The behavior of distributed search algorithms on DisCSPs can be studied on a set of three very different families of DisCSP algorithms. All search algorithms on DisCSPs can be divided into two families: single search process algorithms (SPAs) and concurrent (multiple) search process algorithms (CSAs). The only former experimental study of the performance of DisCSP algorithms compared two asynchronous single search algorithms [4].

The state of single process algorithms is defined by a single tuple of assignments, one for each agent. When this set of assignments is complete (containing assignments to all variables of all agents) and consistent, the SPA stops and reports a solution. A simple representation for the state of any synchronous SPA, like SBT [61] or CBJ [68], is a data structure that holds the current partial assignment of the search (CPA). Single search process algorithms can be asynchronous. In asynchronous backtracking (ABT) [9, 61], each agent holds its view of the current assignments of other agents in a single Agent_View (Chapter 5). When all agents are idle, all Agent_Views are consistent and a solution is reported [9, 61].

In concurrent search, multiple concurrent processes perform search on non-intersecting parts of the global search space of a DisCSP (25, 55, 67). All agents in a CSA participate in every search process, since each agent holds some variables of the search space. Each agent holds the current domains of its variables, for each of the search processes. Messages of CSAs carry the IDs of their related search process and the agents use the corresponding current domains for consistent assignments. The concurrent backtracking algorithm (ConcBT), distributes among agents a dynamically changing number of search processes (25, see Chapter 7). Agents generate and terminate search processes dynamically during the run of the ConcBT algorithm. The concurrent dynamic backtracking (ConcDB) algorithm incorporates dynamic backtracking to the concurrent performing search processes. As a result, one search procedure can reveal a dead end of another concurrent search procedure and terminate it (75).
In interleaved asynchronous backtracking, agents participate in multiple processes of asynchronous backtracking. Each agent keeps a separate Agent View for each search process in IDIBT [25]. The number of search processes is fixed by the first agent. The performance of concurrent asynchronous backtracking [25, 55] was tested and found to be ineffective for more than two concurrent search processes [25].

The general model of DisCSPs has variables owned by agents, who assign them values. The distinction between the two families of algorithms is in the number of concurrent assignments that agents maintain. In SPAs each agent can have no more than one assignment to its variable at any single instance. In multiple process algorithms (MPAs), on the other hand, agents maintain multiple concurrent assignments to their variable. To give an example, synchronous backtracking (SBT) is a single process algorithm. During search, a single CPA carries the assignments of some of the agents. The other agents which are waiting for the message with assignments to arrive are still unassigned. Therefore, each agent, in every step of the search, has either one assignment or none. Asynchronous backtracking (ABT) is also an SPA. All the variables in ABT have exactly one assignment at each instant of its run [9].

To maintain two concurrent assignments in a DisCSP, let us go back to Chapter [7]. Think of the first agent as assigning two of its values to its variable. It then puts each assignment on a different message and initializes a backtracking process for each one. Each backtrack process traverses all agents, not in the same order, to accumulate assignments to all variables of all agents. All agents eventually receive two messages. One message has the first assignment for the first agent and the other has the second assignment that the first agent performed. Agents that receive a message either add their compatible assignment to the partial assignment already on the message or backtrack by sending the message back. All agents use a different current domain for each of the messages. It is easy to see that all agents react to the two messages in exactly the same way, assigning their variable to it or backtracking. This process stops when one of the messages accumulates a complete assignment and reports a solution or when both messages return to the first agent and find no more values to assign. In this case the two-process algorithm reports failure.

Several single process DisCSP search algorithms have appeared in the literature in the last decade: synchronous algorithms like synchronous backtrack (SBT) and conflict-based backjumping (CBJ) [61, 68]; asynchronous algorithms like asynchronous backtracking (ABT), asynchronous aggregations search (AAS) and asynchronous forward-checking (AFC) [44, 54]. In contrast, only a few multiple process DisCSP search algorithms appear in the literature [25, 55, 75]. The concurrent dynamic backtracking algorithm (ConcDB), with dynamic splitting of search processes, will be the representative of this family in the present study. ConcDB incorporates dynamic splitting, generating a variable number of search processes. Furthermore, the search processes