Complex Chebyshev Optimization Using Conventional Linear Programming
- A versatile and comprehensive solution

by

Mattias Dahl, Sven Nordebo
and Ingvar Claesson
Complex Chebyshev Optimization Using Conventional Linear Programming
- A versatile and comprehensive solution
by Mattias Dahl, Sven Nordebo, Ingvar Claesson

ISSN 1103-1581
ISRN HKR-RES—00/6—SE

Copyright © 2000 by Mattias Dahl
All rights reserved
Printed by Psilander Grafiska, Karlskrona 2000
Complex Chebyshev Optimization Using Conventional Linear Programming

A versatile and comprehensive solution

Research report April 2000
Complex Chebyshev Optimization Using Conventional Linear Programming

A versatile and comprehensive solution

Mattias Dahl  Sven Nordebo  Ingvar Claesson

April 2000
Abstract

This paper presents a new practical approach to semi-infinite complex Chebyshev approximation. By using a new technique, the general complex Chebyshev approximation problem can be solved with arbitrary base functions taking advantage of the numerical stability and efficiency of conventional linear programming software packages. Furthermore, the optimization procedure is simple to describe theoretically and straightforward to implement in computer coding. The new design technique is therefore highly accessible. The complex approximation algorithm is general and can be applied to a variety of applications such as conventional FIR filters, narrow-band as well as broad-band beamformers with any geometry, the digital Laguerre networks, and digital FIR equalizers. The new algorithm is formally introduced as the Dual Nested Complex Approximation (DNCA) linear programming algorithm.

The design example in limelight is array pattern synthesis of a mobile base-station antenna array. The corresponding design formulation is general and facilitates treatment of the solution of problems with arbitrary array geometry and side-lobe weighting.

The complex approximation problem is formulated as a semi-infinite linear program and solved by using a front-end applied on top of a software package for conventional finite-dimensional linear programming.

The essence of the new technique, justified by the Caratheodory dimensionality theorem, is to exploit the finiteness of the related Lagrange multipliers by adapting conventional finite-dimensional linear programming to the semi-infinite linear programming problem.

The proposed optimization technique is applied to several numerical examples dealing with the design of a narrow-band base-station antenna array for mobile communication. The flexibility and numerical efficiency of the proposed design technique are illustrated with these examples where even hundreds of antenna elements are optimized without numerical difficulties.
Chapter 1

Introduction

The array pattern synthesis of a non-uniformly spaced sensor array or beamformer \([1, 2]\) is closely related to the design of an FIR filter with arbitrary phase response. The essential similarity is the finite-dimensional nature of the complex approximating functions. The design and application of FIR filters with non-linear phase have previously been extensively studied, see for example \([3]-[6]\).

For beamformers as for FIR filters the design problem is often cast as a finite-dimensional complex approximation problem \([2, 3]\). Classical least squares approximation methods can in many cases be used to obtain a desired solution \([1]\). However, when the design specification is given as a bound on the complex design error, the problem is naturally converted to a complex Chebyshev approximation problem.

It was shown that the (non-linear) complex Chebyshev approximation problem can be reformulated as an equivalent real semi-infinite linear program \([2, 7]\). However, the complex error was approximated in \([2]\) by a finitization in order to solve the problem numerically. Other possible methods to find the Chebyshev solution include quadratic programming \([8]\) (true complex error and approximation by a finite response domain) and a functional inequality approach \([9]\) (true complex error, infinite response domain and approximation by numerical evaluation of integrals).

The complex Chebyshev approximation problem for the design of FIR filters has been intensively studied over the last few years, see for example \([3]-[6]\). Earlier approaches approximate the optimal solution by finitization and employs conventional finite-dimensional linear programming \([3]\). However, it was shown that the semi-infinite linear program corresponding to the (real) Chebyshev approximation problem can be solved by using numerically efficient simplex extension algorithms \([10]\). These results were later exploited for the design of digital FIR filters and digital Laguerre networks with complex Chebyshev error criteria, see for example \([6]\) and \([11]\).

Finitization, see for example \([3]\), can in principle give an arbitrarily accurate approximation of the complex Chebyshev solution but becomes exceedingly
memory intensive as the grid spacing decreases. The semi-infinite simplex extension is much more computationally efficient since the constraint set can be represented in functional form rather than stored in memory as numerical values. Furthermore, the semi-infinite formulation deals directly with the true complex error and not an approximation thereof.

The quadratic programming [8], and functional inequality approaches [9] are also in general more computationally intensive than the semi-infinite linear programming approach.

It was shown that the celebrated Remez exchange algorithm employed for the design of linear phase FIR filters can be put into the framework of semi-infinite linear programming, see [10]. In this context, it is also noted that some other recent approaches to complex FIR filter design such as a generalized Remez algorithm [12], and Tang’s algorithm [13, 5] in fact also employ simplex extension algorithms.

An obstacle with the semi-infinite simplex extension algorithm as described in [6, 10, 11] is the lack of commercially available software for efficient and reliable numerical solution of general complex approximation problems. Furthermore, the recent semi-infinite linear programming theory [10] is rather involved and not very easily accessed by non-specialists. It is therefore an extensive task to develop specific software for semi-infinite linear programming to solve the problem at hand if this software is also required to be stable and reliable with respect to numerical problems such as cycling phenomena, ill-conditioned matrix inversions, etc. On the other hand, the conventional finite-dimensional linear programming technique is well established and there is a plethora of good, numerically reliable and easily accessible softwares available [14, 15].

In order to overcome the difficulties mentioned above, we present in this paper an applied semi-infinite front-end for complex Chebyshev approximation which is based on conventional finite-dimensional linear programming. The essence of the new technique, justified by the Caratheodory dimensionality theorem, is to exploit the finiteness of the related Lagrange multipliers by adapting conventional finite-dimensional linear programming to the semi-infinite linear programming problem.

By the proposed front-end, the complex Chebyshev approximation problem can be solved taking advantage of the numerical stability and efficiency of the given linear programming software package. Furthermore, the optimization procedure is simple to describe theoretically and straightforward to implement in computer coding. The new design technique should therefore be highly accessible for most design engineers.

In order to illustrate the flexibility and numerical efficiency of the proposed design technique we have included several design examples concerning the optimization of a narrow-band base-station antenna array for mobile communication in the 450 MHz band.

The paper is organized as follows: In Section 2 we introduce the planar hexag-
onal antenna array for the far-field as a numerical example and formulate the design problem as a complex Chebyshev approximation problem. In Section 3 we formulate the associated semi-infinite linear programming problem and discuss briefly the existing semi-infinite simplex extension algorithms. The new semi-infinite front-end technique is then formally introduced and a comprehensive convergence proof is given. Section 4 describes the numerical design examples and Section 5 the summary. Appendix A and B contain proofs for the finiteness of related Lagrange multipliers and the convergence of the DNCA algorithm, respectively.
Chapter 2

Problem Formulation

To demonstrate the versatility of the semi-infinite front-end the optimization is performed from an application point of view. The design examples are taken from the mobile communication base-station area, or more precisely, design of an antenna array in the 450 MHz band. As a numerical example we consider the planar hexagonal array where the sensor elements are evenly distributed in sections on a hexagon, and with the incident wave front propagating in the same plane as the array, see Fig. 2.1. The rather unusual configuration also illustrates the generality and the applicability for arbitrary base functions (array response) for arbitrarily geometries.

We consider the far-field and narrow-band case where the phase of the wave front is given by \( e^{j(2\pi f_0 t - k^T r)} \) where \( f_0 \) is the frequency, \( t \) the time, \( k = -\frac{2\pi f_0}{c} (\cos \varphi, \sin \varphi) \) the wave vector, \( c \) the speed of wave propagation, \( \varphi \) the angle of incidence and \( r \) the evaluated spatial point, see Fig. 2.1. The hexagonal sectioned antenna array consists of antenna elements distributed along the ground-plane sections, see Fig. 2.2. One antenna element and the ground-plane together form a dipole. The dipoles have the spatial positions \((r_m, \varphi_m)\) where \( m = 0, \ldots, M - 1 \) and \( r_m \) and \( \varphi_m \) are the distance and angle, respectively, between the middle of the hexagon to the dipole centre. The phase center and origin of coordinates are located in the middle of the hexagon and the total array response \( H(\varphi) \) is given as

\[
H(\varphi) = \sum_{m=0}^{M-1} w_m a_m(\varphi)e^{j2\pi f_0 \frac{r_m}{c} \cos(\varphi_m - \varphi)}
\]  

(2.1)

where \( M \) is the total number of used/active dipoles in the antenna and \( w_m \) is a complex weight. The radiation characteristic for a dipole located in \((r_m, \varphi_m)\) is denoted \( a_m(\varphi) \). We only use three of the six array-sections at a time and consequently a subset of all dipoles are used. Depending on the angle of incidence \( \varphi \) the three nearest heading sections with angle \( \varphi_g \) are selected complying with the requirements in Table 2.1.
The dipole characteristic \( a_{dp}(\varphi) \) for one single (omnidirectional) antenna element shielded by the ground-plane is shown in Fig. 2.3 and is given by

\[
\begin{align*}
\left\{ \begin{array}{l}
a_{dp}(\varphi) = e^{j2\pi \frac{d_g}{\lambda} \cos \varphi} - e^{-j2\pi \frac{d_g}{\lambda} \cos \varphi}, & |\varphi| \leq \frac{\pi}{2} \\
0, & \text{otherwise}
\end{array} \right. \\
\end{align*}
\]

(2.2)

where \( d_g \) is the spacing between the antenna element and the ground-plane (shield) and \( \lambda \) is the design wavelength. Thus, one single dipole in the array consists of an antenna element placed in front of a ground-plane with a corresponding virtual mirror source behind the ground-plane. Together they form the characteristics of every individual dipole, as in Fig. 2.3. The dipole radiation depends on the antenna element placement with respect taken to the ground-plane, see Fig. 2.2.

In this formulation the distance between the source antenna element and the ground-plane is \( d_g = \frac{\lambda}{4} \), and consequently the quotient \( \frac{d_g}{\lambda} = \frac{1}{4} \). The mirror source will virtually appear with the opposite radiation sign (phase shifted \( 180^\circ \)) but at the same distance \( d_g = \frac{\lambda}{4} \) behind the ground-plane. The corresponding radiation pattern \( a_m(\varphi) \) placed at position \((r_m, \varphi_m)\) for element \( m \) in the array is given by \( a_m(\varphi) = a_{dp}(\varphi - \varphi_g) \). In this formulation \( \varphi_g \) is the angle heading to the ground-plane section in which antenna element \( m \) is positioned. Due to the ground-plane the resulting response \( a_m(\varphi) \) in the reverse direction and along the ground-plane sections (shield) is assumed to be zero. Each dipole \( a_m(\varphi) \) is controlled in the amplitude and phase domains by complex weights \( w_m \).

The complex antenna array response for the angle \( \varphi \) using vector notation is given by

\[
H(\varphi) = w^H d(\varphi) = \tilde{w}^T \tilde{d}(\varphi) \]

(2.3)

(2.4)

(2.5)

where the complex array vector \( w = \Re\{w\} + j\Im\{w\} \) is an \( M \times 1 \) array vector of complex coefficients \( w_m \) and \( d(\varphi) \) the corresponding array response vector of complex continuous and linear independent transfer functions \( d_m(\varphi) = \)

<table>
<thead>
<tr>
<th>Config. no.</th>
<th>Angle of incidence ( \varphi )</th>
<th>Selected sections Bold faced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-30^\circ \leq \varphi &lt; 30^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
<tr>
<td>2</td>
<td>(30^\circ \leq \varphi &lt; 90^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
<tr>
<td>3</td>
<td>(90^\circ \leq \varphi &lt; 150^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
<tr>
<td>4</td>
<td>(150^\circ \leq \varphi &lt; 210^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
<tr>
<td>5</td>
<td>(210^\circ \leq \varphi &lt; 270^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
<tr>
<td>6</td>
<td>(270^\circ \leq \varphi &lt; 330^\circ)</td>
<td>( \varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ] )</td>
</tr>
</tbody>
</table>

Table 2.1: Angle of incidence \( \varphi \) vs. selected sections \( \varphi_g \).
Complex Chebyshev Optimization Using Conventional...

\[ a_m(\varphi) e^{j2\pi f_0 \frac{m}{M} \cos(\varphi_m - \varphi)}, \quad m = 1, \ldots, M - 1. \]

The \( \tilde{w}^T \) and \( \tilde{d}(\varphi) \) in (2.4) are defined as is defined as in (2.5), respectively. Consequently, \( \tilde{w}^T \) is an \( N \times 1 \) real vector and \( \tilde{d}^T(\varphi) \) an \( N \times 1 \) complex vector where \( N = 2M \).

In order to make the passband extremely narrow, the passband in this formulation was restricted to a single point \( \varphi_p \). The stopband is defined as \( \Phi = [\varphi_p + \varphi_s, \varphi_p + 2\pi - \varphi_s] \).

### 2.1 The Design Specification

Consider the following design specification

\[
\begin{align*}
|H(\varphi)| & \leq \sigma(\varphi), \quad \varphi \in \Phi \\
H(\varphi_p) &= 1 
\end{align*}
\]

(2.6)

where \( \sigma(\varphi) \) is a prescribed strictly positive magnitude bound. The specification (2.6) is equivalent to the specification

\[
\begin{align*}
\max_{\varphi \in \Phi} \frac{1}{\sigma(\varphi)} |H(\varphi)| & \leq 1 \\
H(\varphi_p) &= 1 
\end{align*}
\]

(2.7)

which leads to the minimax design formulation

\[
\begin{align*}
\min_{H} \max_{\varphi \in \Phi} v(\varphi) |H(\varphi)| \\
H(\varphi_p) &= 1 
\end{align*}
\]

(2.8)

where \( v(\varphi) = \frac{1}{\sigma(\varphi)} \).

It is concluded that a solution to the specification (2.6) exists if and only if the optimal objective value in (2.8) is less than or equal to one. Hence, the optimization formulation (2.8) will give us the answer to the question if there exists a feasible solution to (2.6), and furthermore, if a solution to (2.6) exists we will obtain the solution which is furthest away from the upper bound in a "logarithmic" minimax sense.

To elaborate on this last property, let \( \delta \) denote the objective value in (2.8). The optimal solution to (2.8) will give us the smallest \( \delta \) such that \( |H(\varphi)| \leq \delta \sigma(\varphi) \), \( \forall \varphi \in \Phi \) or

\[
20 \log |H(\varphi)| \leq 20 \log \delta + 20 \log \sigma(\varphi), \quad \forall \varphi \in \Phi.
\]

(2.9)

Thus, if a feasible solution to (2.6) exists, the optimal solution to (2.8) will give us the solution to (2.6) which maximizes the minimum distance to the specification \( \sigma(\varphi) \) in decibels (dB). If a feasible solution to (2.6) does not exist, the solution to (2.8) will give us the solution which minimizes the maximum constraint violation in (2.6) in decibels (dB).
Figure 2.1: A planar hexagonal antenna array with shield for the far-field where the outer ‘•’ denote the sensor elements, the solid hexagon the ground-plane and the inner ‘○’ denote the mirror source. The passband (main-lobe look direction) is defined by the angle $\varphi_p$ ($\varphi_p = 0$ in the figure) and the stopband (side-lobe region) by the interval $[\varphi_p + \varphi_s, \varphi_p - \varphi_s + 360^\circ]$. The ground-plane sections are positioned in direction $\varphi_g = [0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ]$. 
Figure 2.2: One section in the hexagonal antenna array.
Figure 2.3: The dipole model for $a_{dp}(\varphi)$. 
Chapter 3

Semi-Infinite Linear Programming

The optimal solution to the minimax formulation in (2.8) is given by the equivalent formulation

\[
\begin{align*}
\min & \quad \delta \\
\text{s.t.} & \quad v(\varphi)|H(\varphi)| - \delta \leq 0, \quad \varphi \in \Phi \\
& \quad H(\varphi_p) = 1
\end{align*}
\]

(3.1)

where \( \delta \) is an additional real variable.

3.1 Semi-Infinite Linear Programming Formulation

The problem (3.1) corresponds to a non-linear optimization problem which is very difficult to treat as it stands. We will therefore convert (3.1) into a semi-infinite linear programming problem. According to the real rotation theorem [16], a magnitude inequality in the complex plane can be expressed in the equivalent form

\[
|z| \leq \sigma \iff \Re\{ze^{j\theta}\} \leq \sigma \quad \forall \theta \in [0, 2\pi]
\]

(3.2)

where \( z \) is a complex number and \( \sigma \) a real and positive number.

By making use of (3.2), the design problem (3.1) is now reformulated as

\[
\begin{align*}
\min & \quad \delta \\
\text{s.t.} & \quad v(\varphi)\Re(H(\varphi)e^{j\theta}) - \delta \leq 0, \quad (\varphi, \theta) \in \Phi \times \Theta \\
& \quad H(\varphi_p) = 1
\end{align*}
\]

(3.3)

In order to emphasize the linear structure of this formulation we finally rewrite
(3.3) as the following semi-infinite linear program

$$\begin{align*}
\min_{\delta} & \quad a^T(\varphi, \theta)\hat{w} - \delta \\ a^T(\varphi, \theta)\hat{w} - \delta & \leq 0, \quad (\varphi, \theta) \in \Phi \times \Theta
\end{align*}$$

where $a(\varphi, \theta) = v(\varphi)\Re(\hat{d}(\varphi)e^{j\theta})$, $P$ is an $L \times N$ constraint matrix and $p$ an $L \times 1$ constraint vector. The main-lobe constraint $H(\varphi_p) = 1$ in (3.3) is obtained by choosing $P^T = [\Re \{\hat{d}(\varphi_p)\} \ \Im \{\hat{d}(\varphi_p)\}]$ and $p^T = [1, 0]$.

The linear program (3.4) is called semi-infinite since the number of variables (unknowns) are finite but the constraint set is infinite. For practical purposes in the implementation of the optimization algorithm, it is assumed that the set $\Phi$ is finite. Note however, that the total corresponding approximation problem is with respect to the true complex error since the phase parameter $\theta$ belongs to the infinite set $\Theta = [0, 2\pi]$.

### 3.2 The Kuhn-Tucker Conditions

The necessary and sufficient Kuhn-Tucker conditions related to (3.4) are given by [10]

$$\begin{align*}
\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \int_{D} \begin{pmatrix} a(\varphi, \theta) \\ -1 \end{pmatrix} d\Lambda + \begin{pmatrix} P^T \\ 0^T \end{pmatrix} \mu &= 0 \\
\int_{D} (a^T(\varphi, \theta)\hat{w} - \delta) d\Lambda &= 0 \\
a^T(\varphi, \theta)\hat{w} - \delta &\leq 0, \quad (\varphi, \theta) \in D \\
P\hat{w} &= p \\
\Lambda &\geq 0
\end{align*}$$

where the Lagrange multiplier $\Lambda$ is a regular Borel measure [10], $D = \Phi \times \Theta$ and $\mu$ a vector of Lagrange multipliers. The vector $0$ denotes a zero vector of compatible dimension.

It can be shown [10] that the optimal (non-unique) measure $\Lambda$ satisfying (3.5)-(3.9) can always be represented by a measure with finite support (atomic measure) at no more than $N + 1$ points. The Kuhn-Tucker conditions may therefore be written

$$\begin{align*}
\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \sum_{i=1}^{r} \lambda_i \cdot \begin{pmatrix} a(\varphi_i, \theta_i) \\ -1 \end{pmatrix} + \begin{pmatrix} P^T \\ 0^T \end{pmatrix} \mu &= 0 \\
\sum_{i=1}^{r} \lambda_i (a^T(\varphi_i, \theta_i)\hat{w} - \delta) &= 0
\end{align*}$$
\[ \mathbf{a}^T(\varphi, \theta)\tilde{\mathbf{w}} - \delta \leq 0, \quad (\varphi, \theta) \in \mathcal{D} \]  
(3.12)

\[ \mathbf{P}\tilde{\mathbf{w}} = \mathbf{p} \]  
(3.13)

\[ \lambda_i \geq 0 \]  
(3.14)

where \((\varphi_i, \theta_i) \in \mathcal{D}, i = 1, \ldots, r \leq N + 1\) and the \(\lambda_i\)'s are the values of the atomic measure. The proof follows the same technique as given in [10] on page 73-76 by an application of Caratheodory theorem [17], see Appendix A.

We note that in our application the set \(\Phi\) is assumed to be finite and the conditions (3.10)-(3.14) can be motivated by a simple argument as follows: An active constraint \(|H(\varphi_i)| = \delta\) in (3.4) corresponds to a phase angle uniquely given by \(\theta_i = -\arg\{H(\varphi_i)\}\). Thus, there can be only a finite number of active constraints in (3.4).

The implication of the conditions (3.10)-(3.14) is extremely useful since it allows us to solve the problem (3.4) by considering a sequence of finite subsets \(\mathcal{R}_k = \{(\varphi_1, \theta_1), \ldots, (\varphi_r, \theta_r)\}\) consisting of no more than \(N + 1\) points of \(\mathcal{D} = \Phi \times \Theta\).

**Key observation 1** Since the number of variables is \(N + 1\), we note that an optimization software will usually give us a total of \(N + 1\) Lagrange multipliers greater than zero including the multipliers in \(\mu\). Therefore, the size of the so-called reference set \(\mathcal{R}_k\) is in fact \(r \leq N + 1 - L\). Note that we will come to the same conclusion if we transform the equality constraints in (3.8) and (3.13) to inequality constraints and apply Caratheodory theorem as in Appendix A.

### 3.3 The Dual Formulation

The **primal** formulation (3.4) can be rewritten using the Lagrangian as

\[
\min_{(\tilde{\mathbf{w}}, \delta) : \mathbf{P}\tilde{\mathbf{w}} = \mathbf{p}} \delta + \int_\mathcal{D} \left( \mathbf{a}^T(\varphi, \theta)\tilde{\mathbf{w}} - \delta \right) d\Lambda
\]  
(3.15)

and the corresponding **dual** formulation is then given by

\[
\max_{\Lambda \geq 0} \min_{(\tilde{\mathbf{w}}, \delta) : \mathbf{P}\tilde{\mathbf{w}} = \mathbf{p}} \left\{ \delta + \int_\mathcal{D} \left( \mathbf{a}^T(\varphi, \theta)\tilde{\mathbf{w}} - \delta \right) d\Lambda \right\}
\]  
(3.16)

where \(\Lambda\) is the Lagrange multiplier as in (3.5).

It can be shown that the two problems (3.15) and (3.16) above have the same optimal solution and the optimal values are the same, see e.g. [6]. Hence, there is no duality gap associated with (3.15) and (3.16) and the optimal solution is a saddle point.

Due to the Caratheodory dimensionality result mentioned above, we need only to consider measures with finite support and the dual formulation (3.16) can therefore be stated as

\[
\max_{(\varphi_i, \theta_i) \in \mathcal{D}} \min_{\lambda_1, \ldots, \lambda_r} \max_{(\tilde{\mathbf{w}}, \delta) : \mathbf{P}\tilde{\mathbf{w}} = \mathbf{p}} \left\{ \delta + \sum_{i=1}^r \lambda_i(\mathbf{a}^T(\varphi_i, \theta_i)\tilde{\mathbf{w}} - \delta) \right\}
\]  
(3.17)
where \( \lambda_i \geq 0, \ i = 1, \ldots, r \leq N + 1 - L \).

Since (3.17) can also be written

\[
\max_{(\varphi_i, \theta_i) \in D} \min_{\lambda_1, \ldots, \lambda_r} \max \left\{ \delta + \sum_{i=1}^{r} \lambda_i (a^T(\varphi_i, \theta_i)\hat{w} - \delta) \right\}
\]

we finally conclude that the dual formulations (3.16) and (3.17) can also be formulated as

\[
\max_{(\varphi_i, \theta_i) \in D} \left\{ \min \delta \ a^T(\varphi_i, \theta_i)\hat{w} - \delta \leq 0, \ i = 1, \ldots, r \right. \\
\left. P\hat{w} = p \right\}
\]  

(3.19)

**Key observation 2**  
The dual formulation (3.19) suggests that the primal problem (3.4) can be solved by considering a sequence of subproblems as in (3.19) with increasing minimum cost and which is based only on finite subsets \( R_k = \{(\varphi_1, \theta_1), \ldots, (\varphi_r, \theta_r)\} \) consisting of no more than \( N + 1 - L \) points of \( D = \Phi \times \Theta \). This observation constitutes the foundation for the development of the optimization algorithm described below.

### 3.4 The Semi-Infinite Simplex Algorithm

The optimization problem (3.4) can be solved by employing a semi-infinite simplex algorithm as described in e.g. [6, 10, 11]. Such an algorithm aims at finding the optimal Lagrange multipliers \( \lambda_i \) and \( \mu_i \) in (3.10)-(3.14) by directly applying the simplex algorithm to the dual problem of (3.4), see e.g. [11].

The applicability of this approach relies on two important theoretical results on semi-infinite linear programming. The first result regards the finite dimensionality of the dual formulation which is essentially the same as the finite dimensionality of the related Lagrange multipliers (3.10)-(3.14) mentioned in the previous section. The second result regards the strong duality for the problem formulation (3.4) and its dual, which basically means that the two dual problems have the same optimal value. The strong duality ensures that the solution to (3.4) can be obtained by solving the related dual problem, cf. [10].

At each step of the semi-infinite simplex algorithm, the dual variables (Lagrange multipliers) are solved for by inverting the so called simplex basis which corresponds to a square system of linear equalities defined by a subset of \( D \). Further, a pivoting takes place in which a new constraint is chosen to enter the basis and an “old” active constraint is chosen to leave the basis. The constraint index \((\varphi_e, \theta_e)\) which is chosen to enter the basis is usually defined by the maximum constraint violation in (3.4) and can thus be stated as

\[
(\varphi_e, \theta_e) = \arg \max_{\varphi, \theta} \left\{ a^T(\varphi, \theta)\hat{w} - \delta \right\}
\]

(3.20)
assuming that the maximum value in (3.20) is non-negative (otherwise the current solution is optimal due to the duality theorem). The constraint index chosen to leave the basis is given by the so called ratio test which is designed so as to maintain feasibility of the dual variables at each step of the algorithm.

It can be shown that the semi-infinite simplex algorithm referred to above converges [6, 10, 11]. During the iteration, and corresponding to the current simplex basis, the cost function for the (feasible) dual problem is equal to the cost $\delta$ for the (infeasible) primal problem (3.4). This cost is monotonically increasing and converges to the optimal value for the dual problem provided that no numerical difficulties are encountered such as cycling or ill-conditioned basis matrices etc., cf. [10]. It can be shown that when the dual solution is optimum, the corresponding primal solution is feasible and is therefore optimum due to the duality theorem [10].

3.5 Complex Chebyshev Optimization using Conventional Linear Programming

The approach taken in this paper is different from the semi-infinite simplex algorithm described in the previous section. We will describe and prove the convergence of a semi-infinite linear programming algorithm solely based on subproblems of (3.4) and without the explicit reference to the related dual formulation. At each iteration of this algorithm, a subproblem to (3.4) is solved using conventional linear programming. It is important to note that each subproblem requires no more than $N + 1$ linear constraints according to the finite dimensionality property of the related Lagrange multipliers as described in Section 3.2.

Before we proceed and describe the proposed algorithm in detail, we comment on the following three main advantages of the new algorithm in comparison to the existing semi-infinite simplex algorithm: The new algorithm is

1. Simple to describe theoretically.

2. Simple to implement practically.

3. Numerically reliable.

The first two items above are due to the fact that the theoretical and numerical issues and problems related to the calculation of the dual variables (Lagrange multipliers) is subordinated to a subroutine for conventional linear programming. It is assumed that this subroutine accurately calculates and returns the required Lagrange multipliers for each subproblem. The optimization problem is therefore simple to treat from a theoretical point of view. It is furthermore assumed that the subroutine handles the ratio test as well as the numerical difficulties associated
with linear programming such as cycling phenomena, matrix inversion, accuracy, etc. The software development is therefore simple and straightforward to pursue.

The third item above is due to the fact that the proposed optimization DNCA procedure itself is a stable procedure. Furthermore, the DNCA inherits the good numerical properties of the linear programming subroutine. It is thereby assumed that we are in the possession of a good, numerically stable and reliable software for conventional linear programming.

It may be noted that since the simplex algorithm is a fairly general concept [6], the proposed algorithm could in fact be regarded as a generalized version of the semi-infinite simplex algorithm. However, the present formulation strongly emphasizes the motivating points 1-3 above.

The DNCA computer code is significantly simplified in comparison with a computer code which is tailored for semi-infinite linear programming. Moreover, the computational complexity is asymptotically the same.

### 3.6 The DNCA-LP Optimization Algorithm

The proposed semi-infinite Dual Nested Complex Approximation (DNCA) Linear Programming (LP) algorithm to solve (3.3) proceeds with the following basic steps:

1. Given a reference set \( R_k = \{ (\varphi_1, \theta_1), \ldots, (\varphi_r, \theta_r) \} \subset D = \Phi \times \Theta \), let \((\tilde{w}_k, \delta_k)\) and \( H_k(\varphi) = \tilde{w}_k^T \tilde{d}(\varphi) \) denote the optimal solution to the subproblem

\[
\begin{align*}
\min & \delta \\
\text{subject to} & v(\varphi_i) \Re(H(\varphi_i)e^{j\theta_i}) - \delta \leq 0, & (\varphi_i, \theta_i) \in R_k \\
& H(\varphi_p) = 1
\end{align*}
\]

and let \( \lambda_1, \ldots, \lambda_r \) denote the corresponding Lagrange multipliers. The subproblem (3.21) is solved by using conventional finite-dimensional linear programming techniques.

2. Define the entering index \((\varphi_e, \theta_e)\) by

\[
(\varphi_e, \theta_e) = \arg \max_{\varphi, \theta} \{ v(\varphi) \Re(H_k(\varphi)e^{j\theta}) \} \tag{3.22}
\]

where

\[
\begin{align*}
\varphi_e & = \arg \max_{\varphi} \{ v(\varphi) |H_k(\varphi)| \} \\
\theta_e & = - \arg(H_k(\varphi_e))
\end{align*}
\]

and calculate \( \|H_k\|_{\infty} = \max \{ v(\varphi) |H_k(\varphi)| \} = v(\varphi_e) \Re(H_k(\varphi_e)e^{j\theta_e}) \).
3. Stop if
\[ \|H_k\|_\infty < \delta_k(1 + \varepsilon) \] (3.25)
where \( \varepsilon \) is a predefined tolerance parameter. If (3.25) is satisfied then
\[ \|H_k\|_\infty < \delta_o(1 + \varepsilon) \] where \( \delta_o \) is the optimal Chebyshev deviation related to the problem (3.3).

4. Define the leaving indices by
\[ R_l = \{ (\varphi_i, \theta_i) \in R_k \mid \lambda_i = 0 \} \] (3.26)
which essentially consists of the inactive constraints in (3.21).

5. Define the new reference set by
\[ R_{k+1} = (R_k \setminus R_l) \cup R_e \] (3.27)
and return to step 1.

To ensure that the first subproblem is numerically well-conditioned the choice of starting reference \( R_1 \subset D \) is important. We suggest that this is done by using a finite-dimensional linear programming technique using discretization similar to [16] pp. 120-129. For this purpose the phase variable \( \theta \) is restricted to \( \theta \in \{0, \frac{\pi}{3}, \frac{2\pi}{3}\} \) and the initial starting reference is given by
\[ R_1 = \{ \varphi_1, \ldots, \varphi_q \} \times \{0, \frac{\pi}{3}, \frac{2\pi}{3}\} \] (3.28)
where \( q \geq \left\lceil \frac{N-1}{3} \right\rceil \) so that \( r = 3q \geq N - 1 \). The points \( \varphi_1, \ldots, \varphi_q \) are uniformly sampled in the domain \( \Phi \).

The optimal variable \( \delta_o \) given by the optimization process yields the amplitude margin with respect to the desired amplitude function \( \sigma(\varphi) \). Three different cases are possible:

1. \( \delta_o < 1 \Rightarrow \) the solution satisfies the specification given by \( 20 \log \sigma(\varphi) \) [dB] with a (maximized) minimum amplitude margin of \( 20 \log \delta \) [dB].

2. \( \delta_o = 1 \Rightarrow \) the solution satisfies the specification given by \( 20 \log \sigma(\varphi) \) [dB] exactly, without margin.

3. \( \delta_o > 1 \Rightarrow \) the solution violates the specification given by \( 20 \log \sigma(\varphi) \) [dB] with a (minimized) maximum of \( 20 \log \delta_o \) [dB].

**Key observation 3** The number of variables is \( N + 1 \) and the size of the reference set \( R_k \) is only \( r \leq N + 1 - L \). The constraint index \( R_e = (\varphi_e, \theta_e) \) which is chosen to enter the basis \( R_k \) is usually defined by the maximum constraint violation. Hence, this entering constraint \( R_e \) is very likely to be independent of the small reference set \( R_k \). This is the primary reason ensuring that the DNCA itself is a highly numerical stable procedure.

In Appendix B we give a “convergence proof” for the optimization algorithm described in this section.
3.7 Matlab Structure

The MATLAB™ structure of the DNCA is briefly outlined in Appendix C. This description corresponds to the MATLAB help command of the semi-infinite Dual Nested Complex Approximation Linear Programming (DNCALP) implementation. This is a basic and efficient way to understand the syntax and behavior of DNCALP optimization function.

The solution of the hexagonal antenna array problem as defined in Section 2 is thus obtained by the DNCA algorithm as described in Section 3. The DNCALP-function with the input and output arguments is stated below (3.29)-(3.32). In Table 3.1 the corresponding sizes and classes are displayed.

\[
f = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} v(\varphi_1)\tilde{d}^H(\varphi_1) & 0 \\ \vdots & \vdots \\ v(\varphi_L)\tilde{d}^H(\varphi_L) & 0 \end{bmatrix}, \quad b = c = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & -1 \end{bmatrix}
\]

Theta = [] (or Theta = Inf), Aeq = \( \tilde{d}^H(\varphi_P) \), beq = [0] (3.30)

\[
LB = UB = X_0 = TOL = K = OPTIONS = []
\]

(3.31)

The DNCALP function \( X = \text{DNCALP}(f, A, b, \text{Theta}, c, D, \text{Aeq}, \text{beq}, LB, UB, X_0, TOL, K, OPTIONS) \) returns the vector

\[
X = \begin{bmatrix} \tilde{w} \\ \delta \end{bmatrix} = \begin{bmatrix} \tilde{w}_1 \\ \vdots \\ \tilde{w}_N \\ \delta \end{bmatrix}
\]

(3.32)
<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>(N+1) × 1</td>
<td>Array (real)</td>
</tr>
<tr>
<td>f</td>
<td>(N+1) × 1</td>
<td>Array (real)</td>
</tr>
<tr>
<td>A</td>
<td>I × (N+1)</td>
<td>Array (complex)</td>
</tr>
<tr>
<td>b</td>
<td>I × 1</td>
<td>Array (real)</td>
</tr>
<tr>
<td>Theta</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>c</td>
<td>I × 1</td>
<td>Array (complex)</td>
</tr>
<tr>
<td>D</td>
<td>I × (N+1)</td>
<td>Array (real)</td>
</tr>
<tr>
<td>Aeq</td>
<td>1 × (N+1)</td>
<td>Array (complex)</td>
</tr>
<tr>
<td>beq</td>
<td>1 × 1</td>
<td>Array (complex)</td>
</tr>
<tr>
<td>LB</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>UB</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>X0</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>TOL</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>K</td>
<td>0 × 0</td>
<td>Array (real)</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>0 × 0</td>
<td>Char Array</td>
</tr>
</tbody>
</table>

Table 3.1: The corresponding MATLAB vector sizes and classes where I is the initial angular grid resolution.
Chapter 4

Design Examples

As an application example we consider the planar hexagonal array as defined in Section 2. If the weighting function $v(\varphi)$ is uniformly distributed we achieve an equiripple solution, that is we achieve the lowest possible side-lobe level in the stopband with respect taken to one or more linear constraints. The array response used is given by (2.1) and the corresponding real variables used, $\tilde{w}_m$, are defined as in (2.4). The specification in (2.6) is used to state the desired array design. The solution is obtained by using the DNCA algorithm as described in Section 3. The performance of using a hexagonal sectioned ground-plane shielded antenna array as in Fig. 2.1 is investigated. The individual sensor responses, $a_m(\varphi)$, corresponds to a dipole heading in the direction $\varphi_g$. Each dipole, $a_m(\varphi)$, is controlled in the amplitude and phase domains by a complex weight $w_m$. Due to the shielding the resulting dipole response $a_m(\varphi)$ behind and along the shield ground-plane sections is zero in this evaluation, see Fig. 2.3.

The examples show the flexibility in design using the proposed semi-infinite front-end algorithm. The algorithm is capable of solving huge design problems with many antenna elements, see Figs. 4.1-4.2. Main-lobe steering, see Fig. 4.3-4.4, and side-lobe control by incorporating arbitrary weighting functions, see Figs. 4.5-4.7, can at the same time be taken into consideration during the optimization process as well. All figures and examples show the corresponding array pattern using configuration no. 1 as described in Section 2.

All antenna array responses are designed for the frequency $f_0 = 450$ MHz and the interspacing between antenna elements in the array is $d = \lambda/2 \approx 0.3$ meter. The distance between the source element and the ground-plane is $d_g = \frac{\lambda}{4} \approx 0.15$ meter. Note that by choosing $f_0$ the design problem can easily be scaled or translated into different frequency bands.

Fig. 4.1 illustrates a typical equiripple solution for an antenna with $M = 102$ elements. Each linear antenna section consists of 34 ($M = 3 \cdot 34 = 102$) isotropic dipole elements. The antenna look direction for the main-lobe is $\varphi_p = 0^\circ$, that is the main-lobe (passband) consists only of one point and is defined by one single point constraint $H(\varphi_p) = H(0) = 1$. The radius of the antenna
is approximately 9 meters. There are 102 complex weights $w$, which implies that $N = 204$ real variables $\tilde{w}$ are involved in the optimization process. The corresponding convergence behavior in Fig. 4.2 shows the advance of variable $\delta$ which is continuously increasing to the optimal value $\delta_o$. The maxnorm $\|H\|$ has a more irregular but still overall decreasing behavior also approaching the optimal value $\delta_o$. It is in accordance as described in Section 3 that $\delta < \delta_o < \|H\|$ during the optimization process. In this design example the side-lobe (stopband) region starts at $\pm1.5^\circ$, and in all the other examples at $\pm5.5^\circ$ from the desired main-lobe direction. The angular grid resolution is $0.25^\circ$ in this example and $0.5^\circ$ in all others.

In the examples concerning the main-lobe steering the look direction $\varphi_p$ for the main-lobe is gradually increased from $0^\circ$ to $35^\circ$ in steps of $5^\circ$. The main-lobe look direction is defined by one single point constraint $H(\varphi_p) = 1$, see Figs. 4.3-4.4. Each linear antenna section consists of 10 ($M = 3 \cdot 10 = 30$) isotropic dipole elements. The unusual symmetry in the hexagonal antenna compared to a circular design is obvious. However, by using the proposed design method it is possible to apply lobe steering to direct the main-lobe in the angular domain $\varphi$. The drawback is that several sets of weights must be used. Figs. 4.3-4.4 illustrate the performance of the 3-sectioned hexagonal antenna using 30 antenna elements for main-lobe directions $\varphi_p \in [0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ]$. The radius of the antenna is approximately 3 meters. The loss in stopband attenuation performance between the $0^\circ$ and $30^\circ$ design is $\sim 3$ dB. Note that lobe steering in the direction $35^\circ$ degrees can be obtained by counterclockwise switching antenna sections as described in Table 2.1 from configuration no.1 to no.2 and reuse a mirrored weight setup of the $25^\circ$ design. In this way, the number of weight sets will be kept reasonably low.

The ability to incorporate a weighting function $v(\varphi) = \frac{1}{\sigma(\varphi)}$ is a useful option when designing antenna arrays $M = 30$, see Figs. 4.5-4.8. The array response in the angular domain $\varphi$ can in that way be shaped in an arbitrary sense. The antenna look direction for the main-lobe is $\varphi_p = 0^\circ$. In the design example illustrated in Fig. 4.5 we define four different side-lobe regions: two near side-lobe regions (one left near side-lobe region and one right side-lobe region) and two far side-lobe regions. The corresponding desired near side-lobe levels are asymmetrically chosen (left) $-15$ dB and (right) $-20$ dB and the far side-lobe levels are symmetrically chosen (left) $-35$ dB and (right) $-40$ dB. The convergence behavior illustrated in Fig. 4.6 corresponds to the weighted design optimization process.

The weighting function $v(\varphi) = \frac{1}{\sigma(\varphi)}$, $\varphi \in \Phi$ can be chosen arbitrarily. In Fig. 4.7 a non-linear weighting is used. The weighting function is linear in dB (that is, non-linear in linear scale) defined by

$$
20 \log \sigma(\varphi) = k_1 \varphi + c_1 \text{ dB}, \quad \varphi \in [\varphi_p + \varphi_s, \varphi_p + \pi]
$$
\[ 20 \log \sigma(\varphi) = k_2 \varphi + c_2 \text{ dB}, \quad \varphi \in [\varphi_p + \pi, \varphi_p - \varphi_s + 2\pi] \] (4.2)

where \( k_1 = \frac{-25}{\pi - \varphi_s}, \ c_1 = -20 + \frac{25\varphi_s}{\pi - \varphi_s}, \ k_2 = \frac{25}{\pi - \varphi_s}, \) and \( c_2 = -45 - \frac{25\varphi_s}{\pi - \varphi_s}. \) The optimization complexity using a weighting function differing in shape see Fig. 4.7 or a uniform one see Fig. 4.5 is equal. The optimization process converges in the same manner and needs approximately 1000 iterations to reach the desired tolerance \( \varepsilon = 0.0001 \) only \(-80\) dB from the theoretically optimal solution, that is the accuracy is more than sufficient for this design, since \(-40\) dB is designed and we have approximately \(-80\) dB numerical error.

The final value of variable \( \delta \) yields information about the amplitude margin with respect to the function \( \sigma(\varphi) \) and is plotted as \( \delta_o \) in the convergence behavior plots. This value of variable \( \delta \) is an important design parameter. If the uniform weighting function \( \sigma(\varphi) = 1 \) (0 dB) is used we simply achieve the maximum side-lobe suppression by observing the final value of variable \( \delta \).

It is possible to add additional point constraints such as nulls in the stopband or constraints on the main-lobe.
Figure 4.1: Minimax design of an hexagonal antenna using 3 sections containing 34 antenna elements each. Side-lobe suppression \( \approx 18.5 \) dB. The design example consists of total \( M = 102 \) complex weights i.e. \( N = 204 \) real variables in the optimization.

Figure 4.2: The monotonic convergence behaviour of \( \delta \) corresponding to the minimax design in fig. 4.1. The fluctuating maxnorm \( \|H\| \) and \( \delta \) will converge to the same optimal value \( \delta_0 \)
Figure 4.3: Main-lobe steering $\varphi_p \in [0^\circ, 5^\circ, 10^\circ, \ldots, 35^\circ]$. Minimax design of an hexagonal antenna using 3 sections containing 10 antenna elements each using different passband angles $\varphi_p$.

Figure 4.4: Optimum side-lobe suppression in dB vs. main-lobe steering angle/direction $\varphi_p$ corresponding to fig. 4.3.
Figure 4.5: Minimax design showing the capability of using a uniform weighting functions $\sigma(\varphi)$ in the far and near side-lobe region. In this case a non-symmetric weighting function is used in the left and right near side-lobe regions.

Figure 4.6: Convergence behavior corresponding to the design in Fig. 4.5.
Figure 4.7: Minimax design showing the capability of using a non-linear weighting function $\sigma$ in the side-lobe region.

Figure 4.8: Convergence behavior corresponding to the design in Fig. 4.7.
Chapter 5

Summary and Further Improvements

This paper solves antenna array Chebyshev approximation problems by exploiting Caratheodory dimensionality theorem in a conventional linear programming software front-end. The semi-infinite linear programming theory is fairly recent[10] and to the authors’ knowledge, there is no commercial software available for efficient numerical solution of the problem (3.3). It is therefore an extensive task to develop a specific software for semi-infinite linear programming to solve (3.3) if this software is also required to be stable and reliable with respect to numerical problems such as cycling phenomena, ill-conditioned matrix inversions, etc. On the other hand, the conventional finite-dimensional linear programming technique is well established and there is a lot of good, numerically reliable software available. The front-end optimization technique proposed in this paper is therefore a convenient alternative which inherits the good numerical properties of the given linear programming subroutine. The front-end computer code is significantly simplified in comparison with computer code which is tailored for semi-infinite linear programming. Moreover, the computational complexity is asymptotically equal. The proposed method is capable of solving large optimization problems such as huge antenna arrays and optimization variables. Extensive evaluations indicate the flexibility in design using the proposed front-end method.

Further possible improvements to extend the formulation to make it possible to take the coupling between antenna elements into consideration. This can be done either by using measurements from a real setup or by incorporating a coupling model. If an approximation of the error function is known (which normally is the case in the antenna array application) it will be possible to create a multiple exchange front-end. A multiple exchange algorithm will probably converge in a faster manner as compared to the single exchange algorithm.

The Dual Nested Complex Approximation Linear Programming described in this paper applied to antenna arrays can probably be a subject for an interdisciplinary exchange, since the method is by no means restricted to antenna array
or FIR filter design.
Acknowledgments

This work has been supported by NUTEK, the National Swedish Board for Technical Development.
Appendix A

Finiteness of Lagrange multipliers

In this section we briefly outline the proof regarding the finiteness of the Lagrange multipliers related to (3.3) as described in Section 3.6.

Rearranging (3.5) as

\[- \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} P^T \\ 0^T \end{pmatrix} \mu = \int_D \begin{pmatrix} a(\varphi, \theta) \\ -1 \end{pmatrix} d\Lambda \quad (A.1)\]

where \( \Lambda \geq 0 \) and following the proof of Theorem 4.6 in [10], it is concluded that the vector on the left hand side of (A.1) belongs to the convex cone \( C \) generated by the compact set

\[ G = \left\{ \begin{pmatrix} a(\varphi, \theta) \\ -1 \end{pmatrix} : (\varphi, \theta) \in D \right\} \subset R^{N+1}. \quad (A.2) \]

Caratheodory theorem [17] states that a vector \( x \in R^{N+1} \) belongs to the convex cone \( C \) generated by a set \( G \subset R^{N+1} \) if and only if \( x \) can be expressed as a non-negative linear combination of \( N + 1 \) vectors in \( G \). Hence, we may take \( \Lambda \) as an atomic measure with positive masses \( \lambda_i \) concentrated at no more than \( N + 1 \) points \( (\varphi_i, \theta_i) \in D \) as in (3.10).
Appendix B

Convergence Proof

In this Appendix we give a “convergence proof” for the DNCA optimization algorithm. The proof is only briefly outlined and the emphasis is on an intuitive understanding for the optimization procedure. We show that this can be accomplished without explicit reference to the conceptually abstract dual formulation.

Assume that we are given the solution $(\tilde{w}_k, \delta_k)$ to subproblem given by (3.21). This solution will remain unchanged if we delete the constraints corresponding to zero Lagrange multipliers (basically the inactive constraints). This can be readily seen by considering the Kuhn-Tucker conditions for the subproblem (3.21). Thus, the problem

\[
\begin{align*}
\min \delta \\
v(\varphi_i)\Re(H(\varphi_i)e^{j\theta_i}) - \delta &\leq 0, \quad (\varphi_i, \theta_i) \in \mathcal{R}_k \setminus \mathcal{R}_l \\
H(\varphi_p) & = 1 \\
\end{align*}
\]

(B.1)

has the same solution $(\tilde{w}_k, \delta_k)$ as (3.21).

If we now add one single constraint row as below

\[
\begin{align*}
\min \delta \\
v(\varphi_i)\Re(H(\varphi_i)e^{j\theta_i}) - \delta &\leq 0, \quad (\varphi_i, \theta_i) \in \mathcal{R}_k \setminus \mathcal{R}_l \\
v(\varphi_e)\Re(H(\varphi_e)e^{j\theta_e}) - \delta &\leq \|H_k\|_\infty - \delta_k, \\
H(\varphi_p) & = 1 \\
\end{align*}
\]

(B.2)

we will still have the same solution $(\tilde{w}_k, \delta_k)$ as in (3.21). This is due to the fact that $v(\varphi_e)\Re(H_k(\varphi_e)e^{j\theta_e}) = \|H_k\|_\infty$ and the last added constraint row is thus satisfied with equality for the particular solution $(\tilde{w}_k, \delta_k)$. Furthermore, we may associate a new Lagrange multiplier with value zero to this new constraint, and
the solution \((\tilde{w}_k, \delta_k)\) is therefore optimum for (B.2) due to the Kuhn-Tucker conditions.

The solution \((\tilde{w}_{k+1}, \delta_{k+1})\) of the next iteration is given by the following formulation

\[
\begin{align*}
\min \delta \\
v(\varphi_i) \Re(H(\varphi_i)e^{j\vartheta_i}) - \delta & \leq 0, \quad (\varphi_i, \theta_i) \in R_k \setminus R_l \\
v(\varphi_p) \Re(H(\varphi_p)e^{j\vartheta_p}) - \delta & \leq 0, \\
H(\varphi_p) & = 1
\end{align*}
\]

(B.3)

where we have put the right hand side of the entering constraint in (B.2) equal to zero.

We now establish the following important inequality relations

\[
\delta_k \leq \delta_o \leq \|H_k\|_\infty.
\]

(B.4)

The first inequality is due to the fact that \(\delta_o\) is the minimum value for the problem (3.3) and \(\delta_k\) is the minimum value for the subproblem (3.21) where the constraint set constitutes a proper subset of the constraint set for (3.3). The second inequality in (B.4) is due to the fact that \(\delta_o\) is the optimum Chebyshev deviation.

Assuming that the solution \((\tilde{w}_k, \delta_k)\) is not optimum for the problem (3.3), we see from (B.4) that \(\|H_k\|_\infty - \delta_k > 0\). Thus, the constraints in (B.3) are more restrictive than the constraints in (B.2) and the feasible region for (B.3) is a proper subset of the feasible region for (B.2). We therefore conclude that

\[
\delta_{k+1} \geq \delta_k
\]

(B.5)

and convergence is therefore guaranteed since the sequence \(\delta_k\) is monotonically increasing and upper bounded. Convergence to the optimum value \(\delta_k \to \delta_o\) is motivated by Section 3.2 where the existence of a finite optimal reference set \(R_{opt}\) is established.

We note that in this context, convergence of the algorithm is based on the assumption that numerical difficulties (such as cycling, etc.) are avoided due to the presence of a reliable software for the solution of (3.21). In practice however, any software may encounter numerical difficulties if the problem (3.3) is numerically ill-conditioned.

In practice, the optimal value \(\delta_o\) of (3.3) will never be reached. However, we may obtain a solution with arbitrary accuracy as follows. Let \(\varepsilon\) denote a positive tolerance parameter. Typically, the value of \(\varepsilon\) may be e.g. \(\varepsilon = 0.01\) meaning 1% error tolerance in the approximation \(\|H_k\|_\infty\) of the optimal value \(\delta_o\). From (B.4) we draw the following conclusion

\[
\|H_k\|_\infty < \delta_k(1 + \varepsilon) \Rightarrow \|H_k\|_\infty \leq \delta_o(1 + \varepsilon)
\]

(B.6)
which yields a practically useful stopping criteria for the optimization algorithm.

We note that the algorithm outlined above can be regarded as a generalized version of the simplex algorithm or as a cutting plane algorithm, cf. [14]. In our presentation however, we emphasize the simplicity and efficiency obtained when the corresponding exchange rule is based directly on the given Lagrange multipliers.
%DNCALP Solves complex semi-infinite Chebyshev constrained optimization
% problems using the Dual Nested Complex Approximation (DNCA)
% Algorithm.
%
% Requirements: MATLAB Optimization Toolbox Version 2.0 (or later)
%
% X=DNCALP(f,A,b) solves the complex linear-programming problem using
% the DNCA algorithm:
%  
%          min f'*x  subject to:  |A*x| <= b
%         x
%  
% X=DNCALP(f,A,b,Theta) solves the complex linear-programming
% problem using the DNCA algorithm:
%  
%          min f'*x  subject to:  Re(A*x*EXP(i*Theta)) <= b
%         x
%  
% For complex semi-infinite linear programming: Theta=Inf (=[]) (Default)
%  
% For complex finite dimensional linear programming: Theta=[W1 W2 ... WN]
% where 0<=Theta<=2*pi
%  
% For absolute real valued linear programming: Theta=[0,pi]  =>
%  Re(A*x)<=b and -Re(A*x)<=b
%  
% For real valued linear programming: Theta=0  =>  Re(A*x)<=b
%  
% X=DNCALP(f,A,b,Theta,c,D,Aeq,beq) solves the problem above while
% additionally satisfying the equality complex constraints Aeq*x = beq.
%  
%          min f'*x  subject to:  Re((A*x+c)*EXP(i*Theta))+D*x <= b
% x
%
X=DNCALP(f,A,b,Theta,c,D,Aeq,beq,LB,UB) defines a set of lower and upper
% bounds on the design variables, X, so that the solution is in
% the range LB <= X <= UB. Use empty matrices for LB and UB
% if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below;
% set UB(i) = Inf if X(i) is unbounded above.
%
X=DNCALP(f,A,b,Theta,c,D,Aeq,beq,LB,UB,X0) sets the starting point to X0.
% This option is only available with the active-set algorithm. Use empty
% matrix for X0 if no starting point exist.
%
X=DNCALP(f,A,b,Theta,c,D,Aeq,beq,LB,UB,X0,TOL) sets the DNCA final
% tolerance TOL > 0 such as max(Re((A*x+c)*EXP(i*Theta))+D*x-b) <= TOL
% (Default TOL = 0.01)
%
X=DNCALP(f,A,b,Theta,c,D,Aeq,beq,LB,UB,X0,TOL,K) sets the maximum number
% of DNCA steps K. If K = Inf the DNCA will iterate until the tolerance TOL
% is reached. (Default K = 500).
%
X=DNCALP(f,A,b,Theta,c,D,Aeq,beq,LB,UB,X0,TOL,K,OPTIONS) minimizes with
% the default optimization parameters replaced by values in the structure
% OPTIONS, an argument created with the OPTIMSET function. See OPTIMSET
% for details.
%
[X,FVALS]=DNCALP(f,A,b) returns the values of the objective function for
% each DNCA step k: FVALS(k) = f'*X(k). The final value for the objective
% function at X: FVALS(end)=f'*X
%
[X,FVALS,EXITFLAGS] = DNCALP(f,A,b) returns EXITFLAGS that describes the
% exit conditions of the underlying linear-programming LINPROG for each
% DNCA step k.
%
If EXITFLAGS(k) is:
% > 0 then LINPROG converged with a solution X(k).
% = 0 then LINPROG reached the maximum number of LINPROG-iterations
% without converging.
% < 0 then the problem was infeasible or LINPROG failed.
%
If EXITFLAGS(end) is:
% > 0 then DNCA converged with a solution X.
% = 0 then DNCA reached the maximum number of DNCA-iterations
% without converging.
% < 0 then the problem was infeasible or DNCA failed.
Bibliography


