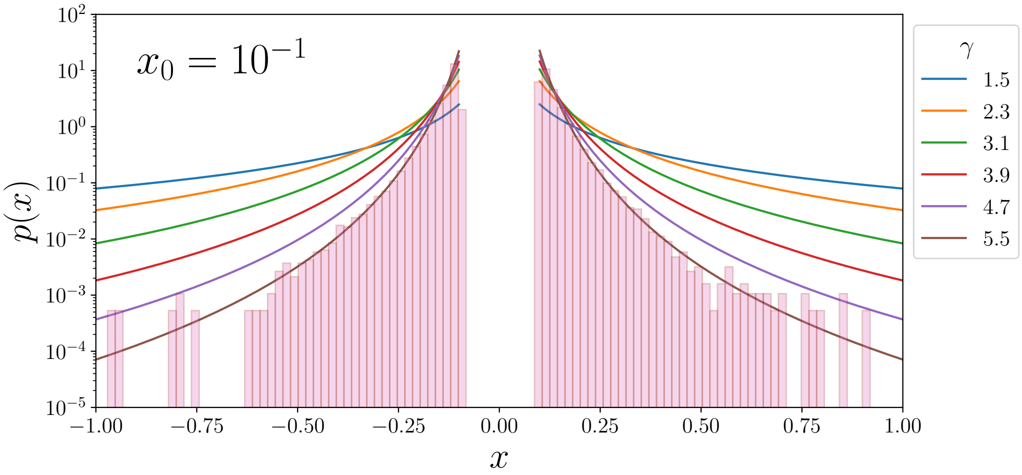


Figure 4.4: Two-tailed power law probability density function $p(x; x_0, \gamma)$ for $x_0 = 10^{-1}$ for different values of γ. The histogram is obtained with points sampled with equation (4.9) for $\gamma = 5.5$ in order to demonstrate the sampling method used.

Since both x and y coordinates are sampled from the same distribution, this configuration of points is axially symmetric. Another possibility is to generate radially symmetric points (here labeled *Power law II*) using only two uniform random variables, ξ_1 to determine the direction and ξ_2 to determine the distance from the origin:

$$\begin{cases} x = r_0 \cos(2\pi\xi_1)\xi_2^{\frac{1}{1-\gamma}} \\ y = r_0 \sin(2\pi\xi_1)\xi_2^{\frac{1}{1-\gamma}}, \end{cases} \tag{4.11}$$

both of these approaches are illustrated in the figure 4.5.

The Gaussian distribution along with the plot of *Power Law I* with $\gamma = 3.4$ are two cases that will be important for our results later on. Immediately, we notice the abundance of outlier points in the cases with $\gamma \leq 3$ for which the distribution has no second moment defined. This effect is weakened for the cases with higher γ.

Having studied the environments for which we will apply simulated annealing to obtain approximate optimal cycles of the traveling salesman problem, we will present our results. We divided them in two parts: a preparatory part which is more pedagogical and a second part where our main results are presented.

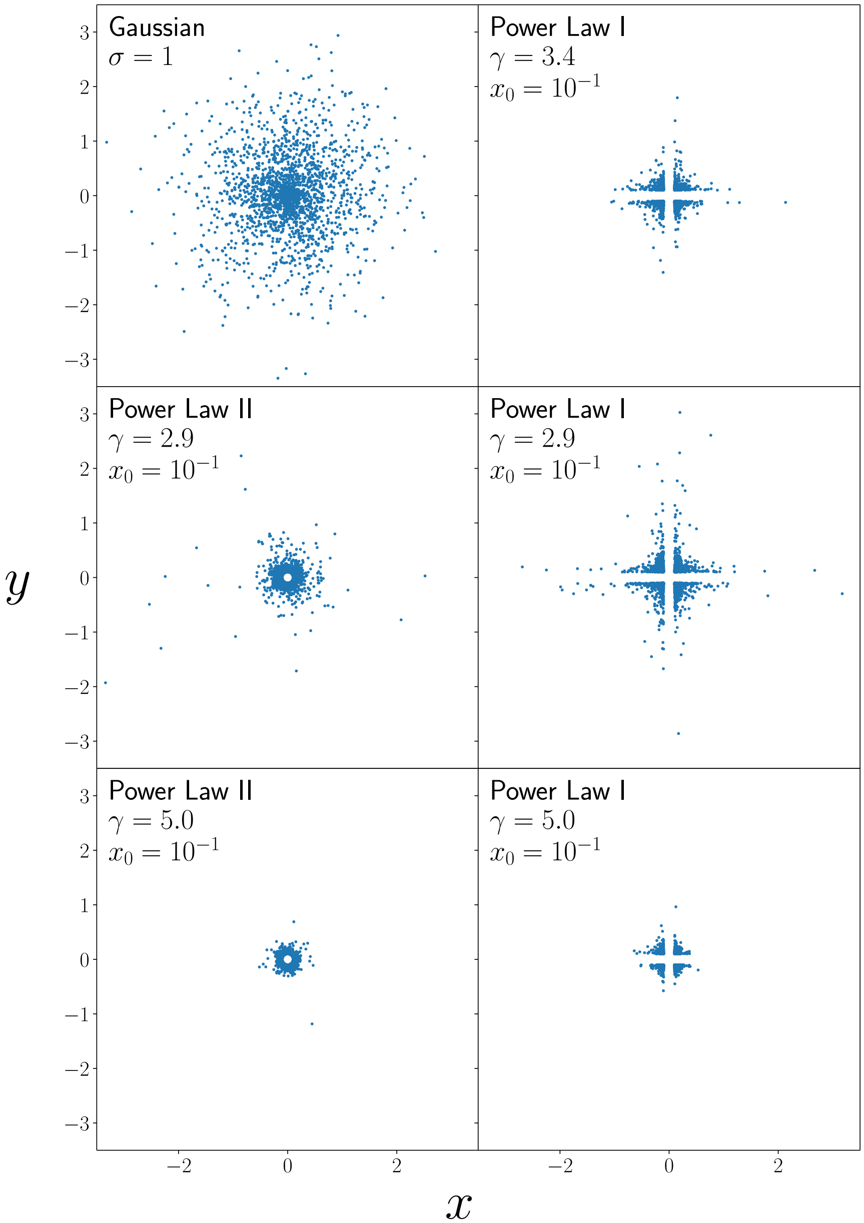


Figure 4.5: Scattering of points comparing coordinates generated using standard Gaussian distribution and power law distributions using both equations (4.10) and (4.11) for different exponents γ. Notice the gap around the origin for the case of the power laws.

Chapter 5

Computational results

In this chapter we present the results obtained when analyzing the performance of the simulated annealing applied to the environments described in the previous chapter. We begin by exposing some general and pedagogical aspects of the heuristic with the geometrical cooling schedule. Then, an analysis of the effects of the correlations between the coordinates on the performance of the simulated annealing is performed. Finally, the effects of a long-tailed distribution of points is considered.

5.1 Pedagogical aspects of simulated annealing

In this preliminary study of the simulated annealing heuristic we begin by considering the chosen annealing schedule, i.e., the geometrical annealing schedule, and then evaluate the performance of the heuristic when applied to the traveling salesman problem over uniformly distributed points in a unit square.

5.1.1 Geometrical annealing schedule

The geometrical annealing schedule has been used since the earliest implementations of simulated annealing to the TSP [5]. Despite the lack of concrete mathematical proof for its performance, it has been an attractive choice for yielding very good results in a reasonable amount of time. It consists in starting at a given initial temperature T_0 and at each annealing step decrease the temperature by multiplying the current temperature by a given $\alpha \in (0, 1)$. Hence, the temperature at the t-th annealing step is

$$T_t = \alpha^t T_0 \tag{5.1}$$

This procedure is repeated until a final temperature T_f is reached after n_{steps} annealing steps.

$$T_f = \alpha^{n_{steps}-1} T_0 \tag{5.2}$$

The integer n_{steps} can be thought of as how many different temperatures the system was subject to during the annealing process. For instance, the case for which $n_{steps} = 2$ corresponds to sampling the system at T_0 and then decreasing the temperature to $T_f = \alpha T_0$ and sampling the system again before ending the process.

Another important variable is n_{iter} which is the number of iterations that the system was sampled at each annealing step. It is straightforward to note that the total number of samples generated n_{total} is simply the product of the number of annealing steps n_{steps} and the number of iterations at each step n_{iter}.

$$n_{total} = n_{steps} \times n_{iter} \tag{5.3}$$

In order to quantify how divided the temperature range is by a given cooling schedule, we introduce the following ratio:

$$R_{cs} = \frac{\log(n_{steps})}{\log(n_{total})} \tag{5.4}$$

A less divided temperature range will have very few annealing steps, that is $n_{steps} \ll n_{total}$ yielding $R_{cs} \to 0$. On the other hand, for a more subdivided temperature range $n_{steps} \to n_{total}$ which in turn implies $R_{cs} \to 1$.

5.1.2 Uniformly distributed points

Now, we apply the simulated annealing heuristic using the geometrical cooling schedule to traveling salesman problem instances consisting of N uniformly distributed points over the unit square $[0, 1]^2$. The measure for performance used throughout this work is the ratio between the average cost achieved $\overline{C}$ and the expected cost of a cycle drawn randomly for a given system specification. The average cost achieved is calculated over n_{run} runs:

$$\overline{C} = \frac{1}{n_{run}} \sum_{i=1}^{n_{run}} C_i \tag{5.5}$$

where C_i is the cost achieved at the i-th run.

And the expected cost of a cycle is given by the expected distance between two points multiplied by the size of the cycle:

$$\langle C_0 \rangle = N \frac{\sum_{i<j} d(i,j)}{\binom{N}{2}} = \frac{2}{N-1} \sum_{i<j} d(i,j) \tag{5.6}$$

For the two dimensional unit uniform domain the expected value for the distance between two points can be calculated exactly (see Appendix 1). Hence, the expected cost of a random cycle composed of N randomly distributed points is

$$\langle C_0 \rangle = \frac{N}{15} \left(2 + \sqrt{2} + 5 \ln(\sqrt{2} + 1) \right) . \tag{5.7}$$

In figure 5.1, we present the results from the annealing of systems with N uniformly distributed points using both the swap and the 2-opt moves. We begin at the highest temperature $T_0 = 10^2$ and decrease the temperature geometrically until the final temperature $T_f = 10^{-7}$ is reached. At each annealing step the system is sampled for $2^{22} \approx 4.19 \times 10^6$ iterations to allow it thermalize and then the average cost is calculated over the next 2^{22} iterations. This process is repeated for $n_{run} = 60$ different systems. Finally, the performance is defined as the average cost at each temperature step divided by the expected random cycle cost.

We can see that the 2-opt move offers a better performance than the swap move especially for larger instances. Additionally, we note that $T_f = 10^{-7}$ is clearly low enough since even larger instances are frozen at this temperature. As an argument in support of the geometrical annealing schedule we note in figure 5.1 how smoothly the average cost varies in a log-log plot as a function of the temperature.

An illustration of the state of a sample system being annealed using the 2-opt move is shown in figure 5.2. We note how the system goes from very messy and costly cycles at high temperatures to remarkably inexpensive ones at lower temperatures.

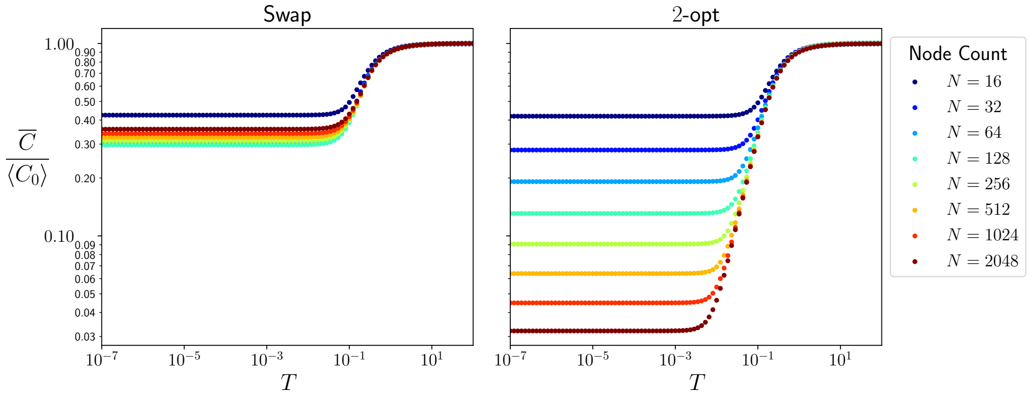


Figure 5.1: Annealing of systems with N uniformly distributed points using both the swap and 2-opt moves.

At low temperatures the cycle cost obtained using the 2-opt move follows a power law of the form

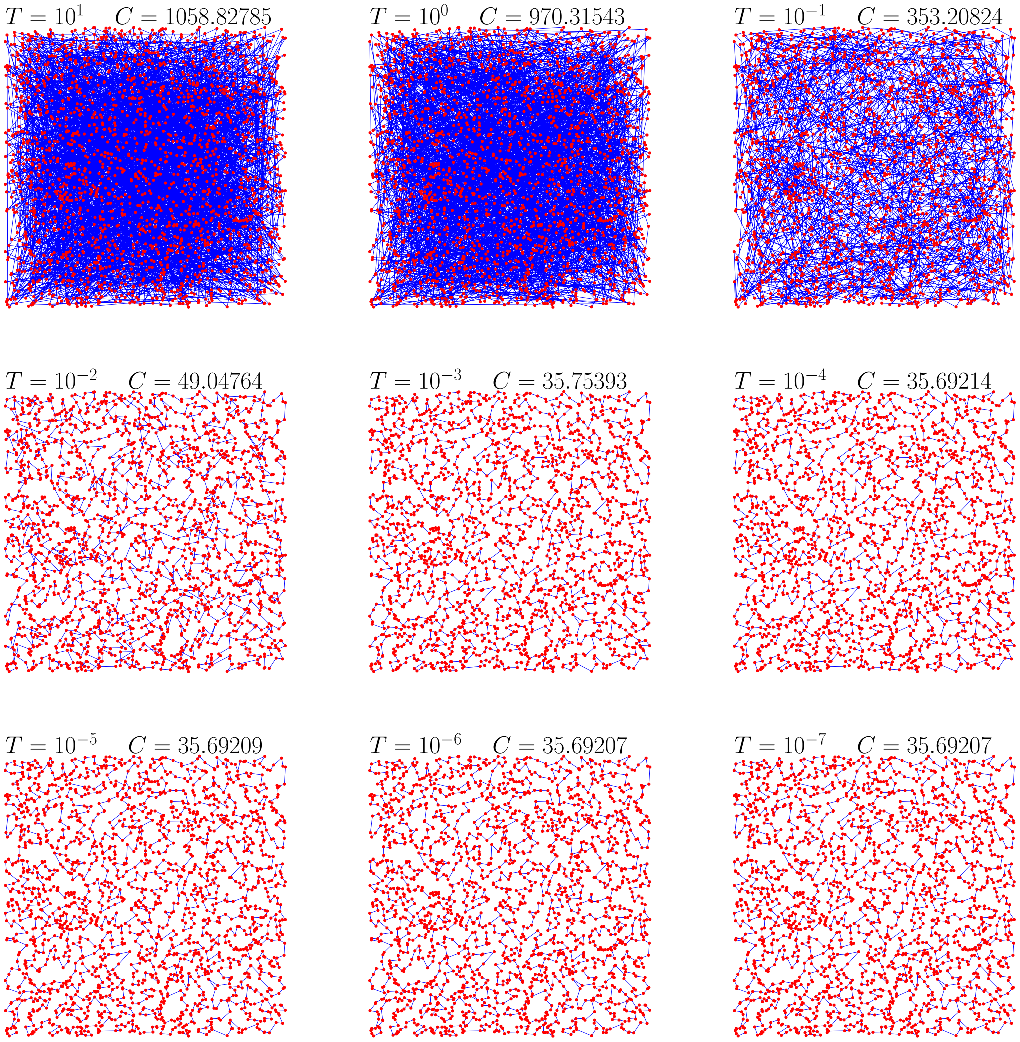
$$\frac{\overline{C}(T_f)}{\langle C_0 \rangle} \propto N^{-\delta}, \tag{5.8}$$

as shown in figure 5.3, along with the coefficient of determination r^2 of the fit. Once again, we note how much better the 2-opt move performs when compared to the swap move. For this reason we will use the 2-opt move in the remainder of this section.

An important question regarding the annealing schedule is how the temperature range subdivision impacts the system's final cost. That is, how the choice of α impacts the final cost for uniformly distributed domains. In order to assess this effect, we fix the initial temperature T_0, the final temperature T_f and the total number of samples n_{total} generated by the heuristic and then vary the number of temperature steps n_{steps} choosing an appropriate number of iterations at each temperature step $n_{iter} = \frac{n_{total}}{n_{steps}}$ to maintain n_{total} constant. This also implies that

$$\alpha = \left(\frac{T_f}{T_0}\right)^{\frac{1}{n_{steps}-1}}. \tag{5.9}$$

We apply the simulated annealing heuristic using the 2-opt move and geometrical annealing schedule on TSP instances consisting of uniformly distributed points using as initial temperature $T_0 = 1$ and a final temperature



$T = 10^{1}$ $C = 1058.82785$

$T = 10^{0}$ $C = 970.31543$

$T = 10^{-1}$ $C = 353.20824$

$T = 10^{-2}$ $C = 49.04764$

$T = 10^{-3}$ $C = 35.75393$

$T = 10^{-4}$ $C = 35.69214$

$T = 10^{-5}$ $C = 35.69209$

$T = 10^{-6}$ $C = 35.69207$

$T = 10^{-7}$ $C = 35.69207$

Figure 5.2: State of a TSP cycle on $N = 2048$ uniformly distributed points sampled at different temperatures using 2-opt moves. The values of the cycle costs C for each state are also shown on top of each figure. We can observe the interesting unfolding of the cycle.

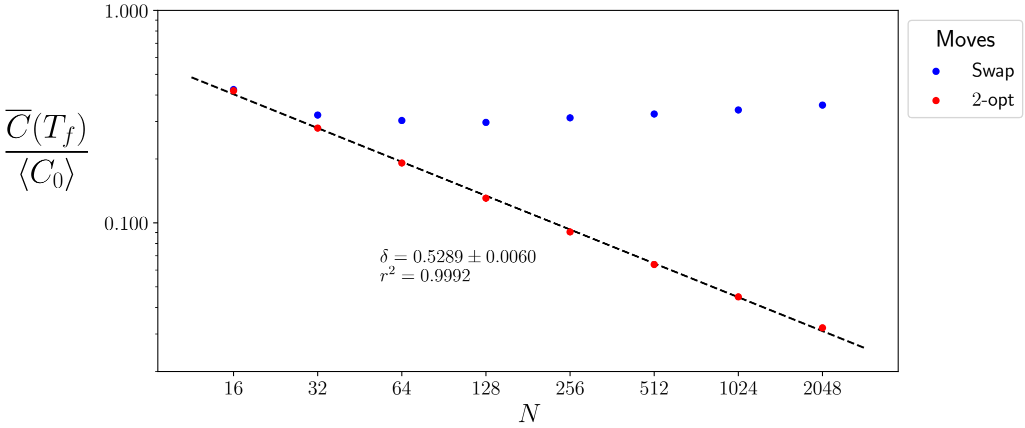


Figure 5.3: Low temperature $T_f = 10^{-7}$ normalized costs obtained by using both the swap and 2-opt moves.

$T_f = 10^{-7}$ and plot the final cost obtained as a function of the total number of iterations n_{total} in figure 5.4.

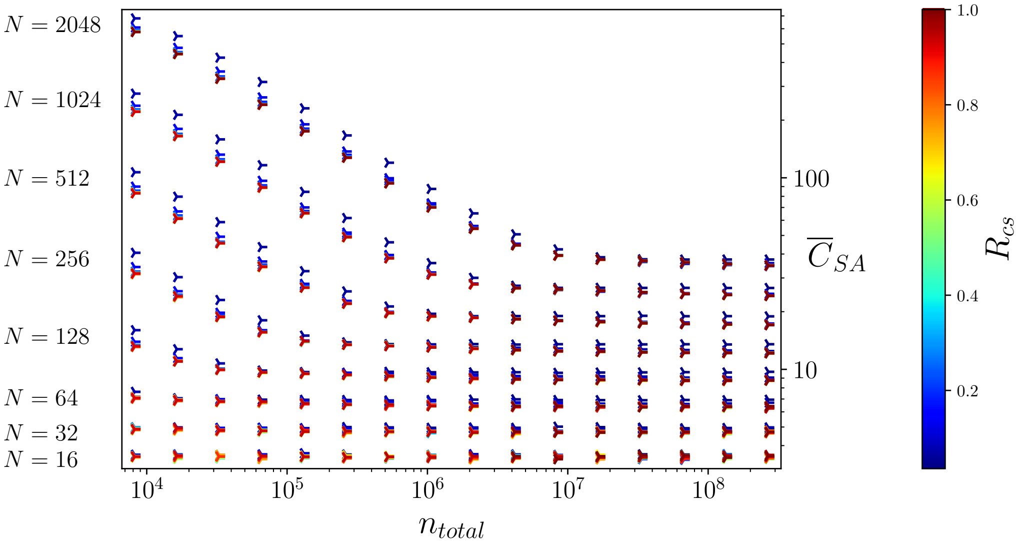


Figure 5.4: Final cost obtained as a function of the total number of samples n_{total} generated by the simulated annealing with 2-opt moves. The color of each marker represent the value of R_{cs} which indicates how subdivided the temperature range is.

We note that the final cost obtained does not depend heavily on the way the temperature range is divided, that is on the choice of α; however, a smaller number of annealing steps n_{steps} is still disfavored. In fact, as figure 5.5 suggests, for the final cost obtained, $\overline{C}_{SA}$, follows a power law as a function of n_{total}:

$$\overline{C}_{SA} \propto n_{total}^{-\delta}. \tag{5.10}$$

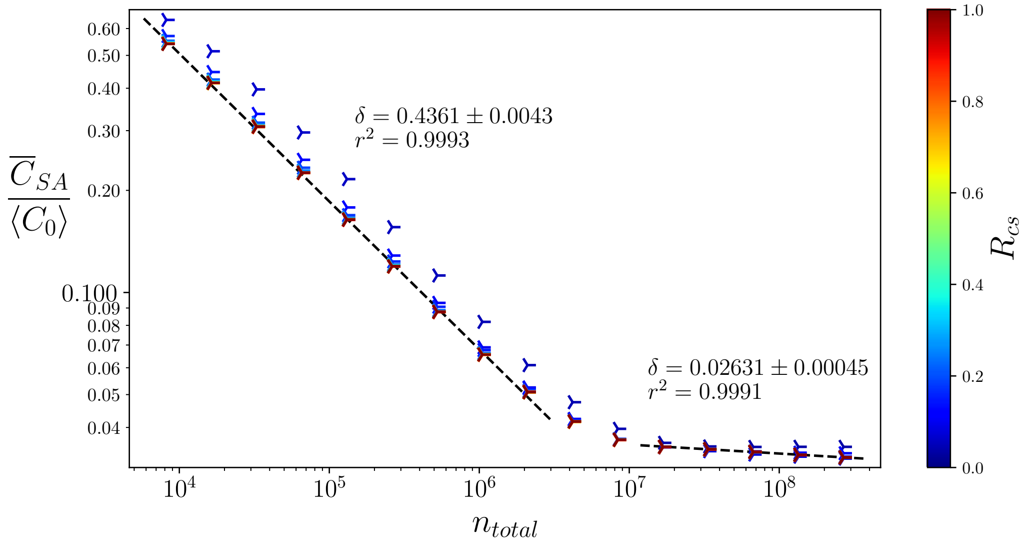


Figure 5.5: Power law fit on the final cost obtained as a function of the total number of samples n_{total} generated the simulated annealing with 2-opt moves. The fit is performed on instances $N = 2048$ and using the $R_{cs} = 1$ curve, but similar results can be obtained for other values of R_{cs}.

We observe a transition between two power laws with very different exponents: from $\delta \approx 0.44$ to $\delta \approx 0.032$, which shows that for $n_{total} > 2^{22} \approx 4.2 \times 10^6$ the optimal cost found does not show a meaningful improvement for increased number of samples.

In order to show that there are indeed several data points sitting on top of one another we show in figure 5.6 the high n_{total} region of the previous plot in more detail and in figure 5.7 the final costs obtained for using $n_{total} = 2^{24} \approx 1.68 \times 10^7$ as a function of R_{cs}.

Figure 5.7 shows that apart from the very first point which corresponds to having only two annealing steps, one at $T_0 = 1$ and the other at $T_f = 10^{-7}$,

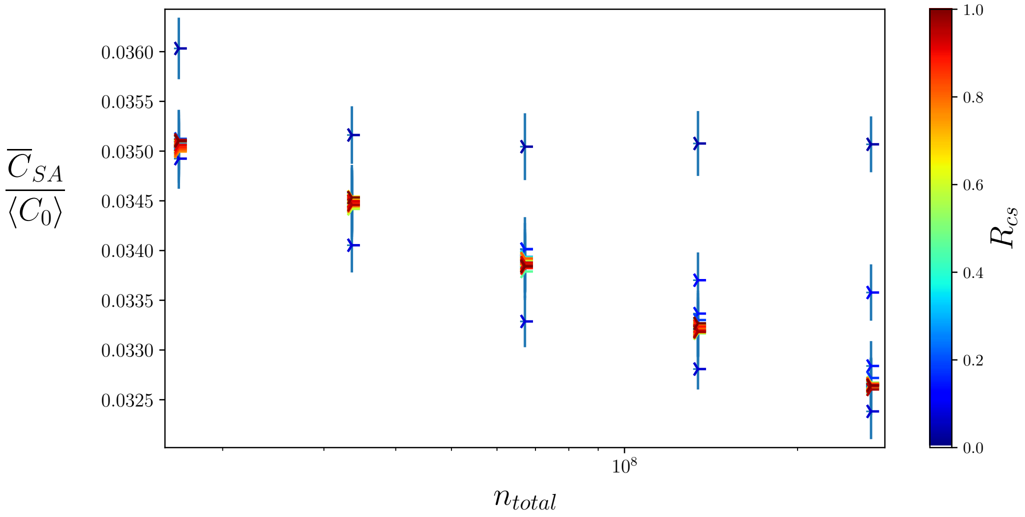


Figure 5.6: Same plot as in figure 5.5 but with a linear y-axis, highlighting the multiple points lying on top of each other and their uncertainties.

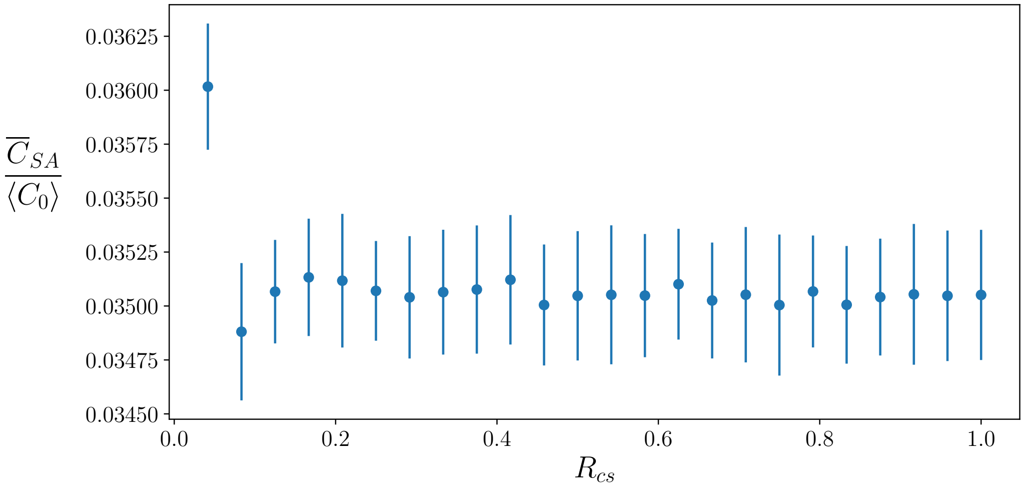


Figure 5.7: Final costs for Simulated Annealing with 2-opt moves of domains with $N = 2048$ points.

there is no substantial difference in the final cost obtained.

For the range of temperatures studied, we observe in figure 5.8 that as the total number of samples generated n_{total} grows the normalized final cost approaches a power law $\overline{C}_{SA}/\langle C_0\rangle \propto N^{-\delta}$ with exponent $\delta \approx 0.53$.

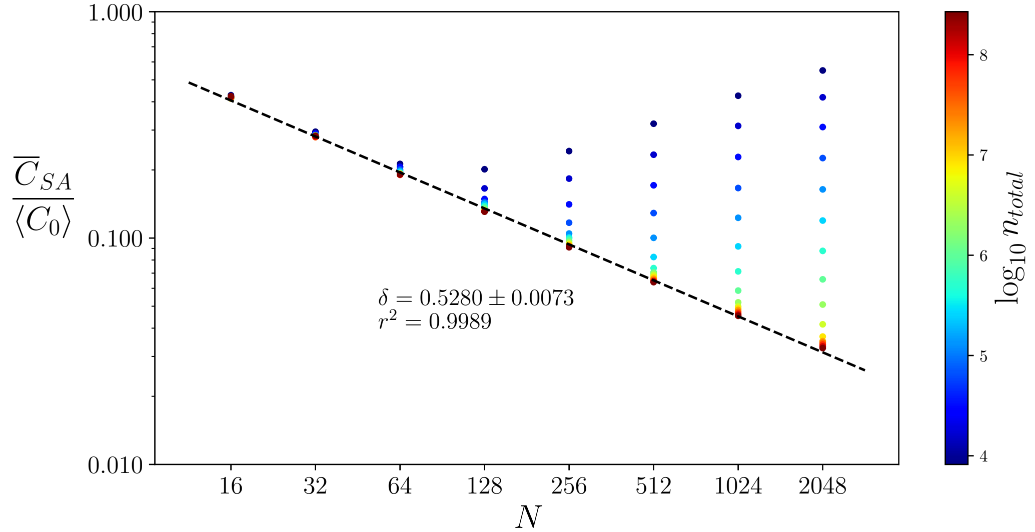


Figure 5.8: Normalized final costs obtained by simulated annealing with 2-opt moves for uniformly distributed points. Different curves represent different number of total samples n_{total} generated.

We compare the simulated annealing heuristic with the nearest neighbor heuristic, recalling that the latter is a constructive heuristic which only takes into account local constraints to build the cycle. In figure 5.9 we plot the ratio between the average cost obtained by simulated annealing with 2-opt moves and the average cost obtained by the nearest neighbor heuristic for different sizes of the uniform domain.

We observe that for sufficiently high n_{total} the cost obtained by simulated annealing remains consistently lower for all system sizes considered.

In order to assess the final costs obtained in the absence of the actual optimal costs for each cycle we can use a theorem in [35] that for the case of two dimensional uniform domains states that:

$$\lim_{N\to\infty} \frac{C_{optimal}(N)}{\sqrt{N}} = \beta \quad \text{with probability 1,} \tag{5.11}$$

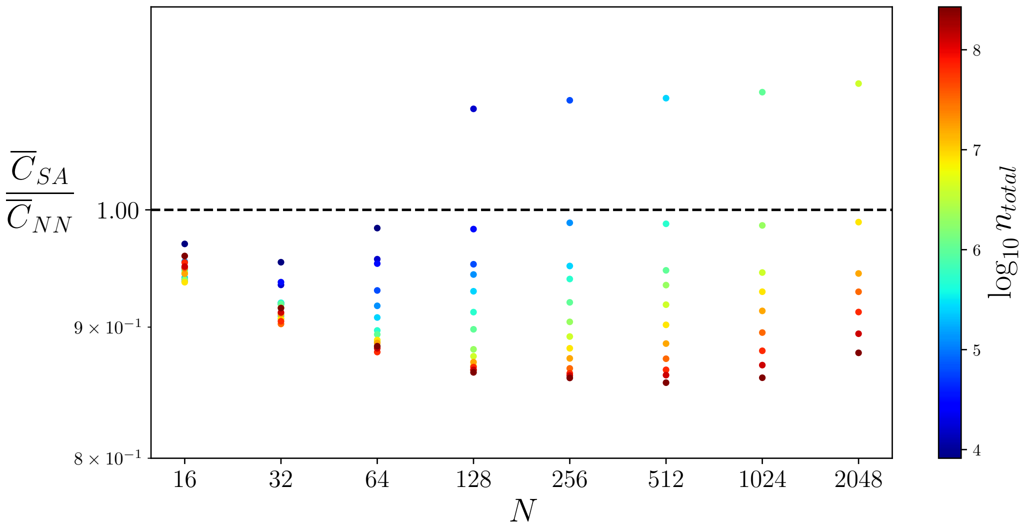


Figure 5.9: Final costs obtained by simulated annealing with 2-opt moves over the average cost obtained by the nearest neighbor heuristic for uniformly distributed points. Different curves represent different number of total samples n_{total} generated by simulated annealing.

where $C_{optimal}(N)$ is the optimal cost of the TSP for N uniformly distributed points over the unit square and β is a constant often called the traveling salesman constant. The currently available upper and lower bounds to the traveling salesman constant were calculated in [36] and can be used to judge how good a TSP cycle over a large number of points N is. Figure 5.10 shows the cycle cost obtained divided by the square root of the point count N as a function of N.

Importantly, we do not claim that the costs obtained by simulated annealing for the TSP over uniformly distributed points are optimal, rather this suggests that they are close to optimal.

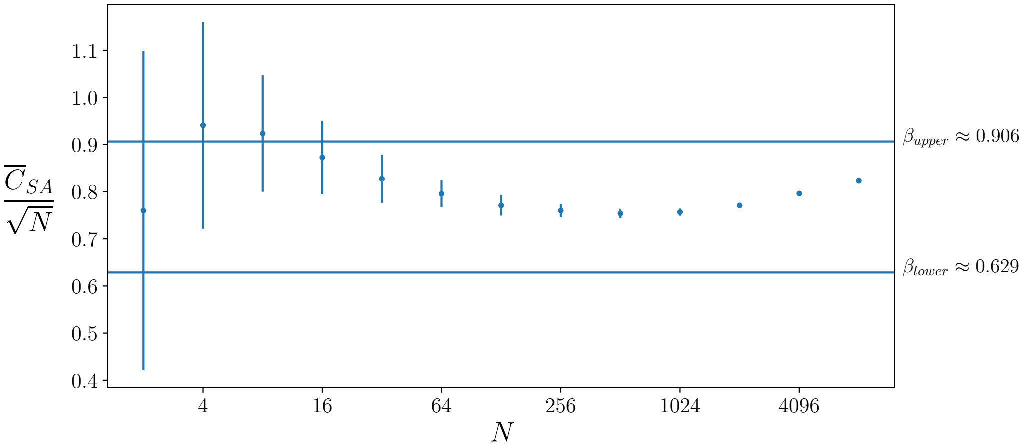


Figure 5.10: Final costs $\overline{C}_{SA}$ obtained by simulated annealing using 2-opt moves. The upper and lower bounds shown here were calculated in [36].

In this preliminary investigation, we were able to determine that when applied to the TSP over uniformly distributed points simulated annealing with 2-opt move is much more efficient than when using the swap move. Regarding the geometrical annealing schedule, we have shown that the cycle cost obtained by the heuristic does not depend heavily on how the temperature range is divided, that is, on the choice of α. Additionally, this heuristic suffers a considerable performance reduction after a certain number of iterations. Finally, the efficiency, in this context defined as the average cost cycle obtained over the expected cost of a random cycle was shown to improve for larger systems following a power law of the form

$$\frac{\overline{C}_{SA}}{\langle C_0 \rangle} \propto N^{-\delta}, \tag{5.12}$$

with exponent $\delta \approx 0.53$.

5.2 Main results

In this section we will expose our main results. First, in subsection 5.2.1 we explore the effects of the correlation between the coordinates of the points on the performance of simulated annealing. Then, in subsection 5.2.2 the effects of long tailed distributions of points are studied.

5.2.1 Correlation effects

In this section, we investigate the effects of correlations between the coordinates of points on the efficiency of the simulated annealing when applied to the TSP. The systems are generated following the procedures outlined in the previous chapter. We analyze the performance of the heuristic for systems with a correlation ranging from $\rho_{xy} = 0$, which is the uniform domain considered in the previous section, to $\rho_{xy} = 1$, which consists essentially of N collinear points. It is important to point out that an optimal TSP cycle on such a finite one-dimensional aperiodic domain can be found by simply sorting the points position. Hence, we are essentially investigating the behavior of the simulated annealing heuristic on the boundary between a NP and a P problem.

We begin by probing the behavior of the heuristic in such systems for different values of correlation ρ_{xy} in figure 5.11. The annealed systems consist of $N = 2048$ points and each curve corresponds to a different correlation ρ_{xy} between its coordinates. Importantly, since there is no analytical expression for the expected cost of a random cycle on such a system for arbitrary ρ_{xy}, we will calculate an average initial cost over $n_{runs} = 2^{14}$ runs:

$$\overline{C}_0(\rho_{xy}) = \frac{1}{n_{run}} \sum_{i=1}^{n_{run}} C_0^i(\rho_{xy}) \tag{5.13}$$

where $C_0^i(\rho_{xy})$ is the initial cost for the i-th run, and then define the performance as the average cost obtained $\overline{C}$ divided by the average initial cost.

The same temperature range as in the preliminary investigation of uniform domains is used. At each annealing step the system is sampled for $2^{22} \approx 4.19 \times 10^6$ iterations in order to thermalize and then the average cost is calculated over the next 2^{22} iterations. The values at each temperature are then averaged over 60 different systems.

The values of correlation range from $\rho_{xy} = 0$ to $\rho_{xy} = 0.99$ in steps of 0.01 and then from $\rho_{xy} = 0.99$ to $\rho_{xy} = 1$ in finer steps of 0.001 in order to take into account the rapid variation of the final cost near $\rho_{xy} = 1$. We observe once again that the 2-opt move is vastly more efficient than the swap move and is able to attain lower costs at low temperatures. Another interesting point is that the normalized final cost for low temperatures quickly diminishes as ρ_{xy} approaches one, i.e., in the $2D \to 1D$ limit.

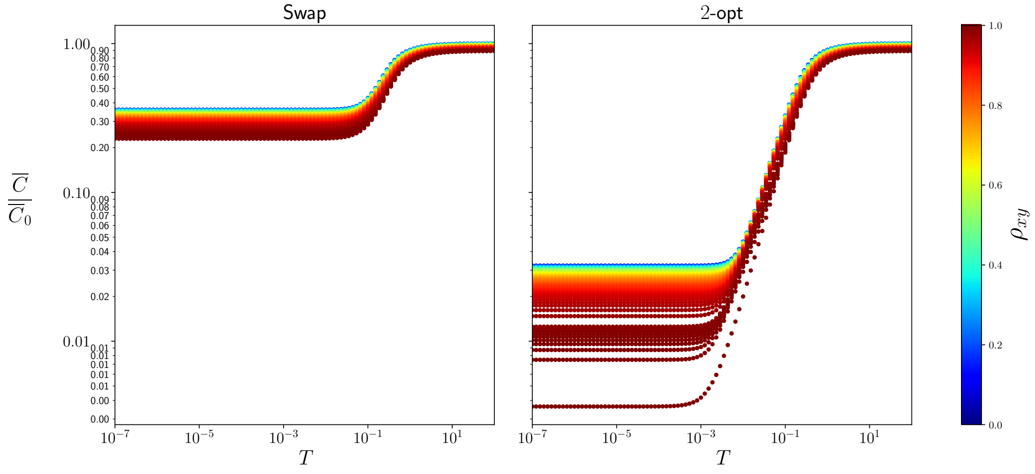


Figure 5.11: Annealing of a system with $N = 2048$ points with correlated coordinates using both the swap and 2-opt moves.

An analysis of the geometrical annealing schedule similar to the one done for the uniformly distributed domain is shown in figure 5.12 by setting the initial temperature to $T_0 = 1$ and the final temperature to $T_f = 10^{-7}$, we apply the heuristic dividing the temperature range with different values of R_{CS}. All systems considered are composed of $N = 2048$ points.

Similarly to the case of the uniform domain, the way that the temperature range is subdivided by the annealing schedule does not play a very important role in determining the average final cost obtained. Once again, we observe that at first the final cycle cost obtained quickly diminishes as the total number of samples generated by the heuristic grows (first region, $n_{total} \lesssim 10^6$), until a point after which the performance quickly diminishes (second region, $n_{total} \gtrsim 10^7$). In both of these regions the average final cost obtained $\overline{C}_{SA}$ displays a power law behavior as a function of n_{total}:

$$\overline{C} \propto n_{total}^{-\delta} \tag{5.14}$$

The exponents δ of the fits performed for the different values of correlation ρ_{xy} in the first region are presented in table 5.1 along with the coefficient of determination for each fit.

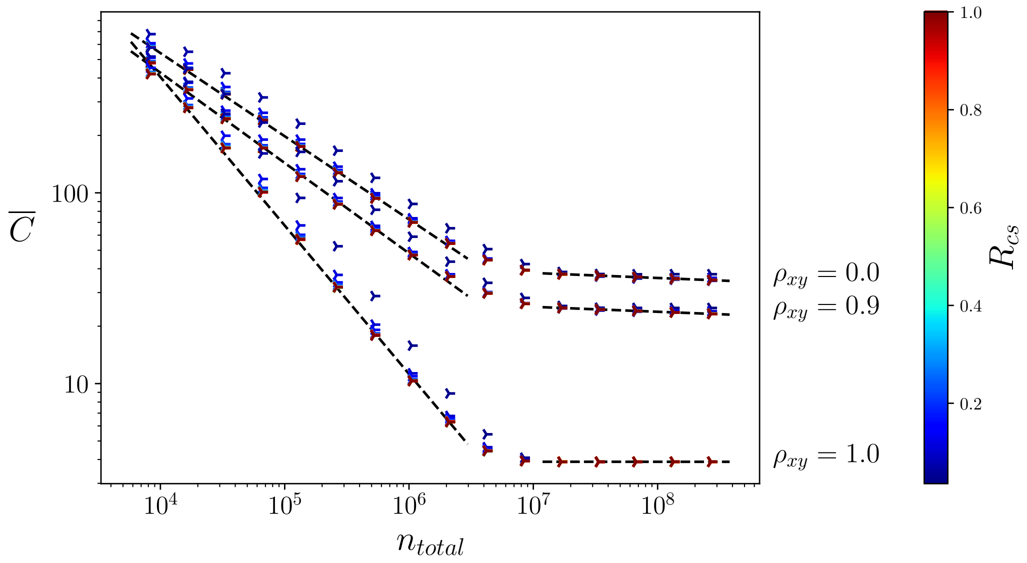


Figure 5.12: Final cost obtained by simulated annealing using 2-opt move for two dimensional domains with correlated coordinates as a function of the total number of samples n_{total} generated by the heuristic. The fit is performed using the $R_{cs} = 1$ curve, but similar results can be obtained for other values of R_{cs}.

ρ_{xy}	First region	Second region
0.0	$\delta = 0.4361 \pm 0.0043$ $\boldsymbol{r^2 = 0.9993}$	$\delta = 0.02631 \pm 0.00045$ $\boldsymbol{r^2 = 0.9991}$
0.9	$\delta = 0.4734 \pm 0.0074$ $\boldsymbol{r^2 = 0.9983}$	$\delta = 0.02583 \pm 0.00043$ $\boldsymbol{r^2 = 0.9992}$
1.0	$\delta = 0.7784 \pm 0.012$ $\boldsymbol{r^2 = 0.9983}$	$\delta = 0.00016 \pm 0.00083$ $\boldsymbol{r^2 = 0.01279}$

Table 5.1: Exponents for power law fit of the average final cost obtained $\overline{C}_{SA}$ as a function of n_{total} in both regions.

Now we examine how the performance of the heuristic behaves as we vary the correlation ρ_{xy} for different system sizes (i.e., different values of N). First, we consider simulated annealing with swap moves in figure 5.13.

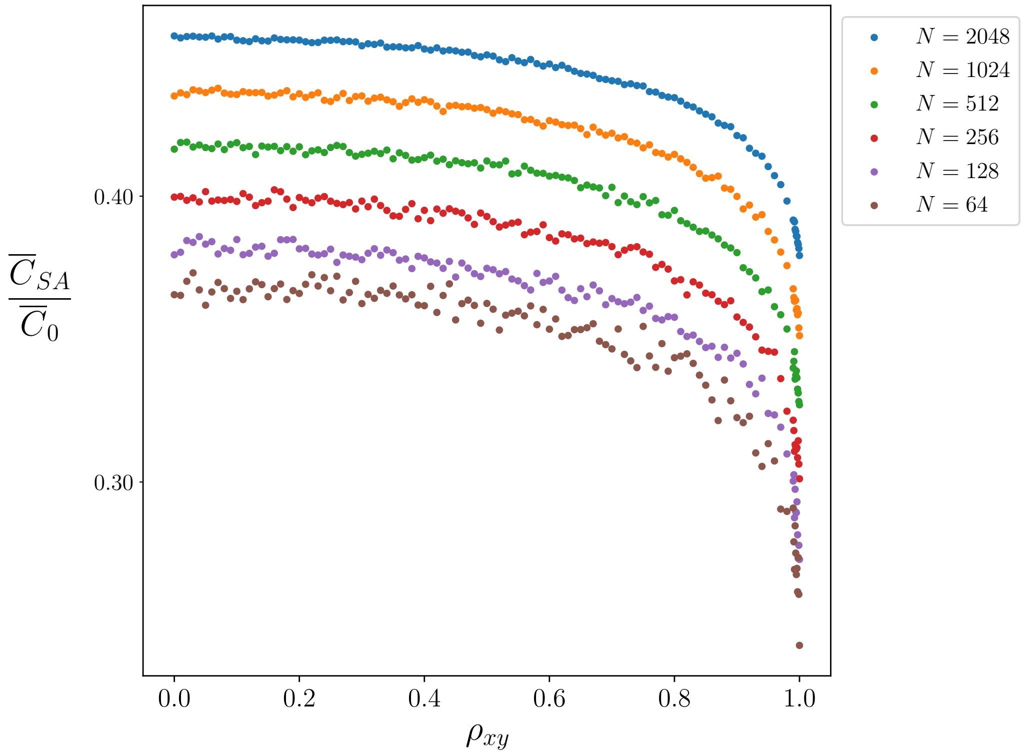


Figure 5.13: Performance of the simulated annealing using swap moves as function of ρ for different number of cities.

We observe a performance improvement as $\rho_{xy} \to 1$ which was already apparent from the low temperature behavior shown in figure 5.11. Additionally, the performance decreases as the number of points in the system grows. In fact, a reasonable empirical scaling on the system size of the performance curves of the form can be observed in figure 5.14. Choosing $b = \frac{N_{max}}{N}$, where $N_{max} = 2048$ is the size of the largest system considered, the base $z \approx 0.0760$ allows for a collapse of the curves $z^b \frac{\overline{C}_{SA}}{\overline{C}_0}$.

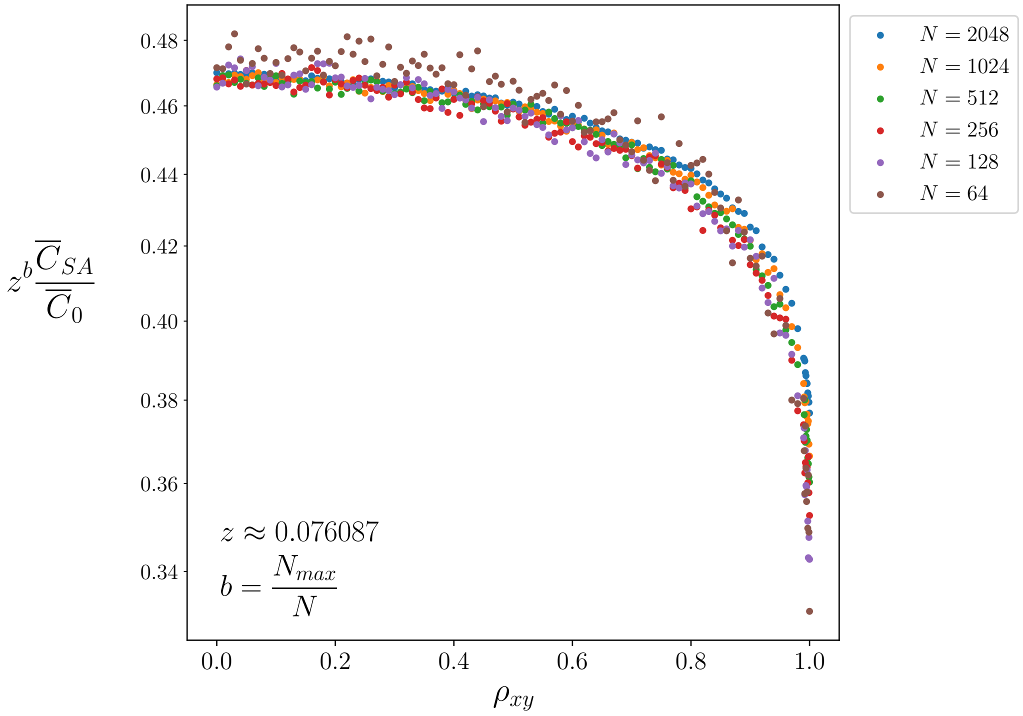


Figure 5.14: An approximate scaling on the system size for the performance of the simulated annealing with swap moves as a function of the correlation ρ_{xy}.

The same procedure is followed in figures 5.15 and 5.16, this time considering the 2-opt move. The base for the scaling for the 2-opt move is $z \approx -0.502$, a negative value, since for this move the performance improves with the size of the system.

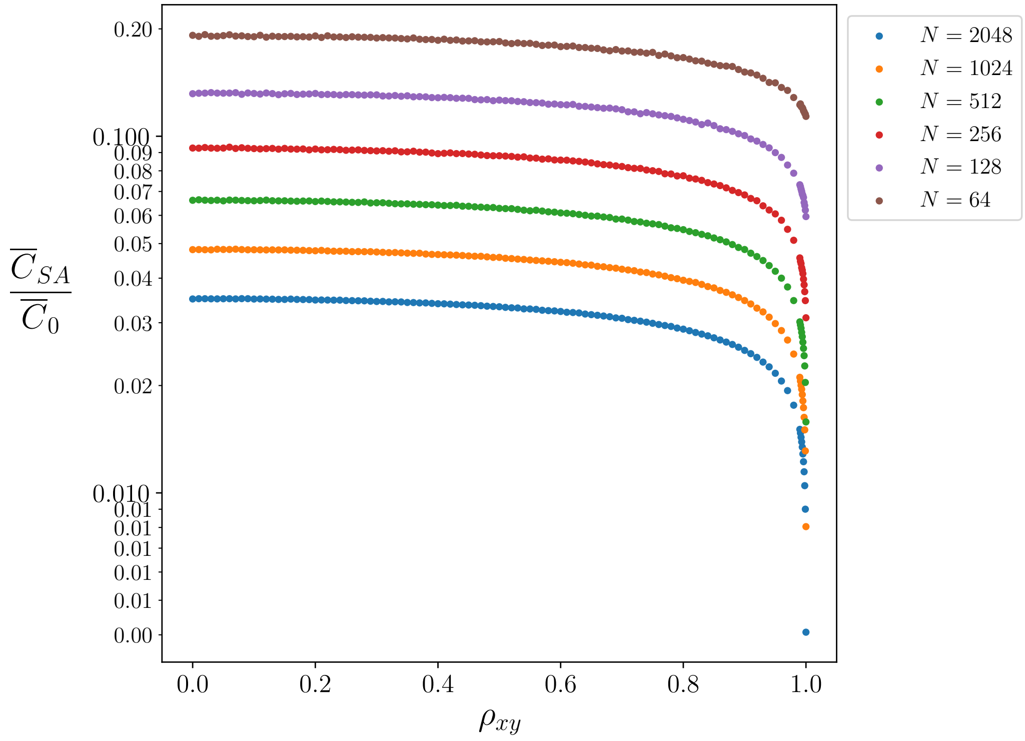


Figure 5.15: Performance of the SA as function of ρ for different number of cities.

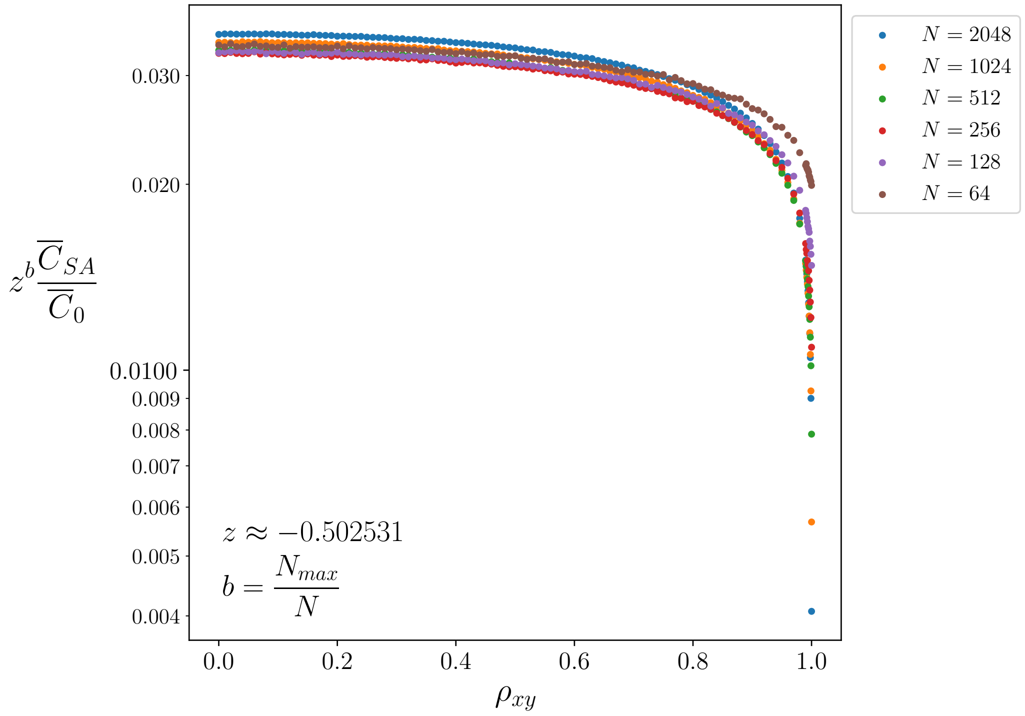


Figure 5.16: An approximate scaling on the system size for the performance of the simulated annealing with 2-opt moves as a function of the correlation ρ_{xy}.

As for the dependence of the performance on the system size, another power law behavior can be identified:

$$\frac{\overline{C}_{SA}}{\overline{C}_0} \propto N^{-\delta}, \tag{5.15}$$

with a different exponent δ for each value ρ_{xy}. Figure 5.17 shows how the heuristic performance behaves as function of the system size highlighting the exponents δ for both of the extreme cases ($\rho_{xy} = 0$ and $\rho_{xy} = 1$).

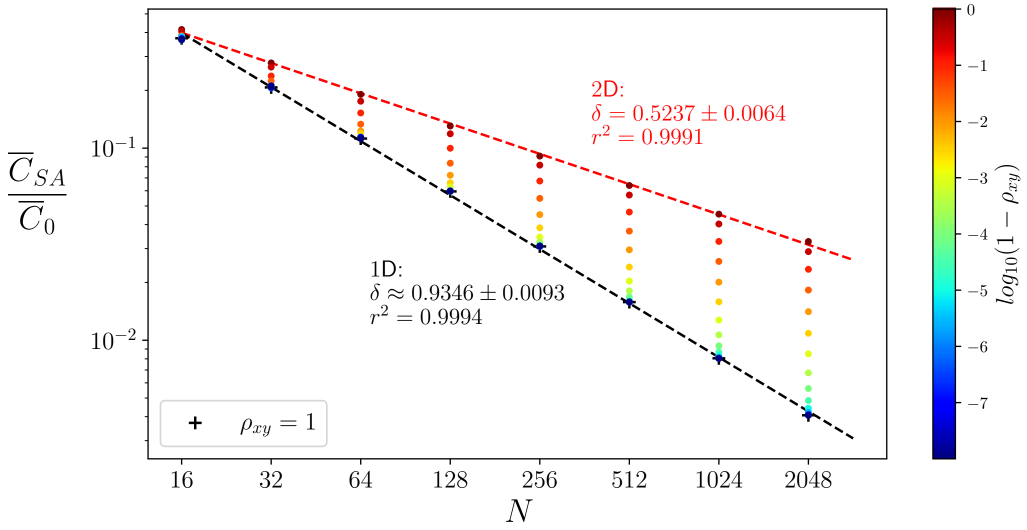


Figure 5.17: Performance of the simulated annealing with 2-opt moves as a function of system size N for different values of correlation ρ_{xy}. Power law fits were performed for both the one dimensional case ($\rho_{xy} = 1$, in black) and the two dimensional case ($\rho_{xy} = 0$, in red).

The exponent goes from $\delta \approx 0.52$ for the two dimensional case (we note that it is, as expected, very close to the one calculated in figure 5.8) to $\delta \approx 0.93$ for the one dimensional case.

We also attempt to find good fits for the performance of the simulated annealing $\frac{\overline{C}_{SA}}{\overline{C}_0}$ as a function of ρ_{xy}. In the following we make the change $\rho_{xy} = \rho$ to simplify the notation. As a preliminary investigation, we attempt a polynomial fit:

$$p_n(\rho) = \sum_{k=0}^{n} a_k \rho^k, \tag{5.16}$$

by testing the orders $n = 2$, 3 and 4 (with 3, 4 and 5 parameters, respectively).

Then, we test exponential decay fits with only two parameters:

$$e_I(\rho) = e_I(0) + c_1 e^{-\rho/\rho_1}, \tag{5.17}$$

where the term $e_I(0)$ is fixed and equal to the data point value $\frac{\overline{C}_{SA}}{\overline{C}_0}(\rho = 0)$.

Next, we consider a simple linear combination of exponential decays (using 4 parameters):

$$e_{II}(\rho) = e_{II}(0) + c_1 e^{-\rho/\rho_1} + c_2 e^{-\rho/\rho_2} \tag{5.18}$$

also assuming $e_{II}(0) = \frac{\overline{C}_{SA}}{\overline{C}_0}(\rho = 0)$.

Alternatively, other functions with 4 parameters were explored and the one that yielded a good result was the rational function:

$$r(\rho) = \frac{a + b\rho}{1 + c\rho + d\rho^2} \tag{5.19}$$

The results are shown in figures 5.18 and 5.19 for the performance of simulated annealing with swap and 2-opt moves respectively. The values for the coefficient along with the coefficient of determination for each fit are summarized in the table 5.2.

As expected, the higher the degree of the polynomial fitting the function, the better the coefficient of determination r^2 (the closer to one, the better it is). Our analysis also shows that in both the swap and 2-opt cases, the single exponential (labeled Exponential I, requiring 2 parameters) is not enough to nicely fit the curve.

Using the rational function shown in equation (5.19) a good fit can be obtained using 4 parameters. It is important to mention that the coefficient a found in both cases is exactly $\frac{\overline{C}_{SA}}{\overline{C}_0}(\rho = 0)$ so the fit can effectively be performed using only the remaining 3 parameters.

However the best result is achieved using the two exponentials from equation (5.18) (labeled Exponential II, requiring 4 parameters) yielding a fit with coefficient of determination $r^2 \approx 0.999$ for both cases. This linear combination of exponentials seems to be a universal fit for $\frac{\overline{C}_{SA}}{\overline{C}_0}(\rho)$ for both the simulated annealing with swap and 2-opt moves.

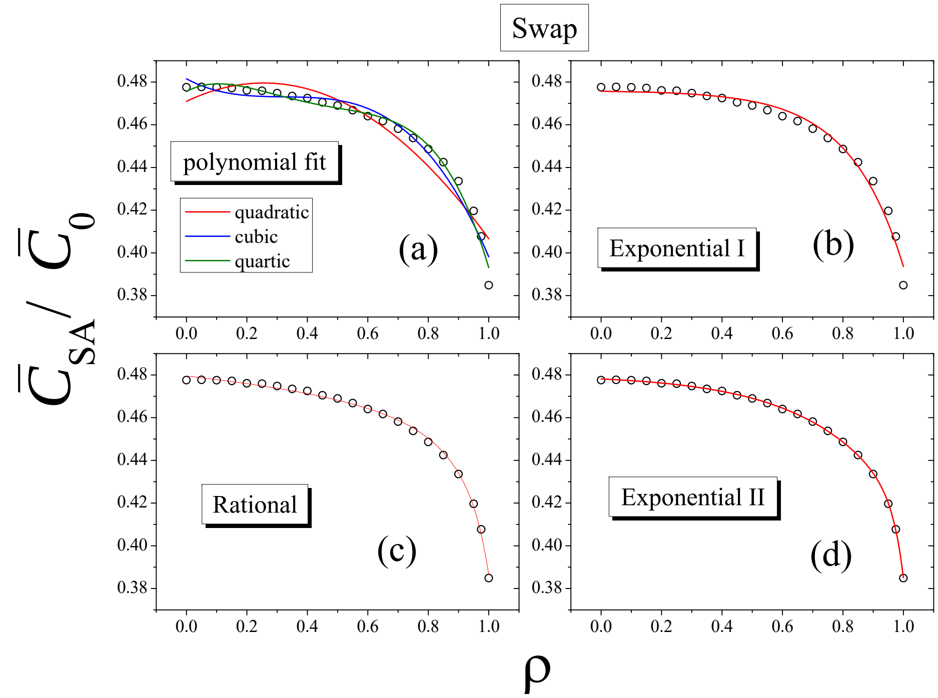


Figure 5.18: Fits for the performance of the simulated annealing with swap moves versus ρ: (a) preliminary polynomial fit, (b) simple exponential, (c) a reasonable fit with rational function and (d) the best fit obtained by combining two exponentials.

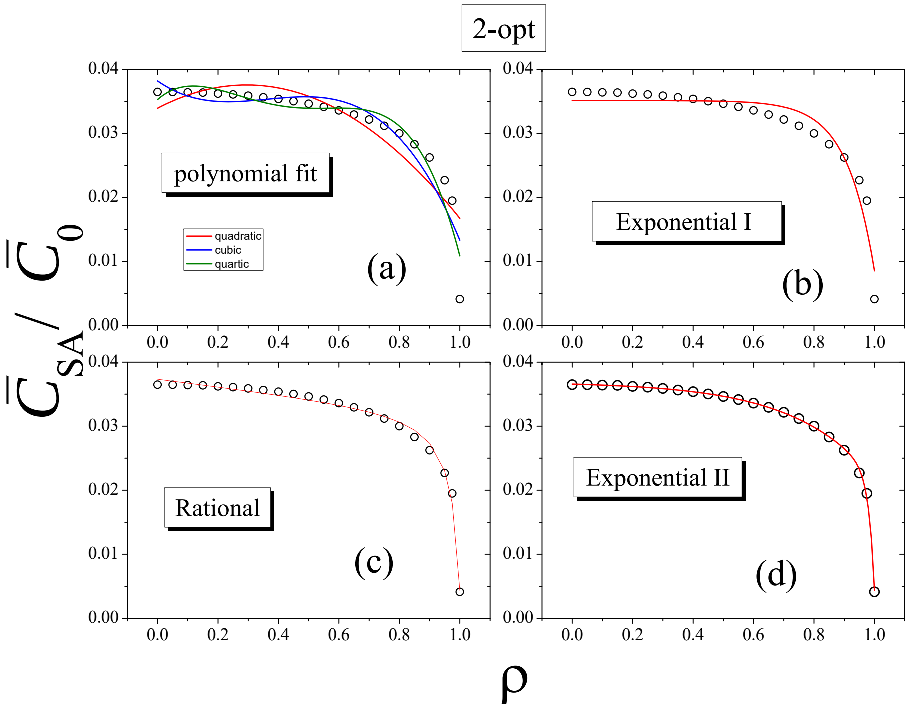


Figure 5.19: Fits for the performance of the simulated annealing with simple 2-opt versus ρ: (a) preliminary polynomial fit, (b) simple exponential, (c) a reasonable fit with rational function and (d) the best fit obtained by combining two exponentials.

Move	Quadratic	Cubic	Quartic
Swap	$a_0 = 0.4710(42)$ $a_1 = 0.067(19)$ $a_2 = -0.132(18)$ $\boldsymbol{r^2 = 0.92124}$	$a_0 = 0.4815(34)$ $a_1 = -0.076(30)$ $a_2 = 0.230(70)$ $a_3 = -0.237(45)$ $\boldsymbol{r^2 = 0.96729}$	$a_0 = 0.4755(26)$ $a_1 = 0.077(38)$ $a_2 = -0.49(16)$ $a_3 = 0.90(24)$ $a_4 = -0.57(12)$ $\boldsymbol{r^2 = 0.98495}$
2-opt	$a_0 = 0.0339(21)$ $a_1 = 0.0247(96)$ $a_2 = -0.0419(91)$ $\boldsymbol{r^2 = 0.78373}$	$a_0 = 0.0382(21)$ $a_1 = 0.0330(19)$ $a_2 = 0.104(43)$ $a_3 = -0.096(28)$ $\boldsymbol{r^2 = 0.86162}$	$a_0 = 0.0353(20)$ $a_1 = 0.041(29)$ $a_2 = -0.25(12)$ $a_3 = -0.46(19)$ $a_4 = -0.275(92)$ $\boldsymbol{r^2 = 0.90423}$
Move	Exponential I	Rational	Exponential II
Swap	$c_1 = -3.1(1.2) \cdot 10^{-4}$ $\rho_1 = 0.179(13)$ $\boldsymbol{r^2 = 0.98309}$	$a = 0.47951(65)$ $b = -0.4443(29)$ $c = -0.8923(39)$ $d = -0.0165(51)$ $\boldsymbol{r^2 = 0.99798}$	$c_1 = -0.00190(13)$ $\rho_1 = -0.2852(62)$ $c_2 = -(2.4 \pm 3.3) \cdot 10^{-15}$ $\rho_2 = -0.0331(15)$ $\boldsymbol{r^2 = 0.99984}$
2-opt	$c_1 = -4.9(7.9) \cdot 10^{-7}$ $\rho_1 = -0.092(14)$ $\boldsymbol{r^2 = 0.93539}$	$a = 0.03733(34)$ $b = -0.03719(35)$ $c = -0.832(24)$ $d = -0.136(25)$ $\boldsymbol{r^2 = 0.99312}$	$c_1 = -1.5(5.4) \cdot 10^{-26}$ $\rho_1 = -0.0180(12)$ $c_2 = -3.27(61) \cdot 10^{-4}$ $\rho_2 = -0.261(14)$ $\boldsymbol{r^2 = 0.99914}$

Table 5.2: Values of coefficients found for nonlinear fitting using Levenberg-Marquadt method. The parameter r^2 corresponds to the coefficient of determination for each fit.

Finally, we move on to consider the effects of the shape of the distribution used to draw the coordinates. If before we considered z_1 and z_2 drawn from identically distributed uniform distributions, now we also consider them as identically distributed Gaussian random variables. It is important to mention once again that the x and y coordinates generated have the same average (equal to zero) and variance of the variables z_1 and z_2 regardless of the value chosen for ρ_{xy}. As previously stated, the effects observed are only dependent on ρ_{xy} since average and variance remain unchanged.

Hence, in figure 5.20 we plot the performance of both the simulated annealing using swap and 2-opt moves when considering the following distributions:

- Uniform distribution with $\sigma = \frac{1}{\sqrt{12}}$
- Gaussian distribution with $\sigma = 1$
- Gaussian distribution $\sigma = 1$ using scaled costs inside the Metropolis prescription of the form

  $$C_s = \frac{C}{\sqrt{(x_{max} - x_{min})(y_{max} - y_{min})}} \tag{5.20}$$

  where $x(y)_{max(min)} = max(min)\{x(y)_1, x(y)_2, \cdots, x(y)_N\}$. This is used to take into account the fact that, in contrast to the uniform distribution, the normal distribution does not have finite support.
- Gaussian with same standard deviation as the uniform distribution: $\sigma = \frac{1}{\sqrt{12}}$

We can observe that in all cases, the shape of the distribution plays a much more important role than the variance since the Gaussian distributions with different variances display practically the same behavior while the uniform and Gaussian distributions with the same variance produce very different results. Additionally the scaling performed for the costs in the case of the Gaussian distribution shows no effect.

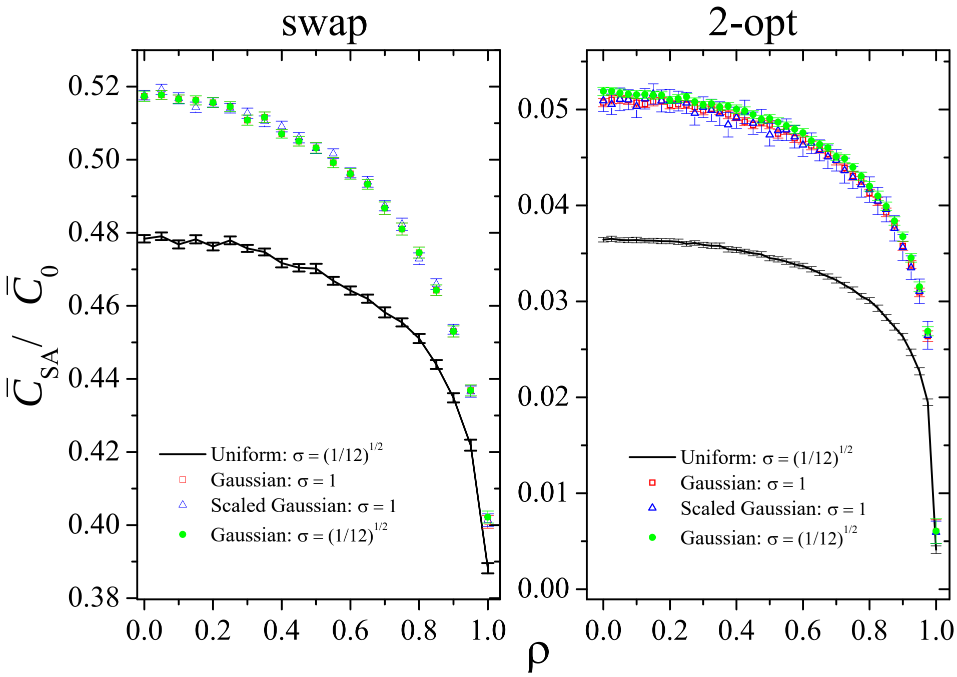


Figure 5.20: Analysis for different distributions of $N = 2048$ points: Gaussian and uniform. The plot on left corresponds to the simulated annealing with swap moves and the the plot on the right corresponds to the one using with 2-opt moves.

5.2.2 Long tail effects

Finally, we analyze the effects of coordinates drawn from long-tailed distributions on the performance of the simulated annealing heuristic. Thus, we employ the power law probability density function from equation (4.7) to generate the coordinates of the N points in dimensions in both radially symmetric and axially symmetric configurations.

We start by considering the simulated annealing with swap moves applied to the axially symmetric power law distribution of points generated by equation (4.10). The scaling of equation (5.20) is used here in order to take into account the fact that the power law distribution also does not have finite support.

Figure 5.21 shows the performance of the simulated annealing with swap moves as a function of γ for different values of x_0 for the axially symmetric domain, the number of points is $N = 2048$. The inset plot does not take into account the scaling.

The first thing we observe is that the performance improves as the exponent γ gets higher. There is also a noticeable instability in the region $2 \leq \gamma \leq 3$ where the variance is not defined but the average is. For $\gamma < 2$, that corresponds to the region where the average also cannot be defined, the performance is negligible with overwhelming instability which suggests that the heuristic should not be applied to such instances. Interestingly, the scaling results in a collapse of all curves into one, regardless of the value of x_0.

In the more detailed figure 5.22 the error bars show the instability for the low γ region. Also in this picture we identify the values of γ for which the performance coincides with that of the Gaussian and uniform distribution analyzed before.

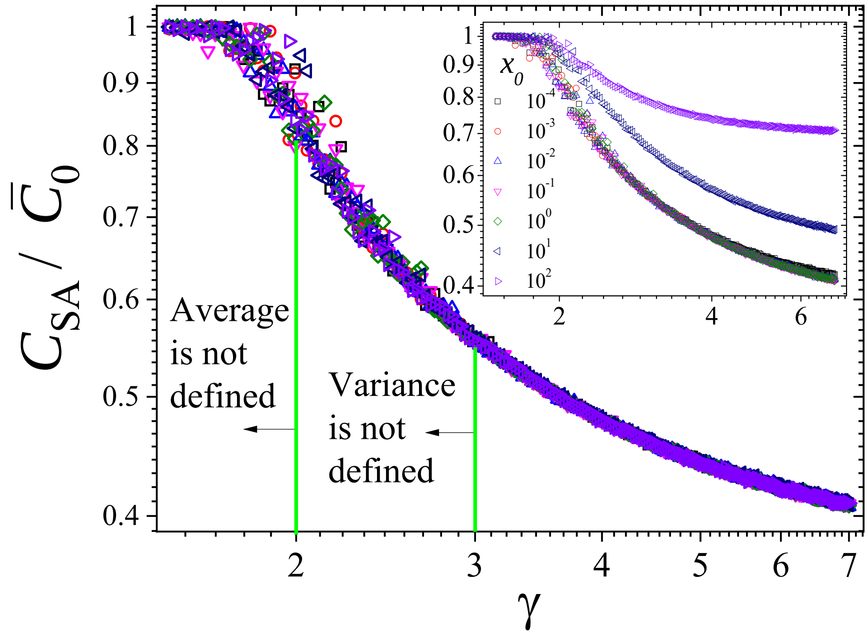


Figure 5.21: Performance of the simulated annealing with swap moves applied to the system with axially symmetric power law coordinates as a function of γ for different values of x_0 using the scaling of equation (5.20). The inset plot shows the same curves without the scaling.

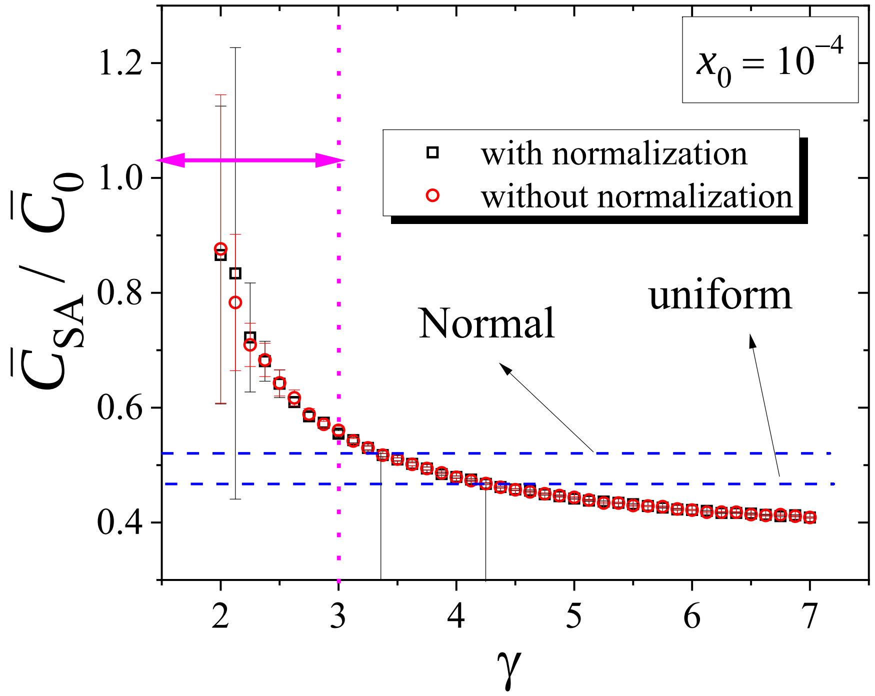


Figure 5.22: Refinement of the region presenting the error bars. We observe that for $\gamma \approx 3.4$ we obtain the same improvement as for the Gaussian distribution and that $\gamma \approx 4.3$ has an improvement equal to that obtained for the uniform distribution.

Finally we consider the simulated annealing with 2-opt moves applied to both the axially symmetric and radially symmetric power law point distributions. For the axially symmetric domain (figure 5.23), we can draw basically the same conclusions as before: the increase of performance as γ grows, the small instability with some performance for $2 \leq \gamma \leq 3$ and the lack of improvement for $\gamma < 2$. Surprisingly, the performance for the radially symmetric distribution (shown in figure 5.24) exhibits essentially the same behavior, hinting that the phenomenon of the performance improvement of the simulated annealing is universal as a function of γ. In both configurations the scaling is able to take into account the effect of different values of x_0.

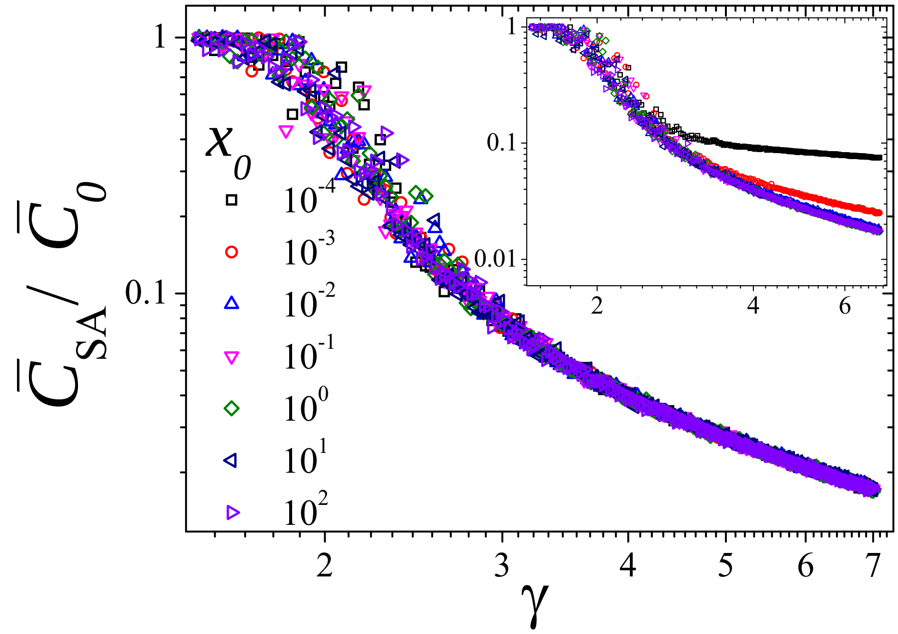


Figure 5.23: Improvement of the performance for simulated annealing with 2-opt moves applied to the system with axially symmetric power law coordinates (equation 4.10) as function of γ for different values of x_0 using the scaling of equation (5.20). The inset plot shows the same curves without scaling.

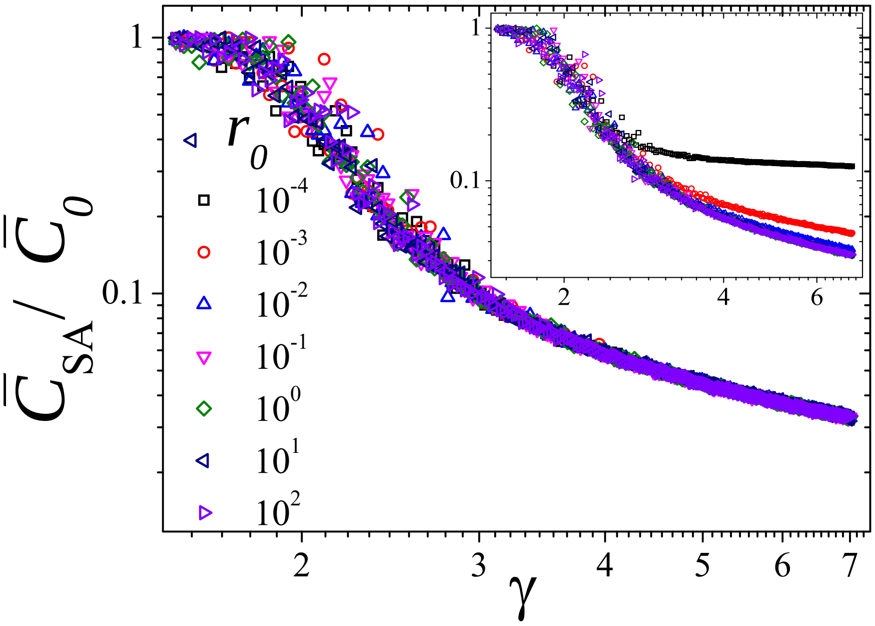


Figure 5.24: Improvement of the performance for simulated annealing with 2-opt moves applied to the system with radially symmetric power law coordinates (equation 4.11) as function of γ for different values of x_0 using the scaling of equation (5.20). The inset plot shows the same curves without scaling.

Chapter 6

Conclusion

In this work we explored an example of the use of the framework provided by statistical physics applied to a hard combinatorial optimization problem. More specifically a standard version of the simulated annealing heuristic applied to the traveling salesman problem.

We were able to determine that the 2-opt move is able to explore the state space better and as a consequence achieve lower costs when compared to the swap move in all systems considered. For the domain formed by uncorrelated and uniform random points we identified two power laws: one regarding the performance of the heuristic and the system size and the other regarding the performance and the total number of samples generated by the heuristic.

As the main focus of this work, we analyzed how the statistics of the distribution of points impacts the performance of the heuristic. First, we considered the correlation between the coordinates which is equivalent to a dimensionality reduction (2D to 1D), essentially transforming the problem from a TSP into a sorting problem. We showed that the performance of the simulated annealing increases as the correlation increases and performed a reasonable finite size scaling for this data as well as experimented with some functions in order to fit its behavior. We also identified a power law behavior of the performance on the total number of samples generated showing that after a given point there is no meaningful improvement. Additionally, we identified power law fits for the performance as a function of the system size for both the one dimensional end two dimensional case. Furthermore, when using different types of distributions, we found that the shape played a more important role than its variance.

Finally, considering the long tail effects on the coordinates, we observed that the higher the exponent of the power law, the better the performance of the heuristic. For the special case with $\gamma < 2$, i.e., distributions without the first moment, we observe no cost improvement at all. We were also able to obtain an universal behavior of the performance as a function of the power law exponent regardless of whether the distribution is axially symmetric or radially symmetric.

A considerable part of the results obtained in this dissertation can be found in our recent publication: R. da Silva, E. Venites Filho, A. Alves, Physica A 577, 126067 (2021).

Bibliography

[1] A. Percus, G. Istrate, and C. Moore, *Computational Complexity and Statistical Physics*. Proceedings volume in the Santa Fe Institute studies in the sciences of complexity, OUP USA, 2006.

[2] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, (New York, NY, USA), p. 151–158, Association for Computing Machinery, 1971.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction To Algorithms*. Introduction to Algorithms, MIT Press, 2001.

[4] J. Schneider and S. Kirkpatrick, *Stochastic Optimization*. Scientific Computation, Springer Berlin Heidelberg, 2006.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671–680, may 1983.

[6] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, jan 1985.

[7] U. H. E. Hansmann and Y. Okamoto, "Tackling the protein folding problem by a generalized-ensemble approach with Tsallis statistics," *Brazilian Journal of Physics*, vol. 29, pp. 187 – 198, mar 1999.

[8] F. Mauger, C. Chandre, and T. Uzer, "Simulated annealing algorithm for finding periodic orbits of multi-electron atomic systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 7, pp. 2845–2852, 2011.

[9] G. Dantzig, R. Fulkerson, and R. Johnson, “Solution of a large-scale traveling-salesman problem,” *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[10] W. J. Cook, *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation.* Princeton University Press, 2011.

[11] M. Padberg and G. Rinaldi, “A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems,” *SIAM Rev.*, vol. 33, p. 60–100, Feb. 1991.

[12] A. H. Land and A. G. Doig, “An automatic method of solving discrete programming problems,” *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.

[13] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the tsp,” *Discrete Applied Mathematics*, vol. 117, no. 1, pp. 81–86, 2002.

[14] H. E. Stanley and S. V. Buldyrev, “The salesman and the tourist,” *Nature*, vol. 413, pp. 373–374, sep 2001.

[15] G. F. Lima, A. S. Martinez, and O. Kinouchi, “Deterministic walks in random media,” *Phys. Rev. Lett.*, vol. 87, p. 010603, jun 2001.

[16] C. Voudouris and E. Tsang, “Guided local search and its application to the traveling salesman problem,” *European Journal of Operational Research*, vol. 113, no. 2, pp. 469–499, 1999.

[17] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[18] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[19] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of Statistical Physics*, vol. 34, pp. 975–986, Mar. 1984.

[20] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, jun 1953.

[21] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics.* Clarendon Press, 1999.

[22] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics.* Cambridge University Press, 2015.

[23] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues.* Texts in Applied Mathematics, Springer International Publishing, 2020.

[24] T. P. Peixoto, "The graph-tool python library," *figshare*, 2014.

[25] R. da Silva, *Universalidade e leis de escala em sistemas fora do equilíbrio.* PhD thesis, FFCLRP-USP, Av. Bandeirantes, 3900 - Vila Monte Alegre, Ribeirão Preto - SP, Brazil, 12 2002. In portuguese.

[26] H. Szu and R. Hartley, "Fast simulated annealing," *Physics Letters A*, vol. 122, pp. 157–162, jun 1987.

[27] P. J. van Laarhoven and E. H. Aarts, *Simulated Annealing: Theory and Applications.* Mathematics and Its Applications, Springer Netherlands, 1987.

[28] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *Journal of Physics A Mathematical General*, vol. 31, pp. 8373–8385, Oct. 1998.

[29] R. Azencott, *Simulated Annealing: Parallelization Techniques.* Wiley Series in Discrete Mathematics and Optimization, Wiley, 1992.

[30] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, 1993.

[31] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 721–741, nov 1984.

[32] E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* Wiley Series in Discrete Mathematics & Optimization, Wiley, 1989.

[33] R. da Silva and S. D. Prado, "A simple study of the correlation effects in the superposition of waves of electric fields: The emergence of extreme events," *Physics Letters A*, vol. 384, no. 11, p. 126231, 2020.

[34] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* USA: Cambridge University Press, 3 ed., 2007.

[35] J. Beardwood, J. H. Halton, and J. M. Hammersley, "The shortest path through many points," *Proceedings of the Cambridge Philosophical Society*, vol. 55, p. 299, Jan. 1959.

[36] S. Steinerberger, "New bounds for the traveling salesman constant," *Advances in Applied Probability*, vol. 47, no. 1, pp. 27–36, 2015.

Appendix A

Average Distance Between Two Uniformly Distributed Points on a Square

Considering points uniformly distributed on a square $S = [0, L]^2$ of sides L, the average distance between two points is:

$$\begin{aligned}
\langle d \rangle &= \frac{1}{L^4} \int_S dx_i dy_i \int_S dx_j dy_j \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\
&= \frac{1}{L^3} \int_0^L dx_i \int_0^L dy_i \int_0^L dx_j \int_0^L dy_j \sqrt{\left(\frac{x_i - x_j}{L}\right)^2 + \left(\frac{y_i - y_j}{L}\right)^2}
\end{aligned}$$

Performing the change of variables $z_1 = \frac{x_i}{L}$, $z_2 = \frac{x_j}{L}$, $z_3 = \frac{y_i}{L}$, and $z_4 = \frac{y_i}{L}$ yields

$$\langle d \rangle = L \int_0^1 dz_1 \int_0^1 dz_2 \int_0^1 dz_3 \int_0^1 dz_4 \sqrt{(z_1 - z_2)^2 + (z_3 - z_4)^2} \qquad \text{(A.1)}$$

Here we see that the average distance between two points is indeed the length L of each side of the square times the average distance for a unit square.

Since z_1 and z_2 are uniformly distributed, the random variable $u = |z_1 - z_2|$ has a triangular probability density function $2(1 - u)$. The same applies to

$u = |z_3 - z_4|$ and thus considering the distribution of the differences u and v the integral becomes:

$$\begin{aligned}\langle d\rangle &= L\int_0^1 du\, 2(1-u)\int_0^1 dv\, 2(1-v)\sqrt{u^2+v^2}\\ &= 4L\int_0^1 du\int_0^1 dv\,(1-u)(1-v)\sqrt{u^2+v^2}\end{aligned}$$

We now note that the integrand is symmetric with respect to the $u = v$ line and hence the integral can be performed only on the right triangle below this line.

$$\langle d\rangle = 8L\int_0^1 du\int_0^u dv\,(1-u)(1-v)\sqrt{u^2+v^2} \tag{A.2}$$

Changing from the Cartesian coordinates (u, v) to the polar coordinates (r, θ) the integral becomes

$$\begin{aligned}\langle d\rangle &= 8L\int_0^{\frac{\pi}{4}} d\theta\int_0^{\sec\theta} r\,dr\,(1-r\cos\theta)(1-r\sin\theta)r\\ &= 8L\int_0^{\frac{\pi}{4}} d\theta\int_0^{\sec\theta} dr\;\left[r^2 - r^3(\cos\theta+\sin\theta) + r^4\cos\theta\sin\theta\right]\\ &= 8L\int_0^{\frac{\pi}{4}} d\theta\;\left[\frac{\sec^3\theta}{3} - \frac{\sec^4\theta}{4}(\cos\theta+\sin\theta) + \frac{\sec^5\theta}{5}\cos\theta\sin\theta\right]\\ &= 8L\int_0^{\frac{\pi}{4}} d\theta\;\left[\frac{\sec^3\theta}{10} - \frac{\sec^4\theta\sin\theta}{20}\right]\\ &= \frac{4L}{5}\int_0^{\frac{\pi}{4}} d\theta\;\left[\sec^3\theta - \frac{\sec^3\theta\tan\theta}{2}\right]\end{aligned}$$

The first term is a well know integral and can be found to be

$$\int d\theta\,\sec^3\theta = \frac{1}{2}\left(\sec\theta\tan\theta + \ln|\sec\theta+\tan\theta|\right) + C \tag{A.3}$$

the second term can be easily calculated using the substitution $w = \sec\theta$ and evaluates to

$$\int d\theta\,\sec^3\theta\tan\theta = \frac{\sec^3\theta}{3} + C \tag{A.4}$$

Finally, the average distance is

$$\langle d \rangle = \frac{L}{15}\left(2 + \sqrt{2} + 5\ln(\sqrt{2}+1)\right) \approx 0.5214L \tag{A.5}$$