An ACO Algorithm and its Application to The Travelling Salesman Problem

Elmo Barratt

Supervised by Dr Ayalvadi Ganesh
Level 4
20 Credit Points

May 9, 2022

Acknowledgement of Sources

Acknowledgement of Sources

For all ideas taken from other sources (books, articles, internet), the source of the ideas is mentioned in the main text and fully referenced at the end of the report.

All material which is quoted essentially word-for-word from other sources is given in quotation marks and referenced.

Pictures and diagrams copied from the internet or other sources are labelled with a reference to the web page or book, article etc.

Signed Elmo Barn

Date 09/05/2022

Contents

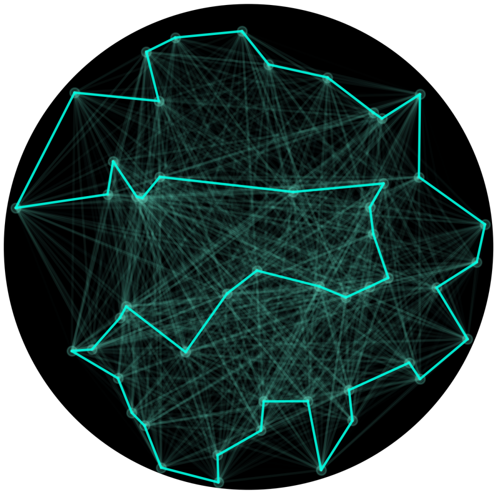


Figure 1: An ACO derived solution to the TSP on K_{50}^D

1 Introduction

1.1 Motivation

In this project we study a recursive algorithm that can tackle large solution space combinatorial optimisation problems which would otherwise be infeasible to solve through the use of brute force exhaustive searches. There are a number of well documented mathematical problems in which Ant Colony Optimisation (ACO) is effective, with one particular application of the algorithm studied in great depth throughout this paper - the *Travelling Salesman Problem*. It gives great intuition into both the mechanics of this algorithm and its source of biological inspiration. We will explore how under certain conditions, ACO can do better than just approximating solution but in fact converge to global optimal solution with probability arbitrarily close to 1. Finally, we will discuss the limitations of the theoretical results with a supplementary experimental investigation running simulations to try to understand the behaviour of ACO algorithms.

The idea behind the algorithm is simple, yet effective, and takes inspiration from the decentralised communicative behaviour of real animal populations based on interaction with environment and surroundings rather than directly with one another. This process is referred to as stigmergy and often, the purpose of it is to encourage members of the species to change their behaviour in a more favourable way, thus leading to greater success for the collective population. Stigmergy is seen across many different species of insects, with the most notable example being that of foraging ants and outlines the disproportionately greater power of swarm intelligence when compared to the sum of the intelligence of the individuals.

1.2 Biological Inspiration

The self organising behaviour of large colonies of ants bewildered scientists as it was not obvious how simple autonomous agents, often limited in capacity of traditional communicative 'senses' such as vision or acoustics, might form complex structures several orders of magnitude larger than the individuals themselves. Early research showed that it was via the use of pheromones, myriad fine-tuned chemical cocktails deposited into the environment, that ants relay specific signals to other members of its species. One particular well documented illustration of this is the so called 'double bridge' experiment [2].

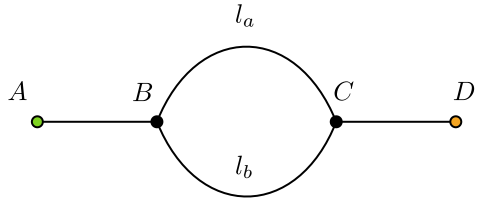


Figure 2: The Double Bridge experiment: a simple illustration of how ants may optimise foraging paths analogous to real world behaviour

The double bridge experiment is schematically depicted in Figure 2. The setup shows an Argentine ant (*Linepithema humile*) colony positioned at A and a food source postioned at B. The ants have the goal of foraging for food and then to return to the nest without the ability of sensing food sources at distances further than their immediate surrounding.

An ant leaving A and travelling to B is given the option to traverse BC via path a or path b, of length l_a and l_b respectively. Initially, as the ant cannot sense which path may be shorter (or if indeed it does lead to a food source), the two paths are chosen uniformly at random. Once at C, the ant continues onto D where food is found. Now, the ant will collect as much food as it can carry and return to the nest. During this journey, it will deposit trail pheromones along the path to signal to other ants a route to food. The process is then repeated with the new assumption that when faced with a decision between two paths to take, the probability of choosing a specific path is proportional to the pheromone concentration on it. The experiment considered two cases.

In the first case $l_a = l_b$, thus there was no difference between path a and path b. It was observed that, initially, ants would not favour either path but eventually all ants would converge to the primary usage of one path. It was of equal chance whether path a or b would be selected for convergence when the experiment was repeated, however in the long term there was never a significant split in ant traffic between the two paths. The explanation of this was as follows; since both paths were of the same length, it would take the same amount of time for each path to be traversed, thus initially pheromone concentration would be about the same on either path. However, there exists a positive feedback loop mechanism whereby initially small stochastic fluctuations in pheromone concentration along one path lead to higher probability of selection and in turn more ants traversing it leading to a higher concentration of pheromones along it and so on. This leads to convergence on one path but it is the uniform random fluctuations at the beginning which determine whether path a or b is converged upon.

In the second case one path was longer than the other, say without loss of generality $l_a > l_b$. As before, it was observed initially that ants would randomly select either path with equal chance as no pheromone lay on either. However, ants that chose b would get to the food first, and thus back to the colony faster, as l_b is a shorter distance. This means pheromones can be deposited on path b at a greater rate and as a result, future ants selected this path with higher probability. Then via the same positive feedback loop as described above, convergence to path b occurred in almost all of the experimental trials. In the long term, even though most ants would choose shorter path b, it is a real-world system thus the event that the longer path a is chosen is - although small - a non-zero probability event, so some ants may still choose this. Although this may seem like a flaw in the time allocation of the ant workforce with unoptimised productivity, it is in fact a crucial factor that allows for the self organising structure to remain dynamic and nonrigid.

This was demonstrated when an analogous single bridge experiment was conducted with only one path for ants to follow from colony to food. Once the ants had formed a strong pheromone trail a second path was introduced of shorter length than the original one. It was the initially low-probability decisions of some rogue ants to chose this unfamiliar, pheromone-baron path which then allowed for a new more efficient route to be discovered. The same positive feedback mechanism combined with pheromone evaporation then led to a total switch of preferred route and convergence to the new shorter path. This shows how the randomness allowed for convergence towards globally optimal routes over what seemed to be the locally optimal route.

We can draw inspiration from these swarm-intelligent, self organising systems to tackle many mathematical optimisation problems by setting relatively simple parameters for agents performing some stochastic process with the ability to iteratively and dynamically update those parameters.

1.3 A generalised Ant Colony Optimisation Algorithm

After research into the self-organising behaviour of Argentine ants was published [2], it set the scene for a whole new class of heuristic - an algorithm used for the partial search of a solution space where certainty in the optimality of solution is traded off against speed and reduced computational load - first formally outlined by Dorigo et al [1] . The ideas discussed in this section are a generalisation of those he proposed. Before we begin, we must first put a rigorous definition on the type of problem we aim to solve using ACO. The class of mathematical problem is referred to as a combinatorial optimisation problem and is defined as such.

Definition 1.0.1 (**Combinatorial Optimisation Problem**)**.** A combinatorial optimisation problem is defined as the triple $(\mathbf{W}, \Omega, f)$ where:

- Ω is the set of constraints to the problem.
- $\mathbf{W}$ is the solution space and $\mathbf{W}_\Omega \subseteq \mathbf{W}$ is the feasible solution space subject to given constraints.
- $f : \mathbf{W}_\Omega \to \mathbb{R}$ is the cost objective function we aim to minimise.

We then note the optimal solution $\mathbf{w}^* \in \mathbf{W}_\Omega$ satisfies $f(\mathbf{w}^*) \leq f(\mathbf{w})$ (strict inequality if $\mathbf{w}^*$ is unique) for all $\mathbf{w} \in \mathbf{W}_\Omega \setminus \{\mathbf{w}^*\}$. The goal of an Ant Colony algorithm is to find the optimal solution to a combinatorial optimisation problem through non-exhaustive, and therefore computationally less intensive, methods. In this section we will describe the intuition behind a generic Ant Colony Algorithm in a non-rigorous sense before later going onto a rigorous definition in the context of an application.

Definition 1.0.2 (**Generalised Ant Colony Optimisation Algorithm**)**.** A generalised ACO algorithm is a recursive algorithm with the following components:

- Combinatorial optimisation problem $(\mathbf{W}, \Omega, f)$ we intend to solve.
- A finite set $A = \{A_1, A_2, \ldots, A_S\}$ containing the colony of identical agents/ants (these terms can be used interchangeably) with population size S.
- Iteration number m with $1 \leq m \leq M$ where the final iteration number of the algorithm M may be predetermined or decided under some stopping condition during execution. The iteration number can be thought of as the discrete time index for the algorithm.
- 'Pheromone' based update rule including evaporation constant $\rho \in (0, 1)$ which adjusts the influential strength that the results of a given iteration have on behaviour of the algorithm in succeeding iterations.

We begin the algorithm by assigning an initial discrete probability distribution, $\mu_1(\cdot)$, to each element of the feasible solution space, $\mathbf{w} \in \mathbf{W}_\Omega$. In the double bridge experiment, figure 2, the solution space is $\mathbf{W} = \{\text{path } a, \text{path } b\}$ equipped with $\mu_1(\mathbf{w}) = \frac{1}{2}$ for all $\mathbf{w} \in \mathbf{W}$ - the uniform initial distribution. Each agent $A_s \in A$ then randomly samples a solution, $\mathbf{w}_s$ identically and independently of one another generating a sample, $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_S\}$. For each solution in our sample, the cost objective function maps to a real number which correlates more desirable solutions to lower costs, and vice versa. Our set of sample costs

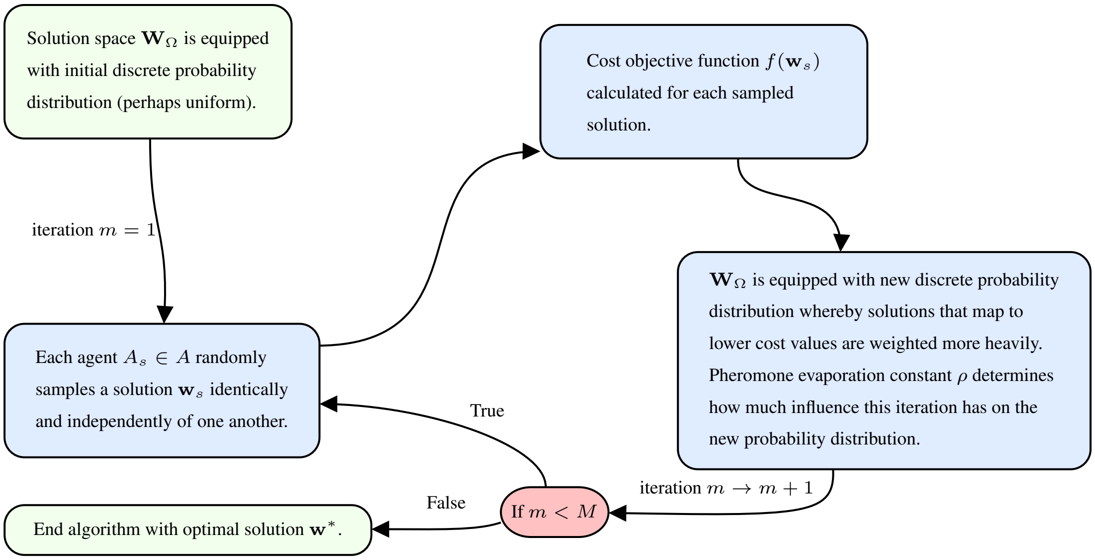

Solution space $\mathbf{W}_\Omega$ is equipped with initial discrete probability distribution (perhaps uniform).
iteration $m = 1$
Each agent $A_s \in A$ randomly samples a solution $\mathbf{w}_s$ identically and independently of one another.
Cost objective function $f(\mathbf{w}_s)$ calculated for each sampled solution.
$\mathbf{W}_\Omega$ is equipped with new discrete probability distribution whereby solutions that map to lower cost values are weighted more heavily. Pheromone evaporation constant ρ determines how much influence this iteration has on the new probability distribution.
iteration $m \to m + 1$
If $m < M$
True
False
End algorithm with optimal solution $\mathbf{w}^*$.

Figure 3: Flow chart outlining the structure of a generalised ACO Algorithm

is $\{f(\mathbf{w}_1), f(\mathbf{w}_2), \ldots, f(\mathbf{w}_S)\}$. In the double bridge experiment, we could define a cost objective function as such

$$f : \mathbf{W} \to \mathbb{R} \text{ such that } f(\text{path } a) = l_a \text{ and } f(\text{path } b) = l_b$$

This would then intuitively agree with both the asymmetric case, $l_a > l_b$, since the cost of traversing path b is lower and therefore more desirable, and the symmetric case, $l_a = l_b$, and there is no cost advantage of either route. Say we are in iteration m, the next step is to generate a new discrete probability distribution by taking $\mu_m(\cdot)$ and increasing the weighting of solutions that appear in our sample / are 'close' to elements in our sample with inverse proportion to their cost. This means that solutions (and close neighbours) that appear multiple times in the sample or have a lower cost should be given higher weighting mimicking its biological analogue whereby pheromone concentration is higher on shorter (less costly), freshly / heavily travelled routes. We note here that the idea of two solutions being 'close' in the solution space relies on the notion of a metric which we will go into more detail later once we have specified a particular solution space in a given application of this algorithm. Normalisation of the distribution will cause solutions that do not appear in the sample to have a decreased weighting in our new distribution. The amount we let the sample from iteration m influence the distribution $\mu_{m+1}(\cdot)$ is limited by pheromone evaporation constant ρ. It becomes clear why ρ is named as such when we see that pheromone evaporation happening at a faster rate (larger ρ) would cause ants to sense most recent / fresher trails proportionally more than older trails, thus choosing fresher trails with higher probability. An informal sketch of a pheromone update rule might be,

$$\mu_{m+1} \propto (1 - \rho) * \mu_m + \rho * \text{Sample}.$$

Now that iteration m is complete we increase the iteration number to $m+1$ and continue this loop providing we are not at the predetermined stopping iteration, M, for the algorithm.

If $m = M$, the algorithm halts and we take the mode of distribution $\mu_M(\cdot)$, i.e. $\mathbf{w}^* \in \mathbf{W}_\Omega$ such that for all $\mathbf{w} \in \mathbf{W}_\Omega \setminus \{\mathbf{w}^*\}$, $\mu_M(\mathbf{w}^*) > \mu_M(\mathbf{w})$, to be solution to the combinatorial optimisation problem. However, it is often not clear before the algorithm has started what a suitable value of M may be and instead it can be useful to allow to algorithm to run indefinitely under some stopping condition. Such a stopping condition might be of the form; for a given $\epsilon > 0$, stop the algorithm when $\exists \mathbf{w}^* \in \mathbf{W}_\Omega$ and $M \in \mathbb{N}$ such that $\mu_M(\mathbf{w}^*) > 1-\epsilon$, i.e. stop when there is a iteration where distribution on solution space has probability density concentrated at one solution. Again, we choose the mode of the distribution $\mu_M(\cdot)$ - clearly coinciding with $\mathbf{w}^*$ - to be solution to the combinatorial optimisation problem. The almost sure finiteness of such a stopping time should be assessed for each specific ant colony algorithm implementation to avoid running into infinite loops.

1.4 Overview of Project

Having outlined the basic intuition and motivation of ACO, we will now give a brief overview of the direction in which this paper takes our understanding of the concept.

The travelling salesman problem (TSP) is perhaps the most natural application of ACO, so using this as a tool to better understand and investigate such algorithms will provide great insight into the potential of ACO.

At this point, the ability of ACO to give exact results to the TSP is unknown. Very little has been rigorously proved in this area however one insightful paper allows, in theory, for a particular ACO to give exact solutions to the travelling salesman problem with high probability under fine tuning of algorithm parameters. We will give a detailed proof for the simplified case of the complete graph K_n as a series of Lemmas followed by the central theorem.

Following this we will discuss the practical limitations of the theoretical results derived in this section and talk about how we might be able to gain further insight through an experimental investigation where simulations on the algorithm will be run for a particular type of random graph. Analysing the results of simulations will reveal how valid / accurate solutions produced by ACO are when compared with exact solutions to the TSP.

Finally due to the in-feasibility of verifying the exactness of ACO solutions for large number of nodes n, we will build predictive models to discuss how we expect solutions for the TSP to behave for large n and compare this to the ACO proposed solutions.

2 Application to the Travelling Salesman Problem

The travelling salesman problem (TSP) is a classical problem in combinatorics formulated in the 19th century by W.R. Hamilton and goes as such;

> *Given a list of n cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once returning to the initial city at the end?*

This problem is formalised to the well-known graph theory problem of finding the shortest Hamiltonian cycle (SHC).

Definition 2.0.1 (Hamiltonian Cycle)**.** For a directed graph $G = (V, E)$ with vertices V such that $|V| = n$ and edges $E \subseteq V \times V$, a Hamiltonian cycle is defined as an ordered list of vertices $\mathbf{w} = (w_1, w_2, \ldots, w_n, w_{n+1})$ such that every vertex of V appears exactly once in the list with the exception of $w_1 = w_{n+1}$ appearing both at the start and end such that all

edges edges $(w_1, w_2), (w_2, w_3), \ldots, (w_{n-1}, w_n), (w_n, w_{n+1}) \in E$.
If in addition each directed edge $(i, j) \in E$ has weight e_{ij} then the total weight of the cycle is $\sum_{i=1}^{n} e_{w_i, w_{i+1}}$.

Remark. The number of unique Hamiltonian cycles on K_n up to repetitions due to shifts and reflections is given by,

$$\frac{n!}{2n} = \frac{(n-1)!}{2} \approx \sqrt{\frac{\pi(n-1)}{2}} \left(\frac{n-1}{e}\right)^{n-1}.$$

To see this, we note there are $n!$ permutations of vertices $1, \ldots, n$. The condition of returning to origin does not change the number of permutations as this is not a degree of freedom. Then there is a shift symmetry of these permutations corresponding to the n possible starting nodes (e.g. $(1, 2, 3, 1) \sim (3, 1, 2, 3)$ where '$\sim$' meaning the permutations are equivalent) leading us to divide by n. Finally we have to divide by 2 to take account for the reflections of permutations e.g. $(1, 2, 3, 1) \sim (1, 3, 2, 1)$. This gives the above quantity. The approximation follows directly from Stirling's approximation for factorials.
The consequence of this result is that as n increases, the number of unique Hamiltonian cycles in K_n grows asymptotically faster than any polynomial and even any exponential in n, making an exhaustive search of the solution space infeasible for even moderately large n. We note here that solving such a problem exactly does not require exhaustive search of the solution space, as we will see later. However, even the most efficient algorithms for exact solves of the TSP are still of complexity at least exponential in n [4], validating a heuristic approach to such a problem.

2.1 The Travelling Salesman Problem

We will be focusing on solving the TSP on the complete graph K_n, so every vertex is accessible from every other vertex. This may have limitations in real world applications as this is often not the case, so we note that the condition of the graph being complete is not necessary for results shown in this paper but rather is imposed for the purpose of simplicity.
We can write the travelling salesman problem as a combinatorial optimisation problem $(\mathbf{W}, \Omega, f)$ where:

- $K_n = (V, E)$ is the complete undirected graph on n vertices where each city corresponds to a vertex and the edge weight $e_{i,j} = d(i, j)$ the distance between city i and city j. For a nice visual output in the simulations we take euclidean distance but if applied to a real world scenario we could set edge weights according to a more practical metric such as the average time taken to drive from city i to city j.

- $\mathbf{W}$ is the set of all finite sequences on the set $\{1, \ldots, n\}$. We can interpret a sequence $\mathbf{w} = (w_1, w_2, \ldots, w_k) \in \mathbf{W}$ as a list of vertices to visit in order, thus defining a path on K_n.

- Ω is the set of constraints on $\mathbf{W}$ meaning feasible solution space $\mathbf{W}_\Omega$ is set of length-$n+1$ sequences $\mathbf{w} = (w_1, w_2, \ldots, w_n, w_{n+1}) \in \mathbf{W}$ such that $w_1 = w_{n+1} = 1$ and $w_i \neq w_j$ for $1 \leq i, j \leq n$, $i \neq j$, thus ensuring each vertex is visited exactly once with exception of start and end node which are the same.

- Cost objective function $f : \mathbf{W}_\Omega \to \mathbb{R}^+$ is defined as $f(\mathbf{w}) = \sum_{i=1}^{n} e_{w_i, w_{i+1}}$ the sum of edge weights along the cycle defined by $\mathbf{w}$, therefore giving the total distance traversed. We aim to find $\mathbf{w}^* = \text{argmin}_{\mathbf{w} \in \mathbf{W}_\Omega} f(\mathbf{w})$ as the solution of this combinatorial optimisation problem.

The ant colony algorithm we will use to tackle this problem is an interpretation of Gutjahr's Ant System [3]. We use this specifically as it has rigorous theoretical results proven relating to the probability of convergence to optimal solution. It mostly follows the same logic as the generalised ACO algorithm defined in the previous section, however, there are some additional conditions which allow for the theoretical results to hold. The rest of this section will focus on understanding the specific mechanisms of this algorithm in relation to the TSP.

2.2 Gutjahr's Ant System

Let $A = \{A_1, \dots, A_S\}$ be the set of S agents. In a given iteration $m \leq M$ each agent independently and identically performs a random walk on the graph K_n starting from a random uniformly chosen vertex and visiting each vertex exactly once returning to starting point at the end thus ensuring the walk constructed by agent s in iteration m denoted $\mathbf{w}^s = (w_1^s, \dots, w_n^s, w_{n+1}^s)$ is an element of feasible solution space $\mathbf{W}_\Omega$. We define the partial random walk constructed by agent s up to the t^{th} step of iteration m as $\mathbf{u}^s(t) = (u_1^s, \dots, u_t^s)$ where $1 \leq t \leq n+1$ (i.e. where the agent has walked already in that iteration). We have $\mathbf{u}^s(n+1) = \mathbf{w}^s$ the full random walk.
Let,

$$f^*(1) = \infty,$$

and,

$$f^*(m+1) = \min\{\min_{A_s \in A} f(\mathbf{w}^s), f^*(m)\},$$

be the lowest value of cost objective function achieved by any agent in iterations $1, \dots, m$. We note that f^* is therefore monotone decreasing in m.
Finally we define the pheromone concentration associated with edge (i,j) in iteration m to be $\tau_{i,j}(m)$. All together the pheromone concentration values form an $n \times n$ matrix $\boldsymbol{\tau}(m)$ with i, j^{th} entry equal to $\tau_{i,j}(m)$.
The transition probabilities for agent A_s's random walk before the $t+1^{\text{th}}$ step of iteration m must depend on three inputs; the partial random walk up until step t, $\mathbf{u}^s(t)$, the pheromone matrix $\boldsymbol{\tau}(m)$, and predetermined edge weights $\{e_{i,j}\}_{i,j \in \{1,\dots,n\}}$.

$$\begin{aligned} p_{i,j}(m, \mathbf{u}^s(t)) &:= \mathbb{P}(u_{t+1}^s = j | u_t^s = i, \mathbf{u}^s(t), \text{ iteration \# is } m) \\ &= \begin{cases} \frac{(\tau_{i,j}(m))^\alpha (e_{i,j})^{-\beta}}{\sum_{k \in \{1,\dots,n\} \setminus \mathbf{u}^s(t)} (\tau_{i,k}(m))^\alpha (e_{i,k})^{-\beta}} & \text{if } j \in \{1, \dots, n\} \setminus \mathbf{u}^s(t) \\ 1 & t = n \text{ and } j = u_1^s \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Where $\alpha, \beta \geq 0$ are weighting parameters controlling the influence of pheromone values and edge weights respectively on the transition probability. For example, setting $\alpha = 0, \beta = 1$ would result in a purely greedy heuristic whereby transition probabilities are determined only by edge weights and pheromone values are redundant, defeating the point of an ACO algorithm. Of course, the first case of this probability is only relevant for $1 \leq t \leq n-1$ since for $t = n$ every vertex has been visited by the partial walk and the set $\{1, \dots, n\} \setminus \mathbf{u}^s(t) = \emptyset$, so the walk then returns to initial point with probability 1. For an unvistied vertex j, the transition probability of an agent going from vertex i to j is therefore proportional, up to exponent $\alpha > 0$, to the pheromone value on edge (i, j) and inversely proportional, up to exponent $\beta > 0$, to the weight of edge (i, j). Moreover, we impose the normalising condition on edge weights so that $\sum_{(i,j)\in E} e_{i,j}^{-1} = 1$, again this is not essential for convergence but keeps things neat.

Instead of viewing each agent as performing a random walk, we can instead think of each agent sampling a solution from the feasible solutions space $\mathbf{W}_\Omega$ according to the discrete probability measure $\mu_m(\cdot)$ which can be calculated directly by multiplying along transition probabilities as such,

$$\mu_m(\mathbf{w}) = \prod_{i=1}^{n} p_{w_i,w_{i+1}}(m, \mathbf{u}(i)).$$

The pheromone values $\tau_{i,j}(m+1)$ are dynamically updated using information from iteration m. For simplicity we assume $\sum_{i,j=1}^{n} \tau_{i,j}(m) = 1$ as a normalising condition. This is not necessary for the algorithm's efficacy but prevents pheromone values from blowing up exponentially and is thus easier to analyse. Since we will define the pheromone values as a function of the previous iterations outcomes, we need to set an initial pheromone value to each vertex, so we do so uniformly. Since there are $n(n-1)$ directed edges, for $i, j \in \{1, \ldots, n\}$ we let,

$$\tau_{i,j}(1) = \begin{cases} \frac{1}{n(n-1)} & i \neq j \\ 0 & i = j \end{cases}.$$

After each agent has constructed a walk $\mathbf{w}^s$ in iteration m, we calculate the cost objective function $f(\mathbf{w}^s)$ and define a new variable corresponding to the change in pheromone value on edge (i, j) by agent s as,

$$\Delta\tau_{i,j}^{s} = \begin{cases} \frac{1}{f(\mathbf{w}^s)} & (i,j) \in \mathbf{w}^s \text{ and } f(\mathbf{w}^s) \leq f^*(m) \\ 0 & \text{otherwise} \end{cases}.$$

This rule is what is referred to as an elitist strategy and is essential for proving results about convergence. In essence, it means that only solutions mapping to a cost at least as low as any solution constructed in any previous iteration will have an impact on the new pheromone values in the subsequent iteration. The elitist strategy implemented here does not have a biological analogue as it would require the ants to have knowledge about the future cost of a trial before walking it and deciding whether or not they lay trail pheromones which of course cannot happen. We take reciprocal of the cost of a given walk to relate lower cost walks to greater influence in the change in pheromone values.

Let,

$$\mathcal{C}_m := \sum_{i,j=1}^{n} \sum_{s=1}^{S} \Delta\tau_{i,j}^{s},$$

be the total change in pheromone value on any edge by any agent in iteration m and,

$$\Delta\tau_{i,j} = \frac{1}{\mathcal{C}_m} \sum_{s=1}^{S} \Delta\tau_{i,j}^{s},$$

be the total normalised change in pheromone value on edge (i,j) by any agent in iteration m. As well as a normalising factor, $\mathcal{C}_m$ somewhat acts as an indicator as to whether any agent has constructed a solution of at least as low cost as to one seen in any iteration before.

$$\begin{aligned}\mathcal{C}_m = 0 &\iff \Delta\tau^s_{i,j} = 0 \ \forall i,j \in \{1,\ldots,n\}, s \in \{1,\ldots,S\}\\ &\iff f(\mathbf{w}^s) > f^*(m) \ \forall A_s \in A\end{aligned}$$

The pheromone update rule between iteration m and $m+1$ is then defined by,

$$\tau_{i,j}(m+1) = \begin{cases}(1-\rho)\tau_{i,j}(m) + \rho\Delta\tau_{i,j} & \mathcal{C}_m > 0\\ \tau_{i,j}(m) & \mathcal{C}_m = 0\end{cases}.$$

We notice that the change in pheromone value on edge (i,j) is normalised by constant $\mathcal{C}_m$ which then allows for the convex combination of $\tau_{i,j}(m)$ and $\Delta\tau_{i,j}$ to preserve this normalising condition over $(i,j) \in E$ and pass it onto the next iterations' pheromone values. This further illustrates the role of evaporation constant; for small ρ most of the pheromone value in iteration $m+1$ is just carried over from what it was before with a small amount being updated due to the change calculated whereas for large ρ, we place far more weight upon the most recently calculated pheromone changes. We will see more on this later but intuitively this has the affect that a large ρ may cause higher strength paths to be carved out in fewer iterations at the expense of certainty about their optimality.

As pheromone concentration on a particular edge increases, the weighting of all solutions that share that particular edge increases in the discrete probability distribution μ_m. This confirms our intuition that solutions that are 'close' to one another, meaning they share one or more edges, will benefit collectively in their chance of selection under μ_m, if one is sampled, and satisfied the elitist condition, in previous iterations.

The algorithm is complete once a certain stopping iteration, M, is completed. This variable can be deterministic and set prior to the execution of the algorithm or can be a random variable determined under some condition based upon the state of variables within the algorithm, more on this later. When the algorithm has finished, the lowest cost solutions achieved in all cycles is proposed as the ACO algorithm's suggested optimal solution.

3 Convergence Results of the ACO algorithm

3.1 Outline of theoretical results and proof

In this section, we describe the proof in [3] that under certain assumptions the ACO algorithm converges to the optimal solution with probability arbitrarily close to 1. The proof in [3] proves the result for general graphs, but we have simplified the exposition by restricting it to the complete graoh K_n. We will first outline a sketch detailing how we will go about proving this and then break it down into a series of lemmas and corollaries before approaching the main theorem.

Sketch of Proof. We begin by showing that this ACO algorithm is a Markov process indexed by the iteration number in Lemma 3.1, meaning the state of the ACO algorithm in iteration m is totally determined probabilistically based upon its state in iteration $m-1$. Next, in Lemma 3.3 we will find a lower bound on the probability that the optimal solution is sampled by some agent in some given iteration, in terms of the ant colony population, S, the pheromone evaporation constant, ρ and the number of vertices in the graph, n. Moreover we will show this bound can be made arbitrarily close to 1 under manipulation of these parameters.

Following this, we will condition on the algorithm on having already discovered the optimal solution in some iteration and show in Lemma 3.4 that the pheromone values will then become concentrated exclusively and evenly along the edges of that optimal solution with high probability. It is then natural to show that the transition probabilities for agents performing random walks on the graph will become be arbitrarily close to 1 in Lemma 3.5, subsequently resulting in the overall probability of an agent sampling the optimal solution to be arbitrarily close to 1 in Corollary 3.5.1.

The last step required conditioning on the optimal path being discovered by at least one agent in some iteration, so we then relax this strict condition in conjunction with the use of Baye's theorem to show that the probability of never sampling the optimal solution can be made arbitrarily small in Lemma 3.7.

At this point we have gathered all the necessary components to prove the central theorem. Theorem 3.8 states that we can make the probability of sampling the optimal solution for every iteration after some point, arbitrarily close to 1 in two different ways; prior to executing the algorithm we can either make the ant colony population, S, sufficiently large or the pheromone evaporation constant, ρ, sufficiently small.

3.2 Theoretical convergence to optimal solution of the ACO algorithm

Assumptions for Convergence. We must assume that for our optimisation problem, the optimal solution, denoted $\mathbf{w}^* = (w_1^*, \dots, w_n^*, w_{n+1}^*)$, is unique i.e. $f(\mathbf{w}^*) < f(\mathbf{w})$ for all $\mathbf{w} \in \mathbf{W}_\Omega \setminus \{\mathbf{w}^*\}$. On K_n with symmetric directed edges, the optimal solution will have multiple representations due to the symmetries of reflections and translations. We therefore group all symmetrically equivalent solutions into one to satisfy this property of uniqueness. We further assume for all $(i,j) \in E$, $0 < e_{i,j} < \infty$ i.e. no two vertices can be at the same point nor have infinite distance between them.

For simplicity we let $\alpha = 1$ which does not restrict the algorithm since we can adjust β accordingly for desired balance of weighting between edge weight values and pheromone values for the agents' random walk transition probabilities.

Lemma 3.1 (Interpretation of ACO algorithm as a Markov Process)**.**
The triple $(\boldsymbol{\tau}(m), \mathbf{w}(m), f^*(m))$ is a time discrete Markov process

Proof. In order to show this is a Markov Process we must show that the distribution of the random triple $(\boldsymbol{\tau}(m), \mathbf{w}(m), f^*(m))$ only depends on $(\boldsymbol{\tau}(m-1), \mathbf{w}(m-1), f^*(m-1))$.
We see that for each $(i,j) \in E$,

$$\tau_{i,j}(m) = \begin{cases} (1-\rho)\tau_{i,j}(m-1) + \rho\Delta\tau_{i,j} & \mathcal{C}_{m-1} > 0 \\ \tau_{i,j}(m-1) & \mathcal{C}_{m-1} = 0 \end{cases}$$

is a deterministic function of $\tau_{i,j}(m-1)$, $\Delta\tau_{i,j}$ and $\mathcal{C}_{m-1}$ where both of the latter are functions of $\mathbf{w}(m-1)$ and $f^*(m-1)$ and the former is just an entry in the matrix $\boldsymbol{\tau}(m-1)$. Hence $\boldsymbol{\tau}(m)$ only depends on $(\boldsymbol{\tau}(m-1), \mathbf{w}(m-1), f^*(m-1))$.

Given any walk $\mathbf{w} = (w_1, \dots, w_n) \in \mathbf{W}_\Omega$ with $\mathbf{u}(t) = (w_1, \dots, w_t)$ the partial walk up to the t^{th} step, we can write the probability of any agent independently constructing it in the m^{th} iteration as,

$$\begin{aligned}\mathbb{P}(\mathbf{w}) &= \prod_{i=1}^{n} p_{w_i,w_{i+1}}(m, \mathbf{u}(i)) \\ &= \begin{cases} \prod_{i=1}^{n-1} \frac{(\tau_{w_i,w_{i+1}}(m))^{\alpha}(e_{w_i,w_{i+1}})^{-\beta}}{\sum_{k\in\{1,\dots,n\}\setminus\mathbf{u}(i)}(\tau_{w_i,k}(m))^{\alpha}(e_{w_i,k})^{-\beta}} & \text{if } \mathbf{w}\in\mathbf{W}_{\Omega} \\ 0 & \text{otherwise} \end{cases}.\end{aligned}$$

Thus since each $e_{i,j}$ are nonrandom predetermined constants and each $\tau_{i,j}(m)$ depends only on $(\boldsymbol{\tau}(m-1), \mathbf{w}(m-1), f^*(m-1))$ we have that the probability distribution used to sample random walks in iteration m depends only on information from iteration $m-1$.
Finally $f^*(m) = \min\{\min_{A_s\in A} f(\mathbf{w}^s), f^*(m-1)\}$ where we first minimise the cost over walks $\mathbf{w}^s \in \mathbf{w}(m-1)$ constructed by agents in the previous iteration and then choose the minimum of this value and $f^*(m-1)$ so clearly it only depends on $(\boldsymbol{\tau}(m-1), \mathbf{w}(m-1), f^*(m-1))$. Thus the triple $(\boldsymbol{\tau}(m-1), \mathbf{w}(m-1), f^*(m-1))$ is a discrete time Markov process indexed by iteration number m. □

Definition 3.1.1. Now we will define some probability events that will be used throughout the rest of this section, let

- $E_m^{(s)} = \{\mathbf{w}^s(m) = \mathbf{w}^*\}$ be the event that agent A_s traverses the optimal path in iteration m.
- $B_m = \bigcup_{s=1}^{S} E_m^{(s)}$ be the event that at least one of the S agents traverses the optimal path in iteration m.
- $F_m = (\bigcap_{i=1}^{m-1} B_i^c) \bigcap B_m$ be the event that no agent traverses the optimal path in the first $m-1$ iterations but at least one agent does (for the first time) in iteration m.
- $H = \bigcup_{m=1}^{M} F_m$ be the event that the optimal walk is traversed by any agent in any iteration during the execution of the algorithm. We notice each F_m are mutually exclusive so the union is disjoint.

We further define the constant $\gamma := \min_{(i,j)\in E}\{e_{i,j}^{-\beta}\}$, which will be achieved by the edge of highest weight, and $\Gamma := \max_{(i,j)\in E}\{e_{i,j}^{-\beta}\}$, which will be achieved by the edge of smallest weight.

Lemma 3.2. *For sequence of probability events $A_1, \dots, A_n$ not necessarily independent where there is a uniform bound on probability of each event that is not affected by conditioning on other any combination of the other events i.e. if there exists $C > 0$ such that,*

$$\mathbb{P}(A_i), \mathbb{P}(A_i | A_{a_1}, \dots, A_{a_m}) \leq C,$$

for all $i \in \{1, \dots, n\}$ and any $A_{a_1}, \dots, A_{a_m}$ not including A_i, we have that ,

$$\mathbb{P}(\cap_{k=1}^{n} A_k) \leq C^n.$$

Proof.

$$\begin{aligned}\mathbb{P}(\cap_{k=1}^n A_k) &= \mathbb{P}(A_1 | \cap_{k=2}^n A_k)\mathbb{P}(\cap_{k=2}^n A_k)\\ &= \mathbb{P}(A_1 | \cap_{k=2}^n A_k)\mathbb{P}(A_2 | \cap_{k=3}^n A_k)\mathbb{P}(\cap_{k=3}^n A_k)\\ &\vdots\\ &= (\prod_{i=1}^{n-1} \mathbb{P}(A_i | \cap_{k=i+1}^n A_k))\mathbb{P}(A_n)\end{aligned}$$

But then since the bound is uniform for each event A_i, $i \in \{1, \dots, n\}$ and independent of conditioning on other events, each term in the product, and the final term $\mathbb{P}(A_n)$ can be bound by C. Hence the result follows. □

The above lemma will be used extensively throughout this section as it allows us to bound the probability of intersections of events maintaining a complicated interdependence which may have no analytically closed form.

Lemma 3.3 (Lower bound on probability of optimal path travelled in iteration m)**.**
Let $c := (1-\rho)^{n-1}$ *and* $p := (\frac{\gamma}{n(n-1)})^{n-1}$ *then,*

$$\mathbb{P}(B_m) \geq 1 - (1 - (\frac{(1-\rho)^{m-1}\gamma}{n(n-1)})^{n-1})^S.$$

That is that we can bound the probability that at least one agent traverses the optimal path in iteration m below by some non-zero constant.

Proof. Since $\Delta\tau_{i,j} \geq 0$ for all $(i,j) \in E$ and $\rho \in (0,1)$ we see that,

$$\begin{aligned}\tau_{i,j}(m+1) &= \begin{cases} (1-\rho)\tau_{i,j}(m) + \rho\Delta\tau_{i,j} \geq (1-\rho)\tau_{i,j}(m) & \mathcal{C}_m > 0 \\ \tau_{i,j}(m) \geq (1-\rho)\tau_{i,j}(m) & \mathcal{C}_m = 0 \end{cases}.\\ &\geq (1-\rho)\tau_{i,j}(m)\end{aligned}$$

Therefore we can reapply this inequality to get,

$$\tau_{i,j}(m+1) \geq (1-\rho)^m\tau_{i,j}(1) = \frac{(1-\rho)^m}{n(n-1)}. \tag{1}$$

Now due to the normalising condition on $e_{i,j}^{-1}$ we have that for all $(i,j) \in E$, $e_{i,j}^{-\beta} \leq 1$ thus for any partial walk $\mathbf{u}(t)$,

$$\sum_{k\in\{1,\dots,n\}\setminus\mathbf{u}(t)} \tau_{i,k}(m)(e_{i,k})^{-\beta} \leq \sum_{k\in\{1,\dots,n\}\setminus\mathbf{u}(t)} \tau_{i,k}(m) \leq \sum_{k\in\{1,\dots,n\}} \tau_{i,k}(m) \leq 1.$$

Therefore for $j \in \{1, \dots, n\} \setminus \mathbf{u}^s(t)$, we can bound the transition probabilities $p_{i,j}(m, \mathbf{u}(t))$ below as such,

$$\begin{aligned}p_{i,j}(m, \mathbf{u}(t)) &= \frac{\tau_{i,j}(m)(e_{i,j})^{-\beta}}{\sum_{k\in\{1,\dots,n\}\setminus\mathbf{u}^s(t)} \tau_{i,k}(m)(e_{i,k})^{-\beta}}\\ &\geq \tau_{i,j}(m)(e_{i,j})^{-\beta}.\end{aligned} \tag{2}$$

Now we want to use this to put a lower bound on $\mathbb{P}(E_m^{(s)})$ the probability agent A_s traverses the optimal path in iteration m. We find an expression for this by multiplying the transition probabilities along the path $\mathbf{w}^* = (w_1^*, \dots, w_n^*)$ to get,

$$\begin{aligned}
\mathbb{P}(E_m^{(s)}) &= \prod_{i=1}^{n} p_{w_i^*, w_{i+1}^*}(m, (w_1^*, \dots, w_i^*)) \\
&\geq \prod_{i=1}^{n-1} \tau_{w_i^*, w_{i+1}^*}(m)(e_{w_i^*, w_{i+1}^*}^{-\beta}) \times 1 \\
&\geq \gamma^{n-1} \prod_{i=1}^{n-1} \tau_{w_i^*, w_{i+1}^*}(m) \\
&\geq \gamma^{n-1} \prod_{i=1}^{n-1} (1-\rho)^{m-1} \qquad \leftarrow \left(\frac{1-\rho}{n(n-1)}\right)^{m-1} \\
&= (\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1}.
\end{aligned}$$

Finally we can use the independence of each agent's walk in the same iteration to find a bound on the probability no agent traverses the optimal path in iteration m,

$$\begin{aligned}
\mathbb{P}(B_m^c) &= \mathbb{P}(\bigcap_{s=1}^{S} (E_m^{(s)})^c) \\
&= (1 - \mathbb{P}(E_m^{(s)}))^S \\
&\leq (1 - (\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1})^S \\
\implies \mathbb{P}(B_m) &\geq 1 - (1 - (\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1})^S.
\end{aligned}$$

□

One useful result from this Lemma is noting that the bounds derived in the above Lemma are dependent on only the iteration number m and parameters fixed before the start of the algorithm. Hence information about the events $B_1, \dots, B_{m-1}$ will not affect the bound. Thus the following corollary also holds.

Corollary 3.3.1.

$$\mathbb{P}(B_m | \bigcap_{i=1}^{m-1} B_i^c) \geq 1 - (1 - (\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1})^S.$$

Given that no agent has traversed the optimal walk in the first $m-1$ iterations, the probability of at least one agent traversing it in the m^{th} iteration, is also bounded below by the same quantity.

As with the biological analogue, it is intuitive for the algorithm to have settled on a solution when the pheromone trail is concentrated almost exclusively and distributed evenly along the path corresponding to the optimal solution. Since pheromone values are normalised in this algorithm, the above intuition translates to $\tau_{i,j} \approx \frac{1}{n-1}$ - since there are $n-1$ edges in a path - for $(i,j) \in \mathbf{w}^*$ and $\tau_{i,j} \approx 0$ otherwise. The following lemma formalises this idea.

Lemma 3.4 (W.H.P exclusive and even spread of pheromones along optimal path)**.** $\forall \epsilon > 0, \forall m \in \mathbb{N}, \exists d_{\epsilon,m} \in \mathbb{N}$ *such that* $\forall \hat{m} \geq m + d_{\epsilon,m}$*,*

$$\textbf{\textit{(i)}}\ \forall (i,j) \in w^* \text{ we have } \mathbb{P}(\{|\tau_{i,j}(\hat{m}) - \frac{1}{n}| < \epsilon\}|F_m) \geq 1 - \epsilon,$$
$$\textbf{\textit{(ii)}}\ \forall (i,j) \notin w^* \text{ we have } \mathbb{P}(\{\tau_{i,j}(\hat{m}) < n\epsilon\}|F_m) \geq 1 - \epsilon.$$

That is given that the optimal path is traversed by at least one agent for the first time in iteration m, there is some number of iterations, $d_{\epsilon,m}$, after m whereby all subsequent iterations will have the pheromone values concentrated on the optimal walk with high probability.

Proof. Throughout this proof we assume the condition F_m, i.e. at least one agent traverses the optimal path $\mathbf{w}^*$ for the first time in iteration m.
For some $\hat{m} > m$ we call upon $\mathcal{C}_{\hat{m}}$ to indicate whether an at-least-as-low cost solution has been constructed in iteration $\hat{m}$ than any seen before. Since the lowest cost solution after iteration m is simply the optimal solution $\mathbf{w}^*$ (by assumption of F_m), therefore $\mathcal{C}_{\hat{m}}$ indicates whether any agent has traversed the optimal path in iteration $\hat{m}$. If $\mathcal{C}_{\hat{m}} = 0$ then $\tau_{i,j}(\hat{m}+1) = \tau_{i,j}(\hat{m})$ for all $(i,j) \in E$ and no agent has traversed the optimal path in iteration $\hat{m}$, so let us first focus on the case $\mathcal{C}_{\hat{m}} > 0$ implying there is some non-empty subset of the agents, say $\{A_1, \ldots, A_r\} \subseteq A$ that have traversed the optimal path in iteration $\hat{m}$. Thus the change in pheromone concentration on edge (i,j) by agent s is given by,

$$\Delta\tau^s_{i,j} = \begin{cases} \frac{1}{f(\mathbf{w}^*)} & (i,j) \in \mathbf{w}^* \text{ and } 1 \leq s \leq r \\ 0 & \text{otherwise} \end{cases}.$$

Hence for $(i,j) \in \mathbf{w}^*$,

$$\begin{aligned} \Delta\tau_{i,j} &= \frac{1}{\mathcal{C}_{\hat{m}}} \sum_{s=1}^{S} \Delta\tau^s_{i,j} \\ &= \frac{\sum_{s=1}^{S} \Delta\tau^s_{i,j}}{\sum_{k,l=1}^{n} \sum_{s=1}^{S} \Delta\tau^s_{k,l}} \\ &= \frac{\sum_{s=1}^{r} \frac{1}{f(\mathbf{w}^*)} + \sum_{s=r+1}^{S} 0}{\sum_{(k,l)\in\mathbf{w}^*} (\sum_{s=1}^{r} \frac{1}{f(\mathbf{w}^*)} + \sum_{s=r+1}^{S} 0)} \\ &= \frac{r\frac{1}{f(\mathbf{w}^*)}}{nr\frac{1}{f(\mathbf{w}^*)}} \\ &= \frac{1}{n}, \end{aligned}$$

whilst clearly for $(i,j) \notin \mathbf{w}^*$ we have $\Delta\tau_{i,j} = 0$. Now we need to see how this affects the pheromone update rule between iteration $\hat{m}$ and $\hat{m}+1$, so for $(i,j) \in \mathbf{w}^*$,

$$\begin{aligned}\tau_{i,j}(\hat{m}+1) &= (1-\rho)\tau_{i,j}(\hat{m}) + \rho\Delta\tau_{i,j} \\ &= (1-\rho)\tau_{i,j}(\hat{m}) + \frac{\rho}{n}\end{aligned}$$

,

which can be rearranged in two different ways for two useful results,

$$\tau_{i,j}(\hat{m}+1) - \frac{1}{n} = (1-\rho)(\tau_{i,j}(\hat{m}) - \frac{1}{n}) \Rightarrow \text{ if } \tau_{i,j}(\hat{m}) \geq \frac{1}{n} \text{ then } \tau_{i,j}(\hat{m}+1) \geq \frac{1}{n}, \tag{3}$$

$$\tau_{i,j}(\hat{m}+1) - \tau_{i,j}(\hat{m}) = \rho(\frac{1}{n} - \tau_{i,j}(\hat{m})) \Rightarrow \text{ if } \tau_{i,j}(\hat{m}) < \frac{1}{n} \text{ then } \tau_{i,j}(\hat{m}+1) > \tau_{i,j}(\hat{m}). \tag{4}$$

Combining these we get that,

$$\tau_{i,j}(\hat{m}+1) \geq \min\{\tau_{i,j}(\hat{m}), \frac{1}{n}\}.$$

The above result was shown for the case $\mathcal{C}_{\hat{m}} > 0$ but this still holds for $\mathcal{C}_{\hat{m}} = 0$ as in this case $\tau_{i,j}(\hat{m}+1) = \tau_{i,j}(\hat{m})$ for all $(i,j) \in E$ and the above inequality is satisfied. This essentially means pheromone values along the edges of the optimal path are monotone increasing (not strict) from one iteration to the next or will stay above $\frac{1}{n}$. We can apply this inequality repeatedly to get for all $\hat{m} > m$,

$$\begin{aligned}\tau_{i,j}(\hat{m}+1) &\geq \min\{\tau_{i,j}(\hat{m}), \frac{1}{n}\} \\ &\geq \min\{\frac{(1-\rho)^{m-1}}{n(n-1)}, \frac{1}{n}\} \\ &\geq \frac{(1-\rho)^{m-1}}{n(n-1)}.\end{aligned}$$

where the second line is given by inequality (1).
We now recall $\gamma := \min_{(i,j)\in E}\{e_{i,j}^{-\beta}\}$ and $p_{i,j}(m, \mathbf{u}(t)) \geq \tau_{i,j}(m)e_{i,j}^{-\beta}$ from inequality (2). We can bound the following transition probability,

$$\begin{aligned}p_{i,j}(\hat{m}, \mathbf{u}(t)) &\geq \tau_{i,j}(\hat{m})\gamma \\ &\geq \frac{\gamma(1-\rho)^{m-1}}{n(n-1)}.\end{aligned}$$

Thus the probability a given agent A_s traverses the optimal path in iteration $\hat{m} > m$ given that the optimal path was traversed for the first time in iteration m is given by,

$$\begin{aligned}\mathbb{P}(E_{\hat{m}}^{(s)}|F_m) &= \prod_{i=1}^{n} p_{w_i,w_{i+1}}(\hat{m}, (w_1, \dots, w_i)) \\ &\geq \prod_{i=1}^{n-1} \frac{\gamma(1-\rho)^{m-1}}{n(n-1)} \times 1 \\ &= (\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1}.\end{aligned}$$

Crucially, this bound is independent of $\hat{m}$ and any event occurring in iterations $m+1,\dots,\hat{m}$. To bound above the probability no agent traverses the optimal path in iteration $\hat{m} > m$ given that the optimal path was traversed for the first time in iteration m, we look at the intersection of the complement of event $E^{(s)}_{\hat{m}}$,

$$\begin{aligned}\mathbb{P}(B^c_{\hat{m}}|F_m) &= \mathbb{P}(\bigcap_{s=1}^{S}(E^{(s)}_{\hat{m}})^c|F_m)\\ &= (1-\mathbb{P}(E^{(1)}_{\hat{m}}|F_m))^S\\ &\leq (1-(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1})^S.\end{aligned}$$

Where the second line is due to the fact each agent draws a solution identically and independently from the same distribution. Now for some $d \in \mathbb{N}$, since we have the uniform bound,

$$\mathbb{P}(B^c_{\hat{m}}|F_m) \leq \left(1-\left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S \in (0,1),$$

with the bound independent of any information from iterations $i \in \{m+1,\dots,m+d\}\setminus\{\hat{m}\}$, we can apply Lemma 3.2, and since,

$$\mathbb{P}(B_{\hat{m}}|F_m) \leq 1,$$

we can consider the probability of $B_{\hat{m}}|F_m$ occurring on a given k iterations from iteration $m+1,\dots,m+d$ and $B^c_{\hat{m}}|F_m$ occurring on the remaining $d-k$ iterations from iteration $m+1,\dots,m+d$ and bound it above as such.

$$\begin{aligned}&\mathbb{P}(\{B_{\hat{m}} \text{ occurs on } k \text{ iterations}\}\bigcap\{B^c_{\hat{m}} \text{ occurs on } d-k \text{ iterations}\}|F_m)\\ \leq& 1^k\left(\left(1-\left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S\right)^{d-k}.\end{aligned}$$

Thus since there are a possible $\binom{d}{k}$ ways of choosing on which of the d iterations $B_{\hat{m}}|F_m$ occurs, we can write the cumulative probability as follows,

$$\begin{aligned}&\mathbb{P}(B_{\hat{m}} \text{ occurs for at most } k-1 \text{ of iterations } m+1,\dots,m+d|F_m)\\ \leq&\sum_{i=0}^{k-1}\binom{d}{i}1^i\left(\left(1-\left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S\right)^{d-i}\\ \leq& d^k\left(\left(1-\left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S\right)^{-k}\left(\left(1-\left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S\right)^{d}\\ =&o(d^kc^d)\xrightarrow{d\to\infty}0 \text{ for constant } c\in(0,1).\end{aligned}$$

Hence we can make the probability that at least one agent traverses the optimal path on at least k of the iterations $m+1,\dots,m+d$ arbitrarily close to 1 by making d sufficiently large i.e. for all $\delta > 0$ there exists $d \in \mathbb{N}$ such that,

$$\mathbb{P}(B_{\hat{m}} \text{ occurs for at least } k \text{ of iterations } m+1,\dots,m+d|F_m) \geq 1-\delta.$$

Thus to find an upper bound on the probability that $B_i|F_m$ occurs for any k consecutive iterations from iteration $\hat{m} > m$ to $\hat{m}+k$ we have,

$$\mathbb{P}(\bigcap_{i=\hat{m}}^{\hat{m}+k} B_i^c|F_m) \leq (1-(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1})^{Sk}.$$

Which due to the fact $0 < 1-(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)})^{n-1} < 1$ we can make the above bound arbitrarily small for large k. Even though for each $i \in \{\hat{m},\dots,\hat{m}+k\}$ the events B_i are not mutually independent, we have found a uniform bound that is indeed independent of conditioning on any other events B_j for $j \in \{\hat{m},\dots,\hat{m}+k\} \setminus \{i\}$ due to Corollary 3.3.1. This therefore satisfies conditions for Lemma 3.2 and the inequality holds.
Thus for any $\delta > 0$, we can choose $k_\delta \in \mathbb{N}$ such that,

$$\begin{aligned}
&\mathbb{P}(\bigcap_{i=m+1}^{m+k_\delta} B_i^c|F_m) \leq \delta \\
\Rightarrow &\mathbb{P}(\bigcup_{i=m+1}^{m+k_\delta} B_i|F_m) \geq 1-\delta \\
\Rightarrow &\mathbb{P}(\bigcap_{h'=1}^{h} \bigcup_{i=m+1+(h'-1)k_\delta}^{m+k_\delta h'} B_i|F_m) \geq (1-\delta)^h \geq 1-2\delta h \text{ for } \delta \in (0,\frac{1}{2}).
\end{aligned}$$

Where we have divided the sequence of iterations into $h \in \mathbb{N}$ independent periods of length k_δ as shown in Figure 3.2, whereby the probability that at least one agent traverses the optimal path in a period of k_δ iterations is bounded below by $1-\delta$.

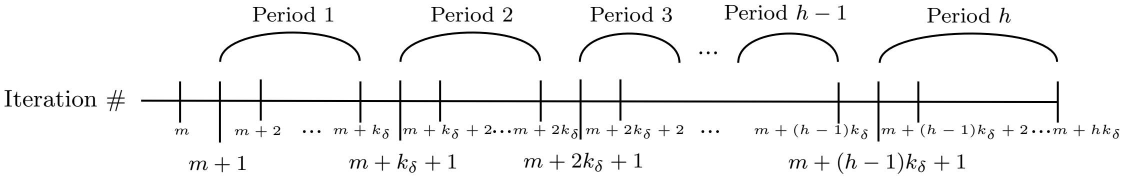


Figure 4: Time scale showing how iterations are divided into periods on which we can determine probability bounds.

Fix $\epsilon > 0$, we can choose $\delta = \frac{\epsilon}{2h}$ and the above probability is bounded below by $1-\epsilon$. We can choose the number of these period, h, such that $(1-\rho)^h < \epsilon$. Let $d_{\epsilon,m} = hk_\delta$. By equation 3 in an iteration, $\hat{m}$ where the optimal path is traversed, for an edge $(i,j) \in \mathbf{w}^*$,

$$\tau_{i,j}(\hat{m}+1) - \frac{1}{n} = (1-\rho)(\tau_{i,j}\left(\hat{m}) - \frac{1}{n}\right).$$

Thus the distance between $\tau_{i,j}(\hat{m}+1)$ and $\frac{1}{n}$ decreases by a factor of $1-\rho$ if an agent traverses the optimal walk in iteration $\hat{m}$ and remains the same if not. As a result, since in the $d_{\epsilon,m}$ iterations after m, we will get at least h successful iterations with probability greater than $1-\epsilon$, hence with at least the same probability again,

$$\begin{aligned}|\tau_{i,j}(m+d_{\epsilon,m}) - \frac{1}{n}| &\leq (1-\rho)^h |\tau_{i,j}(m) - \frac{1}{n}| \\ &\leq (1-\rho)^h \\ &\leq \epsilon,\end{aligned}$$

so,

$$\mathbb{P}(|\tau_{i,j}(m+d_{\epsilon,m}) - \frac{1}{n}| \leq \epsilon | F_m) \geq 1-\epsilon,$$

which proves part **(i)** of the lemma.
The second part follows as a direct result of the normalising condition that for any $\hat{m}$,

$$\sum_{(i,j)\in E} \tau_{i,j}(\hat{m}) = 1.$$

We have just shown that with probability greater than $1-\epsilon$ and conditional on F_m, the pheromone value on each edge of $\mathbf{w}^*$ can be made larger than $\frac{1}{n}-\epsilon$. Therefore the sum of the pheromone values along the n edges of $\mathbf{w}^*$ can be made larger than,

$$n(\frac{1}{n}-\epsilon) = (1-n\epsilon).$$

Thus any edge $(i,j) \notin \mathbf{w}^*$ may have at most pheromone value $\tau_{i,j}(\hat{m}) \leq n\epsilon$, by normalising condition, which shows part **(ii)**. □

Lemma 3.5. *Given optimal walk* $\mathbf{w}^* = (w_1^*, \dots, w_{n+1}^*)$ *and its partial walk upto the* i^{th} *step* $\mathbf{u}^*(t) = (w_1^*, \dots, w_i^*)$, *for any* $\epsilon > 0$ *and each* $m \in \mathbb{N}$ *there exists* $d'_{\epsilon,m} \in \mathbb{N}$ *such that for all* $\hat{m} \geq m + d'_{\epsilon,m}$ *we have,*

$$\mathbb{P}(p_{w_i^*,w_{i+1}^*}(\hat{m}, \mathbf{u}^*(i)) \geq 1-\epsilon | F_m) \geq 1-\epsilon.$$

This lemma tells us given that the optimal solution was sampled in iteration m for the first time, we can carry out another $d'_{\epsilon,m}$ iterations to make transition probabilities along edges contained in the optimal solution within an ϵ distance of, or arbitrarily close to, 1 for all subsequent iterations with high probability.

Proof. First we assume without loss of generality that $\alpha = 1$. This assumption does not constrict the result as we can just adjust β to offset this but allows for nicer algebra manipulation. Let $\epsilon > 0$ and consider the transition probability along edges of the optimal solution $\mathbf{w}^*$ in iteration $\hat{m}$,

$$\begin{aligned}p_{w_i^*,w_{i+1}^*}(\hat{m}, \mathbf{u}^*(i)) &= \frac{(\tau_{w_i^*,w_{i+1}^*}(\hat{m}))^\alpha (e_{w_i^*,w_{i+1}^*})^{-\beta}}{\sum_{k\in\{1,\dots,n\}\setminus \mathbf{u}^*(i)} (\tau_{w_i^*,k}(\hat{m}))^\alpha (e_{w_i^*,k})^{-\beta}} \\ &= \frac{\tau_{w_i^*,w_{i+1}^*}(\hat{m}) (e_{w_i^*,w_{i+1}^*})^{-\beta}}{\sum_{k\in\{1,\dots,n\}\setminus \{\mathbf{u}^*(i), w_{i+1}^*\}} \tau_{w_i^*,k}(\hat{m}) (e_{w_i^*,k})^{-\beta} + \tau_{w_i^*,w_{i+1}^*}(\hat{m}) (e_{w_i^*,w_{i+1}^*})^{-\beta}}.\end{aligned}$$

Where in the final line we have split the sum in the denominator into feasible indices not including w^*_{i+1}, the next step of the optimal solution, and the next step of optimal solution on its own.

Now from here we condition the following statements on F_m occurring. Using the normalising condition on edge weights, we have $(e_{i,j})^{-\beta} \leq 1$ for all $(i,j) \in E$ then combining this with Lemma 3.4 part **(ii)** we have with probability greater than $1 - \hat{\epsilon}$,

$$\begin{aligned}\sum_{k \in \{1,\dots,n\} \setminus \{\mathbf{u}^*(i), w^*_{i+1}\}} \tau_{w^*_i,k}(\hat{m})(e_{w^*_i,k})^{-\beta} &\leq \sum_{k \in \{1,\dots,n\} \setminus \{\mathbf{u}^*(i), w^*_{i+1}\}} \tau_{w^*_i,k}(\hat{m}) \\ &\leq (n-1) \cdot n\hat{\epsilon},\end{aligned}$$

where $n-1$ is the degree of any vertex and the $n\hat{\epsilon}$ comes from the bound in Lemma 3.4 part **(ii)**.

Now from Lemma 3.4 part **(i)** we have with probability greater than $1 - \hat{\epsilon}$,

$$\begin{aligned}&|\tau_{w^*_i,w^*_{i+1}}(\hat{m}) - \frac{1}{n}| \leq \hat{\epsilon} \\ \implies &\frac{1}{n} - \hat{\epsilon} \leq \tau_{w^*_i,w^*_{i+1}}(\hat{m}) \leq \frac{1}{n} + \hat{\epsilon}.\end{aligned}$$

Combining this with the previous result yields,

$$\begin{aligned}p_{w^*_i,w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) &\geq \frac{(\frac{1}{n} - \hat{\epsilon})(e_{w^*_i,w^*_{i+1}})^{-\beta}}{(n-1)n\hat{\epsilon} + (\frac{1}{n} + \hat{\epsilon})(e_{w^*_i,w^*_{i+1}})^{-\beta}} \\ &= \frac{1 - n\hat{\epsilon}}{1 + \hat{\epsilon}(n^2(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + n)}.\end{aligned}$$

Now for any $x \geq 0$ we have the following inequality,

$$(1+x)^{-1} \geq 1 - x,$$

so setting $x = \hat{\epsilon}(n^2(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + n)$ we get ,

$$\begin{aligned}p_{w^*_i,w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) &\geq (1 - n\hat{\epsilon})(1 - \hat{\epsilon}(n^2(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + n)) \\ &= 1 - \hat{\epsilon}(n^2(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + 2n) + \hat{\epsilon}^2 n^2(n(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + 1) \\ &\geq 1 - \hat{\epsilon}(n^2(n-1)(e_{w^*_i,w^*_{i+1}})^{\beta} + 2n) \\ &\geq 1 - \hat{\epsilon}(\frac{n^2(n-1)}{\gamma} + 2n),\end{aligned}$$

where we recall $\gamma := \min_{(i,j) \in E}\{e_{i,j}^{-\beta}\}$.

At this point for any $\epsilon > 0$, we can simply choose $\hat{\epsilon}$ - with corresponding $d_{\hat{\epsilon},m}$ - as in Lemma 3.4 such that $\hat{\epsilon} = \frac{\epsilon}{(\frac{n^2(n-1)}{\gamma} + 2n)}$ then set $d'_{\epsilon,m} = d_{\hat{\epsilon},m}$. Now for all $\hat{m} \geq m + d'_{\epsilon,m}$ we have,

$$\mathbb{P}(p_{w^*_i,w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) \geq 1 - \epsilon | F_m) \geq 1 - \hat{\epsilon} \geq 1 - \epsilon.$$

□

Definition 3.5.1. Let,

$$P^*_{\hat{m}} := \prod_{i=1}^{n} p_{w^*_i, w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)),$$

be the random variable giving the probability a given agent samples the optimal solution in iteration $\hat{m}$.

Corollary 3.5.1. *For all $\epsilon > 0$ and each $m \in \mathbb{N}$ there exists $d''_{\epsilon,m} \in \mathbb{N}$ such that for all $\hat{m} \geq m + d''_{\epsilon,m}$ we have,*

$$\mathbb{P}(P^*_{\hat{m}} \geq 1 - \epsilon | F_m) \geq 1 - \epsilon.$$

We have already shown in the previous Lemma that we can make transition probabilities along the optimal path arbitrarily close to 1 with high probability. A quick result from this can be derived by multiplying transition probabilities to get the probability of an agent sampling the entire optimal walk and then finding the associated high probability bound on this.

Proof. Let $\epsilon > 0$. Let $\hat{\epsilon} > 0$ and $d'_{\hat{\epsilon},m}$ be those given by Lemma 3.5 and set $\hat{\epsilon} = \frac{\epsilon}{2n}$ and $d''_{\epsilon,m} = d'_{\hat{\epsilon},m}$. For $\hat{m} \geq m + d''_{\epsilon,m}$ we notice the probability events satisfy the following set inclusions,

$$\begin{aligned} \bigcap_{i=1}^{n} \{p_{w^*_i, w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) \geq 1 - \frac{\epsilon}{2n} | F_m\} &\subseteq \{P^*_{\hat{m}} \geq (1 - \frac{\epsilon}{2n})^n | F_m\} \\ &\subseteq \{P^*_{\hat{m}} \geq 1 - 2n\frac{\epsilon}{2n} | F_m\} \\ &= \{P^*_{\hat{m}} \geq 1 - \epsilon | F_m\}, \end{aligned}$$

where the penultimate line follows from the inequality that for small $\hat{\epsilon} > 0$ we have $(1-\hat{\epsilon})^n \geq 1 - 2n\hat{\epsilon}$. By monotonicity of probability measure we therefore have,

$$\begin{aligned} \mathbb{P}(P^*_{\hat{m}} \geq 1 - \epsilon | F_m) &\geq \mathbb{P}(\bigcap_{i=1}^{n} \{p_{w^*_i, w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) \geq 1 - \frac{\epsilon}{2n}\} | F_m) \\ &\geq (1 - \frac{\epsilon}{2n})^n \\ &\geq 1 - \epsilon, \end{aligned}$$

for all $\hat{m} \geq m + d''_{\epsilon,m}$ where the penultimate line follows from Lemma 3.2 and the fact that for each i, $\mathbb{P}(p_{w^*_i, w^*_{i+1}}(\hat{m}, \mathbf{u}^*(i)) \geq 1 - \frac{\epsilon}{2n} | F_m)$ has uniform lower bound of $1 - \frac{\epsilon}{2n}$. □

Lemma 3.6. *For all $\epsilon > 0$ there exists $d'''_{\epsilon,m}$ such that for a fixed agent A_s and for all $\hat{m} \geq m + d'''_{\epsilon,m}$ we have,*

$$\mathbb{P}(E^{(s)}_{\hat{m}} | F_m) \geq 1 - \epsilon.$$

Equivalently for any $\epsilon > 0$ and $m \in N$ we can find some number of iterations, $d'''_{\epsilon,m}$, such that for any subsequent iterations from $m + d'''_{\epsilon,m}$ we can make the probability that some fixed agent, A_s, traverses the optimal walk given that the optimal walk was traversed for the first time in iteration m arbitrarily close to 1.

Proof. We first notice that $\mathbb{P}(E_{\hat{m}}^{(s)}) = P_{\hat{m}}^*$ as defined in Definition 3.5.1 as in a given iteration, each agent samples a walk IID. Let $\hat{\epsilon} > 0$ and $d''_{\hat{\epsilon},m}$ be as in Corollary 3.5.1. Then for all $\hat{m} \geq m + d''_{\hat{\epsilon},m}$,

$$\begin{aligned}\mathbb{P}(E_{\hat{m}}^{(s)}|F_m) &\geq \mathbb{P}(E_{\hat{m}}^{(s)} \cap \{P_{\hat{m}}^* \geq 1-\hat{\epsilon}\}|F_m) \\ &= \mathbb{P}(E_{\hat{m}}^{(s)}|\{P_{\hat{m}}^* \geq 1-\hat{\epsilon}\} \cap F_m)\mathbb{P}(P_{\hat{m}}^* \geq 1-\hat{\epsilon}|F_m),\end{aligned}$$

which follows from Bayes' theorem on conditional probability. Now it is possible to bound both these probabilities below by $1-\hat{\epsilon}$ due to the condition of $\{P_{\hat{m}}^* \geq 1-\hat{\epsilon}\}$ for the first probability and as a direct result of Corollary 3.5.1 for the second probability. Hence for any $\epsilon > 0$ we can simply choose $\hat{\epsilon} = \frac{\epsilon}{2}$ and $d'''_{\epsilon,m} = d''_{\hat{\epsilon},m}$ then for all $\hat{m} \geq m + d'''_{\epsilon,m}$,

$$\begin{aligned}\mathbb{P}(E_{\hat{m}}^{(s)}|F_m) &\geq (1-\hat{\epsilon})(1-\hat{\epsilon}) \\ &= 1 - 2\hat{\epsilon} + \hat{\epsilon}^2 \\ &\geq 1 - 2\hat{\epsilon} = 1 - \epsilon.\end{aligned}$$

□

Recall from Definition 3.1.1 that B_m is the event that at least one of the S agents traverses the optimal path in iteration m.

Lemma 3.7. *For any $\epsilon > 0$, we either make colony size, S, sufficiently large or pheromone evaporation constant, ρ, sufficiently small such that if let run for an indefinite number of iterations, the algorithm will induce a probability for some agent to traverse the optimal walk in some iteration arbitrarily close to* 1 *i.e.*

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) \geq 1 - \epsilon.$$

Proof. Taking compliments of Corollary 3.3.1 tells us that the probability of no agents traversing the optimal path in a given iteration, m, given that it had not been traversed in any previous iteration as well, is bounded as such,

$$\mathbb{P}\left(B_m^c \middle| \bigcap_{i=1}^{m-1} B_i^c\right) \leq \left(1 - \left(\frac{\gamma(1-\rho)^{m-1}}{n(n-1)}\right)^{n-1}\right)^S.$$

Hence if we consider the probability that no agent samples the optimal solution in the first m cycles we get,

$$\begin{aligned}\mathbb{P}\left(\bigcap_{i=1}^{m} B_i^c\right) &= \mathbb{P}(B_1^c)\mathbb{P}(B_2^c|B_1^c)\dots\mathbb{P}\left(B_m^c \middle| \bigcap_{i=1}^{m-1} B_i^c\right) \\ &\leq \prod_{i=1}^{m}\left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1}\right)^S.\end{aligned}$$

Thus to bound the probability that at least one agent traverses the optimal path in at least one iteration indexed by $\mathbb{N}$, we consider,

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) &= 1 - \mathbb{P}\left(\bigcap_{i=1}^{\infty} B_i^c\right) \\
&= 1 - \lim_{m\to\infty} \mathbb{P}\left(\bigcap_{i=1}^{m} B_i^c\right) \\
&\geq 1 - \lim_{m\to\infty} \prod_{i=1}^{m} \left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1}\right)^{S} \\
&= 1 - \prod_{i=1}^{\infty} \left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1}\right)^{S}.
\end{aligned}$$

Thus taking logarithms of the infinite product and using the inequality $\log(x) \leq x - 1$ we have,

$$\begin{aligned}
\log\left(\prod_{i=1}^{\infty} \left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1}\right)^{S}\right) &= \sum_{i=1}^{\infty} S \log\left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1}\right) \\
&\leq \sum_{i=1}^{\infty} S \left(1 - \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1} - 1\right) \\
&= -S \sum_{i=1}^{\infty} \left(\frac{\gamma(1-\rho)^{i-1}}{n(n-1)}\right)^{n-1} \\
&= -S \left(\frac{\gamma}{n(n-1)}\right)^{n-1} \sum_{i=0}^{\infty} \left((1-\rho)^{n-1}\right)^{i} \\
&= -S \left(\frac{\gamma}{n(n-1)}\right)^{n-1} \frac{1}{1-(1-\rho)^{n-1}}.
\end{aligned}$$

By use of the formula for infinite geometric sum noting that since $\rho \in (0,1)$, $|(1-\rho)^{n-1}| < 1$ for all $n \in \mathbb{N}$. Substituting this into the previous working yields,

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) \geq 1 - \exp\left(-S \left(\frac{\gamma}{n(n-1)}\right)^{n-1} \frac{1}{1-(1-\rho)^{n-1}}\right).$$

Thus for any fixed $\epsilon > 0$ and ρ we can choose,

$$S \geq -\log(\epsilon)(1-(1-\rho)^{n-1})\left(\frac{n(n-1)}{\gamma}\right)^{n-1},$$

or equivalently for any fixed $\epsilon > 0$ and S we can choose,

$$\rho \leq 1 - \left(1 + \frac{S}{\log(\epsilon)}\left(\frac{\gamma}{n(n-1)}\right)^{n-1}\right)^{\frac{1}{n-1}},$$

such that,

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) \geq 1-\epsilon.$$

□

At this point, we have gathered all the components to prove the central theorem.

Theorem 3.8. *Let $P_{\hat{m}}^*$ be as in Definition 3.5.1, then the following two assertions hold* ***(i)*** *For all $\epsilon > 0$ and any fixed ρ, β we can find S sufficiently large such that there exists $m_\epsilon \in \mathbb{N}$ such that for all iterations $\hat{m} \geq m_\epsilon$ we have,*

$$P_{\hat{m}}^* \geq 1-\epsilon.$$

(ii) *For all $\epsilon > 0$ and any fixed S, β we can find ρ sufficiently small such that there exists $m_\epsilon \in \mathbb{N}$ such that for all iterations $\hat{m} \geq m_\epsilon$ we have,*

$$P_{\hat{m}}^* \geq 1-\epsilon.$$

This essentially says we can bound the probability that a given agent samples the optimal path in some iteration, m_ϵ, arbitrarily close to 1 and for this probability bound to hold in every subsequent iteration as well. Furthermore, we can achieve this result in two possible ways; **(i)** states this can be done by making the number of agents in the colony, S, sufficiently large so that in each iteration a large number of solutions in $\mathbf{W}_\Omega$ are sampled thus leading to a greater exploration of solution space. **(ii)** states we can achieve the result by making the pheromone evaporation constant, ρ, sufficiently small (close to zero) meaning that particular pheromone trails (along edges of solutions) will take longer to strengthen, leaving time for more iterations to occur, thus more exploration of the solution space, before converging upon a solution. We note that both of these approaches will increase the computational running time of the algorithm.

Proof. Let $\epsilon > 0$. Recall from Definition 3.1.1, F_m is the event that no agent traverses the optimal path in the first $m-1$ iterations but at least one agent does (for the first time) in iteration m. Notice the events F_m are mutually disjoint and their union over $\mathbb{N}$ can be written as,

$$\bigcup_{i=1}^{\infty} B_i = \bigcup_{i=1}^{\infty} F_i.$$

Hence,

$$\begin{aligned}\mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) &= \mathbb{P}\left(\bigcup_{i=1}^{\infty} F_i\right)\\ &= \sum_{i=1}^{\infty} \mathbb{P}(F_i).\end{aligned}$$

We can find $k \in \mathbb{N}$ such that the tail of this convergent sum is bounded above by $\frac{\epsilon}{4}$,

$$\sum_{i=k+1}^{\infty} \mathbb{P}(F_i) \leq \frac{\epsilon}{4}.$$

This results in the following bound on the probability of any agent traversing the optimal walk for the first time in any of the first k iterations,

$$\begin{aligned}\mathbb{P}\left(\bigcup_{i=1}^{k} F_i =\right) &= \mathbb{P}\left(\bigcup_{i=1}^{\infty} F_i \setminus \bigcup_{i=k+1}^{\infty} F_i\right) \\ &= \mathbb{P}\left(\bigcup_{i=1}^{\infty} F_i\right) - \mathbb{P}\left(\bigcup_{i=k+1}^{\infty} F_i\right) \\ &= \mathbb{P}\left(\bigcup_{i=1}^{\infty} B_i\right) - \sum_{i=k+1}^{\infty} \mathbb{P}(F_i).\end{aligned}$$

So we can choose S large enough or ρ close enough to 0 as in Lemma 3.7 such that $\mathbb{P}(\cup_{i=1}^{\infty} B_i) \geq 1 - \frac{\epsilon}{4}$ to get,

$$\mathbb{P}\left(\bigcup_{i=1}^{k} F_i =\right) \geq 1 - \frac{\epsilon}{4} - \frac{\epsilon}{4} = 1 - \frac{\epsilon}{2}.$$

Now recall from Lemma 3.6 for any m and $\epsilon > 0$ we can find $d'''_{\epsilon,m} \in \mathbb{N}$ such that for a fixed agent A_s and for all $\hat{m} \geq m + d'''_{\epsilon,m}$ we have $\mathbb{P}(E^{(s)}_{\hat{m}} | F_m) \geq 1 - \epsilon$. Thus choose $\frac{\epsilon}{2}$ and take,

$$d_k = \max\{d'''_{\frac{\epsilon}{2},m} : 1 \leq m \leq k\}.$$

Then for any fixed $m \leq k$ and any $\hat{m} \geq m + d_k$ we have that, $\mathbb{P}(E^{(s)}_{\hat{m}} | F_m) \geq 1 - \frac{\epsilon}{2}$. This leads us to the final step in the proof of this theorem which begins by relating these to the random variable, $P^*_{\hat{m}}$. Let $\hat{m} \geq m + d_k$,

$$\begin{aligned}P^*_{\hat{m}} &= \mathbb{P}(E^{(s)}_{\hat{m}}) \\ &= \sum_{m=1}^{k} \mathbb{P}(E^{(s)}_{\hat{m}} | F_m)\mathbb{P}(F_m) + \mathbb{P}\left(E^{(s)}_{\hat{m}} \middle| \bigcap_{m=1}^{k} F^c_m\right) \mathbb{P}\left(\bigcap_{m=1}^{k} F^c_m\right) \\ &\geq \sum_{m=1}^{k} \mathbb{P}(E^{(s)}_{\hat{m}} | F_m)\mathbb{P}(F_m) \\ &\geq (1 - \frac{\epsilon}{2}) \sum_{m=1}^{k} \mathbb{P}(F_m) \\ &\geq (1 - \frac{\epsilon}{2})(1 - \frac{\epsilon}{2}) \\ &= 1 - \epsilon + \frac{\epsilon^2}{4} \\ &\geq 1 - \epsilon.\end{aligned}$$

□

3.3 Limitations on theoretical convergence results of the ACO algorithm

The convergence result shown in this section is remarkable and outlines a key characteristic that not all heuristic algorithms share. This algorithm is said to be optimisation-capable meaning there is no principle obstacle in obtaining the true optimal solution. However, the bounds derived on S and ρ in Lemma 3.7 that implied the convergence result are not at all useful as a guide for setting the parameters. To put this into perspective, for a graph with just $n = 10$ nodes, a minimum inversely exponentiated edge weight of $\gamma = 0.001$ and pheromone evaporation constant $\rho = 0.05$, if the probability of convergence to the optimal solution is desired to be greater than 0.9 then we would require the ant colony population S, to be at least,

$$
\begin{aligned}
S &\geq -\log(\epsilon)(1-(1-\rho)^{n-1})\left(\frac{n(n-1)}{\gamma}\right)^{n-1} \\
&= -\log(0.1)(1-(1-0.05)^{10-1})\left(\frac{10\cdot(10-1)}{0.001}\right)^{10-1} \\
&= 8.9 \times 10^{44}.
\end{aligned}
$$

Completing even just one iteration would require vast computation power with this many agents. The absurdity of such a task becomes even more apparent when we realise the number of unique Hamiltonian cycles on the complete graph K_{10} is equal to $\frac{(10-1)!}{2} = 181440$ - many orders of magnitude smaller than the bound on S. This entirely defeats the point of ACO as a means to solve combinatorial optimisation problems, when even an exhaustive search would obtain a solution faster with the addition of not having to compromise on exactness.

This motivates the next sections investigation into the efficacy of this particular ACO algorithm for more realistic / pragmatic parameter values.

4 Experimental Investigation

4.1 Experimental Model

In order to derive experimental results on this algorithm through data collection from simulations, it is necessary to define a model for the graphs upon which this algorithm will be run.

Definition 4.0.1 (Complete Random Graph on the Euclidean unit disc)**.** Let

$$D := \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x}| \leq 1\}$$

be the unit 2D euclidean disc.

X
Not
uniform

Let $\theta_1, \dots, \theta_n \sim \text{Unif}[0, 2\pi)$ and $r_1^2, \dots, r_n^2 \sim \text{Unif}[0, 1]$ be uniform random variables. We take the points $\mathbf{x}_1, \dots, \mathbf{x}_n \in D$ such that $\mathbf{x}_i = (r_i \cos(\theta_i), r_i \sin(\theta_i))$. This construction gives a method of sampling n uniformly distributed random points on D.

Now we define K_n^D to be the complete random graph on n nodes on the Euclidean unit disc with directed, weighted edges given by $e_{i,j} = |\mathbf{x}_i - \mathbf{x}_j|$, the euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

From this point forwards we will be considering the ACO algorithm defined in 2.2 applied to the TSP on K_n^D.

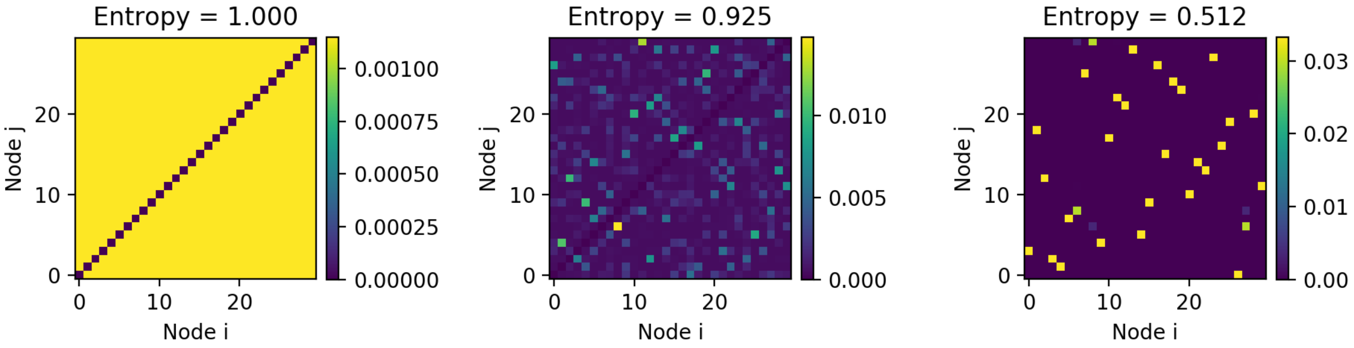
4.2 Entropy of Pheromone Matrix

A useful way to measure the degree to which this ACO algorithm has settled into equilibrium with one particular solution it has proposed to be optimal is via the use of entropy. Shannon first invented the idea of probabilistic entropy [5] as a way to quantify the certainty of a probability distribution, uniform being the least certain and non-random degenerate being the most certain. Formally, for a discrete random variable X equipped with probability measure, μ, he defined the entropy as $\mathbb{E}(-\log(\mu(X))) = \sum_x -\mu(x)\log(\mu(x))$. Here we extend this idea to find the entropy of pheromone concentration matrices which analogously measures how certain the algorithm is one particular solution (corresponding to a set of n edges).

Definition 4.0.2 (Shannon matrix Entropy)**.** For an $n \times n$ non-negative, off-diagonal matrix, $\boldsymbol{\tau}$, with elements (τ_{ij}) such that $\tau_{ij} \geq 0$ for all $i, j \in \{1, \ldots, n\}^2$, $\tau_{ii} = 0$ for all $i \in \{1, \ldots, n\}$ and $\sum_{i,j=1}^{n} \tau_{ij} = 1$, the Shannon matrix entropy H is defined as follows

$$H(\boldsymbol{\tau}) = -\sum_{1 \leq i,j \leq n \,:\, i \neq j} \tau_{ij} \frac{\log(\tau_{ij})}{\log(n(n-1))}$$

Where the log refers to the natural logarithm and we use the convention $x\log(x)|_{x=0} = 0$.



Images generated using matplotlib.pyplot python library

Figure 5: Pheromone concentration matrix, $\boldsymbol{\tau}$, for K_{30}^{D} at three different stages of the algorithm.

Note that due to the normalisation of the pheromone concentration matrix, the scale for the colour map in each plot of figure 5 is different. We interpret the color value at pixel (i, j) as the pheromone concentration along edge (i, j). These three plots were constructed from snapshots of the pheromone matrix prior to execution of the algorithm at iteration 1, during execution of the algorithm at iteration 6 and after 'completion' of the algorithm at iteration 98 respectively from left to right. Contrary to the second law of thermodynamics, the ACO algorithm is designed to decrease entropy as time moves forwards.

Remark. Our algorithm begins with a uniform spread of pheromone values on each of the $n(n-1)$ edges, and thus the pheromone matrix has all off diagonal elements equal to $\frac{1}{n(n-1)}$

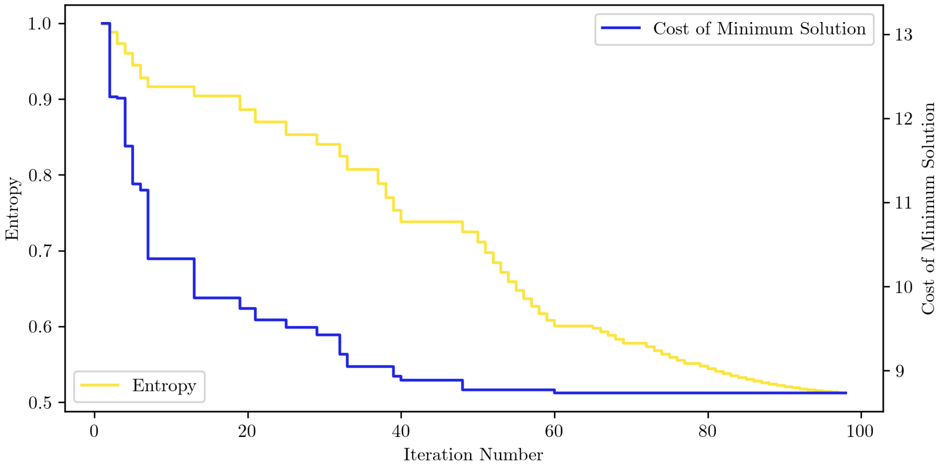


Figure 6: Plot showing how the entropy of the pheromone matrix and the cost of the minimum solution of the same ACO simulation as in Figure 5 develop with iteration number.

whilst all diagonal elements fixed permanently at 0, as there is no edge from a vertex to itself. This initial setup corresponds to an entropy given by,

$$\begin{aligned} H(\boldsymbol{\tau}) &= -\sum_{1\leq i,j\leq n\,:\,i\neq j} \tau_{ij}\frac{\log(\tau_{ij})}{\log(n(n-1))} \\ &= -\sum_{1\leq i,j\leq n\,:\,i\neq j} \frac{1}{n(n-1)}\frac{\log(\frac{1}{n(n-1)})}{\log(n(n-1))} \\ &= \frac{n(n-1)}{n(n-1)}\frac{\log(n(n-1))}{\log(n(n-1))} \\ &= 1, \end{aligned}$$

which, as it turn out, maximises the entropy function. An example of this can be seen in the first plot of figure 5.

As per Lemma 3.4, we find our algorithm has selected a particular path, say $\mathbf{w}^*$, and is therefore in equilibrium when the pheromone value is approximately $\frac{1}{n}$ for each of the n edges of $\mathbf{w}^*$ and approximately 0 everywhere else. This pheromone matrix would corresponds to an equilibrium entropy given by,

$$\begin{aligned}
H(\boldsymbol{\tau}) &= -\sum_{1\leq i,j\leq n\,:\,i\neq j} \tau_{ij}\frac{\log(\tau_{ij})}{\log(n(n-1))} \\
&= -\sum_{(i,j)\in \mathbf{w}^*} \tau_{ij}\frac{\log(\tau_{ij})}{\log(n(n-1))} - \sum_{(i,j)\notin \mathbf{w}^*} \tau_{ij}\frac{\log(\tau_{ij})}{\log(n(n-1))} \\
&\approx -\sum_{(i,j)\in \mathbf{w}^*} \frac{1}{n}\frac{\log(\frac{1}{n})}{\log(n(n-1))} - \sum_{(i,j)\notin \mathbf{w}^*} 0\frac{\log(0)}{\log(n(n-1))} \\
&= n\times\frac{1}{n}\frac{\log(n)}{\log(n(n-1))} + n(n-2)\times 0 \\
&= \frac{\log(n)}{\log(n(n-1))} \approx \frac{1}{2} \text{ for large } n.
\end{aligned}$$

In figure 5, plot 3, visually we can see $\boldsymbol{\tau}$ is very close to equilibrium with one bright spot per each row and column corresponding to high pheromone concentration along 30 edges, and negligible pheromone concentration everywhere else, with an approximate entropy of 0.512. Since there are 30 nodes, a theoretical equilibrium entropy would be given by

$$H(\boldsymbol{\tau}) = \frac{\log(30)}{\log(30\cdot 29)} \approx 0.503$$

We can therefore use entropy as a measure of how 'settled' the algorithm is, where higher values closer to 1 correspond to it being totally unsettled and smaller values closer to $\frac{1}{2}$ correspond to it being settled on a solution. We note this is not the minimum of H as we see if $\tau_{i_0 j_0} = 1$ for some i_0, j_0 and 0 for all other i, j, then $H(\boldsymbol{\tau}) = 0$. Further we note that just because the the algorithm has settled on a particular solution, it does not imply this solution is certain to be the global optimal. This is the main limitation of a heuristic algorithm; although we can make probabilistically 'good' guesses at the optimal solution, they are still just guesses and in order to say for certain whether the solution is globally optimal, it would require a more computationally intensive search.

4.3 Stopping time of ACO algorithm

The ACO algorithm outlined in [3] does not give an explicit protocol for stopping, so in this section we explore stopping times and justify the use of a specific stopping time devised by the author. Formally, a stopping time is the number of iterations executed before halting the algorithm. The algorithm cannot simply be instructed to execute until the optimal solution is found as such a stopping time requires knowledge of the optimal solution, which if were already known, would defeat the point of such an algorithm. One choice of stopping time might be to set a predetermined fixed number of iterations prior to execution, however since this does not depend on any information gained during execution of the algorithm and due to the many possible variations on parameters of the algorithm, to use a 'one-stopping-time-fits-all' approach is likely to result in redundant extra iterations or an insufficient number of iterations to get a confident solution. This dilemma motivates the following definition.

Definition 4.0.3 (Entropy driven stopping time)**.** For an ACO algorithm solving the TSP on n nodes with pheromone matrix in iteration m given by $\boldsymbol{\tau}(m)$ and $\delta > 0$, define the Entropy driven stopping time of lenience δ as,

$$M_{H,\delta} = \inf\{m > 0 : H(\boldsymbol{\tau}(m)) - \frac{\log(n)}{\log(n(n-1))} < \delta\}.$$

Intuitively, this is the first iteration in which the entropy of the pheromone matrix drops below an δ threshold of the entropy of a pheromone matrix in equilibrium state.

It is important to determine whether $M_{H,\delta}$ is finite to reduce the risk of the algorithm encountering an infinite loop. We know from Lemma 3.4 that conditioned upon the optimal walk being traversed in iteration m, we can continue to run the algorithm for some number of iterations after this to achieve pheromone values arbitrarily close to $\frac{1}{n}$ along the optimal path and arbitrarily small everywhere else. Although it may seem that this lemma is only useful when the algorithm has already discovered the optimal solution, we can apply the same logic on any solution the algorithm has deemed optimal at the time.

The stopping rule in Def 4.0.3 & this lemma are the student's original work.

Lemma 4.1. *The entropy driven stopping time of lenience δ, $M_{H,\delta}$, for our ACO algorithm is finite almost surely i.e.*

$$\mathbb{P}(M_{H,\delta} < \infty) = 1.$$

Proof. Say the ACO algorithm is in iteration m and finds a new best solution $\mathbf{w}'$. At this point Lemma 3.4 tells us that the pheromone concentration will to begin accumulating along this path $\mathbf{w}'$ and two cases can occur; case 1 is if no better solution is found, then in some number of iterations after m, this path will constitute for all but an arbitrarily small amount of the total pheromone concentration on the graph with high probability. If before this happens, case 2 occurs and a new optimal solution is discovered, then the process begins again. However, the remark on Definition 2.0.1 tells us there are only finitely many (albeit a vast number of) elements in the feasible solution space, so this can only happen finitely many times.

For case 1 we can show that the pheromone matrix entropy can be made arbitrarily close to the equilibrium entropy. We note that the function $x \mapsto -x\log(x)$ is non negative and strictly increasing on the interval $[0, e^{-1})$ and recall from Lemma 3.4, for any $\epsilon > 0$ we can find $d \in N$ such that for $\hat{m} \geq m + d$ with probability greater than $1 - \epsilon$ we can bound the following quantities as such,

$$\begin{cases} \frac{1}{n} - \epsilon \leq \tau_{ij}(\hat{m}) \leq \frac{1}{n} + \epsilon & (i,j) \in \mathbf{w}' \\ \tau_{ij}(\hat{m}) \leq n\epsilon & (i,j) \notin \mathbf{w}' \end{cases}.$$

Hence assume $n \geq 3$ and choose $\epsilon < \min\left\{(ne)^{-1}, e - n^{-1}\right\}$ thus causing, $\tau_{ij}(\hat{m}) \in [0, e^{-1})$

$$\begin{aligned}
H(\boldsymbol{\tau}(\hat{m})) &= -\sum_{1\leq i,j\leq n\,:\, i\neq j} \tau_{ij}(\hat{m}) \frac{\log(\tau_{ij}(\hat{m}))}{\log(n(n-1))} \\
&= \frac{1}{\log(n(n-1))}\left(\sum_{(i,j)\in\mathbf{w}'} -\tau_{ij}(\hat{m})\log(\tau_{ij}(\hat{m})) + \sum_{(i,j)\notin\mathbf{w}'} -\tau_{ij}(\hat{m})\log(\tau_{ij}(\hat{m}))\right) \\
&\leq \frac{1}{\log(n(n-1))}\left(\sum_{(i,j)\in\mathbf{w}'} -\left(\frac{1}{n}+\epsilon\right)\log\left(\frac{1}{n}+\epsilon\right) + \sum_{(i,j)\notin\mathbf{w}'} -(n\epsilon)\log(n\epsilon)\right) \\
&= \frac{1}{\log(n(n-1))}\left(-n\left(\frac{1}{n}+\epsilon\right)\log\left(\frac{1}{n}+\epsilon\right) - n(n-2)(n\epsilon)\log(n\epsilon)\right) \\
&= \frac{\log(n)}{\log(n(n-1))} + \frac{\log\left(n^{n\epsilon}(1+n\epsilon)^{-(1+n\epsilon)}(n\epsilon)^{-n^2(n-2)\epsilon}\right)}{\log(n(n-1))}
\end{aligned}$$

Noticing for any fixed $n \in \mathbb{N}$ and for small enough ϵ, the second term in the above sum is non negative and can be made arbitrarily small (in fact its limit as $\epsilon \to 0$ is 0), it is now clear that we can make the pheromone matrix entropy arbitrarily close to its equilibrium entropy. Lemma 3.4 then implies for such an ϵ, there exists a $d_{\epsilon,m}$ number of iterations after iteration m which by this point, the pheromone matrix entropy will be within the δ distance of its equilibrium with high probability thus proving $\mathbb{P}(M_{H,\delta} < \infty) \geq 1 - \epsilon$, since this holds for all $\epsilon > 0$, the result of this lemma follows. □

Although reassuring, this result is often not instructive and there are two main problems; almost sure finiteness still allows for the algorithm to be run for an arbitrarily long time under the right conditions which of course is something we want to avoid. Secondly, as previously discussed, as it is a heuristic algorithm, it does not assess the fitness of the solution proposed.
To address the former we will set a cap on the number of iterations for which entropy can remain constant before halting the algorithm.

Definition 4.1.1 (Capped entropy driven stopping time)**.** For an ACO algorithm solving the TSP on n nodes with pheromone matrix in iteration m given by $\boldsymbol{\tau}(m)$, $\delta > 0$ and $k \in N$, let $M_{H,\delta}$ be the entropy driven stopping time of lenience δ.
Let $M_k = \inf\{m > k : H(\boldsymbol{\tau}(m)) = H(\boldsymbol{\tau}(m-k))\}$ to be the first iteration with k consecutive iterations leading up to it on which the pheromone matrix entropy remains constant.
Now define the k-repetition capped entropy driven stopping time of lenience δ as

$$\overline{M}_{H,\delta,k} = M_{H,\delta} \wedge M_k$$

where $a \wedge b$ refers to the minimum of a and b.

$\overline{M}_{H,\delta,k}$ inherits almost sure finiteness from $M_{H,\delta}$ and further benefits the algorithm by preventing it from getting stuck in a state where pheromone matrix is not in equilibrium yet entropy is not changing. This occurs when a new optimal solution is discovered but is then not rediscovered again for many iterations which due to the elitist strategy of the ACO algorithm does not allow for pheromone concentration values to change and thus for entropy to change. This is one of the biggest flaws in the elitist strategy of the ACO algorithm.

4.4 Testing the exactness of the ACO algorithm's proposed solution for $n \leq 16$

Now that we have developed a stopping time for the algorithm based upon the certainty of the solution proposed, we must assess how far the ACO algorithm's proposed solution is from the true value of the solution. For a relatively small number of nodes, this can be verified using an exhaustive search of the solution space or better yet, the Held-Karp [4] algorithm which gives exact solutions to the TSP with computational complexity of $O(n^2 2^n)$. In this paper we will not go into the details of this algorithm but will use it to reduce running time of simulations.

For K_n^D, the complete random graph on n nodes on the Euclidean unit disc, let the solution to the TSP (also referred to as the shortest Hamiltonian cycle - SHC) be denoted as $\mathbf{w}_n^*$ with corresponding cost $f(\mathbf{w}_n^*)$. We consider the ACO algorithm's proposed SHC when compared with the exact value of the SHC whilst varying the number of nodes in the graph, n, as well as various other parameters including; ant colony population, S, pheromone evaporation constant, ρ and edge weight importance parameter, β. For each of the 189 unique combination of parameters,

$$(n, S, \rho, \beta) \in \{10, \ldots, 16\} \times \{50, 100, 200\} \times \{0.05, 0.10, 0.15\} \times \{0.5, 1.0, 1.5\},$$

150 independent and identically distributed K_n^D random variables were sampled with the ACO and Held-Karp algorithm and subsequently run upon its graph. This ensures independence of the collected data. The first statistic we observe is the proportion of trials in which ACO gave exact SHC, the proportion of trials in which ACO gave inexact SHC and the proportion of trials in which ACO was interrupted prematurely due to the stopping time M_k (k consecutive trials where entropy remains constant). Overall we use the capped entropy driven stopping time, $\overline{M}_{H,\delta,k}$ with $\delta = 0.03$ and $k = 50$. The percentage figure (to 1 decimal place) was calculated by grouping all simulated data by one parameter (shown in column header), choosing a value for that parameter and then averaging over all data entries with that given fixed parameter.

	n						
%	10	11	12	13	14	15	16
Exact	91.7	88.8	83.6	77.4	69.9	62.4	55.4
Inexact	0.8	1.9	3.8	7.0	9.5	12.8	17.1
Interrupted	7.5	9.3	12.6	15.6	20.6	24.8	27.5

Figure 7: Table showing the proportion of ACO simulations solving the TSP giving exact solutions for the SHC, inexact solutions for the SHC and the proportion of simulations that were prematurely interrupted compared against varying number of nodes, n, of K_n^D.

As an example, we can read data from the $n = 16$ column of Figure 7 as; of all the simulations run on K_{16}^D, (to 1 d.p.) 55.4% gave exact solutions to the TSP, 17.1% gave inexact solutions to the TSP and 27.5% were interrupted before they could give a solution to the TSP.

First of all we must must disclaim the fact that due to there being 4 independent parameters for each simulation, the statistics in a given column only see a narrow 'slice' through the

	S			ρ			β		
%	50	100	200	0.05	0.10	0.15	0.5	1.0	1.5
Exact	66.2	76.7	83.8	59.8	85.4	81.6	59.9	78.7	88.2
Inexact	11.3	7.3	4.2	0.7	6.9	15.1	12.0	7.0	3.8
Interrupted	22.5	16.0	12.0	39.5	7.7	3.3	28.2	14.3	8.0

Figure 8: Table showing the proportion of ACO simulations solving the TSP giving exact solutions for the SHC, inexact solutions for the SHC and the proportion of simulations that were prematurely interrupted compared against varying parameters of the algorithm.

data set. This can mislead us into thinking each parameter has a simple relationship with the exactness of proposed solutions but in reality, the 4 dimensional parameter input space $\mathbb{N}\times\mathbb{N}\times(0,1)\times(0,\infty)$ will have a non-straight forward mapping to the percentage/probability of exactness on $[0,100]$. Whats more, due to the inavailability of computation power, a limited number of values for each parameter were simulated (for n, 7 values, and for the other parameters, only 3 values). This again tells us we do not have enough data to draw confident conclusions about trends.
In spite of this, we will speculate some results based upon these averaged statistics with the claim that although many of the independent parameters are varying behind each statistic, they do so consistently and with the same number of trials for each unique combination.

Speculation 1. *The average percentage of number of simulations in which the algorithm gave exact solutions to the SHC decreases as the number of nodes n increases from* 10 *to* 16*.*

This behaviour is expected. As n grows, the number of possible solutions in the solution space grows like $\frac{(n-1)!}{2}$, so we would expect the algorithm to select non-exact solutions more as n grows as there are simply many more solutions to choose from.

Speculation 2. *The average percentage of number of simulations in which the algorithm gave exact solutions to the SHC increases as the ant colony population S increases from* 50 *to* 200*.*

This behaviour is expected. With more ants in the colony, in each iteration a greater sample of the solution space is taken resulting in the likelihood of selecting a better solution than one seen so far to increase. For example, an ACO algorithm with 200 ants in the colony will sample the same number of solutions from the solution space in one iteration as and ACO algorithm with 50 ants does in 4 iterations. The difference being that the distribution from which solutions are sampled is updated in between each iteration so the in the latter case, 200 sampled solutions spans over 4 different iterations and will not all be from identical distributions. Moreover, the stopping time cap M_k, when entropy of the pheromone concentration matrix remains unchanged for 50 consecutive iterations, is less likely to interrupt the ACO algorithm when the colony population S is 200 as opposed to 50, simply because there is greater opportunity to explore the solution space and thus propensity for better solutions to be found and pheromone concentrations to change. This is illustrated by the decreasing percentage of interrupted simulations as S grows. This agrees with the result in Theorem 3.8 that a larger colony size S results in the probability of sampling the exact solution to be bounded closer to 1. One note of caution is that as S increases, the computational load in one given iteration increases with direct proportion, which is why it is not feasible to continue increasing S and why the bounds suggested in the proof of Theorem 3.8 are purely theoretical, not practical.

Speculation 3. *The average percentage of number of simulations in which the algorithm gave inexact solutions to the SHC increases as the pheromone evaporation constant, ρ, increases from* 0.05 *to* 0.15*.*

Crudely, pheromone evaporation constant, ρ, influences the speed at which the algorithm can evolve. For larger ρ, more weight is placed upon the solutions sampled in more recent iterations when updating the probability distribution on the solution space. This allows pheromone concentration along edges to accumulate faster which in turn can result in faster convergence upon a particular solution giving less time for the algorithm to search the solution space for other, potentially better, solutions. This explains the higher proportion of trials in which inexact solutions were proposed by the algorithm for higher values of ρ. We do not comment here upon the possible increase in the average percentage of number of simulations in which the algorithm gave exact solutions to the SHC as ρ increases as the data is not monotone and needs a greater level of detail on the range of ρ to discuss.

Speculation 4. *The average percentage of number of simulations in which the algorithm gave exact solutions to the SHC increases as the edge weight importance parameter, β, increases from* 0.5 *to* 1.5*.*

This behaviour is expected. β controls the influence of edge weights in comparison to edge pheromone concentration when an agent calculates transition probabilities during the sampling of a solution from the solution space. A increased value of β would mean agents are more likely to choose closer vertices to move to (smaller edge weight) over vertices connected by stronger pheromone trail. If we set $\beta = 0$, then ants would make decisions souly based upon pheromone trails, disregarding the (maybe obvious) use of shorter edges. It is intuitive that for a small number of vertices, $n \leq 16$, agents constructing a solution should indeed choose edges of shortest weight with higher probability as the global topology of the solution is likely to be a simple convex cycle. However for larger, n, due to the increased density of nodes upon the D, we are likely to see the topology of the SHC to contain both locally convex and concave segments. We can therefore conjecture that simple greedy heuristic (i.e. indefinitely increasing β) may not be the best strategy for larger n.

Figures 7 and 8 are limited in the sense they have a simple tertiary approach to the data collected; solutions are partitioned into 3 disjoint sets; exact, inexact or interrupted. For those solution which are inexact but not interrupted, we can quantitatively analyse exactly how 'inexact' the ACO algorithm's proposed solution is. The percentage error of the ACO algorithm's solution is given by,

$$100 \cdot \frac{\text{Cost of ACO inexact solution} - \text{Cost of exact solution}}{\text{Cost of exact solution}}.$$

Here we pruned the data set to allow only ACO simulations which gave an inexact but non-interrupted solution. Subsequently, the pruned data was then grouped by a particular parameter, (e.g. number of nodes, n) and then the sample mean and sample standard deviation of the percentage error statistics were calculated to 2 decimal places. The results are as follows.

As an example, we could read a particular statistic from Figure 9 as follows; for all the uninterrupted ACO simulations which gave inexact solutions on K_n^D for $n = 16$, the sample

	n						
	10	11	12	13	14	15	16
Mean of % Error	1.66	1.38	1.23	1.66	1.59	1.69	1.76
Standard Dev of % Error	1.72	1.17	1.16	1.60	1.65	1.73	1.78

Figure 9: Table showing the sample mean and sample standard deviation of % error for the inexact solutions proposed by the ACO algorithm when number of nodes, n, for K_n^D is varied.

	S			ρ			β		
	50	100	200	0.05	0.10	0.15	0.5	1.0	1.5
Mean of % Error	1.90	1.45	1.31	0.89	1.33	1.83	1.98	1.40	1.06
Standard Dev of % Error	1.86	1.47	1.30	0.75	1.29	1.81	1.87	1.40	1.05

Figure 10: Table showing the sample mean and sample standard deviation of % error for the inexact solutions proposed by the ACO algorithm when ACO parameters are varied.

mean and sample standard deviation of the percentage difference between the exact solution and the ACO algorithm's inexact solution is given by 1.76% and 1.78% respectively. We would read data from Figure 10 analogously.
We have to apply the same disclaimer mentioned in association with Figures 8 and 7 about the varying of multiple independent parameters contained within each statistic with the additional disclaimer that each column of Figures 9 and 10 have differing sample sizes corresponding to the % Inexact row given in Figures 8 and 7.

Speculation 5. *The standard deviation of the percentage error in the cost of the ACO algorithm's inexact solution compared with the cost of the exact solution increases as the number of nodes n increases from* 10 *to* 16*.*

Although we cannot use this data to comment on behaviuor for the sample mean of % error, we can say that the sample standard deviation increases with n. This tells us that as n increases, the variability in quite how 'wrong' a proposed solution is, increases, with the exception of an anomaly at $n = 10$. This anomalous statistic is likely explained by examining the average percentage of number of simulations in which the algorithm gave inexact solutions to the SHC for $n = 10$ as seen in Figure 7, which is equal to only 0.8% or in true value, only 34 data points. It therefore could be inferred that the sample size for this static was not large enough to give confident estimates of the standard deviation / mean of the % error.

Speculation 6. *Both the standard deviation and the mean of the percentage error in the cost of the ACO algorithm's inexact solution compared with the cost of the exact solution decreases as the ant colony population S increases from* 50 *to* 200*.*

Speculation 7. *Both the standard deviation and the mean of the percentage error in the cost of the ACO algorithm's inexact solution compared with the cost of the exact solution increase as the pheromone evaporation constant ρ increases from* 0.05 *to* 0.15*.*

Speculation 8. *Both the standard deviation and the mean of the percentage error in the cost of the ACO algorithm's inexact solution compared with the cost of the exact solution decrease as the edge weight importance parameter, β, increases from* 0.5 *to* 1.5*.*

The explanation for Speculation 6,7,8 is much the same as that for speculation 2,3,4 respectively so will be omitted.
Overall, across the whole range of parameters we varied, we see a relatively small % error in the cost of the ACO algorithm's proposed solution and the cost of the exact value of the solution. As a blanket statement, we can therefore say that although ACO may not give exact solutions for a large proportion of the simulations, the inexact solutions are often very near approximations. This indeed should be the metric to which we measure the success of a heuristic algorithm as perhaps it is too stringent of an expectation to find the exact solution. This is the trade off with which we can suffice in exchange for a reduced computational complexity.

4.5 Bias in the interruption protocol of the ACO algorithm for $n \leq 16$

One potential bias that must be considered during this experiment is whether the ACO algorithm is more likely to be interrupted on particular instances of K_n^D than others. It was suspected that instances of of K_n^D with larger than average costs for the exact SHC would have a more contorted geometry for the exact value of the SHC therefore somehow making it more difficult for ACO to solve without getting stuck and consequently getting interrupted. Over all possible combinations of parameters, S, ρ, β and n, the frequency of interrupted and complete simulations (from a total of 28350) were plotted as a histogram against the exact value of the cost of the SHC. Figure 11 shows the result.

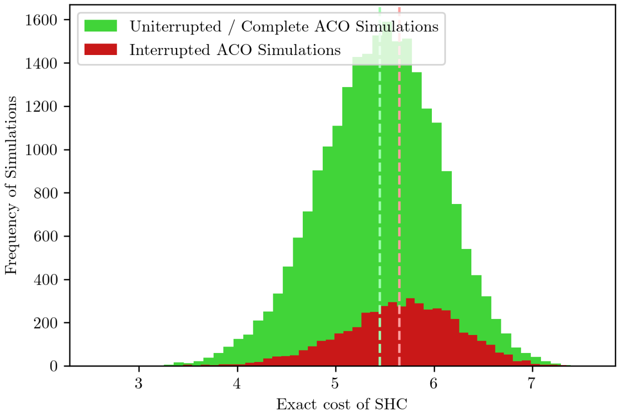


Figure 11: Historgram showing the frequency of ACO simulations which were uninterrupted versus those that were interrupted over all simulated parameters' ranges.

The dashed vertical lines in Figure 11 correspond to the mean exact cost of the SHC for the uninterrupted simulations (green dashed line ≈ 5.451) and the interrupted simulations (red dashed line ≈ 5.650) from left to right respectively, thus showing a larger average exact cost of SHC for the instances of K_n^D for which that ACO algorithm was interrupted. If there were no bias for the algorithm to be interrupted each set in this partition of the data would show a similar density plot. This is evidence to suggest our suspicions about the bias in the ACO algorithm are true. However, this plot is limited as it groups together data over varying n, and thus non identically distributed samples of K_n^D. To overcome this, the data was grouped by n and similar histograms were plotted as shown in Figure 12 and then a two sample Kolmogorov-Smirnov test on the samples for the exact cost of the SHC for simualtions which were interrupted vs uninterrupted was carried out. The null hypothesis was that the two samples (uninterrupted and interrupted) were drawn from identical probability distributions. The p-value can be understood by asking the following question; if the two samples were randomly sampled from identical populations, what is the probability that the two cumulative frequency distributions would differ by as much as was observed? This gives the intuition that for small p-values, we reject the null hypothesis meaning it is likely the two samples were drawn from different populations thus confirming ACO is more likely to be interrupted for some instances of K_n^D over others.

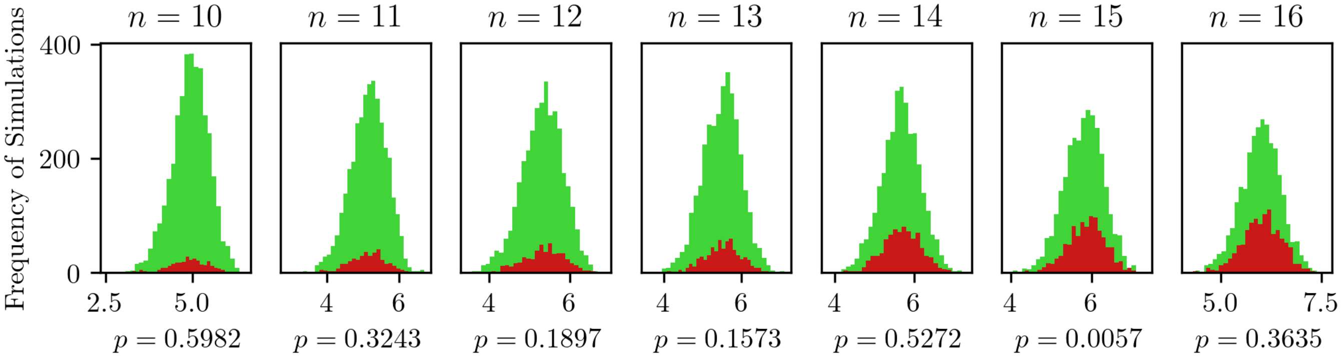


Figure 12: Historgrams showing the frequency of ACO simulations which were uninterrupted (green) versus those that were interrupted (red) against the exact cost of the SHC over all simulated parameters' ranges for each value of n. The p-value is that given by a 2 sample Kolmogorov-Smirnov test. Color labelling is the same as in Figure 11.

At a 5% significance level, the only case in which we can reject the null hypothesis is for $n = 15$. That is to say, the samples of K_{15}^D on which the algorithm was uninterrupted and interrupted were not drawn from identical distributions with a Kolmogorov-Smirnov p-value of 0.0057. The qualitative difference in these two samples are difficult to say, however we can quantitatively say that the mean exact cost of the SHC was ≈ 5.810 for the uninterrupted simulations and larger at ≈ 5.871 for the interrupted simulations in the group of $n = 15$. Due to the insignificance of the p-values for the other values of n, we cannot draw validated conclusions about the bias of the ACO algorithm's interruption protocol.

Furthermore, as each data set for a fixed n contained a variety of permutations for the parameters (S, ρ, β), we can only say that on average over these parameters ranges the results above show. Contradictory to this, it may in fact hold that certain combinations of parameters lead to the ACO algorithm having a greater chance of interruption on instances of K_n^D with a smaller exact cost of SHC. More simulations / computational power would be required to verify this.

One way to reduce this bias is as such; given an instance of K_n^D, if the ACO algorithm terminates prematurely due to the interruption protocol, we simply reinitialise and run the ACO algorithm again on the same instance of K_n^D repeating this process until the ACO algorithm successfully executes without interruption. This would ensure particular instances of K_n^D are not favoured/ left out when statistically analysing the results of a large number of ACO simulations. However, this approach was not taken in this investigation due to the inavailability of computation power, thus we must recognise the possibility of underlying bias in the results produced in later sections of this investigation.

4.6 The relationship between the Minimum Spanning Tree and Shortest Hamiltonian Cycle on K_n^D

The previous subsections' analysis of exactness and accuracy of the ACO algorithm is totally reliant upon the ability to verify exact solutions of the TSP. However, as previously discussed, this approach becomes infeasible for larger n, giving the initial motivation for a heuristic algorithm. Instead, we will develop a probabilistic model derived through a combination of theoretical results and regression of simulated data for smaller n.

Definition 4.1.2 (Minimum Spanning Tree)**.** For a graph $G = (V, E)$, the minimum spanning tree, $T^{\min}$, is defined as,

$$T^{\min} = \operatorname{argmin} \left\{ f(T) : T \subseteq E, T \text{ is a connected component and reaches every node} \right\},$$

where $f(T)$ is the cost objective function summing up weights of edges in the set T.

Tomescu [8] showed that for a symmetric euclidean TSP, we can bound the cost of the optimal solution using the minimum spanning tree (MST). Since the problem of finding the MST on K_n has been shown to be achievable in $O(n^2)$ computational complexity (based upon the number of edges in the graph) through the use of Prim's algorithm [9], for a large number of nodes, the following theorem provides a useful starting point for assessing the fitness of the proposed solution.

Theorem 4.2 (Tomescu [8] bounds on cost of optimal solution to TSP)**.** *Let K_n^D be the complete random graph on n nodes on the Euclidean unit disc.*
Let $\mathbf{w}_n^*$ be the optimal solution to the TSP and $T_n^{\min}$ the minimum spanning tree on K_n^D, with $f(\mathbf{w}_n^*)$ and $f(T_n^{\min})$ their respective costs (summed weights along edges).
Since all weights are euclidean distances, they are symmetric and obey the triangle inequality. We then have the following bound,

$$f(T_n^{\min}) \leq f(\mathbf{w}_n^*) \leq 2f(T_n^{\min}).$$

Definition 4.2.1. For a particular instance of K_n^D, the difference in minimum cost of TSP and MST is the random variable given by,

$$\Delta_n = f(\mathbf{w}_n^*) - f(T_n^{\min}).$$

Remark. A direct result of Theorem 4.2 is that Δ_n exists on the interval $[0, f(T_n^{\min})]$ according to some probability distribution of which we can estimate the expectation and variance through a Monte-Carlo approach. We will do so for $n \leq 16$, as again we require the ability to find exact solutions for the SHC in order to calculate Δ_n, and then build a regression based model to extrapolate expectation of Δ_n for larger n. The following theorem provides us with some insight into the asymptotic behaviour of $\mathbb{E}(\Delta_n)$.

Theorem 4.3 (Steele [6][7] - Expected cost of euclidean MST and SHC)**.** *If n points are sampled independently and identically at random from $X \subseteq \mathbb{R}^d$ according to probability density $g(x)$ then for large n,*

$$\mathbb{E}(f(T_n^{\min})) \sim a_d n^{\frac{d-1}{d}} \int_X g(x)^{\frac{d-1}{d}} dx,$$

$$\mathbb{E}(f(\mathbf{w}_n^*)) \sim b_d n^{\frac{d-1}{d}} \int_X g(x)^{\frac{d-1}{d}} dx,$$

where a_d, b_d are constants depending on the dimension d and,

$$\alpha(n) \sim \beta(n) \iff \lim_{n\to\infty} \frac{\alpha(n)}{\beta(n)} = 1.$$

Corollary 4.3.1. *The expected length of the minimum spanning tree for K_n^D is given by,*

$$\mathbb{E}(f(T_n^{\min})) \sim a_2\sqrt{n},$$

and the expected length of the shortest Hamiltonian cycle on K_n^D is given by,

$$\mathbb{E}(f(\mathbf{w}_n^*)) \sim b_2\sqrt{n},$$

which both in turn imply

$$\mathbb{E}(\Delta_n) \begin{cases} \sim (b_2 - a_2)\sqrt{n} & a_2 < b_2 \\ = o(\sqrt{n}) & a_2 = b_2 \end{cases}$$

for large n.

So we know all three expectation quantities grow asymptotically equivalently to $\sqrt{n}$ unless $a_2 = b_2$ in which case $\mathbb{E}(\Delta_n) = o(\sqrt{n})$ meaning,

$$\lim_{n\to\infty} \frac{\mathbb{E}(\Delta_n)}{\sqrt{n}} = 0.$$

$\mathbb{E}(\Delta_n)$ could potentially even decay to a constant value at $n = \infty$.

Remark. The variance of Δ_n is given by,

$$\begin{aligned} \operatorname{Var}(\Delta_n) &= \operatorname{Var}(f(\mathbf{w}_n^*) - f(T_n^{\min})) \\ &= \operatorname{Var}(f(\mathbf{w}_n^*)) + \operatorname{Var}(f(T_n^{\min})) - 2\operatorname{Cov}(f(\mathbf{w}_n^*), f(T_n^{\min})) \\ \Rightarrow \operatorname{Var}(\Delta_n) \leq \operatorname{Var}(f(\mathbf{w}_n^*)) &\iff \operatorname{Var}(f(T_n^{\min})) \leq 2\operatorname{Cov}(f(\mathbf{w}_n^*), f(T_n^{\min})) \end{aligned}$$

If we can show this condition of $\operatorname{Var}(\Delta_n) \leq \operatorname{Var}(f(\mathbf{w}_n^*))$, then we have validated the use of estimating Δ_n for larger n as a supplementary to $f(\mathbf{w}_n^*)$ since a smaller variance will lead to less uncertainty in our estimation. Indeed, it is intuitive that $\operatorname{Cov}(f(\mathbf{w}_n^*), f(T_n^{\min})) > 0$ due to the obvious dependence of $f(\mathbf{w}_n^*)$ and $f(T_n^{\min})$. Whether the covariance satisfies the above inequality needs experimental results to verify.

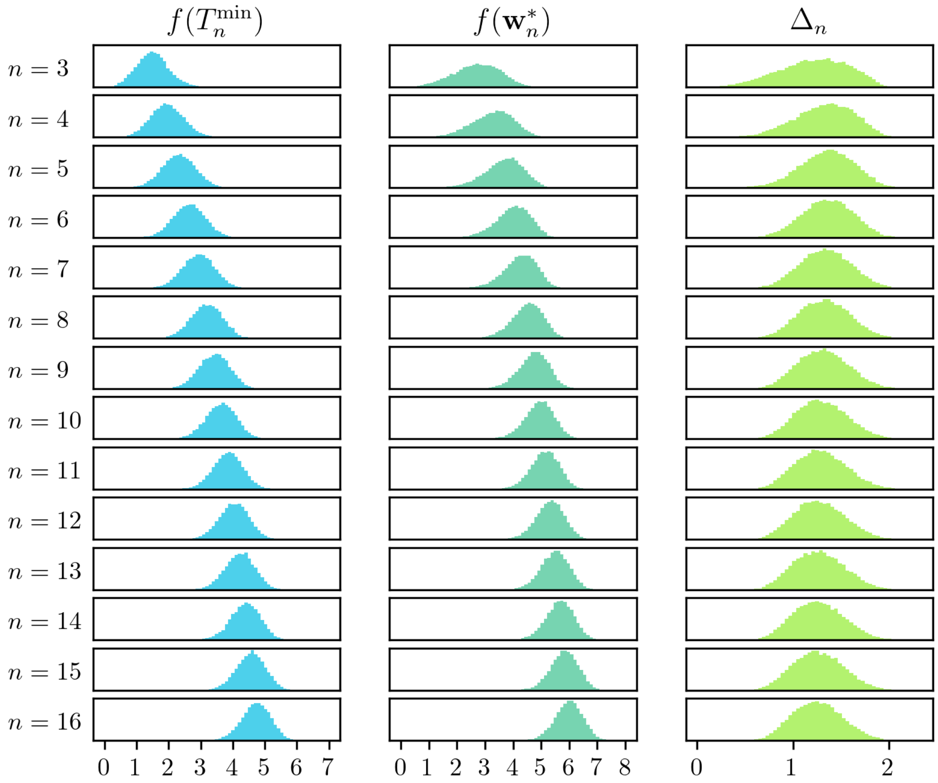


Figure 13: Density plots showing the results of the 20000 simulated K_n^D random variables cost of MST, cost of SHC and their difference Δ_n for each value of $n \in \{3, 4, \ldots, 16\}$

4.7 Simulated data on the MST and SHC for $n \leq 16$

For $n \in \{3, 4, \ldots, 16\}$, 200000 independent and identically distributed copies of K_n^D were sampled and the exact values of $f(\mathbf{w}_n^*)$ and $f(T_n^{\min})$ were calculated, from this we were able to calculate the value of Δ_n for each data point. The resulting density plots are given by Figure 13. Note, within each column of Figure 13, the x-axis remains fixed for each plot thus for neatness, it is explicitly measured on plots only in the bottom row. Having compared these densities with several standard continuous distributions, to much disappointment, none produced a significant p-value when passed through a Kolmogorov–Smirnov test. Thus it was concluded these random variables do not belong to a standard continuous distribution and instead we must infer results from the simulated data alone.

Figure 13 illustrates that the cost of the MST, $f(T_n^{\min})$, and the cost of the SHC, $f(\mathbf{w}_n^*)$, grow with n. This is expected due to Corollary 4.3.1. From comparing the variance of $f(\mathbf{w}_n^*)$ and Δ_n, we generated Figure 14.

One explanation for the behaviour exhibited in figure 14 is due to the very implicit dependence of $f(\mathbf{w}_n^*)$ and $f(T_n^{\min})$, since instances of K_n^D that are likely to give abnormally large / small values of $f(T_n^{\min})$ are also likely to give abnormally large / small values of $f(\mathbf{w}_n^*)$. There is a clear dependence which leads to the $\operatorname{Cov}(f(\mathbf{w}_n^*), f(T_n^{\min}))$ term the equa-

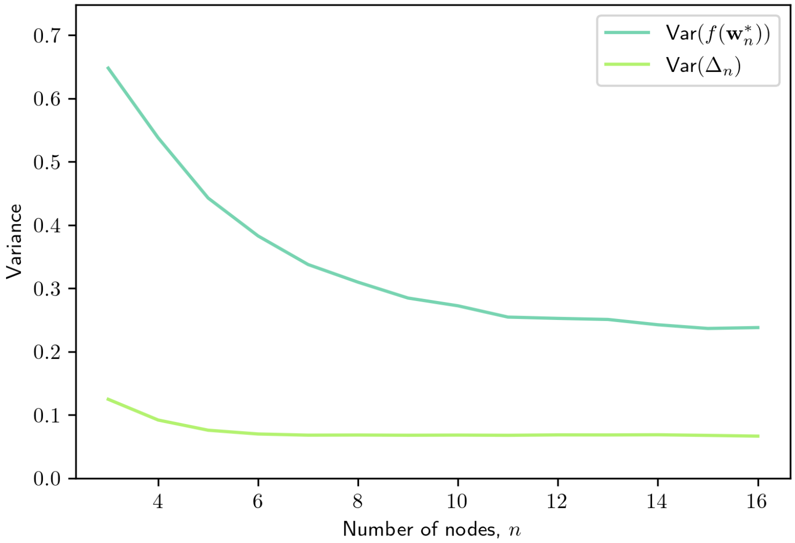


Figure 14: Comparison of the sample variance of $f(\mathbf{w}_n^*)$ and Δ_n as n increases from 3 to 16

tion in the remark on Corollary 4.3.1 being strictly positive.
Since there is less variability in the random variable Δ_n than $f(\mathbf{w}_n^*)$ for $n \leq 16$, we can conjecture that this would also hold for larger n.

We will now build a predictive regression model based upon the simulated data shown in Figure 13 to extrapolate the expected value of $f(\mathbf{w}_n^*)$ for larger n. Corollary 4.3.1 tells us that $\mathbb{E}(f(\mathbf{w}_n^*)) \sim \sqrt{n}$. We will therefore perform curve fitting via nonlinear least squares regression on the function,

$$\mathbb{E}(f(\mathbf{w}_n^*)) \approx \alpha(n) = k_1\sqrt{n-k_2} + k_3.$$

The transformation due to parameters k_1, k_2, k_3 allows for shifts and scaling of the curve whilst indeed the condition of $\alpha(n) \sim k_1\sqrt{n}$ still holds since,

$$\lim_{n\to\infty} \frac{k_1\sqrt{n-k_2}+k_3}{k_1\sqrt{n}} = \lim_{n\to\infty} \sqrt{1-\frac{k_2}{n}} + \frac{k_3}{k_1\sqrt{n}} = 1.$$

The assumption of this regression curve is limited for two reasons; firstly, it does not account for $o(\sqrt{n})$ terms (other than the constant term k_3) thus the behaviour may in fact differ outside of the range of $n \leq 16$. Whats more, we only know that $\mathbb{E}(f(\mathbf{w}_n^*))$ behaves like $k_1\sqrt{n}$ asymptotically so it is perhaps unwise to assume similar behaviour for small n, thus leading us to a potential over fitting of the data for $n \leq 16$ which may result in misleading extrapolation results for large n. In spite of this and due to the inability to simulate data to estimate $\mathbb{E}(f(\mathbf{w}_n^*))$ for larger n, this is the most astute method of estimation and extrapolation available, so we will continue the investigation.
Based upon the 20000 sampled K_n^D for each $n \in \{3, \dots, 16\}$, the values of the parameters,

k_1, k_2, k_3, that minimised the mean square error of $\alpha(n)$ were,

$$\begin{cases} k_1 &= 1.1023647 \\ k_2 &= 2.47805211 \\ k_3 &= 1.92260287 \end{cases}.$$

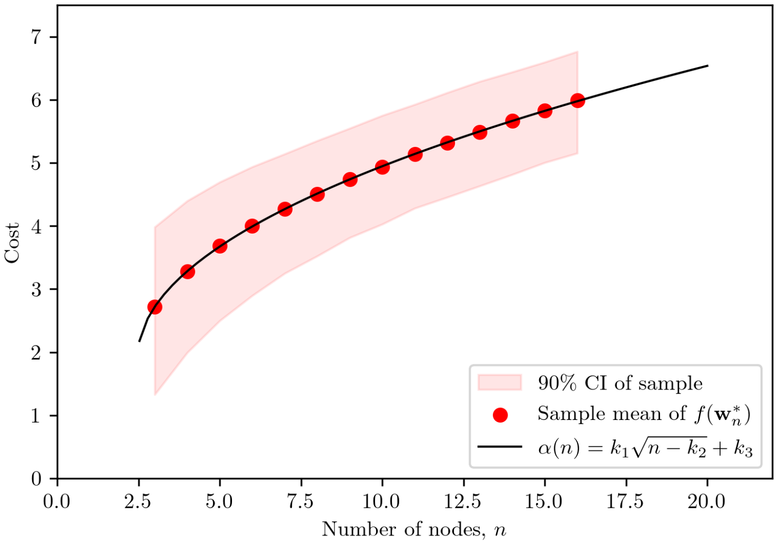


Figure 15: Plot showing a 90% confidence interval for the cost of the SHC for simulated data along with the sample mean approximating $\mathbb{E}(f(\mathbf{w}_n^*))$ and the regression curve $\alpha(n)$.

From Figure 15 we see that the regression curve given by $\alpha(n)$, parameterised by k_1, k_2, k_3, remains within the red region containing 90% of the sample data ($5^{\text{th}} - 95^{\text{th}}$ percentile) and virtually intersects the sample mean of $f(\mathbf{w}_n^*)$ for every $n \in \{3, \ldots, 16\}$. Since this is a non-linear model we cannot find the Pearson correlation coefficient to assess the goodness of fit quantitatively. However, upon visual inspection, it is clear that $\alpha(n)$ is indeed a good fit for the sample mean within this range of n.

We will now build our second predictive regression model based upon the simulated data shown in Figure 13 with the new aim of extrapolating the expected value of Δ_n for larger n. Corollary 4.3.1 tells us that $\mathbb{E}(\Delta_n) \sim k_4\sqrt{n}$. There is the possibility that $k_4 = 0$ in which case $\mathbb{E}(\Delta_n) = o(\sqrt{n})$, however in this investigation we will make the assumption that $k_4 > 0$. We will therefore perform curve fitting via nonlinear least squares regression on the function,

$$\mathbb{E}(\Delta_n) \approx \beta(n) = k_4\sqrt{n} + k_5 n^{k_6} + k_7,$$

where k_4, k_5, k_6, k_7 parameterise the model and we restrict $k_4 > 0$ and $k_6 < 0.5$. The condition of $\beta(n) \sim k_4\sqrt{n}$ still holds since,

$$\lim_{n\to\infty} \frac{k_4\sqrt{n} + k_5 n^{k_6} + k_7}{k_4\sqrt{n}} = \lim_{n\to\infty} 1 + \frac{k_5}{k_4} n^{k_6 - 0.5} + \frac{k_7}{k_4\sqrt{n}} = 1.$$

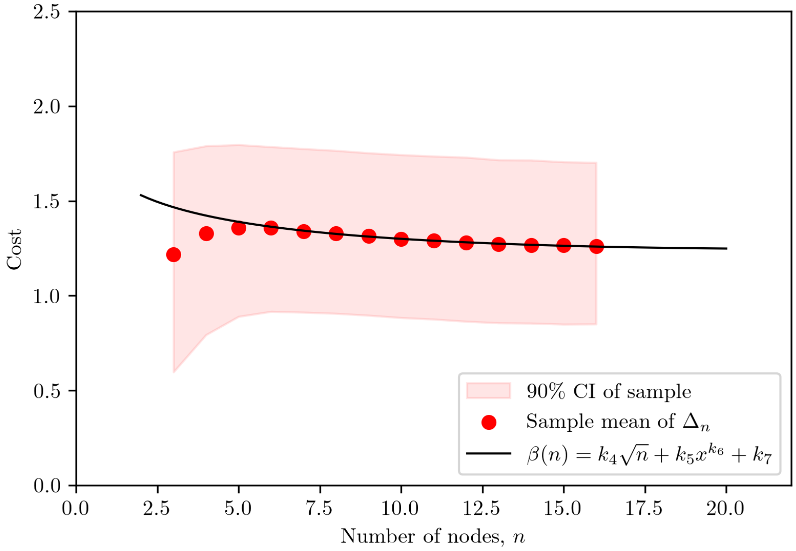


Figure 16: Plot showing a 90% confidence interval for Δ_n for simulated data along with the sample mean approximating $\mathbb{E}(\Delta_n)$ and the regression curve $\beta(n)$.

This model has all of the same limitations mentioned with regards to the $\alpha(n)$ model, with the additional limitation that we do not know for certain $k_4 > 0$ as it is possible $\mathbb{E}(\Delta_n) = o(\sqrt{n})$. We have introduced the extra polynomial term $k_5 n^{k_6}$ to give $\beta(n)$ the freedom to be a decreasing function for small n. We have ensured that $k_6 < 0.5$ as to not increase the asymptotic growth rate of $\beta(n)$ past the upper bound of $\sqrt{n}$. The extra degree of freedom in this model is even more likely to cause overfitting to the generated dataset which is a strong argument as to why this model is limited. However, as before, we will continue investigating this model for variety of analysis of the SHC for large n.
Based upon the 20000 sampled K_n^D for each $n \in \{7, \dots, 16\}$, the values of the parameters, k_4, k_5, k_6, k_7, that minimised the mean square error of $\beta(n)$ were,

$$\begin{cases} k_4 & = 0.45820196 \\ k_5 & = -1.10867801 \\ k_6 & = 0.34189248 \\ k_7 & = 2.28701057 \end{cases}.$$

The reason we chose to leave out data for $n \leq 6$ when fitting the parameters of the model is clear upon inspection of Figure 16. There is an increasing section of the sample mean plot which in order to fit to would require additional polynomial terms and thus additional degrees of freedom which is again more likely to increase the chances of over-fitting the model to our data.

4.8 Testing the exactness of the ACO algorithm's proposed solution for larger n

Having now developed the models $\alpha(n)$ and $\beta(n)$ for extrapolating the value of $\mathbb{E}(f(\mathbf{w}_n^*))$ and $\mathbb{E}(\Delta_n)$ respectively for larger n, we have the ability to take this investigation of the exactness of the ACO algorithm's proposed solutions further.

Suppose for some $n \geq 16$ we sample an instance of K_n^D. We can then allow the ACO algorithm to propose an approximation of the optimal solution to the TSP, say $\tilde{\mathbf{w}}_n$. If a large number of K_n^D are independently and identically sampled, we can then find a sample mean approaching the expectation $\mathbb{E}(f(\tilde{\mathbf{w}}_n))$ by the strong law of large numbers. This sample mean can then be compared with the model $\alpha(n)$.

Similarly, for each instance of K_n^D, we compute the exact MST, $T_n^{\min}$, (which is easily done with $O(n^2)$ complexity) and calculate an approximation of Δ_n based upon ACO suggested solution as such,

$$\tilde{\Delta}_n = f(\tilde{\mathbf{w}}_n) - f(T_n^{\min}).$$

Since we have conjectured that there is small variability in the true value of the random variable Δ_n for large n and since we have extrapolated $\mathbb{E}(\Delta_n)$ using simulated data for $n \leq 16$, we can then compare the ACO algorithm's approximation, $\tilde{\Delta}_n$, and our $\beta(n)$ model's estimated value of $\mathbb{E}(\Delta_n)$ to test the exactness our solutions.

Before we embark upon this, we must disclaim that using the non-certain predictive models (that as mentioned already, have much potential to be flawed) $\alpha(n)$, $\beta(n)$, as a tool to measure the exactness of this heuristic algorithm's solution is analogous to measuring an unknown length with a ruler for which we cannot say for certain whether its marked increments are precise / accurate. Thus we comment on the results of this experiment cautiously with this limitation planted in our mind.

Due to the limited computing power available, instead of varying all parameters of the algorithm as was done in subsection 4.4, we have chosen a particular combination of parameters,

$$(S, \rho, \beta) = (150, 0.1, 2).$$

This set of parameters were chosen using Figure 8 as a guide with the aim of balancing the desire to maximise the exactness of the ACO algorithm's proposed solutions and the computational intensity of the simulations. For the rest of this subsection, these parameters will remain fixed and the number of nodes, n, in K_n^D will be varied over the range,

$$n \in \{20, 25, 30, 35, 40, 45\},$$

as our independent variable. For each n in this range, 1000 independent and identically distributed samples of K_n^D were taken and upon each, ACO approximated the SHC and calculated its cost, $f(\tilde{\mathbf{w}}_n)$. Then the cost of the exact MST, $f(T_n^{\min})$, and the approximated difference term $\tilde{\Delta}_n$ were subsequently calculated.

As before, the ACO algorithm was executed using capped entropy driven stopping time $\overline{M}_{H,\delta,k}$ with $\delta = 0.03$ and $k = 50$. Because of the cap on this stopping time (when entropy remains the same for k consecutive iterations), there is a certain proportion of the simulations in which the algorithm was interrupted. Figure 17 shows how this proportion varies with n. As already discussed in subsection 4.5, since the ACO algorithm's interruption protocol has been shown to have a bias leading to a non-uniform sampling / representation of K_{15}^D

in ACO produced results, there is a risk that this bias also exists for larger n. This is likely to distort the data more on values of n for which the percentage of interrupted ACO simulations is larger. We have already discussed the measures that may be taken to reduce this bias, but due to the limited computation power, this investigation was unable to take them.

n	20	25	30	35	40	45
% of ACO Simulations Uninterrupted	96.1	95.4	90.7	87.8	82.5	75.8
% of ACO Simulations Interrupted	3.9	5.6	9.3	12.2	17.5	24.2

Figure 17: A table showing how the percentage of the ACO simulations that were uninterrupted and interrupted varies with n.

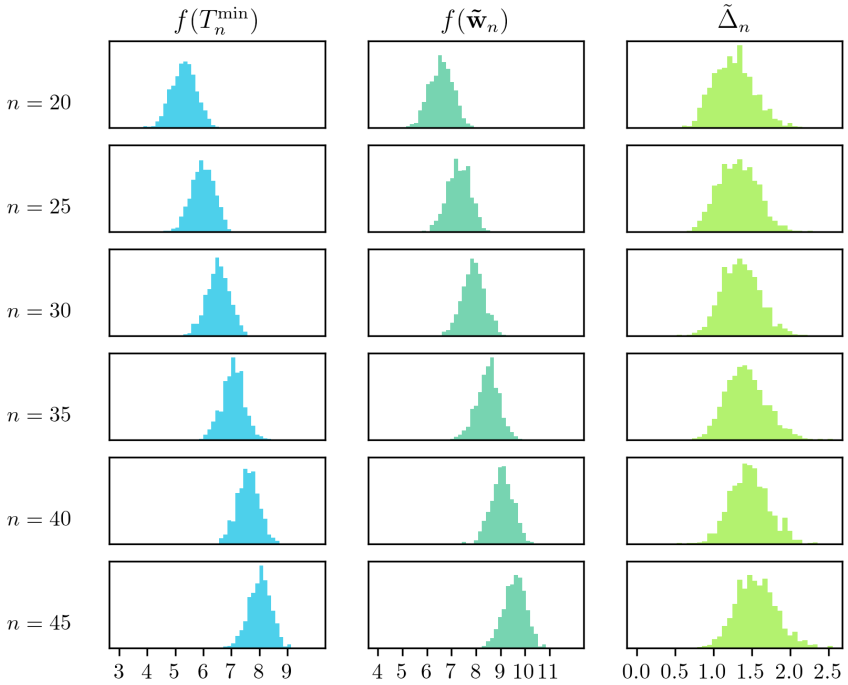


Figure 18: Density plots showing the results of the 1000 simulated K_n^D random variables cost of MST, $f(T_n^{\min})$, cost of ACO approximated SHC, $f(\tilde{\mathbf{w}}_n)$, and their difference $\tilde{\Delta}_n$ for each value of $n \in \{20, 25, 30, 35, 40, 45\}$.

Figure 17 shows that for this particular fixed set of parameters (S, ρ, β), as the number of nodes, n, increases, the proportion of simulations for which the interruption protocol was invoked increases. This is expected due to the increasing complexity of the solution to the TSP on instances of K_n^D with larger n which can result in an increased likelihood of the algorithm getting 'stuck'.

Figure 18 shows the density plots for the variables $f(T_n^{\min}), f(\tilde{\mathbf{w}}_n), \tilde{\Delta}_n$ over the experimental range of n. Within each column of plots, the x-axis remains fixed so for neatness, it is

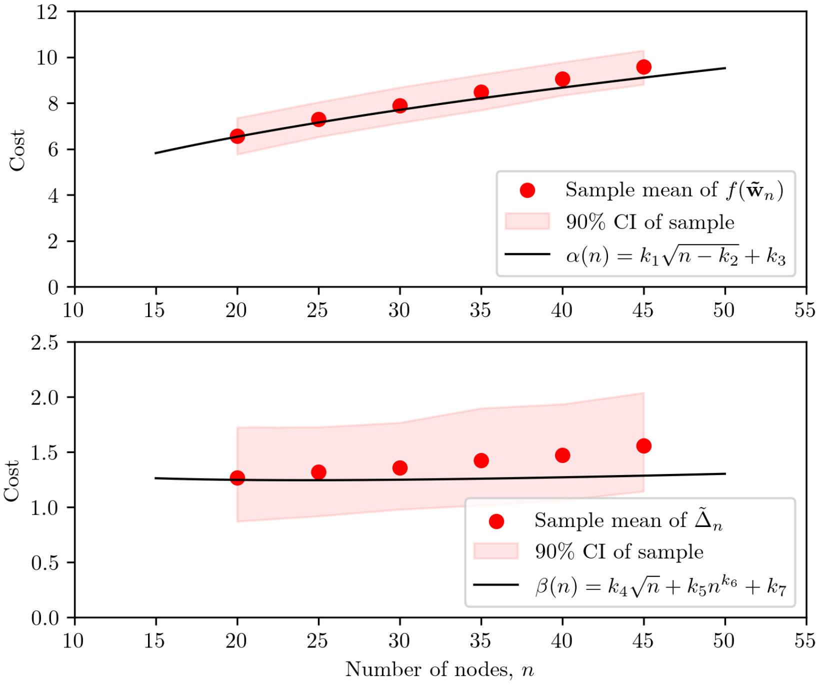


Figure 19: Plots showing how ACO approximated sample mean and 90% CI of $f(\tilde{\mathbf{w}}_n)$ and $\tilde{\Delta}_n$ compare to the predictive nonlinear models $\alpha(n)$ and $\beta(n)$ respectively.

explicitly measured on plots only in the bottom row.

From Figure 19, we can see that the region containing the central 90% of the sample data ($5^{\text{th}} - 95^{\text{th}}$ percentile) for both $f(\tilde{\mathbf{w}}_n)$ and $\tilde{\Delta}_n$ contains the curve described by models $\alpha(n)$ and $\beta(n)$ respectively. However, in both cases the mean of the sample drifts upwards faster than the predictive mean given by the nonlinear model with n. If we calculate the error in the sample mean when compared with their respective model using the formulas,

$$\text{Error in sample mean of } f(\tilde{\mathbf{w}}_n) = \text{Sample mean of } f(\tilde{\mathbf{w}}_n) - \alpha(n),$$

$$\text{Error in sample mean of } \tilde{\Delta}_n = \text{Sample mean of } \tilde{\Delta}_n - \beta(n),$$

we get the table in Figure 20.

n	20	25	30	35	40	45
Error in sample mean of $f(\tilde{\mathbf{w}}_n)$	0.040	0.145	0.187	0.280	0.394	0.472
Error in sample mean of $\tilde{\Delta}_n$	0.021	0.076	0.109	0.169	0.202	0.274

Figure 20: Table showing the error in ACO approximated sample mean of $f(\tilde{\mathbf{w}}_n)$ and $\tilde{\Delta}_n$ when compared to the predictive models $\alpha(n)$ and $\beta(n)$ respectively.

From Figure 20, it is evident that the ACO algorithm's approximated solutions are more costly in comparison to both models $\alpha(n)$, $\beta(n)$. Since the units in both rows of the table are the same, we can compare the model $\alpha(n)$ and $\beta(n)$ directly.

Since the error in sample mean of $\tilde{\Delta}_n$ is smaller and grows slower than the error in sample mean of $f(\tilde{\mathbf{w}}_n)$ for all values of n, we can conclude that the model $\beta(n)$ suggest the ACO algorithm gives stronger approximations to the exact cost of the SHC on K_n^D than the model $\alpha(n)$ suggests. One explanation for this is that the quantity $\tilde{\Delta}_n$ takes into account the MST for instances of K_n^D. We have already shown the dependence of $f(T_n^{\min})$ and $f(\mathbf{w}_n)$ and concluded that they have a positive covariance due to the Remark on Corollary 4.3.1 and experimental results for shown in Figure 14. Since $f(\tilde{\mathbf{w}}_n)$ approximates $f(\mathbf{w}_n)$, we can assume that $f(T_n^{\min})$ and $f(\tilde{\mathbf{w}}_n)$ have a positive covariance. Overall, this gives the $\beta(n)$ model the flexibility to contextualise instances of $f(\tilde{\mathbf{w}}_n)$ that were abnormally larger than the predicted mean as 'bad' instances of K_n^D rather than bad approximations by the ACO algorithm. The model $\alpha(n)$ does not consider the MST and consequentially fails to contextualize abnormally large instances of $f(\tilde{\mathbf{w}}_n)$ in the way we just described.

The fact that the error in sample mean for both ACO approximated quantities begins to grow as n grows could suggest two possibilities. The models $\alpha(n), \beta(n)$ may become worse at estimating the true expected value of $f(\mathbf{w}_n)$ and Δ_n as n grows due to an over-fitting to data for $n \leq 16$. This results in $\alpha(n), \beta(n)$ underpredicting the true expected values they aim to model. Alternatively, it is possible that ACO does indeed provide worse approximations as n grows due to the super-exponentially increasing number of possible Hamiltonian paths in the solution space therefore leading to an increased likelihood of selecting sub optimal solutions. In reality, both explanations are likely responsible for the observed behaviour to some degree.

4.9 Experimental Conclusions

This section has explored the ability of a particular ACO solving the TSP on the random euclidean graph K_n^D for varying parameter inputs of the ACO algorithm S, ρ, β and varying number of nodes n. Overall, this investigation has demonstrated the viability of ACO with much success at approximations of the shortest Hamiltonian path, with exact solves in some cases. Although this investigation has not included a comparative analysis of computation times for this heuristic algorithm and exact solve algorithms such as Held-Karp, the fact that ACO simulations, for graphs as large as $n = 45$, showed evidence to suggest the approximated solutions were strong is incredible.

Section 3 showed the capability of this ACO algorithm to give the optimal solution with probability arbitrarily close to 1 through maximum exertion of its parameter. It was then demonstrated that, if some degree of compromise is acceptable in the accuracy of ACO solution, then the parameters of the algorithm did not need to be pushed to unrealistic scales in order to get good results with high probability.

Throughout the investigation, we have self-evaluated the legitimacy and limitations of the approach taken whilst simultaneously addressing how one might improve the investigation through the removal of bias or the expansion of parameter domain. Whilst using as much theory as was available, we devised predictive models in Section 4.7 and commented on the uncertainty recognising that there is much need for refinement but also that as a basic test of the ACO algorithm, this was still informative.

Overall, the genius of ant colony optimisation has proven to be a fascinating and propitious approach to this instance of combinatorial optimisation offering approximate solutions that are often indistinguishable from exact. It serves as a reminder to us that some of the most elegant solutions to problems in mathematics were masterfully devised by nature long before being pondered by the human mind.

References

[1] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: An autocatalytic optimizing process. *Technical Report 91-016*, 1991.

[2] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[3] Walter J Gutjahr. A graph-based ant system and its convergence. *Future generation computer systems*, 16(8):873–888, 2000.

[4] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

[5] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[6] J. Michael Steele. Growth Rates of Euclidean Minimal Spanning Trees with Power Weighted Edges. *The Annals of Probability*, 16(4):1767 – 1787, 1988.

[7] J. Michael Steele. Probability and problems in euclidean combinatorial optimization. *Statistical Science*, 8(1):48–56, 1993.

[8] Ioan Tomescu. An upper bound for the shortest hamiltonian path in the symmetric euclidean case. *RAIRO-Operations Research*, 17(3):297–306, 1983.

[9] Ben Zinberg, Dana Moshkovitz, and Bruce Tidor. Design and analysis of algorithms: Minimum spanning trees 2. *MIT Electrical Engineering and Computer Science Mathematics Lecture notes*, 2012.