Advances in Fuzzy Mathematics.
ISSN 0973-533X Volume 10, Number 2 (2015), pp. 99-103

http://www.ripublication.com

A Study on NP-completeness in Fuzzy Context

Mathews M. George

Department of Mathematics, B.A.M College, Thuruthicad, Kerala
e-mail:mathewjas@gmail.com

Abstract

The classical set theory models the world as black or white and makes no provision for sets of grey. Life is not always black and white. This two valued logic has proved effective and successful in solving well defined problems. However, a class of problems exists which are typically complex in nature, and are often left to human beings to deal with. It is known that no polynomial algorithm could be found for the problems in NP or NP-hard problems. While a method for computing the solutions to NP-complete problems using a reasonable amount of time remains undiscovered, computer scientists and programmers still frequently encounter NP-complete problems. NP-complete problems are often addressed by using algorithms. The purpose of this paper is to introduce a basic concept of NP-Completeness in the fuzzy context.

Key words: fuzzy algorithm, the classes fuzzy P and fuzzy NP, fuzzy NP-completeness

1. INTRODUCTION

The origin of the word ‘algorithm’ dates back to Euclid who obtained a step-by-step procedure to find the greatest common divisor of two integers. During the time of Leibnitz (1646 – 1716) and perhaps earlier, there were attempts to provide algorithmic proofs for problems involving numerical computation. David Hilbert’s (1862-1943) aim was to device a formal mathematical system in which all problems can be precisely formulated in terms of statements which are either true or false. If Hilbert’s aim was fulfilled then any problem which was well defined could be solved by executing the algorithm. In 1931 Kurt Gödel discovered that no algorithm of the type desired by Hilbert exists. Gödel’s work is the famous *incompleteness theorem.* Gödel’s incompleteness theorem states that there is no algorithm whose input can be any statement involving integers and whose output tells whether or not the statement is true. Unlike other papers, this article contains no theorems and no proof.

2. PRELIMINARIES

In the computer science field, complexity theory attempts to answer questions about the amount of use of computational resources *time, memory and hardware*. The complexity of a problem is the complexity of the best algorithm which solves that problem. The amount of any resource used by an algorithm may vary with size of the input data. We are interested only in finding the most 'efficient' algorithm for solving a problem. The time requirements of an algorithm are conveniently expressed in terms of a single variable, the size of a problem. That is, the amount of input data needed to describe the problem.

Definition 2.1 A function f(n) is said to be O(g(n)) if there exists a constant c such that $|f(n)| \leq c|g(n)|$ for all $n \geq n_0$.

Definition 2.2 The *time complexity function* of an algorithm expresses its time requirements by giving, for each possible input length, the largest amount of time needed by the algorithm to solve a problem.

Definition 2.3 A *polynomial time algorithm* is defined to be one whose time complexity function is O(p(n)) for some polynomial function p, where n is the input length. An algorithm whose time complexity function cannot be so bounded is called an *exponential time algorithm*. A problem has not been 'well solved' until a polynomial time algorithm is known for it. A problem is so hard that no polynomial time algorithm can possibly solve it.

Definition 2.4 Any problem for which the answer is either zero or one is called a *decision problem*. An algorithm for a decision problem is termed a *decision algorithm*. Any problem that involves the identification of an optimal solution (either maximum or minimum) value of a given cost function is known as an *optimization problem*.

Definition 2.5 Algorithms such that the result of every operation is uniquely defined are called *deterministic algorithms*. A machine capable of executing a deterministic algorithm is called a deterministic machine.

Definition 2.6 Algorithms such that the result of every operation is not uniquely defined are called *non deterministic algorithms*. A machine capable of executing a nondeterministic algorithm is called a nondeterministic machine.

Definition 2.7 A decision problem is in P if there is a known polynomial-time algorithm to get that answer. A decision problem is in NP if there is a known polynomial-time algorithm for a non-deterministic machine to get the answer.
NP-complete is a subset of NP, the set of all decision problems whose solutions can be verified in polynomial time; *NP* may be equivalently defined as the set of decision problems that can be solved in polynomial time on a nondeterministic Turing machine. A problem *p* in NP is also in NPC if and only if every other problem in NP

can be transformed into *p* in polynomial time. NP-complete can also be used as an adjective: problems in the class NP-complete are known as NP-complete problems.

Definition 2.8 A decision problem L is NP-complete if:
1. L ∈NP, and
2. Every problem in NP is reducible to L in polynomial time.

A consequence of this definition is that if we had a polynomial time algorithm for L, we could solve all problems in NP in polynomial time. Some NP-complete problems, indicating the reductions typically used to prove their NP-completeness. Thus, the easiest way to prove that some new problem is NP-complete is first to prove that it is in NP, and then to reduce some known NP-complete problem to it. Therefore, it is useful to know a variety of NP-complete problems.

Definition 2.9 The concept of fuzzy algorithm may be viewed as a generalization, through the process of fuzzification, of the conventional (non fuzzy) concept of an algorithm. A fuzzy algorithm may contain fuzzy statements containing names of fuzzy sets.

Illustration 2.10 Fuzzy algorithm may contain fuzzy instructions such as:
(a) "Set y approximately equal to 8 if x is approximately equal to 4"
(b) "If x is large, increase y by several units"

The sources of fuzziness in these instructions are fuzzy sets which are identified by their underlined names. Fuzzy instruction which is a part of a fuzzy algorithm can be assigned a precise meaning by making use of the concept of the membership function of a fuzzy set.

Definition 2.10 Fuzzy algorithms such that the result of every operation is uniquely defined are called deterministic fuzzy algorithms. A machine capable of executing a deterministic fuzzy algorithm is called a *deterministic fuzzy machine*.

Definition 2.11 Fuzzy algorithms such that the result of every operation is uniquely defined are called deterministic fuzzy algorithms. A machine capable of executing a nondeterministic fuzzy algorithm in this way is called a *nondeterministic fuzzy machine*.

3. FUZZY NP-COMPLETENESS

In this section, we introduce the classes *fuzzy P* and *fuzzy NP* using fuzzy algorithm.

Definition 3.1 NP-complete is a subset of NP, the set of all decision problems whose solutions can be verified in polynomial time; *NP* may be equivalently defined as the set of decision problems that can be solved in polynomial time on a nondeterministic fuzzy Turing machine.

Definition 3.2 The class *fuzzy P* is defined to be the class of all decision problems that can be solved in polynomial time by a non-deterministic fuzzy algorithm. The class *fuzzy NP* is defined to be the class of all decision problems that can be solved in polynomial time by a non-deterministic fuzzy algorithm.

Definition 3.3 A decision problem L is *fuzzy NP-complete* if:

1. L ∈ *fuzzy NP*, and
2. Every problem in fuzzy NP is reducible to L in polynomial time.

Note that a problem satisfying condition 2 is said to be *fuzzy NP-hard*, whether or not it satisfies condition 1. There is often only a small difference between a problem in *fuzzy P* and a *fuzzy NP-complete* problem.

Limitations of the study and Scope for Future Studies

In this study mainly utilized descriptive research method to address the study objectives. Several limitations of the study provide additional research opportunities. One of the major limitations is that the study is theoretical in nature and is based on the documents and other information from journals and articles. Also this study is mainly focused on complexity, only a few areas were considered for evaluation of the results. There is more scope for future research in the above mentioned areas. NP-complete problems are interesting for future studies. Why?

1. It is unknown whether or not a polynomial time algorithm exists for NP-complete problems.
2. If a polynomial time algorithm exists for any NP-complete problem then polynomial time algorithms exists for all of them.
3. Several NP-complete problems are similar, but not identical, no problems for which we do know of polynomial time algorithms.

Further studies can focus on the following:

Is the class of problems NP-Complete more difficult to solve than the class of problems NP-Hard in fuzzy context? In other words, NP-Complete problems are in NP, and also are NP-Hard, but not all NP-Hard problems are in NP... How to tell which one is more difficult to solve in fuzzy environment? Further research is needed to explore and deepen the understanding of the nature of complexity in fuzzy context. This study opens new avenues for further investigation for this new concept fuzzy NP-completeness.

References

1. Cook, S.A. (1971). "The complexity of theorem proving procedures". *Proceedings, Third Annual ACM Symposium on the Theory of Computing, ACM, New York*. pp. 151–158.
2. Garey, M.R.; Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman. ISBN 0-7167-

1045-5.This book is a classic, developing the theory, and then cataloguing *many* NP-Complete problems.

3. Mathews M. George, *Stephan Cook's theorem and NP-Completeness,* M. Phil dissertation, Calicut University, Kerala India 1992
4. Michael E. Albertson, Joan P. Hutchinson, *Discrete Mathematics with Algorithms*, John Wiley & Sons, 2003.
5. L.A. Zadeh, *Fuzzy sets, Information and Control* 8 (1965), 338-352
6. L.A. Zadeh, *Fuzzzy Algorithms*, Information and Control, 12, 91-102 (1968)
7. H.Z.Zimmermann, *Fuzzy Set Theory and Applications*, Allied Publishers Limited, New Delhi 1996