Influential Variables in Constraint Networks

Qiancheng Li[*] **Tenda Okimoto**[*] **Katsutoshi Hirayama**[*]

1.Introduction

Constraint satisfaction problem (CSP) [8, 15] is a fundamental problem that can formalize various applications of Artificial Intelligence, e.g., facility location, planning, job shop scheduling, vehicle routing, and graph problems [2, 10, 11, 18]. CSP consists of a set of variables, finite, discrete domains for each variable, and a set of constraints. The objective of this problem is to find a consistent assignment of values to variables. CSP can be represented using a graph, called a constraint network, in which each node represents a variable and each edge corresponds to a constraint. A graph-coloring problem is one of the representative examples of CSPs. This problem can be represented as a constraint network [10], where the aim is to find an assignment of colors to the nodes such that no two adjacent nodes share the same color.

In a constraint network, some nodes have a significant influence on the global result, namely satisfiability and unsatisfiability. The existence of such variables, e.g., backbone [7, 13, 14] and influential variables [16], is well-known in Satisfiability problem (SAT) and CSPs. The SAT is one of the first problems that was proven to be NP-complete [3]. The objective of a SAT problem is to decide whether a given boolean formula has any satisfying truth assignment. The backbone is a variable which takes the same value for all satisfiable solutions in a SAT. In general, the SAT solver terminates when it finds the set of literals that are true in every satisfying truth assignment, or finds that there exists no such assignment. However, in order to identify a backbone variable, it is required to solve all satisfiable solutions of the problem. It is well-known that identifying the backbone variables in a SAT (and also CSP) is co-NP-complete problem [5].

The influential variable is a variable if it enables a given CSP satisfiable and also unsatisfiable. More specifically, an influential variable x has at least two values d and d' that lead to the phase transition, i.e., from satisfiable to unsatisfiable solutions and vice versa. For instance, in case the influential variable x takes the value d, one can find a satisfiable solution of the problem. However, when it takes the another value namely d', one cannot find a satisfiable solution of the problem whatever we choose the values for other variables. The concept of backbone and influential variables are very similar. The difference of these two variables is that the former has the same value in all satisfiable solutions in CSPs, while the latter is not necessary to have the same one. In the previous work [16], it is shown that the influential variable includes the backbone variable for $|D| \geq 2$, where D is a set of finite, discrete domains in a CSP. Finally, similar concepts such as backdoor[14], frozen pairs[4], and spines[1] have been frequently appeared in SAT and CSP literature, and they are widely investigated in research on phase transition phenomena [9, 12, 17].

There exists virtually no work on investigating the backbone and influential variables in different constraint network structures. In the previous works on CSPs, there exists several works on examining the effect of network structures [6, 19, 20]. For instance, Walsh showed several application domains of CSPs such as graph-coloring problems generated from register allocation problems, time-tabling, and quasi-group problems, actually have small-world like structures, and the cost of solving such problem instances can have a heavy-tailed distribution [19, 20]. Also, there exists several works focused on the backbone and influential variables in CSPs. However, as far as the authors are aware, there exists no work focused on identifying the existence of backbone and influential variables in different constraint network structures.

In this paper, the main focus is laid on the existence of backbone and influential variables in different constraint network structures. In the experiments, the graph coloring problems are generated as our CSP instances. Then, the number of backbone and influential variables in small-world and random constraint networks is investigated on the number of benchmarks. We empirically show that there exists less influential variables in small-world constraint networks compared to random constraint networks. [†]

2.Preliminaries

Constraint Satisfaction Problem

A surprisingly wide variety of Artificial Intelligence problems can be formalized as a constraint satisfaction problem (CSP) [8, 15]. CSP consists of m variables $x_1, x_2, ..., x_m$, whose values are taken from finite, discrete domains $D_1, D_2, ..., D_m$, respectively, and a set of constraints on their values. The aim of this problem is to find a consistent assignment of values to variables. A constraint is defined by a predicate. That is, the constraint $p(k; x_{k_1}, x_{k_2}, ..., x_{k_j})$ is a predicate that is defined on Cartesian product $D_{k_1} \times D_{k_2} \times ... \times D_{k_j}$. This predicate is true iff the value assignment of these variables satisfies this con-

---

[*] Graduate School of Maritime Sciences, Kobe University

[†] This article is the extended version of [16]

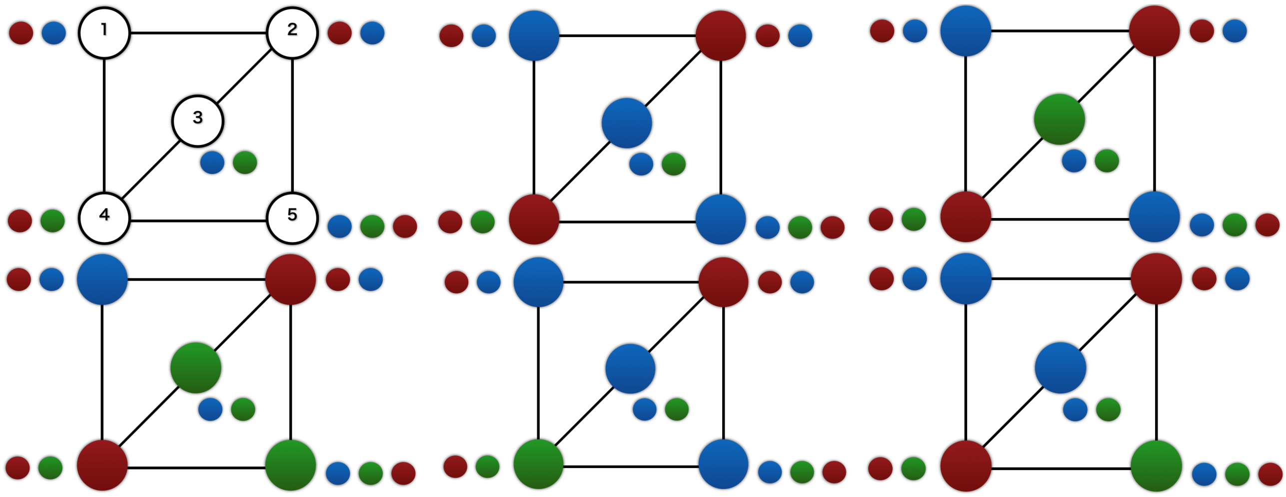


図 1: Graph coloring problem with five variables.

straint. CSP can be represented using a graph, called a constraint network, in which each node represents a variable and each edge corresponds to a constraint.

Example 1 (Graph-coloring [16]). Figure 1 shows a graph-coloring problem with five variables. The top-left, which is not colored, is the original problem. Each node represents a variable and each edge between nodes corresponds to a constraint. Each node takes its value from a discrete, finite domain, e.g., node 1 can take the value red or blue. The aim of this problem is to find an assignment of colors to the nodes such that no two adjacent nodes share the same color. The rests show all possible solutions for it.

Backbone variable

Backbone[7, 13, 14] is the set of literals which are true in every satisfying truth assignment in a SAT. In CSPs over finite domains, a variable is said to be a backbone variable if it takes the same value in all satisfiable solutions. Formally, for a given CSP $= \langle X, D, C \rangle$, where X is a set of variables, D is a set of domains, and C is a set of constraints. Now, let $\mathcal{S}^{sat}$ be a set of all satisfiable solutions. Also, let $s_{(x_i,d_i)} \in \mathcal{S}^{sat}$ be a solution which includes an assignment (x_i, d_i). In CSPs over finite domains, a variable $x_i \in X$ is said to be a backbone variable if it holds the following condition: For $d_i, d'_i \in D_i$ $(\subseteq D)$,

$$s_{(x_i,d_i)} \in \mathcal{S}^{sat} \wedge s_{(x_i,d'_i)} \in \mathcal{S}^{sat} \Rightarrow d_i = d'_i \qquad (1)$$

Example 1 (continued). Consider the same graph-coloring example in figure 1. Since node 1 and 2 have the same values in all satisfiable solutions, namely node 1 has the value blue and red for node 2, they are backbone variables. On the other hand, since node 3, 4 and 5 have two different assignments in the satisfiable solutions, these three variables are not backbone.

2.1.Influential variable

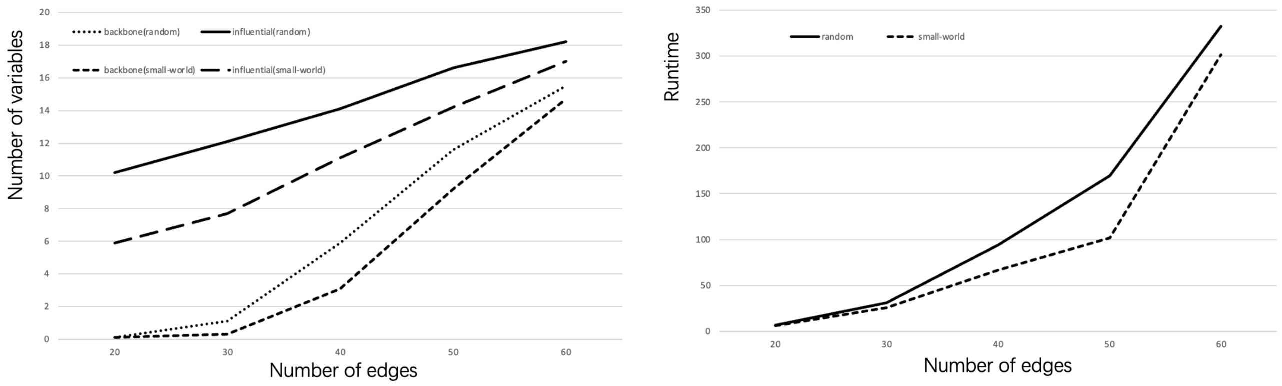
In CSPs, a variable is called an influential variable [16] if it enables a given CSP satisfiable and also unsatisfiable. More specifically, an influential variable has at least two values that lead to the phase transition, i.e., from satisfiable to unsatisfiable solutions and vice versa. Formally, for a given CSP $= \langle X, D, C \rangle$, let $\mathcal{S}^{sat}$ be a set of all satisfiable solutions and $\mathcal{S}^{unsat}$ be a set of other solutions. For a variable $x_j \in X$ and its domain D_j with $|D_j| \geq 2$, x_j is called an influential variable if it holds the following two conditions:

$$\exists d_j \in D_j : s_{(x_j,d_j)} \in \mathcal{S}^{sat} \qquad (2)$$

$$\exists d'_j \in D_j : s_{(x_j,d'_j)} \in \mathcal{S}^{unsat} \qquad (3)$$

Intuitively, an influential variable x_j has at least two values, and the decision of x_j (i.e. d_j or d'_j) leads to the phase transition, e.g., from the satisfiable solution to the unsatisfiable solution and vice versa. In case a variable x_j takes a value d_j, one can find a satisfiable solution. However, when a variable x_j takes another value d'_j, one cannot find a satisfiable solution whatever we choose the values for other variables.

Example 1 (continued). Consider the same example in figure 1. First, node 1 and 2 are influential variables. When node 1 and 2 take blue and red respectively, the equation (2) is hold. Also, it holds the equation (3). One can easily see that this problem becomes unsatisfiable, in case node 1 takes red and/or node 2 takes blue. Next, node 3 is not an influential variable, since the problem becomes always satisfiable whatever it chooses. That is, it does not hold the equation (3). Similarly, node 4 is not an influential variable. Finally, node 5 is an influential variable. When it takes blue, it holds the equation (2). In case it takes red, the equation (3) is also hold.



(a) Number of backbone and influential variables

(b) Runtime

図 2: The average number of influential and backbone variables for random and small-world constraint networks with 20 nodes, varying the number of edges from 20 to 60.

The concept of backbone and influential variables are very similar. The main difference between these two variables is that the former has the same value in all satisfiable solutions in a CSP, while the latter is not necessary to have the same one. In the previous work [16], it is showed that if a variable is the backbone, it is also the influential variable. However, an influential variable is not necessary to be a backbone.

3.Experiments

In the section, the number of backbone and influential variables in small-world constraint networks is investigated by using the graph coloring problems, and the results are compared with those in random constraint networks. The runtime of identifying backbone and influential variables is also reported.

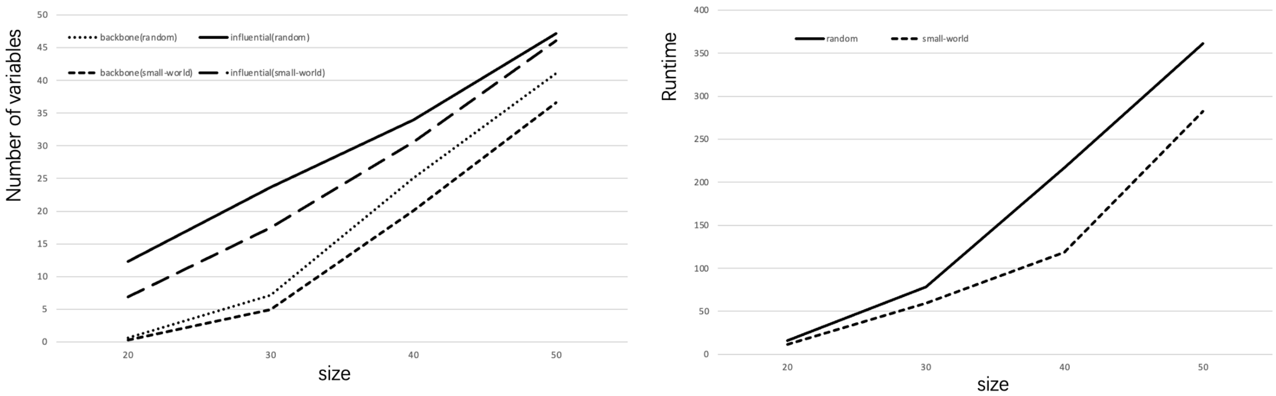
3.1.Experimental setting

The following problem instances are utilized: The random and small-world constraint networks are generated varying the number of nodes and edges. The domain size of each variable is randomly chosen from two to four. Each data point in the results represents the average value of 50 problem instances. First, we investigate the number of backbone and influential variables in constraint networks with 20 nodes, varying the number of edges. Next, we investigate the number of backbone and influential variables in constraint networks with the density 0.2, varying the number of nodes. The density is the constraint tightness of a problem instance that is provided by $|E| = \delta \times \frac{1}{2}|X|(|X|-1)$, where $|X|$ is the number of nodes and $|E|$ is the number of edges.

3.2.Results

Figure 2 represents the average number of influential and backbone variables for constraint networks with 20 nodes, varying the number of edges from 20 to 60. Note that this article is targeted on only satisfiable problem instances, otherwise, there exists neither backbone nor influential variables in unsatisfiable problems. The x axis shows the number of edges and the y axis represents the average number of backbone and influential variables. Compared to random constraint networks, there exists less backbone and influential variables in small-world constraint networks. The average number of influential variables in small-world constraint networks is 17.0 (and 14.7 for backbone) for 60 edges, while it is 18.2 (and 15.5 for backbone) in random constraint networks. Furthermore, the number of influential variables increases when the constraint networks become dense. For small-world constraint networks, the average number of influential variables is 5.9 (and 0.1 for backbone) for 20 edges, while it is 17.0 (and 14.7 for backbone) for 60 edges. We observed the similar results in random constraint networks. Finally, the average runtime for identifying influential variables in small-world constraint networks is smaller than random constraint networks.

Figure 3 shows the average number of influential and backbone variables for constraint networks with the density 0.2, varying the number of nodes from 20 to 50. We observed the similar results to those in Fig. 2, that is, compared to random constraint networks, there exists less backbone and influential variables in small-world constraint networks. The average number of influential variables in small-world constraint networks is 46.1 (and 36.6 for backbone) for 50 nodes, while it is 47.2 (and 41.1 for backbone) in random constraint networks. Furthermore, the number of backbone and influential variables increases when the constraint networks become larger. For small-world constraint networks, the average number of influential variables is 6.9 (and 0.3 for backbone) for 20 nodes, while it is 46.1 (and 36.6 for backbone) for 50 nodes. We observed the similar results in random



(a) Number of backbone and influential variables

(b) Runtime

図 3: The average number of influential and backbone variables for random and small-world constraint networks with the density 0.2, varying the number of nodes from 20 to 50.

constraint networks. Also, the average runtime for finding influential variables in small-world constraint networks is smaller than random constraint networks.

4.Conclusion

Constraint satisfaction problem (CSP) is a fundamental problem of Artificial Intelligence. This problem can be represented as a graph called a constraint network. In a constraint network, some nodes have a significant influence on the global result, namely satisfiability and unsatisfiability. In CSPs, such variable is called an influential variable. In this paper, we empirically investigated the number of influential variables in random and small-world constraint networks. The experimental results revealed that (i) compared to random constraint networks, there exists less backbone and influential variables in small-world constraint networks, and (ii) the number of backbone and influential variables increases when a constraint network becomes dense, and also it becomes larger.

As a perspective for further research, we will develop an efficient heuristic algorithm for finding an influential variable, that is specialized to small-world network structure. Also, we will intend to apply our work to some real-world problems, e.g, time-tabling, meeting scheduling and quasi-group problems.

参考文献

[1] B. Bollobás, C. Borgs, J. Chayes, J. Kim, and D. Wilson. The scaling window of the 2-sat transition. *Random Structure Algorithms*, 18(3):201–256, 2001.

[2] S. Brailsford, C. Potts, and B. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119:557–581, 1999.

[3] S. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158, 1971.

[4] J. Culberson and I. Gent. Frozen development in graph coloring. *Theoretical Computer Science*, 265(1-2):227–264, 2001.

[5] R. de Haan, I. Kanj, and S. Szeider. Small unsatisfiable subsets in constraint satisfaction. In *ICTAI*, pages 429–436, 2014.

[6] D. Devlin and B. OŚullivan. Preferential attachment in constraint networks. In *ICTAI*, pages 708–715, 2009.

[7] O. Dubois and G. Dequen. A backbone-search heuristic for efficient solving of hard 3-sat formulae. In *IJCAI*, pages 248–253, 2001.

[8] E. Freuder. Partial constraint satisfaction. In *IJCAI*, pages 278–283, 1989.

[9] C. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1/2):67–100, 2000.

[10] E. Hebrard and G. Katsirelos. Constraint and satisfiability reasoning for graph coloring. *Journal of Artificial Intelligence Research*, 69:33–65, 2020.

[11] H.Kautz and B.Selman. Planning as satisfiability. In *ECAI*, pages 359–363, 1992.

[12] B. Huberman and T. Hogg. Phase transitions in artificial intelligence systems. *Artificial Intelligence*, 33(2):155–171, 1987.

[13] M. Janota, I. Lynce, and J. Marques-Silva. Algorithms for computing backbones of propositional formulae. *AI Communications*, 28(2):161–177, 2015.

[14] P. Kilby, J. Slaney, S. Thiébaux, and T. Walsh. Backbones and backdoors in satisfiability. In *AAAI*, pages 1368–1373, 2005.

[15] A. Mackworth. Constraint satisfaction. In *Encyclopedia of Artificial Intelligence*, pages 285–293, 1992.

[16] T. Okimoto and K. Hirayama. Identifying influential variables in csp. In *ISIS*, pages 320–325, 2019.

[17] A. Parkes. Clustering at the phase transition. In *AAAI*, pages 340–345, 1997.

[18] P. van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 58:297–326, 1992.

[19] T. Walsh. Search in a small world. In *IJCAI*, pages 1172–1177, 1999.

[20] T. Walsh. Search on high degree graphs. In *IJCAI*, pages 266–274, 2001.