Collective Problem Solving through Coordinated Reaction

JyiShane Liu and Katia P. Sycara

Abstract—**We present a methodology, called Constraint Partition and Coordinated Reaction (CP&CR), for collective, evolutionary problem solving. Problem solving is viewed as an emergent functionality from the evolving process of a group of diverse, interacting, and well-coordinated reactive agents. Cheap and effective search knowledge is extracted from local interactions and embedded in the coordination mechanism. Our domain of problem solving is constraint satisfaction problems. We have applied the methodology to job shop scheduling, an NP-complete constraint satisfaction problem. Experimental results on a benchmark suite of problems show that CP&CR outperformed three other state-of-the-art direct search scheduling techniques, in both efficiency and number of problems solved. In addition, CP&CR was experimentally tested on problems of larger sizes and showed favorable scaling-up characteristics.**

KeyWords—**Collective Problem Solving, Agent Society, Coordination Strategy, Coordinated Search**

I. Introduction

PROBLEM-SOLVING has been an important research area for Artificial Intelligence. Most traditional approaches rely on variants of *direct heuristic search* on the state-space of the problem [11]. Problem solving knowledge and heuristics are used to prune the search space and guide the exploration [6]. In contrast to direct search, *evolutionary approaches* to computation, where the solution is achieved incrementally through asynchronous interactions of a society of agents, have been gaining ground. These approaches investigate and exploit implications of evolutionary principles for computation [14] [5] [2]. Systems that incorporate evolutionary approaches have been developed by, for example [13] [1] and [4], where global system functionality emerges from interaction of its subcomponents. Building upon these ideas, we have developed a computational framework to perform goal-directed search efficiently. We present the approach and experimentally demonstrate that it outperforms direct search methods for a set of benchmark problems.

The focus of our research is Constraint Satisfaction Problems (CSPs). Many problems of theoretical and practical interest (e.g., parametric design, resource allocation, time-dependent scheduling) can be formulated as CSPs. Informally, a CSP is defined by (1) a set of *variables*, each of which *is instantiated* (i.e. takes its value) from a given domain, and (2) a set of *constraints* that restrict the admissible variable instantiations. A solution to a CSP is an assignment of values (an instantiation) for all variables, such that all constraints are satisfied. Numerical CSPs (NCSPs) [7] are a subset of CSPs, in which constraints are represented by numerical relations between quantitative variables usually with fairly large domains of possible values. Direct search constraint satisfaction algorithms typically suffer from feasibility/efficiency problems for NCSPs.

In this paper, we present an approach, called Constraint Partition and Coordinated Reaction (CP&CR), in which an NCSP is collectively solved by a set of agents with simple local reactions and effective coordination. CP&CR divides an NCSP into a set of subproblems according to considerations of functionality (constraint type) and responsibility (variable instantiation). Each subproblem is assigned to an agent. Interaction characteristics among agents are identified. Agent coordination defines agent roles, the information they exchange, and the rules of interaction. Problem solving emerges as a result of the evolving process of the group of interacting and coordinating agents. The fundamental characteristics of CP&CR are: (1) divide and conquer, (2) simple local reactions, and (3) effective coordination. Experimental results (presented in section 4) on a set of benchmark problems attest to the effectiveness of the approach as compared with other constraint-based, direct search methods.

II. Collective Problem Solving

CP&CR is a framework of collective problem solving which employs several principles of evolutionary computation, such as emergence, diversity, solution evolution. The underlying idea of CP&CR is to solve a problem collectively and incrementally by a group of specialized and well-coordinated agents. Each agent reacts to each other's action and communicates with each other by leaving and perceiving particular messages on the objects it acts on. A solution emerges from the evolutionary process of the group of diverse agents interacting with each other. Specifically, CP&CR provides a framework to decompose an NCSP into a set of subproblems, identify their interaction characteristics and, accordingly construct effective coordination mechanism.

The constraint set of an NCSP is partitioned into a set of *constraint clusters* according to constraint type (e.g. adherence constraint, exclusion-around constraint etc, see figure 2.1) and constraint connectivity (see Figure 2.2). In the Figures, X's denote the variables and V's their values. Two constraints are *connected* when they have constrained variables in common. A constraint cluster is a set of con-

JyiShane Liu and Katia P. Sycara are both with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

straints which are of the same constraint type and are connected to each other. A problem is decomposed into a set of subproblems, each of which is concerned with the satisfaction of constraints in a constraint cluster. A solution to a constraint cluster subproblem is an instantiation of the variables in the constraint cluster such that none of the constraints in the constraint cluster are violated. When all subproblems are solved, a solution of the NCSP is found. For detailed description of CP&CR see [8].

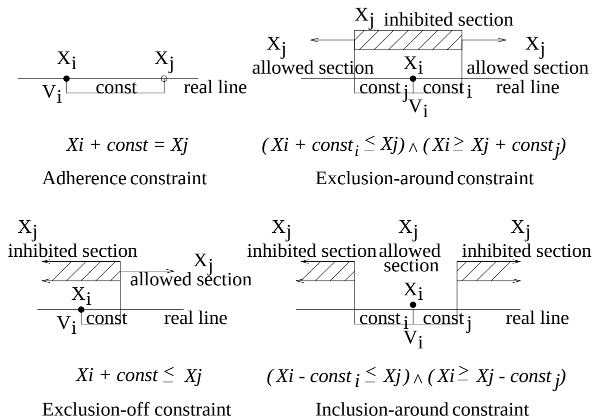

Figure 2.1 Constraint Types Classification : Xi instantiated to a value Vi

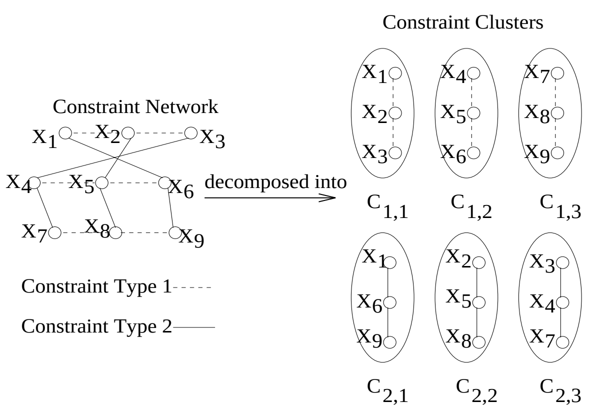

Figure 2.2 Constraint Partition

A group of agents, each corresponding to a constraint cluster, work as a team to solve an NCSP. All variables are instantiated to a set of initial values. The agents then asynchronously and collectively modify the instantiation and transform it into a solution. Each agent is responsible for variables in its variable set in the sense that the agent constantly checks and changes, when necessary, the values of the variables to make sure the constraints under its jurisdiction are not violated. A variable constrained by more than one type of constraint is under the jurisdiction of more than one agent. Agents responsible for the same variable have the same authority on its value, i.e. they can independently change the value as they need to. Therefore, a given value of a given variable is part of a solution, if it is *agreed* upon by all its responsible agents in the sense that no agent seeks to further change it.

III. Coordination Strategies and Group Search

In a coordinated group of agents, individual behavior is regulated by rules so that the agents' collective actions achieve the common goals. Communication facilitates coordinated group behavior. Given the tasks of solving complex, large-scale NCSPs, our coordination mechanism emphasizes convergence efficiency by exploiting characteristics of agent group structure, agent tasks, and informative communication. We have developed coordination strategies that promote rapid convergence by considering the following principles of interaction control.

1. *Least Disturbance* - When an agent is resolving conflicts, interactions should be initiated only when necessary and, in such a way as to reduce the chances of causing other concerned agents to subsequently initiate further interaction.
2. *Island of Reliability* - A group of problem-solving agents should reach consensus by a process of evolving coherent group decision-making based on *islands of reliability*, and modifying *islands of reliability* only when group coherence is perceived as infeasible under current assumptions.
3. *Loop Prevention* - A group of agents should prevent looping behaviors, such as oscillatory value changes by a subset of agents.

An *island of reliability* is a subset of variables whose consistent instantiated values are regarded as more likely than others to be part of the solution. In other words, a consistent instantiation of these variables is considered reliable. In particular, islands of reliability should correspond to the most critical constraint clusters, i.e. clusters whose variables have the least flexibility in satisfying their constraints. A variable which is a member of an island of reliability is called a *seed variable*. A variable which is not a seed variable is a *regular variable*.

Depending on the types of variables it governs, each agent in the population of agents falls into one of three categories. *Dominant agents* are responsible only for seed variables and therefore, are in charge of making decisions within islands of reliability. *Intermediate agents* have variable sets including both seed variables and regular variables. *Submissive agents* are responsible for only regular variables. Intermediate agents interact with submissive agents in a group effort to evolve an instantiation of regular variables compatible with the decisions of dominant agents regarding seed variables. An intermediate agent may override dominant agents' decisions by changing values of its seed variables to be compatible with its regular variables under only two conditions. First, the intermediate agent is not able to find a feasible instantiation on its regular variables compatible with the values of its seed variables (assigned by dominant agents). Secondly, the group effort, gauged by a threshold mechanism, of evolving a compatible instantiation on regular variables is perceived as having failed. When values of seed variables are changed by intermediate agents, dominant agents modify the values on the seed variables so as to adapt to the changes. The group of intermediate and submissive agents, then, are trying again to find a compatible instantiation on regular variables with the current values of seed variables.

Because of the principle of least disturbance, the infeasibility of a group effort in finding a compatible instantiation with dominant agents' decisions is typically manifested by cyclic value changes within a subset of intermediate and submissive agents. To detect such a loop requires complicated communication between agents which may not be justified by computational efficiency. Therefore, a threshold mechanism is provided for gauging search effort and preventing endless loops. Cyclic behavior may also occur when a dominant agent changes values of its seed variables to previously assigned values in response to constraint violations with values of regular variables. To prevent such looping behavior, a dominant agent keeps a history of value changes so it does not repeat the same variable instantiations.

Coordination information among agents is associated with each variable by its responsible agents. When an agent is resolving constraint violations on a variable under its responsibility, the coordination information provided by the other agents that govern the same variable is used in decision-making. Two types of information are exchanged between agents. *Disturbance information* reveals how the variable is constrained by an agent regarding potential value changes. *Dominance information* indicates special status of seed variables and measures the search effort of intermediate and submissive agents between each modification on islands of reliability. Search efforts are estimated in the form of the number of times the value of each regular variable is changed by a submissive agent.

The coordination mechanism, embedded in the roles of agents, principles governing agent interactions, and exchanged coordination information, facilitate efficient transformation from initial disorder into group coherence. From the point of view of search, the collective problem solving process is a coordinated, localized search with partially overlapping local search spaces (the values of variables that are the common responsibility of more than one agents). The process starts from an initial instantiation of all variables. The search behavior of the group of agents is guided by knowledge of the interaction structure and, static and emergent critical areas in the search space. Islands of reliability provide a means of anchoring the search, thus providing long term stability of partial solutions. The principle of least disturbance provides short term opportunistic search guidance. Search space is explored based on local feedback. The search ends when a solution is found, or when dominant agents have exhausted all possible instantiation of the seed variables.

IV. Experimental Results

Our domain of application of this methodology is job shop scheduling, one of the most difficult constraint satisfaction problems. Job shop scheduling deals with allocating a limited set of resources to a number of activities (operations) associated with a set of orders (jobs). Job shop scheduling is a well-known NP-complete problem [3]. CP&CR views each activity as a *variable*. A variable's *value* corresponds to a reservation for an activity. A reservation consists of a start time and the set of resources needed by the activity. The dominant constraints in job shop scheduling are *temporal activity precedence* and *resource capacity* constraints. The temporal precedence constraints along with a job's release date, due date and activity durations restrict the set of acceptable start times for each activity. The capacity constraints restrict the number of activities that can use a resource at any particular point in time and create conflicts among activities that are competing for the use of the same resource at overlapping time intervals. The goal of a scheduling system is to produce schedules that respect the problem constraints, i.e. release and due dates, as well as temporal relations and resource capacity constraints.

CP&CR has been implemented in a system, called CORA (COordinated Reactive Agents), for job shop scheduling problems. We experimentally (1) compared CORA's capabilities to other constraint satisfaction methods on the suit of benchmark constraint satisfaction scheduling problems proposed in [12], and (2) investigated CORA's scaling up characteristics on large constraint satisfaction scheduling problems.

The benchmark consists of 6 groups of 10 problems, each of which has 10 jobs of 5 activities and 5 resources. Each group of problems represents different scheduling conditions. Each problem is viewed as having 50 variables. For a more detailed description of the benchmark see [12]. CORA was compared to three other heuristic search scheduling techniques which had reported performances on the benchmark problem suite.

Micro-Opportunistic Search (MICRO OPP) [12] incrementally constructs a solution by traditional backtracking search guided by specialized variable and value ordering heuristics. MICRO OPP was run with two benchmark configurations - CHRON BKTRK, chronological backtracking and INTEL BKTRK, intelligent backtracking [15]. Min-Conflict Iterative Repair (MIN-CONF) [9] starts with an initial, inconsistent assignment for all variables and searches through the space of possible repairs. The search is guided by a *min-conflicts* value-ordering heuristic which attempts to minimize the number of constraint violations after each step. Conflict Partition Scheduling (CPS) [10] employs a search space analysis methodology based on stochastic simulation which iteratively prunes the search space by posting additional constraints. The procedure is repeated if the resulting network is inconsistent.

For the job shop scheduling problems in the benchmark, CORA views each order as a constraint cluster of *exclusion-off* constraints and each resource as a constraint cluster of *exclusion-around* constraints. Bottleneck resources, corresponding to the most contended resources in job shop scheduling, are the most critical constraint clusters and therefore, are regarded as islands of reliability and associated with dominant agents. Each order is governed by an intermediate agent and each non-bottleneck resource is assigned to a submissive agent. An instantiation of all variables is initiated by resource agents considering their capacity constraints. Subsequently, order agents and resource agents engage in an evolutionary process of reacting to constraint violations and making changes to the current instantiation. A solution emerges when none of the agents detect any constraint violation.

	CORA	CPS	MIN-CONF	MICRO OPP	
				CHRON BKTRK	INTEL BKTRK
w/1	10	10	9.8	10	10
w/2	10	10	2.2	10	10
n/1	10	10	7.4	8	10
n/2	10	10	1	9	10
0/1	10	10	4.2	7	10
0/2	10	10	0	8	10
Total	60	60	24.6	52	60
AVG. CPU time	4.8 seconds	78.43 seconds	298.42 seconds	234.72 seconds	128.78 seconds

Table 4.1
Comparative results: number of problems solved and average CPU time

Table 4.1 reports the number of problems solved and the average CPU time spent over all the benchmark problems for each technique (implementations are in Common Lisp on a DEC 5000/200). Each row represents a different benchmark category. For example, n/2 represents the category of 10 problems, each with narrow spread and two bettlenecks. Although CORA can operate asynchronously, it was sequentially implemented for fair comparison. We did not implement the other techniques to which CORA was compared ourselves but obtained the results from published reports of the developers of the techniques. To wit, the results for CPS and MIN-CONF were obtained from [10], the results for MICRO-OPP (CHRON-BKTRK) were obtained from [12] and the results for MICRO-OPP (INTEL-BKTRK) from [15]. CORA outperforms all the other techniques. In addition, the same problem generator function producing the benchmark problems was used to produce problem sets of 250 variables and 500 variables.

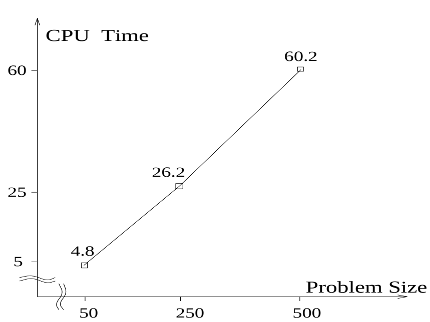

Figure 4.1 Scaling-up performace of CORA

Figure 4.1 shows CORA's performance on these larger sized problems, which exhibits favorable, near-linear scaling-up characterisitcs. As opposed to global search approaches, in which a single search is performed on the whole search space and search knowledge is obtained by analyzing the whole space at each step, CP&CR provides a framework to extract search knowledge and embed it in coordination between asynchronous local searches. The experimental results attest to the efficiency of CP&CR.

V. Conclusion

We show that a large-scaled NCSP can be decomposed and assigned to different problem solving agents according to discrete functionality (constraint type) and overlapping responsibility (variable instantiation). This decomposition results in interaction characteristics which can be exploited as sources of cheap and effective search knowledge. The coordination mechanism incorporates search knowledge and guides the search space exploration by the society of interacting agents, facilitating rapid convergence to a solution. In addition, the search knowledge is independent of the problem size and, therefore, the search complexity grows only linearly in problem size. We are currently applying the CP&CR methodology to Constraint Optimization Problems (COPs). Preliminary experiments show encouraging results compared to both direct search and simulated-annealing-based techniques.

References

[1] Rodney A. Brooks. "Intelligence Without Representation", *Artificial Intelligence*, Vol. 47, 1991, pp. 139-159.
[2] J. Ferber & E. Jacopin. "The Framework of Eco Problem Solving", in *Decentralized AI 2*, Demazeau & Muller (Ed.), Elsevier, North-Holland, 1991.
[3] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Co., 1979.
[4] Hiroaki Kitano. "Chanllenges of Massive Parallelism", *Proceedings of IJCAI-93*, 1993, pp. 813-834.
[5] Christopher G. Langton (Ed.). *Artificial Life*, Addison-Wesley, 1989.
[6] D. B. Lenat and E. A. Feigenbaum. "On the threshold of knowledge", *Artificial Intelligence*, Vol. 47, 1991, pp. 185-250.
[7] Olivier Lhomme. "Consistency Techniques for Numerical CSPs", *Proceedings of IJCAI-93*, 1993, pp. 232-238.
[8] JyiShane Liu and Katia P. Sycara. "Constraint Partition and Coordinated Reaction (CP&CR) - A Framework for Distributed Problem Solving in Constraint Satisfaction Problems", Tech. report CMU-RI-94, Forthcoming, Robotics Institute, School of Comoputer Science, Carnegie-Mellon University, 1994.
[9] S. Minton, M. D. Johnston, A. B. Philips, P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems", *Artificial Intelligence*, Vol. 58, 1992, pp. 161-205.
[10] Nicola Muscettola. "HSTS: Integrated Planning and Scheduling", in *Knowledge-Based Scheduling*, Mark Fox and Monte Zweben (Ed.), Morgan Kaufmann, 1993.
[11] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984.
[12] Norman Sadeh. "Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling", Tech. report CMU-CS-91-102, School of Computer Science, Carnegie-Mellon University, 1991.
[13] Luc Steels. "Towards a Theory of Emergent Functionality", *Proc. First Int. Conf. on Simulation of Adaptive Behavior*, MIT Press, 1990, pp. 451-461.
[14] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*, MIT Press, 1988.
[15] Yalin Xiong, Norman Sadeh, and Katia Sycara. "Intelligent Backtracking Techniques for Job Shop Scheduling", *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992, pp. 14-23.