کارگاه محاسبات نرم

On Theoretical Foundations of Soft Computing

Amir Daneshgar, Rasool Ramezanian
Department of Mathematical Sciences, Sharif University of Technology

Abstract

In this talk we try to present a brief introduction to the theoretical foundations of soft computing. In this regard we begin with a concise introduction to *theory of computation* along with some historical notes, and we present a general definition of a *computation* that consists of *given data (input), dynamics (model)* and the *result (output).* In this sense the *classical* (i.e. hard) computing is usually identified with the model of standard Turing machines and is justified through Church-Turing thesis. Therefore, the term *soft computing* is usually assigned to the cases when any one of the above mentioned three concepts are allowed to be given in a *relaxed* sort of way.

From this point of view the subjects of *approximation algorithms* and *randomized algorithms* fall into the class of soft computation since in these models of computation the output is allowed to be nonexact.

In this sense, we start with the case of soft *data* and will try to present a survey of mathematical tools usually used, containing theories of *fuzzy sets* and *vague sets*. Also, in this regard we try to present a short introduction to the theory of Dioids and Max-Plus algebras as some related basic tools and active areas of research.

In the second part we focus on relaxations of the dynamics and consider the concept of computation as a discrete dynamical system. In this regard, first, we try to present basic well-known concepts of soft computing as *soft optimization, soft memories (classifiers), learning algorithms, games* and *interactive computation* in the framework of discrete dynamics and then as the basic mathematical tools to model and analyze these systems, we briefly introduce the theories of *modal logic* and *combinatorial landscapes*.

At the end of this talk we try to present a prelude to the theory of Max-Plus-based nonlinear dynamics and its relations to *dynamic programming.*