

SOLVING CSP BY LAGRANGIAN METHOD WITH IMPORTANCE OF CONSTRAINTS

Takahiro Nakano and Masahiro Nagamatu

Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology

Abstract: We proposed a neural network called LPPH-CSP for solving constraint satisfaction problem (CSP). The LPPH-CSP is not trapped by any point which is not a solution of the CSP, and it can update all neurons simultaneously. In this paper, we propose two methods to improve the efficiency of the LPPH-CSP. Though the LPPH-CSP can deal with several types of constraints of the CSP, it treats all constraints evenly. One of the proposed methods distinguishes the types of constraints for solving the CSP more efficiently. Another one of the proposed methods applies fast local search (FLS) to the LPPH-CSP. Experimental results show the effectiveness of our proposals.

Key words: CSP; constraint satisfaction problem; neural network; Lagrangian method.

1. INTRODUCTION

Many AI problems can be represented by the CSP (Constraint Satisfaction Problem). The CSP is a problem to find a variable assignment which satisfies all given constraints. We proposed a neural network called LPPH-CSP¹ for solving the CSP by extending a neural network called LPPH² which is proposed for solving the SAT which is a problem to find an assignment of truth values to the variables which satisfies the given CNF (Conjunctive Normal Form).

In this paper, we propose two methods to improve the LPPH-CSP. First we apply the FLS (Fast Local Search)³ which is useful for the speedup of the neighborhood search process. Next, we extend the LPPH-CSP by introducing importance of the types of the constraints. The LPPH-CSP can deal with several types of constraints of the CSP, however, it treats all

constraints evenly. Our extended LPPH-CSP distinguishes the importance of the types of constraints. It can solve the CSP more efficiently.

2. CSP

The CSP is a combinatorial problem to find a solution which satisfies all given constraints. The CSP is defined by a triple (X, D, C) . $X = \{X_1, X_2, \dots, X_n\}$ is a finite set of variables. $D = \{D_1, D_2, \dots, D_n\}$ is a finite set of domains. Each domain D_i is a finite set of values and each variable X_i is assigned a value in D_i . $C = \{C_1, C_2, \dots, C_m\}$ is a finite set of constraints. A solution of the CSP is a variable assignment to X which satisfies C . Let x_{ij} be a Boolean variable which represents "variable X_i is assigned the j th value of D_i ". x_{ij} is called a VVP (Variable-Value Pair). Constraint C_r consists of a set of VVPs. In this paper we consider ALT(n, S) [at-least- n -true constraint]¹ which requires that at least n of VVPs in S must be true where S is a finite set of VVPs. We also consider ALF(n, S) [at-least- n -false constraint], AMT(n, S) [at-most- n -true constraint], and AMF(n, S) [at-most- n -false constraint].

3. LPPH-CSP

From now on, each VVP x_{ij} is not a Boolean variable, but a variable which has a continuous value between 0 and 1. The dynamics of the LPPH-CSP is defined as follows¹:

$$\begin{aligned} \frac{dx_{ij}}{dt} &= x_{ij}(1-x_{ij}) \sum_{r=1}^m w_r s_{rij}(\mathbf{x}), \\ \frac{dw_r}{dt} &= -\alpha w_r + h_r(\mathbf{x}), \end{aligned} \quad (1)$$

where $s_{rij}(\mathbf{x})$ represents a force put on x_{ij} for satisfying constraint C_r , w_r is the weight of constraint C_r , $h_r(\mathbf{x})$ represents the degree of unsatisfaction of constraint C_r , α is a constant called attenuation coefficient. The LPPH-CSP searches a solution of the CSP by numerically solving the above dynamics.

3.1 Applying the fast local search to LPPH-CSP

Bently⁴ proposed the approximate 2-opt method for the TSP (traveling salesman problem). The FLS (Fast Local Search)³ is a generalization of the approximate 2-opt method. In the FLS algorithm, the neighborhood of the

problem is broken down to a small number of sub-neighborhoods and an activation bit is attached to each of them. Only the sub-neighborhoods with the activation bit set to 1 are searched. These sub-neighborhoods are called active sub-neighborhood, and the other are called inactive sub-neighborhood, and they are not searched. We applied the FSL to the LPPH-CSP by the following algorithm:

1. Initially, assign activation bit to all neuron (initial value of activation bit is 1).
2. According to (1) update x_{ij} 's of neurons for which the activation bit is 1. Inactivate a neuron x_{ij} , if x_{ij} does not change its value and all constraints which contain x_{ij} are satisfied. Otherwise activate a neuron x_{ij} .
3. Update weights by (1).
4. Repeat 2 and 3 until the LPPH-CSP find a solution.

3.2 Considering importance of constraints

We consider four types of constrains which are ALT, ALF, AMT and ALF. The LPPH-CSP treats these constraints equally. We extended the LPPH-CSP so as to deal with the importance of the types of constraints. This is called LPPH-CSP with IC (Importance of Constraints). For the dynamics of LPPH-CSP with IC, we replace $w_r s_{rij}(\mathbf{x})$ with $\rho_r w_r s_{rij}(\mathbf{x})$ in (1). ρ_r indicates the importance of constraint C_r . The value of ρ_r depends on the types of constraint C_r .

4. EXPERIMENTAL RESULTS

We examined the efficiency of proposed methods by experiments. Fig.1 shows the average CPU time for solving Car Sequencing Problems and N-Queen Problems. The average of each problem is calculated by changing initial points 30 times. Furthermore the average of these average CPU time is calculated for the problems in each problem class. The constraints in Car Sequencing Problems and N-Queen Problems are represented by ALT and AMT. We tried various values of ρ_r for CSPs. As the result, we obtained that the best values of ρ_r for solving Car Sequencing Problems and N-Queen problems is around $\rho_r = 1$ for ALT and around $\rho_r = 2$ for AMT.

From experimental results, we can see that the LPPH-CSP with FLS & IC is most effective within the four methods.

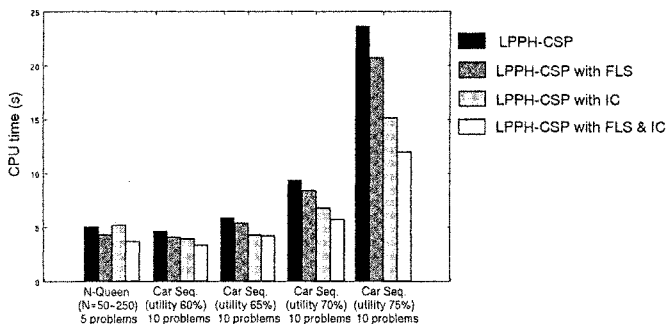


Figure 1. CPU time of proposed methods for N-Queen Problems and Car Sequencing Problems

5. CONCLUSION

We proposed two acceleration methods for the LPPH-CSP. First, we applied the FLS which can speed up the neighborhood search process of the LPPH-CSP. Next, we extended the LPPH-CSP so as to deal with the importance of constraints.

Experimental results show that these methods are effective especially for hard problems. For future works we are going to study theoretical analysis of experimental results obtained in this paper, and we plan to determine more effective values of ρ_r 's of the LPPH-CSP with IC.

REFERENCES

1. T. Nakano and M. Nagamatu, "Solving CSP via Neural Network Which Can Update All Neurons Simultaneously", *In Proceedings of the SCIS & ISIS 2004*, (to appear).
2. M. Nagamatu and T. Yanaru, "On the Stability of Lagrange Programming Neural Networks for Satisfiability Problems of Propositional Calculus", *Neurocomputing*, 13, 119-133, 1995.
3. C. Voudouris, and E. Tsang, "Partial Constraint Satisfaction Problems and Guided Local Search", *In Proceedings of the PACT-96*, 337-356, 1996.
4. J.L.Bently, "Fast Algorithms for Geometric Traveling Salesman Problems", *ORSA Journal on Computing*, 4, 387-441, 1992.