

Abstract

Constraint programming is a paradigm wherein relations between variables are stated in the form of constraints. It is well-known that many real-life problems can be modeled as Constraint Satisfaction Problems (CSPs). Due to their pervasive use in solving these problems, much effort has been spent to increase the efficiency of algorithms for solving CSPs. However, many of these techniques assume that the set of variables, domains and constraints involved in the CSP are completely known and fixed when the problem is modeled. This is a strong limitation because many problems come from uncertain and dynamic environments, where the original problem, and consequently its associated CSP model, may evolve due to the environment, the user or other agents. In such situations, a solution that holds for the original problem can become invalid after changes occur in the original problem.

There exist two main approaches for dealing with these situations: reactive and proactive. Using reactive approaches entails re-solving the CSP after a solution is no longer a solution, which is time consuming. That is a clear disadvantage, especially when we deal with short-term changes. This is also motivated in [Verfaillie and Jussien, 2005], which is an important survey on constraint solving in uncertain and dynamic environments. In this survey, the authors state that a desirable objective in dynamic and uncertain frameworks is:

First *“Limit as much as possible the need for successive online problem solvings.”*

Motivated by the **first** statement, in this dissertation we develop proactive approaches, which try to offer a resistance to the possible future alterations of the problem. Therefore, these approaches are applied before changes in the original problem occur. There exist two main types of proactive approaches, which can be distinguished on the basis of the characteristics of the solutions that they obtain: *robust* and *flexible*. The flexibility concept implies modifications over the original solution whilst the robustness concept does not. In [Verfaillie and Jussien, 2005], the authors make the **second** statement as another desirable objective.

Second *“Limit as much as possible changes in the produced solution.”*

For this reason, this thesis mainly focuses on the search of robust solutions, which have a high probability of remaining solutions after changes over the CSP. Furthermore, in [Verfaillie and Jussien, 2005] the authors also mention as future interesting work the possibility of developing proactive strategies that combine the solution features of robustness and flexibility.

Third *“The production of solutions that are at the same time robust and flexible, that have every chance to resist changes and can be easily adapted when they did not resist, is obviously a desirable objective.”*

According to this objective, expressed in the **third** statement, in this thesis we develop approaches that meet the condition of combining solution robustness and *stability* (the stability is a special case of flexibility). To the best of our knowledge, this combination has not yet been developed for CSPs.

Proactive approaches use the available knowledge about possible future changes in order to avoid or minimize their effects. However, for many real-life problems the information about the uncertain and dynamic environment is unknown or hard to obtain. In many proactive approaches that search for robust solutions, the form of the algorithm is dependent on detailed knowledge about the dynamic environment. As a result, in the lack of such a priori information, these methods can not be applied. For this reason, the author of [Hebrard, 2006] makes the **fourth** statement for a desirable characteristic of approaches dealing with such uncertain and dynamic framework.

Fourth *“Ideally, no additional knowledge over the data used to build the classical constraint network is required and no more expertise than for solving the problem without taking uncertainty into account.”*

In this dissertation, we try to answer an interesting question in dynamic and uncertain environments: when additional information about the possible changes in the problem is unknown, is it possible to define what is the robustness of a solution of a CSP and to build appropriate algorithms? We found that it is possible and justifiable to extract some limited (and intuitively reasonable) assumptions about the dynamism of problems for which the order over the domain elements is significant. Therefore, in this work we present approaches for dealing with types of problems that are, therefore, consistent with the **fourth** statement. The fulfilment of these four statements has been the motivation and the main objective of the work presented in this thesis.