An Intelligent System to Manage Soft Constraints

Gabriel Higareda
Department of Computer Science, University of Texas at El Paso, ghigareda@miners.utep.edu

Eduardo Breijo
Department of Computer Science, University of Texas at El Paso, eduardo.breijo@upr.edu

Martine Ceberio
Department of Computer Science, University of Texas at El Paso, mceberio@utep.edu

Follow this and additional works at: http://digitalcommons.utep.edu/couri_abstracts_sum12

Recommended Citation
http://digitalcommons.utep.edu/couri_abstracts_sum12/64

This Article is brought to you for free and open access by the COURI Symposium Abstracts at DigitalCommons@UTEP. It has been accepted for inclusion in COURI Symposium Abstracts, Summer 2012 by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.
An Intelligent System to Manage Soft Constraints

Gabriel Higareda, Eduardo Breijo, and Martine Ceberio

Most problems can be expressed in terms of requirements that must be met by their expected solutions. Constraints are requirements and constraint programming is a declarative programming paradigm that simply consists in modeling problems as lists of their requirements. Many constraint solvers exist that provide solutions to such list of requirements.

However, constraint solvers are stiff: if a problem was ill modeled, no solution will be returned and the original problem called over-constrained. Nevertheless there is hope even in these cases: such overconstrained problems can be handled as soft constraints, meaning that their processing will not be stiff anymore and solvers will be lenient on constraint satisfaction, sometimes stretching constraints, sometimes discarding them altogether. There exist plenty of frameworks for addressing soft constraints and deciding what to do with them.

Unfortunately, having to deal with an overconstrained problem and facing many frameworks as solving options, one might wonder which option makes most sense. The aim of the presented work is to design a decision-aiding tool that analyzes the reasons why a problem is overconstrained and turns to the user to elicit preferences on constraints based on the results of the analysis. Specifically, we propose to analyze how constraints are clustered in groups of compatible constraints as a mean to generate high-level questions to users and provide guidance for the choice of a framework. We report on preliminary experiments and show how our findings can be extended to a more complex analysis.

This article is available at DigitalCommons@UTEP: http://digitalcommons.utep.edu/couri_abstracts_sum12/64