A Preprocessing Algorithm Applicable to the Multiuser Detection Problem

Daniel Axehill, Fredrik Gunnarsson, Anders Hansson

Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: daniel@isy.liu.se, fred@isy.liu.se, hansson@isy.liu.se

14th December 2005

Report no.: LiTH-ISY-R-2716
Submitted to RadioVetenskap och Kommunikation 2005

Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications.
Abstract
In this paper a preprocessing algorithm for binary quadratic programming problems is presented. For some types of binary quadratic programming problems, the algorithm can compute the optimal value for some or all integer variables without approximations in polynomial time. When the optimal multiuser detection problem is formulated as a maximum likelihood problem, a binary quadratic programming problem has to be solved. Fortunately, the low correlation between different users in the multiuser detection problem enables the use of the preprocessing algorithm. Simulations show that the preprocessing algorithm is able to compute almost all variables in the problem, even though the system is heavily loaded and affected by noise.

Keywords: CDMA, Gold sequences, multiuser detection, polynomial complexity, binary quadratic programming
A PREPROCESSING ALGORITHM APPLICABLE TO THE MULTIUSER DETECTION PROBLEM*

Daniel Axehill, Fredrik Gunnarsson and Anders Hansson

Division of Automatic Control, Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden, {daniel,fred,hansson}@isy.liu.se

ABSTRACT
In this paper a preprocessing algorithm for binary quadratic programming problems is presented. For some types of binary quadratic programming problems, the algorithm can compute the optimal value for some or all integer variables without approximations in polynomial time. When the optimal multiuser detection problem is formulated as a maximum likelihood problem, a binary quadratic programming problem has to be solved. Fortunately, the low correlation between different users in the multiuser detection problem enables the use of the preprocessing algorithm. Simulations show that the preprocessing algorithm is able to compute almost all variables in the problem, even though the system is heavily loaded and affected by noise.

1. INTRODUCTION
Multiuser detection is the process to demodulate multiple users sharing a common multi-access channel. A first approach is to demodulate each user independently and treat the signal from other users as additive Gaussian noise, [14]. An improvement to this strategy is to use the known correlation between users in the demodulation process. Better performance can be achieved if the detector makes the most likely decision, which formally is achieved by solving a so-called Maximum Likelihood (ML) problem. When the optimum multiuser detection problem is cast on the form of an ML problem it requires the solution of a so-called Binary Quadratic Programming (BQP) problem. Unfortunately, these problems are generally known to be \textit{NP}-hard, [7]. If the signature waveform produces a cross-correlation matrix with some special structures, the problem can sometimes turn out to have lower complexity, [15, 12, 13].

Many contributions to the area of multiuser detection have already been published. The objective has been to find an algorithm which solves the multiuser detection problem in reasonable time in order to make a real-time implementation possible. So far, this has been done either by restricting the class of possible cross-correlation matrices or by employing some sub-optimal procedure. In [15], an algorithm with polynomial complexity has been derived for systems with only negative cross-correlations. A similar requirement on the cross-correlation matrix is found in [12], where the multiuser detection problem is solved with a polynomial complexity algorithm if the cross-correlation between the users are non-positive. Another paper also dealing with a special class of cross-correlations is [13], where a polynomial complexity algorithm is derived for the case of identical, or a few different, cross-correlations between the users. Thorough work in the field of approximate algorithms for multiuser detection is found in [14]. Several different algorithms, optimal as well as sub-optimal, are presented and evaluated in [5]. The sub-optimal algorithm local search is evaluated in [6]. Branch and bound methods are investigated in [10]. Another near optimal approach is presented in [8]. Also the well-known Kalman filter has been applied to the problem. This approach is presented in [9].

In this paper, a preprocessing algorithm with polynomial complexity for the BQP problem is derived. A preprocessing algorithm is an algorithm which processes the optimization problem in the step previous to the one when the actual solver is applied. Because the preprocessing algorithm executes in polynomial time and the BQP solver, generally, executes in exponential time, the required CPU time can be reduced if optimal variables can be detected already in the preprocessing step. In [1], the algorithm has previously been successfully applied to Model Predictive Control (MPC) for systems including binary variables. When the algorithm is applied to the multiuser detection problem, it not only works as a preprocessing algorithm, but it also shows some important properties of the solution to the problem.

Most algorithms involved when solving BQP problems either focus on producing approximative solutions or only on handling various special cases of the general problem, [3]. The algorithm presented in this paper belongs to the latter type of algorithms. Some approximative heuristic algorithms can be found in, e.g., [7], [2], [11] and [4].

2. THE BQP PROBLEM
In this paper a BQP problem without any constraints is considered. First, a BQP problem with no linear term is studied

\[
\mathcal{P}_1 : \min \quad x^T H x \\
\text{subject to} \quad x \in \{0, 1\}^{n_b} \tag{1}
\]

where \( H \in \mathbb{R}^{n_b \times n_b} \) is symmetric. Thereafter, the result is extended to a BQP problem of the form

\[
\mathcal{P}_2 : \min \quad \frac{1}{2} x^T H x + f^T x \\
\text{subject to} \quad x \in \{0, 1\}^{n_b} \tag{2}
\]

where a linear term has been incorporated in the objective function.

For what follows it is practical to define

\[
H = H_d + H^+ + H^- \tag{3}
\]

where

\[
H_{d,ij} = \begin{cases} H_{ij}, & i = j \\ 0, & i \neq j \end{cases} \tag{4}
\]

\[
H^+_{ij} = \max(0, H_{ij} - H_{d,ij}) \\
H^-_{ij} = \min(0, H_{ij} - H_{d,ij})
\]

---

*The research has been supported by the Swedish Research Council for Engineering Sciences under contract Nr. 621-2002-3822.
3. SYNCHRONOUS CDMA

In this section, the synchronous CDMA model is presented. It is also shown how the multiuser detection problem can be formulated as a BQP problem.

3.1. System Model

In this paper a CDMA channel for \( K \) simultaneous users is considered. The symbol length is assumed to be \( T \) seconds. Each user is assigned a certain signature sequence, a so-called chip sequence. The chip sequence is a sequence consisting of \( N \) chips, each taking a value from \( \{-1, +1\} \). The constant \( N \) is known as the spreading factor, spreading gain or processing gain, [16].

The notation used in this text has been chosen similar to the one used in [16]. The K-user channel consists of the sum of \( K \) antipodally modulated synchronous signature waveforms embedded in additive white Gaussian noise

\[
y(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t), \quad t \in [0, T]
\]  

(5)

where

- \( s_k(t) \) is the deterministic signature waveform assigned to user \( k \), normalized to have unit energy, i.e.

\[
\|s_k\|^2 = \int_0^T s_k^2(t) dt = 1
\]  

(6)

Because the waveforms are assumed to be zero outside the interval \([0, T]\), there is no inter-symbol interference.

- \( A_k \) is the received amplitude of the signal from user \( k \), and therefore, \( A_k^2 \) is referred to as the energy of user \( k \).

- \( b_k \in \{-1, +1\} \) is the data bit transmitted by user \( k \).

- \( n(t) \) is white Gaussian noise with unit power spectral density.

The similarity of different signature waveforms is expressed in terms of the cross-correlation defined by

\[
\rho_{ij} = \langle s_i, s_j \rangle = \int_0^T s_i(t) s_j(t) dt
\]  

(7)

From (6), (7) and Cauchy’s inequality it follows that

\[
|\rho_{ij}| = |\langle s_i, s_j \rangle| \leq \|s_i\| \|s_j\| = 1
\]  

(8)

The normalized cross-correlation matrix \( R = \{\rho_{ij}\} \) has ones in the diagonal and is symmetric nonnegative definite, [16]. If \( \rho_{ij} = 0 \) whenever \( i \neq j \), the signature sequences are orthogonal. In this paper non-orthogonal sequences have been used and these usually give low cross-correlation for all possible non-zero offsets. Common choices of such sequences are Gold sequences and Kasami sequences.

At the receiver, the signal \( y(t) \) in (5) is received. After the reception, the procedure of despreading begins. Here the low cross-correlation between the different signature sequences is useful. The despreading is performed by \( K \) matched filters, where the output from filter \( k \) can be written as

\[
y_k = \int_0^T y(t) s_k(t) dt
\]  

(9)

Using (5), (6) and (7), (9) can be equivalently expressed as

\[
y_k = A_k b_k + \sum_{j \neq k} A_j b_j \rho_{jk} + n_k
\]  

(10)

where

\[
n_k = \sigma \int_0^T n(t) s_k(t) dt \in \mathcal{N}(0, \sigma^2)
\]  

(11)

Using vector notation, this can be written more compactly as

\[
y = RAb + n
\]  

(12)

where \( R \) is the normalized cross-correlation matrix and

\[
y = [y_1, ..., y_K]^T
\]

\[
b = [b_1, ..., b_K]^T
\]

\[
A = \text{diag}\{A_1, ..., A_K\}
\]

Furthermore, the unnormalized cross-correlation matrix is denoted as

\[
H = ARA
\]  

(14)

Because only the synchronous case is treated in this paper, no inter-symbol interference will occur. Hence, it is only necessary to look at one time instant and therefore no time index on \( y, b \) or \( n \) is necessary.

3.2. Derivation of the BQP Problem

The matched filter output is described by equation (12). According to [16], the bits most likely sent by the users are given by the solution to the optimization problem

\[
\max_b \exp \left( -\frac{1}{2\sigma^2} \int_0^T (y(t) - \sum_{k=1}^{K} b_k A_k s_k(t))^2 dt \right)
\]  

(15)

Alternatively, it is equivalent to maximize

\[
\Omega(b) = 2 \int_0^T \left[ \sum_{k=1}^{K} A_k b_k s_k(t) \right] y(t) dt
\]

\[
- \int_0^T \left[ \sum_{k=1}^{K} A_k b_k s_k(t) \right]^2 dt
\]

\[
= 2b^T Ay - b^T Hb
\]  

(16)

where \( A, H, b \) and \( y \) are defined in (13) and (14). By altering the sign of the objective and dividing it by two, the optimization problem can be rewritten as an equivalent minimization problem

\[
\min_b \frac{1}{2} b^T Hb - y^T A^T b
\]  

(17)

where \( b \in \{-1, +1\}^K \). After a variable substitution, this problem can be identified as a BQP problem on the form \( \mathcal{P}_2 \).
4. PREPROCESSING FOR THE BQP PROBLEM

The BQP problem to be solved is a pure combinatorial problem. The algorithm presented in this section makes it possible to speed up the solution of BQP problems having large diagonal elements compared to the non-diagonal elements. For each binary variable the algorithm delivers one out of three possible results: 1 is the optimal value, 0 is the optimal value or nothing can be said for sure.

The preprocessing algorithm is built upon the following theorem:

**Theorem 1** For a BQP problem of type $\mathcal{P}_1$, the optimal value of one or more components $x_i$ can be found in polynomial time if for some $i \in \{1, \ldots, n_b\}$ any of the following conditions is fulfilled

\[
\begin{align*}
H_{ii} &\geq -2 \sum_{j \neq i} n_j H_{ij} \quad (i) \\
H_{ii} &< -2 \sum_{j \neq i} n_j H_{ij} \quad (ii)
\end{align*}
\]

If any of the conditions (i) or (ii) is fulfilled for a certain value of $i$, the optimal value of $x_i$ is given by

\[
x_i = \begin{cases} 
0, & \text{if (i) holds} \\
1, & \text{if (ii) holds}
\end{cases}
\]

**Proof:** Consider the optimization problem $\mathcal{P}_1$, where $H$ is a symmetric $n_b \times n_b$ matrix. Denote the objective function by $Q(x)$ and rewrite it as follows

\[
Q(x) = x^T H x = \sum_i \sum_j H_{ij} x_i x_j
\]

where $x_i, x_j \in \{0, 1\}, \forall i, j \in \{1, \ldots, n_b\}$. For each $i \in \{1, \ldots, n_b\}$ the objective function $Q(x)$ can be written on the form

\[
Q(x) = H_{ii} x_i x_i + 2 x_i \sum_{j \neq i} H_{ij} x_j + g_i(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_b})
\]

where $g_i$ is a function that is independent of $x_i$ and where the last equality follows from the fact that $x_i^2 = x_i$ when $x_i \in \{0, 1\}$.

Define

\[
h_i(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_b}) = H_{ii} + 2 \sum_{j \neq i} H_{ij} x_j
\]

Note that $h_i$ is independent of $x_i$. With this definition the objective function can be written as

\[
Q(x) = h_i(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_b}) x_i + g_i(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_b})
\]

Denote the optimal value of $x$ with $x^* = [x_1^*, \ldots, x_{n_b}^*]^T$. It is also convenient to make the following definitions

\[
h_i^* = h_i(x_1^*, x_2^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_{n_b}^*)
\]

\[
g_i^* = g_i(x_1^*, x_2^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_{n_b}^*)
\]

It now follows that

\[
\min Q(x) = \min_{x_i} h_i^* x_i + g_i^* = \begin{cases} 
g_i^*, & \text{if } h_i^* \geq 0 \\
h_i^* + g_i^*, & \text{if } h_i^* < 0
\end{cases}
\]

From (23) the conclusion can be drawn

\[
x_i^* = \begin{cases} 
0, & \text{if } h_i^* \geq 0 \\
1, & \text{if } h_i^* < 0
\end{cases}
\]

Unfortunately, $h_i^*$ is usually not known before the optimal solution $(x_1^*, x_2^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_{n_b}^*)$ is known. A solution to this problem is to try to make an estimate of $h_i^*$.

To simplify the notation, define

\[
\begin{align*}
\tilde{h}_i &= \max_h h_i \\
\tilde{h}_i &= \min_x h_i
\end{align*}
\]

It now follows that $h_i \leq h_i^* \leq \tilde{h}_i$. From this observation the following implications can be stated

\[
\begin{align*}
\tilde{h}_i > 0 &\Rightarrow h_i^* < 0 \\
\tilde{h}_i \geq 0 &\Rightarrow h_i^* \geq 0
\end{align*}
\]

By combining (24) and (26), the following conclusion can be drawn

\[
x_i^* = \begin{cases} 
0, & \tilde{h}_i \geq 0 \\
1, & \tilde{h}_i < 0
\end{cases}
\]

From (4) and (20) it follows that

\[
\tilde{h}_i = \max_x h_i = \max_x (H_{ii} + 2 \sum_{j \neq i} H_{ij} x_j)
\]

\[
= H_{ii} + 2 \max_x \sum_{j \neq i} H_{ij} x_j = H_{ii} + 2 \sum_{j} H_{ij}^+
\]

where the last equality follows from the fact that the sign of $H_{ij}$ determines whether the optimal value of $x_j$ is 0 or 1. Analogously it follows that

\[
\tilde{h}_i = \min_x h_i = H_{ii} + 2 \sum_{j} H_{ij}^-
\]

Equation (27) can then finally be written on the desired form

\[
x_i^* = \begin{cases} 
0, & H_{ii} + 2 \sum_{j} H_{ij}^+ \geq 0 \Rightarrow H_{ii} \geq -2 \sum_{j} H_{ij}^- \\
1, & H_{ii} + 2 \sum_{j} H_{ij}^+ < 0 \Rightarrow H_{ii} < -2 \sum_{j} H_{ij}^+
\end{cases}
\]

From (30) it is clear that the computational complexity of the tests (i) and (ii) is polynomial in the number of variables, i.e. in $n_b$.

Now the result is extended to problems of type $\mathcal{P}_2$.

**Corollary 1** For a BQP problem of type $\mathcal{P}_2$ the optimal value of one or more components $x_i$ can be found in polynomial time if for some $i \in \{1, \ldots, n_b\}$ any of the following conditions is fulfilled

\[
\begin{align*}
H_{ii} &\geq -2 f_i - 2 \sum_{j \neq i} H_{ij}^+ \quad (i) \\
H_{ii} &< -2 f_i - 2 \sum_{j \neq i} H_{ij}^+ \quad (ii)
\end{align*}
\]
If any of the conditions (i) or (ii) is fulfilled for a certain value of \( i \), the optimal value of \( x_i \) is given by

\[
x_i = \begin{cases} 
0, & \text{if (i) holds} \\
1, & \text{if (ii) holds}
\end{cases}
\]

**Proof:** The result follows directly from Theorem 1 by recognizing the fact that

\[
Q(x) = \frac{1}{2} x^T H x + f^T x = x^T \left( H + 2 \text{diag}(f) \right) x
\]

for \( x_i \in \{0, 1\} \).

5. APPLICATION OF THE RESULTS TO SYNCHRONOUS CDMA

In this section, it is shown how to apply the preprocessing algorithm derived in Section 4 to the BQP problem (17) and how the result can be interpreted.

5.1. Using Preprocessing

In order to be able to apply the preprocessing algorithm, the optimization problem (17) has to be rewritten on the BQP form \( P_2 \). Note especially the domain of the optimization variable \( x \). In order to get an optimization problem with binary variables the following variable substitution is performed

\[
b = 2\bar{b} - 1
\]

where \( \bar{b} \in \{0, 1\}^K, b \in \{-1, +1\}^K \) and \( 1 \) denotes a column vector with all elements equal to one. Using (32), neglecting constant terms and dividing by 4, the objective function in (17) can be rewritten as

\[
\frac{1}{2} \bar{b}^T H \bar{b} + \bar{f}^T \bar{b}
\]

where

\[
\bar{f} = -\frac{1}{2} H 1 - \frac{1}{2} Ay
\]

The problem is now on the form \( P_2 \), on which preprocessing can be performed.

5.2. From Theorem to Algorithm

When implementing the algorithm, the conditions in Corollary 1 are used. For each element in the vector \( x \), the inequalities (i) and (ii) are tested. If any of the inequalities are fulfilled for a certain element, the optimal value of that element has been found. The optimal value of elements already calculated can be used in the following tests for other elements in order to tighten the upper and the lower bounds. If elements remain to be computed after all elements have been tested once, as long as at least one element was computed in the previous run it is possible to start over again and try to compute the remaining ones. This procedure is in this text referred to as the iterated implementation.

When the preprocessing algorithm has terminated, another optimization method has to be applied to compute any remaining variables. Depending on the time available, the other method may be chosen to produce optimal solutions or sub-optimal solutions. References to such algorithms can be found in Section 1.

6. SIMULATIONS

In this section, the preprocessing algorithm is applied to the multiuser detection problem and tested in Monte Carlo simulations. In the first simulations, the joint Bit Error Rate (BER) for the optimal detector implemented by using the preprocessing algorithm is compared to the joint BER of the conventional detector, [5],

\[
\hat{b}_i = \text{sign}(y)
\]

and to the joint BER of the decorrelating detector, [5],

\[
\hat{b}_i = \text{sign}(H^{-1} y)
\]

To be able to make a fair comparison only the variables computed by the preprocessing algorithm were used in the BER calculations for all methods in the comparisons. The tests were performed with Gold Sequences of length 128. The algorithms were compared for the loads 1 to 127.
users. Each load was tested 10000 times. In each test, a new noise realization and a new random bit was assigned to each user. In Figure 2 it can be noticed that the preprocessing algorithm computes nearly all variables in average. Even in the worst case, 75 % of the variables were computed.

The next issue to verify is that the problem instances tested were not "trivial" in the sense that the existing lowest complexity algorithms also could compute them optimally. This verification is performed by calculating the average BER for the different methods during the Monte Carlo simulations. The result from this simulation is shown in Figure 3. In the test, the SNR for user 1 varied from 7 dB to 6.7 dB. The conclusion drawn from the simulation is that the optimal multiuser detector implemented by the preprocessing algorithm gives lower BER than the two other algorithms. Note that the plots for all three methods only includes bits possible to calculate with the preprocessing algorithm. The same plot also verifies that the iterated implementation of the algorithm, enables more "wise" decisions than the one only using a single iteration. In the non-iterated case, it follows from (35) that the optimal decision coincides with the decision taken by (37), in the region where variables can be computed by the preprocessing algorithm. The region where the solution from the algorithms coincide is the region outside the grey region in Figure 1. When the iterated implementation of the preprocessing algorithm is used it can sometimes be possible to decrease the size of the grey area for some variables in the problem. This means that in those cases more variables are possible to compute by preprocessing.

The computational time is illustrated in Figure 4. The tests of the computational times were performed on a Sun UltraSPARC-IIf 500 MHz with 640 Mb RAM running SunOS 5.9 and Matlab 7.0.1.

Gold sequences have very low cross-correlation. To test the algorithm as the cross-correlation increases, the cross-correlation matrix was manually modified. This was done by adding 0.01 − 0.6 in 60 equidistant steps to 16 symmetric off-diagonal elements (8 elements on each side of the diagonal). For each of the 60 steps, 10000 Monte Carlo simulations were performed. The test was performed for 100 users and the result from this simulation can be found in Figure 5. The number of variables possible to compute by preprocessing when the correlation is increased is illustrated by Figure 6.

7. CONCLUSIONS

In this paper a preprocessing algorithm for BQP problems has been derived. These problems are generally known to have exponential complexity. Due to the low cross-correlation between the signature sequences, the BQP problem gets a special structure with small off-diagonal elements. This structure is exploited by the preprocessing algorithm and makes it possible to compute the optimal values of some, or all, of the optimization variables.
in polynomial time. In the paper, the preprocessing algorithm has been successfully applied to the multiuser detection problem when signature sequences of Gold type have been used. The algorithm can be combined both with optimal and suboptimal optimization algorithms. Compared to previously presented low complexity optimal algorithms, the proposed algorithm does not demand any special care in the design of the signature sequences, except from the already existing desire that the cross-correlation should be chosen as low as possible.

REFERENCES


