A New Approach for Weighted Constraint Satisfaction: Theoretical and Computational Results

02991440 東京工業大学 LAU Hoong Chuin

1 Introduction

An instance of the binary Weighted Constraint Satisfaction Problem (W-CSP) is defined by a set of variables, their associated domains of values and a set of weighted binary constraints. Our goal is to find an assignment which maximizes the weighted sum of satisfied constraints. W-CSP is a generalization of combinatorial optimization problems such as MAX CUT.

Finding optimal solutions of W-CSP is known to be computationally hard. Freuder and Wallace [1] gave the first formal definition of PCSP which is a special case of W-CSP having unit weights. They proposed a general framework based on branch-and-bound.

Our work is motivated mainly by the potential application of W-CSP in scheduling. With the rapid increase in the speed of computing and the growing need for efficiency in scheduling, it becomes increasingly important to explore ways of obtaining better schedules at some extra computational cost, short of going all the way towards the usually futile attempt of finding an optimal schedule. Our paper describes a new approach meant to achieve this goal. Another highlight of our approach is that the solution obtained has a provable worst-case bound. In a previous paper [3], we performed a worst-case analysis of local search for PCSP, and in this paper, we improve that result. The knowledge of the worst-case performance gives us some peace of mind: that our algorithm will never perform embarrasingly poorly.

Our approach is heavily based on the notion of *randomized rounding*, due to Raghavan and Thompson [4]. The key idea is to formulate a given optimization problem as a quadratic integer program and solve a polynomial-time solvable semidefinite program. Then treat the optimal fractional solution of the semidefinite program as a probability distribution and obtain an integer solution using this distribution. This approach yields a randomized algorithm. which can be derandomized using the *method of conditional probabilities*. Recently, Goemans and Williamson applies a similar approach to approximate MAX CUT [2]. Our experiments illustrate that this approach can handle problem sizes beyond what enumerative search algorithms can handle, and thus is a candidate for solving real-world large-scale problems.

2 Preliminaries

Let $V = \{1,\ldots,n\}$ be a set of variables. Each variable has a *domain* and for simplicity, we assume that all domains have fixed size k and are equal to the set $K = \{1,\ldots,k\}$. A binary *constraint* between two variables i and l is a relation over $K \times K$ which defines the pairs of values that i and l can take simultaneously. Given an assignment $\sigma : V \longrightarrow K$, the constraint is said to be *satisfied* iff the pair (σ_i, σ_l) is an element of the relation. A W-CSP instance is defined by a set V of variables, a collection M of constraints, integer k, and a weight function $w : M \longrightarrow Z^+$. Its output is an assignment such that the weighted sum of satisfied constraints is maximized. Denote by W-CSP(k) the class of instances with domain size k.

For each constraint $j \in M$ let w_j, R_j and $s_j = \|R_j\|/k^2$ denote its weight, relation and *strength* respectively; let α_j and β_j denote the two variables connected by j; and let $c_j(u,v) = 1$ if $(u,v) \in R_j$ and 0 otherwise. Let $s = \sum_{j\in M} w_j s_j / \sum_{j\in M} w_j$ denote the strength of a W-CSP instance. A W-CSP instance is *satisfiable* iff there exists an assignment which satisfies all constraints simultaneously.

We say that a maximization problem P *can be approximated within* $0 < \epsilon \leq 1$ iff there exists a polynomial-time algorithm A such that for all input instances y of P, A computes a solution whose objective value is at least ϵ times the optimal value of y (denoted $OPT(y)$). The quantity ϵ is commonly known as the *performance guarantee* or *approximation ratio* for P.

3 Theoretical Results

Linear-time Greedy Algorithm Suppose we are given a n by k probability matrix $\Pi = (p_{iu})$ such that all $p_{i:u} \in [0..1]$ and $\sum_{u=1}^{k} p_{i:u} = 1$ for all $1 \leq i \leq n$. If we assign each variable i independently to value u with probability $p_{i:u}$, then the expected weight of the resulting assignment is given by $\hat{W} = \sum_{j\in M} w_j \left(\sum_{u,v\in K} p_{\alpha_j:u} \cdot p_{\beta_j:v} \cdot c_j(u,v) \right)$.
The method of conditional probabilities specifies that there must exist an assignment whose

weight is at least $\hat{W}$ and such an assignment can be found deterministically via greedy in linear time. Now consider the random assignment, i.e. $p_{i,k} = 1/k$ for all i, k. By linearity of expectation, we can show that:

Theorem 3.1. W-CSP(k) can be approximated within absolute ratio s in $O(mk^2)$ time.

Next, we will consider two types of rounding of semidefinite programs and analyse their respective worst-case performances.

Simple Rounding Formulate an instance of W-CSP(k) by a Quadratic Integer Program (Q′):

$$
\begin{aligned}
Q'\text{: maximize } & \sum_{j \in M} w_j f'_j(x) \\
\text{subject to } & \sum_{u \in K} x_0 x_{i,u} = -(k-2) \quad \text{for } i \in V \\
& x_{i,u} \in \{-1, +1\} \text{ for } i \in V,\ u \in K \\
& x_0 = +1
\end{aligned}
$$

where

$f'_j(x) = \frac{1}{4} \sum_{u,v} c_j(u,v) \left(1 + x_0 x_{\alpha_j,u}\right) \left(1 + x_0 x_{\beta_j,v}\right)$

encodes the satisfiability of constraint j.

Interpret each variable x as a 1-dimensional vector of unit length and relax it to a unit-vector X lying in the sphere S_N where $N = nk + 1$. The notation $A \cdot B$ means the inner product of vectors A and B. The resulting relaxation problem (P) is the following:

$$
\begin{aligned}
\text{P: maximize } & \sum_{j \in M} w_j F_j(X) \\
\text{subject to } & \sum_{u \in K} X_0 \cdot X_{i,u} = -(k-2) \quad \text{for } i \in V \\
& X_0, X_{i,u} \in S_N \quad \text{for } i \in V,\ u \in K
\end{aligned}
$$

where $F_j(X) = \frac{1}{4} \sum_{u,v} c_j(u,v)\ (1 + X_{\alpha_j,u} \cdot X_{\beta_j,v} + X_0 \cdot X_{\alpha_j,u} + X_0 \cdot X_{\beta_j,v})$.

We propose the following randomized algorithm to approximate (Q′) for $k = 2$:

1. (Relaxation) Solve (P) to optimality and obtain an optimal set of vectors X^*.

2. (Randomized Rounding) Construct a corresponding assignment for (Q′) as follows. For each i, with probability $1 - \frac{\arccos(X_0^* \cdot X_{i,u}^*)}{\pi}$, assign $x_{i,u}$ to $+1$ and $x_{i,v}$ $(v \neq u)$ to -1.

For the case of $k = 3$, we can show by adding valid inequalities that it reduces to the case of $k = 2$.

Theorem 3.2. W-CSP(k) ($k \leq 3$) can be approximated within 0.408.

Rounding Via Hyperplane Partitioning Adopting a rounding scheme in the veins of Goemans and Williamson [2], we can improve the approximation ratio for $k = 2$:

Theorem 3.3. W-CSP(2) can be approx. within 0.634, and 0.878 for satisfiable instances.

4 Computational Experience

In this section, we report our computational experience. We experiment on *satisfiable* instances so that we can compute the approximation ratio without having to obtain optimal solutions.

Four algorithms are compared. Greedy LS refers to hill-climbing local search with an initial assignment generated greedily, i.e. arrange the variables in a linear order and assign them in sequence the value that maximizes the weighted sum of satisfied constraints. Random LS refers to hill-climbing local search with a random initial assignment. Rand Round refers to our simple rounding algorithm, and RR LS refers to hill-climbing local search with an initial assignment generated by Rand Round.

Our experiments yield these observations:

1. Greedy LS performs well on dense instances, but not so well on sparse ones.

2. Random LS performs reasonably well on sparse instances but not on dense instances.

3. Rand Round performs consistently well on all instances, achieving at least 97% optimality for $k = 2$ and 86% optimality for $k = 5$. Rand Round outperforms Greedy LS and Random LS for $k = 5$.

4. RR LS outperforms all other approaches in all cases, achieving 99% optimality for $k = 2$ and 96% optimality for $k = 5$.

References

[1] Eugene C. Freuder and Richard J. Wallace. Partial Constraint Satisfaction. *Artif. Intell.*, 58(1-3):21–70, 1992.

[2] M. Goemans and D. Williamson. Approximation algorithms for MAX CUT and MAX 2SAT. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 422–431, 1994.

[3] H. C. Lau. Approximation of constraint satisfaction via local search. In *Proc. 4th Wksp. on Algorithms and Data Structures (WADS)*, pages 461–472, 1995.

[4] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.