Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Mosso-Vázquez, Alejo; Juárez-Romero, David; Cruz-Chávez, Marco Antonio; Sucar, Luis Enrique
A KKT Simplex Method for Efficiently Solving Linear Programs for Grasp Analysis Based on the Identification of Nonbinding Constraints
Computación y Sistemas, vol. 18, núm. 2, 2014, pp. 225-242
Instituto Politécnico Nacional
Distrito Federal, México

A KKT Simplex Method for Efficiently Solving Linear Programs for Grasp Analysis Based on the Identification of Nonbinding Constraints

Alejo Mosso-Vázquez[1], David Juárez-Romero[1], Marco Antonio Cruz-Chávez[1], and Luis Enrique Sucar[2]

[1] Centro de Investigación en Ingeniería y Ciencias Aplicadas, Cuernavaca, Morelos, Mexico

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Tonantzintla, Puebla, Mexico

{alejo.mosso, djuarez7}@gmail.com, mcruz@uaem.mx, esucar@inaoep.mx

Abstract: A one-phase efficient method to solve linear programming (LP) problems for grasp analysis of robotic hands is proposed. Our method, named as KKT Simplex method, processes free variables directly while choosing the entering and leaving variables, which makes it a one-phase method able to start at any point of the set of feasible solutions. Besides, the proposed method lowers the number of simplex steps by an angular pricing strategy to choose the entering variable. Moreover, the method reduces the size of an LP problem by the identification of nonbinding constraints that preserves the Karush-Kuhn-Tucker (KKT) cone. We developed the KKT Simplex method by incorporating to the well-known revised simplex method the following components: a method to process free variables, a pricing strategy, and an identification method. We solve LP problems of grasp analysis to test the efficiency and the one-phase nature of the proposed method.

Keywords. KKT Simplex method, linear programming, grasp analysis, nonbinding constraints.

Un método simplex KKT para resolver eficientemente programas lineales para análisis de la sujeción basado en la identificación de restricciones no atadas

Resumen: Se propone un método eficiente de una fase para resolver problemas de programación lineal (LP) para análisis de la sujeción por manos robóticas. El método, nombrado como método Simplex KKT, procesa variables libres directamente mientras selecciona las variables entrante y saliente, lo que lo convierte en un método de una fase que es capaz de iniciar en cualquier punto del conjunto de soluciones factibles. Además, el método disminuye el número de pasos simplex por una estrategia angular de costo para seleccionar la variable entrante. Aún más importante, el método reduce el tamaño del problema LP por identificación de restricciones no atadas que preserva el cono Karush-Kuhn-Tucker (KKT). Desarrollamos el método Simplex KKT por la incorporación al bien conocido método simplex revisado de los siguientes componentes: un método para procesar variables libres, una estrategia de costo, y un método de identificación. Resolvemos problemas LP de análisis de la sujeción para probar la eficiencia y la naturaleza de una fase del método propuesto.

Palabras clave. Método Simplex KKT, programación lineal, análisis de la sujeción, restricciones no atadas.

1 Introduction

Linear programming (LP) is perhaps the most important and best-studied optimization problem. Many real problems can be formulated as linear problems [4, 10, 11, 12, 13, 21, 22, 23, 27, 34, 35]. LP of grasp analysis we will deal with in this paper is an example [11, 23]. In this kind of problems, the analytic center of the polyhedron of feasible solutions is known. Therefore, for real-time grasp analysis, two important challenges are addressed: to solve an LP problem by starting at an interior point and to solve it in a computational time as small as possible. Ding [11] reported 238 milliseconds to solve an LP problem with 400 constraints using the simplex method, and Roa and Suárez [32] reported about 687 milliseconds to solve the grasp analysis which involves 32 constraints by a geometrical approach.

The problem of starting at an interior point has been solved by simplex methods using two strategies: (i) two simplex phases [3, 24, 28] and (ii) the transformation of free variables into nonnegative variables [4]. The first strategy has the disadvantage of solving two LP problems. The second increases the original size of the LP problem. The criss-cross method is worth to mention here because it is also a one-phase simplex method that can start at any, not necessarily infeasible, basis solution [13, 14, 18, 20, 36, 37, 45]. This method is useless if the starting point is an interior point of the polyhedron.

The problem of efficiency has been addressed by reducing the number of simplex steps. One approach consists in bringing the initial solution closer to the optimal solution [1, 3, 9, 16, 19, 40]. An outstanding instance of this approach is the work of Andersen *et al*. [3], which combines an interior-point method and two simplex phases to solve LP problems in standard form. The second approach simplifies a given LP problem by suppressing superfluous constraints [7, 15, 21, 29, 30, 38, 39]. A drawback of these works is that they are computationally expensive.

The main purposes of this paper are (i) to design a one-phase simplex method by dealing with free variables directly while choosing the entering and leaving variables, (ii) to reduce the size of a given LP problem by means of a method of identification of nonbinding constraints that preserves the Karush-Kuhn-Tucker (KKT) cone, and (iii) to lower the number of simplex steps by a pricing strategy based on angular measures to choose the entering variable.

The paper is organized as follows: in Section 2 we state the problem to be solved, in Section 3 we develop the *KKT Simplex method*, in Section 4 the performance of the proposed method is tested, in Section 5 conclusions are presented.

2 Problem Statement

The notation described below will be used in this paper. There may be some alterations where appropriated. Upper-case letters will be used to represent matrices. Vectors are regarded as column vectors and will be denoted by boldface characters. Lower indexes $x_1, \dots, x_n$ denote the different components of the vector **x**. The superscript T denotes transpose operation. B^{-1} denotes the inverse of a matrix B. The identity matrix is denoted by I, $\|\mathbf{x}\|$ is the Euclidian norm of a vector **x**.

2.1 Linear Programming Problem for Grasp Analysis

The LP problem with free variables to deal with in this paper is stated as follows:

$$\underset{\mathbf{x}}{\text{maximize}}\ f=\mathbf{c}^T\mathbf{x} \tag{1}$$

$$\text{subject to: } A\mathbf{x}+\mathbf{s}=\mathbf{b}$$

$$L_{Ni} < x_i < U_{Ni},\ i=1, \dots n$$

$$L_{Bj} \le s_j < U_{Bj},\ j=1, 2, \dots m,$$

where $L_{Ni}=-\infty$, $U_{Ni}=\infty$, $L_{Bj}=0$, $U_{Bj}=\infty$, $\mathbf{c},\mathbf{x}\in R^n$, $\mathbf{s},\mathbf{b}\in R^m$, $A=[\mathbf{a}_1,\cdots,\mathbf{a}_m]^T\in R^{m\times n}$, $s_j=(b_j - \mathbf{a}_j^T\mathbf{x}) \ge 0$ is a slack variable, $f:R^n\to R$ is the linear objective function, $h_j:R^n\to R$ with $h_j(\mathbf{x})=\mathbf{a}_j^T\mathbf{x} - b_j \le 0$, $j=1, 2, \dots m$ are linear constraints functions, and the gradient of the j-th constraint function is the constant vector $\nabla h_j(\mathbf{x})=\mathbf{a}_j$. We will assume $\|\mathbf{a}_j\|=1$ and $\|\mathbf{c}\|=1$ for convenience.

The free optimization variable **x** is feasible for (1) if $s_j(\mathbf{x})\ge 0$ $j=1\cdots m$. The polyhedron of feasible solutions of the primal problem, denoted by P, is defined as:

$$\mathcal{P}=\{\mathbf{x}\in R^n \mid A\mathbf{x}\le\mathbf{b}\}. \tag{2}$$

For LP problem (1) we will consider the following assumptions:

1. The polyhedron $\mathcal{P}$ is full-dimensional.
2. The polyhedron $\mathcal{P}$ is bounded towards the optimization direction.
3. The number of constraints m is greater than the number n of optimization variables.
4. The origin of the R^n space is at the analytic center of $\mathcal{P}$, which implies the optimization variables are free and $b_j>0$, j=1, …, m.
5. rank[A,**I**]=m.

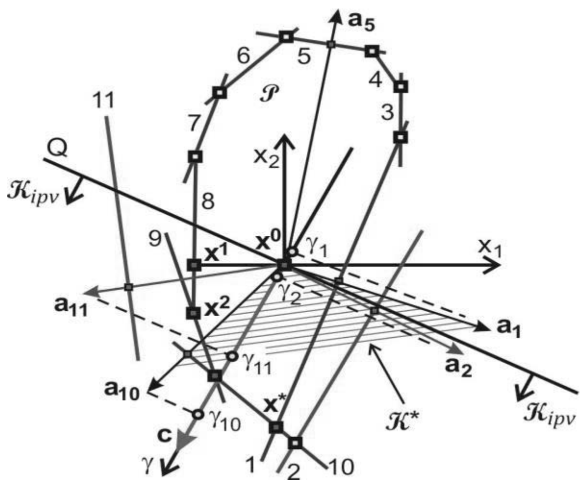


Fig. 1. Orthogonal projection of gradients $\mathbf{a}_j$,j=1,2,10,11 onto **c**

Assumptions 3 and 4 are critical for LP problems of grasp analysis. In a more general problem, where assumption 4 is not satisfied, the analytic center of $\mathcal{P}$ should be computed from an interior or exterior point of $\mathcal{P}$ by maximizing the logarithmic barrier function $\phi(\mathbf{s})=\sum_{j=1}^{m}\log(s_j)$ in the way shown in [43]. In grasp analysis, however, $m \gg n$ and the analytic center is given as part of the LP problem [11, 23]. Notice that problem (1) is neither in standard nor in canonical form, but it is adequate for treating the free variables directly avoiding their conversion to nonnegative variables.

The dictionary for (1) can be written as

$$\begin{aligned} \mathbf{x}_B &= B^{-1}\mathbf{b} - B^{-1}N\mathbf{x}_N, \\ f &= \mathbf{c}_B^T B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{y}^T N)\mathbf{x}_N, \end{aligned} \tag{3}$$

where $\mathbf{x}_N$ is a vector of non-basic variables x_{Ni},i=1, …, n, $\mathbf{x}_B$ is a vector of basic variables x_{Bj},j=1, …, m, N=[$\mathbf{N}_1$, …, $\mathbf{N}_n$] is the non-basic m×n matrix, B=[$\mathbf{B}_1$, …, $\mathbf{B}_m$] is the basic m×m matrix, and $\mathbf{y}^T = \mathbf{c}_B^T B^{-1}$. The pair {$\mathbf{x}_N$,$\mathbf{x}_B$} with $\mathbf{x}_N = \mathbf{0}$ is called a basic feasible solution (BFS). In any BFS x_{Bj}, j=1, …, m are in general real numbers in case the optimization variables are free. However, the slack variables must remain nonnegative. If all components of $\mathbf{x}_N$ are slack variables, the BFS is a vertex of the polyhedron $\mathcal{P}$. It may happen in dealing with free variables that all or some components of $\mathbf{x}_N$ are optimization variables; in such a case the BFS is not a vertex, it is just a feasible point.

2.2 Test Problem

In order to illustrate the methodology that will be presented in this paper, we will refer to the following test LP problem:

$$\underset{\mathbf{x}}{\text{maximize}}\ f(\mathbf{x}) = [-0.4472 \quad -0.8944] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

subject to:

$$\begin{array}{rcrcrcl} 0.9359x_1 & - & 0.3523x_2 & + & s_1 & = & 0.7707 \\ 0.8869x_1 & - & 0.4619x_2 & + & s_2 & = & 1.3858 \\ 0.9998x_1 & + & 0.0202x_2 & + & s_3 & = & 1.5148 \\ 0.8601x_1 & + & 0.5101x_2 & + & s_4 & = & 2.7004 \\ 0.1709x_1 & + & 0.9853x_2 & + & s_5 & = & 3.4183 \\ -0.6816x_1 & + & 0.7317x_2 & + & s_6 & = & 2.5059 \\ -0.9404x_1 & + & 0.3401x_2 & + & s_7 & = & 1.7007 \\ -0.9999x_1 & + & 0.0101x_2 & + & s_8 & = & 1.2121 \\ -0.9573x_1 & - & 0.2892x_2 & + & s_9 & = & 1.3960 \\ -0.6816x_1 & - & 0.7317x_2 & + & s_{10} & = & 1.9045 \\ -0.9939x_1 & - & 0.1104x_2 & + & s_{11} & = & 2.1082 \end{array} \tag{4}$$

$$-\infty < x_i < +\infty,\ i=1, 2,$$
$$0 \le s_j < +\infty,\ j=1, 2, \ldots, 11.$$

The polyhedron $\mathcal{P}$ of the test problem is shown in Fig. 1. Constraints from 1 to 11 are shown as lines. An initial vertex $\mathbf{x}^0$ and the optimal vertices **x*** are also shown in the figure. The gradient **c** of the objective function, the gradients of constraints $\mathbf{a}_j$, j=1,2,5,10,11 and a cone $\mathcal{K}^*$ delimited by $\mathbf{a}_1$ and $\mathbf{a}_{10}$ are also shown there. The entities γ_1, γ_2, γ_{10}, γ_{11}, Q, and $\mathcal{K}_{ipv}$ will be described in the next section.

The following definitions classify sets of constraints that will be used in this paper.

Definition 1. Let $\mathcal{M}$ be the set of constraints in a given LP problem. For the test problem we have $\mathcal{M}$={1,2,3,4,5,6,7,8,9,10,11}.

Definition 2. Let the *set of binding constraints*, denoted by $\mathcal{N}^*_{bind}$, be the set of constraints whose slack variables are zero at the optimal vertex **x***: $\mathcal{N}^*_{bind}$={k|s_k(**x***)=0}. $\mathcal{N}^*_{bind}$={1,10} for the test problem.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

Definition 3. Let the *Karush-Kuhn-Tucker cone* (*KKT cone*), denoted by $\mathcal{K}^*$, be the set of points determined by a positive linear combination of gradients $\nabla h_i(\mathbf{x^*})$ of constraints $i \in \mathcal{N}^*_{bind}$:

$$\mathcal{K}^*=\{\mathbf{x}\in R^n \mid \mathbf{x}=\sum_{i\in\mathcal{N}^*_{bind}} \lambda_i \mathbf{a_i}, \lambda_i \geq 0\}. \quad (5)$$

In Fig. 1 we have the *KKT cone:* $\mathcal{K}^*=\{\mathbf{x}\in R^2 \mid \mathbf{x}=\lambda_1\mathbf{a_1}+\lambda_{10}\mathbf{a_{10}}, \lambda_1, \lambda_{10}\geq 0\}$, which is characterized by constraints in $\mathcal{N}^*_{bind}=\{1,10\}$. The definition (5) of the *KKT cone* is derived from the concept of the *Karush-Kuhn-Tucker optimality conditions* presented in [4].

3 The KKT Simplex Method

In this section we propose a method, which we name as the *KKT Simplex method*, to efficiently solve LP problems for grasp analysis. As it will be shown here, this method will be the result of the incorporation of three different methods into the revised simplex method (RSM) [8]: (i) a method of identification of nonbinding constraints, (ii) a method to processes free variables directly while choosing the entering and leaving variables, and (iii) an *angular pricing strategy* to choose the entering variable.

3.1 Method of Identification of Nonbinding Constraints

Computing the solution of a LP problem in the R^n space can be reduced to find a set of "n" linearly independent hyperplanes such that their intersection point is feasible and optimal [6]. The corresponding "n" constraints are binding constraints, which are known after an optimal solution has been computed. The rest of constraints are superfluous. Constraints can be superfluous in two ways: nonbinding constraints which, although they delimit the polyhedron of feasible solutions, are over fulfilled at the optimal solution, and redundant constraints which do not delimit the polyhedron [38].

Since nonbinding constraints can be identified after a problem has been solved, a question that arises is whether there is a way to force them to be manifested as nonbinding before the optimal solution is reached. In this subsection, we show that the nonbinding condition may be identified by means of angular information of constraints before the simplex algorithm starts. In formulating a LP problem for grasp analysis of robotic hands [11, 23], there appear a lot of nonbinding constraints at the optimal solution. Many of these constraints may be suppressed, then the dimension of the problem could be decreased and the computational effort reduced.

Given the set $\mathcal{M}$ of constraints of a LP problem, we will identify a set of candidates of binding constraints of reduced cardinality by using angular measures of them with respect to the objective.

3.1.1 Angular Coordinates

Let γ_j be a one-dimensional coordinate on the real axis γ corresponding to the orthogonal projection of ∇h_j onto γ. Fig. 1 shows γ_j, j=1, 2, 10, 11. Based on this definition, the following theorem is obtained easily.

Theorem 1. The coordinate γ_j of the orthogonal projection of $\nabla h_j=\mathbf{a_j}$ onto γ is

$$\gamma_j=(\mathbf{a_j^T c}),\ j=1,\cdots m. \quad (6)$$

Proof: Since γ is spanned by $\mathbf{c}$, the orthogonal projector onto γ is the following $n \times n$ idempotent and symmetric matrix [31]:

$$P_\mathbf{c}=\mathbf{c}(\mathbf{c^T c})^{-1}\mathbf{c^T}. \quad (7)$$

Therefore, the orthogonal projection of $\mathbf{a_j}$ onto the axis γ can be computed as follows:

$$P_\mathbf{c}\mathbf{a_j}=(\mathbf{c}(\mathbf{c^T c})^{-1}\mathbf{c^T})\mathbf{a_j}. \quad (8)$$

After some algebraic manipulation we have:

$$P_\mathbf{c}\mathbf{a_j}=(\mathbf{a_j^T c})\mathbf{c}. \quad (9)$$

From (9) the result (6) follows. ■

The coordinate γ_j will be named as *angular coordinate* because it is an angular measure of the gradient $\mathbf{a_j}$ of the constraint $j\in\mathcal{M}$ with respect to the gradient $\mathbf{c}$ of the objective function. Since $||\mathbf{a_j}||=1$ and $||\mathbf{c}||=1$, γ_j has the property:

$\gamma_j=\cos(\theta_j)\in[-1, 1]$ where θ_j is the angle between $\mathbf{a_j}$ and $\mathbf{c}$.

3.1.2 Improvement Cone

Taking advantage of *angular coordinates*, we will introduce the concept of *improvement cone*, denoted as $\mathcal{K}_{ipv}$. The improvement cone is referred to a hyperplane in R^n orthogonal to the axis γ, illustrated as the line Q in the R^2 space in figure 1, and specified by vectors $\mathbf{v}$ that improve the value of the objective function:

$$\mathcal{K}_{ipv}=\{\mathbf{v}\in R^n|\mathbf{v}^T\mathbf{c}>0\}. \tag{10}$$

As it can be seen from γ_j given in (6), $\mathbf{a_j}$ belongs to $\mathcal{K}_{ipv}$ if its *angular coordinates* $\gamma_j>0$. That is, a constraint $j\in\mathcal{M}$ such that $\mathbf{a_j}\in\mathcal{K}_{ipv}$, or equivalently $\gamma_j>0$, is a candidate constraint to determine the optimal solution.

Notice that $\mathcal{K}_{ipv}$ may be specialized to the finite set of vectors $\mathbf{a_j}$, $j\in\mathcal{M}$. Even better, instead of grouping vectors, we may group constraints $j\in\mathcal{M}$ using the criterion $\gamma_j>0$ to isolate a set of candidates of binding constraints. The following partition of the set $\mathcal{M}$ into a set $\mathcal{N}_\gamma$ of candidates of binding constraints and $\mathcal{F}_\gamma$ of candidates of superfluous constraints

$$\mathcal{N}_\gamma=\{j\in\mathcal{M}|\gamma_j>0\}, \quad \mathcal{F}_\gamma=\{j\in\mathcal{M}|\gamma_j\leq 0\} \tag{11}$$

may be a good selection if $\mathcal{N}^*_{bind}\subseteq\mathcal{N}_\gamma$ & $\mathcal{N}^*_{bind}\not\subset\mathcal{F}_\gamma$. However, for the test problem with $\mathcal{N}^*_{bind}=\{1,10\}$, we have: $\gamma_1=-0.10$ and $\gamma_{10}=+0.96$. Since constraint 1 belongs to $\mathcal{N}^*_{bind}$, the fact $\gamma_1<0$ excludes constraint 1 from the set $\mathcal{N}_\gamma$. Therefore, the criterion $\gamma_j>0$ that leads to the partition (11) is not useful to identify $\mathcal{N}^*_{bind}$ as a subset of $\mathcal{N}_\gamma$. In other words, the *improvement cone* $\mathcal{K}_{ipv}$ does not always contain the *KKT cone* $\mathcal{K}^*$.

3.1.3 Identification of Nonbinding Constraints

The *KKT cone* $\mathcal{K}^*$ is not always a subset of the *improvement cone* $\mathcal{K}_{ipv}$. Equivalently, the set $\mathcal{N}^*_{bind}$ is not always a subset of $\mathcal{N}_\gamma$. If we could find a set $\mathcal{N}_\gamma$ that satisfies $\mathcal{N}^*_{bind}\subseteq\mathcal{N}_\gamma$, we would immediately identify $\mathcal{F}_\gamma$ as a set of superfluous constraints.

Since constraints $j\in\mathcal{N}^*_{bind}$ correspond to $\mathbf{a_j}$ whose hyperplane delimit the *KKT cone* $\mathcal{K}^*$, we name the property $\mathcal{N}^*_{bind}\subseteq\mathcal{N}_\gamma$ as the *KKT binding condition* and specify it as follows:

$$\mathcal{N}^*_{bind}\subseteq\mathcal{N}_\gamma \;\&\; \mathcal{N}^*_{bind}\not\subset\mathcal{F}_\gamma \tag{12}$$

Notice that (12) is satisfied when $\mathcal{N}_\gamma=\mathcal{M}$. However, a set $\mathcal{N}_\gamma$ of reduced cardinality is preferred. Therefore, for our purpose $\mathcal{N}_\gamma=\mathcal{M}$ is the worst case while $\mathcal{N}_\gamma=\mathcal{N}^*_{bind}$ is the best case.

Let us consider the *angular coordinate* $\gamma_1<0$ that excludes constraint 1 from the set: $\mathcal{N}_\gamma=\{j\in\mathcal{M}|\gamma_j>\gamma_{TH}\}$ for a threshold coordinate $\gamma_{TH}=0$. From the axis γ located at the right side of figure 2 below we have:

$$\mathcal{N}_\gamma=\{\mathbf{10},9,11,8,7,2\}, \quad \mathcal{F}_\gamma=\{\mathbf{1},6,3,4,5\}. \tag{13}$$

Since $\mathcal{N}^*_{bind}=\{\mathbf{1},\mathbf{10}\}$ and $\boldsymbol{\gamma_1}<0$, we find that $1\notin\mathcal{N}_\gamma$, that is, $\mathcal{N}^*_{bind}\not\subset\mathcal{N}_\gamma$. However, to make $\mathcal{N}^*_{bind}\subseteq\mathcal{N}_\gamma$ all we have to do is to move $\gamma_{TH}=0$ to a negative value.

Since $\gamma_1=-0.10$, we may propose $\gamma_{TH}=-0.2$.

Proposition 1: From the foregoing discussion, a *method of identification of nonbinding constraints* is proposed which consists in partitioning $\mathcal{M}$ into the set $\mathcal{N}_\gamma$ of candidates of binding constraints and $\mathcal{F}_\gamma$ of candidates of superfluous constraints:

$$\mathcal{N}_\gamma=\{j\in\mathcal{M}|\gamma_j\geq\gamma_{TH}\}, \quad \mathcal{F}_\gamma=\{j\in\mathcal{M}|\gamma_j<\gamma_{TH}\}, \tag{14}$$

where the threshold coordinate γ_{TH} must be chosen such that $\mathcal{N}_\gamma$ and $\mathcal{F}_\gamma$ satisfy the *KKT binding condition.* If so, $\mathcal{F}_\gamma$ may be discarded and $\mathcal{N}_\gamma$ recast as $\mathcal{M}$. At this point, the *identification of nonbinding constraints* ends and the *KKT Simplex method* starts solving the given LP problem by processing the reduced set $\mathcal{M}$.

3.1.4 Tuning Threshold Coordinate

As an illustration for proposing the adequate threshold coordinate $\gamma_{TH}\in(1, -1]$ we will identify $\mathcal{N}_\gamma$

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

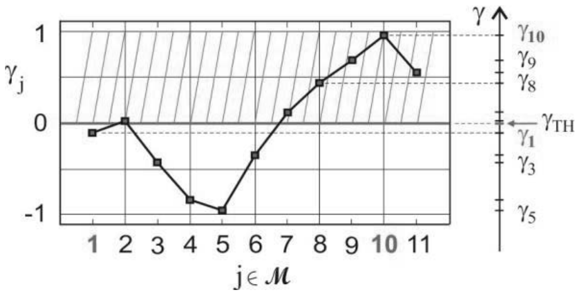


Fig. 2. Distribution of angular coordinates

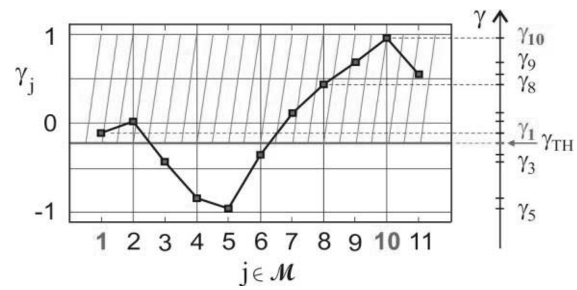


Fig. 3. Identification of $\mathcal{N}_\gamma$ with γ_{TH}=-0.2

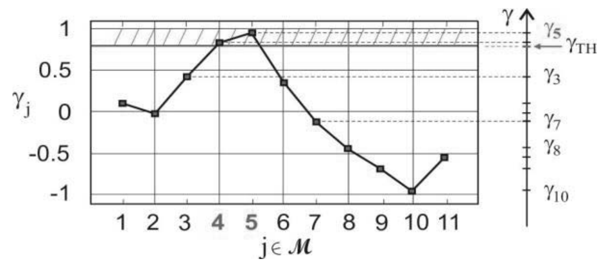


Fig. 4. A positive threshold coordinate γ_{TH}=+0.8

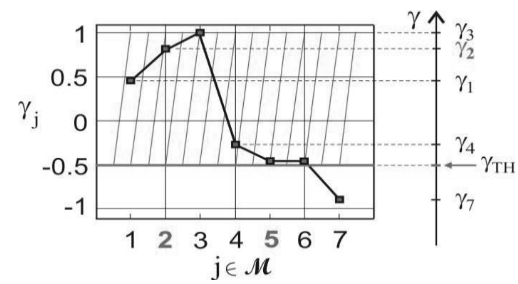


Fig. 5. A negative threshold coordinate γ_{TH}=-0.5

in three test problems with: γ_{TH}=-0.2, +0.8, -0.5. The graphs of figures 3, 4, and 5 show constraints in the dashed areas that belong to $\mathcal{N}_\gamma$, whose coordinates satisfy $\gamma_j \geq \gamma_{TH}$.

Since $\mathcal{N}^*_{bind}$={**1,10** }, the coordinates above the line γ_{TH}=-0.2 belong now to $\mathcal{N}_\gamma$ and the rest to $\mathcal{F}_\gamma$: $\mathcal{N}_\gamma$={**10**,9,11,8,7,2,**1**}, $\mathcal{F}_\gamma$={6,3,4,5}.

A second test problem is obtained from the first one by changing the sign of the given gradient of the objective function: **c**=$[\,0.4472 \quad 0.8944]^T$. Since the set of binding constraints for this problem is $\mathcal{N}^*_{bind}$={**4,5** }, we may propose γ_{TH}=+0.8 as it is shown in figure 4. In this case we have: $\mathcal{N}_\gamma = \mathcal{N}^*_{bind}$.

A third test problem has **c**=$[-0.4472 \quad -0.8944]^T$ and the set of constraints adapted from [19]:

$$\begin{array}{rcrcrcl}
1.0000x_1 & + & 0.0000x_2 & + & s_1 & = & 0.5918 \\
0.8944x_1 & + & 0.4472x_2 & + & s_2 & = & 0.9044 \\
0.4472x_1 & + & 0.8944x_2 & + & s_3 & = & 2.8037 \\
-0.9806x_1 & + & 0.1961x_2 & + & s_4 & = & 4.0949 \\
-1.0000x_1 & + & 0.0000x_2 & + & s_5 & = & 0.4082 \\
-1.0000x_1 & + & 0.0000x_2 & + & s_6 & = & 5.4082 \\
0.0000x_1 & - & 1.0000x_2 & + & s_7 & = & 1.1613
\end{array} \qquad (15)$$

Since the set of binding constraints for this problem is $\mathcal{N}^*_{bind}$={**2,5** }, we ought to propose γ_{TH}=-0.5 as it is shown in figure 5.

The test problem with positive threshold coordinate γ_{TH}=+0.8 of figure 4 exhibits the best reduction of $\mathcal{N}_\gamma = \mathcal{N}^*_{bind}$={**4,5** }. In contrast, the test problem with the negative threshold coordinate γ_{TH}=-0.5 of figure 5 exhibits the worst reduction $\mathcal{N}_\gamma$={1,**2**,3,4,**5**,6} because γ_{TH} is near to the case: γ_{TH}=-1. Since γ_{TH}=-0.5 works for the best and a worst case, it may be used as a conservative value for a general LP problem.

3.2 Entering Variable Selection

As it will be shown in this subsection, the direct processing of free optimization variables modifies the way the entering and leaving variables are chosen. The new methods to determine these variables will endow the one-phase property to the *KKT Simplex method*.

While moving from a vertex to an adjacent vertex, the free variables x_i, i=1,⋯,n are distributed among the components of the vectors of non-basic variables $\mathbf{x_N}$ and basic variables $\mathbf{x_B}$.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

Recall that an entering variable is chosen from $\mathbf{x_N}$ and the leaving variable from $\mathbf{x_B}$. Here we will deal with the selection of an entering variable which consists of two steps. The first step determines a set of non-basic variables that are candidates to enter the basis. The second step chooses just one entering variable.

3.2.1 Set of Candidates of Entering Variables

Let us consider the equation for the objective function in terms of non-basic variables:

$$f=\mathbf{c_B^T}B^{-1}\mathbf{b}+(\mathbf{c_N^T}\text{-}\mathbf{y^T}N)\mathbf{x_N} \ . \quad (16)$$

The criterion to select a candidate to be an entering variable is to choose one entry x_{Nk} of $\mathbf{x_N}$ whose cost coefficient increases the value of the objective function. In order to refer to non-basic, optimization, and slacks variables, we will use sets of their indices, denoted by Ω_N, Ω_o, and Ω_s respectively, which are defined as follows:

$$\begin{gathered}\Omega_N=\{k_1,\cdots,k_n\},\\ \Omega_o=\{1,\cdots,n\},\\ \Omega_s=\{1,\cdots,m\}.\end{gathered} \quad (17)$$

A non-basic variable $x_{Nk}, k\in\Omega_N$ may be an optimization variable $x_i, i\in\Omega_o$ or a slack variable $s_j, j\in\Omega_s$. Since x_i is free, x_{Nk} maybe positive or negative. In other words, the current zero value of x_{Nk} may change to a negative or positive value:

$$x_{Nk}=x_{Nk}\pm\xi, \quad (18)$$

where the number $\xi>0$ will be computed in subsection 3.3. Let us search (18) for its effect on the objective function. Replacing x_{Nk} into (16) the objective function is updated as follows:

$$f=f\pm(c_{Nk}\text{-}\mathbf{y^T N_k})\xi, \quad (19)$$

where $\mathbf{N_k}$ is the k-th column of the non-basic matrix N. Since $\xi>0$ and due to the presence of the double sign ±, the value of the objective function may be increased by the positive or negative signs of the cost coefficient $(c_{Nk}\text{-}\mathbf{y^T N_k})$.

Let us consider $f=f+(c_{Nk}\text{-}\mathbf{y^T N_k})\xi$ which may be increased if we can find a positive coefficient:

$$(c_{Nk}\text{-}\mathbf{y^T N_k})>0,\ k\in\Omega_N \ . \quad (20)$$

If such a k-th coefficient exists, we record our finding by the sign indicator $k^+=k$, otherwise $k^+=0$ as it is shown in the second and third rows of table 1 below. Since $x_{Nk}=x_{Nk}+\xi$ has been considered, x_{Nk} may be an optimization variable x_i or a slack variable s_j as it is shown in the first column of table 1 below.

Now let us consider $f=f\text{-}(c_{Nk}\text{-}\mathbf{y^T N_k})\xi$ which is a consequence of taking $x_{Nk}=x_{Nk}\text{-}\xi$. This negative value means that x_{Nk} must be just an optimization variable x_i, since a slack variable cannot be negative. The fact x_{Nk} is an optimization variable may be detected by its lower bound $L_{Nk}=-\infty$ as it is shown in the second column of table 1 below. Therefore, $f=f\text{-}(c_{Nk}\text{-}\mathbf{y^T N_k})\xi$ may be increased by finding a negative k-th coefficient:

$$(c_{Nk}\text{-}\mathbf{y^T N_k})<0,\ k\in\Omega_N \ \&\ L_{Nk}=-\infty \ . \quad (21)$$

If such a coefficient exists, we record our finding by the sign indicator $k^-=k$, otherwise $k^-=0$ as it is shown in the fourth row of table 1 below. The table 1 summarizes the way the sign of the k-th cost coefficient $(c_{Nk}\text{-}\mathbf{y^T N_k})$ determines the index k of an entering variable x_{Nk} that increases the value of the objective function.

Table 1. Selection of an entering variable

x_{Nk}	L_{Nk}	The k-th coefficient	Indices of x_{Nk} k^+	k^-
s_j	0	$(c_{Nk}\text{-}\mathbf{y^T N_k})>0$	k	0
x_i	$-\infty$	$(c_{Nk}\text{-}\mathbf{y^T N_k})>0$	k	0
x_i	$-\infty$	$(c_{Nk}\text{-}\mathbf{y^T N_k})<0$	0	k

In a LP problem with nonnegative variables, the detection of positive coefficients is enough. With free variables, however, it is critical to detect the negative coefficients because these may also improve the value of the objective function.

Based on the foregoing analysis, the following proposition is stated.

Proposition 2: The non-basic variables $x_{Nk}, k\in\Omega_N$ that are candidates to enter the basis are identified in two sets Ω^+ and Ω^-, which are determined from table 1:

$$\begin{gathered}\Omega^+=\{k\in\Omega_N \mid (c_{Nk}\text{-}\mathbf{y^T N_k})>0\},\\ \Omega^-=\{k\in\Omega_N \mid (c_{Nk}\text{-}\mathbf{y^T N_k})<0 \ \&\ L_{Nk}=-\infty\}.\end{gathered} \quad (22)$$

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

3.2.2 Angular Pricing Strategy

In this subsection we propose a pricing strategy based on *angular coordinates* for the *KKT Simplex method* to choose one entering variable from the sets Ω^+ and Ω^- defined in (22).

Some pricing strategies have been suggested [10, 17, 44]. One of the most widely used is the Dantzig's rule [10], where the entering variable is an element of the set $\Omega^+ \cup \Omega^-$ which corresponds to the maximum cost coefficient among those that improve the value of the objective function. The basic idea of a pricing strategy based on an angular criterion has been developed for LP problems with nonnegative variables [44].

Here, we develop a pricing strategy in the context of free variables, which we name as *angular pricing strategy.* The strategy will use the *angular coordinates* $\gamma_j, j \in \Omega_s$ introduced in subsection 3.1.1 and angular coordinates of the Cartesian axis which will be introduced in this subsection.

Assume that the origin $\mathbf{x^0}$=0 of the R^n space is at the analytic center of the polyhedron $\mathscr{P}$. The *KKT Simplex method* should start at $\mathbf{x^0}$, migrate to a feasible vertex $\mathbf{x^\#}$ on the boundary of the polyhedron $\mathscr{P}$, and proceed until the optimal vertex $\mathbf{x^*}$ is reached. Therefore, the non-basic vector $\mathbf{x_N}$ contains n optimization variables $x_i, i \in \Omega_o$ at $\mathbf{x^0}$, n-1 optimization variables at $\mathbf{x^1}$, … and zero optimization variables at $\mathbf{x^\#}$. However, $\mathbf{x_N}$ contains just slack variables $s_j, j \in \Omega_s$ from $\mathbf{x^\#}$ to $\mathbf{x^*}$. Since the *angular coordinate* γ_j is defined for each constraint $j \in \Omega_s$, it must be associated to a vertex from $\mathbf{x^\#}$ to $\mathbf{x^*}$.

Let us consider $\mathcal{N}_\gamma=\{j \in \mathcal{M} | \gamma_j \geq \gamma_{TH}\}$ that contains the set of binding constraints $\mathcal{N}^*_{bind}$. Since $\gamma_j=\cos(\theta_j) \in [-1, 1]$, the fact $\gamma_j \to 1$ when $\theta_j \to 0$, for some j chosen from $\mathcal{M}$, means the j-th constraint may determine the optimal vertex $\mathbf{x^*}$ when $\gamma_j \to 1$. That is, coordinates with the property $\gamma_j \to 1$ may be used to choose an entering variable that better improve the value of the objective function at vertices from $\mathbf{x^\#}$ to $\mathbf{x^*}$.

For the migration from x^0 to $x^\#$ there is not yet any angular measure. We propose the direction cosines of the gradient **c** of the objective function as an angular measure of each Cartesian axis of $x_i, i \in \Omega_o$ with respect to **c**. Let us name this angular measure as *Cartesian angular coordinate*, denoted as γ_i^c and defined as the absolute value of the direction cosines of **c**:

$$\gamma_i^c=|c_i|, i \in \Omega_o. \tag{23}$$

Notice that $\gamma_i^c \in [0, 1]$. The usefulness of γ_i^c relays on the fact that **c** is directed toward a region of $\mathscr{P}$ where the optimal vertex $\mathbf{x^*}$ is located. That is, coordinates with the property $\gamma_i^c \to 1$ may be used to choose an entering variable that better improve the value of the objective function. Since $i \in \Omega_o$, γ_i^c may be used just when x_i is a member of the non-basic vector $\mathbf{x_N}$. That is, γ_i^c may be used during the migration from x^0 to $x^\#$.

As it is stated in proposition 2, x_{Nk} is a candidate to enter the basis if $k \in (\Omega^+ \cup \Omega^-)$. Notice that for each candidate x_{Nk} there must be an *angular Cartesian coordinate* $\gamma_k^c, k \in \Omega_o$ or an *angular coordinate* $\gamma_k, k \in \Omega_s$. Since both angular coordinates will be used to choose one entering variable, they must be associated to Ω^+ and Ω^- as follows:

If $k \in (\Omega_o \cap \Omega^+)$ we may group γ_k^c in the set:

$$\Omega^+_{\gamma c}=\{\gamma_k^c | k \in (\Omega_o \cap \Omega^+)\}. \tag{24}$$

If $k \in (\Omega_o \cap \Omega^-)$ we may group γ_k^c in the set:

$$\Omega^-_{\gamma c}=\{\gamma_k^c \mid k \in (\Omega_o \cap \Omega^-)\}. \tag{25}$$

If $k \in (\Omega_s \cap \Omega^+)$ we may group γ_k in the set:

$$\Omega^+_{\gamma}=\{\gamma_k | k \in (\Omega_s \cap \Omega^+)\}. \tag{26}$$

Since $\gamma_k \to 1$ and $\gamma_k^c \to 1$ means that the best improvement of the value of the objective function may be obtained, we propose the following strategy to choose the entering variable.

Proposition 3: For $\mathbf{x^0} \to \mathbf{x^{n-1}}$ use γ_k^c to determine the entering variable indicated by k^+ or k^-:

$k^+=k \in \Omega^+ | \gamma_k^+= \max_k \Omega^+_{\gamma c}$,

$k^-=k \in \Omega^- | \gamma_k^-= \max_k \Omega^-_{\gamma c}$,

if $\gamma_k^+ > \gamma_k^-$, k^-=0 otherwise k^+=0.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

For $\mathbf{x^\#} \rightarrow \mathbf{x^*}$ use γ_k to determine the entering variable indicated by k^+:

$k^+ = k \in \Omega^+ \mid \gamma_k^+ = \max_k \Omega_\gamma^+$.

Based on proposition 3, we propose the *angular pricing strategy* presented in Algorithm 1 to determine the entering variable x_{Nk}, whose index "k" is k^+ or k^-. The strategy is implemented in two parts for an iteration of the *KKT Simplex method*. The first part, named the *Cartesian pricing strategy*, chooses one entering variable at each feasible solution from $\mathbf{x^0}$ to $\mathbf{x^{n-1}}$ using γ_k^c. The second part, named the *constraint pricing strategy*, chooses one entering variable at each basic feasible solution from $\mathbf{x^\#}$ to $\mathbf{x^*}$ using γ_k.

3.3 Leaving Variable Selection

Assume that the entering variable x_{Nk} has been selected. Then the current zero value of x_{Nk} may change to a negative or positive value: $x_{Nk}=x_{Nk}\pm\xi$. The positive number $\xi>0$ will be computed here.

Let us consider the system of equations given in the dictionary (3) at a simplex iteration "t":

$$\mathbf{x_B}=B^{-1}\mathbf{b}-B^{-1}N\mathbf{x_N}\,. \qquad (27)$$

When the new value $x_{Nk}=x_{Nk}\pm\xi$ is substituted in k-th element of $\mathbf{x_N}$, the vector of basic variables $\mathbf{x_B}$ is updated as follows:

$$\mathbf{x_B}=\mathbf{x_B}\mp B^{-1}\mathbf{N_k}\xi\,. \qquad (28)$$

To preserve its feasibility, $\mathbf{x_B}$ should let one of its components x_{Br} become zero, which is called the *leaving variable*. In this way, the entering variable $x_{Nk}=x_{Nk}\pm\xi$ becomes nonzero and one leaving variable x_{Br} becomes zero.

The number ξ must be chosen such that $\mathbf{x_B}$ given in (28) and the *entering variable* $x_{Nk}=x_{Nk}\pm\xi$ remain feasible, that is:

$$\begin{aligned} &L_{Nk}<(x_{Nk}\pm\xi)<U_{Nk}\,, \\ &\mathbf{L_B}\leq(\mathbf{x_B}\mp B^{-1}\mathbf{N_k}\xi)<\mathbf{U_B}\,. \end{aligned} \qquad (29)$$

Since $x_{Nk}=0$, $L_{Nk}=-\infty$, $U_{Nk}=+\infty$, and $\xi>0$, the first inequality in (30) does not impose any bound to ξ. Let us define $\mathbf{d}=B^{-1}\mathbf{N_k}=[d_1,\cdots,d_m]^T$ and express the second inequality of (29) for each element of $\mathbf{x_B}$:

Algorithm 1. The *angular pricing strategy* at an iteration "t"

Reset the indexes of the entering variable:
$k^+=0,\ k^-=0$
PART I: *The Cartesian pricing strategy*: $\mathbf{x^0} \rightarrow \mathbf{x^{n-1}}$
if t < n choose an entering variable "k" (use γ_k^c):
$\Omega^+=\phi$, $\Omega^-=\phi$, $\Omega_{\gamma c}^+=\phi$, $\Omega_{\gamma c}^-=\phi$, $\gamma_k^+=0$, $\gamma_k^-=0$
for k=1 to *n* construct: Ω^+, $\Omega_{\gamma c}^+$, Ω^-, and $\Omega_{\gamma c}^-$:
if $k\in\Omega_o$
if $(c_{Nk}-\mathbf{y^T N_k})>0$
$\Omega^+=\Omega^+\cup\{k\}$
$\Omega_{\gamma c}^+=\Omega_\gamma^+\cup\{\gamma_k^c\}$
if $(c_{Nk}-\mathbf{y^T N_k})<0$ & $L_{Nk}=-\infty$
$\Omega^-=\Omega^-\cup\{k\}$
$\Omega_{\gamma c}^-=\Omega_\gamma^-\cup\{\gamma_k^c\}$
if $\Omega_{\gamma c}^+\neq\phi$, $k^+=k\in\Omega^+\mid\gamma_k^+=\max_k\Omega_{\gamma c}^+$
if $\Omega_{\gamma c}^-\neq\phi$, $k^-=k\in\Omega^-\mid\gamma_k^-=\max_k\Omega_{\gamma c}^-$
if $\gamma_k^+>\gamma_k^-$, $k^-=0$ otherwise $k^+=0$
if $k^+\neq0$, $\Omega_o=\Omega_o\backslash\{k^+\}$
if $k^-\neq0$, $\Omega_o=\Omega_o\backslash\{k^-\}$
PART II: *The constraint pricing strategy*: $\mathbf{x^\#} \rightarrow \mathbf{x^*}$
if $t\geq n$ choose an entering variable "k" (use γ_k):
$\Omega^+=\phi$, $\Omega_\gamma^+=\phi$, $\gamma_k^+=0$
for k=1 to *n* construct: Ω^+ and Ω_γ^+:
if $k\in\Omega_s$
if $(c_{Nk}-\mathbf{y^T N_k})>0$
$\Omega^+=\Omega^+\cup\{k\}$
$\Omega_\gamma^+=\Omega_\gamma^+\cup\{\gamma_k\}$
if $\Omega_\gamma^+\neq\phi$, $k^+=k\in\Omega^+\mid\gamma_k^+=\max_k\Omega_\gamma^+$
END of the algorithm 1.

$$L_{Bj}\leq(x_{Bj}\mp d_j\xi)<U_{Bj},\ j\in\{1,\ldots m\}. \qquad (30)$$

Let us substitute in (30): $U_{Bj}=+\infty$ and

$$L_{Bj}=\begin{cases} 0, & \text{if } x_{Bj} \text{ is a slack variable } s_j, \\ -\infty, & \text{if } x_{Bj} \text{ is an optimization variable } x_i\,. \end{cases} \qquad (31)$$

Then, we get from (30) two inequalities which depend on the nature of basic variable x_{Bj} or, more precisely, on the values of its lower bound L_{Bj}:

$$\begin{aligned} &0\leq\left(x_{Bj}\mp d_j\xi\right)<+\infty \quad \text{if } L_{Bj}=0,\ (x_{Bj}=s_j)\,, \\ &-\infty<\left(x_{Bj}\mp d_j\xi\right)<+\infty \quad \text{if } L_{Bj}=-\infty,\ (x_{Bj}=x_i)\,. \end{aligned} \qquad (32)$$

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

If the non-basic variable is increased $x_{Nk}=x_{Nk}+\xi$, the basic variable is changed to $(x_{Bj}-d_j\xi)$ and the first inequality of (32) derives in the result:

$$\xi_j \in (-\infty, \frac{x_{Bj}}{d_j}],\ j=1,\cdots,m,\ \text{ if } L_{Bj}=0,\ (x_{Bj}=s_j). \quad (33)$$

Since the value of the objective function must increase with ξ by means of $x_{Nk}=x_{Nk}+\xi$, we have to choose ξ_j as large as constraint (33) allows:

$$\xi_j=+\frac{x_{Bj}}{d_j},\ j=1,\cdots,m,\ L_{Bj}=0 \text{ and } k^+=k. \quad (34)$$

The formula (34) is valid when: (i) $L_{Bj}=0$, and (ii) the non-basic variable is increased $x_{Nk}=x_{Nk}+\xi$, which is detected by $k^+=k$.

Now, if the non-basic variable is decreased: $x_{Nk}=x_{Nk}-\xi$, the basic variable is changed to $(x_{Bj}+d_j\xi)$ and again the first inequality of (32) derives now in the result:

$$\xi_j \in [-\frac{x_{Bj}}{d_j}, +\infty),\ j=1,\cdots,m,\ \text{ if } L_{Bj}=0,\ (x_{Bj}=s_j). \quad (35)$$

Since the value of the objective function increases with ξ by means of $x_{Nk}=x_{Nk}-\xi$, we have to choose ξ_j as small as constraint (35) allows:

$$\xi_j=-\frac{x_{Bj}}{d_j},\ j=1,\cdots m,\ L_{Bj}=0 \text{ and } k^-=k. \quad (36)$$

The formula (36) is valid when: (i) $L_{Bj}=0$, and (ii) the non-basic variable is decreased $x_{Nk}=x_{Nk}-\xi$, which is detected by $k^-=k$.

The second inequality in (32) derives simply into the following constraint:

$$\xi_j \in (-\infty,+\infty)\big|,\ j=1,\cdots,m,\ L_{Bj}=-\infty. \quad (37)$$

Condition (37) means that ξ_j is a real number. However, it is valid just when: $x_{Bj}=x_i$, which is detected by $L_{Bj}=-\infty$.

For a non-degenerated LP problem, numbers $\xi_j\neq 0$ are computed by formulas (34) or (36), which are characterized by the condition $L_{Bj}=0$. The rest numbers must be zero; as it is implied by constraint (37); these zero valued numbers are identified by the condition $L_{Bj}=-\infty$:

$$\xi_j=0,\ j\in\{1,\cdots,m\} \text{ and } L_{Bj}=-\infty \quad (38)$$

Summarizing formulas (34), (36) and (38) that compute the numbers ξ_j, $j=1,\cdots,m$, we have:

$$\xi_j=\begin{cases} \frac{x_{Bj}}{d_j} & L_{Bj}=0 \text{ and } k^+=k, \\ -\frac{x_{Bj}}{d_j} & L_{Bj}=0 \text{ and } k^-=k, \\ 0 & L_{Bj}=-\infty. \end{cases} \quad (39)$$

Let us name ξ_j, $j=1,\cdots,m$ given in (39) as *simplex coordinates* because they are delivered by the *KKT Simplex method* in the context of free variables. Let us notice that for each constraint $j\in\{1,\cdots,m\}$ there is just one number ξ_j.

Let us remark that the set $\{\xi_j, j=1,\cdots,m\}$ are calculated at the current vertex $\mathbf{x}^t$ at each iteration "t" of the *KKT Simplex method*. Also, as it is shown in [8, 26] the strictly positive coordinate, denoted as ξ_r, specified by:

$$\xi_r=\min_j\{\xi_j|\xi_j>0\} \quad (40)$$

defines the adjacent vertex $\mathbf{x}^{t+1}$ at which the value of the objective function is improved. Therefore, the *leaving variable* x_{Br} corresponds to ξ_r. Then the desired value for $\xi>0$ in $x_{Nk}=x_{Nk}\pm\xi$ should be ξ_r.

3.4 The KKT Simplex method

The *KKT Simplex method* is presented in Algorithm 2. Steps 1 to 5 were transcribed from the RSM as it is presented in [8] and adapted to solve LP problems for grasp analysis. The identification of nonbinding constraints is incorporated in step 0 as a pre-solution procedure [7, 39]. The *entering variable* x_{Nk} is chosen in step 2 by the *angular pricing strategy* of table 2. In step 4 the *leaving variable* x_{Br} is selected. *Step 5* updates the dictionary. Steps 1 and 3 are the same as that of the RSM in [8]. The input data may be obtained from (1) and (3).

3.5 Discussion of the KKT Simplex Method

The one-phase property of the *KKT simplex method* is derived from the direct processing of the free variables. We distinguish three different classes of starting points:

(i) Points in the interior of $\mathcal{P}$,

(ii) Points on a face of $\mathcal{P}$,

(iii) Vertices on the boundary of $\mathcal{P}$.

Therefore, the *KKT simplex method* generates in general a sequence of feasible points: $\mathbf{x^0}$, $\mathbf{x^1}$, … $\mathbf{x^\#}$, … $\mathbf{x^*}$ in just one simplex phase. If the starting point is a vertex of $\mathcal{P}$ the sequence begins at $\mathbf{x^\#}$. If the starting point is $\mathbf{x^0}$, the first vertex $\mathbf{x^\#}$ is reached in exactly n simplex steps. As a consequence, the *KKT simplex method* is able to start at any point on the polyhedron $\mathcal{P}$. Therefore, the *KKT Simplex method* is suitable to be combined with an interior-point method for solving LP problems [2, 3, 42].

The efficiency of the *KKT simplex method* is due to: (i) the reduction of the size of the LP problem and (ii) the reduction of the number of simplex steps by means of the *angular pricing strategy*. However, the method bears the same complexity as the usual simplex method. That is, the number of simplex steps taken to solve a LP problem is bounded [2] by

$$C(m)=\binom{m+n}{m}=\frac{(m+n)!}{m!\,n!}. \qquad (41)$$

However, this bound changes as a result of the reduction of the size of the LP problem from m to 0.6m, as it will be shown graphically in section 4.

4 Numerical Experiments

In this section we test the performance of the *KKT Simplex method* with respect to the RSM [8] in solving LP problems for grasp analysis. These LP problems are discussed in Appendix A. A single problem with 400 constraints and a sequence of problems of 40, 80, …,800 constraints will be solved using the same threshold coordinate γ_{TH}=-0.07. The starting point is the analytic center of $\mathcal{P}$.

In particular, we will test the efficiency and the one-phase nature of the *KKT Simplex method*.

Algorithm 2. The KKT Simplex method

Step 0. Identify the set of candidates of binding constraints $\mathcal{N}_\gamma$:

Propose the threshold coordinate: γ_{TH}=-0.07 just for LP of grasp analysis.

Compute the *angular coordinates*:

$$\gamma_j=(\mathbf{a}_j^T\mathbf{c}),\ j\in\Omega_s,\ \gamma_i^c=|c_i|,\ i\in\Omega_o$$

Partition the set $\mathcal{M}$ of constraints:

$$\mathcal{N}_\gamma=\{j\in\mathcal{M}|\gamma_j\geq\gamma_{TH}\},\ \mathcal{F}_\gamma=\{j\in\mathcal{M}|\gamma_j<\gamma_{TH}\}$$

Recast N_γ as M and proceed.

An iteration "t" of the KKT Simplex method:

Step 1. Solve the system $\mathbf{y}^T B=\mathbf{c}_B^T$.

Step 2. Choose an entering variable x_{Nk} by means of the *angular pricing strategy* of table 2. If there is no such entering variable (k^+=0 and k^-=0), then the current solution is optimal.

Step 3. Solve the system $B\mathbf{d}=\mathbf{N}_k$, $\mathbf{d}=[d_1,\cdots,d_m]^T$.

Step 4. Find the living variable x_{Br}:

Compute the *simplex coordinates* ξ_j, j=1,⋯m

If $k^+\neq0$ and k^-=0:

$$\xi_j=\begin{cases}\dfrac{x_{Bj}}{d_j} & \text{if } L_{Bj}=0\\ 0 & \text{if } L_{Bj}=-\infty\end{cases}$$

If k^+=0 and $k^-\neq0$:

$$\xi_j=\begin{cases}\dfrac{-x_{Bj}}{d_j} & \text{if } L_{Bj}=0\\ 0 & \text{if } L_{Bj}=-\infty\end{cases}$$

Find $\xi_r=\min_j\{\xi_j|\xi_j>0\}$.

If there is no such ξ_r, the problem is unbounded; otherwise, at least one component "r" of ($\mathbf{x}_B$-$\xi\mathbf{d}$) equals zero if $k^+\neq0$, or ($\mathbf{x}_B$+$\xi\mathbf{d}$) equals zero if $k^-\neq0$, and the associated variable x_{Br} is leaving the basis.

Step 5. Set the value of the *entering variable* x_{Nk} equals to +ξ_r if $k^+\neq0$, or to -ξ_r if $k^-\neq0$. Replace $\mathbf{x_B}$ by ($\mathbf{x}_B$-$\xi_r\mathbf{d}$) if $k^+\neq0$, or by ($\mathbf{x}_B$+$\xi_r\mathbf{d}$) if $k^-\neq0$. Replace the leaving column $\mathbf{B_r}$ of the basic matrix B by the entering column $\mathbf{N_k}$, and replace the *leaving variable* x_{Br} by the *entering variable* x_{Nk}.

END of the algorithm 2.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

4.1 Reduction of the Problem Size

Fig. 6 shows the graph of reduced sizes m_{KKT} *vs*. the initial size *m* of the sequence of LP problems solved by the *KKT Simplex method* with respect to the RSM. A slope of 0.6 is exhibited in the graph for the reduced sizes m_{KKT}. That is, a reduction of 40% of the size of each of these problems was reached.

Fig. 7 shows the ratio

$$\eta = \frac{C(0.6m)}{C(m)} \qquad (42)$$

of the reduced bound C(0.6m) to the original bound C(*m*) for problem sizes 40, 80, …,800. From figure 7 we can see that a size reduction from *m* to *0.6m* causes a bound reduction from C(m) to about C(0.6m)=0.05C(m) for *m*≥200.

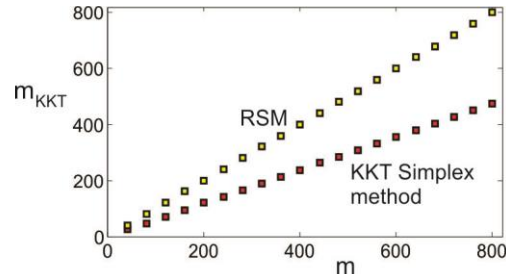


Fig. 6. Reduced sizes of a sequence of problems

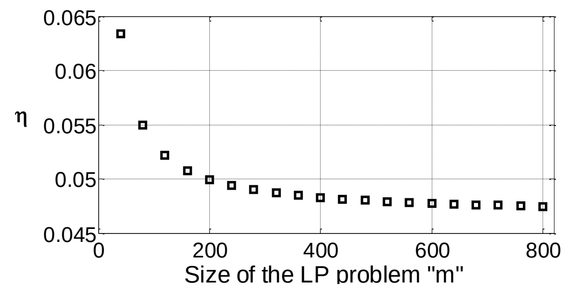


Fig. 7. Bound ratio η

4.2 Reduction of the Number of Simplex Steps

The two graphs in figure 8 show the reduction of the number of simplex steps due to the *angular pricing strategy* implemented in the *KKT Simplex method* in solving a LP problem of 400 constraints.

The optimal solution is reached in 35 steps when the *angular pricing strategy* is implemented, while 42 steps are taken when the Dantzig's rule is used.

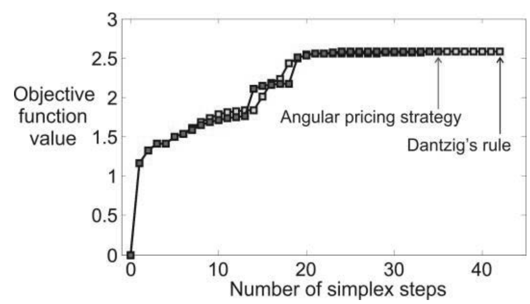


Fig. 8. Reduction of number of simplex steps

4.3 Computation Time of the Problem

A single problem: The time in milliseconds $\tau_{KKT}\cong 2$ of the *KKT Simplex method* is compared with $\tau_{RSM}\cong 9$ of the RSM spent per step in solving a LP problem of 400 constraints, as it is shown in figure 9. The identification of nonbinding constraints takes $\tau_{bind}\cong 2.5$ milliseconds as part of the above methods. That is, τ_{bind} is on the order of τ_{KKT}.

Therefore, the *KKT Simplex method* is about 4.5 times faster than the RSM in this experiment.

In 2001 Ding [11] reported 238 milliseconds for solving the same LP problem of 400 constraints by means of a Simplex method; in 2009 Roa and Suárez [32] reported about 687 milliseconds to solve the grasp analysis which involves 32 constraints by means of a convex hull based method; the *KKT Simplex method* solves a LP problem of 400 constraints in $36\tau_{KKT}\cong 72$ milliseconds. Since these results were obtained on different machines and/or methods, we just may say that the *KKT Simplex method* solves a LP problem of grasp analysis in a reasonable computational time. We must point out that there is not still a benchmark on grasp analysis for execution time comparison.

A sequence of problems: Figures 10 and 11 below show the solution time spent by the *KKT Simplex method* and the IBM ILOG CPLEX

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

optimization software respectively in solving a sequence of LP problems.

Graph ① of Fig. 10 shows the time spent by the RSM in solving a sequence of LP problems of dimensions m=40,80,…,800 in standard form with no size reduction (using γ_{TH}=-1). While graph ② shows the time spent by the *KKT Simplex method* in solving the same sequence but in the free variable form sated in (1) with size reduction (using γ_{TH}=-0.07). We can see that the *KKT Simplex method* is about 4.5 times faster than the RSM in solving each of these problems.

In a grasp analysis situation, the user may choose the precision of the grasp linearization and the corresponding solution time τ_m by changing the problem size m. A reasonable precision is obtained with m=400 [11].

Fig. 11 shows the time spent by the IBM ILOG CPLEX optimization software in solving a sequence of LP problems of dimensions m=25, 50, …, 400.

From Fig. 11 we can see that the reduced LP problem (using γ_{TH}=-0.07) with free variables ② is solved about 2 times faster than the non-reduced LP problem (using γ_{TH}=-1) in standard form ①. Comparing the graphs of figures 10 and 11 we see that the CPLEX software is completely faster than our Matlab implementation. We neither optimized the programming code nor performed any numerical improvement. Even though the great difference in time response, both implementations show that the solution ② of the reduced LP problem is better than that of ①.

4.4 Combination of a Primal-Dual Interior-Point Method and the KKT Simplex Method

The one-phase nature of the *KKT Simplex method* is shown in Fig. 12 by means of its combination with a primal-dual interior-point method (PD-IPM) [2, 3, 42]. The figure shows the times $\tau_{IPM}\cong 13$ milliseconds and $\tau_{KKT}\cong 2$ milliseconds spent per step by the PD-IPM and *KKT Simplex method* respectively in solving a LP problem of 400 constraints. The LP problem in standard form and the PD-IPM are briefly described in appendix B.

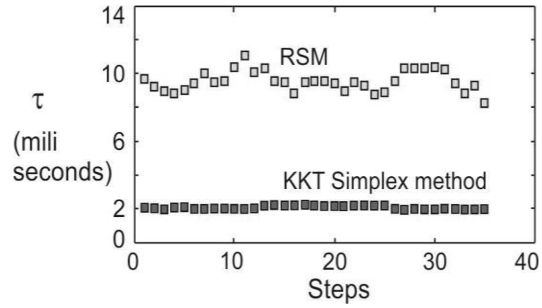


Fig. 9. Time τ per step

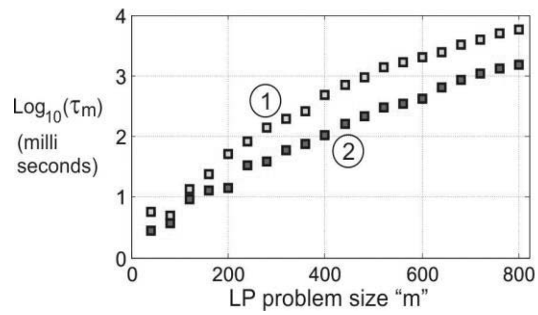


Fig. 10. Solution time spent per LP problem of size "m" by the *KKT Simplex method*

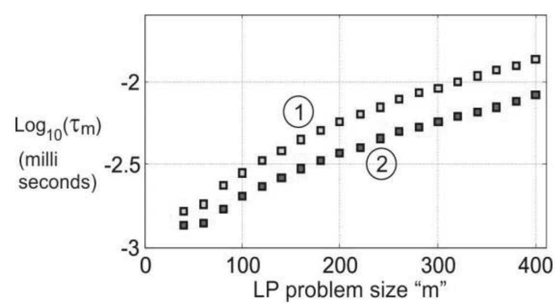


Fig. 11. Solution time spent per LP problem of size "m" by the IBM ILOG CPLEX optimization software

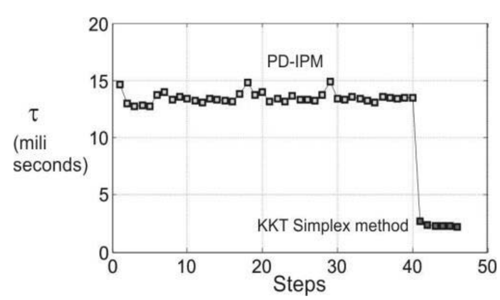


Fig. 12. Time spent per step

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

The PD-IPM brings the starting point, denoted as x_{IPM}, of the *KKT Simplex method* close to the optimal vertex $\mathbf{x}^*$ in 40 steps as it is shown in the graph of figure 12. Then in just 6 more steps the *KKT Simplex method* reaches $\mathbf{x}^*$ in just one simplex phase:

$$\mathbf{x_{IPM}}=[\,2.025,1.470,1.147,0.267,0.448,-0.843]^T\,, \quad \mathbf{x}^*=[\,2.023,1.468,1.153,0.262,0.455,-0.842]^T. \tag{43}$$

Since $\mathbf{x_{IPM}}$ is close enough to $\mathbf{x}^*$: $\|\mathbf{x}^*-\mathbf{x_{IPM}}\|\cong 10^{-3}$, the first vertex found on the polyhedron $\mathscr{P}$ is $\mathbf{x}^*$. The first vertex is reached in "n" simplex steps in general (n=6 for grasp analysis). The tested combination is not efficient because the PD-IPM solved a LP problem in standard form. However, the *KKT Simplex method* maintains its efficiency.

5 Conclusions

In this paper, we have proposed a one-phase method, named as *KKT Simplex method*, which efficiently solves linear programming problems for grasp analysis of robotic hands. The proposed method possesses the three following properties: (i) The one-phase nature by dealing with free variables directly while choosing the entering and leaving variables, which enables the method to start at any point of the polyhedron of feasible solutions, (ii) the reduction of the problem size by the identification of nonbinding constraints as a pre-solution procedure which is based on an angular measure and preserves the *KKT cone*, and (iii) the reduction of the number of simplex steps by means of an *angular pricing strategy*.

The one-phase property of the *KKT simplex method* is derived from the direct processing of the optimization free variables. The efficiency of the *KKT Simplex method* derives from: (i) the suppression of nonbinding constraints, and (ii) the reduction of the number of simplex steps by means of the *angular pricing strategy*.

Numerical experiments on LP problems for grasp analysis show a reduction of 40% of the problem size by the identification of nonbinding constraints. Although a high size reduction is achieved, the computational cost of the identification is negligible: on the order of the time for one simplex step, at least when solving a LP problem of 400 constraints.

The fact that the *KKT Simplex method* can start at any point of the polyhedron and reach the first vertex on the polyhedron of feasible solutions in "n" simplex steps is significant. That is, this fact makes the *KKT Simplex method* suitable to be combined with an interior-point method as it was shown in the experiments. Since the PD-IPM is a polynomial-time method in solving LP problems [2, 43] and that the *KKT Simplex method* reaches the optimal solution in a reduced number of steps when it starts at the solution delivered by the PD-IPM, we suggest, as a future work, to search the proposed combination for a polynomial-time property [3].

Although the method of identification of nonbinding constraints works in LP problems for grasp analysis, the method is subjected to the tuning of the threshold coordinate γ_{TH} for a general LP problem. However, γ_{TH}=-0.5 is suggested as a conservative value.

Appendix A. LP Problem for Grasp Analysis

In grasp and manipulation planning, the two most important classes of grasp are known as *form-closure* and *force-closure* grasps [5, 25, 41]. In this paper, grasp analysis refers to the decision whether a grasp is *force-closure* or not.

Many works have been reported to solve grasp analysis, among them are: [11, 23, 25, 32, 33, 35, 41]. Liu [23] and Ding [11] proposed a new LP formulation of the grasp analysis based on the duality between convex hull and convex polytopes. Roa and Suárez [32, 33] presented a geometrical approach to compute *force-closure* grasp with or without friction.

We will focus on the LP problem introduced by Liu [23] and Ding [11]. The details of the LP formulation may be found in these works.

Assume a polyhedral rigid object is grasped with 4 fingers in a 3D workspace. Each finger is in a point to point frictional contact with the object with the same frictional coefficient μ_f=0.5. The grasping position $\boldsymbol{r_i}$ and normal $\boldsymbol{n_i}$ vectors, with respect to the center of mass of the object, are

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

the same as that used in [11]: $\mathbf{r_1}=[2,0,0]^T$, $\mathbf{r_2}=[0,1.5,0]^T$, $\mathbf{r_3}=[0,0,2]^T$, $\mathbf{r_4}=[1.2,-2,0]^T$, $\mathbf{n_1}=[1,0,0]^T$, $\mathbf{n_2}=[0,1,0]^T$, $\mathbf{n_3}=[0,1,0]^T$, $\mathbf{n_4}=[0,-1,0]^T$. At each contact points, the friction cone is linearized by a polyhedral convex cone with $\overline{m}$ sides. Therefore, the number of constraints of the resultant LP problem is $m=4\overline{m}$.

Appendix B. PD-IPM

By applying to (1) the following transformation [4]

$$\mathbf{x}=(\mathbf{x}^+-\mathbf{x}^-),\ \mathbf{x}^+,\mathbf{x}^-\geq\mathbf{0}\ , \qquad \textbf{(B.1)}$$

we get the LP problem in the standard form:

$$\begin{aligned}&\underset{x}{\text{maximize}}\ \mathbf{c}^T\mathbf{x}\\ &\text{subject to: } A\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq\mathbf{0},\end{aligned} \qquad \textbf{(B.2)}$$

where the parameters and variables have been recast as follows:

$$\begin{aligned}\mathbf{c}^T&\equiv[\mathbf{c}^T,-\mathbf{c}^T,\mathbf{0}]\in R^{(2n+m)},\\ A&\equiv[A,-A,I]\in R^{m\times(2n+m)},\\ \mathbf{x}&\equiv[\mathbf{x}^+,\mathbf{x}^-,\mathbf{s}]^T\in R^{(2n+m)},\\ &=[x_1,x_2,\cdots x_n,x_{n+1},\cdots x_{2n},s_1,s_2,\cdots s_m]^T.\end{aligned} \qquad \textbf{(B.3)}$$

The parameters and variables in the right hand side of (B.3) correspond to the LP problem (1) with free variables, while those in the left hand side correspond to the standard form (B.2), which will be used just in this appendix.

The first order optimality conditions to (B.2) are

$$\begin{bmatrix}A\mathbf{x}-\mathbf{b}\\ A^T\mathbf{y}+\mathbf{v}-\mathbf{c}\\ X\mathbf{v}-\gamma\mu\mathbf{e}\end{bmatrix}=0, \qquad \textbf{(B.4)}$$

where the first equality enforces primal feasibility, the second enforces dual feasibility, $\mathbf{y}\in R^m$ are Lagrange multipliers, $\mathbf{v}\in R^{(2n+m)}$ are dual slacks variables, $\mathbf{e}=[1,\cdots,1]^T\in R^{(2n+m)}$, $\mu=\mathbf{x}^T\mathbf{v}/(2n+m)$, and $\gamma\in[0,1)$ is proposed. Assuming $\mathbf{x}>\mathbf{0}$ and $\mathbf{v}>\mathbf{0}$, then by a Taylor approximation of (B.4) to the first order, we obtain [2, 42]

$$\begin{bmatrix}A & 0 & 0\\ 0 & A^T & I\\ V & 0 & X\end{bmatrix}\begin{bmatrix}\mathbf{d_x}\\ \mathbf{d_y}\\ \mathbf{d_v}\end{bmatrix}=-\begin{bmatrix}A\mathbf{x}-\mathbf{b}\\ A^T\mathbf{y}+\mathbf{v}-\mathbf{c}\\ X\mathbf{v}-\gamma\mu\mathbf{e}\end{bmatrix}, \qquad \textbf{(B.5)}$$

where

$$\begin{aligned}V&=\text{diag}\{v_1,\cdots v_{(2n+m)}\},\\ X&=\text{diag}\{x_1,\cdots x_{(2n+m)}\},\\ \mathbf{r_P}&=\mathbf{b}-A\mathbf{x},\\ \mathbf{r_D}&=\mathbf{c}-A^T\mathbf{y}-\mathbf{v},\\ \mu&=\mathbf{x}^T\mathbf{v}/(2n+m).\end{aligned} \qquad \textbf{(B.6)}$$

The PD-IPM of the Algorithm A.1 computes a feasible solution $\mathbf{x}_{IPM}$ to the LP problem in the standard form (B.2) with parameters $A,\mathbf{b},\mathbf{c}$ given in the left hand side of (B.3). We choose $(\mathbf{x^0},\mathbf{y^0},\mathbf{v^0})=(0.1\mathbf{e},\mathbf{0},0.1\mathbf{e})$ as the initial point, the parameter $\epsilon_G=0.00007$ to terminate the algorithm, $\gamma=0.005$, and $\theta=0.5$.

Algorithm A.1. The PD-IPM adapted from [2, 42]

Step 1. Choose $(\mathbf{x^0},\mathbf{y^0},\mathbf{v^0})$ such that $\mathbf{x^0},\mathbf{v^0}>\mathbf{0}$, $\mathbf{y^0}=\mathbf{0}$, and $\epsilon_G>0$. Let k = 0.

Step 2. Let $\mathbf{r_P^k}=\mathbf{b}-A\mathbf{x^k}-\mathbf{u^k}$, $\mathbf{r_D^k}=\mathbf{c}-A^T\mathbf{y^k}-\mathbf{v^k}$, and $\mu^k=(\mathbf{x^k})^T\mathbf{v^k}/(2n+m)$.

Step 3. If $(\mathbf{x^k})^T\mathbf{v^k}\leq\epsilon_G$, then terminate.

Step 4. Pick $\gamma\in[0,1)$ and compute the increments $(\mathbf{d_y},\mathbf{d_v},\mathbf{d_x})$:

$$\begin{aligned}\mathbf{d_y}&=M^{-1}\mathbf{r},\\ \mathbf{d_v}&=\mathbf{r_D^k}-A^T\mathbf{d_y},\\ \mathbf{d_x}&=-\mathbf{x^k}+(V^k)^{-1}[\gamma\mu^k\mathbf{e}-X^k\mathbf{d_v}],\end{aligned}$$

where $M=A(V^k)^{-1}X^kA^T$ and $\mathbf{r}=\mathbf{b}+A(V^k)^{-1}(X^k\mathbf{r_D^k}-\gamma\mu^k\mathbf{e})$.

Step 5. Compute the step size: $\alpha=\min\{1,\theta\alpha^{max}\}$ for some $\theta\in(0,1)$, where: $\alpha^{max}=\min\{\alpha_P^{max},\alpha_D^{max}\}$, $\alpha_P^{max}=\min_j\{-x_j^k/d_{xj}|d_{xj}<0\}$, $\alpha_D^{max}=\min_j\{-v_j^k/d_{vj}|d_{vj}<0\}$.

Step 6. Update:

$$\begin{bmatrix}\mathbf{x^{k+1}}\\ \mathbf{v^{k+1}}\\ \mathbf{y^{k+1}}\end{bmatrix}=\begin{bmatrix}\mathbf{x^k}\\ \mathbf{v^k}\\ \mathbf{y^k}\end{bmatrix}+\alpha\begin{bmatrix}\mathbf{d_x}\\ \mathbf{d_v}\\ \mathbf{d_y}\end{bmatrix}.$$

Step 7. Let k = k+1 and return to step 2.

Step 8. Recover the optimal solution to the original LP problem: $x_{IPM}=[x_1,x_2,\cdots x_n]^T-[x_{n+1},\cdots x_{2n}]^T$ from: $x=[x_1,x_2,\cdots x_n,x_{n+1},\cdots x_{2n},u_1,u_2,\cdots u_m]^T$.

END of the algorithm A.1.

ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-2-2014-029

References

1. **Al-Najjar, C. & Behnam, M. (2011).** Hybrid LP: Finding Advanced Starting Points for Simplex, and Pivoting LP Methods. *Computers & Operations Research,* 38(2), 427–434.
2. **Andersen, E.D. (2001).** *Linear optimization: Theory, methods and extensions*. Retrieved from http://plato.asu.edu/ftp/linopt.pdf.
3. **Andersen, E.D. & Ye, Y. (1996).** Combining Interior-Point and Pivoting Algorithms for Linear Programming. *Management Science*, 42(12), 1719–1731.
4. **Bazaraa, M.S., Sherali, H.D., & Shetty, C.M. (2006).** *Nonlinear Programming Theory and Algorithms*. Hoboken, N.J.: Wiley-Interscience.
5. **Bicchi, A. (1995).** On the Closure Properties of Robotics Grasping. *The International Journal of Robotics Research,* 14(4), 319–334.
6. **Borgwardt, K.H. (1987).** *The Simplex Method, A probabilistic Analysis.* Berlin; New York: Springer Verlag.
7. **Brearley, A.L., Mitra, G., & Williams, H.P. (1975).** Analysis of Mathematical Programming Problems Prior to Applying the Simplex Algorithm. *Mathematical Programming, 8(1), 54–83.*
8. **Chvatal, V. (1983).** *Linear Programming.* New York: W. H. Freeman.
9. **Corley, H.W., Rosenberger, J., Yeh, W.C., & Sung, T.K. (2006).** The Cosine Simplex Algorithm. *The International Journal of Advanced Manufacturing Technology,* 27(9-10), 1047–1050*.*
10. **Dantzig, G.B. (1963).** *Linear Programming and Extensions*. New Jersey, UK: Princeton University Press.
11. **Ding, D. (2001).** *Grasp Synthesis of Multi-fingered Robotic Hands. PhD Thesis.* Chinese University of Hong Kong, Hong Kong, China*.*
12. **El-Khoury, S., Sahbani, A., & Bidaud, P. (2011).** 3D Objects Grasps Synthesis: A Survey. *13th World Congress in Mechanism and Machine Science,* Guanajuato, México.
13. **Floudas, C.A. & Pardalos, P.M. (2009).** *Encyclopedia of Optimization*. New York: Springer.
14. **Fukuda, K. & Terlaky, T. (1997).** Criss-Cross Methods: A Fresh View on Pivot Algorithms. *Mathematical Programming,* 79(1-3), 369–395.
15. **Gal, T. (1995).** *Postoptimal Analyses, Parametric Programming, and Related Topics: degeneracy, multicriteria decision making, redundancy*. Berlin; New York: Walter de Gruyter.
16. **Gondzio, J. & González-Brevis, P. (2012).** *A New Warmstarting Strategy for the Primal-Dual Column Generation Method*. University of Edinburgh, Edinburgh, UK. (Technical Report ERGO 12-007).
17. **Harris, P.M.J. (1973).** Pivot Selection Methods of the Devex LP Code. *Mathematical Programming, 5(1), 1–28*.
18. **Illés, T., Szirmai, Á., & Terlaky, T. (1996).** *The Finite Criss-Cross Method for Hyperbolic Programming*. (Report 96-103). Delft, The Netherlands: The Faculty of Technical Mathematics and Informatics.
19. **Junior, H.V. & Lins, M.P.E. (2005).** An improved initial basis for the Simplex algorithm. *Computer and Operations Research,* 32(8), 1983–1993*.*
20. **Kaluzny, B.L. (2001).** *Finite Pivot Algorithms and Feasibility*. Master of Science Thesis. McGill University, Montreal, Canada.
21. **Karwan, M.H., Lotfi, V., Telgen, J., & Zionts, S. (1983).** *Redundancy in Mathematical Programming*. Berlin; New York: Springer-Verlag.
22. **Lippiello, V., Siciliano, B., & Villani, L. (2013).** Multi-fingered grasp synthesis based on the object dynamic properties. *Robotics and Autonomous Systems,* 61(6), 626–636*.*
23. **Liu, Y.H. (1999).** Qualitative Test and Force Optimization of 3-D Frictional Form-Closure Grasps Using Linear Programming. *IEEE Transaction of Robotics and Automation*, 15(1), 163–173.
24. **Megido, N. (1991).** On Finding Primal-Dual Optimal Bases. *ORSA Journal on Computing,* 3(1), 63–65*.*
25. **Mishra, B., Schwartz, J.T., & Sharir, M. (1987).** On the Existence and Synthesis of Multifinger Positive Grips. *Algorithmica, 2(1-4), 541–558*.
26. **Mosso-Vázquez, A. (2012).** *Método Simplex KKT para la Solución del Análisis de la Sujeción por Manos Robóticas*. Tesis de Doctorado. Universidad Autónoma del Estado de Morelos (UAEM), México.
27. **Mulmuley, K. (1994).** *Computational Geometry*. Englewood Cliffs, NJ: Prentice Hall.
28. **Paparrizos, K., Samaras, N., & Stephanides, G. (2003).** A New Efficient Primal Dual Simplex Algorithm. *Computers & Operations Research,* 30(9), 1383–1399.
29. **Paulraj, S. & Sumathi, P. (2010).** A Comparative Study of Redundant Constraints Identification Methods in Linear Programming Problems. *Mathematical Problems in Engineering,* Volume 2010, Article ID 723402.
30. **Paulraj, S. & Sumathi, P. (2012).** A New Approach for Selecting a Constraint in Linear Programming Problems to Identify the Redundant Constraints.

International Journal of Scientific and Engineering Research, 3(8), 1345–1348.

31. **Rao, C.R. (2002).** *Linear Statistical Inference and Its Applications* (2nd ed.). New York: Wiley.

32. **Roa, M.A. & Suarez, R. (2007).** Geometrical approach for grasp synthesis on discretized 3D objects. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 2007), San Diego, CA, 3283–3288.

33. **Roa, M.A. & Suarez, R. (2009).** Computation of Independent Contact Regions for Grasping 3-D Objects. *IEEE Transactions on Robotics,* 25(4), 839–850.

34. **Sahbani, A., El-Khoury, S., & Bidaud, P. (2012).** An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems,* 60(3), 326–336.

35. **Salisbury Jr., J.K. (1982).** *Kinematic and Force Analysis of Articulated Hands*. PhD. dissertation, Stanford University, Stanford, CA.

36. **Shang, S. (1997).** New Variants of Finite Criss-Cross Pivot Algorithms for Linear Programming. *European Journal of Operational Research,* 116(3), 607–614.

37. **Terlaky, T. (1985).** A Convergent Criss-Cross Method. *Optimization: A Journal of Mathematical Programming and Operations Research,* 16(5), 683–690.

38. **Thompson, G.L., Tonge, F.M., & Zionts, S. (1966).** Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems. *Management Science*, 12(7), 588–608.

39. **Trigos, F. (2002).** Sobre la Reducción de Modelos de Programación Lineal. *Métodos Numéricos en Ingeniería y Ciencias Aplicadas,* 1, 409–418.

40. **Trigos-Salazar, F., Frausto-Solís, J. F. and Rivera-López, R. (2002).** A Simplex-Cosine Method for Solving Hard Linear Problems. *Advances in Simulation, System Theory and Systems Engineering, WSEAS Press, ISBN 960852 70X, pp 27-32*.

41. **Trinkle, J.C. (1992).** On the Stability and Instantaneous Velocity of Grasped Frictionless Objects. *IEEE Transactions on Robotics and Automation,* 8(5), 560–572.

42. **Wright, S.J. (1997).** *Primal-Dual Interior-Point Methods*. Philadelphia: The Society for Industrial and Applied Mathematics.

43. **Ye, Yinyu (1997).** *Interior point Algorithms Theory and Analysis*. John Wiley and Sons, Inc.

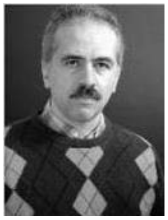
44. **Yeh, W.C. & Corley, H.W. (2009).** A simple direct cosine simplex algorithm. *Applied mathematics and Computation,* 214(1), 178–186.

45. **Zionts, S. (1969).** The Criss-Cross Method for Solving Linear Programming Problems. *Management Science,* 15(7), 426–445.

Alejo Mosso Vazquez received an engineering degree in Communication and Electronics from the School of Electrical and Mechanical Engineering of the National Polytechnic Institute (IPN) of Mexico in 1975. He holds M.Sc. from CINVESTAV-IPN, Mexico (1986), M.Sc. from ITESM, Mexico (2003), and Ph.D. in Engineering and Applied Sciences from CIICAP-UAEM, Mexico (2012). His research interests are optimization and linear control, and their applications to robotics.

David Juárez-Romero received his engineering degree in Chemical Engineering from the Chemical Faculty-UNAM. He has M.Sc. and Ph.D. from the Imperial College, University of London, U.K. His research interest is the improvement of the design and operation of processes of separation-transformation related with energy machines. He develops methodologies of analysis, design, and control of these processes. He is a member of the National System of Researchers of Mexico.

Marco Antonio Cruz-Chávez holds Ph.D. in Computer Science from ITESM, Mexico (2005). He has been a Research Professor since 2004 at CIICAP-UAEM, Mexico, where he was appointed as Head of the Academic Staff of Research on Optimization and Software. He is a member of the National System of Researchers of Mexico. He is currently Technical President of CICos Congress and Technical Editor of the journal *Optimization and Software* published by the UAEM. His research interests are resources assignment optimization.



Luis Enrique Sucar has Ph.D. in Computing from the Imperial College, London (1992) and M.Sc. in Electrical Engineering from Stanford University (1982). He is a Senior Research Scientist at INAOE, Puebla, Mexico. Dr. Sucar is a member of the National System of Researchers of Mexico, the Mexican Academy of Sciences, and Senior Member of the IEEE. His main research interests are in graphical models and probabilistic reasoning, and their applications in computer vision, robotics and biomedicine.

Article received on 11/06/2012, accepted on 07/03/2014.