1996

Randomized approximation of the constraint satisfaction problem

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Osamu WATANABE
Tokyo Institute of Technology

Citation
LAU, Hoong Chuin and WATANABE, Osamu. Randomized approximation of the constraint satisfaction problem. (1996). *Nordic Journal of Computing*. 3, (4), 405-424. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/150

Published in Nordic Journal of Computing, 2016 Winter, 3 (4), 405-424.

RANDOMIZED APPROXIMATIONS OF THE CONSTRAINT SATISFACTION PROBLEM

HOONG CHUIN LAU
Information Technology Institute
11 Science Park Rd
Singapore 117685
hclau@cs.titech.ac.jp

OSAMU WATANABE
Tokyo Institute of Technology
Department of Computer Science
Meguro-ku, Ookayama
Tokyo 152, Japan
watanabe@cs.titech.ac.jp

Abstract. We consider the Weighted Constraint Satisfaction Problem (W-CSP) which is a fundamental problem in Artificial Intelligence and a generalization of important combinatorial problems such as MAX CUT and MAX SAT. In this paper, we prove non-approximability properties of W-CSP and give improved approximations of W-CSP via randomized rounding of linear programming and semidefinite programming relaxations. Our algorithms are simple to implement and experiments show that they are run-time efficient.

CR Classification: Please add missing CR-classification !!!

Key words: Approximation algorithms, constraint satisfaction problem, randomized rounding.

1. Introduction

An instance of the Weighted Constraint Satisfaction Problem (W-CSP) is defined by a set of variables, their associated domains of values and a set of constraints governing the assignment of values to variables. Each constraint is associated with a positive integer weight. The output is an assignment which maximizes the weighted sum of satisfied constraints.

W-CSP is a fundamental problem in Artificial Intelligence and Operations Research. Many real-world problems can be represented as W-CSP, among which are scheduling and timetabling problems. In scheduling for example, our task is to assign resources to jobs under a set of constraints, some of which are more important than others. Most often, instances are over-constrained and no solution exists that satisfies all constraints. Thus, our goal is to find an assignment which maximizes the weights of the satisfied constraints.

W-CSP is interesting theoretically because it is a generalization of several key NP-optimization problems. A W-CSP instance has *arity* t iff all its constraints are defined on a set of t or less variables. It has domain size k iff the sizes of all domains are k or less. When $k = 2$, we get a generalization of the maximum satisfiability problem (MAX SAT), while the case of $k = t = 2$

Received July 30, 1996; revised September 9, October 11 and 23, 1996; accepted October 24, 1996.

is a generalization of the maximum cut problem (MAX CUT). These problems are MAX SNP-complete. Papadimitriou and Yannakakis [1991] proved that for every MAX SNP-complete problem, there exists a constant $c > 0$ within which the problem can be approximated in polynomial time. On the other hand, Arora *et al.* [1992] proved that for every MAX SNP-complete problem, there exists a constant $d < 1$ within which the problem cannot be approximated in polynomial time unless P=NP. Hence, designing polynomial time approximation algorithms to close the gap between constants c and d is a key concern.

Recently, many surprising and interesting approximation results for MAX CUT, MAX SAT and their variants have been obtained using *randomized rounding*. The notion of *randomized rounding* was originally proposed by Raghavan and Thompson [1987]. The key idea is to formulate a given optimization problem as an integer program and then solve a polynomial-time solvable relaxation, which is usually a convex mathematical program such as a linear or semidefinite program. A semidefinite program seeks to optimize a linear function of a symmetric matrix subject to linear constraints and the constraint that the matrix be *positive semidefinite* (abbrev. PSD). Given the optimal fractional solution of the relaxation, the values of the fractional solution are treated as a probability distribution and an integer solution is obtained by rounding with respect to this distribution. This approach yields a randomized algorithm. Using the *method of conditional probabilities*, one can convert it into a deterministic algorithm which always produces a solution whose objective value is at least the expected value of the solution produced by the randomized algorithm. This method is implicitly due to Erdös and Selfridge [1989] and clearly explained in the text of Alon and Spencer [1992]. The seminal work of Goemans and Williamson [1994] demonstrated that MAX CUT and MAX 2SAT can be approximated within a factor of 0.878... by randomized rounding of semidefinite programming relaxations.

1.1 Related Work

Freuder [1989] gave the first formal definition of PCSP which is a special case of W-CSP having unit weights. Freuder and Wallace [1992] proposed a polynomial time algorithm based on reverse breadth-first search to solve PCSP whose underlying constraint network is a tree. For the general PCSP, they proposed a general framework based on branch-and-bound and its enhancements (see Wallace and Freuder [1993] and Wallace [1995]). Existing heuristic methods include the connectionist architecture GENET by Tsang [1993] and guided local search by Voudouris and Tsang [1995]. However, these algorithms may perform badly in the worst case.

Approximation algorithms are algorithms which have worst-case performance bounds. Recent works in the approximation of W-CSP are as follows. Amaldi and Kann [1994] considered the problem of of finding maximum feasible subsystems of linear systems. Their problem may be seen as unit-weight W-CSP where the domains are real numbers. They showed that their problem remains MAX SNP-hard even if the domains are bipolar (specifically $\{-1, +1\}$) and the constraints are restricted to binary inequalities of certain forms. In Khanna *et al.* [1994], one section is devoted to the approximation of W-CSP of domain size 2 and fixed arity t. They showed that it is approximable within a factor of $1/2^t$ by a fairly sophisticated local search technique. That ratio was very recently improved by Trevisan [1996] to $1/2^{t-1}$ via randomized rounding of a linear program. Lau [1995] considered W-CSP in terms of the *arc-consistency* property and obtained tight approximation via local search. Feige and Goemans [1995] recently proved a 0.859 bound for W-CSP of domain size and arity 2.

1.2 Our Contributions

In this paper, we are concerned with the approximation of W-CSP with binary constraints, although some of our results are shown to be easily extended to instances of higher arity.

This paper is organized as follows. Section 2 gives the definitions and notations which will be used throughout the paper. In Section 3, we give non-approximability results via interactive proofs. We show that there exists a constant $t > 0$ such that W-CSP of n variables cannot be approximated within a factor of $1/2^{(\log n)^t}$ (which is stronger than $1/polylog(n)$). We also show that there exists a constant $c > 0$ such that for every $k \geq 2$, W-CSP of domain size k is not approximable within $(1/k)^c$. In Section 4, we use the method of conditional probabilities to derive a linear-time derandomization scheme which is essentially a greedy algorithm parameterized by the probability distribution. On uniform distribution, this scheme gives an approximation ratio of $1/k^2$. In Section 5, we use randomized rounding of linear program relaxation to obtain a tight approximation ratio of $1/k$. This ratio can be generalized to $1/k^{t-1}$ for instances of arity t, thereby generalizing Trevisan's result. In Section 6, we obtain higher ratios for instances with domain sizes of 2 and 3, based on rounding of semidefinite program relaxation. We obtain a constant-approximation of 0.408 for the case of $k \leq 3$. Unfortunately, one cannot improve the ratio to more than 0.5 because the formulation is inherently weak, in the sense that, given any pre-determined rounding scheme, we can construct an instance where the expected weight of the derived assignment is no more than 0.5 times the optimal value. The strength of the above algorithms is that, once the linear/semidefinite programs have been solved, the rounding part can be derandomized in linear-time using the scheme proposed in Section 4. Finally, we extend the hyperplane-rounding method of Goemans and Williamson [1994] for the case $k = 2$ and derive a ratio of 0.878-ϵ, where ϵ depends on the *type* of constraints. This offers a better bound than the 0.859 bound of Feige and Goemans in some cases, and is simpler to implement. Derandomization of this algorithm was recently proposed by Mahajan and Ramesh [1995].

2. Preliminaries

Let V be a set of n variables indexed by 1 to n. Each variable takes a value from the domain $K = \{1, \ldots, k\}$, for a given constant k All results presented in this paper also hold for variables having unequal-sized domains provided that the maximum domain size is k. Let $E = \{C_1, \ldots, C_m\}$ be a set of binary constraints on V. Each constraint C_j is associated with:

- α_j, β_j : the indices of variables related by C_j;
- R_j : a non-empty relation over $K \times K$ (i.e. a set of value pairs);
- w_j : a positive integer weight.

Constraints of higher arity are defined analogously.

The Weighted Constraint Satisfaction Problem (W-CSP) is to find an assignment $\sigma : V \to K$ that maximizes the sum of the weights of satisfied constraints. A constraint C_j is said to be *satisfied* iff $(\sigma_{\alpha_j}, \sigma_{\beta_j}) \in R_j$, i.e. the assigned values are related by the relation. W-CSP can be represented by a *constraint graph* $G = (V, E)$ whose vertices and edges represent the variables and constraints respectively.

We will use the following notations throughout the paper. Let W-CSP(k) denote the class of instances with domain size at most k. Let W denote the total weight of all constraints. Let c_j be a Boolean predicate $\{1, \ldots, k\} \times \{1, \ldots, k\} \to \{0, 1\}$ where $c_j(u, v) = 1$ if the pair (u, v) is an element of R_j. Let $s_j = \|R_j\|/k^2$ and $s = \sum_{C_j \in E} w_j s_j / W$. The quantity s is called the *strength* (the weighted average strengths of all its constraints). Note that $s \geq 1/k^2$ because every constraint relation contains at least 1 out of the k^2 possible pairs.

A W-CSP instance is termed *satisfiable* iff there exists an assignment which satisfies all constraints simultaneously. A constraint is said to be *2-consistent* iff there are at least two value pairs in the relation. Otherwise, it is said to be *1-consistent*. Clearly, every 1-consistent constraint is satisfied by one unique instantiation of its variables.

We say that a maximization problem P *can be approximated within* $0 < c \leq 1$ 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 c times the optimal value of y (denoted $OPT(y)$). The quantity c is commonly known as the *performance guarantee* or *approximation ratio* for P. Observe that the ratio is at most 1. The ratio is *absolute* if we consider the *maximum possible* objective value instead of $OPT(y)$. In the case of W-CSP for example, the maximum possible objective value is the sum of edge weights, although the optimal value can be much smaller. Hence, the absolute ratio is always a lower bound of (and therefore better bound than) the performance guarantee.

3. Non-Approximability of W-CSP

In this section, we give two non-approximability results for W-CSP. First, by mimicking the proof of Feige and Lovász [1992], we show that W-CSP is not approximable within a factor of $1/2^{(\log n)^t}$ factor for some constant $0 < t < 1$, unless EXP=NEXP. Next, by extending the proof and using the recent result of Raz [1995], we show that for all $\epsilon > 0$, there exists a constant k depending on ϵ such that W-CSP(k) cannot be approximated within ϵ unless P=NP.

3.1 Two Prover One Round Interactive Proof

In a two-prover proof system, two provers P_1 and P_2 try to convince a probabilistic polynomial-time verifer V that a common input x of size n belongs to a language L. V sends messages s and t respectively to P_1 and P_2 according to a distribution π which is a polynomial-time computable function of input x and a random string r. The provers return answers $\mathsf{P}_1(s)$ and $\mathsf{P}_2(t)$ respectively without communicating with each other.

Definition 1. *A language L has a two-prover one-round interactive proof system of parameters ϵ, f_1, f_2 (abbreviated IP(ϵ, f_1, f_2)) if, in one round of communication:*

(1) $\forall x \in L,\ \exists\ \mathsf{P}_1, \mathsf{P}_2\ Pr[(\mathsf{V}, \mathsf{P}_1, \mathsf{P}_2)$ *accepts* $x] = 1$*;*

(2) $\forall x \notin L,\ \forall\ \mathsf{P}_1, \mathsf{P}_2\ Pr_\pi[(\mathsf{V}, \mathsf{P}_1, \mathsf{P}_2)$ *accepts* $x] < \epsilon$*;*

(3) V *uses* $O(f_1)$ *random bits; and*

(4) the answer size is $O(f_2)$*.*

Several results relating language classes to interactive proofs have appeared recently. Particularly, we need the following properties:

(1) (Feige and Lovász [1992]) All languages in NEXP have IP$(2^{-n}, n^q, n^q)$, for some constant $q \geq 1$;

(2) (Fortnow *et al.* [1988],Arora *et al.* [1992],Raz [1995]) For all $L \in NP$, there exists a constant $0 < c < 1$ such that for every integer $t \geq 1$, L has IP$(2^{-ct}, t \log n, t)$ (t is the number of parallel repetitions used in Raz [1995]).

A two-prover one-round proof system can be modelled as a problem on a two-player game G. Let S and T be the sets of possible messages. Hence, the sizes of S and T are $O(2^{O(f_1)})$. A pair of messages $(s,t) \in S \times T$ is chosen at random according to probability distribution π and sent to the players respectively. A strategy of a player is a function from messages to answers. Let U and W be the sets of answers returned by the two players respectively, whose sizes are $O(2^{O(f_2)})$. The objective is to choose strategies P_1 and P_2 which maximizes the probability over π that $\mathsf{V}(s, t, \mathsf{P}_1(s), \mathsf{P}_2(t))$ accepts x. Let the value of the game, denoted $\omega(G)$, be the probability of success of the players' optimal strategy in the game G.

3.2 Proof of Non-Approximability

We can formulate the problem of finding the optimal strategy for the game G as a W-CSP instance with a bipartite constraint graph as follows. The set of nodes in the constraint graph is,

$$V = \{x_s \ : \ s \in S\} \bigcup \{y_t \ : \ t \in T\}.$$

Edges are given by:

$$E = \{(x_s, y_t) \ : \ \pi(s,t) \neq 0\}.$$

The domain of each x_s (resp., y_t) is U (resp., W). For each edge $(x_s, y_t) \in E$, the corresponding relation contains exactly those pairs (u, v) such that $\mathsf{V}(s,t,u,w)$ accepts x. Finally, define the weight of the constraint $(x_s, y_t) \in E$ as the number of random strings on x which generate the query pair (s,t) (i.e. the value $\pi(s,t)$ scaled up to an integer). Since each variable must be assigned exactly one value, the assignment of variables in S and T encodes a strategy for the two players respectively. By definition, the scaled optimum value of this W-CSP instance (i.e. optimal value divided by the total number of random strings) is exactly $\omega(G)$ and hence the accepting probability of the proof system.

THEOREM 1. *There exists a constant* $0 < t < 1$ *such that W-CSP of* n *variables cannot be approximated within a factor of* $1/2^{(\log n)^t}$ *factor, unless* EXP $=$ NEXP*.*

PROOF. Consider an arbitrary language L in NEXP and an input x. By the property of NEXP languages mentioned above, there is a two-prover one-round proof system such that the acceptance probability reflects membership of x in L. We can construct a W-CSP instance y with $n = 2^{|x|^q}$ (for some constant $q \geq 1$) variables whose scaled optimal value is the acceptance probability. Suppose there is a polynomial time algorithm which approximates y to $1/2^{(\log n)^t}$ factor. Then, if $x \in L$, the solution value returned by the algorithm is at least $1/2^{|x|^{qt}}$ and if $x \notin L$, the optimal value is less than $2^{-|x|}$. Hence, by choosing $t < 1/q$ and applying the polynomial time approximation algorithm to y, we obtain an exponential time decision procedure for L, implying EXP=NEXP. □

Next, we consider the non-approximability of W-CSP with fixed domain size k. The result is given by following theorem whose proof was suggested by Trevisan through personal communication:

THEOREM 2. *There exists a constant* $0 < c \leq 1$ *such that for all* $k \geq 2$*, W-CSP(k) cannot be approximated within* $(1/k)^c$*, unless P=NP.*

PROOF. Consider an arbitrary language L in NP and an input x of size n. First suppose the given $k = 2^t$, where t is a positive integer. By the property of NP languages mentioned above, a two-prover one-round proof exists, which can be simulated by a W-CSP instance y with $O(2^{t \log n}) = O(n^t)$ variables and domain size k. Suppose there is a polynomial time which approximates y within $(1/k)^c$ for some $0 < c \leq 1$. Then, we have again a gap in the acceptance probability. This enables us to determine membership of x in polynomial time, implying P=NP. If k is not a power of 2, we let t be the smallest integer where $2^t \geq k$. We can conclude similarly that W-CSP(k) cannot be approximated within $(1/k)^{c'}$, for some constant $0 < c' \leq 1$. □

4. Method of Conditional Probabilities

In this section, we derive a linear-time greedy algorithm based on the method of conditional probabilities. This algorithm will be used to derandomize the randomized rounding schemes proposed in Sections 5 and 6.

Consider an instance of W-CSP(k) of n variables. Suppose we are given an n by k 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 a value u to each variable i independently with probability $p_{i,u}$, we obtain a probabilistic assignment whose expected weight is given by:

$$\hat{W} = \sum_{C_j \in E} w_j \times \Pr[\, C_j \text{ is satisfied}] \; = \; \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \cdot p_{\alpha_j,u} \cdot p_{\beta_j,v} \right).$$

Hence, there must exist an assignment whose weight is at least $\hat{W}$. The method of conditional probabilities specifies that such an assignment can be found deterministically by computing certain conditional probabilities. The following greedy algorithm performs the task:

> Assign variables 1 to n iteratively. At the beginning of iteration i, let $\tilde{W}$ denote the expected weight of the partial assignment where variables $1, \ldots, i-1$ are fixed and variables $i, \ldots, n$ are assigned according to distribution Π. Let $\tilde{W}_u$ denote the expected weight of that partial assignment with variable i fixed to the value u. Assign value v to variable i maximizing $\tilde{W}_v$.

From the law of conditional probabilities, we know:

$$\tilde{W} = \sum_{u=1}^{k} \tilde{W}_u \cdot p_{i,u}.$$

Since we always pick v such that $\tilde{W}_v$ is maximized, $\tilde{W}$ is non-decreasing in all iterations, and the complete assignment has weight no less than the initial expected weight, which is $\hat{W}$.

Therefore, to obtain assignments of large weights, the key factor is to obtain the probability distribution matrix Π such that the *expected weight* is as large as possible. In the following, we consider the most naive probability distribution – the *random* assignment, i.e. for all i and u, we have $p_{i,u} = 1/k$.

By linearity of expectation (i.e. expected sum of random variables is equal to the sum of expected values of random variables), the expected weight of the random assignment is given by,

$$\hat{W} \;=\; \sum_{C_j \in E} w_j \cdot s_j \;=\; s \sum_{C_j \in E} w_j .$$

That is, the expected weight is s times the total edge weights, implying that W-CSP(k) can be approximated within absolute ratio s. Since each constraint contains at least one value pair, this gives an absolute approximation ratio of $1/k^2$.

Time Complexity

We show how the conditional probabilities can be efficiently computed. $\tilde{W}_u$ can be derived from $\tilde{W}$ as follows. Maintain a vector r where r_j stores the probability that C_j is satisfied given that variables $1, \ldots, i-1$ are fixed and the remaining variables assigned randomly. Then, $\tilde{W}_u$ is just $\tilde{W}$ offset by the change in probabilities of satisfiability of those constraints incident to variable i. More precisely,

$$\tilde{W}_u = \tilde{W} + \sum_{C_j \text{ incident to } i} w_j (r'_j - r_j)$$

where r'_j is the new probability of satisfiability of C_j. Letting l be the second variable connected by j, r'_j is computed as follows:

> `if` $l < i$ (i.e. l has been assigned)
> `then set` r'_j `to 1 if` $(\sigma_l, u) \in R_j$ `and 0 otherwise`
> `else set` r'_j `to the fraction` $\#\{v \in K \;:\; (u,v) \in R_j\}/k$.

Clearly, the computation of each $\tilde{W}_u$ takes $O(m_i k)$ time, where m_i is the number of constraints incident to variable i. Hence, the total time needed is $O(\sum m_i k^2) = O(mk^2)$, which is linear in the size of the input (assuming that the constraint value pairs are explicitly listed).

5. Randomized Rounding of Linear Program

In this section, we present randomized rounding of linear program and analyze its performance guarantee.

For every variable $i \in V$, define k Boolean variables $x_{i,1}, \ldots, x_{i,k}$ such that value u is assigned to i in the W-CSP instance iff $x_{i,u}$ is assigned to 1. A W-CSP instance can be formulated by the following integer linear program:

(IP) :	maximize	$\sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) z_{j,u,v} \right)$	
subject to	$z_{j,u,v} \leq x_{\alpha_j,u}$	for $C_j \in E$ and $u,v \in K$	(I1)
	$z_{j,u,v} \leq x_{\beta_j,v}$	for $C_j \in E$ and $u,v \in K$	(I2)
	$\sum_{u \in K} x_{i,u} = 1$	for $i \in V$	(I3)
	$x_{i,u} \in \{0,1\}$	for $i \in V$ and $u,v \in K$	(I4)
	$0 \leq z_{j,u,v} \leq 1$	for $C_j \in E$ and $u,v \in K$	(I5)

Inequalities (I1) and (I2) ensure that $z_{j,u,v}$ is 1 only if $x_{\alpha_j,u}$ and $x_{\beta_j,v}$ are both 1. Equation (I3) ensures that each W-CSP variable is assigned exactly one value. Since the edge weights are positive and we are maximizing a linear function of z, the inner sum of the objective function is 1 if C_j is satisfied and 0 otherwise.

Given (IP), solve the corresponding linear programming problem (LP) by relaxing the integrality constraints (I4). Let (x^*, z^*) denote the optimal solution obtained. We propose the following rounding scheme:

> assign u to variable i with probability $\frac{1}{2}\left(x^*_{i,u} + \frac{1}{k}\right)$, for all $i \in V$ and $u \in K$.

This is a valid scheme since the sum of probabilities for each variable is exactly 1 by equation (I3).

CLAIM 1. *The expected weight of this probabilistic assignment is at least* $\frac{1}{k}$ *OPT(IP).*

PROOF. The expected weight of the probabilistic assignment is given by,

$$\begin{aligned}
\hat{W} &= \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \cdot \frac{1}{2}(x^*_{\alpha_j,u} + \frac{1}{k}) \cdot \frac{1}{2}(x^*_{\beta_j,v} + \frac{1}{k}) \right) \\
&\geq \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \cdot \frac{1}{4}(z^*_{j,u,v} + \frac{1}{k})^2 \right)
\end{aligned}$$

where the inequality follows from (I1) and (I2). By simple calculus, one can derive that the minimum value of the function

$$f(z) = \frac{(z + \frac{1}{k})^2}{4z}$$

in the interval $[0,1]$ is $1/k$ at the point $z = 1/k$. Hence, the expected weight

$$\hat{W} \geq \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \cdot \frac{1}{k} z^*_{j,u,v} \right) \geq \frac{1}{k} OPT(\mathrm{LP}) \geq \frac{1}{k} OPT(\mathrm{IP})$$

which completes the proof. □

The above analysis is tight. In fact, the above rounding scheme is best possible with respect to the (IP) formulation, as can be shown by considering a W-CSP instance in which all constraints are full relations (i.e. contain all possible value pairs). Then, the optimal solution is the sum of weights W. On the other hand, a feasible solution of the linear program where all variables are equal to $1/k$ has objective value kW.

The above formulation and rounding scheme can be extended in a straightforward manner to handle instances of arbitrary arity t. In this case, we assign variable i the value u with probability $\frac{1}{t}\left(x^*_{i,u} + \frac{t-1}{k}\right)$.

The rounding step can be derandomized in linear-time using the greedy method proposed in Section 4. Hence:

THEOREM 3. *For any fixed t, W-CSP(k) of arity t can be approximated within an absolute ratio of* $\frac{1}{k^{t-1}}$.

This ratio is almost the best that we can hope for in light of Theorem 2.

6. Randomized Rounding of Semidefinite Program

In Section 5, we saw that a linear programming relaxation gives a performance ratio of $1/k$. Is it possible to improve this ratio for small domain sizes, such as $k = 2, 3$? In this section, we present improved approximation via semidefinite programming.

6.1 A Simple Rounding Scheme

Consider an instance of W-CSP(k). Formulate a corresponding quadratic integer program (Q) as follows. For every variable $i \in V$, define k decision variables $x_{i,1}, \ldots, x_{i,k} \in \{-1, +1\}$ such that i is assigned value u in the W-CSP instance iff $x_{i,u}$ is assigned to $+1$ in (Q).

$$
\begin{array}{lll r}
\text{(Q) : maximize} & \displaystyle\sum_{C_j \in E} w_j f_j(x) & & \\
\text{subject to} & \displaystyle\sum_{u \in K} x_0 x_{i,u} = -(k-2) & \text{for } i \in V & \text{(I6)} \\
 & x_{i,u} \in \{-1, +1\} & \text{for } i \in V \text{ and } u \in K & \\
 & x_0 = +1 & &
\end{array}
$$

In this formulation, $f_j(x) = \frac{1}{4}\sum_{u,v} c_j(u,v)\left(1 + x_0 x_{\alpha_j,u}\right)\left(1 + x_0 x_{\beta_j,v}\right)$ encodes the satisfiability of C_j and hence the objective function gives the weight of the assignment. Equation (I6) ensures that every W-CSP variable gets assigned exactly one value. The reason for introducing a dummy variable x_0 is so that all terms occurring in the formulation are quadratic, which is necessary for the subsequent semidefinite programming relaxation.

The essential idea of the semidefinite programming relaxation is to coalesce each quadratic term $x_i x_j$ into a matrix variable $y_{i,j}$. Let Y denote the $(kn+1) \times (kn+1)$ matrix comprising these matrix variables. The resulting relaxed problem (P) is the following:

(P) : maximize	$\sum_{C_j \in E} w_j F_j(Y)$		
subject to	$\sum_{u \in K} y_{0,iu} = -(k-2)$	for $i \in V$	
	$y_{iu,iu} = 1$	for $i \in V$ and $u \in K$	(I7)
	$y_{0,0} = 1$		
	Y symmetric PSD.		

Here, $F_j(Y) = \frac{1}{4} \sum_{u,v} c_j(u,v)(1 + y_{\alpha_j u, \beta_j v} + y_{0,\alpha_j u} + y_{0,\beta_j v})$.

This semidefinite program can be solved in polynomial time within an additive factor (see Alizadeh [1995]). By a well-known theorem in Linear Algebra, a $t \times t$ matrix Y is symmetric positive semidefinite iff there exists a full row-rank matrix $r \times t$ $(r \leq t)$ X such that $Y = X^T X$ (see for example, Lancaster and Tismenetsky [1985]). One such matrix X can be obtained in $O(t^3)$ time by an incomplete Cholesky's decomposition. Since Y has all 1's on its diagonal (by inequality (I7)), the decomposed matrix X corresponds precisely to a list of t unit-vectors $X_1, \ldots, X_t$ which are the t columns of X. Furthermore, these vectors have the nice property that the inner product $X_c \cdot X_{c'} = y_{c,c'}$. Henceforth, for simplicity, we will regard that the program (P) returns a set of vectors X (instead of matrix Y) as the solution.

We propose the following randomized approximation algorithm for the case of $k = 2$ (which can also be used for $k = 3$ as shown later):

1. (Relaxation)
Solve the semidefinite program (P) to optimality (within an additive factor) and obtain an optimal set of vectors X^*.
2. (Rounding)
Construct an assignment for the W-CSP instance as follows.
Assign value u to variable i with probability $1 - \frac{\arccos(X_0^* \cdot X_{i,u}^*)}{\pi}$.

The Rounding step has the following intuitive meaning: the smaller the angle between $X_{i,u}^*$ and X_0^*, the higher the probability that the value u would be assigned to i. Since the vector assignment is constrained by the equation $X_0^* \cdot X_{i,1}^* + X_0^* \cdot X_{i,2}^* = 0$ for all i, the sum of angles between X_0^* and $X_{i,1}^*$ and between X_0^* and $X_{i,2}^*$ must be 180 degrees (or π). Thus, the sum of probabilities of assigning values 1 and 2 to i is exactly 1, implying that the assignment obtained is valid. Furthermore, the variable x_0 is always assigned $+1$.

Before proving the performance guarantee, we present a technical lemma:

LEMMA 1. *For all unit vectors a, b and c,* $b \cdot c \leq \cos(\arccos(a \cdot b) - \arccos(a \cdot c))$.

PROOF. The vectors a, b and c span a unit 3-D sphere. Since the vectors have unit length, the angles between the vectors (denoted $\theta()$) are equal in cardinality to the distance between the respective endpoints on the sphere. Using the form of triangle inequality on spherical distances (see for example, O'Neill [1966],pages 346–347), we get:

$$\theta(b,c) \geq \mid \theta(a,b) - \theta(a,c) \mid .$$

Since the cosine function is monotonically decreasing in the range $[0, \pi]$, the lemma follows. □

CLAIM 2. *The expected weight of this probabilistic assignment is at least* $0.408\times$ *OPT(Q).*

PROOF. The expected weight of the probabilistic assignment is given by,

$$\hat{W} = \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \left[1 - \frac{\arccos(X_0^* \cdot X_{\alpha_j,u}^*)}{\pi} \right] \left[1 - \frac{\arccos(X_0^* \cdot X_{\beta_j,v}^*)}{\pi} \right] \right)$$

Let $p = \frac{\arccos(X_0^* \cdot X_{\alpha_j,u}^*)}{\pi}$, and $q = \frac{\arccos(X_0^* \cdot X_{\beta_j,v}^*)}{\pi}$. One can show that

$$(1-p)(1-q) \geq 0.102 \left[\cos(p\pi - q\pi) + \cos(p\pi) + \cos(q\pi) + 1 \right]$$

in the range $0 \leq p, q \leq 1$ by graph plotting. By Lemma 1, the right-hand-side is at least

$$0.102 \left[X_{\alpha_j,u}^* \cdot X_{\beta_j,v}^* + X_0^* \cdot X_{\alpha_j,u}^* + X_0^* \cdot X_{\beta_j,v}^* + 1 \right].$$

Hence,

$$\hat{W} \geq \frac{0.408}{4} \sum_{C_j \in E} w_j \left(\sum_{u,v \in K} c_j(u,v) \left[X_{\alpha_j,u}^* \cdot X_{\beta_j,v}^* + X_0^* \cdot X_{\alpha_j,u}^* + X_0^* \cdot X_{\beta_j,v}^* + 1 \right] \right)$$

which is equal to $0.408 \times OPT(\mathrm{P})$. □

The above analysis is almost tight, because one cannot achieve a ratio better than 0.5. This can be shown by considering a W-CSP instance in which all constraints are full relations. Here, the optimal solution is W, while a feasible solution of the relaxation (P) where all vectors $X_{i,u}$ are equal and orthogonal to X_0 has objective value $2W$.

For the case of $k = 3$, the technical difficulty is in ensuring that the probabilities of assigning the three values to each variable sum up to exactly 1. Fortunately, by introducing additional valid inequalities, it is possible to enforce this condition, which we will now explain.

Call two vectors X_1 and X_2 *opposite* if $X_1 = -X_2$. The following lemma provides the trick.

LEMMA 2. *Given 4 unit vectors a, b, c, d, if*

$$a \cdot b + a \cdot c + a \cdot d = -1 \tag{6.1}$$

$$b \cdot a + b \cdot c + b \cdot d = -1 \tag{6.2}$$

$$c \cdot a + c \cdot b + c \cdot d = -1 \tag{6.3}$$

$$d \cdot a + d \cdot b + d \cdot c = -1 \tag{6.4}$$

then a, b, c and d must form two pairs of opposite vectors.

PROOF. $\frac{1}{2}[(3) + (4) - (1) - (2)]$ gives:

$$a \cdot b = c \cdot d.$$

Similarly, one can show that $a \cdot c = b \cdot d$ and $a \cdot d = b \cdot c$. This means that they form either two pairs of opposite vectors or two pairs of equal vectors. Suppose we have the latter case, and w.l.o.g., suppose $a = b$ and $c = d$. Then, by (1), $a \cdot c = a \cdot d = -1$, implying that we still have two pairs of opposite vectors (a, c) and (b, d). $\square$

Thus, for $k = 3$, we add the following set of $4n$ valid equations into (Q). For all i:

$$\begin{aligned} x_0(x_{i,1} + x_{i,2} + x_{i,3}) &= -1 \\ x_{i,1}(x_0 + x_{i,2} + x_{i,3}) &= -1 \\ x_{i,2}(x_0 + x_{i,2} + x_{i,3}) &= -1 \\ x_{i,3}(x_0 + x_{i,2} + x_{i,3}) &= -1 \end{aligned}$$

By Lemma 2, the corresponding relaxed problem will return a set of vectors with the property that for each i, there exists at least one vector $\tilde{X} \in \{X_{i,1}, X_{i,2}, X_{i,3}\}$ which is *opposite* to X_0 while the remaining two are opposite to each other. Noting that $1 - \frac{\arccos(X_0 \cdot \tilde{X})}{\pi} = 0$, the sum of probabilities of assigning the other two values to i is exactly 1. Thus, we have reduced the case of $k = 3$ to the case of $k = 2$. The following result follows after derandomization via the method of conditional probabilities:

THEOREM 4. *W-CSP(3) can be approximated within 0.408.*

Note that this ratio is an improvement over the linear programming bound of 0.333.

6.2 Limits of Simple Rounding

The above formulation is inherently weak. We prove by adversary arguments that, with the the above formulation, regardless of the randomized rounding scheme we choose, there exists a W-CSP(2) instance such that the expected weight of the solution is no more than 0.5 times the optimal weight.

Let S be the set consisting of two constraint relations $\{(1,1),(2,2)\}$ and $\{(1,2),(2,1)\}$. Define W-CSP$_S$ to be the collection of W-CSP(2) instances whose constraints are drawn from the set S.

LEMMA 3. *Let $\hat{X}$ be the set of vectors $\{\hat{X}_0\} \bigcup \{\hat{X}_{i,u} : i \in V\ , u \in K\}$ such that all $\hat{X}_{i,u}$'s are equal and orthogonal to $\hat{X}_0$. Then, $\hat{X}$ is an optimal solution for the relaxed problem (P) associated with any instance of W-CSP$_S$.*

PROOF. Consider an arbitrary instance of W-CSP$_S$. For any feasible solution X of the relaxed problem (P), the objective value is,

$$\frac{1}{4} \sum_{C_j \in E} w_j \left(\sum_{u,v \in \{1,2\}} c_j(u,v) \left[1 + X_0 \cdot X_{\alpha_j,u} + X_0 \cdot X_{\beta_j,v} + X_{\alpha_j,u} \cdot X_{\beta_j,v} \right] \right)$$

which is no greater than the total weight W since $X_0 \cdot X_{i,1} = -X_0 \cdot X_{i,2}$ for all i. On the other hand, $\hat{X}$ is a feasible solution of (P) whose objective value is exactly W. □

LEMMA 4. *Let $\{p_{i,u}\}$ be a fixed probabilistic distribution. There exists an instance in W-CSP$_S$ such that the expected weight of the assignment is no more than 0.5 times the optimal weight.*

PROOF. Construct the following W-CSP$_S$ instance. Let the constraint graph be a simple chain connecting n variables. For each constraint C_j connecting variables i and l, let u_{max} (resp. u_{min}) be the value in $\{1,2\}$ such that $p_{i,u}$ is the larger (resp. smaller) quantity, ties broken arbitrarily. Let v_{max} and v_{min} be defined similarly for $p_{l,v}$. Define the constraint relation of C_j to be $\{(u_{max}, v_{min}), (u_{min}, v_{max})\}$, which is an element of S. Now, one can verify by simple arithmetic that the expected weight of the solution is at most 0.5 times the sum of weights. On the other hand, there exists an assignment which can satisfy all constraints simultaneously. □

THEOREM 5. *Using the above semidefinite formulation, W-CSP cannot be approximated by more than 0.5 regardless of randomized rounding scheme, even for $k = 2$.*

PROOF. Given a randomized rounding scheme, let $\{p_{i,u}\}$ be the fixed probability distribution associated with the fixed set of vectors $\hat{X}$. By Lemma 4, we can construct at least one instance I in W-CSP$_S$ for which the probabilistic assignment has expected weight no more than 0.5 times the optimal and by Lemma 3, $\hat{X}$ is an optimal solution of the corresponding relaxed problem (P) of I. The theorem follows if we suppose that $\hat{X}$ is returned by the Relaxation step of the algorithm. □

6.3 Rounding via Hyperplane Partitioning

Several combinatorial problems such as MAX 2SAT, MAX CUT and MAX DICUT are special cases of W-CSP(2) having various types of constraints. Feige and Goemans recently showed that these problems are approximable within the ratios of 0.931, 0.878 and 0.859 respectively. What is nice about their approach is that, given a W-CSP(2) instance, which contains constraints of mixed types, the approximation ratio guaranteed by the rounding scheme for the *most difficult* type of constraints holds simultaneously for all constraints. Hence, they obtained an approximation ratio of 0.859 for W-CSP(2).

The analysis of Feige and Goemans suggests that 1-consistent constraints (i.e. DICUT constraints in their terminology) make the problem harder to approximate. This leads us to consider a form of *parameterized* ratio, namely, if we know that the instance contains a weighted fraction $t < 1$ of 1-consistent constraints (and $1 - t$ of 2-consistent constraints), can we do better? The approach taken by Feige and Goemans seems to suggest that even if the given instance has just *one* (or very few) DICUT constraint, we have to reduce the ratio from 0.878 downto 0.859.

Goemans and Williamson [1994] proposed a nice rounding scheme for approximating MAX 2SAT. This scheme can be adopted to give a parameterized bound for W-CSP(2) which we will now present. The strength of Goemans and Williamson's approach lies in its simplicity. Moreover, it does not involve computation of trigonometric functions (which are heavily used in the rounding scheme of Feige and Goemans), thereby eliminating precision issues in implementation.

Since the domain has only two values, we can directly use the decision variable $x_i \in \{-1, +1\}$ to indicate which value (*false*/*true*) is assigned to variable i. Introduce an additional variable x_0. The value of x_0 will determine whether -1 or $+1$ will correspond to *true* in the W-CSP instance. Model a given instance of W-CSP by the following quadratic program.

$$\begin{array}{lll} \text{Q: maximize} & \displaystyle\sum_{i<l} w_j f_j(x) & \\ \text{subject to} & x_i \in \{-1, +1\} & \text{for } i \in V \bigcup \{0\} \end{array}$$

where $f_j(x)$ encodes the satisfiability of C_j. Table I gives the function f_j associated with all 16 possible constraint relations. We may conveniently ignore those constraints which contain all pairs because they are always satisfied. From the table, we see that (Q) can be expressed as:

$$\text{Q}'\text{: maximize} \sum_{i<l} \left[a_{il}(1 - x_i x_l) + b_{il}(1 + x_i x_l) - c_{il}\right]$$

TABLE I: The first four columns indicate whether each of the four value pairs, namely $(+1,+1),(+1,-1),(-1,+1),(-1,-1)$, is an element of the relation. For simplicity of notation, we assume that constraint C_j relates the variables i and l.

(1)	(2)	(3)	(4)	$f_j(x)$
$\surd$	$\surd$	$\surd$	$\surd$	not a constraint
$\surd$	$\surd$	$\surd$	$\times$	$\frac{1}{4}((1+x_0x_i)+(1+x_0x_l)+(1-x_ix_l))$
$\surd$	$\surd$	$\times$	$\surd$	$\frac{1}{4}((1+x_0x_i)+(1-x_0x_l)+(1+x_ix_l))$
$\surd$	$\times$	$\surd$	$\surd$	$\frac{1}{4}((1-x_0x_i)+(1+x_0x_l)+(1+x_ix_l))$
$\times$	$\surd$	$\surd$	$\surd$	$\frac{1}{4}((1-x_0x_i)+(1-x_0x_l)+(1-x_ix_l))$
$\surd$	$\surd$	$\times$	$\times$	$\frac{1}{2}(1+x_0x_i)$
$\surd$	$\times$	$\surd$	$\times$	$\frac{1}{2}(1-x_0x_i)$
$\surd$	$\times$	$\times$	$\surd$	$\frac{1}{2}(1+x_ix_l)$
$\times$	$\surd$	$\surd$	$\times$	$\frac{1}{2}(1-x_ix_l)$
$\times$	$\surd$	$\times$	$\surd$	$\frac{1}{2}(1-x_0x_l)$
$\times$	$\times$	$\surd$	$\surd$	$\frac{1}{2}(1+x_0x_l)$
$\surd$	$\times$	$\times$	$\times$	$\frac{1}{4}((1+x_0x_i)+(1+x_0x_l)+(1+x_ix_l)-2)$
$\times$	$\surd$	$\times$	$\times$	$\frac{1}{4}((1+x_0x_i)+(1-x_0x_l)+(1+x_ix_l)-2)$
$\times$	$\times$	$\surd$	$\times$	$\frac{1}{4}((1-x_0x_i)+(1+x_0x_l)+(1+x_ix_l)-2)$
$\times$	$\times$	$\times$	$\surd$	$\frac{1}{4}((1-x_0x_i)+(1-x_0x_l)+(1+x_ix_l)-2)$
$\times$	$\times$	$\times$	$\times$	not a constraint

$$\text{subject to} \qquad x_i \in \{-1,+1\} \quad \text{for } i \in V \bigcup \{0\}$$

where the coefficients a_{il}, b_{il} and c_{il} are *non-negative*.

Consider the following rounding scheme proposed in Goemans and Williamson [1994] based on hyperplane partitioning:

1. (Rounding)
Let r be a unit-vector chosen uniformly at random.
Construct an assignment x for (Q$'$) as follows.
For each $i = 0, \ldots, n$, if $r \cdot X_i^* \geq 0$, then set $x_i = +1$ else set $x_i = -1$.
2. (Normalizing)
Construct an assignment for the given W-CSP instance as follows.
If $x_0 = +1$ then return x as the assignment
else ($x_0 = -1$) return x with all values flipped as the assignment.

In the Rounding step, a random hyperplane through the origin of the unit sphere (with r as its normal) is chosen and the variables are partitioned according to those vectors that lie on the same side of the hyperplane. Intuitively, the distance between any two vectors gives a sense of how different their values will be in the W-CSP instance. In the extreme case, if the two vectors are *opposite*, then their corresponding values will always be different. The Normalizing step is needed to undo the effect of the additional variable x_0 in case it is set to -1. More precisely, variable i is assigned $+1$ if $x_i = x_0$ and -1 otherwise, as in the case of Goemans and Williamson [1994].

Let $\hat{W}$ be the expected weight of the assignment returned by the algorithm.

CLAIM 3. $\hat{W} \geq \gamma \left(\sum_{i<l} a_{il}(1 - X_i^* \cdot X_l^*) + \sum_{i<l} b_{il}(1 + X_i^* \cdot X_l^*) \right) - \sum_{i<l} c_{il}$, *where* $\gamma = 0.878\ldots$.

The proof is a direct extension of that given in Goemans and Williamson [1994]. For the sake of completeness, it is given as follows. Let the function $sgn()$ return the sign $(+/-)$ of its argument. In Goemans and Williamson [1994], it has been shown that, for any two vectors X and Y, the probability that $sgn(r \cdot X) = sgn(r \cdot Y)$ (resp. $sgn(r \cdot X) \neq sgn(r \cdot Y)$) is $1 - \frac{\arccos(X \cdot Y)}{\pi}$ (resp. $\frac{\arccos(X \cdot Y)}{\pi}$). Furthermore, the following inequalities were proved: for all $-1 \leq \theta \leq 1$,

$$1 - \frac{\arccos(\theta)}{\pi} \geq \frac{\gamma}{2}(1 + \theta) \quad \text{and} \quad \frac{\arccos(\theta)}{\pi} \geq \frac{\gamma}{2}(1 - \theta).$$

Hence, by linearity of expectation,

$$\begin{aligned}
\hat{W} &= 2\sum_{i<l} a_{il} \Pr[sgn(r \cdot X_i^*) \neq sgn(r \cdot X_l^*)] + \\
&\quad 2\sum_{i<l} b_{il} \Pr[sgn(r \cdot X_i^*) = sgn(r \cdot X_l^*)] - \sum_{i<l} c_{il} \\
&= 2\sum_{i<l} a_{il} \left(\frac{\arccos(X_i^* \cdot X_l^*)}{\pi} \right) + 2\sum_{i<l} b_{il} \left(1 - \frac{\arccos(X_i^* \cdot X_l^*)}{\pi} \right) - \sum_{i<l} c_{il} \\
&\geq \gamma \sum_{i<l} \left(a_{il}(1 - X_i^* \cdot X_l^*) + b_{il}(1 + X_i^* \cdot X_l^*) \right) - \sum_{i<l} c_{il}.
\end{aligned}$$

From this claim, it follows that two subclasses of W-CSP(2), namely, *2-consistent instances* and *satisfiable instances* have approximation ratio γ. The first case follows from the observation that 2-consistent contraints have no constant terms (i.e. c_{il}). For the second case, we can iteratively satisfy all 1-consistent constraints by uniquely fixing the values of their variables. The remaining constraint graph is 2-consistent and still satisfiable, and hence approximable within γ.

Now, consider an instance y which contains 2-consistent constraints *plus* a weighted fraction of t 1-consistent constraints. The latter introduces negative constant terms which will inevitably reduce the approximation ratio. However, observe that each coefficient c_{il} is at most half times the weight of the constraint between i and l and therefore the total contribution of the negative constant terms is at most $\frac{1}{2}tW$.

Let ϕ be the ratio of the optimal value to the total weight (i.e. $\phi = OPT(y)/W$). Hence,

$$\hat{W} \geq \gamma \left(\sum_{i<l} [a_{il}(1 - X_i^* \cdot X_l^*) + b_{il}(1 + X_i^* \cdot X_l^*) - c_{il}] \right) - (1 - \gamma) \sum_{i<l} c_{il}$$

$$\geq \left(\gamma - \frac{(1-\gamma)t}{2\phi}\right) OPT(y)$$

Thus, we obtain a bound of $\gamma - \frac{(1-\gamma)t}{2\phi}$. From Section 4, we learned that the naive random assignment gives an expected weight of at least $sW \geq \frac{2-t}{4W}$, thus giving a ratio of $(2-t)/4\phi$, which is a good bound for small ϕ. Balancing the two bounds, we obtain the approximation ratio for W-CSP(2) instances having a fraction of t 1-consistent constraints as:

$$\theta_t = \min_{(2-t)/4 \leq \phi \leq 1} \left(\max \left(\frac{2-t}{4\phi}, \gamma - \frac{(1-\gamma)t}{2\phi} \right) \right).$$

Numerically, the ratios (to three significant digits) are summarized as follows:

t	θ_t
0.00	0.878
0.01	0.877
0.03	0.875
0.05	0.873
0.07	0.870
0.09	0.868

t	θ_t
0.10	0.867
0.30	0.842
0.50	0.812
0.70	0.776
0.90	0.732
1.00	0.706

Furthermore, we observe that our ratios breakeven with the ratio 0.859 of Feige and Goemans at $t \approx 0.17$. Moreover, we claim a ratio of 0.706 for W-CSP(2) by a simple extension of Goemans and Williamson's technique.

The above algorithm can be derandomized via the technique of Mahajan and Ramesh [1995]. Unfortunately, that technique cannot be efficiently implemented to date In fact, we were told by Mahajan that the worst-case time complexity of the derandomized algorithm is $O(n^{50})$!.

7. Conclusion

We have given new results for the approximation of the Weighted CSP (W-CSP). There remains much room for improving the performance guarantee, especially for small domain sizes. Particularly, we believe that the 0.408 ratio can be improved with better formulation. Hardness results are also interesting to explore. Trevisan [1996] has shown that for any arity $t \geq 11$, W-CSP(2) is not approximable within $2^{-\lfloor t/11 \rfloor}$ unless P=NP. But what about the case $t = 2$?

It would also be interesting to experiment with the proposed techniques. Recently, Goemans and Williamson [1994] applies semidefinite programming to find approximate solutions for the Maximum Cut Problem. Their computational experiments show that, on a number of different types of random graphs, their algorithm yields solutions which are usually within 4% from the optimal solution. In the same vein, we conducted experiments comparing the proposed simple rounding of semidefinite programs with existing approaches. Our implementation indicates that our approach is run-time efficient. It can handle problems of sizes beyond what enumerative search algorithms can handle, and thus is a candidate for solving large-scale real-world problem instances.

Acknowledgements

We thank David Williamson for discovering a fatal error in an early analysis of the semidefinite programming bound; Madhu Sudan for suggesting interactive proofs to obtain the non-approximability result; Luca Trevisan for notifying us the new result of Raz [1995] and its application; Magnús Halldórsson and Carlos Domingo for their constructive comments. We also thank the two referees for suggestions of improvement to the paper, especially for bringing our attention to the results of Feige and Goemans [1995].

References

Alizadeh, F. 1995. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM J. Optimiz. 5*, 1, 13–51.

Alon, Noga and Spencer, Joel. 1992. *The Probabilistic Method.* Wiley Interscience Ser. Disc. Math. and Optimiz.

Amaldi, Edoardo and Kann, Viggo. 1994. On the Approximability of Finding Maximum Feasible Subsystems of Linear Systems. In *Proc. Symp. Theo. Aspects of Comp. Sci. (STACS).* Springer Verlag Lect. Notes. Comp. Sci. (775), 521–532.

Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. 1992. Proof Verification and hardness of Approximation Problems. In *Proc. 33th IEEE Symp. on Found. of Comp. Sci.*, 2–13.

Erdös P. and Selfridge, L. 1989. On a Combinatorial Game. *J. Comb. Theory Series A 14*, 298–301.

Feige, Uriel and Goemans, Michel. 1995. Approximating the Value of Two-Prover Proof Systems, with Applications to MAX 2SAT and MAX DICUT. In *Israel Symp. on Theoretical Computer Science*, 182–189.

Feige, Uriel and Lovász, László. 1992. Two-Prover One-Round Proof Systems: Their Power and Their Problems. In *Proc. 24th ACM Symp. on Theory of Computing*, 733–744.

Fortnow, L., Rompel, J., and Sipser, M. 1988. On the Power of Multiprover Interactive Protocols. In *Proc. Third Annual Conf. on Structures in Complexity Theory*, 156–161.

Freuder, Eugene C. 1989. Partial Constraint Satisfaction. In *Proc. Int'l Joint Conf. Artif. Intell. (IJCAI-89).* Detriot, MI, 278–283.

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

Goemans, Michel X. and Williamson, David P. 1994. Approximation Algorithms for MAX CUT and MAX 2SAT. In *Proc. 26th ACM Symp. on Theory of Computing*, 422–431.

Khanna, S., Motwani, R., Sudan, M., and Vazirani, U. 1994. On Syntactic versus Computational Views of Approximability. In *Proc. 35th IEEE Symp. on Found. of Comp. Sci.*, 819–830.

Lancaster, P. and Tismenetsky, M. 1985. *The Theory of Matrices.* Academic Press, Orlando, FL.

Lau, H. C. 1995. Approximation of Constraint Satisfaction via Local Search. In *Proc. 4th Wrksp. on Algorithms and Data Structures (WADS).* Springer Verlag Lect. Notes

Comp. Sci. (955), 461–472.

MAHAJAN, S. AND RAMESH, H. 1995. Derandomizing Semidefinite Programming Based Approximation Algorithms. In *Proc. 36th IEEE Symp. on Found. of Comp. Sci.*, 162–168.

O'NEILL, BARRETT. 1966. *Elementary Differential Geometry*. Academic Press, New York.

PAPADIMITRIOU, CHRISTOS H. AND YANNAKAKIS, MIHALIS. 1991. Optimization, Approximation, and Complexity Classes. *J. Comput. Sys. Sci. 43*, 425–440.

RAGHAVAN, P. AND THOMPSON, C. D. 1987. Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica* 7, 4, 365–374.

RAZ, RAN. 1995. A Parallel Repetition Theorem. In *Proc. 27th ACM Symp. on Theory of Computing*, 447–456.

TREVISAN, LUCA. 1996. Positive Linear Programming, Parallel Approximation and PCPs. In *Proc. European Symp. on Algorithms (ESA)*. Springer Verlag Lect. Notes Comp. Sci.

TSANG, EDWARD. 1993. *Foundations of Constraint Satisfaction*. Academic Press.

VOUDOURIS, CHRIS AND TSANG, EDWARD. 1995. Partial Constraint Satisfaction Problems and Guided Local Search. Tech. Report No. CSM-250, Dept. Computer Science, University of Essex.

WALLACE, RICHARD J. 1995. Directed Arc Consistency Preprocessing as a strategy for Maximal Constraint Satisfaction. In *Constraint Processing*. Springer Verlag Lect. Notes Comp. Sci. (923), 121–138.

WALLACE, RICHARD J. AND FREUDER, EUGENE C. 1993. Conjunctive Width Heuristics for Maximal Constraint Satisfaction. In *Proc. Nat'l Conf. on Artif. Intell. (AAAI-93)*, 762–768.