University of Zurich[UZH]

Dynamic Distributed Constraint Optimization in Signal/Collect

Thesis January 11, 2015

Daniel Hegglin
of Oerlikon ZH, Switzerland

Student-ID: 08-721-102
dani.hegglin@gmail.com

Advisor: **Mihaela Verman**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

Acknowledgements

- Mihaela - Prof. Bernstein - UZH - Proofreaders

Zusammenfassung

- Stand Constraint Optimization - Stand Dynamic Constraint Optimization - Implementation drei unterschiedlicher Approaches fÃ¼r Constraint Optimization fÃ¼r MeetingScheduling Problem - Benchmark in statischer und dynamischer Umgebung mit unterschiedlichen Konfiguration von Problemen, Agenten, Meetings, Dynamics - Ev. Conclusion Vorgeschmack

Abstract

This thesis is about understanding the performance and behaviour of various existing distributed constraint optimization algorithm approaches (complete, local-iterative message-passing, local-iterative best-response) in context of a dynamic environment, e.g changing constraints or other parameters. The use case for this analysis will be the scheduling problem, which needs to be mapped to the algorithms accordingly. The goals are adding a general module to the existing framework for dcops, which can simulate dynamic environments / parameter changes in various ways, mapping existing dcop algorithms to signal collect and evaluate their performance. An additional goal would be to suggest or test a blended algorithm with local-iterative characteristics that handles change better by applying techniques from dynamic approaches which are not local-iterative. The benchmarking is done with respect to change (resilience to change / stability, amount of variable value changes necessary to bounce back), solution quality (how fast can the algorithms reach a defined quality), Time-to-Convergence (how long does it take to converge).

Table of Contents

1

Introduction

1.1 Motivation & Goal

The goal of this thesis is explore dynamic distributed constraint optimization. Constraint optimization itself is well-researched field and even distributed constraint optimization has been under research. The aspect of having a dynamic environment and dynamic meaning changing constraints is very under-developed and underinvestigated.

The thesis tries to bring further what previously has been achieved in several bachelor and master thesis. The main goal is to show the capabilities of the signal-collect framework and the performance in distributed environments. Important is also mapping of the chosen Meeting Scheduling Problem and And how algorithms like the Max-Sum algorithm can be extended to better perform in dynamic environments. All those implementations shall be tested in benchmarking situations.

1.2 Structure

First I will give an overview about various definitions and aspects of constraint optimization in general, as well as the aspects of distributed and dynamic environments. I am also going to give an overview about different approaches of algorithms to solve constraint optimization problems and their advantages and disadvantages in various contexts. The I will choose appropriate algorithms for the experiments and comparisons and map them to the meeting scheduling problem. I also will design an attempt of introducing dynamic environments that fits into the given framework provided by DDIS. Then I will describe the implemenation details of the algorithms and the testbed, as well as the dynamic environment modelling. After that I will conduct experiments in various testsets, various algorithms, various setups and analyze the limitations of existing algorithms to the problem in dynamic environments, as well as options to optimize defined benchmarks like bounce-back time after amount of change, time to reach certain quality, etc.

2

Background & Related Work

2.1 Distributed Constraint Optimization Definition

2.2 DCO Problems

2.2.1 Graph Coloring Problem

2.2.2 Meeting Scheduling Problem

2.3 Algorithm Design Approaches

2.3.1 Complete

2.3.2 Local-Iterative - Best Response

2.3.3 Local-Iterative - Message Passing

2.3.4 Other Approaches

2.4 Dynamic Environments

3

Design

3.1 Meeting Scheduling Problem

3.1.1 Formal Definition

- General problem and definitions - General constraints Different, Same (sources for that) - Considerations for all algorithms

3.1.2 Problem Generation Considerations

3.2 General Considerations for Dynamics Framework

3.3 Complete - DPOP

3.3.1 Design Considerations

3.3.2 Graph

3.3.3 Vertices

3.4 Local Iterative Best-Respone - Maximum Gain Messaging

3.4.1 Design Considerations

3.4.2 Graph

3.4.3 Vertices

3.5 Local Iterative Message Passing - MaxSum

3.5.1 Design Considerations

3.5.2 Graph

3.5.3 Vertices

-¿ Local Iterative -¿ Good fit for dynamic -¿ Better than the above -¿ Fewer information disclosure

3.6 Dynamic Environment Module

3.7 Testbed, Monitoring

4

Implementation

4.1 Signal / Collect

4.2 Complete - DPOP

Graph

Vertices

4.3 Local Iterative Best-Response - Maximum Gain Messaging

Graph

Vertices

4.4 Local Iterative Message-Passing - Max-Sum

Graph

Vertices

4.5 Monitoring Platform

5

Benchmark

5.1 Experiment Design

The experiments are designed in a two-step procedure accoring to ?

5.2 Testing Environment

5.3 Results I: Algorithms Performance Basic

- Mainly to proof flaws of the other approaches when it comes to dynamic

5.3.1 Solution Quality

- Static / Dynamic - Mainly to proof the ability of new approaches to keep stable and reach certain goals

5.3.2 Time to Convergence

- Heatmap

5.3.3 Scalability

- Number of Messages - Number of Agents - Number of conflicts - Mainly to show that local-iterative scales well because the number of messages is low

5.4 Results II: Algorithms Performance in Dynamic Environments

- Rate / Avg. Conflicts - Density - Stability - Different Scenarios: Constraints, Variables, Domain

5.4.1 Resilience to dynamic Environments

5.4.2 Time to Convergence

5.4.3 Scalability

6

Limitations & Future Work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nullam a tellus. Aliquam commodo dui non ipsum. Duis mollis nisi id turpis. Donec quis ipsum. Curabitur sed nibh. Morbi suscipit justo quis orci. Ut massa tortor, ultricies vitae, lacinia eu, facilisis eu, nisl. Nulla mattis urna sed metus imperdiet ornare. Praesent sodales. Etiam laoreet. Mauris quam magna, sagittis et, pharetra eget, congue vitae, arcu. Fusce sollicitudin justo. Suspendisse lectus. Sed lobortis dolor quis lectus scelerisque ornare. Integer purus. Phasellus vel elit at nibh sagittis lobortis. Aliquam iaculis malesuada eros. Mauris metus.

7

Conclusions

- Ausgangslage - Was wurde gebaut - Was wurde getestet - Was wurde herausgefunden
- Was kÃ$\mathbf{P}_n$nte nÃ$\frac{1}{4}$tzlich sein (testbed)

References

A

Appendix 15

A.1 Results I: Selected Data

A.2 Results II: Selected Data

List of Figures

List of Tables