Chapter 1

Introduction

This thesis addresses search algorithms for Distributed Constraint Satisfaction problems. Distributed Constraint Satisfaction is a novel research topic related to two well-known research areas: Constraint Satisfaction and Distributed Algorithms. Constraint Satisfaction is the process of finding a solution to a Constraint Satisfaction problem (*CSP*). A *CSP* consists of a set of *variables*, each one taking a value in *finite domain*. Values are related by *constraints* that impose restriction to the values that variables can take. A *CSP solution* is an evaluation of these variables that satisfies all constraints.

A Distributed Constraint Satisfaction problem (*DisCSP*) is, thus, a *CSP* whose variables and/or constraints are geographically distributed among communicating agents[1] and cannot be solved by using the centralized approach.[2] Analogously to *CSP*, a *DisCSP solution* is an assignment of values to variables which satisfies every constraint. In an algorithm for *DisCSP*, agents cooperate and exchange messages in order to find a solution.

A *distributed algorithm* can be classified in two main classes: *synchronous* and *asynchronous*. In between these two classes are *hybrid* algorithms, which combine elements of both algorithm type. In general, a synchronous algorithm is based on the notion of *privilege*, a token that is passed among agents. Only one agent is active at any time, the one having the privilege, while the rest of agents are waiting. When the process in the active agent terminates, it passes the privilege to another agent, which now becomes the active agent. In an asynchronous algorithm every agent is active at any time, and they do not have to wait for any event. In a hybrid algorithm, an agent may be required to wait for some special event, but not for every event. In this thesis we consider these three different classes of distributed algorithms for *DisCSP*.

Search is one of the common approaches for *CSP* as well as for *DisCSP*. A *search algorithm* consists in searching a solution within the search space defined by all possible solutions to the problem. A search method is *complete* if the

[1]By agent we mean a software and/or hardware component capable of acting exactingly in order to accomplish tasks on behalf of its user [Nwana and Ndumu, 1997].

[2]By *centralized* we mean single processor, as opposed to *distributed*.

exploration of the search space is systematic and it is conducted until a solution is found or the absence of a solution is proven.

1.1 Motivation

Distributed Constraint Satisfaction is motivated by the existence of naturally distributed *CSP*, for which it is impossible or undesirable to gather the whole problem knowledge into a single agent and to solve it using the centralized approach. There are several reasons for that. The cost of collecting all information into a single agent could be taxing. This includes not only communication costs, but also the cost of translating the problem knowledge into a common format, which could be prohibitive for some applications. Furthermore, gathering all information into a single agent implies that this agent knows every detail about the problem, which could be undesirable for security or privacy reasons [Yokoo et al., 2002].

Distributed *CSPs* can be found in many real domains such as scheduling, planning and as part of many coordination processes in multi-agent systems. This thesis contributes to the development of *DisCSP* solving methods according to two issues: efficiency and privacy.

Distributed Constraint Satisfaction is an NP-complete task and hence all *DisCSP* algorithms have, in the worst case, exponential time in the number of variable of the problem. In such context, the development of algorithms to solve *DisCSPs* as efficient as possible is necessary. This intractability of *DisCSP* has motivated the part of our work. We present new approaches and heuristics in order to improve the efficiency of some state-of-the-art algorithms.

As mentioned above privacy is one of the main motivations to solve distributed *CSP* in a distributed form. *Privacy* in *DisCSP* is concerned with the desire of agents to conceal their information about the problem. Most of the existing algorithms for *DisCSP* were conceived without taking into account privacy issues (their agents give constraints and exchange value freely). This makes them to be unsuitable to solve naturally distributed problems where privacy is the main concern. Examples of this kind of problems are: Stable Marriage and Stable Roommate problems [Gusfield and Irving, 1989] where agents often want to keep their personal preferences private. There exists two main approaches: those that use cryptographic techniques [Silaghi and Mitra, 2004, Yokoo et al., 2005, Nissim and Zivan, 2005] and those that enforce privacy by different strategies but excluding cryptography [Silaghi, 2002, Brito and Meseguer, 2003, Brito and Meseguer, 2005b, Zivan and Meisels, 2005a]. For many applications the use of secure algorithms based on cryptographic methods is costly and difficult to implement. In our work we are concerned with *DisCSP* algorithms that leak less information that existing ones. We present new *DisCSP* algorithms that achieve a higher degree of privacy than existing approaches.

1.2 Scope and Orientation

The boundaries of this work are established by the following decisions:

- *Practical Application.* This work is oriented to further extend the applicability of *DisCSP* algorithms to solve naturally distributed problems. Our contributions are new approaches and heuristics that improve the performance of existing algorithms according to two issues: efficiency, in terms of computation cost, and privacy.

- *General Distributed Constraint Solving.* Except for the problems considered in Part "Applications", all algorithms and ideas that we propose in this thesis are applicable for solving any *DisCSP* whose agents hold only one variable. The extension of proposed approaches and heuristics for naturally handling multi-variable variants could be part of further research.

- *Complete Search.* A different approach to avoid the computationally intractability of Distributed Constraint Satisfaction uses incomplete search schemas, and they do not guarantee to find a solution to a solvable instance. In our work, however, we are concerned with complete search algorithms with polynomial space complexity.

- *Empirical evaluation.* Because of the practical orientation of our work and the intractability of *DisCSP*, the assessment of our contributions is mainly supported by empirical methods. In our experiments, we have used a set of benchmarks widely used in the *CSP* community.

1.3 Contributions

In the following we give the main contributions of this thesis:

- *A Family of Asynchronous Backtracking Algorithms.* We present a basic kernel for grouping asynchronous backtracking algorithms. We believe that this characterization of asynchronous backtracking will help better understand these non-trivial mechanisms. By implementing the condition for termination in this kernel, we obtain four asynchronous algorithms. One of these approaches does not add links between agents not sharing constraints, which can be useful for solving problems where privacy is the main concern. These ideas are gathered in:

  – C. Bessière, A. Maestre, I. Brito, P. Meseguer. *Asynchronous Backtracking without Adding Links: a New Member to ABT Family.* **Artificial Intelligence**. Volume 161, Issues 1-2. pp. 7-24. January, 2005.

- *Synchronous Backtracking.* The use of variable reordering heuristics for constraint satisfaction problems has been shown to be a powerful strategy in order to improve efficiency. Inspired in this idea, we present two approximations of the popular minimum-domain heuristic for dynamic variable reordering.

  - I. Brito. *Synchronous, Asynchronous and Hybrid Algorithms for DisCSP*. Proceeding of the Tenth International Conference on Principles and Practice of Constraint Programming (CP-2004). **Lecture Notes in Computer Science**, Volume 3258, p. 791, Jan 2004.

- *Hybrid Algorithms.* We present a novel algorithm which combines synchronous and asynchronous elements. This algorithm outperforms the reference asynchronous backtracking algorithm. A comparison about synchronous, asynchronous and hybrid algorithms as well as variable reordering heuristics for synchronous ones can be found in:

  - I. Brito, P. Meseguer. *Synchronous, Asynchronous and Hybrid Algorithms for DisCSP*. **Fifth International Workshop on Distributed Constraint Reasoning at the Tenth International Conference on Principles and Practice of Constraint Programming (CP-2004)**. Toronto, Canada. September, 2004.
  - I. Brito, F. Herrero, P. Meseguer. *On the Evaluation of DisCSP Algorithms*. **Fifth International Workshop on Distributed Constraint Reasoning at the Tenth International Conference on Principles and Practice of Constraint Programming (CP-2004)**. Toronto, Canada. September, 2004.

- *Non-binary Constraints.* Although most of state-of-the-art methods for *DisCSP* assume that every constraint involves two variables, they can be extended to handle constraints involving more than two variables. We present new versions of existing algorithms to deal with non-binary constraints, including the addition of redundant constraint projections.

  This ideas were published in:

  - I. Brito, P. Meseguer. *Asynchronous Backtracking Algorithms for Non-binary DisCSP*. **Workshop on Distributed Constraint Satisfaction Problems at the 17th European Conference on Artificial Intelligence (ECAI-2006)**, Riva del Garda, 2006.

- *Assignment Privacy.* We propose an asynchronous algorithm that allows agents to maintain their variable assignments private during problem resolution. This algorithm is based on the idea that agents exchange sets of consistent values instead of their own assignments.

- *Constraint Privacy.* We present the Partially Known Constraint model (*PKC*), a new *DisCSP* model in which constraints are kept private and are only partially known to agents. We propose two algorithms to solve *DisCSPs* expressed under the *PKC* model. These algorithms also preserve agents' assignments. The ideas related to assignment and/or constraint can be found in:

   – I. Brito, P. Meseguer. *Distributed Forward Checking.* Proceeding of the Night International Conference on Principles and Practice of Constraint Programming, CP-2003. **Lecture Notes in Computer Science**, Volume 2833, pp. 801 - 806, November 2003.

- *Enforcing Privacy with Lies.* We present a novel algorithm to further enforce constraint privacy. This is algorithm is based on the idea that agents may lie. It requires a single extra condition: if an agent lies, it has to tell the truth in finite time afterwards.

   – I. Brito, P. Meseguer. *Distributed Forward Checking May Lie for Privacy*, **Recent Advances in Constraints, Lecture Note in Artificial Intelligence**. Volume 4651, 2007.

- *Applications.* We consider naturally distributed problems which have a clear motivation to be tried with distributed techniques. We examine some *DisCSP* algorithms for solving several versions of two well-known Stable Marriage problems [Gusfield and Irving, 1989]: Stable Marriage and Stable Roommates. We propose a way to resolve these problems while keeping personal preference private. Four publications develop privacy issues related to several versions of the Stable Matching:

   – I. Brito, P. Meseguer. *The Distributed Stable Marriage Problem with Ties and Incomplete Lists.* **Workshop on Distributed Constraint Satisfaction Problems at the 17th European Conference on Artificial Intelligence (ECAI-2006)**, Riva del Garda, 2006.

   – I. Brito, P. Meseguer. *Distributed Stable Matching Problems with Ties and Incomplete Lists.* Proceeding of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP-2006). **Lecture Notes in Computer Science**, Volume 4204, pp. 675-680, Nantes, 2006.

   – I. Brito, P. Meseguer. *The Distributed Stable Marriage Problem.* **Sixth International Workshop on Distributed Constraint Reasoning at the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005)**. Edinburgh, Scotland, 30 July - 5 August, 2005.

   – I. Brito, P. Meseguer. *Distributed Stable Matching Problems.* Proceeding of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP-2005). **Lecture Notes in Computer Science**. Volume 3709. pp. 152-166. Sitges, Spain. September, 2005.

1.4 Thesis Structure

This document is divided in five Parts and one Appendix. Parts II, III and IV contain the contributions of our work.

- *Part I Background*: This part includes two Chapters. Chapter 2 and 3 are an overview of Constraint Satisfaction and Distributed Constraint Satisfaction frameworks, respectively. In both, we formally define the corresponding problems and briefly describe the main solving algorithms that will useful to understand the following Chapters.

- *Part II Approaches*: This part has four chapters. In Chapter 4, we study synchronous backtracking algorithms and present two approaches for variable ordering. In Chapter 5 we consider four asynchronous backtracking algorithms. In Chapter 6 we present a novel hybrid algorithm, which we evaluate against synchronous and asynchronous approaches. In Chapter 7 we present several distributed algorithms to deal with non-binary constraints and consider the idea of add constraints projections in order to speed up the search.

- *Part III Privacy in DisCSP*: This part contains two chapters. In Chapter 8 we study three types of privacy in *DisCSP*: domain, assignment and constraint privacy. We propose three asynchronous algorithms for enforcing assignment and/or constraint privacy. In Chapter 9, we further enforce constraint privacy in of the algorithms studied in Chapter 8 by allowing agents to lie.

- *Part IV Applications*: In this part we resolve two problems with privacy requirements: Meeting Scheduling (Chapter 10) and Stable Matching problems (Chapter 11).

- *Part V Conclusions and Appendix*: This part contain two Chapters: Chapter 12 where we present the conclusions and further research of our work and Appendix A, where we analyze centralized and distributed specialized algorithms to Stable Matching problems with privacy requirements.

1.5 Abbreviations

The abbreviations used in this thesis are summarized next:

CSP	Constraint Satisfaction Problem
DisCSP	Distributed Constraint Satisfaction Problem
BT	Chronological Backtracking algorithm
DepDB	Dependency-directed Backtracking algorithm
GBJ	Graph-based Backjumping algorithm
CBJ	Conflict-based Backjumping algorithm
FC	Forward Checking algorithm
GLS	Generic Local Search algorithm
BO	Break-out algorithm
MAC	Maintaining Arc-Consistency algorithm
DisMS	Distributed Meeting Scheduling problem
SensorDCSP	Sensor-mobile problem
SBT	Synchronous Chronological Backtracking algorithm
SCBJ	Synchronous Conflict Backjumping algorithm
ABT	Asynchronous Backtracking algorithm
AWC	Asynchronous Weak-commitment algorithm
DIBT	Asynchronous Graph-based Backjumping algorithm
AAS	Asynchronous Aggregation Search
DMAC-ABT	Distributed Maintaining Asynchronous Consistency algorithm
ABT-DO	Dynamic Ordering Asynchronous Backtracking algorithm
ConBT	Concurrent Backtracking Search
ConDB	Concurrent Dynamic Backtracking
DisBO	Distributed Break-out algorithm
amd1	An Approach of the Minimum-Domain heuristic
amd2	An Approach of the Minimum-Domain heuristic
$SCBJ_{amd1}$	*SCBJ* plus *amd1*
$SCBJ_{amd2}$	*SCBJ* plus *amd2*
ABT_{kernel}	Kernel of *ABT* family
ABT_{all}	*ABT* with all potentially new links added in advance
ABT_{temp}	*ABT* with temporary new links
ABT_{not}	*ABT* without adding new links
ABT_{hyb}	*ABT*-like algorithm with synchronous and asynchronous elements
ABT_{proj}	Non-binary *ABT* with constraint projections
$SCBJ_{proj}$	Non-binary *SCBJ* with constraint projections
ABT_1	*ABT* with the single phase strategy
ABT_2	*ABT* with the two-phase strategy
$DisFC_1$	*DisFC* with the single phase strategy
$DisFC_2$	*DisFC* with the two-phase strategy
$DisFC_{lies}$	$DisFC_1$ with lies
SM	Stable Marriage problem
EGS	Extended Gale-Shapley algorithm
SMI	Stable Marriage problem with Incomplete Lists
DisEGS	Distributed Extended Gale-Shapley algorithm
DisSM	Distributed Stable Marriage problem
DisSMI	Distributed Stable Marriage problem with Incomplete Lists

SR	Stable Roommates problem
SRI	Stable Roommates problem with Incomplete Lists
DisSR	Distributed Stable Roommates problem
DisSRI	Distributed Stable Roommates problem with Incomplete Lists
SMT	Stable Marriage problem with Ties
SMTI	Stable Marriage problem with Ties and Incomplete Lists
DisSMT	Distributed Stable Marriage problem with Ties
DisSMTI	Distributed Stable Marriage problem with Ties and Incomplete Lists
SRT	Stable Roommates problem with Ties
SRTI	Stable Roommates problem with Ties and Incomplete Lists
DisSRT	Distributed Stable Roommates problem with Ties
DisSRTI	Distributed Stable Roommates problem with Ties and Incomplete Lists
DisFC-SM	Distributed Forward Checking algorithm for finding Stable Matchings
MS	Meeting Scheduling problem
DisMS	Distributed Meeting Scheduling problem