# Semidefinite Programming in Randomized Approximation Algorithms

Diego Puppin
diego@mit.edu

December 8, 2000

## Abstract

Here I am presenting briefly some of the main results in the area of semidefinite programming, applied to randomized versions of approximation algorithms for NP-complete problems. Most of the ideas here discussed have been proposed by Goemans and Williamson in their paper *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming* [GW95]. Their technique, actually, resulted in a breakthough in the field, having a strong impact on many successive works.

## 1 Introduction

Semidefinite programming has proved to be a powerful technique, useful to overcome some limitation over linear programming: e.g. it is known that classical linear programming relaxations for MAX CUT can give an optimal value that isas close as desired to $\frac{1}{2}$ of the true maximum cut.

In their paper [GW95], Goemans and Williamson describe a randomized approximation algorithm for the *maximum cut* (MAX CUT) and *maximum 2-satisfiability* (MAX 2SAT) problems, that offers a good (at least .87856 times) approximation of the optimum value, with respect to previously known .5 and .75 approximations, respectively. The main contribution is the use of *semidefinite programming* in an approximation algorithm. The authors prove the technique to be useful also with the *maximum directed cut* (MAX DICUT) and *maximum satisfiability* (MAX SAT) problems, improving the approximation to .79607 and .758 times, respectively, with respect the previously known .25 and .75-approximations, respectively.

This work is organized as follows: section 2 describes the MAX CUT problem; section 3 presents and analyzes the algorithm proposed by Goemans and Williamson; section 4 introduces semidefinite programming (according to Alizadeh [Ali95]), and section 5 explains how it can be used to approximate the MAX CUT; section 6 shows how the introduced technique as proven useful with other problems, and section 7 concludes.

## 2 The MAX CUT problem

The *maximum cut* problem is that of splitting the nodes of a given graph into two sets, $S$ and $\bar{S}$, so to maximize the sum of the weight of the edges crossing the partition, i.e. with one endpoint in $S$ and the other in $\bar{S}$.

If $G$ is an undirected graph $G = (V, E)$ and $w_{ij}$ are the weight of the edges ($w_{ij} = 0$ if $(i, j) \notin E$), the problem can be expressed as:

$$\boxed{\begin{array}{ll} \text{maximize} & \displaystyle\frac{1}{2}\sum_{i<j} w_{ij}(1 - y_i y_j) \\ \text{subject to} & y_i \in \{-1, 1\} \qquad \forall i \in V \end{array}} \tag{1}$$

The problem is known to be NP-complete [Kar72], even in the unweighted version ($w_{ij} = 1$ for all

$(i,j) \in E$). So, the main interest is to find a good $\rho$-approximation of the optimum.

The first result was by Sahni and Gonzalez [SG76], who showed a $\frac{1}{2}$-approximation. Their deterministic algorithm is equivalent in performance to assigning nodes to one or the other side of the partition with random coin flips. This was improved, with small steps, up to a $(\frac{1}{2} + \frac{1}{2\Delta})$-approximation (with $\Delta$ the maximum degree) by Hofmeister and Lefmann [HL95].

In [GW95], a $(\alpha - \varepsilon)$-approximation is presented, with

$$\alpha = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta} > 0.87856$$

and $\varepsilon$ any positive real number. The same approach leads to a $(\alpha - \varepsilon)$-approximation for MAX 2SAT, for which the best known result was a $\frac{3}{4}$-approximation by Yannakakis [Yan76].

In [DP93], the MAX CUT problem is reduced to another problem called eigenvalue maximization (EIG) (the interested reader is invited to refer to the cited paper for more information about EIG). The authors show an example of a graph that gives, with their algorithm, a worst case of a 0.88445-approximation. The algorithm proposed by Goemans and Williamson reaches a worst case of $(\alpha - \varepsilon)$-approximation, very close to the limit just described, showing so to be a very good solution.

It is also interesting to note that MAX CUT, along with MAX 2SAT and MAX DICUT, are known to be MAX SNP-hard, i.e. they are *NP-hard to approximate to within a constant factor* [Kar00]. Bellare showed that this constant is equal to $\frac{83}{84}$ for MAX CUT.

# 3 The algorithm for MAX CUT

If G is a weighted undirected graph, it is easy to see that (1) solves the MAX CUT problem. Let $S = \{i : y_i = 1\}$; the edges $(i,j)$ not in the cut will have $y_i y_j = 1$, and so they will not contribute to the sum $\frac{1}{2}\sum_{i<j} w_{ij}(1 - y_i y_j)$. As said, MAX CUT is NP-complete, so we are interested in an approximate solution, to be obtained with a relaxed problem.

The authors go a bit further with respect to what could be considered a natural relaxation, such as:

$$\boxed{\begin{array}{ll} \text{maximize} & \displaystyle\frac{1}{2}\sum_{i<j} w_{ij}(1 - y_i y_j) \\ \text{subject to} & -1 \leq y_i \leq 1 \qquad \forall i \in V \end{array}}$$

and use instead:

$$\boxed{\begin{array}{ll} \text{maximize} & \displaystyle\frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i \cdot v_j) \\ \text{subject to} & v_i \in S_n \qquad \forall i \in V \end{array}} \tag{2}$$

where $S_n$ is the unit sphere in $\mathbb{R}^n$ ($S_n = \{v \in \mathbb{R}^n : \|v\| = 1\}$).

As usual, the optimum for the relaxation is an upper bound for the original problem. As we will see in section 4, this relaxation can be easily (in polynomial time) solved.

The proposed algorithm is the following:

1. solve problem (2), obtaining a solution $\{v_i\}$;
2. let $r$ a random vector in $S_n$ (uniform distribution);
3. set $S = \{i : v_i \cdot r \geq 0\}$.

So, a random hyperplane (as defined by its normal $r$) is chosen to partition the vectors into two set: vertex $i$ goes to $S$ or $\bar{S}$, according to the fact that $v_i$ lies above or below the plane. This technique makes the solution independent of the coordinate system: any orthonormal transformation of the optimum $\{v_i\}$, gives the same objective value.

Let $W$ be the value of the cut, found this way (after using $r$).

**Theorem 3.1**

$$Pr[sgn(v_i \cdot r) \neq sgn(v_j \cdot r)] = \frac{1}{\pi}\arccos(v_i \cdot v_j)$$

**Proof.** By symmetry, $Pr[sgn(v_i \cdot r) \neq sgn(v_j \cdot r)] = 2Pr[v_i \cdot r \geq 0, v_j \cdot r < 0]$. The set $\{v_i \cdot r \geq 0, v_j \cdot r < 0\}$ is the intersection of two half-spaces, that form a dihedral angle of $\theta = \arccos(v_i \cdot v_j)$. This intersects the sphere $S_n$ in a spherical digon of angle

$\theta$, the area of which is equal to $\frac{\theta}{2\pi}$ times the area of the sphere. So $Pr[v_i \cdot r \geq 0, v_j \cdot r < 0] = \frac{\theta}{2\pi}$ and the conclusion follows. □

**Theorem 3.2**

$$E[W] = \frac{1}{\pi} \sum_{i<j} w_{ij} \arccos(v_i \cdot v_j)$$

**Proof.** We have that

$$W = \sum_{i<j} w_{ij}(sgn(v_i \cdot r) \neq sgn(v_j \cdot r))$$

with $sgn(x) = 1$ if $x \geq 0$, and $-1$ otherwise. So,

$$E[W] = \sum_{i<j} w_{ij} Pr[sgn(v_i \cdot r) \neq sgn(v_j \cdot r)]$$

and the conclusion follows from the previous theorem. □

Let

$$\alpha = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta}$$

we have:

**Theorem 3.3** *For* $-1 \leq y \leq 1$,

$$\frac{\arccos(y)}{\pi} \geq \alpha \frac{1}{2}(1 - y)$$

**Proof.** This can be proved substituting $y = \cos\theta$, and studying the resulting function (see figure 1). □

**Theorem 3.4**

$$E[W] \geq \alpha \frac{1}{2} \sum_{i,j} w_{ij}(1 - v_i \cdot v_j)$$

**Proof.** For theorem 3.2, we have:

$$E[W] = \frac{1}{\pi} \sum_{i<j} w_{ij} \arccos(v_i \cdot v_j)$$

but, using lemma 3.3 applied to $y = v_i \cdot v_j$, we have:

$$\frac{\arccos(v_i \cdot v_j)}{\pi} \geq \alpha \frac{1}{2}(1 - v_i \cdot v_j)$$

The conclusions follows from the fact that $w_{ij}$ are nonnegative. □

With simple calculus, it is possible to prove that:

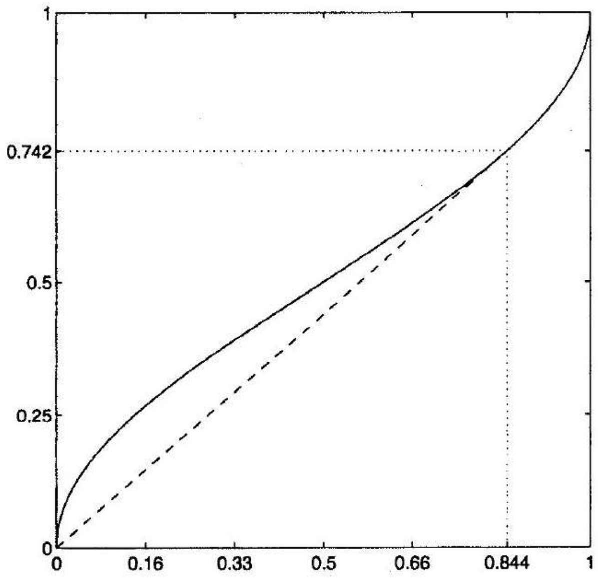


Figure 1: This is a plot of $z = \arccos(y)/\pi$ as a function of $t = \frac{1}{2}(1 - y)$. The ratio $z/t$ is thus the slope of the line between $(0, 0)$ and $(z, t)$. The minimum slope is $\alpha = 0.742/0.44$.

**Theorem 3.5**

$$\alpha > .87856$$

Let $Z_P^* = \frac{1}{2}\sum_{i<j}(1 - v_i \cdot v_j)$ to be the objective value of (2) (relaxed problem), and $Z_{MC}^*$ to be the objective value of (1) (original problem). Clearly $Z_P^* \geq Z_{MC}^*$.

So far, we have proven that

$$E[W] \geq \alpha Z_P^* \geq \alpha Z_{MC}^*$$

showing thus a performance guarantee of $\alpha$ for the discussed algorithm for the MAXCUT problem. As we wills see, we can find, in polynomial time, an $\varepsilon$-approximation for the relaxed problem. This means that we can find a set of vectors that gives an objective value greater than $Z_P^* - \varepsilon$. So, $E[W] \geq \alpha(Z_P^* - \varepsilon) \geq (\alpha - \varepsilon)Z_{MC}^*$.

As a consequence of this analysis, a new formulation of the problem can be proposed, as follows:

$$\boxed{\begin{array}{lll} \text{maximize} & \sum_{i<j} w_{ij} \dfrac{\arccos(v_i \cdot v_j)}{\pi} & \\ \text{subject to} & v_i \in S_n & \forall i \in V \end{array}} \tag{3}$$

**Theorem 3.6** *The optimal value $Z_R^*$ of (3) is equal to the optimal value $Z_{MC}^*$ of (1).*

**Proof.** We note that (3) is a relaxation of (1), as it reduces to (1) if we force the vectors to lie in $S_1$ instead of $S_n$. This implies $Z_R^* \geq Z_{MC}^*$.

Also, from theorem 3.2, we know that the expected value of the solution given by the proposed algorithm is exactly $Z_R^*$, so implying that there exist a cut of value at least $Z_R^*$, and so $Z_R^* \leq Z_{MC}^*$. □

Through reducing the relaxed problem to an eigenvalue maximization problem (EIG), Delorme and Poljak [DP93] showed that there exists a graph (see figure 2) such that:

$$\frac{Z_{MC}^*}{Z_{EIG}^*} = \frac{32}{25 + 5\sqrt{5}} = 0.88445...$$

with $Z_{MC}^*$ the optimal MAXCUT, and $Z_{EIG}^*$ the solution found with eigenvalue maximization. (Nonetheless, a tight bound for this ratio is not known. The only bound we know is just $\frac{1}{2}$.)

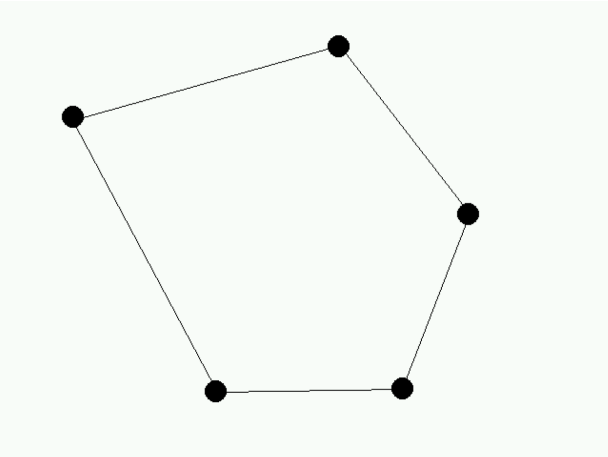

Figure 2: The 5-cycle is a bad case for the eigenvalue maximization

As a consequence of theorem 3.4, we have that, for any graph:

$$\frac{Z_{MC}^*}{Z_P^*} \geq \alpha = 0.878$$

with $Z_P^*$ the solution of the relaxed problem (2). This is so a very good approximation for the worst case. The authors give also some experimental results showing a much better performance (within 9%) in practice.

They also push further the analysis, giving better bounds for the case when the cut is large, and extending it to the case when negative edge weights are allowed.

# 4 Semidefinite programming

A semidefinite programming problem consists of optimizing a linear function of a symmetric matrix, subject to linear equality constraints, and the constraint of the matrix being positive semidefinite.

**Definition 4.1 (Positive semidefinite)** *A matrix is said to be positive semidefinite if for every vector $x \in \mathbb{R}^n$, $x^T Ax \geq 0$.*

**Theorem 4.2** *For a symmetric $n \times n$-matrix $A$, the following statements are equivalent:*

1. *$A$ is positive semidefinite,*

2. *all eigenvalues of $A$ are nonnegative,*

3. *there exists a $m \times n$-matrix $B$ $(m \leq n)$ such that $A = B^T B$.*

Note that there exists an $O(n^3)$ algorithm (Cholesky decomposition) that can determine the matrix B.

Semidefinite programming inherits many results, such as the duality theory, from the broader *linear programming over cones* (*cone-LP*). Simplex methods used in cone-LP can be generalized for semidefinite programming [Ali95].

For a problem of maximum, as (1), a simplex algorithm mainly moves from one vertex of the admissible polytope folllowing the direction that brings to the largest increase in the value of the target function, using also information about the dual problem.

For a simple linear program as:

$$\boxed{\begin{array}{rl} \text{maximize} & b^T y \\ \text{subject to} & y^T A \leq c^T \\ \text{and} & y \geq 0 \end{array}}$$

the dual is:

$$\boxed{\begin{array}{rl} \text{minimize} & c^T x \\ \text{subject to} & Ax \geq b \\ \text{and} & x \geq 0 \end{array}}$$

For more information about the primal-dual techniques, see for instance the interesting talk by Williamson [Wil95].

An interesting result is also that, for any given $\varepsilon > 0$, semidefinite programs can be solved in polynomial time in $\log \frac{1}{\varepsilon}$, within an additive error of $\varepsilon$ (see again [Ali95], among others).

We define, the bullet ($\bullet$) operation as the inner product between matrices ($A \bullet B = \sum_{i,j} A_{ij} B_{ij} = trace(A^T B)$), and we say that the relation $A \succeq B$ holds if $A - B$ is positive semidefinite. Clearly, $A \succeq 0$ iff $A$ is positive semidefinite.

**Definition 4.3 (Semidefinite programming)** *A semidefinite programming problem is defined as:*

$$\boxed{\begin{array}{rll} \textit{minimize} & C \bullet X & \\ \textit{subject to} & A_i \bullet X = b_i & \forall 1 \geq i \geq m \\ \textit{and} & X \succeq 0, X \, \textit{symmetric} & \end{array}} \qquad (4)$$

*where $C$, $A_i$ and $X$ are $n \times n$-matrices.*

If $X$ is diagonal, the semidefinite programming problem reduces to linear programming. Note that there exists an equivalent definition in term of maximization, that will be used in the next section.

Its dual problem is defined as:

$$\boxed{\begin{array}{rl} \text{maximize} & b^T \cdot y \\ \text{subject to} & C - \sum_{i=1}^m y_i A_i \succeq 0 \end{array}} \qquad (5)$$

In [Ali95], we can find the following theorem, describing the relation between primal and dual problem.

**Theorem 4.4** *Let $X$ a feasible matrix for the primal and $y$ a feasible vector for dual. Then $C \bullet X \geq b^T y$.*

Developing this idea, in the paper the author presents an algorithm for the primal problem that finds an $\varepsilon$-approximation in time $O(\sqrt{n} \log(1/\varepsilon))$. The interested reader is invited to refer to it for additional information.

# 5 A semidefinite programming problem for MAX CUT

As seen, for every symmetric positive semidefinite matrix $Y$, there exists a matrix $B$ such that $Y = B^T B$. This means that if $y_{ii} = 1$, the columns $v_1 \ldots v_n$ of $B$ are vectors in $S_n$ (with norm equal to 1), as $Y_{ii} = (B^T B)_{ii} = v_i \cdots v_i = 1$. Using this fact, the relaxed MAX CUT problem can be rewritten as:

$$\boxed{\begin{array}{rll} \text{maximize} & \frac{1}{2} \sum_{i<j} w_{ij}(1 - y_{ij}) & \\ \text{subject to} & y_{ii} = 1 & \forall i \in V \\ \text{and} & Y \succeq 0, Y \text{symmetric} & \end{array}} \qquad (6)$$

This cannot be solved optimally in polynomial time, as its optimal value $Z_P^*$ may be irrational.

Nonetheless, as said, for any given $\varepsilon > 0$, it is possible to find a solution larger than $Z_P^* - \varepsilon$, in time polynomial with the input size and $\log \frac{1}{\varepsilon}$ (see [Ali95]).

Once a solution Y is found, with a Cholesky decomposition we can find the matrix B, and so the vectors $v_i$ such that:

$$\frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i \cdot v_j) \geq Z_P^* - \varepsilon$$

We discuss the dual of (6). First of all, we want the objective function to be symmetric, and so we rewrite it as:

$$\frac{1}{4} \sum_{i,j} w_{ij}(1 - y_{ij})$$

or

$$\frac{1}{2} W_{tot} - \frac{1}{4} \sum_{i,j} w_{ij} y_{ij} = \frac{1}{2} W_{tot} - \frac{1}{4} W \bullet Y$$

with $W_{tot} = \sum_{i<j} w_{ij}$. With some manipulations, we reduce it to the general primal (using an equivalent definition for the maximization problem):

$$\boxed{\begin{array}{rl} \text{maximize} & \frac{1}{2} W_{tot} - \frac{1}{4} W \bullet Y \\ \text{subject to} & A_i \bullet Y = 1 \\ \text{and} & Y \succeq 0 \end{array}}$$

with $(A_i)_{jk} = 1$ if $j = k$, and 0 otherwise. With respect to the general problem (4), here $b_i = 1$, $C$ is $W$, $X$ is $Y$. Now, we can express the dual as:

$$\boxed{\begin{array}{rl} \text{minimize} & \frac{1}{2} W_{tot} + \frac{1}{4} \sum_i \gamma_i \\ \text{subject to} & W + \sum \gamma_i A_i = W + diag(\gamma) \succeq 0 \end{array}} \qquad (7)$$

where $diag(\gamma)$ is a diagonal matrix, whose entries are $\gamma_i$. The optimal solution for (7) $Z_D^*$ is, for the theory of duality, larger than or equal to the optimal value for (2) $Z_P^*$. Nonetheless, in [Ali95], it is proven that, in this case, $Z_P^* = Z_D^*$.

Now that the problem is in the form primal-dual semidefinite programming, we can use Alizadeh's algorithm to find an $\varepsilon$-approximation in polynomial time.

# 6 Other examples of semidefinite programming formulations

## 6.1 MAX RES CUT

The proposed algorithm for MAX CUT is extended, by the original authors, to MAX RES CUT, i.e. the problem of finding the maximum cut in a graph with the further constraint that pairs of vertices are forced to be either on the same side of the cut, or on different sides. It is enough to add some constraints $v_i \cdot v_j = 1$ (respectively $-1$) for the pairs $(i, j)$ that are forced to be on the same (respectively different) side, to the relaxed problem (2). When we build $S$ and $\bar{S}$ as described in the algorithm, we are guaranteed that the requirements will be met: if $v_i \cdot v_j = 1$, certainly $y_i y_j = 1$, as $v_i = v_j$. If $v_i \cdot v_j = -1$, the only case in which $y_i y_j \neq -1$ is when $v_i \cdot r = v_j \cdot r = 0$, but this happens with probability equal to 0.

So, the same analysis is still valid, and the algorithm offers a $(\alpha - \varepsilon)$-approximation to MAX RES CUT.

## 6.2 MAX 2SAT

We can define the MAX 2SAT problem as follows: given a collection $C$ of Boolean clauses, each one associated with a nonnegative weight, in the form of disjunctions of at most two literals, the goal is to choose an assignment of the variables $\{x_1 \dots x_n\}$ so to maximize the sum of the weights of the satisfied clauses. MAX 2SAT is known to be NP-complete [GJS76]. The best solution known was a $\frac{3}{4}$-approximation by Yannakakis [Yan76]. As said, MAX 2 SAT is MAX SNP-hard; Bellare showed that MAX 2SAT cannot be approximated better than up to $\frac{95}{96}$, unless $P = NP$.

MAX 2SAT can be modeled as a quadratic program of the type:

$$\boxed{\begin{array}{rl} \text{maximize} & \sum_{i<j} \left( a_{ij}(1 - y_i y_j) + b_{ij}(1 + y_i y_j) \right) \\ \text{subject to} & y_i \in \{-1, 1\} \qquad \forall 0 \leq 1 \leq n \end{array}}$$

with $a_{ij}$ and $b_{ij}$ nonnegative. The problem is modeled defining a variable $y_i$ for every variable $x_i$ in the problem, plus another variable $y_0$, with the property that $x_i$ will be true in the final assignment iff $y_0 = y_i$.

We define $v(\phi) = 1$ if $\phi$ is true, 0 otherwise. We have:

$$v(x_i) = \frac{1 + y_0 y_i}{2}$$

$$v(\bar{x_i}) = 1 - v(x_i) = \frac{1 - y_0 y_i}{2}$$

$$\begin{aligned} v(x_i \vee x_j) &= 1 - v(\bar{x_i} \wedge \bar{x_j}) \\ &= 1 - v(\bar{x_i}) v(\bar{x_j}) \\ &= 1 - \frac{1 - y_0 y_i}{2} \frac{1 - y_0 y_j}{2} \\ &= \frac{1}{4}(3 + y_0 y_i + y_0 y_j + y_0^2 y_i y_j) \\ &= \frac{1 + y_0 y_i}{4} + \frac{1 + y_0 y_j}{4} + \frac{1 - y_i y_j}{4} \end{aligned}$$

Similarly, we can express all the clause with up to two literals as expression of $y_i$s. So, MAX 2SAT can be expressed as:

$$\boxed{\begin{array}{lll} \text{maximize} & \sum_{c_i \in C} w_i v(c_i) & \\ \text{subject to} & y_i \in \{-1, 1\} & \forall 0 \leq i \leq n \end{array}} \quad (8)$$

with $v(c_i)$ nonnegative linear combinations of $1 \pm y_i y_j$. We can relax it, as we did with MAX CUT:

$$\boxed{\begin{array}{ll} \text{maximize} & \sum_{i<j} \left(a_{ij}(1 - v_i \cdot v_j) + b_{ij}(1 + v_i \cdot v_j)\right) \\ \text{subject to} & v_i \in S_n \qquad \forall i \in V \end{array}} \quad (9)$$

Using the same analysis for MAX CUT, we have:

$$\begin{aligned} E[V] \quad = \quad & 2 \sum_{i<j} a_{ij} Pr[sgn(v_i \cdot r) \neq sgn(v_j \cdot r)] \\ & + 2 \sum_{i<j} b_{ij} Pr[sgn(v_i \cdot r) = sgn(v_j \cdot r)] \end{aligned}$$

and

$$Pr[sgn(v_i \cdot r) = sgn(v_j \cdot r)] = 1 - \frac{\arccos(v_i \cdot v_j)}{\pi}$$

hence:

**Theorem 6.1**

$$E[V] \geq \alpha \sum_{i<j} \left(a_{ij}(1 - v_i \cdot v_j) + b_{ij}(1 + v_i \cdot v_j)\right)$$

The authors also extend their results to present solutions to MAX SAT and MAX DICUT, but, to my opinion, they are too technical, and not interesting for the purpose of this work.

## 6.3 Graph coloring

Using the technique proposed by Goemans and Williamson, other authors could develop new interesting solution to some well-known problems.

Karger et al. [KMS98] showed a technique to color a $k$-colorable graph with $\min\{O(\Delta^{1-2/k} \log^{1/2} \Delta \log n), O(n^{1-3/(k+1)} \log^{1/2} n)\}$ colors in polynomial time (with $\Delta$ maximum degree of any node in the graph).

They do that relaxing the problem, defining the *vector $k$-coloring* of a graph.

**Definition 6.2 (Vector $k$-coloring)** *Given a graph $G = (V, E)$ on $n$ vertices, and a real number $k \geq 1$, a vector $k$-coloring of $G$ is an assignment of unit vectors $v_i$ from the space $\mathbb{R}^n$ to each vertex $i \in V$, such that for any two adjacent vertices $i$ and $j$, we have:*

$$v_i \cdot v_j \leq -\frac{1}{k-1}$$

They prove that vector $k$-coloring can be found in time polynomial in $k$, $n$ and $\log \frac{1}{\varepsilon}$, solving a suitable semidefinite program. They present two techniques for rounding the solution, the first of which is analog to the one proposed by Goemans and Williamson: they partition the vectors with $r$ random hyperplanes, assigning the same color to nodes $i$ and $j$ if $v_i$ and $v_j$ are in the same region, with respect to the hyperplanes.

## 6.4 Scheduling

In his paper [Sku00], Skutella applies the authors' results to a scheduling problem. The problem he face is a scheduling problem of the type $R||\sum w_j C_j$, i.e. unrelated parallel machines, optimized for total weighted completion time, in absence of nontrivial release dates. He could represent the problem as an integer quadratic program:

$$\boxed{\begin{array}{rl} \text{minimize} & \sum_{j \in J} w_j C_j \\ \text{subject to} & \sum_{i=1}^{m} a_{ij} = 1 \qquad \forall j \\ \text{and} & C_j = \sum_{i=1}^{m} a_{ij} \left( p_{ij} + \sum_{k \prec_i j} a_{ik} p_{ik} \right) \qquad \forall j \\ \text{and} & a_{ij} \in \{0,1\} \qquad \forall i,j \end{array}} \tag{10}$$

that, for the case of two machines, is relaxed by the author to the following vector program:

$$\boxed{\begin{array}{rl} \text{minimize} & Z \\ \text{subject to} & Z \geq \sum_{j} w_j \sum_{i=-1}^{0} \left( \frac{1 + v_i v_j}{2} p_{ij} + \right. \\ & \left. + \sum_{k \prec_i j} \frac{v_j v_k + v_i v_j + v_i v_k + 1}{4} p_{ik} \right) \\ \text{and} & v_{-1} \cdot v_0 = -1 \\ \text{and} & v_i \in S_n \qquad \forall j \in J \cup \{0,-1\} \end{array}} \tag{11}$$

that is solved and then rounded with the hyperplane technique discussed above. This led to a 1.2752-approximation of the optimum.

### 6.5 Other works

We want also to cite Frieze and Jerrum [FJ00], that applied the discussed technique to develop an algorithm for the maximum $k$-way cut problem, and Chor and Sudan [CS95] that used it to solve the *betweenness* problem, all of them reaching provably improved performance figures.

## 7 Conclusion

In [GW95], it is shown how semidefinite programming can help to get tighter relaxations to some problem, usually faced with linear programming (LP): this new technique overcomes many limits of LP (it is known that classical linear programming relaxations for MAX CUT can give optimal value that is as close as desired to $\frac{1}{2}$ of the true maximum cut).

They present two open questions.

1. Is it possible to find the $\alpha$-approximation, without explicitly solving the semidefinite programming? See for instance the interesting result by Young [You95], that proposed a way to the rounded solution of linear programming problems without solving them: he used complex pessimistic estimators to build a very efficient algorithm.

2. Is it possible to have tighter bounds to the solution, adding constraints to the problem? This is known to be true for some cases of linear programming.

Another theoretical question is how close we can get to the constant for MAX SNP-hard problems.

As shown, the technique here discussed had a strong impact in many successive works, that showed successful application of it in many different areas, such as scheduling, coloring, and computational geometry.

## References


[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal of Optimization*, 5(1):13–51, 1995.

[CS95] B. Chor and M. Sudan. A geometric approach to betweenness. Algorithms - ESA '95, pages 227–237, 1995.

[DP93] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a maxcut approximation. *European Journal of Combinatorics*, 14:313–333, 1993.

[FJ00] A. Frieze and M. Jerrum. Improved approximation algorithms for max $k$-cut and max bisection. *Algorithmica*, 2000. To appear.

[GJS76] M. Garey, D. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[HL95] T. Hofmeister and H. Lefmann. A combinatorial design approach to maxcut. In *Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science.* 1995.

[Kar72] R. M. Karp. Reducibility among combinatorial programs. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[Kar00] David Karger. 2000. Personal communication :-).

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, March 1998.

[SG76] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, July 1976.

[Sku00] Martin Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *Journal of the ACM*, 2000. To appear.

[Wil95] David P. Williamson. Primal-dual methods. *Course on Approximation Algorithms at IBM Almaden*, October 1995. Available at http://www.research.ibm.com/ people/w/williamson/ Almaden/primal-dual.ps.

[Yan76] Y. Yannakakis. On the approximation of maximum satisfiability. *Journal of Algorithms*, 17:475–502, 1976.

[You95] Neal E. Young. Randomized rounding without solving the linear program. 6th Annual Symposium on Discrete Algorithms, pages 170–178, 1995.