

# AN EFFICIENT MAXIMIZATION ALGORITHM WITH IMPLICATIONS IN MIN-MAX PREDICTIVE CONTROL

T. Alamo<sup>\*</sup>, D. Muñoz de la Peña<sup>\*</sup> and E.F. Camacho<sup>\*</sup>

<sup>\*</sup> Dpto. de Ingeniería de Sistemas y Automática. Universidad de Sevilla  
Camino de los Descubrimientos s/n, 41092 Sevilla, Spain  
e-mail: {alamo, davidmps, eduardo}@cartuja.us.es

**Keywords:** Binary quadratic programming, band matrices, combinatorial optimization, model predictive control, min-max

## Abstract

Binary Quadratic Programming (BQP) problems arise frequently in robust MPC when min-max techniques are used. In this paper, an efficient algorithm that solves the problem for L-Band matrix structures is presented. The L-Band matrix algorithm has a direct application to min-max MPC. The computational burden of the L-Band max algorithm is polynomial with the dimension of the optimization variable and exponential with  $L$ , the band size. The proposed algorithm makes the implementation in real time of min-max predictive controllers possible.

## 1 Introduction

The objective of a binary quadratic programming (BQP) problem is to find a binary vector  $z \in \{-1, 1\}^N$  that maximizes:

$$F(z) = \sum_{i=1}^N \sum_{j=1}^N h_{ij} z_i z_j + \sum_{i=1}^N q_i z_i \quad (1)$$

where  $H = \{h_{ij}\}$  is a symmetric  $N \times N$  matrix and  $q = \{q_i\}$  a real vector.

BQP problems arise in a variety of applications, e.g. capital budgeting and financial analysis [9], traffic message problems [4], machine scheduling [2], molecular formation [1], etc. Furthermore, it is well known that the BQP is equivalent to many classical combinatorial problems such as maximum cut, maximum clique, maximum vertex packing, minimum vertex cover, maximum independent set and maximum weight independent set problems [8]. In Model Predictive Control (MPC), BQP problems arise when min-max techniques are used with

bounded additive uncertainties models [7].

The BQP belongs to the class of NP hard problems. At the present time, the computation of the exact solution is only possible for small size instances. For this reason, min-max predictive controllers are hardly implementable in real plants.

In this paper, an efficient algorithm for solving BQP L-Band problems is presented. A BQP L-Band problem arises when matrix  $H$  has a band structure, that is,  $h_{ij} = 0$  if  $|i - j| \geq L$ . The computational burden of the proposed L-Band maximization algorithm is polynomial with the dimension of the optimization variable and exponential with  $L$ , the band size. To the knowledge of the authors, there is no equivalent result in the literature.

When the control action of a min-max MPC is obtained computing a sequence of correction control actions to a given stabilizing control law [3, 5, 6], the contribution of a perturbation in the prediction vanishes in an exponential way. This implies that  $|h_{ij}|$  decreases in an exponential way with  $|i - j|$ . In this way, the maximization problem arising from a min-max MPC can be approximated by an L-Band maximization problem. The error in the approximation can be made arbitrarily small choosing conveniently the band width  $L$ .

This paper is organized as follows. Section 2 presents the BQP L-Band optimization problem. Section 3 presents the preliminary notation. In section 4, the L-Band algorithm is presented. In section 5, an example is provided. Section 6 is devoted to some preliminary results needed to prove the main contribution of the paper. In section 7 the main theorem is proved. The computational efficiency of the algorithm is addressed in section 8. The paper ends with a section of conclusions.

## 2 L-Band maximization problem

The L-Band maximization problem is:

$$\max_{\|z\|_\infty \leq 1} F(z) = z^T H z + q^T z \quad (2)$$

with  $z \in \mathbb{R}^N$  and  $H$  a semidefinite positive L-Band matrix:

$$|i - j| \leq L \Rightarrow h_{i,j} = 0$$

We propose an algorithm that builds a set of  $2^L$  hypothesis that contains the optimum solution. To build this set we need to make  $(N - L - 1)2^L$  comparisons. So, the algorithm requires a number of evaluations of  $F(z)$  proportional to  $(N - L)2^L$ .

Note that for  $L = 1$  the optimization problem is trivial and the maximum value of the quadratic function is  $\text{tr}(H) + \|q\|_1$  and it is attained at  $x = \text{sign}(q)$ .

## 3 Preliminary notation

**Notation 1** (Hypercube in  $\mathbb{R}^n$ )

$B_r^n$  denotes the following hypercube in  $\mathbb{R}^n$ :

$$B_r^n = \{ z \in \mathbb{R}^n : \|z\|_\infty \leq r \}$$

**Notation 2** (Vertices of a Hypercube)

The vertices of the unit hypercube  $B_1^n$  in  $\mathbb{R}^n$  will be enumerated as follows:

$$\vartheta_i^n = \begin{bmatrix} 1 - 2b_0(i-1) \\ 1 - 2b_1(i-1) \\ \vdots \\ 1 - 2b_{n-1}(i-1) \end{bmatrix}, i = 1, \dots, 2^n$$

where  $b_j(i), j = 0, \dots, n-1$  are the binary digits of  $i$ , that is,  $i = \sum_{j=0}^{n-1} b_j(i)2^j$ .

**Notation 3** Suffix and Prefix

1. Denote the length of vector  $x$  as

$$l(x) = \begin{cases} 0 & \text{if } x = 0 \\ \max \{ i : x(i) \neq 0 \} & \text{otherwise} \end{cases}$$

that is,  $l(x)$  is zero if  $x = 0$ . If  $x \neq 0$  then  $l(x)$  is defined as the greatest index  $i$  that satisfies  $x(i) \neq 0$ .

2. If  $n \leq l(x)$  then  $s_n(x)$  denotes the suffix of length  $n$  of  $x$ , that is,

$$s_n(x) = \begin{bmatrix} x(l(x) + 1 - n) \\ x(l(x) + 2 - n) \\ \vdots \\ x(l(x)) \end{bmatrix}$$

3. Given vector  $x \in \mathbb{R}^m$  and  $n \leq m$ ,  $p_n(x)$  denotes the prefix of length  $n$  of  $x$ , that is,

$$p_n(x) = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(n) \end{bmatrix}$$

The following property can be easily obtained from the properties of  $b_j(\cdot)$ .

**Property 1** .

1. The vertices  $\vartheta_i^L, i = 1, \dots, 2^L$  can be obtained from the vertices  $\vartheta_j^{L-1}, j = 1, \dots, 2^{L-1}$ :

$$\vartheta_i^L = \begin{cases} \begin{bmatrix} \vartheta_i^{L-1} \\ 1 \end{bmatrix} & \text{if } i \leq 2^{L-1} \\ \begin{bmatrix} \vartheta_{i-2^{L-1}}^{L-1} \\ -1 \end{bmatrix} & \text{otherwise} \end{cases}$$

2.  $s_{L-1}(\vartheta_{2i}^L) = s_{L-1}(\vartheta_{2i-1}^L) = \vartheta_i^{L-1}$

## 4 L-Band Max Algorithm

The following algorithm solves the L-Band max problem making a number of operations that, for a fixed width band  $L$ , grows in a polynomial way with the number of optimization variables.

**Theorem 1** Let us consider the quadratic function  $F(z) = z^T H z + q^T z$  where  $H \in \mathbb{R}^{N \times N}$  is a semidefinite positive matrix with band structure, that is,  $H_{ij} = 0$  for all  $|i - j| \geq L > 1$ . Then

$$\max_{\|z\|_\infty \leq 1} F(z) = \max_{i=1 \dots 2^{L-1}} F(z_i^N)$$

where  $z_i^N, i = 1, \dots, 2^{L-1}$  are obtained from the following recursion:

1. Define  $z_i^{L-1} \in \mathbb{R}^N, i = 1, \dots, 2^{L-1}$  as

$$z_i^{L-1} = \sum_{j=0}^{L-2} (1 - 2b_j(i-1))e_{j+1}$$

where  $e_j$  is the  $j$ -esime unitary vector in  $\mathbb{R}^N$  and  $b_j(n), j = 0, \dots, L-2$  are the binary digits of  $n \in [0, 2^{L-1} - 1]$ , that is,  $n = \sum_{j=0}^{L-2} b_j(n)2^j$ .

- Given an integer  $k \in [L, N]$ , and vectors  $z_i^{k-1}, i = 1, \dots, 2^{L-1}$ , vectors  $m_i^k, i = 1, \dots, 2^L$  are obtained from:

$$m_i^k = \begin{cases} z_i^{k-1} + e_k & \text{if } i \leq 2^{L-1} \\ z_{i-2^{L-1}}^{k-1} - e_k & \text{otherwise} \end{cases}$$

- Given an integer  $k \in [L, N]$ , and vectors  $m_i^k, i = 1, \dots, 2^L$ , vectors  $z_i^k, i = 1, \dots, 2^{L-1}$  are obtained from:

$$z_i^k = \begin{cases} m_{2i-1}^k & \text{if } F(m_{2i-1}^k) \geq F(m_{2i}^k) \\ m_{2i}^k & \text{otherwise} \end{cases}$$

## 5 Example

The algorithm proposed in theorem 1 constructs in a recursive way a set of candidates for the optimum solution to problem (2). This algorithm is shown in the following example.

The algorithm starts with a set of  $2^{L-1}$  different vectors of  $L-1$  non zero components. These are all the possible endings of a vector. Therefore, it is guaranteed that the optimum will have one of these endings. This is the set of candidates of length  $L-1$ .

In each iteration, the algorithm builds a candidate set of  $2^{L-1}$  vectors but with one more non zero component than the previous iteration. When the vectors have  $N$  nonzero components the algorithm finishes and the optimum can be evaluated.

When a new component is added, the number of vectors is doubled, one for each of the possible values of the new component. This is the hypothesis set and it is made of  $2^L$  vectors. To eliminate half of this vectors we use lemma 1. This property allows us to compare vectors with the same suffix of length  $L-1$ , eliminating the one with the lower functional. We obtain the  $2^{L-1}$  new set of candidates vectors.

$$H = \begin{bmatrix} 6 & 1 & -2 & 0 \\ 1 & 6 & 1 & -2 \\ -2 & 1 & 5 & 2 \\ 0 & -2 & 2 & 4 \end{bmatrix} \quad N = 4$$

$$L = 3$$

$$q = \begin{bmatrix} 2 \\ -7 \\ 8 \\ -1 \end{bmatrix}$$

$z_1^2$	$z_2^2$	$z_3^2$	$z_4^2$
1	-1	1	-1
1	1	-1	-1
0	0	0	0
0	0	0	0

This is the initial candidates set. All possible endings are present.

$m_1^3$	$m_2^3$	$m_3^3$	$m_4^3$	$m_5^3$	$m_6^3$	$m_7^3$	$m_8^3$
1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1
1	1	1	1	-1	-1	-1	-1
0	0	0	0	0	0	0	0
$F_1^3$	$F_2^3$	$F_3^3$	$F_4^3$	$F_5^3$	$F_6^3$	$F_7^3$	$F_8^3$
<b>20</b>	20	26	<b>34</b>	8	-8	<b>22</b>	14

This is the hypothesis set. It has the double of vectors because another component has been added with the two possible values. The functional is evaluated for each vector, and then, the next candidates set is made of the ones with the higher functional between those that have the same suffix of length  $L-1$ .

$z_1^3$	$z_2^3$	$z_3^3$	$z_4^3$
1	-1	1	1
1	-1	1	-1
1	1	-1	-1
0	0	0	0

The new candidates set has again  $2^{L-1}$  vectors, but now with one more non zero component.

$m_1^4$	$m_2^4$	$m_3^4$	$m_4^4$	$m_5^4$	$m_6^4$	$m_7^4$	$m_8^4$
1	-1	1	1	1	-1	1	1
1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1
1	1	1	1	-1	-1	-1	-1
$F_1^4$	$F_2^4$	$F_3^4$	$F_4^4$	$F_5^4$	$F_6^4$	$F_7^4$	$F_8^4$
23	<b>45</b>	3	<b>25</b>	25	<b>31</b>	21	<b>27</b>

Again the hypothesis set is constructed and the functional evaluated.

$z_1^4$	$z_2^4$	$z_3^4$	$z_4^4$
-1	1	-1	1
-1	-1	-1	-1
1	-1	1	-1
1	1	-1	-1
$F_1^4$	$F_2^4$	$F_3^4$	$F_4^4$
<b>45</b>	25	31	27

This is the last candidates set, where the optimum is guaranteed to be in. After evaluating each vector's functional, the optimum is  $z_1^4$  and the value of the functional is 45. This result can be proved by evaluating the max with a full matrix algorithm.

## 6 Preliminary results

In this section some preliminary results needed for the proof of the main contribution of the paper are presented.

**Lemma 1** *Let us consider the quadratic function  $F(z) = z^T H z + q^T z$  where  $H \in \mathbb{R}^{N \times N}$  is a matrix with band structure, that is,  $H_{ij} = 0$  for all  $|i - j| \geq L > 1$ . Suppose also that  $s \in \mathbb{R}^{L-1}$  then*

$$F\left(\begin{bmatrix} p_a \\ s \\ y \end{bmatrix}\right) - F\left(\begin{bmatrix} p_b \\ s \\ y \end{bmatrix}\right) =$$

$$F\left(\begin{bmatrix} p_a \\ s \\ 0 \end{bmatrix}\right) - F\left(\begin{bmatrix} p_b \\ s \\ 0 \end{bmatrix}\right), \forall y, \forall p_a, p_b$$

PROOF :

Taking into account that  $H$  is L-Band and given  $s \in \mathbb{R}^{L-1}$ ,  $F(\cdot)$  can be rewritten as

$$F\left(\begin{bmatrix} p \\ s \\ y \end{bmatrix}\right) = q^T \begin{bmatrix} p \\ s \\ y \end{bmatrix} +$$

$$\begin{bmatrix} p \\ s \\ y \end{bmatrix}^T \begin{bmatrix} H_{pp} & H_{ps} & 0 \\ H_{ps}^T & H_{ss} & H_{sy} \\ 0 & H_{sy}^T & H_{yy} \end{bmatrix} \begin{bmatrix} p \\ s \\ y \end{bmatrix}$$

Due to this structure, it is easy to see that

$$F\left(\begin{bmatrix} p_a \\ s \\ y \end{bmatrix}\right) - F\left(\begin{bmatrix} p_b \\ s \\ y \end{bmatrix}\right) =$$

$$F\left(\begin{bmatrix} p_a \\ s \\ 0 \end{bmatrix}\right) - F\left(\begin{bmatrix} p_b \\ s \\ 0 \end{bmatrix}\right), \forall y, \forall p_a, p_b$$

**Lemma 2** *Let us suppose that the vectors  $m_i^k, i = 1, \dots, 2^L, k = L, \dots, N$  are computed according to Theorem 1. Then*

$$s_{L-1}(m_{2i-1}^k) = s_{L-1}(m_{2i}^k), \quad i = 1, \dots, 2^{L-1}$$

PROOF :

Note that  $z_i^{L-1}, i = 1, \dots, 2^{L-1}$  are defined as

$$z_i^{L-1} = \sum_{j=0}^{L-2} (1 - 2b_j(i-1))e_{j+1} = \begin{bmatrix} \vartheta_i^{L-1} \\ 0_{N-L+1} \end{bmatrix}$$

where  $i = 1, \dots, 2^{L-1}$  therefore  $s_{L-1}(z_i^{L-1}) = \vartheta_i^{L-1}$

The lemma will be proved in a recursive way. Suppose that  $s_{L-1}(z_i^{k-1}) = \vartheta_i^{L-1}$ . Then it is easy to see that

$$s_L(m_i^k) = \begin{cases} \begin{bmatrix} s_{L-1}(z_i^{k-1}) \\ 1 \end{bmatrix} & \text{if } i \leq 2^{L-1} \\ \begin{bmatrix} s_{L-1}(z_{i-2^{L-1}}^{k-1}) \\ -1 \end{bmatrix} & \text{otherwise} \end{cases} =$$

$$= \begin{cases} \begin{bmatrix} \vartheta_i^{L-1} \\ 1 \end{bmatrix} & \text{if } i \leq 2^{L-1} \\ \begin{bmatrix} \vartheta_{i-2^{L-1}}^{L-1} \\ -1 \end{bmatrix} & \text{otherwise} \end{cases} = \vartheta_i^L$$

Note that the last equality is due to the properties of the proposed enumeration of the vertices of a hyper-cube (see property 1). Thus, it is inferred that  $s_L(m_i^k) = \vartheta_i^L$ . Recall (see property 1) that  $s_{L-1}(\vartheta_{2i}^L) = s_{L-1}(\vartheta_{2i-1}^L) = \vartheta_i^{L-1}$ , this equality yields

$$s_{L-1}(m_{2i-1}^k) = s_{L-1}(m_{2i}^k) = \vartheta_i^{L-1} \quad (3)$$

Now from the definition of  $z_i^k$  it is easy to see that  $s_{L-1}(z_i^k)$  is equal to

$$\begin{cases} s_{L-1}(m_{2i-1}^k) & \text{if } F(m_{2i-1}^k) \geq F(m_{2i}^k) \\ s_{L-1}(m_{2i}^k) & \text{otherwise} \end{cases}$$

■

Taking into account equation (3) it results that  $s_{L-1}(z_i^k) = \vartheta_i^{L-1}$

## 7 Proof of the algorithm

Let us suppose that  $y \in B_1^N$  and that

$$F(y) = \max_{\|z\| \leq 1} F(z)$$

Suppose also that there is  $k \in [2, N-1]$  and  $i \in [1, 2^{L-1}]$  such that  $p_{k-1}(z_i^{k-1}) = p_{k-1}(y)$ . Then there is  $j \in [1, 2^{L-1}]$  and  $\hat{y} \in B_1^N$  such that

$$\begin{cases} F(y) &= F(\hat{y}) \\ p_k(z_j^k) &= p_k(\hat{y}) \end{cases} \quad \blacksquare$$

In effect, denote

$$r = \begin{cases} i & \text{if } y(k) = 1 \\ i + 2^{L-1} & \text{if } y(k) = -1 \end{cases}$$

From the definition of  $m_i^k$ , and from the equality  $p_{k-1}(z_i^{k-1}) = p_{k-1}(y)$ , it is easily inferred that

$$p_k(y) = \begin{bmatrix} p_{k-1}(y) \\ y(k) \end{bmatrix} = \begin{bmatrix} p_{k-1}(z_i^{k-1}) \\ y(k) \end{bmatrix} = p_k(m_r^k)$$

Note that in order to compute the set of vectors  $\{z_i^k\}$ ,  $i = 1, \dots, 2^{L-1}$ ,  $m_r^k$  will be compared with  $m_{r+a}^k$ , where  $a = 0$  if  $r$  is odd and  $a = 1$  if  $r$  is even. In what follows, it will be shown that

$$F(m_r^k) \geq F(m_{r+a}^k)$$

Note that from lemma 2,  $s_{L-1}(m_r^k) = s_{L-1}(m_{r+a}^k)$ . Thus, applying lemma 1 and the fact that  $y$  is a maximizer of  $F(\cdot)$  in the unit-hypercube:

$$\begin{aligned} F(m_r^k) - F(m_{r+a}^k) &= \\ F\left(\begin{bmatrix} p_k(m_r^k) \\ s_{N-k}(y) \end{bmatrix}\right) - F\left(\begin{bmatrix} p_k(m_{r+a}^k) \\ s_{N-k}(y) \end{bmatrix}\right) &= \\ F(y) - F\left(\begin{bmatrix} p_k(m_r^k) \\ s_{N-k}(y) \end{bmatrix}\right) &\geq 0 \end{aligned}$$

From the definition of  $z_i^k$  it results that choosing  $j = (r-a)/2$  and  $\hat{y} = \begin{bmatrix} p_k(z_j^k) \\ s_{N-k}(y) \end{bmatrix}$ :

$$\begin{cases} F(y) &= F(\hat{y}) = \max_{\|z\| \leq 1} F(z) \\ p_k(z_j^k) &= p_k(\hat{y}) \end{cases}$$

Following this reasoning in a recursive way it is obtained that there is  $j \in [1, 2^{L-1}]$  and  $\hat{y} \in B_1^N$  such that

$$F(\hat{y}) = \max_{\|z\| \leq 1} F(z)$$

$$z_j^N = p_N(z_j^N) = p_N(\hat{y}) = \hat{y}$$

## 8 Computational efficiency

A BQP problem is a NP optimization problem. The complexity of the computational effort is exponential with  $N$ , the dimension of the optimization variable. The algorithm presented in this paper for L-band matrices has a computational effort exponential with  $L$ , the band size, but polynomial with  $N$ .

Figure (1) shows the polynomial nature of the L-Band algorithm and the exponential nature of a full matrix problem. The simulations have been made in Matlab 5.3 with the same band optimization problem. Notice that the axis scale is different. The full matrix algorithm explores all possible vertexes and the computation time is much higher.

In figure (2) the dependence of the computational burden of the L-Band algorithm is shown.

## 9 Conclusions

In this paper an efficient algorithm for solving a BQP L-Band problem is presented. The algorithm requires only  $(N-L)2^L$  evaluations of the quadratic functional.

As the simulations results show, the computational burden is much lower than the corresponding to a full matrix BQP algorithm.

The application of the L-Band algorithm makes it possible the implementation of min-max predictive controller in real processes.

## 10 Acknowledgements

The authors are indebted with Daniel Limon, for his helpful comments and remarks. The authors acknowledge MCYT-Spain for funding this work (contract DPI2002-04375-c03-01).

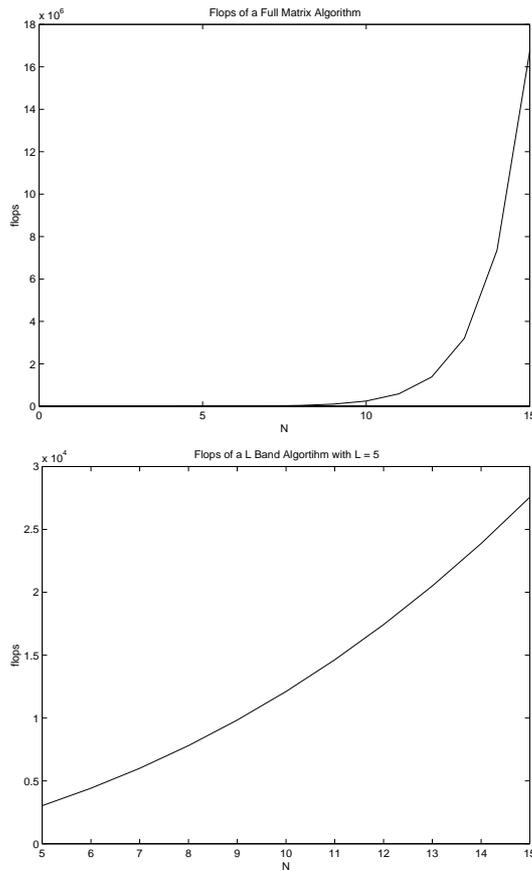


Figure 1: Comparative of the computational effort between the Full Matrix algorithm and the L-Band algorithm

## References

- [1] A. Phillips and J. Rosen. A quadratic assignment formulation for the molecular conformation. *Jnl. of Global Optimization*, 4:229–241, 1994.
- [2] B. Alidaee, B. Kochenberger, and A. Ahmadiam. 0-1-quadratic programming approach for the optimal solution of two scheduling problems. *Int. Jnl. of Systems Science*, 25:401–408, 1994.
- [3] A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 1384–1389, Tampa, Florida USA, December 1998.
- [4] G. Gallo, P. Hammer, and B. Simeone. Quadratic knapsack problems. *Mathematical Programming*, 12:132–149, 1980.

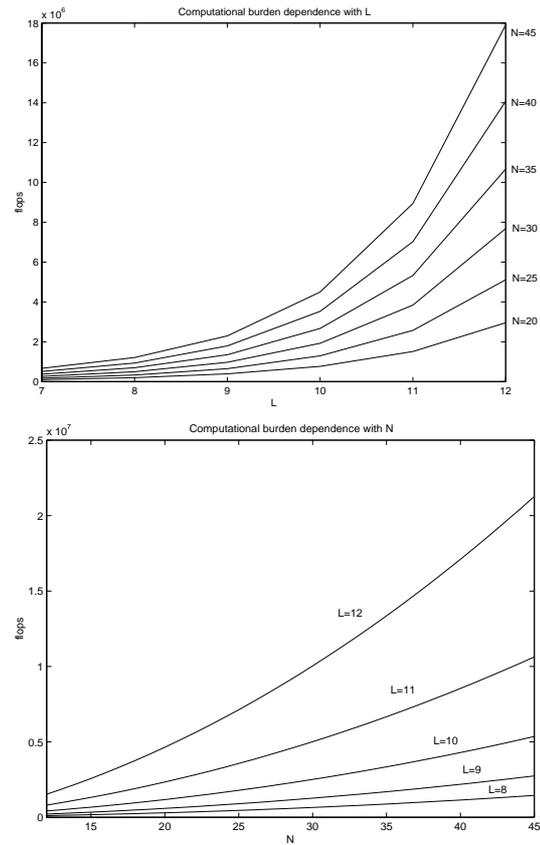


Figure 2: Computational burden of the L-Band algorithm

- [5] B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.
- [6] Y.I. Lee and B. Kouvaritakis. A linear programming approach to constrained robust predictive control. *IEEE Transactions on Automatic Control*, 45(9):1765–1770, 2000.
- [7] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- [8] P.M. Pardalos and J. Xue. The maximum clique problem. *Jnl. of Global Optimization*, 4:301–328, 1994.
- [9] R. Bridge and J. Yorlmark. An evolutionary heuristic for quadratic 0-1 programming. *Management Science*, 26(3):282–296, 1980.