Contributions and Conclusions

Asynchronous search are useful in applications like negotiations (Generalized English Auctions), distributed meeting scheduling with private constraints, distributed configuration with secret know-how.

Asynchronous search also has an important theoretic value, defining generic techniques whose properties, once proven, apply to a large number of not only distributed but also centralized old and new algorithms. Convincing is the technique I propose for using backtracking nogoods in consistency (Section 7.3.1), technique that can be straightforwardly applied to classic centralized versions of backtracking (as I show in Section 5.2) with improvements of efficiency.

Among its main contributions, this thesis:

- Describes an abstraction technique for centralized discrete CSPs with ordered domains. This technique is called "aggregations" and treats subspaces of the search space is an dynamically aggregated manner. Variants for general discrete domains, related to (Hubbe & Freuder 1992), are also analyzed.

- Extends the use of aggregation techniques to numeric constraints in centralized CSPs. This is an optimized technique of removing redundant work performed by previous numeric backtracking that uselessly split and check inequality constraints that are fully feasible on some branches of the search. Domains are split recursively, and the same constraints are recursively reverified until complete satisfaction or an acceptable resolution is obtained.

- I noticed that the use of CSPs techniques for numeric problems over $\mathbb{R}$ was possible by something that can be described as: *letting the domain splits have existential meaning rather than the universal meaning in classical backtracking.* Existing asynchronous search protocols supported only proposals with universal quantifiers (e.g. **require**$(\forall x \in \{1\})$). I modify and generalize existing asynchronous search protocols to support proposals with existential quantifiers (e.g. **require**$(\exists x \in \{[1.5, 10^5] \cup \{0, 1\}\})$). This efficient technique is the first to allow a **proper use of intervals** in proposals and is a first step I took for extending the applicability of asynchronous search to Numeric CSPs. Since this idea came when I tried to integrate in ABT an aggregation technique described in (Silaghi *et al.* 2000c), the first protocol with the desired property is called Asynchronous Aggregation Search (see Chapter 9).

- Local consistency is a technique bringing gains that are exponential in the size of the problem, by deposing an effort that is exponential in the size of the used predicates (constraints). The effort can therefore be polynomial in the size of the global problem. **Exponential gain with polynomial cost** made local consistency the most important technique in constraint satisfaction where it was shown that any opportunity must be used for exploiting it → *consistency maintenance.* I generalize the concept of centralized consistency maintenance in a way that makes it applicable to asynchronous techniques. This is the first technique allowing for consistency maintenance in asynchronous search and is very efficient (Chapter 7). It also:

  - shows how to tune the guarantees of the strength of achieved consistency, and
  - introduces the idea of using backtracking nogoods in consistency maintenance.

- Negotiations are defined by the fact that the participants yield from their constraints to reach common solutions. This is technically called: constraint relaxation. No such technique existed for private problems. I propose such a technique for privacy on constraints (Section 17.10). A classical dualism for CSPs defines a standard transfer between privacy on constraints and privacy on domains (see Section 3.5.2 and Chapter 18). We have also defined a primal version of this technique.

- Details issues related to dynamism in asynchronous search, and proposes a few protocols to deal with some types of events like agents that join, leave, or crash and recover during the search.

- Defines a framework for negotiation and English Auctions, based on distributed valued CSPs. Versions of asynchronous Branch and Bound techniques are described for the new framework (Silaghi *et al.* 2001d).

- Discovers a security problem for the use of asynchronous search in negotiations. Proposes a technique that is slow, but enables verification for security (Silaghi *et al.* 2001c).

- Agents involved in existing algorithms had a static priority enforced on them. This is a drawback. It entangles agents to *verify conflicts in Generalized English Auctions* (Chapter 17), for defending themselves from unfair competition. One only knew to remove that constraint at the expense of requiring an exponentially bounded computer memory (e.g. RAM). This is infeasible and leads to unguaranteed protocols. I designed the first solving protocol that **allows the needed dynamic reordering** and still has **a guaranteed termination** with polynomially bounded computer space requirements. Moreover, the designed technique is very general as it can introduce all important reordering heuristics used for efficiency in centralized techniques (see Chapter 10).

- Besides the aforementioned reordering technique, I propose a suite of other techniques for asynchronous dynamic reordering with other types of heuristics and still offering guaranteed termination with polynomial space requirements: Reordering with increasing delay restarts (Section 10.3.1), and Finite number of reorderings. These two techniques are simple but allow only simplistic reordering heuristics. Dynamic Partial Order is discussed in Section 10.4.

- Describes a few trials for finding efficient reordering heuristics for polynomial space asynchronous search (Silaghi *et al.* 2001h).

- Asynchronous techniques requested an agent only once its proposal. The proposals were therefore necessarily binding. This is incompatible with the practice required for numeric problems, as mentioned at the previous item. After understanding this problem, I proposed a modeling technique called Replica-based search where several replicas of the same agent are asked proposals at different levels of abstractions. As a result, proposals of the replicas are not binding and can be refused by subsequent replicas (e.g. After an agent sends **require**($\exists x \in \{[1.5, 10^5] \cup \{0, 1\}\}$), it can be more restrictive in a subsequent replica, **require**($\exists x \in \{[15, 1000]\}$) and later can even abandon completely its proposal, **require**($x \notin \{[1.5, 10^5] \cup \{0, 1\}\}$)). This technique, described in Chapter 13 closes the sequence of ideas required for developing asynchronous techniques for numeric problems over $\mathbb{R}$.

- One of the main motivations for distributed algorithms is the enforcement of privacy. A framework (DisPrivCSPs) for modeling and quantifying privacy loss and a theoretic comparison on the power of different protocols in saving privacy are proposed in Chapter 16. The proposed metric is feeble but meaningful.

- Proposes and discusses adaptations of secure multi-parties computation techniques to DisCSPs. An expensive secure protocol (SCSPS) is proposed in Chapter 15. I also developed and analyzed a set of cryptographic protocols for both discrete and numeric distributed satisfaction problems (Annex B). An additional contribution of the work described in Annex B

with respect to related work appearing simultaneously, is the discovery of the feebleness of the corresponding techniques, the S-attacks.

The obtained cryptographic protocols have some drawbacks such that a trade-off remains between the competing techniques. Cryptographic protocols and distributed algorithms will continue to coexist for a while.

- Several optimization techniques have been developed so far. Those techniques are not general and abstract enough for allowing extensions like consistency maintenance and reordering. The notion of cost was unclear. I introduce the notion of cost in nogoods, obtaining a general stand-alone communication atom. In Chapter 14 I describe a generalization of all the existing distributed satisfaction and optimization techniques. This generalization allows the introduction of consistency maintenance, reordering, etc. in asynchronous distributed optimization.

- Describes a way for enabling parallel proposals (Silaghi & Faltings 2001), adding additional flexibility compared to the proposal in (Hamadi 1999b).

- Describes a framework extending DisCSPs for in a way adapted to distributed configuration (Chapter 18). This was joint work with Markus Zanker.

- Studies an approach to CSPs called Full Approach, with advantages over primal and dual approaches. The use of the Full Approach is not completely new as it is familiar with numeric constraints, but it has not been often studied as a stand alone framework.

- Proposes a technique for bound consistency, (BC1999), introducing elements that are used later for the well known AC2001/AC3.1

Distributed constraint satisfaction is a promising framework that will be the intermediary step in developing important Internet-based applications. Results of some basic research is still required for arriving there, even if we may already be very close. Techniques like consistency maintenance, abstractions and aggregations proved very useful and are worth the difficulty of the implementation.

Part

Annexes