ABSTRACT

Although modern computer is the most revolutionary and most powerful tool developed in the twenty's century, it is almost useless for the application of this tool to the not well defined humanistic systems such as politics, law, or even the many hour-to-hour small decisions people make routinely and daily. This is in spite of the fact that the human action of the cognitive band, which is of the order of seconds, is much slower than the speed of the modern computer. In this paper, the author shall first examine the basic differences between the scientific systems and the humanistic systems. Then based on these resulting differences, the author shall propose a neural-soft-computing combined approach, which is naturally suited for the vague and difficult to define humanistic systems. These combined systems are developed during the last approximately twenty years. Yet, it has not applied to the humanistic systems in a systematic and extensive manner. Some of the neural-soft-computing systems, also known as neural-evolutionary operational systems are the combined use of neural network, or support vector machine, with fuzzy system or fuzzy logic and evolutionary operational techniques such as genetic algorithm. Several of these systems are summarized and discussed as to why these systems seem more promising for the handling of humanistic systems. To illustrate the effectiveness of the proposed approaches, several initial applications in the literature are summarized.

1. SCIENTIFIC OR HARD COMPUTING VERSUS HUMANISTIC OR SOFT COMPUTING

Humanistic systems can be defined as systems whose behavior strongly influenced by human judgments, perceptions, or emotions (Zadel, 1975). Some of the examples of humanistic systems are: societal systems, political systems, law, psychological systems, some of the economic systems and biological systems.

In order to develop approaches to model humanistic systems on modern computer, the basic differences between scientific systems, which require hard or precise computation, and humanistic systems, which require approximate computation, should be examined.
The most important difference between the two systems is philosophic in nature. In scientific computing, the results must be represented in exact numbers. On the other hand, approximation in humanistic computing is not only necessary but is also desirable. A simple example can illustrate this point. We park our car daily in the parking lot before go to work and never give much thought. Suppose it is required to park the car in a precise location, where precise location means one thousandth of an inch. We would think this requirement is crazy and certainly not necessary. Furthermore, it would be too time consuming and impossible to carry out.

In order to save time or for simplicity, we frequently omit the details even though these details are known. One example is that we frequently say, “John is young” even though we know John is exactly 20 years old. This is because the occasion does not need details.

Because of this desired approximation, numerical representation is not appropriate and the most appropriate approach is to use the original language phrase or term such as young. Thus, the problem is how to model and to manipulate a system which is represented linguistically.

Another difference is the problem of subjectivity. In scientific modeling or computing, the results must be objective. Subjectivity is one of the biggest sins in scientific research. On the other hand, it is almost impossible to be completely objective in the modeling of humanistic activities. For example, we say “John is young”. What do we mean by young? Under some circumstances or by certain people’s judgment, thirty may be considered young. But, a group of very young people such as a group of kindergarteners may consider thirty is old.

A third difference is “domain dependency”, or not universally true in humanistic systems. Scientific truth is universal. Scientific truth remains the same no matter where or when it is applied and is independent of the environment. However, human activities depend on the environment. For example, a shoe company sell shoes in China would require making smaller size shoes than if the shoes were made for sale in Norway. The example “John is young” also can be applied here. Whether “young” or not depends on the concept of the people involved.

A fourth and much more difficult to handle difference is the vagueness and the lack of complete knowledge or definition of the system. Vagueness can be both in value and in other aspects. Vagueness in value is concerned with precision, which was already discussed previously. Usually, in humanistic systems, both the variables and the definition of the problem are vague. For example, in American presidential election, the linguistic variable “national security” is a predominant factor in the 2004 election, but, it is not as important during the 2000 election. Thus, it is even difficult to define which variable is more important and thus the boundary or the largeness of the problem cannot be defined easily.

In summary, the above differences cause the humanistic problems to be vague, lack of complete knowledge, linguistic in nature, and subjective. As discussed above, vagueness is due to many different aspects, some of them are: the difficult in defining precision, in deciding which variable is important, and how to define the variables or even how to define the problem.

2. SOFT COMPUTING ON MODERN COMPUTER

The modeling and computation on modern computer involve two basic steps, namely, representation and aggregation or manipulation. We perform aggregation for scientific or hard computing routinely on modern computer by the use of the various numerical aggregation approaches such as plus, minus, differentiation, integration, and even optimization. However, due to the fact that the knowledge is represented in linguistic forms and not numerical for humanistic systems, the aggregation approaches used easily in hard computing cannot be used for soft computing.

Consider “John is young”, as was pointed out previously, the best way is to use the
Non Linear Dynamical Systems and Chaos Synchronization
www.igi-global.com/article/non-linear-dynamical-systems-chaos/44670?camid=4v1a