Reformulating Constraint Models Using Input Data*

Martin Michalowski[1], Craig A. Knoblock[1], and Berthe Y. Choueiry[1,2]

[1] University of Southern California, Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292 USA
{martinm,knoblock}@isi.edu
[2] Constraint Systems Laboratory, University of Nebraska-Lincoln
choueiry@cse.unl.edu

1 Motivation

Consider the problem of mapping postal addresses to buildings in satellite imagery using publicly available information, defined as the Building Identification (BID) problem in [1]. This problem takes as input a bounding box that defines the area of a satellite image, buildings identified in the image, vector information that specifies streets in the image, and a set of phone-book entries for the area. The task is to find the set of possible address assignments for each building. In [1], we showed how the task can be framed as a Constraint Satisfaction Problem (CSP), which we solved with an existing solver in [1] and a custom solver in [2]. The CSP is given by $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ where $\mathcal{V}$ is the set of buildings, $\mathcal{D}$ the set of their respective potential addresses, and $\mathcal{C}$ a set of constraints that describe the physical layout of the buildings on the map and address numbering strategies.

In the context of a web application, a typical BID scenario is as follows. A user, presented with a map such as a Google map, either selects a specific building in an area of interest and requests the address of the building, or he/she provides an address and requests the buildings that could have this address. This process is repeated for millions of areas throughout the United States. To answer the entire spectrum of user queries, this application needs to contend with the slight addressing variations found in cities across the US. For example, some cities adhere to a block numbering scheme where addresses increment by a fixed factor (i.e., 100 or 1000) across street blocks while others do not. The direction in which addresses increase also varies, in some cities this occurs to the east while in others it is to the west. In other cities, addresses along East-West running streets increase to the West in one part of town but to the East in the other. Finally, expanding this application to support the rest of the world would require the set of constraints to model new addressing characteristics not seen in the US. The globalization of addressing across continents ensures that some general guidelines are followed, but this standardization is typically met with regional/cultural customization such as the districting in Venice or the historical numbering seen in Japan. The creation of individual models, for each city in the

* This research is supported in part by the Air Force Office of Scientific Research under grant numbers FA9550-04-1-0105 and FA9550-07-1-0416.

world, that account for all of the addressing constraints is an overwhelming and unrealistic task. However, the work required of the expert to define constraints that capture all of the characteristics of addressing seen to date is relatively small and manageable. We propose a framework in which the constraint model of the area of interest for a given user is dynamically built by augmenting the set of *basic* constraints, which form the *generic* constraint model, with those constraints that specify the addressing schema that governs the area of interest.

2 Research Approach

We propose to exploit the information found within a problem instance to enrich the generic model of the CSP in order to identify the set of constraints that apply in a given setting, as shown in Fig. 1.

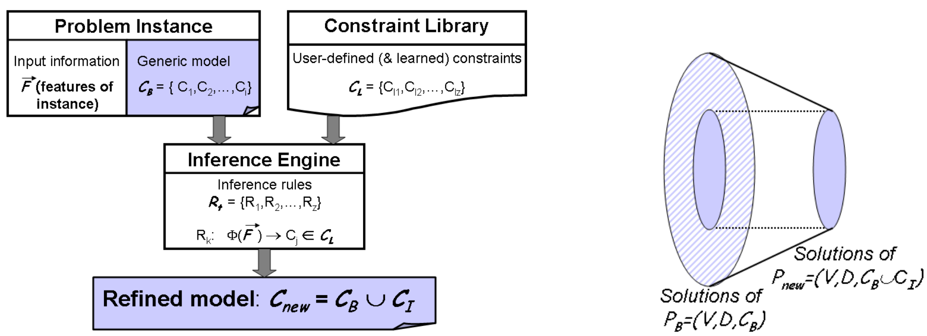

Fig. 1. *Left:* Building the customized constraint model from the generic one. *Right:* Comparing the solution sets of the generic & customized models.

The embedded information that we exploit is a set of instantiated variables, which we call *data points*. Our framework tests the features $\vec{\mathcal{F}}$ of these data points in order to select, from a library of constraints $\mathcal{C}_L$, those constraints $\mathcal{C}_I$ that should be added to the generic constraint model $\mathcal{C}_B$ of the problem. We reduce the load on a human user by limiting their involvement to defining the library of constraints, which is leveraged over the repetitive use of the application over various areas. Subsequently, we use the expert knowledge introduced by the user along with the information found in the problem description to generate a customized problem model that best represents the problem instance at hand. This approach enables a more flexible approach to dynamically modeling problem instances by reformulating problem models and not requiring a collection of individual models that represent all of the foreseeable variations of a problem class. The set of constraints $\mathcal{C}_{new} = \mathcal{C}_B \cup \mathcal{C}_I$ allows us to approach the most accurate model and return more precise solutions (see Fig. 1). We also use constraint propagation on $\mathcal{C}_B$ in order to infer new data points.

References

1. Michalowski, M., Knoblock, C.A.: A constraint satisfaction approach to geospatial reasoning. In: Proc. of AAAI-05. (2005) 423–429
2. Bayer, K., Michalowski, M., Choueiry, B.Y., Knoblock, C.A.: Reformulating constraint satisfaction problems to improve scalability. In: Proc. of SARA-07. (2007)