Methods for Integrating Knowledge with the Three-Weight Optimization Algorithm for Hybrid Cognitive Processing

Nate Derbinsky, José Bento, Jonathan S. Yedidia

Disney Research

Architecture Development

	Sigma [Σ; Rosenbloom ’11]	Soar [Laird ’12]
Approach	Uniform inference over tightly coupled *factor graphs*	Hybrid ecosystem of optimized algorithms (e.g. Rete, Inv. Index)
Benefits	Speed of implementing/ evaluating architectural variants	Real-time efficiency, scalability for long and complex tasks
Challenges	Real-time decision cycle, scaling rich representations	Prototyping and evaluating architectural modifications

This Talk. Three-Weight Algorithm (TWA) for hybrid architecture development

- Fully general: *optimization* over factor graphs
- Efficient & scalable: distributed message-passing
- Knowledge integration: novel methods -> better expressiveness, efficiency, scaling
- Examples: CSP (Sudoku=discrete, packing=continuous), planning (robot trajectory)
    - NOT an architecture; platform for modules and/or solving sub-problems

Relevant Applications of Optimization

- Traversing large modeling-parameter spaces
  - Reward signals [Singh et al. '10]
  - Behavioral strategies [Howes, Vera, and Lewis '07]

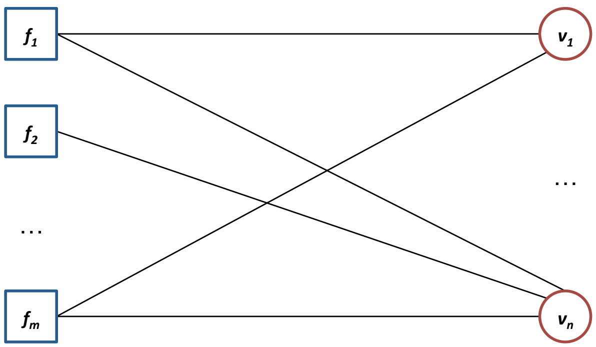
- Analytical framework
  - Rational analysis [Anderson '91]
  - Bounded rationality [Simon '91]
  - Optimality theory [Smolensky and Legendre '11]

Optimization: What and Why

$$\underset{\boldsymbol{v}\in\mathbb{R}^n}{\text{minimize}} : f(\boldsymbol{v}) = f_1(v_1, v_2, \ldots) + f_2(\ldots) + \ldots$$

$$+ \sum c_k(v_1, v_2, \ldots) = \begin{cases} 0 & \text{constraint met} \\ \infty & \text{else} \end{cases}$$

Objective



Factor Graph

<u>Generality</u>

Diverse processing (e.g. constraint satisfaction and vision/perception)

<u>Independence of Problem Specification</u>

Changing objective does *not* require changing solving method
(however, solution time/quality may improve with specialization)

Three-Weight Algorithm (TWA)

Message-passing algorithm [Derbinsky et al. ’13] based on *ADMM* [Boyd et al. ‘11]

General

- Arbitrary objective functions, constraints, and variables
- Global minimum for *convex* problems
- If converges, produces a *feasible* solution (all hard constraints met)

Interruptible

- Iterative algorithm; intermediate results can serve as heuristic start for complementary approaches

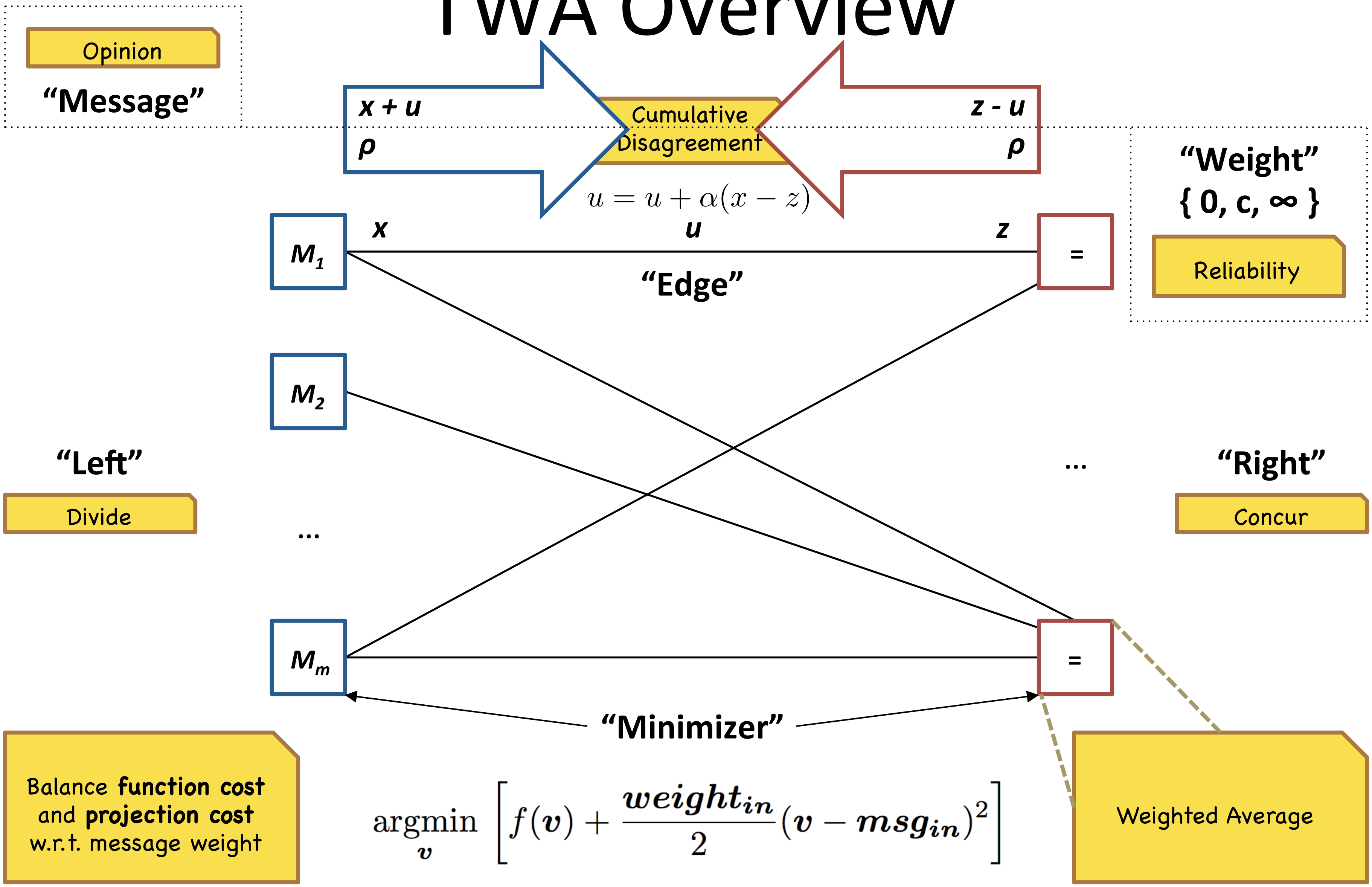
Scalable and Parallelizable

- Formulated as a *decomposition-coordination* problem; leads naturally to concurrency at multiple levels (e.g. MapReduce, multi-core, GPU)

Knowledge Integration – focus of this talk

- Exploit problem structure for efficient sub-problem solutions
- Extract/inject [sub-]symbolic problem state in real-time
- Dynamically re-structure problem

TWA Overview



TWA Iteration

1. Minimize on left – parallel
2. Concur on right – parallel
3. **Reasoner Hierarchy**
   a) **Local - parallel**
   b) **Global - sequential**
      [Graph Dynamics]
4. Convergence check

Knowledge Integration (1)
Minimizers

Integration
- Arbitrary code that implements API: $(x, \rho_{\text{out}}) \Leftarrow f(msg_{\text{in}}, \rho_{\text{in}})$

Expressiveness
- Continuous/discrete variables, hard/soft constraints
- Supports meta-reasoning via recursive/hypothetical problem solving

Efficiency & Scaling
- Graph topology may reduce sub-problem dimensionality
- Operating on edges -> isolated data, all factors [on left/right] operate independently in parallel
- Informed search of variable space often leads to reduced computational complexity, orders-of-magnitude improvement in performance

Knowledge Integration (2)
Reasoner Hierarchy

Integration

- Local: factors* (i.e. executes in parallel)
- Global: code module(s) w/ access to concurred variable values

Expressiveness

- Relational reasoning and rich representations (e.g. rules, perceptual primitives)
- Detect task-specific convergence, feasibility

Efficiency & Scaling

- Yields a discrimination network: local reasoners independently filter data, global acts on results

Knowledge Integration (3)
Graph Dynamics

Integration

- Global reasoners can alter graph topology/parameterization during execution w.r.t. environment, task, and/or agent knowledge/ preferences

Expressiveness

- Dynamic variable sets w/o exhaustive enumeration

Efficiency & Scaling

- Time/Iteration ~ *O*(# factors/variables)
- Time/Variable ~ *O*(# edges)
- Time/Factor may be reduced via reduced edges and/or re-parameterization (e.g. dynamically pruning sub-problem search)

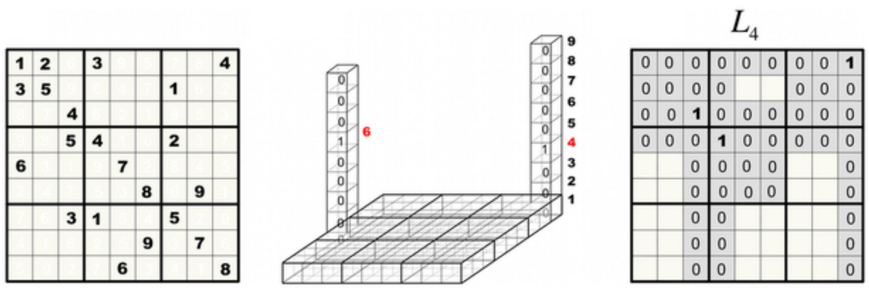
Example Tasks

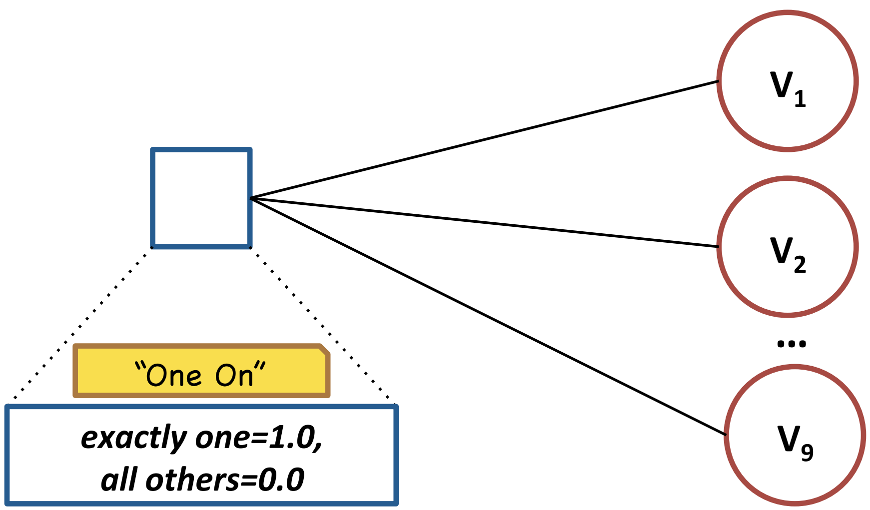
- Constraint Satisfaction (all hard constraints)
  - Sudoku (discrete)
  - Packing (continuous)

- Planning Trajectories [NIPS '13]
  - Global
  - Local

Task 1: Sudoku

All hard constraints, discrete variables



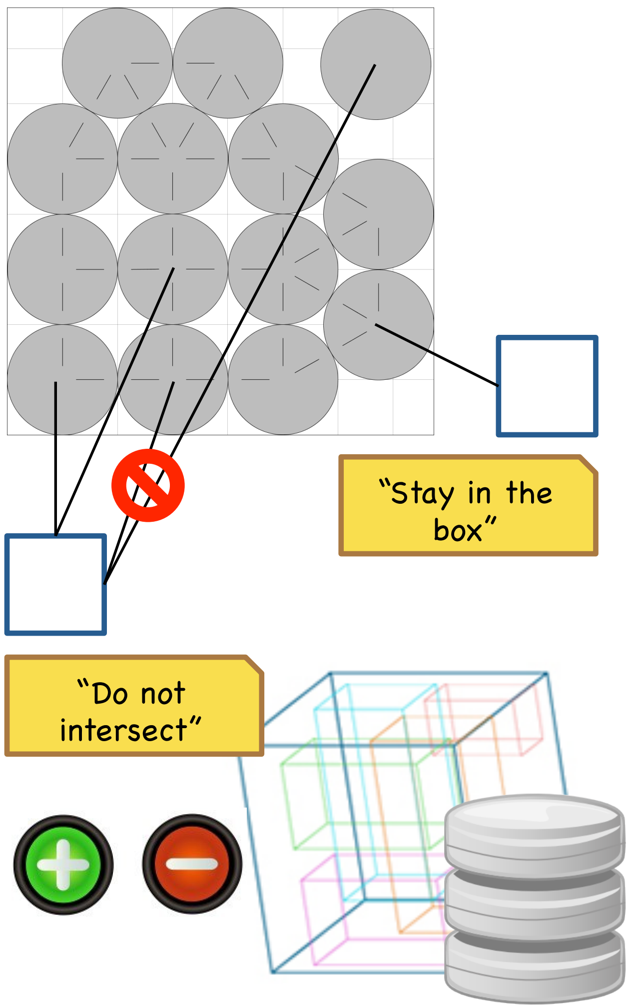


Knowledge Integration

- Minimizers: propagate logical certainty ($\rho=\infty$) from given clues and processing
- Local reasoners: 1-per-cell, discrimination network of possibility-set *changes*
- Global reasoner: remove 4 problem-graph edges per change; remove up to 4 factors and 1 variable
- Results: >10x reduction in graph size and avg. solve time; maintains <11msec avg. iteration for 49x49 on a single core (>125k graph nodes); close to linear speedup (>80%) with added cores for large problems

Task 2: Circle Packing

All hard constraints, continuous variables

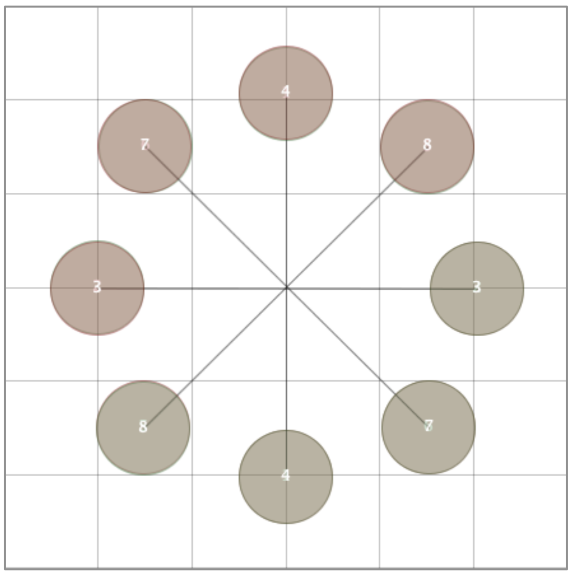


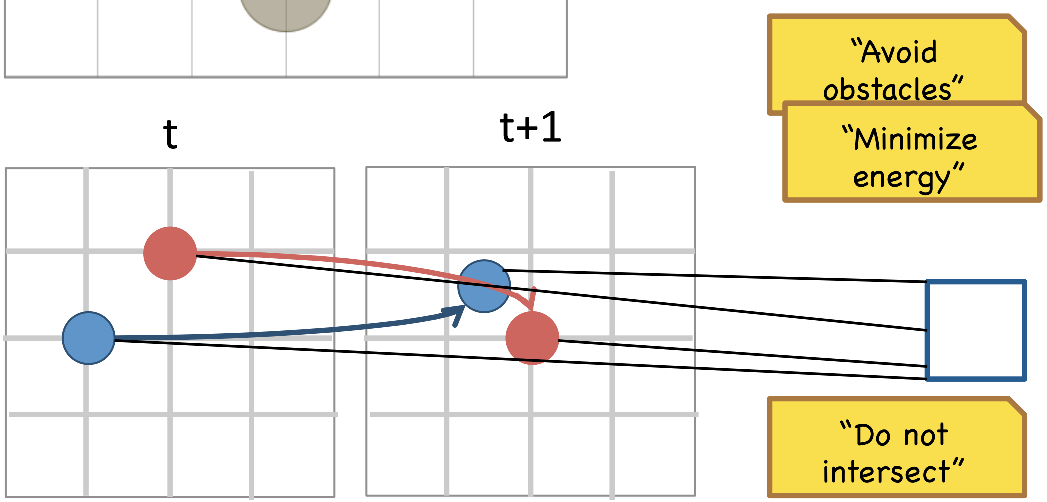
Knowledge Integration

- Minimizers: only influence position of circles violating constraints (else $\rho=0$)
- Global reasoner #1: integrate r-tree to maintain small problem graph (linear vs. quadratic)
- Global reasoner #2: on-demand: identify circles with greatest overlap (via #1)
- Local reasoner: relays positioning messages from top-down reasoning
- Results: record-breaking packing for large instances (1K - 2M); human assistance yields fast, consistent convergence

Task 3: Trajectory Planning [NIPS '13]

Mixed hard/soft constraints, continuous variables





Piecewise Linear: t=1…T

Variants

- Global
- Local (real-time)

Knowledge Integration

- Minimizers: incorporate environmental constraints/ weighting
- Graph Dynamics: prune non-local interactions
- Results: high-quality solutions that scale well with # of agents

LIVE DEMO!

Summary

The Three-Weight Algorithm may serve as a platform for developing cognitive mechanisms that are…

- general
- expressive
- efficient and scalable

Promising candidates:

- Complex constraint satisfaction [w/ high-order K]
- [Motion] planning and simulation
- Probabilistic inference

Questions?

Thank You :-)

Disney Research