Biologically Inspired Cognitive Architectures (2014) 8, 109–119

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Scalable methods to integrate task knowledge with the Three-Weight Algorithm for hybrid cognitive processing via optimization

Nate Derbinsky *, José Bento, Jonathan S. Yedidia

Disney Research Boston, 222 Third Street, Cambridge, MA 02142, United States

Received 14 February 2014; received in revised form 11 March 2014; accepted 24 March 2014

KEYWORDS
Optimization;
Cognitive architecture

Abstract
In this paper we consider optimization as an approach for quickly and flexibly developing hybrid cognitive capabilities that are efficient, scalable, and can exploit task knowledge to improve solution speed and quality. Given this context, we focus on the Three-Weight Algorithm, which is interruptible, scalable, and aims to solve general optimization problems. We propose novel methods by which to integrate diverse forms of task knowledge with this algorithm in order to improve expressiveness, efficiency, and scaling across a variety of problems. To demonstrate these techniques, we focus on two large-scale constraint-satisfaction domains, Sudoku and circle packing. In Sudoku, we efficiently and dynamically integrate knowledge of logically deduced sub-problem solutions; this integration leads to improved system reactivity and greatly reduced solution time for large problem instances. In circle packing, we efficiently integrate knowledge of task dynamics, as well as real-time human guidance via mouse gestures; these integrations lead to greatly improved system reactivity, as well as world-record-breaking solutions on very large packing problems. These results exemplify how cognitive architecture can integrate high-level knowledge with powerful optimization techniques in order to effectively and efficiently contend with a variety of cognitive tasks.

* Corresponding author. Tel.: +1 6173000024.
E-mail addresses: nate.derbinsky@disneyresearch.com (N. Derbinsky), jbento@disneyresearch.com (J. Bento), yedidia@disneyresearch.com (J.S. Yedidia).

http://dx.doi.org/10.1016/j.bica.2014.03.007
2212-683X/

Introduction

A central goal of cognitive architecture is to integrate in a task-independent fashion the broad range of cognitive capabilities required for human-level intelligence, and a core challenge is to implement and interface the diverse processing mechanisms needed to support these capabilities.

The Soar cognitive architecture (Laird, 2012) exemplifies a common approach to this problem: Soar integrates a *hybrid* set of highly *specialized* algorithms, which leads to *flexibility* in the types of task knowledge about which it can reason and learn; *efficiency* for real-time domains; and *scalability* for long-lived agents in complex environments. However, since each algorithm is highly optimized, it can be challenging to experiment with architectural variants.

By contrast, work on the Sigma (Σ) architecture (Rosenbloom, 2011) has exemplified how hybrid cognitive capabilities can arise from *uniform* computation over tightly integrated graphical models. When compared with Soar's hybrid ecosystem, this approach allows for comparable flexibility but much improved speed of integrating and experimenting with diverse capabilities. However, utilizing graphical models as a primary architectural substrate complicates the use of rich knowledge representations (e.g. rules, episodes, images), as well as maintaining real-time reactivity over long agent lifetimes in complex domains (Rosenbloom, 2012).

This paper takes a step towards an intermediate approach, which embraces a *hybrid* architectural substrate (ala Soar), but seeks to leverage *optimization* over factor graphs (similar to Sigma) via the *Three-Weight Algorithm* (TWA; Derbinsky, Bento, Elser, & Yedidia, 2013) as a general platform upon which to rapidly and flexibly develop diverse cognitive-processing modules. We begin by describing why optimization is a promising formulation for specialized cognitive processing. Then we describe the TWA, focusing on its generality, efficiency, and scalability. Finally, we present novel methods for integrating high-level task knowledge with the TWA to improve expressiveness, efficiency, and scaling and demonstrate the efficacy of these techniques in two domains, Sudoku and circle packing.

This paper does *not* propose a new cognitive architecture, nor does the work result from integrating the TWA with an existing architecture. Instead, we propose a paradigm and both present and evaluate a set of methods to enable research in integrated cognition.

Optimization

A general *optimization problem* takes the form

$$\underset{\boldsymbol{v}\in\mathbb{R}^n}{\text{minimize}}: \ f(\boldsymbol{v}) = \sum_{a=1}^{M} f_a(\{v\}_a) \qquad (1)$$

where $f(\boldsymbol{v}): \mathbb{R}^n \rightarrow \mathbb{R}$ is the *objective function* to be minimized[1] over a set of variables $\boldsymbol{v}$ and f_a represents a set of M local cost functions (including ''soft'' costs and/or ''hard'' constraints, those that *must* be satisfied in a feasible solution[2]) over a sub-set of variables $\{v\}_a$.

[1] By convention we consider minimization, but maximization can be achieved by inverting the sign of the objective function.

[2] These functions return 0 when satisfied, ∞ otherwise.

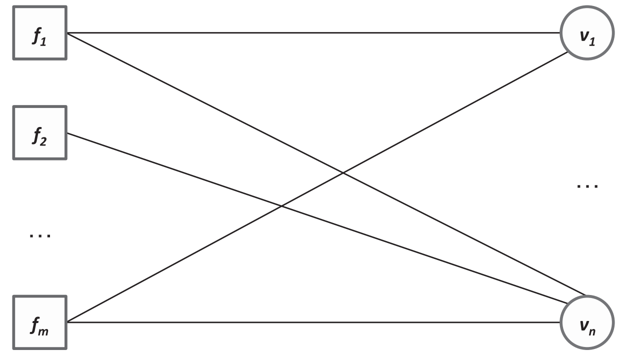

Fig. 1 Factor graph of an optimization problem whose objective function is $f(\mathbf{v}) = f_1(v_1, v_n) + f_2(v_n) + \ldots + f_m(v_1, v_n)$. The circles (right) represent the variables, while the squares (left) represent hard or soft cost functions. If a line connects a square to a circle, that means that the cost function depends on the variable.

As we will exemplify with our discussion of the TWA, it is often useful to consider families or classes of optimization problems, which are characterized by particular forms of the objective and constraint functions. For example, much recent work has been done on *convex* optimization problems, in which both the objective and constraint functions are convex (Boyd & Vandenberghe, 2004). However, neither the TWA nor our proposed approach are constrained to any class of optimization problem.

Optimization is a useful framework in the context of hybrid cognitive processing for two primary reasons: (1) generality of problem formulation and (2) independence of objective function and solution method. First, the form in Eq. (1) is fully general, supporting such diverse processing as constraint satisfaction (a problem with only hard constraints, such as our example tasks) and vision/ perception (e.g. Geman & Geman, 1984). Often these problems are represented as a *factor graph* (Kschischang, Frey, & Loeliger, 2001), as exemplified in Fig. 1. Like other graphical models, factor graphs decompose the objective function into independent local cost functions, reducing the combinatorics that arise with functions of multiple variables.

Another important reason to embrace an optimization framework is that the objective function is formulated independently from the method by which the corresponding problem is solved. This abstraction supports flexibility in experimenting with objective variants without requiring significant effort to change a corresponding algorithm. However, objective-function changes may impact the *speed* and *success rate* of a particular optimization algorithm, and thus it is advantageous to use an optimization algorithm that can specialize to particular classes of objective functions, as well as adapt solving strategies when provided higher-level sources of task knowledge (issues we discuss in greater depth later).

Related work

Broadly speaking, optimization has been applied in three main ways within the cognitive-architecture community. First, optimization has been applied as a methodological