Efficiency of Emergent Constraint Satisfaction in Small-world and Random Agent Networks

Xiaolong Jin and Jiming Liu
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
{jxl,jiming}@comp.hkbu.edu.hk

Abstract

In this paper, we study the efficiency of emergent constraint satisfaction in small-world and random agent networks. We find that emergent constraint satisfaction in a small-world network is less efficient than in some other networks (e.g., regular networks). Further, we find that this finding holds in almost all random networks. Based on these observations, we study the relationship between the efficiency of emergent constraint satisfaction and the randomness of the corresponding network from which the constraint satisfaction problem is generated.

1. Introduction

In recent years, the study of small-world phenomena has attracted much attention. In this paper, we will address the problem of emergent constraint satisfaction in the network of small-world and random topologies. Many natural and man-made systems exist in the form of networks. Researchers have found that many such networks exhibit a small-world topology, which means their nodes are highly clustered, but the shortest distance between any two nodes is relatively small. Some examples of such networks are human society [12], food webs [13], and World Wide Web [1].

In [8, 9, 11], Latora and Marchiori introduced the notion of *efficiency* for network. Based on this notion, they argued that the communication in a network of a small-world topology is efficient both globally and locally. They defined ϵ_{ij} as the communication efficiency between two vertices v_i and v_j. They set $\epsilon_{ij} = \frac{1}{d_{ij}}$, where d_{ij} is the shortest length between v_i and v_j. Based on ϵ_{ij}, they defined the communication efficiency of network G as follows:

$$E_{glob} = E(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \epsilon_{ij} = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}} \quad (1)$$

Assume G_i is a sub-network of G composed of the neighbors of vertex v_i and the edges among them. They defined the local communication efficiency of G as follows:

$$E_{loc} = \frac{1}{N} \sum_{i \in G} E(G_i) \quad (2)$$

Latora and Marchiori found that both E_{glob} and E_{loc} in a network of a small-world topology are larger than those in networks of other topologies. According to this finding, they defined a network with large E_{glob} and E_{loc} as a small-world network.

In [2, 3], Aguirre and Huerta conducted three experiments based on networks of different topologies. In their experiments, they respectively designed different dynamics to deliver reproducible or non-reproducible commodities from a source vertex to a target vertex or all other vertices, or from all vertices to a randomly selected target vertex. Their goal is to maximize an objective function that takes the *efficiency*, *reliability*, and *construction cost* of networks into consideration. They found that in all experiments, the network of a small-world topology maximizes the objective function.

All the above findings show that information communication in a network of a small-world topology is the most efficient among all kinds of networks. In [7], however, we observed a different phenomenon, i.e., in the small-world network, the efficiency of emergent constraint satisfaction is less efficient than in some other networks (e.g., regular networks). In [7], when using a multi-agent system to solve a satisfiability problem (SAT), if based on a clause-based agent representation [1], the resulting agent network shows a small-world topology. On the other hand, if based on a

[1] Each agent represents a clause in a given SAT problem and is responsible for making the clause true.

variable-based agent representation [2], the resulting agent network does not show a small-world topology. Further, we found that with the former representation, it is harder to solve a given SAT problem than with the latter representation. For more details, readers are referred to [7].

1.1 Problem statement

In [7], we observed emergent constraint satisfaction in networks of two special topologies. In this paper, we will:

1. validate the above observations and further systematically examine the efficiency of emergent constraint satisfaction in networks with different topologies, including regular networks, small-world networks, and random networks;
2. further show that the randomness of a network is a determining factor to the efficiency of emergent constraint satisfaction.

We will perform the above studies as follows:

1. generate networks of different topologies;
2. based on the networks, randomly generate some constraint satisfaction problems;
3. through running a local search procedure on the generated constraint satisfaction problems, examine the efficiency of solving constraint satisfaction problems.

1.2 Organization of the Paper

The remainder of this paper is organized as follows. In Section 2, we introduce the notions of agent network and emergent constraint satisfaction. We provide a formal definition of small-world topology as well as introduce Watts and Strogatzs' method for generating small-world networks in Section 3. Section 4 presents a constraint satisfaction problem generator. Section 5 shows our experiments and discusses the experimental results. In Section 6, we compare our results with those of Walsh. Section 7 presents future work. Section 8 concludes the paper.

2. Agent Networks and Emergent Constraint Satisfaction

When using a multi-agent based approach to solve a problem, agents will implicitly or explicitly form relationships of cooperations or coordinations among them [5, 6, 10].

[2] Each agent represents one or several variables in a given SAT problem and is responsible for assigning values to these variables so as to make clauses related to them true.

Definition 1 (Agent Network)
An agent network is a virtual network where nodes are agents and edges are relationships among them.

Definition 2 (The Topology of an Agent Network)
The topology of an agent network is its geometric property, which reflects the connectivity of the agent network.

Generally speaking, in a multi-agent system for problem solving, there are constraints among agents [5, 6, 10]. For example, in a variable-based agent representation of a SAT problem (See [7]), if variables represented by two agents appear in a common clause, agents have to cooperatively assign values to these variables so as to make the clause satisfied. Here, the clause is a constraint between these agents. In a sense, problem-solving is a process where agents cooperate or coordinate to satisfy constraints among them. During the process, all agents behave according to their strategies (e.g., $best-move$,$better-move$, and $random-move$ in [6]). At a certain state, if all constraints are satisfied, we can say that constraint satisfaction emerges and the problem is solved. Fig. 1 gives an example of emergent constraint satisfaction.

3. Small-World Topology

Milgram first proposed the notion of small-world in 1967 [12]. Later, Watts and Strogatz mathematically formulated it with two measurements, *characteristic path length* and *clustering coefficient* [16]. Given a network $G = \langle V, E\rangle$, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of vertices and $E = \{e_1, e_2, \cdots, e_m\}$ is the set of edges, they defined the characteristic path length, L, and clustering coefficient, C, as follows:

- *Characteristic path length L:*

$$L = \frac{2}{n \cdot (n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j} \tag{3}$$

  where $d_{i,j}$ is the shortest distance between vertices v_i and v_j.

- *Clustering coefficient C:*

$$C = \frac{1}{n} \sum_{i=1}^{n} c_{v_i} \tag{4}$$

  where c_{v_i} is called *clustering ratio* (also called *clustering* in short) of vertex v_i. Let $d(v_i)$ be the degree of v_i (i.e., the number of neighboring vertices of v_i) and $f(v_i)$ be the number of existing edges among the neighbors of vertex v_i, c_{v_i} can be defined as follows:

$$c_{v_i} = \frac{f(v_i)}{\frac{(d(v_i)-1)\cdot d(v_i)}{2}} = \frac{2 \cdot f(v_i)}{(d(v_i)-1)\cdot d(v_i)} \tag{5}$$

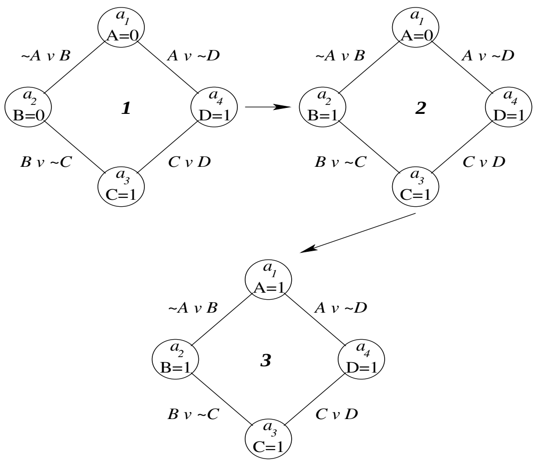

Figure 1. An example of emergent constraint satisfaction. In this example, agents a_1, a_2, a_3, and a_4 represent variables A, B, C, and D, respectively. There are four constraints among them, namely, clauses $\neg A \vee B$, $B \vee \neg C$, $C \vee D$, and $A \vee \neg D$. At the beginning, i.e., $state{-}1$, four variables are randomly assigned values $A = 0, B = 0, C = 1$, and $D = 1$ by corresponding agents. After that, the multi-agent system behaves to satisfy all constraints. Agents find that two constraints, i.e., $B \vee \neg C$ and $A \vee \neg D$, are not satisfied. In this case, agents will try to satisfy the unsatisfied constraints through changing values of their variables. Without lose of generality, we assume the following is a possible process of satisfying all constraints: Based on a certain cooperation mechanism, agent a_2 flips the value of its variable B from 0 to 1. Hence, the system changes to $state{-}2$, and the constraint $\neg A \vee B$ becomes satisfied. In $state{-}2$, constraint $A \vee \neg D$ still remains unsatisfied. Agent a_1 finds this fact and flips the value of its variable A from 0 to 1. Then, the system changes to $state{-}3$. In this state, all constraints among agents are satisfied. Constraint satisfaction emerges.

Generally speaking, L represents the global connectivity of G. On the other hand, C represents the average local connectivity of sub-networks in G. Based on these two measurements, Watts and Strogatz defined network G as a small-world topology if and only if:

$$L \approx L_{random} \quad \texttt{and} \quad C \gg C_{random} \tag{6}$$

where L_{random} and C_{random} are the average characteristic path length and clustering coefficient of random networks with the same size as G, respectively.

Watts and Strogatz further presented a method for generating a small-world network [16] by rewiring edges of a regular lattice with a probability, p. If $p = 0$, the lattice is still regular. If $p = 1$, the lattice becomes a completely random network. They found that the small-world network occurs in the region where $p \in (0, 1)$. The specific algorithm is shown in Fig. 2, where k is the average degree of vertices.

To show the above phenomenon, we sampled several values for probability $p \in [0, 1]$. For each p, we generated 10 networks with $n = 1000$ vertices and an average degree $k = 10$ using Watts and Strogatzs' method. Next, we calculated the average $L(p)$ and $C(p)$ and normalized them with $L(0)$ and $C(0)$, respectively. The results are shown in Fig. 3. The solid curve denotes $\frac{C(p)/C(0)}{L(p)/L(0)}$. We call it *normalized ratio of local and global connectivities*, R_{CL}. We note that as p increases from 0 to 1, the value of R_{CL} increases first and then begins to decrease. When $p \in (0.01, 0.1)$, its value reaches the maximum. Later on, we will call $(0.01, 0.1)$ as "small-world region". Our other experiments have shown that the larger the R_{CL} of a network, the more "small-worldy" the network (i.e., there are more clusters in the network).

4. A Network-Based Constraint Satisfaction Problem (CSP) Generator

In this section, we present a method for generating a constraint satisfaction problem from a network. Given a network, $G = \langle V, E \rangle$, we can generate a constraint satisfaction problem with $|V|$ variables and $|E|$ constraints as follows:

1. With number $i \in \{1, \cdots, |V|\}$ label all vertices in network G;
2. Uniformly select a value, $u_{v_i} \in \{0, 1\}$, for each vertex v_i in network G;
3. Let constraint set $C = \{\}$;
4. For each edge $\langle v_i, v_j \rangle$ in E,
   - if vertices v_i and v_j have the same value, i.e., $u_{v_i} = u_{v_j}$, add a constraint $v_i = v_j$ to C;

Label vertices with $1, \cdots, n$ in clockwise sequence;
For $j = 1$ **to** $k/2$
  For $i = 1$ **to** n
    Consider edge $e(v_i, v_{i_j})$ between vertex v_i and its j^{th} nearest neighbor in a clockwise sense;
    If $e(v_i, v_{i_j})$ exists
      While $True$
        With probability p, uniformly select a vertex v_h;
        If $v_h \neq v_i$ **Then Break**;
      End While
      If $e(v_i, v_h)$ does not exist
        Delete edge $e(v_i, v_{i_j})$;
        Add a new edge, $e(v_i, v_h)$;
      End If
    End If
  End For
End For

Figure 2. The algorithm to generate a small-world network by rewiring edges of a regular lattice with a certain probability, p.

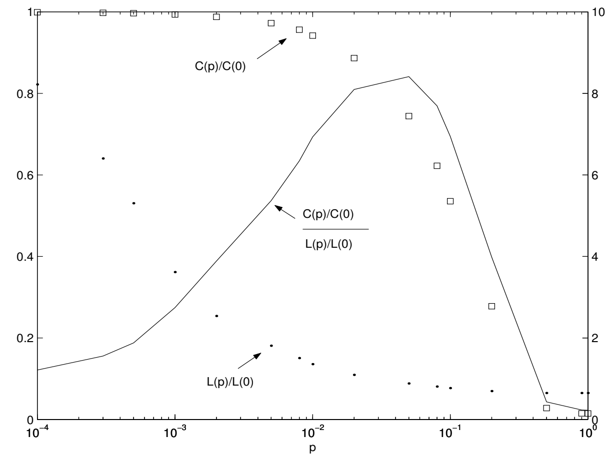

Figure 3. Characteristic path length L and clustering coefficient C of networks generated by Watts and Strogatzs' method with different probabilities p. L and C of all networks are normalized by those of network with probability $p = 0$, respectively. All networks are generated from a regular lattice with $n = 1000$ vertices and an average degree $k = 10$. The solid curve denotes R_{CL}, the right side y-axis is its values.

- if vertices v_i and v_j have the different values, i.e., $u_{v_i} \neq u_{v_j}$, add a constraint $v_i \neq v_j$ to C;

5. Return C as a generated constraint satisfaction problem based on network G, where the domains of all variables are $\{0, 1\}$.

Note that each generated constraint satisfaction problem is satisfiable. Obviously, the value combination of vertices used in the generation of this problem is a solution.

5. Experiments and Results

In this section, we experimentally exam the efficiency of emergent constraint satisfaction in networks of different topologies. All constraint satisfaction problems used in our experiments are generated from four group of networks. These networks are generated from four regular lattices: Lattice 1 ($n = 100$, $k = 4$), Lattice 2 ($n = 200$, $k = 4$), Lattice 3 ($n = 400$, $k = 6$), and Lattice 4 ($n = 500$, $k = 8$). We have found that with the same probability p, different lattices will generate networks with different small-worldinesses (See Fig. 4). This is the reason that we conduct experiments on networks generated from different lattices.

Specifically, in each experiment, we perform the following steps:

1. With each lattice, generate a group of networks with probabilities p sampled in $(0, 1)$;
2. Based on each network, generate 10 constraint satisfaction problems;
3. On each constraint satisfaction problem, run the following *greedy local search* [4, 14] procedure and record or calculate the related data:
   - **Greedy step:** Select a vertex, v_i, the changing of whose value will cause the maximum increment in the number of satisfied constraints[3]. Next, change the value of v_i;
   - **Random step:** If the above v_i does not exist, randomly select a vertex v_j and change its value.

Experiment 1 *In this experiment, we run the greedy local search procedure a fixed number of steps on each generated constraint satisfaction problem, and calculate the ratio of satisfied constraints to all constraints. The results are presented in Fig. 5.*

Fig. 5 shows that as probability p increases, the ratio of satisfied constraints to all constraints decreases. When p enters the small-world region, the ratio decreases more and

[3] If there are more than one such kind of vertices, randomly select one.

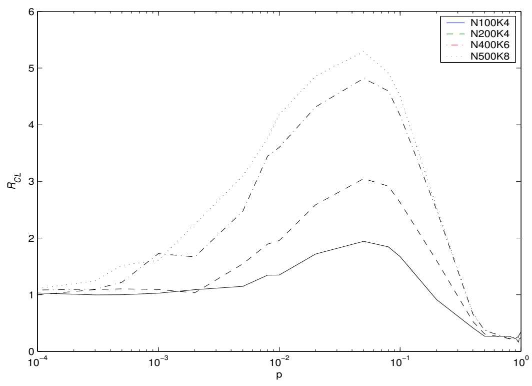

Figure 4. R_{CL} curves of networks generated from four regular lattices, namely, Lattice 1 ($n = 100$, $k = 4$), Lattice 2 ($n = 200$, $k = 4$), Lattice 3 ($n = 400$, $k = 6$), and Lattice 4 ($n = 500$, $k = 8$) using Watts and Strogatzs' method.

more quickly. That means the network of a small-world topology hardens the greedy local search procedure to satisfy more constraints. Even if p increases beyond the small-world region, the ratio keeps decreasing until $p > 0.5$ after which the ratio exhibits purterbations.

At the beginning of running the greedy local search procedure on a constraint satisfaction problem, the procedure can always find a greedy step to perform. After successively performing a certain number of greedy steps, the local search procedure will not always find a greedy step. In this case, it has to perform random steps. We call the total steps before the greedy local search procedure performs the eleventh random step as a *smooth increase stage*.

Experiment 2 *In this experiment, we run the greedy local search procedure on each constraint satisfaction problem. We record the number of satisfied constraints and the total number of steps performed at the smooth increase stage. Next, we calculate the number of satisfied constraints per step. The results are presented in Fig. 6.*

Fig. 6 shows that as probability p increases, the number of satisfied constraints per step decreases. The decrease speed becomes faster after p enters the small-world region. For different network groups, the decrease speeds are obviously different. The order of decrease speeds is the same as that of small-worldinesses of the networks corresponding to four network groups.

Experiment 3 *In this experiment, we run the greedy local search procedure on each constraint satisfaction problem and stop it when more than 85% constraints are satisfied, and then calculate the total number of steps. The results are shown in Fig. 7.*

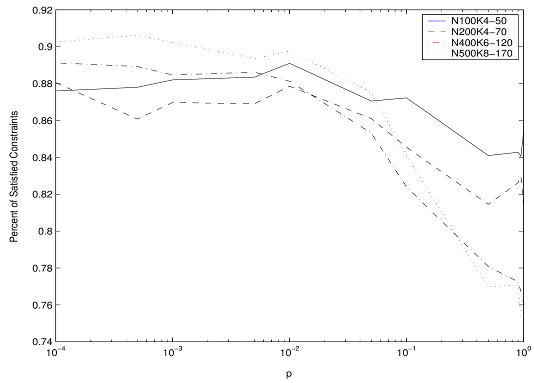

Figure 5. Ratios of satisfied constraints to all constraints after the greedy local search procedure runs on the networks a fixed number of steps. The fixed number for each network group is set as 50, 70, 120, and 170, respectively.

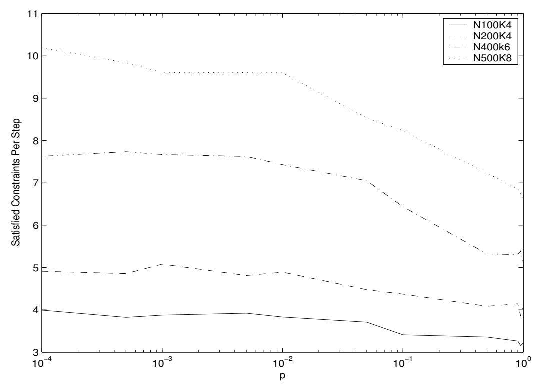

Figure 6. The number of satisfied constraints per step at the smooth increase stage of each network group.

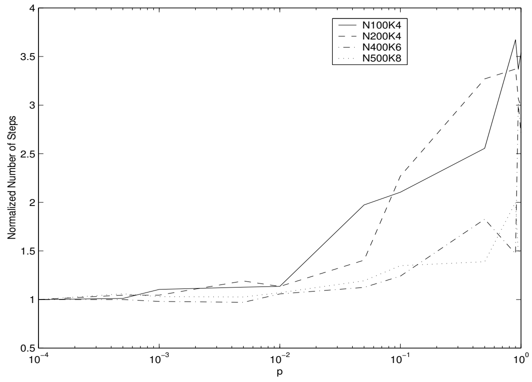

Figure 7. The number of steps that the greedy local search procedure takes before at least 85% constraints are satisfied, which is normalized by that of a problem generated from a network with $p = 0$.

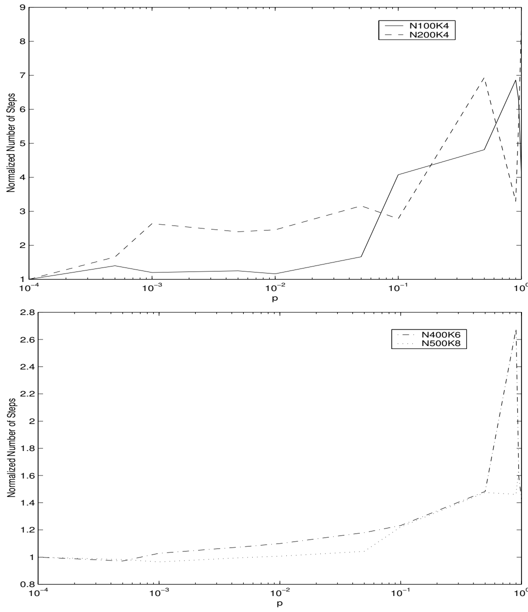

Figure 8. The number of steps that the greedy local search procedure takes before at least 90% constraints are satisfied, which is normalized by that of a problem generated from a network with $p = 0$.

Experiment 4 *This experiment is the same as the previous experiment except that we record how many steps are taken before at least* 90% *constraints are satisfied. The results are shown in Fig. 8.*

Figs. 7 and 8 show that the numbers of steps to satisfy a certain number of constraints almost monotonically increase, as probability p increases from 0 to 0.5. That means, as networks become more small-worldy, the generated constraint satisfaction problems become harder to be solved. When $p > 0.5$, the hardness of the problems become unpredictable.

From the above experimental results, we note that the network of a small-world badly influences the above four empirical measurements. A network of a small-world topology decreases the efficiency of solving a constraint satisfaction problem. In the region $0.1 < p < 0.5$ where networks no longer show obvious small-world topologies, the empirical measurements keep the trends in the small-world region. When $p > 0.5$ where networks become almost purely random, the empirical measurements exhibit perturbations.

Based on the above observations, we can assert that the 'randomness' [4] of a network is a key factor influencing the efficiency of solving the generated constraint satisfaction problems. As the randomness of networks increase, the efficiency will decrease gradually. When the randomness exceeds a threshold $p = 0.5$, the efficiency becomes completely unpredictable.

[4] In this paper, we can regard probability p as an index that reflects the randomness of networks.

6. Comparison with Walsh's Work

In [15], Walsh found that many search problems, i.e., *graph coloring*, *time tabling*, and *quasi-group problems*, involve networks of small-world topologies. Walsh experimentally proved that in search problems the network of a small-world topology computationally hardens the search process.

Walsh observed that when p increases beyond the small-world region and approaches to 1, the search cost decreases to be similar to the case where p has not entered the small-world region. This is entirely different from our above observations. Another key difference between Walsh's work and ours is that Walsh studied the relationship between the topology and the corresponding computational cost of a problem[15], but we addressed that between the topology of a multi-agent system used to solving a problem and the computational cost.

7. Future Work

This paper shows the preliminary results of our research. We can study further in the following aspects:

1. The current constraint satisfaction problem generator can only generate binary constraints with binary vari-

ables. In fact, this model can readily be extended to generate multi-variable and multi-value constraint satisfaction problems;

2. In our presented experiments, we employed the simplest local search procedure, i.e., the greedy local search, to solve problems. We may employ other strategies used in local search, such as, random walk, tabu list, to test the efficiency of these strategies. Therefore, we can better understand the roles of these strategies in guiding a search process;

3. The greedy local search procedure used in the paper is sequential and centralized. We may also employ distributed approaches, such as, MASSAT[6] and Multi-DB[5], to examine emergent constraint satisfaction in networks with different topologies;

4. In this paper, we presented four measurements to check the efficiency of constraint satisfaction in regular, small-world, and random networks. Although these measurements have, to some extent, validated our expected findings, they are in essence empirical measurements. They are not complete enough to reflect the essence of why the efficiency of constraint satisfaction in a small-world or random network is hard. In our future work, we will find additional measurements.

8. Conclusion

Small-world topologies are widely observed in many natural and artificial systems. Researchers have found that information communication in a small-world network is very efficient. In this paper, we validated a different effect of a small-world network. We experimentally confirmed that:

- emergent constraint satisfaction in a small-world or random network is less efficient than in a regular network;

- emergent constraint satisfaction in a small-world network is less efficient than in a random network where $p \in (0, 0.01)$, but more efficient than a random network where $p \in (0.1, 0.5)$. Due to the perturbations, the efficiency of emergent constraint satisfaction in a random network where $p > 0.5$ is unpredictable.

In general, randomness is a key factor that influences the efficiency of emergent constraint satisfaction. The more random a network, the lower efficiency of emergent constraint satisfaction. This conclusion can guide designing a multi-agent based approach to solving a search problem, where we should avoid involving small-world and relatively random networks. This is the significance of the paper.

References

[1] L. A. Adamic. The small world Web. In *Proceedings of European Conference on Digital Libraries (ECDL'99)*, volume 1696 of *LNCS*, pages 443–452, 1999.

[2] C. Aguirre and R. Huerta. Small-world topology for multi-agent collaboration. In *DEXA Workshop 2000*, pages 231–235, 2000.

[3] C. Aguirre, J. M.-M. noz, and R. Huerta. A low cost/high performance schalable topology for multi-agent collaboration. In *Autonomos Agents 2000 Workshop: Agents In Industry*, 2000.

[4] J. Gu. Efficient local search for very large-scale satisfiability problem. *SIGART Bulletin*, 3:8–12, 1992.

[5] K. Hirayama and M. Yokoo. Local search for distributed SAT with complex local problems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pages 1199–1206, 2002.

[6] X. Jin and J. Liu. Multiagent SAT (MASSAT): Autonomous pattern search in constrained domains. In *Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'02)*, pages 318–328, 2002.

[7] X. Jin and J. Liu. Agent network topology and complexity. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*, 2003.

[8] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19), 2001.

[9] V. Latora and M. Marchiori. Economic small-world behavior in weighted networks. *European Physics Journal B*, 32:249, 2003.

[10] J. Liu, H. Jing, and Y. Tang. Multi-agent oriented constraint satisfaction. *Artificial Intelligence*, 136:101–144, 2002.

[11] M. Marchiori and V. Latora. Harmony in the small-world. *Physica A*, 285:539–546, 2000.

[12] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[13] J. M. Montoya and R. V. Solé. Small world patterns in food webs. *Journal of Theoretical Biology*, 214:405–412, 2002.

[14] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, pages 440–446, 1992.

[15] T. Walsh. Search in a small world. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'99)*, 1999.

[16] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.