CHAPTER 1

INTRODUCTION

Metaheuristic methods are known to give good results when solving combinatorial problems and research on these are growing rapidly. Basically metaheuristic is a local search algorithm that has been extended or broadens so that high quality solutions can be found in a lower possible running time. A straightforward extension is to run a simple local search a number of times by using a different start with the best solution kept. This is also known as a *multi-start* approach. Another different approach applies a multiple run of a local search algorithm by combining several neighbourhoods. This is also known as *multi-level*, for example the *iterated local search* (Aarts and Lenstra, 1997).

Examples of combinatorial problems are the Traveling Salesman Problem (TSP), Uniform Graph Partitioning Problem, Job Scheduling Problem, Vehicle Routing Problem (VRP) and Inventory Routing Problem (IRP). Even though some metaheuristics are known to be very powerful in solving combinatorial problems, the efficiency of the method is also related to how hard the problem is. This leads to the theory of NP-hard and NP-complete. Classes P, NP-hard and NP-complete are introduced.

A combinatorial problem is defined. Several definitions of combinatorial problem are defined in Aarts and Lenstra (1997) as follows:

Definition 1.1: *A combinatorial optimization problem is specified by a set of problem instances and is either a minimization problem or a maximization problem.*

Definition 1.2: *An instance of a combinatorial optimization problem is a pair of* $(\mathcal{S}, f)$*where the solution set* $\mathcal{S}$ *is the set of feasible solutions and the cost function* f *is a mapping* $f: \mathcal{S} \to \mathbb{R}$*. The problem is to find a globally optimal solution. i.e., an* $i^* \in \mathcal{S}$ *such that* $f(i^*) \leq f(i)$ *for all* $i \in \mathcal{S}$*. Furthermore,* $f^* = f(i^*)$ *denotes the optimal cost, and* $\mathcal{S}^* = \{i \in \mathcal{S} \mid f(i) = f^*\}$ *denotes the set of optimal solutions.*

The instance $(\mathcal{S}, f)$ from Definition 1.2 is generally not given explicitly; for example via a listing of all solutions and their costs. Often, there are different data representations of the instances. The solution set is usually represented by *decision variables* with a certain range.

Measuring the difficulty of the problem is related to the order of complexity. The time complexity of an algorithm is the order of growth of an algorithm. A more formal definition of a time complexity is given in Definition 1.3.

Definition 1.3 (Korte and Vygen, 2008)**:** *Let A be an algorithm which can accept input from a set X, and let* $f: N \to \mathbb{R}_+$*. If there exists a constant* $\alpha > 0$ *such that A terminates its computation after at most* $\alpha f\big(size(x)\big)$ *elementary steps (including arithmetic operations) for each input* $x \in X$*, then it is said that A* ***runs in*** *O(f)* ***time.*** *It can also be said that the* ***running time*** *(or* ***time complexity****) of A is O(f).*

As mentioned earlier, most of the combinatorial problems are known to be NP-hard problems. Combinatorial problems will first be transformed into decision problems. A decision problem means that the answer or output of the problem is simply given by *zero "*0" or *one "*1" bit (in which 1 represents "yes" and 0 represents "no").

Definition 1.4: *A* ***decision problem*** *is a pair P=(X,Y), where X is a language decidable in polynomial time and* $Y \subseteq X$*. The elements of X are called* ***instances*** *of P; elements of*

*Y are **yes-instances**, those of X\Y are **no-instances**. An algorithm for a decision problem (X,Y) is an algorithm computing the function* $f: X \rightarrow \{0,1\}$*, and defining b* $f(x) = 1y$ *for* $x \in Y$ *and* $f(x) = 0$ *for* $x \in X \backslash Y$.

Class P problems have algorithms that are bounded within polynomial time complexity. P represents the "polynomial time complexity". Problems that can be categorized in class P are those problems that can be solved deterministically using an algorithm, *deterministic Turing machine* with time complexity $O(n^k)$ where k is a constant (Yu and Gen, 2010). The class P is the class of decision problems that are efficiently solvable.

If the problem cannot be solved using polynomial time complexity, the problem belongs to class *NP* where an abbreviation *NP* refers to "nondeterministic polynomial time complexity". It has been proven that $P \subseteq NP$. Understanding *polynomial transformation* will help in understanding what *NP*-hard and *NP*-complete are. If an algorithm with a polynomial time complexity can be designed to transform any solution of the decision problem D_1 of one problem to one solution of the decision problem D_2 of another problem, denoting that $D_1 \propto D_2$, it can be understood that problem D_2 is at least as difficult as problem D_1.

Let's say C is a decision problem. If C belongs to *NP* and there exists polynomial algorithms to transform every problem in *NP* into C, then C is *NP*-complete or C is an *NP*-complete problem. If problem H does not belong to *NP* but there exists an *NP*-complete problem C that can be polynomially transformed into H, then it is said that H belongs to the class of *NP*-hard or H is a *NP*-hard problem, which can be understood that H is at least as hard as problem C. Figure 1.1 illustrates the relationship of P, *NP*, *NP*-complete and *NP*-hard problem. Class P is a subset of *NP*. Intersection between *NP* and *NP*-hard is *NP*-complete.

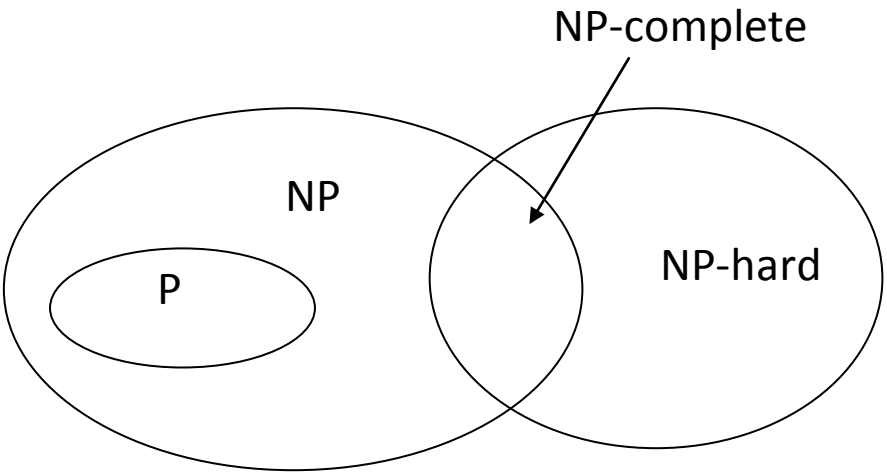

Figure 1.1: The relationship of *P*, *NP*, *NP*-hard and *NP*-complete.

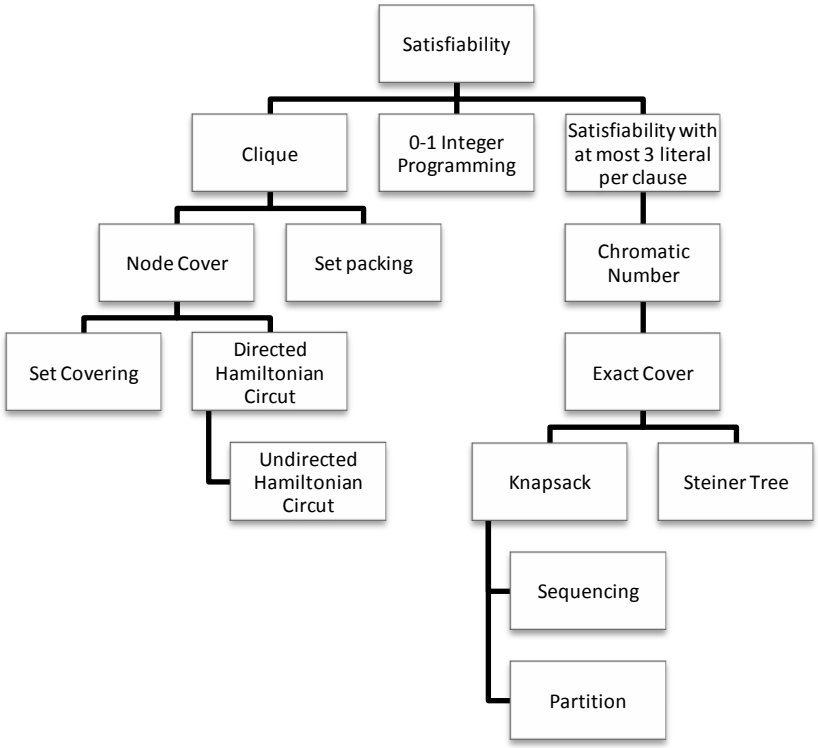

Figure 1.2: Complete problems.

Figure 1.2 gives the list of problems that can be classified as *NP*-complete problems. Note that the problem considered in this thesis are Message Scheduling Problem specifically Point to Multipoint Routing Problems (PMRP) and Inventory

Routing Problems (IRP); where PMRP employs the Steiner Tree Problems and IRP are combinations of Sequencing and Partitioning Problems which are all *NP*-complete problems.

1.1 Network Routing Problem

Network routing problems can be look as a system that route products or items; for example goods, foods, messages and even people from one place to another. Examples of network routing problems among others include distribution network which involves inventory routing problem and vehicle routing problem or transportation problem, telecommunication network (e.g. message routing problem) and scheduling problem. The problems under network routing are conceptually simple, but yet are mathematically complex and challenging. Many problems of network routing can be classified as NP-hard and NP-complete problem as they usually have an underlying combinatorial structure and in addition to an underlying network structure (e.g. communication or transportation network).

Problems considered in this research are Message Scheduling Problem (MSP) specifically Point to Multipoint Routing problem (PMRP) which involves routing a set data or messages from one source nodes to several destination nodes whilst the Inventory Routing Problem (IRP) involves the transportation of goods or products from a set of suppliers to an Assembly Plant.

A. Message Scheduling Problem (MSP)

MSP is defined as process of scheduling a set of requests such that the entire route is optimal. The requests (or messages) are to be sent from one source node to multiple destination nodes for a telecommunications network. The telecommunications network is modeled as a weighted graph with specific cost assigned for each edges. The MSP is to determine an optimal route or routes

such that the total costs involved is minimized. The MSP is explained in details in Chapter 3.

B. Inventory Routing Problem (IRP)

Inventory Routing Problem (IRP) is the coordination of the inventory management and transportation, and seeks to determine simultaneously an optimal inventory and distribution strategy that minimizes the total cost. The resulting inventory and transportation policies usually assign suppliers to routes and then determine the replenishment intervals and collection sizes for each supplier. The distribution network considered includes a depot, suppliers and assembly plant where solving IRP involves finding the minimum total costs of transportation routing (or vehicle routing) and also inventory routing problem. The IRP is discussed in great length in Chapter 4.

1.2 Research Objective and Contribution

The main objective of the research is to evaluate the performance of the Variable Neighbourhood Search (VNS) on different problems: network routing problems, specifically, Point to Multipoint Problem and also Inventory Routing Problems. In addition, the performance of Genetic Algorithm (GA) is also evaluated in the Point to Multipoint Problem. This research concentrates on the metaheuristic methods, VNS and GA. Both methods are known to be very powerful methods as they consist of a mechanism to avoid getting trapped in local optimum. The thesis is presented in five chapters.

In Chapter 2, the metaheuristic method is discussed in detail. Several definitions of the metaheuristic methods are given. The classification of the metaheuristic is presented based on two categories: single solution based and population based. Several

well known metaheuristics such as Simulated Annealing (SA), Tabu Search (TS), evolutionary algorithms (EC) especially GA and VNS are presented according to its classification emphasizing on VNS and GA which are the focus of this study.

The definition of the VNS is given and the underlying concept of VNS is presented. The basic general algorithm is described and a more specific algorithm pertaining to the work in the thesis is given in later chapters (Chapter 3 and Chapter 4). In addition, some variations of VNS, such as Reduced VNS (RVNS) and Variable Neighbourhood Decomposition Search (VNDS) is given and discussed. An explanation of GA, the terminology and how it works are also discussed. The fundamental concepts and the basic operators used in GA are described using the simple GA. Some illustration is used in explaining simple GA which includes three operators that are Selection, Recombination and Mutation. The concept of hybridization of GA by embedding local search algorithms is briefly described. At the end of the chapter, some explanation of the Local Search, focusing on the local search that is applied in this thesis is explained in great length. The Shortest Path Algorithm; Dijkstra and Bellman Ford, and Minimum Spanning Tree heuristics; and Kruskal and Prim Algorithm which are used in the point to multi point routing are discussed in Chapter 3. In the second part of the Local Search algorithms the focus is on the routing based algorithm specifically built for the traveling salesman problem (TSP). Inventory Routing Problem embeds inventory and routing of vehicles which is closely related to TSP. Here, the Local Search is divided into constructive heuristics and improvement heuristics. The well known techniques such as Insertion, Nearest Neighbour, 2-*opt*, 3-*opt* and *k*-*opt* are described in great detail. Savings and Sweep algorithms which are used in the clustering process of the routing of the Vehicle Routing Problem are also given.

In the third and fourth chapters, two cases of combinatorial problems that are studied in this thesis; the network design specifically the Point to Multipoint Routing

Problem and the Inventory Routing Problem are presented. Chapter 3 focuses on the Point to Multipoint Routing Problem (PMRP) which is to determine an optimal route to send a single message from a single source to multiple destinations. Two algorithms are proposed which are VNS and GA. Both algorithms embed the Kou, Markowsky and Berman (KMB) algorithm to solve the Steiner Tree problem which is used to find an optimal routing to send a message. It is noted that the KMB comprises of two related algorithms, Djikstra's algorithm to find the shortest route and Kruskal's algorithm to determine the minimum spanning tree. In addition, VNS also embeds swap, invert, *or-opt* and restricted *or-opt* as part of its local search. Both algorithms are tested on three problems representing small, medium and large test cases.

In Chapter 4 the Integrated Inventory Routing Problem (IRP) which considers inventory control and vehicle routing concurrently in order for the overall cost to be optimized is discussed. VNS is proposed to solve the problem and the neighbourhood structure is defined as a distance function, that is the cardinality of the symmetric difference between any two solutions V_1 and V_2, that is the number of different suppliers that are visited in a period. The algorithm embeds Generalized Insertion (GENI) algorithm in the local search. The results obtained from VNS are compared to the results obtained by Aziz and Moin (2007) that used GA as the main algorithm. There are 14 cases considered: S12T14, S20T21, S50T21 and S98T14 which are reported in the paper, and S12T5, S12T10, S20T5, S20T10, S20T14, S50T5, S50T10, S50T14, S98T5 and S98T10, which are rerun to obtain extended results. The algorithm is modified to Enhanced VNS (EVNS) in which GENI is part of the neighbourhood structure instead of the local search. Nearest neighbor heuristics is employed in the local search. Two different EVNS developed that are EVNS1, in which VNS with 2-*opt* as preoptimization, while EVNS2, which is VNS with 3-*opt*, with the same application as

EVNS1.The results obtained are compared with the Lower Bound and Best Integer found using CPLEX.

In Chapter 5 which is the last part of the thesis, all the work done is concluded and further extensions are outlined.