CHAPTER 2
THEORY BACKGROUND

When solving problems with computers it quite quickly becomes obvious that some problems are fundamentally harder to solve than others. While for some problems it is possible to design ingenious algorithms that solve the problem efficiently, for others it seems considerably harder, or even impossible to come up with any algorithms at all [3].

The theory of computational complexity makes it possible to formalize the concepts of “easy” and “hard” problems and the distinction between them. Problems can be formally classified based on their complexity, and if a problem belongs to the class of NP-hard or complete problems, we know in advance that there is little hope of finding an efficient and exact algorithm for solving it. Any exact algorithm for such a problem has an execution time exploding for increasing problem sizes, and is often useless for most practical purposes [3].

The search for alternative algorithms is thus justified, there is a demand for faster algorithms that do not necessarily produce the exact optimal solution, but in most cases provide solutions of sufficient quality. Such methods are called heuristic algorithms or approximation algorithms. One member of the NP-complete class and possibly the most well– known is the Traveling Salesman Problem (TSP). The Traveling Salesman Problem has commanded much attention of mathematicians and computer scientists specifically because it is so easy to describe and so difficult to solve [3].

Problem can be stated as: given a finite number of “cities" along with the cost of travel between each pair of them, find the cheapest way of visiting all of the cities and returning to your starting point. The travel costs are symmetric in the sense that traveling from city X to city Y costs just as much as traveling from Y to X [3].

In terms of input, the problem takes a list of physical locations or system nodes, along with distance information. Algorithms and equations work on the process of identifying the most efficient paths possible between the locations. Computer programs can do this through the process of elimination or through a