7e Conférence Internationale de MOdélisation et SIMulation - MOSIM'08 - du 31 mars au 2 avril 2008 - Paris - France
«Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination»

SDP AND LAGRANGIAN RELAXATIONS FOR MAXCUT AND CDMA PROBLEMS

A. LISSER, R. LOPEZ

LRI
Université Paris-Sud
91405 Orsay cedex
lisser@lri.fr, lopez@lri.fr

ABSTRACT : *In this paper we compare the results obtained with a SDP relaxation and with the more general Lagrangian form over MaxCut problems taken from SDPLIB or generated from a particular telecommunication problem: multiuser detection in CDMA channels. We show that both formulations lead to very close results, with the spectral bundle method showing some small improvements over known best bounds (SDPLIB), and the Lagrangian method performing better in terms of computation time. We also show that a simple analytical method can be applied in the CDMA cases, which under some conditions gives results close to the SDP and Lagrangian methods, but with a very short computation time.*

KEYWORDS : *Semidefinite Programming, Relaxation, MaxCut, CDMA*

INTRODUCTION

Solving MaxCut problems has been the subject of research for a long time. Recently, solving combinatorial optimization problems such as this one, using semidefinite programming, has received increased interest. One of the first major work was by Lovász (Lovász 1979) on the so-called theta function. This work allowed Grötchel, Lovász and Schrijver (Grötschel, Lovász & Schrijver 1984) to be able to solve the maximum stable set problem for perfect graphs using a polynomial time algorithm they developed. More recently, Goemans and Williamson (Goemans & Williamson 1995) gave a famous improved approximation algorithm for the MaxCut problem (and related problems) using semidefinite programming. Furthermore, benchmarks were created for the purpose of testing the various softwares released using the SDP relaxation, such as SDPLIB (Borchers 1999b). In the past years, Lemarechal and Oustry (Lemaréchal & Oustry 1999) linked the SDP relaxation of some classes of problems to the more general Lagrangian formulation.

One interesting application of the MaxCut problem is in the telecommunication field. In order to allow new services to be provided, like video streaming and music streaming on mobile phones, a good signal quality is required, and therefore as few errors as possible between what is transmitted and what is received. The third generation networks (3G, which can be based on several systems, like UMTS[1] or CDMA2000[2]) use a coding method that lowers the error rate. A crucial step is at the reception by the base, when the signals coming from the various users are detected. Since all of them are emitted simultaneously, their signal is received as a whole by the base, which must identify each individual signal. Several methods of modulation/demodulation exist.

The demodulation and detection phase is important to keep the communication quality. It is also a complex phase, since the signal is not received perfectly: it can be subject to noise, fading and Doppler effects, and the users are not sending their signals synchronized. It has been shown (Verdú 1998) that in general this problem is NP-hard. In particular, it can be formulated as a binary quadratic problem, or as a particular MaxCut problem. (Tan & Rasmussen 2001)

It seemed interesting to us to compare the results we obtain from a straight SDP formulation and Lagrangian formulation of both the SDPLIB MaxCut problems and over randomly generated instances of CDMA problems, for two softwares: Sbmethod (Helmberg 2000) and CSDP (Borchers 1999a).

In section 1, we introduce the notations and formulations of both MaxCut and CDMA problems. In section 2 we detail the relaxations that allow us to find upper (resp. lower) bounds for the MaxCut (resp. CDMA) problems. Finally, in section 3 we present the results we obtain.

[1]UMTS: *Universal Mobile Telecom. Services*
[2]CDMA: *Code Division Multiple Access*

1. FORMULATION

In this section, we present the MaxCut problem, and the CDMA model. An interesting aspect of the CDMA problem is that it can be transformed into a particular MaxCut problem. Both problems are difficult, in fact, MaxCut is part of Karp's 21 NP-complete problems (Karp 1972). Furthermore, Håstad proved in (Håstad 1997) that MaxCut is not approximable within 1.0624.

1.1. MaxCut problem

Given an undirected weighted graph $G = (V, E)$ (where V is the set of vertices, E the set of edges), and W the matrix of weights on the edges, we try to find the maximum cut (partition) $C = (V1, V2)$ of the vertices V, such that the sum of the weights of the edges on the cut is maximized. Formally, we can note $w_{i,j}$ the weight of the edge between vertices i and j. Let $x_i = \begin{cases} 1 & \text{if } x_i \in V1 \\ -1 & \text{if } x_i \in V2 \end{cases}$. The MaxCut problem is NP-Complete in the general case, and can be formulated as follows:

$$\max_{x \in \{-1,1\}} \frac{1}{2} \sum_{i<j \in V} w_{i,j}(1 - x_i x_j) \tag{1}$$

This is equivalent to:

$$\max_{x \in \{-1,1\}} \frac{1}{4} \sum_{i,j \in V} w_{i,j}(1 - x_i x_j) \tag{2}$$

Using vector/matrix notation, we get:

$$\max_{x \in \{-1,1\}} \frac{1}{4} \mathbf{x}^t (\text{diag}(\mathbf{We}) - \mathbf{W})\mathbf{x} \tag{3}$$

where $\mathbf{e}$ is the vector of all ones, and $\text{diag}(\mathbf{x})$ the diagonal matrix made from the values of vector $\mathbf{x}$. $\mathbf{W}$ is the matrix of the weights between the nodes, and $w_{i,j} = 0$ if there is no edge between vertices i and j. Let $\overline{\mathbf{W}} = \text{diag}(\mathbf{We}) - \mathbf{W}$. (3) can be written as:

$$\max_{x \in \{-1,1\}} \frac{1}{4} \mathbf{x}^t \overline{\mathbf{W}} \mathbf{x} = \max \frac{1}{4} \text{tr}\{\overline{\mathbf{W}} \mathbf{x}\mathbf{x}^t\} \tag{4}$$

We will now present the CDMA model, and how it can be formulated as a quadratic $\{-1, 1\}$ program. Our interest in the CDMA model is twofold: first, it is widely used in mobile telecommunication networks, and improvements in this particular aspect (multiuser detection) means improving the quality of the service for the users (or raising the number of concurrent users); and second, it can be seen as a particular MaxCut problem for which we can use known methods, and compare the results obtained here with the results in the general MaxCut case.

1.2. CDMA model

The CDMA model can be reformulated as a particular case of the MaxCut problem.

We consider a CDMA channel with K simultaneous users, each of them is given a signature $p_k(t)$ of length T, where T is the symbol interval. We suppose, without loss of generality, that each of the K signatures has unit energy ($\|p_k\|^2 = \int_0^T p_k^2(t)dt = 1$).
We note the data sequence transmitted by user k as $\{d_k(m)\}$; the value of each symbol can be chosen uniformly in the set $\mathcal{D}$ of the possible symbols. All the data sequences have the same probability and each symbol is statistically independent from the others, as well as between users. We consider a sequence of arbitrary length L. We note $s(t)$ the composite signal transmitted for the K users. The transmitted signal is assumed to be corrupted by additive white gaussian noise (AWGN). Consequently, the received signal may be formulated as $r(t) = s(t) + n(t)$, where $n(t)$ is the noise, with power spectral density σ^2. Using vector notation, the LK Matched Filters (MF) outputs can be written as (Verdú 1998):

$$\mathbf{y} = \mathbf{RCd} + \mathbf{n} \tag{5}$$

where $\mathbf{C}$ is the matrix of the channel coefficients of the users, $\mathbf{R}$ is the $KL \times KL$ matrix of crosscorrelations, $\mathbf{n}$ is the gaussian noise sequence vector, and has zero mean and its autocorrelation matrix (with $\sigma^2 = N_0/2$, the noise spectral density) is:

$$E\left[\mathbf{n}\mathbf{n}^H\right] = \sigma^2 \mathbf{R}$$

The optimum Maximum Likelihood (ML) detector chooses the hypothesis $\hat{\mathbf{d}}$ given the MF output, and assumes perfect knowledge[3] of $\mathbf{C}$ and of $\mathbf{R}$:

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d} \in \mathcal{D}} p(\mathbf{y} \,|\, \mathbf{d}).$$

Since we are working with an AWGN channel, the log-likelihood function based on $p(\mathbf{y} \,|\, \mathbf{d})$ may be written (Tan & Rasmussen 2001):

$$F(\mathbf{d}) = 2\text{Re}\{\mathbf{y}^H \mathbf{Cd}\} - \mathbf{d}^H \mathbf{C}^H \mathbf{RCd}.$$

The constrained ML problem associated with the negative log-likelihood function is described as:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d} \in \mathcal{D}^{KL}} \left\{\mathbf{d}^H \mathbf{C}^H \mathbf{RCd} - 2\text{Re}\{\mathbf{y}^H \mathbf{Cd}\}\right\} \tag{6}$$

Solving (6) requires a search over the $\mathcal{D}^{KL}$ possible combinations of the components of $\mathbf{d}$. Even when considering a single symbol interval (synchronous channel), we still have to search over $\mathcal{D}^K$ possibilities.

[3] The model used here is that of Tan and Rasmussen (Tan & Rasmussen 2001), in particular, we make the same assumptions concerning the knowledge of the channel coefficients, crosscorrelations, and synchronization.

It is clear that computational complexity grows exponentially with the number of users. Verdú showed in (Verdú 1986) that the complexity, however, does not necessarily grow exponentially with the block length L.

In the remainder of this paper, we will only consider synchronous CDMA systems. In this case, $\tau_k = 0$ and each interfering user produces exactly one symbol which interferes with the desired symbol. In single path AWGN channels, it is sufficient to consider the signal received during one signal interval. When the channel coefficients are real valued and the symbols $\mathbf{d}$ are binary ($\mathcal{D} \in \{-1, 1\}$), the optimization problem (6) can be rewritten as:

$$\hat{\mathbf{d}} = \arg \min_{\mathbf{d}\in\{\pm1\}^K} \left\{\mathbf{d}^T\mathbf{C}^T\mathbf{R}\mathbf{C}\mathbf{d} - 2\mathbf{y}^T\mathbf{C}\mathbf{d}\right\} \qquad (7)$$

We will use the real channel - binary symbol formulation, with $\mathbf{Q} = \mathbf{C}^T\mathbf{R}\mathbf{C}$, $\mathbf{c} = \mathbf{C}\mathbf{y}$ and $\mathbf{u} = \hat{\mathbf{d}}$. We get:

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{Q}\mathbf{u} - 2\mathbf{c}^T\mathbf{u}\right\} \\ &\quad \text{s.t. } \mathbf{u} \in \{-1, 1\}^K \end{aligned} \qquad (8)$$

Both the MaxCut and CDMA problems are difficult to solve with a brute force approach (testing all the possible solutions). Therefore, there exist several methods to derive an upper (or lower, in the CDMA case) bound of the optimum. We will present some of these relaxations, and then compare their performances.

2. Bounds returned by the various relaxations

In the particular case of CDMA problems, the problem can be rewritten in its spectral decomposition form, and solved analytically. The other relaxations can be applied to both MaxCut and CDMA problems: the linear relaxation, the Semidefinite (SDP) relaxation, and the Lagrangian relaxation.

2.1. Spectral Decomposition

In (Lemaréchal & Oustry 1999), Lemaréchal and Oustry showed that there is an equivalence between the Semi-definite relaxation and the Lagrangian relaxation for the MAXCUT problem. Furthermore, they give some results related to quadratic problems, in the form of (8). This Lagrangian relaxation problem has a finite lower bound under the conditions presented by the authors:

$$\mathbf{Q} \succcurlyeq 0 \text{ and } -2\mathbf{c} \in \text{range}(\mathbf{Q})$$

In this case, we can use the spectral decomposition of $\mathbf{Q}$: $\mathbf{Q} = \sum_{i=1}^{p-k} \lambda_i \mathbf{q}_i \mathbf{q}_i^T$, where $0 \leqslant k \leqslant p$ is the rank of $\mathbf{Q}$, $\lambda_1 \ldots \lambda_{p-k}$ 0 and $\mathbf{q}_1, \ldots, \mathbf{q}_{p-k}$ are othonormal (eigen)vectors forming a basis of range($\mathbf{Q}$). $-2\mathbf{c} \in$ range($\mathbf{Q}$) means that the linear system $2\mathbf{Q}\mathbf{u} = 2\mathbf{c}$ has at least one solution. The pseudo-inverse $\mathbf{Q}^\dagger$ of $\mathbf{Q}$ is then:

$$\mathbf{Q}^\dagger = \sum_{i=1}^{p-k} \lambda_i^{-1} \mathbf{q}_i \mathbf{q}_i^T$$

The following problem is a relaxation of (8) :

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{Q}\mathbf{u} - 2\mathbf{c}^T\mathbf{u}\right\} \\ &\quad \text{s.t. } \mathbf{u} \in \mathbb{R}^K \end{aligned} \qquad (9)$$

Under the above conditions, the $\bar{\mathbf{u}} \in \mathbb{R}^K$ which minimizes (9) is ((Lemaréchal & Oustry 1999)):

$$\bar{\mathbf{u}} = -\frac{1}{2}\mathbf{Q}^\dagger(-2\mathbf{c}) = \mathbf{Q}^\dagger\mathbf{c}$$

2.2. Linear relaxation and linearization techniques

The idea of this relaxation is to replace the quadratic terms of the problem with linear terms, and to add linearization constraints. More details can be found in (Boros & Hammer 2002) and in (Padberg 1989). In this case, we write: $y_{i,j} = x_i x_j \ \forall i, j \in 1..K$ ($x_i = u_i \ \forall i \in 1..K$ in the CDMA case). We use the matrix $\mathbf{X}$ written as:

$$\mathbf{X}_{lin} = \begin{bmatrix} \mathbf{Y} & \mathbf{x} \\ \mathbf{x}^T & 1 \end{bmatrix}$$

The problem becomes:

$$\begin{aligned} \mathbf{X}^* = \arg \max_{\mathbf{X}_{lin}} \ & \text{tr}\{\overline{\mathbf{W}}\mathbf{X}_{lin}\} \\ \text{s.t.} \quad & x_i \in [-1, 1] \quad \forall i \in 1..K \\ & y_{i,j} \leq x_j \quad \forall i, j \in 1..K \\ & x_i + x_j - y_{i,j} \leq 1 \quad \forall i, j \in 1..K \end{aligned}$$

Or, using matrix/vector notation:

$$\begin{aligned} \mathbf{X}^* = \arg \max_{\mathbf{X}_{lin}} \ & \text{tr}\{\overline{\mathbf{W}}\mathbf{X}_{lin}\} \\ \text{s.t.} \quad & \mathbf{x} \in [-1, 1]^K \\ & \text{tr}\left\{\mathbf{W}^a_{i,j}\mathbf{X}_{lin}\right\} \leq 0 \quad \forall i, j \in \{1, ..., K\} \\ & \text{tr}\left\{\mathbf{W}^b_{i,j}\mathbf{X}_{lin}\right\} \leq 1 \quad \forall i, j \in \{1, ..., K\} \end{aligned} \qquad (10)$$

where $\mathbf{W}^a_{i,j}$ is given $\forall i, j \in \{1, ..., K\}$ by:

$$\text{If } i \neq j, \begin{cases} w^a_{i,j} = w^a_{j,i} = \dfrac{1}{2} \\ w^a_{i,K+1} = w^a_{K+1,i} = -\dfrac{1}{2} \end{cases}$$

$$\text{If } i = j, \begin{cases} w^a_{i,i} = 1 \\ w^a_{i,K+1} = w^a_{K+1,i} = -\dfrac{1}{2} \end{cases}$$

And $\forall i, j \in \{1, ..., K\}, \mathbf{W}^b_{i,j}$ is:

$$\text{If } i \neq j, \begin{cases} w^b_{i,j} = w^b_{j,i} = \frac{1}{2} \\ w^b_{i,K+1} = w^b_{K+1,i} = w^b_{j,K+1} \\ \qquad\qquad = w^b_{K+1,j} = -\frac{1}{2} \end{cases}$$

$$\text{If } i = j, \begin{cases} w^b_{i,i} = 1 \\ w^b_{i,K+1} = w^b_{K+1,i} = -1 \end{cases}$$

2.3. Semidefinite relaxation

We now reformulate the MaxCut problem (4). For all $\mathbf{x} \in \{-1, 1\}^K$, the matrix $\mathbf{x}\mathbf{x}^T$ is positive semidefinite, its diagonal elements are equal to 1, and is a rank-1 matrix. Let $\mathbf{X} = \mathbf{x}\mathbf{x}^T$, such that $\mathbf{X}$ satisfies the properties listed above. We obtain:

$$\max \frac{1}{4}\text{tr}\{\overline{\mathbf{W}}\mathbf{x}\mathbf{x}^T\} = \max \frac{1}{4}\text{tr}\{\overline{\mathbf{W}}\mathbf{X}\} \qquad (11)$$

The semidefinite form of the MaxCut problem is:

$$\begin{aligned} &\mathbf{X}^* = \arg\max_{\mathbf{X}} \ \text{tr}\{\overline{\mathbf{W}}\mathbf{X}\} \\ &\text{s.t.} \quad \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \text{rank}(\mathbf{X}) = 1, \quad \mathbf{X} \succeq 0. \end{aligned} \qquad (12)$$

where diag($\mathbf{X}$) is the vector of diagonal elements of $\mathbf{X}$ and $\mathbf{e}_n$ the all ones vector of length n. $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is positive semidefinite. The equivalence between (11) and (12) is shown in general in (Laurent & Poljak 1995).
Similarly, we can rewrite the CDMA problem in SDP form:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{x}^T\mathbf{L}\mathbf{x} \quad \text{s.t. } \mathbf{x} \in \{-1, 1\}^N. \qquad (13)$$

And a semi-definite relaxation is :

$$\begin{aligned} &\mathbf{X}^* = \arg\ \min_{\mathbf{X}} \ \text{tr}\{\mathbf{L}\mathbf{X}\} \\ &\text{s.t.} \quad \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \text{rank}(\mathbf{X}) = 1, \quad \mathbf{X} \succeq 0. \end{aligned} \qquad (14)$$

If we relax the rank($\mathbf{X}$) = 1 constraint, we get the following semidefinite programs:

$$\begin{aligned} &\text{(MaxCut)} \quad \mathbf{X}^* = \arg\max_{\mathbf{X}} \ \text{tr}\{\overline{\mathbf{W}}\mathbf{X}\} \\ &\qquad\qquad\quad \text{s.t. diag}(\mathbf{X}) = \mathbf{e}_n, \quad \mathbf{X} \succeq 0 \\ &\text{(CDMA)} \quad \mathbf{X}^* = \arg\min_{\mathbf{X}} \ \text{tr}\{\mathbf{L}\mathbf{X}\} \\ &\qquad\qquad\quad \text{s.t. diag}(\mathbf{X}) = \mathbf{e}_n, \quad \mathbf{X} \succeq 0 \end{aligned} \qquad (15)$$

with diag($\mathbf{X}$) the vector of diagonal elements of $\mathbf{X}$.

There are well known results for the MaxCut problem, in particular regarding the use of the semidefinite relaxation. Goemans and Williamson (Goemans & Williamson 1995) showed that there exist an algorithm with a 0.87856-approximation guarantee. The algorithm can be described as follows:

Let $\mathbf{X}$ be the solution of (15), $\mathbf{X}$ can be decomposed into $\mathbf{Y}^t\mathbf{Y}$. From $\mathbf{Y}$, we can extract the vectors $\mathbf{y}_i$. Let $\mathbf{u}$ be a random vector, uniformly distributed over the unit sphere of dimension n. For each vector $\mathbf{y}_i$, calculate $\mathbf{y}_i.\mathbf{u}$. If $\mathbf{y}_i.\mathbf{u} \geq 0$, then set $x_i = 1$, otherwise, set $x_i = -1$.

2.4. Lagrangian relaxation

Based on the work of (Lemaréchal & Oustry 1999), we can use (12) to form the Lagrangian. The objective function is:

$$\begin{aligned} &f(\mathbf{x}) = \mathbf{x}^T\overline{\mathbf{W}}\mathbf{x} \\ &\text{s.t. } x_i^2 = 1 \ \forall i \in 1..N \end{aligned} \qquad (16)$$

We can then formulate the Lagrangian $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{x}^T\overline{\mathbf{W}}\mathbf{x} + \sum_{i=1}^{N} \lambda_i(x_i^2 - 1) \\ &= \mathbf{x}^T(\overline{\mathbf{W}} + D(\boldsymbol{\lambda}))\mathbf{x} - e^T\boldsymbol{\lambda} \ \forall i \in 1..N \end{aligned} \qquad (17)$$

where $D(\boldsymbol{\lambda})$ is the diagonal matrix formed from the vector $\boldsymbol{\lambda}$, and $e \in \mathbb{R}$ is the vector of ones. The dual function of (17) is:

$$\theta(\boldsymbol{\lambda}) = \begin{cases} -e^T\boldsymbol{\lambda} & \text{if } \overline{\mathbf{W}} + D(\boldsymbol{\lambda}) \succcurlyeq 0, \\ -\infty & \text{otherwise} \end{cases} \qquad (18)$$

Therefore, the dual problem is:

$$\begin{aligned} &\sup -e^T\boldsymbol{\lambda}, \\ &\quad \text{s.t. } \overline{\mathbf{W}} + D(\boldsymbol{\lambda}) \succcurlyeq 0 \end{aligned} \qquad (19)$$

For the CDMA problem, we use $\mathbf{L}$ instead of $\overline{\mathbf{W}}$.

We will now present the results obtained in our numerical experiments. The first part will be the instances of the SDPLIB, and the comparison between what we find and the best known value. The second part will be the CDMA case, where we compare the various bounds obtained by the relaxations.

3. Numerical Experiments

3.1. SDPLIB

The SDPLIB (Borchers 1999b) instances are different from the CDMA ones: the weights on the edges of the graph are positive, and the eigenvalues of $\mathbf{W}$ are negative and positive in similar proportion. The CDMA problems have a particular form since $\mathbf{L}$ only has one negative eigenvalue (the submatrix $\mathbf{Q}$ is positive semidefinite). This allows the use of the spectral decomposition, whereas the SDPLIB cases do not.
We summarized the results obtained in the table 1. Overall, the results obtained are in line with the

Known Best	SDP	Lagrangian	SDPLIB data	Number of variables
629.1648	629.040429103251	634.826655582746	maxG11	800
1567.640	1567.76501421199	1579.21142981607	maxG32	2000
4003.809	4006.45796697314	4006.25552096543	maxG51	1000
9999.210	n/a[4]	n/a[4]	maxG55	5000
15222.27	n/a[4]	n/a[4]	maxG60	7000
226.1574	226.125421615193	226.157351345854	mcp100	100
141.9905	141.972966277643	141.990477046969	mcp124-1	124
269.8802	269.870915966374	269.880170252446	mcp124-2	124
467.7501	467.794674836781	467.750114080544	mcp124-3	124
864.4119	864.270065174176	864.411862950212	mcp124-4	124
317.2643	317.264035444626	317.264340169945	mcp250-1	250
531.9301	531.917696086031	531.930083790969	mcp250-2	250
981.1726	981.166174015170	981.172569846116	mcp250-3	250
1681.960	1682.06418683546	1681.96011037827	mcp250-4	250
598.1485	598.147929328005	598.148517065595	mcp500-1	500
1070.057	1070.05882614284	1070.05676540164	mcp500-2	500
1847.970	1848.08624038665	1847.97001980621	mcp500-3	500
3566.738	3566.81241581776	3566.73804479043	mcp500-4	500

Table 1. Comparison of the results of the SDP and Lagrangian forumlations.

known optima, with the SDP[5] and Lagrangian[6] formulations being close from each other. The MaxGXX class of problems are interesting, however, because on these, the Lagrangian formulation is not performing as well as the SDP relaxation. In the case of MaxG51, the SDP and Lagrangian solutions are close, but are not as good as the known best. The maxG11, mcp100, mcp124-1, mcp124-2, mcp124-4, mcp250-1, 2, 3 and mcp500-1 cases shows that SBmethod can give better results than the known best, and in the cases mcp124-3, mcp250-4, and mcp500-2, 3 and 4, SBmethod is outperformed. In all these cases (except MaxG11), the Lagrangian forumlation returns the same bounds as those known . We think the slight variations of our results using the traditional SDP formulation compared to the known best bounds is caused by the software used in our experiments.

3.2. CDMA simulations

3.2.1 *Lower bounds comparison*

We expected to see the following order of quality for the bounds (starting form the worst): spectral decomposition, linear relaxation, SDP and Lagrangian relaxations. Surprisingly, the order is not fully followed. Despite being a relaxation of the variables over $\mathbb{R}$, the spectral decomposition appears to give better bounds than the linear relaxation. The linear relaxation appeared to be inefficient. When the other relaxations find a bound[7] close from the one returned by a rounding heuristic[8] (our optimum is negative), which is what is expected, the linear relaxation returned a considerably lower (and therefore worse) bound.

Furthermore, when the signal to noise ratio (SNR) grows, the gaps between the spectral decomposition, Lagrangian and SDP relaxations get progressively smaller, while the bound given by the linear relaxation does not really improve; in fact, the progression is erratic[9]. In other terms, the first three become better evaluations (the spectral decomposition benefits the most from this), while the linear relaxation stays inefficient. It should also be noted that the spectral decomposition is by far the most efficient in terms of memory usage (the linear relaxation uses $2K^2$ constraints) and in terms of calculation time.

We see on figures 1, 2 and 3 that when the SNR grows, the gap between the spectral decomposition and the SDP and Lagrangian relaxations gets smaller. On all the SNR spectrum, the gap between SDP and Lagrangian relaxations is small, which confirms what was exposed in (Lemaréchal & Oustry 1999): the two formulations are equivalent. However, we see that the values of the SDP relaxation are not significantly different from the Lagrangian relaxation.

[4]The calculations could not be done for memory limits reasons

[5]SBmethod, using the default settings, and Matlab

[6]CSDP, through Matlab and Yalmip(Löfberg 2004)

[7]All calculations are done in Matlab, with calls to SBmethod for the SDP relaxation, to CSDP through Yalmip for the Lagrangian formulation, and to CPLEX for the linear relaxation

[8]Small local search around the nearest integer rounding of the SDP result, which gives a suboptimal, but acceptable solution

[9]A greater number of simulations may flatten the curve, but the calculation time of the linear relaxation is equally inefficient

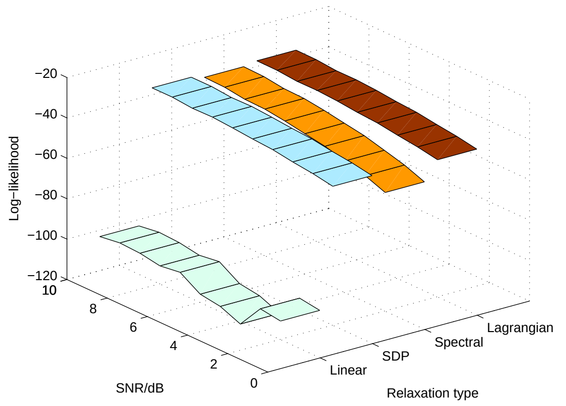

Figure 1. Average Loglikelihood of the various relaxations for $K = 36,\ N = 64$

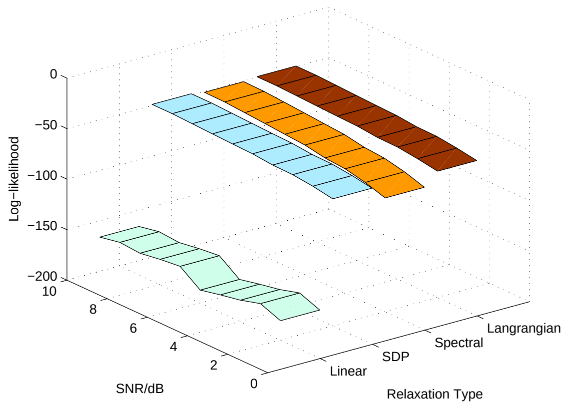

Figure 2. Average Loglikelihood of the various relaxations for $K = 48,\ N = 64$

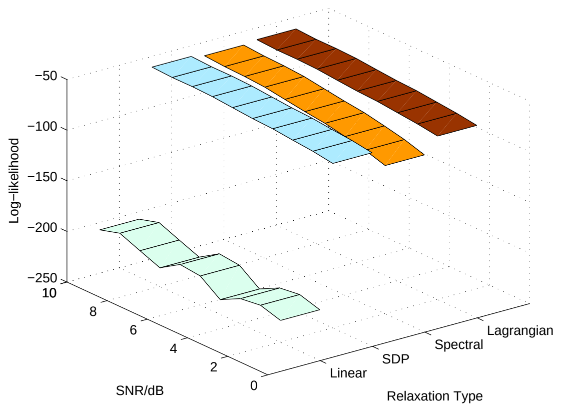

Figure 3. Average Loglikelihood of the various relaxations for $K = 60,\ N = 96$

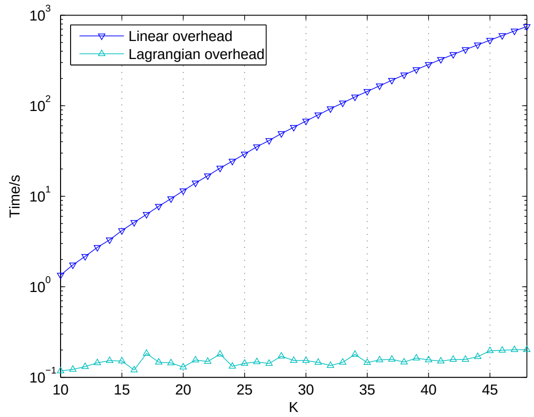

Figure 4. Time (in s) for Yalmip to generate the constraints for the linear and Lagrangian relaxations, with $K \in [10, 48]$

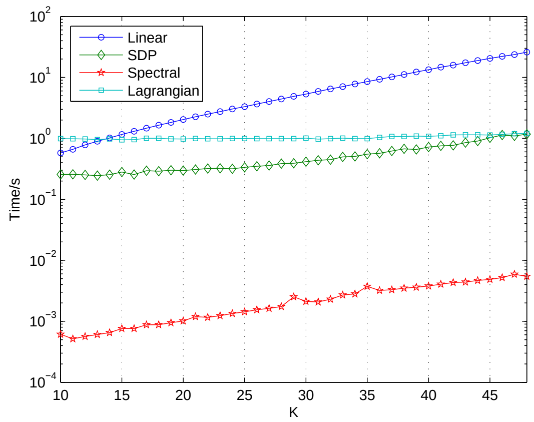

Figure 5. Average time (in s) per simulation, with $K \in [10, 48]$

3.2.2 *Calculation times*

The calculation time for the linear and Lagrangian relaxations is particular, in the sense that there is an initial constraint generation step that can be done once for a given sequence of simulations, which means that the more iterations we do during a simulation, the lower the average time. That constraint generation cost is dependent only on the number of variables (see figure 4). However, it also uses more and more memory[10], up to the point where Matlab is unable to manage the set of constraints. Once these have been generated, for a set of simulations, we can observe the average time taken by each method (see figure 5). We can see that the fastest performance is achieved by the spectral decomposition (in exchange

[10] This is a known issue of Yalmip, which we used at the time for its capacity to interface directly with both CPLEX and CSDP. A much more efficient implementation is possible by directly using cplexint/cplexmex to interface with CPLEX, and the CSDP interface found in the CSDP archive and rewriting the instances in two completely different formats.

for a poorer bound). The linear relaxation is quickly ($K = 14$) the worst in terms of calculation time. The SDP relaxation (with SBmethod) performs well, but the calculation time grows at a faster rate than that of the Lagrangian relaxation. It appears that for large scale problems, where one needs to quickly compute an upper bound, using a Lagrangian formulation might prove more useful, assuming memory efficient tools are used. On the other hand, to obtain a primary solution (to use in an approximation algorithm, for example), then using a more traditional formulation might be more appropriate.
The CDMA problems are a particular class of MaxCut problems, which means that the results obtained do not necessarily hold for other MaxCut problems. Second, in our tests the Lagrangian relaxation is more memory consuming than the SDP method[11]. Last, while the Langrangian method calculates the bound faster, it does not return the vector that leads to that bound, which means we cannot use it as a basis to find a feasible solution.

CONCLUSION

In this paper, we show that with the current available software, calculations made to solve SDP problems (CDMA model or SDPLIB) give very similar results using both the SDP relaxation (using SBmethod) or its Lagrangian form (using CSDP). We also see that some of the results depend on the method (or software): choosing SBmethod rather than CSDP can yield slightly different bounds, and one is not necessarily always better. Indeed, our tests so far showed that in the CDMA case, the average calculation times for Lagrangian form grow slower than for the SDP relaxation. Second, the problems arising from the CDMA model have intersting properties: when the Signal to Noise ratio raises, the gap between the spectral decomposition and the SDP and Lagrangian relaxations becomes smaller, which makes this method a good way to find a lower bound of the solution in a very short time. This bound can then be used as a starting point for other methods.

REFERENCES

Borchers, B. (1999a). CSDP, a C library for semidefinite programming, *Optimization Methods and Software* **11**: 613–623.

Borchers, B. (1999b). SDPLIB 1.2, a library of semidefinite programming test problems, **11**: 683–690.

Boros, E. & Hammer, P. L. (2002). Pseudo-boolean optimization, *Discrete Appl. Math.* **123**(1-3): 155–225.

Goemans, M. X. & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* **42**(6): 1115–1145.

Grötschel, M., Lovász, L. & Schrijver, A. (1984). Corrigendum to our paper "the ellipsoid method and its consequences in combinatorial optimization"., *Combinatorica* **4**(4): 291–295.

Helmberg, C. (2000). SBmethod – a C++ implementation of the spectral bundle method. manual to version 1.1, *ZIB-Report ZR 00-35*, Konrad-Zuse-Zentrum fr Informationstechnik, Berlin.

Håstad, J. (1997). Some optimal inapproximability results, *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, ACM Press, New York, NY, USA, pp. 1–10.

Karp, R. M. (1972). Reducibility among combinatorial problems, *in* R. E. Miller & J. W. Thatcher (eds), *Complexity of Computer Computations*, Plenum Press, pp. 85–103.

Laurent, M. & Poljak, S. (1995). On a positive semidefinite relaxation of the cut polytope, *Linear Algebra and its Applications* **223/224**(1–3): 439–461.

Lemaréchal, C. & Oustry, F. (1999). Semidefinite relaxations and lagrangian duality with application to combinatorial optimization, *Technical Report RR–3710*, INRIA Rhone-Alpes.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB, *Proceedings of the CACSD Conference*, Taipei, Taiwan.

Lovász, L. (1979). On the shannon capacity of a graph, *IEEE Trans. Inform. Theory* **25**(1): 1–7.

Padberg, M. (1989). The boolean quadric polytope: some characteristics, facets and relatives, *Math. Program.* **45**(1): 139–172.

Tan, P. H. & Rasmussen, L. K. (2001). The application of semidefinite programming for detection in CDMA, *IEEE Journal on Selected Areas in Communications* **19**(8): 1442–1449.

Verdú, S. (1986). Minimum probability of error for asynchronous gaussian multiple-access channels., *IEEE Transactions on Information Theory* **32**(1): 85–96.

Verdú, S. (1998). *Multiuser Detection*, Cambridge University Press, New York, NY, USA.

[11]This is most likely due to Yalmip, however.