The Ant Colony Optimisation for the Travelling Salesman Problem

Dylan Galea

November 12, 2018

Contents

1 Introduction

Before defining the Travelling Salesman Problem and proving properties about it, a number of graph theoretic concepts that will be used throughout, must first be defined. Therefore, what follows is a sub-section that introduces a number of graph theoretic concepts which are required for the Travelling Salesman Problem.

1.1 Some Graph Theory

Graph theory is the study of a structure called a graph. A graph can be defined formally as shown in definition 1.1.1 below.

Definition 1.1.1. *A graph G is a pair (V,E) were V is any non empty finite set called the set of vertices of G, and* $E \subseteq \{\{u,v\} : \forall\ u,v \in V$ *and* $u \neq v\}$ *is called the set of edges of G* [1]*. A graph G defined by the pair (V,E) is denoted by G(V,E) or G.*

A graph defined using definition 1.1.1 is called an undirected graph. There is also the concept of a directed graph were $E \subseteq \{(u,v) : \forall u, v \in V, u \neq v\}$ [1]. However, in this thesis it can be assumed that any graph that will be considered is undirected unless otherwise stated. It can also be assumed that there are no edges between same vertices unless otherwise stated. It must also be noted that by this definition, there cannot be multiple edges joining any 2 vertices. The reason is that sets do not allow repetition of elements. Thus, each element in the edge set is unique. The discussion will now proceed by introducing more graph theoretic terminology, with examples that illustrate these terminologies.

When 2 vertices are joined by an edge, they are said to be adjacent. This is defined formally in definition 1.1.2 below.

Definition 1.1.2. *Given a graph G(V,E),* $\forall\ u,v \in V$*, u and v are said to be adjacent if* $\{u,v\} \in E$*.* [2]

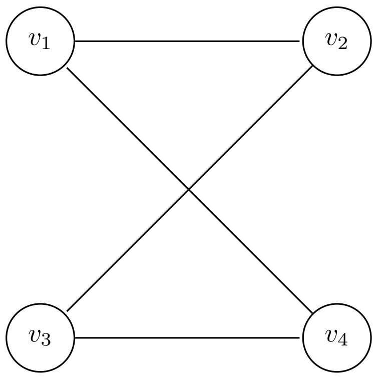
It is sometimes also required to know how many vertices are adjacent to a specific vertex in a graph.

Definition 1.1.3. *The degree of a vertex v in a graph G is the number of adjacent vertices to v in G* [3]*. The degree of a vertex v is denoted as deg(v).*

Any graph $G(V,E)$ can also be represented pictorially by drawing the vertices of G using circles, and by drawing the edges of G using lines between adjacent vertices. As a result, in this thesis a graph is sometimes given formally using sets or as a pictorial representation, assuming that one can be converted into another. Example 1.1.4 below depicts how a graph can be represented pictorially.

Example 1.1.4. Consider the graph *G(V,E)* such that $V=\{v_1, v_2, v_3, v_4\}$ and $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\}$.

Then G can be represented pictorially as :



Using definition 1.1.2, 2 adjacent vertices in G are v_1 and v_2. On the other hand, 2 non adjacent vertices in G are v_1 and v_3.
Using definition 1.1.3, the degree of every vertex in G is 2.

There are many other examples of graphs, one of them being the complete graph on n vertices.

Definition 1.1.5. *A graph G(V,E) is said to be complete if* $\forall$ *v,w* $\in$ *V* $v \neq w$, *v is adjacent to w. The complete graph on n vertices is denoted by* K_n. [4]

Given any graph $G(V, E)$, one can also define graph theoretic terminologies that lie within G, one of them being a path.

Definition 1.1.6. *Given a graph G(V,E), a path in G joining any 2 vertices u, v* $\in$ *V, is a sequence of vertices* $u = u_1$, u_2, ...,$u_n = v$ *in which no vertex is repeated and,* $\forall\ 0 < i < n$, $\{u_i, u_{i+1}\} \in E$. [5]

Definition 1.1.6 can now be used to define cycles and connectivity in a graph.

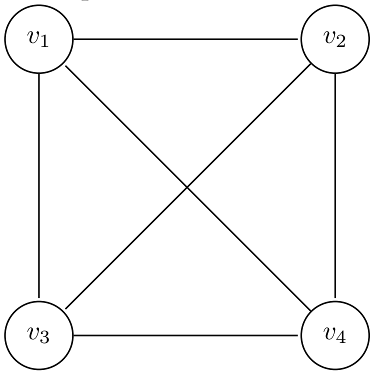
Definition 1.1.7. *A graph G(V,E) is said to be connected if* $\forall$ *u,v* $\in$ *V* $u \neq v$, *u and v are joined by a path.* [6]

Definition 1.1.8. *Given a graph G(V,E), a cycle in G is a path on* $n \geq 4$ *vertices, such that, the first vertex and the last vertex are equal.* [7]

By definitions 1.1.6, 1.1.8 above, it is clear that a cycle is a special instance of a path, with the only difference being that in a cycle, the first vertex and the last vertex are equal. Another thing worth mentioning is that, according to definitions 1.1.6 and 1.1.8, cycles and paths are sequences of vertices and not actual graphs. However, this is not the case because they can be represented easily as graphs. For example, given the path/cycle $u_1, u_2, ..., u_n$ a new graph $G(V, E)$ can be created such that, $V(G) = \{u_1, u_2, ..., u_n\}$ and $E(G) = \{\{u_i, u_{i+1}\} : \forall i, 0 < i < n\}$. For example, consider the cycle v1 v2 v3 v4 v1, then the graph depicted in example 1.1.4, is the graph representing this cycle. Such graphs are known as Cycle/Path graphs and are denoted by C_n/P_n

respecitvely, n being the number of vertices in the graph. Since this construction can be done, cycles/paths will be treated as both graphs and sequences. This will later be useful when defining Hamiltonian cycles. For better understanding of definitions 1.1.5, 1.1.6, 1.1.7 and 1.1.8, example 1.1.9 is constructed.

Example 1.1.9. Consider the graph *G(V,E)* below:



Since every vertex in G is adjacent to every other vertex, G must be complete. Therefore G must be K_4. Since G is complete, it must also be connected because, there is a path P_2 between any 2 distinct vertices of G.
Some examples of paths in G are:
1. v_1 v_2 v_3 v_4
2. v_1 v_4
3. v_4 v_3 v_1
Some examples of cycles in G are:
1. v_1 v_2 v_3 v_4 v_1
2. v_1 v_4 v_2 v_3 v_1
3. v_4 v_3 v_1 v_4

Another important graph theoretic concept is that of subgraphs.

Definition 1.1.10. *Given a graph G(V,E) and a graph H(V′,E′), H is a subgraph of G if* $V' \subseteq V$ *and* $E' \subseteq E$*.* [8]

After defining some important concepts, the next step is to extend definition 1.1.1 to define another class of graphs called weighted graphs. It must be noted that all definitions presented so far apply also to weighted graphs.

Definition 1.1.11. *Given a graph G(V,E), a weight function is a function* $f : E \mapsto \Re$ [8]*. The real numbers assigned to each edge are called weights.*

Definition 1.1.12. *A weighted graph is a graph G(V,E) with a weight function f* [8]*. This is denoted by the triple G(V,E,f) or G.*

According to Bondy and Murty [9], weighted graphs occur regularly in applied graph theory. For example, a railway network can be represented by a weighted graph were, the vertices are the set of towns in the railway network, and there are edges between 2 vertices in the graph if, there is a direct route from one town to another, without visiting other towns in the process. The weight function would then represent the cost of travelling directly from one town to another. In addition to that, the shortest path between 2 towns in the network may be required. It is clear that in order to try and solve such problems, the total weight of a subgraph must first be defined.

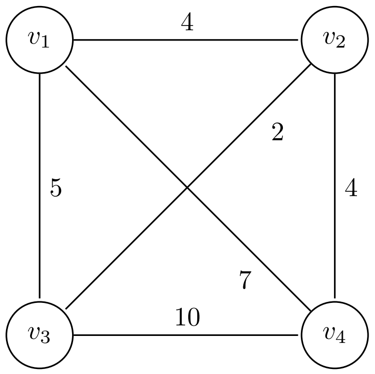
Definition 1.1.13. *Given a weighted graph G(V,E,f), the total weight of any subgraph H(V′,E′,f) of G is:*

$$\sum_{e \in E'} f(e)$$

. [9]

It is important to note that by definition 1.1.10, any weighted graph G is a subgraph of itself, therefore, it's weight can be calculated. This is highlighted in Example 1.1.14 below.

Example 1.1.14. Consider the weighted graph *G(V,E,f)* such that, *G(V,E)* is the graph in example 1.1.9 with weight function *f* such that,
$f(\{v_1, v_2\}) = 4$
$f(\{v_1, v_3\}) = 5$
$f(\{v_2, v_3\}) = 2$
$f(\{v_3, v_4\}) = 10$
$f(\{v_2, v_4\}) = 4$
$f(\{v_4, v_1\}) = 7$
Then by definition 1.1.12, the graph below is a weighted graph.



Also according to definition 1.1.10, the above graph is a subgraph of itself. Therefore it's weight can be calculated, were by definition 1.1.13, the weight of G is 32.

According to Guichard [10], trees are another useful class of graphs.

Definition 1.1.15. *A tree is a connected graph with no cycles.* [10]

Having defined the basic graph theoretic concepts, it is now time to define harder concepts that use previous definitions. It is important to note that the following concepts can be applied to both weighted and unweighted graphs. Therefore, in the remaining definitions the graph being considered can either be weighted or unweighted.

Definition 1.1.16. *$H(V',E')$ is a spanning subgraph of $G(V,E)$ if H is a subgraph of G and $V' = V$.* [11]

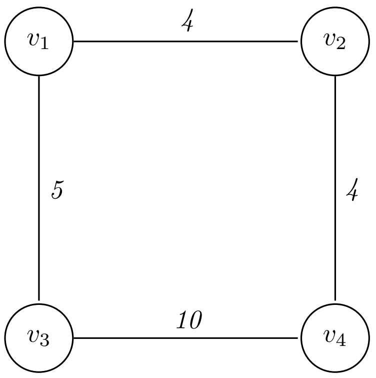
There are many spanning subgraphs, however the ones that are relevant to this thesis are spanning trees and spanning cycles, the latter mostly known as Hamiltonian cycles.

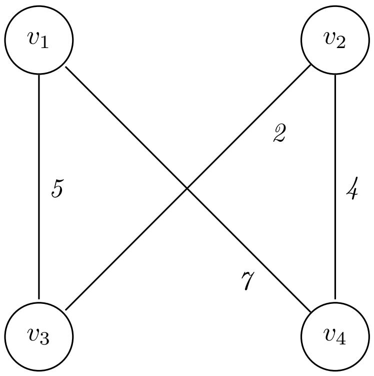
Definition 1.1.17. *A graph H is a spanning tree of G if H is a tree and H is a spanning subgraph of G.* [11]

Definition 1.1.18. *Given a graph G, c is a Hamiltonian cycle in G if c is a cycle and c is a spanning subgraph of G. Also, a graph that contains a Hamiltonian cycle is called a Hamiltonian graph.* [12]

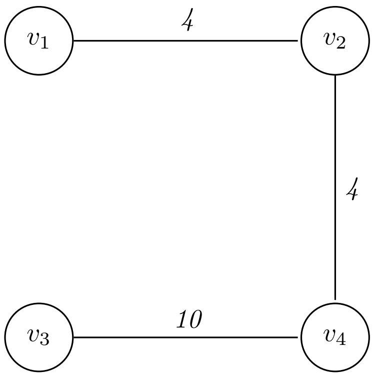
It is worth mentioning that definition 1.1.18 holds because, cycles can be represented by Cycle graphs due to the construction discussed earlier. What follows now is an example that illustrates better definitions 1.1.16, 1.1.17 and 1.1.18.

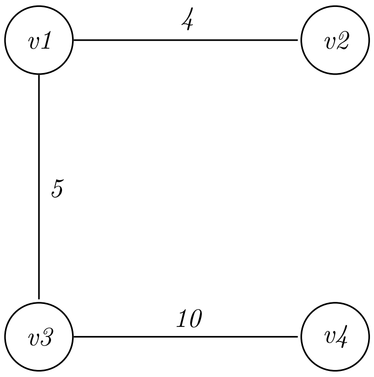
Example 1.1.19. Let G be the graph in example 1.1.14. Then, according to definition 1.1.16, the 2 graphs below are 2 spanning subgraphs of G because, they contain all the vertices of G and are subgraphs of G .





It must also be said that by definition 1.1.18, the above 2 graphs are Hamiltonian cycles in G because, they are spanning subgraphs of G and are Cycle sub-graphs of G. Since the above graphs are subgraphs of G, by definition 1.1.13, their weight can be calculated by summing up the weights of the edges. Thus, the Hamiltonian cycles above have weight 23 and 18 respectively.
Given the same graph G in example 1.1.14, the 2 graphs below are spanning trees of G of weight 18 and 19 respectively.





This example also shows that within the same weighted graph, there could be multiple Hamiltonian cycles and spanning trees of different weight.

Having defined Hamiltonian cycles and spanning trees, it is natural to ask whether there are necessary and sufficient conditions in a graph that guarantee it is Hamiltonian or that it contains a spanning tree as subgraph. In fact, theorem 1.1.20 gives a necessary and sufficient condition for a graph to have a spanning tree.

Theorem 1.1.20. *A graph G has a spanning tree $\iff$ it is connected.*

Proof. ($\implies$) Let $G(V,E)$ be a graph having a spanning tree $T(V',E')$ as one of it's subgraphs. Let $v1, v2 \in V$. Since, T is a spanning tree of G, then, T is a spanning subgraph of G. Thus, $v1, v2 \in V'$. Also, since T is a tree, T must be connected. Therefore, $\exists$ a path P joining vertices v_1 and v_2 in T. But since T is a subgraph of G, then P is also a path in G. Therefore G must be connected.
($\Leftarrow$) Conversely, let $G(V,E)$ be a connected graph. Then, if G has no cycles, G itself must be a spanning tree. If G has cycles, delete an edge from a cycle in G. Clearly, the resultant graph is still connected and contains one less cycle. Repeat this procedure untill no more cycles are left in the graph. Then, the resultant graph G' would be a connected subgraph of G having no cycles(i.e a tree). Also, since by the deletion procedure, no vertex was deleted from G, G' is a spanning subgraph of G. Therefore G' is a spanning tree of G. [11] □

Theorem 1.1.20 confirms that for a graph to have a spanning tree, the graph must be connected and vice-versa. Thus, for spanning trees, the necessary and sufficient condition is connectivity. However, the same cannot be said about Hamiltonian cycles because, no necessary and sufficient conditions are known for a graph to be Hamiltonian. In fact, there are sufficient conditions for a graph to be Hamiltonian, however, these conditions are not necessary. According to Guichard [10], these sufficient conditions typically say that for a graph to be Hamiltonian it must have a lot of edges. But it is also argued in [10], that these conditions are not necessary, because, there are Hamiltonian graphs that have few edges. For example, C_n has only $n-1$ edges but is Hamiltonian. One such sufficient but not necessary condition for Hamiltonianicity is Ore's Theorem below.

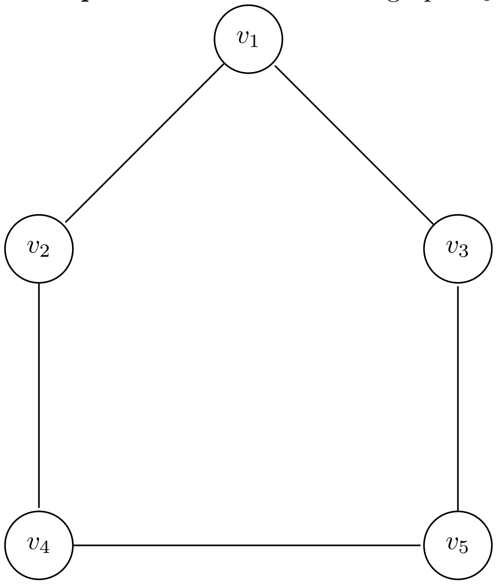
Theorem 1.1.21 (Ore's Theorem)**.** *Let G be a graph on $n \geq 3$ vertices such that if v and w are not adjacent in $G \implies deg(v) + deg(w) \geq n$, then G is Hamiltonian.*

Proof. Suppose that $G(V,E)$ is a graph satisfying all the conditions in the theorem statement but is not Hamiltonian. Then since G is not Hamiltonian and K_n is Hamiltonian, G must be a subgraph of K_n having fewer edges than K_n. Therefore, add edges to G between non adjacent vertices to obtain a subgraph $H(V',E')$ of K_n such that adding an edge to H would create a subgraph of K_n which is Hamiltonian. Let $u, v \in V'$ be 2 non-adjecent vertices in H. Since by construction G is a subgraph of H, u, v must be non-adjcent in G. Therefore $deg(u) + deg(v) \geq n$ in both G and H. Since adding an edge to H creates a resultant graph that is Hamiltonian, then, adding an edge between u and v creates a Hamiltonian graph. Therefore, in H there must be a path joining u

and v containing all the vertices of H. Let the path be $u = v_1, v_2, ..., v_n = v$. Now suppose $deg(v_1) = \alpha$ in H. Now $\forall i, 1 < i < n$, if there is an edge between u_1 and u_i in H, then there must not be an edge between u_{i-1} and u_n because, $u_1, u_i, u_{i+1}, ..., u_n, u_{i-1}, u_{i-2}, ..., u_1$ would be a Hamiltonian cycle in H, thus H would be Hamiltonian. Therefore, $deg(u_n) \leq n - 1 - \alpha$
$\implies deg(u_1) + deg(u_n) \leq \alpha + n - 1 - \alpha$ in H
$\implies deg(u_1) + deg(u_n) \leq n - 1$ in H
$\implies deg(u_1) + deg(u_n) < n$ in G since G is a subgraph of H.
This contradicts the assumption that $deg(u_1 = u) + deg(u_n = v) \geq n$ in G
Therefore, G must be Hamiltonian. [11] □

It is important to note that the above proof uses the fact that K_n is Hamiltonian. This is true because any C_n is always a subgraph of K_n. Therefore any C_n that spans K_n is a subgraph of K_n. Example 1.1.22 below is a counter example to show why Ore's theorem gives a sufficient but not necessary condition.

Example 1.1.22. Consider the graph C_5 *below.*



C_5 above is Hamiltonian because it contains the Hamiltonian cycle $v_1, v_2, v_4, v_5, v_3, v_1$. However, $deg(v_1) + deg(v_5) = 4 < 5 = n$. Therefore, the condition in Ore's Theorem is not a necessary condition.

To conclude, these facts seem to indicate that determining whether a graph is Hamiltonian is a very difficult problem. In addition to that this means that there are certain problems that are harder than other problems. In order to reason about such problems, some computational theory must first be established. This is done in the next subsection.

1.2 Some Computational Theory

After defining some important graph theoretic concepts, it is now time to discuss some important computational theory that is relevant to this thesis. The discussion will now proceed by defining different types of problems and describe the relationship between them.

Definition 1.2.1. *An optimisation problem is a problem were the goal is to find the best solution out of all possible solutions.* [13]

Definition 1.2.1 indicates that when solving an optimisation problem, each solution has an associated value. The aim of the algorithm solving this optimisation problem would then be to find the solution with the best (min/max) value. Another type of problems are decision problems.

Definition 1.2.2. *Decision problems are problems were the solution is either a yes or a no.* [14]

In [14] it is argued that, given an optimisation problem, one can transform this optimisation problem into a decision problem such that, the decision problem is easier to reason about than the optimisation problem. Therefore, if some optimisation problem is easy to solve, the decision problem version is easy to solve aswell [14]. On the other hand, if the decision problem is hard to solve then this would mean that the original optimization problem is also hard to solve [14]. Before going into harder computational theory, an example is given to undestand how an optimisation problem can be transformed into a decision problem.

Example 1.2.3. Two optimisation problems are:
1. The shortest path problem
2. The minimum spanning tree problem
Given a connected weighted graph G, the shortest path problem is the task of finding a path P between 2 vertices in G such that, the weight of P is the minimum amongst all paths joining these 2 vertices in G [15]. On the other hand, given a connected weighted graph G the minimum spanning tree problem is the task of finding a spanning tree in G of minimum weight [16].
The decision problems related to these optimisation problems are the following:
1. Given a connected weighted graph *G,* 2 vertices u and *v,* and an integer *k,* does a path exist from u to v in G of weight at most *k?*
2. Given a connected weighted graph *G,* and an integer k , does G contain a spanning tree of weight at most *k?*
Therefore, to convert an optimisation problem into a decision problem, a bound must be specified on the value to be optimized.

2 The Travelling Salesman Problem

2.1 Heuristics

2.2 The Ant Colony Algorithm

3 Experimental Data

4 Conclusion

References

[1] P. E. Black and P. J. Tanenbaum, "Dictionary of algorithms and data structures," Aug 2017. [Online]. Available: https://xlinux.nist.gov/dads/HTML/graph.html

[2] E. W. Weisstein, "Mathworld–a wolfram web resource," Nov 2018. [Online]. Available: http://mathworld.wolfram.com/AdjacentVertices.html

[3] ——, "Vertex degree." [Online]. Available: http://mathworld.wolfram.com/VertexDegree.html

[4] "Mathonline." [Online]. Available: http://mathonline.wikidot.com/complete-graphs

[5] C. Thompson. [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume18/thompson03a-html/node6.html

[6] E. W. Weisstein, "Connected graph." [Online]. Available: http://mathworld.wolfram.com/ConnectedGraph.html

[7] "Mathematics — walks, trails, paths, cycles and circuits in graph," Jul 2018. [Online]. Available: https://www.geeksforgeeks.org/mathematics-walks-trails-paths-cycles-and-circuits-in-graph/

[8] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff, *COMBINATORICS AND GRAPH THEORY*, 2nd ed. SPRINGER, 2008.

[9] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*, 5th ed. Elsevier Science Publishing, 1982.

[10] D. Guichard, "An introduction to combinatorics and graph theory," Jul 2018.

[11] S. S. Ray, *Graph theory with algorithms and its applications: in applied science and technology.* Springer, 2013.

[12] E. Weisstein, "Hamiltonian cycle," Oct 2018. [Online]. Available: http://mathworld.wolfram.com/HamiltonianCycle.html

[13] P. E. Black, "optimization problem," Apr 2009. [Online]. Available: https://xlinux.nist.gov/dads/HTML/optimization.html

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 3rd ed. MIT PRESS.

[15] "Shortest path algorithms." [Online]. Available: https://www.hackerearth.com/practice/algorithms/graphs/shortest-path-algorithms/tutorial/

[16] "Applications of minimum spanning tree problem." [Online]. Available: https://www.geeksforgeeks.org/applications-of-minimum-spanning-tree/