Measures of Intrinsic Hardness for Constraint Satisfaction Problem Instances

George Boukeas, Constantinos Halatsis,
Vassilis Zissimopoulos, and Panagiotis Stamatopoulos

Department of Informatics and Telecommunications, University of Athens

Abstract. Our aim is to investigate the factors which determine the intrinsic hardness of constructing a solution to any particular constraint satisfaction problem instance, regardless of the algorithm employed. The line of reasoning is roughly the following: There exists a set of distinct, possibly overlapping, trajectories through the states of the search space, which start at the unique initial state and terminate at complete feasible assignments. These trajectories are named solution paths. The entropy of the distribution of solution paths among the states of each level of the search space provides a measure of the amount of choice available for selecting a solution path at that level. This measure of choice is named solution path diversity. Intrinsic instance hardness is identified with the deficit in solution path diversity and is shown to be linked to the distribution of instance solutions as well as constrainedness, an established hardness measure.

1 Introduction

A constraint satisfaction problem consists of a set of variables and a set of constraints. A variable which has been given a value is said to be *instantiated*. A set of instantiations to i distinct variables is an *assignment* α_i of size i. If the size of an assignment α_n equals the number of problem variables n then it is a *complete assignment*. A complete assignment which satisfies all problem constraints is a feasible *solution*. The set of solutions to a problem instance is denoted by $\mathcal{S}$. Given a constraint satisfaction problem instance, the goal is to find a feasible solution or to prove that none exists. In order to accomplish this, *constructive search methods* start from the empty assignment and iteratively extend partial assignments until a feasible solution is found. Therefore, the complete search space comprises all distinct assignments of all sizes, partitioned into disjoint levels according to size. An extensive presentation can be found in [1]. In contrast, *repair search methods* iteratively transform complete assignments until a feasible solution is found. Therefore, the complete search space comprises only complete assignments.

This research aims at investigating the factors which determine the *intrinsic hardness* of constructing a solution to any particular constraint satisfaction problem instance. Other than the assumption that a constructive algorithm is employed, instance hardness is treated in a manner independent of the particular

P. Van Emde Boas et al. (Eds.): SOFSEM 2004, LNCS 2932, pp. 184–195, 2004.

tree-search method used, hence the use of the term "intrinsic". In line with [1], as well as [2], [3], our viewpoint focuses on the structure of the induced search space, which allows an abstraction away from problem-specific properties.

The line of reasoning is roughly the following: There exists a set of distinct, possibly overlapping, trajectories through the states of the search space, which start at the unique initial state (the empty assignment containing no instantiations) and terminate at complete feasible assignments. These trajectories are named *solution paths*. See Fig. 1 for an illustration of the search space and solution paths. The entropy of a distribution is a measure of *choice* in selecting

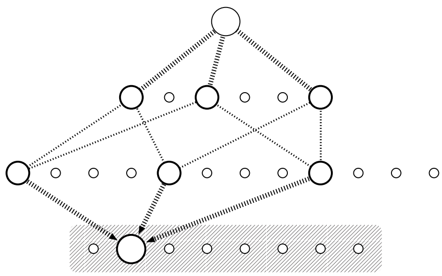

Fig. 1. The search space of a constructive method for instances with $n = 3$ binary variables. The search space is partitioned into disjoint levels, with each level i containing the $2^i C(n, i)$ possible instantiations of size i. The $n!$ possible paths to a particular complete assignment are also depicted. The search space for a repair method comprises the complete assignments in the shaded rectangle

an event [4]. The entropy of the distribution of solution paths among the states of each level of the search space provides a measure of the amount of choice available for selecting a solution path at that level. This measure of choice is named *solution path diversity*. Intrinsic instance hardness is identified with the *deficit* in solution path diversity: the lower the amount of choice available to any algorithm for selecting a solution path, the higher the intrinsic instance hardness. Choice is inherent in the search space of the problem instance and thus independent of the algorithm used to traverse it. In this work, the focus lies on investigating the choices offered by the search space of instances, not on how particular algorithms may make use of such choices. Therefore, throughout the presentation, the reader should bear in mind that the notion of intrinsic instance hardness is not to be confused with computational cost. The former is invariant for a problem instance and manifests itself through the latter, the exact value of which depends on the particular algorithm employed.

To our knowledge, the application of such reasoning in order to characterize intrinsic instance hardness is novel. Moreover, it uncovers a remarkable link between intrinsic instance hardness and *constrainedness* [2]. The latter has been successfully introduced as an estimator of solution cost for problem ensembles