Post-Optimality Analysis of the Optimal Solution
of a Degenerate Linear Program Using a Pivoting Algorithm

Jan Stallaert
University of Connecticut, Storrs CT
stallaert@sba.uconn.edu *

Final Version, June 10, 2005

Abstract

This paper gives a theory and method that specifies how the optimal solution of a linear program changes when the right hand side of the original problem is changed and when the original optimal solution exhibits primal degeneracy. The method determines an optimal change vector as the resource availabilities change, and it calculates a range for which this vector is valid. Resource availabilities are allowed to change simultaneously in any arbitrary proportion, and the concept of an “efficient resource bundle” is introduced. The geometry of the optimal change vector is presented from which the desired results are derived. The connection between the geometrical results and their algebraic calculation in tableau-form is shown. Our method uses a pivoting algorithm and the relationship with post-optimality results from interior-point methods will be established.

Keywords: Linear Programming, Post-Optimality Analysis, Shadow Prices, Simplex method, Degeneracy

Abbreviated Title: Post-Optimality Analysis for Degenerate LPs

*Mailing Address: OPIM Department, Unit 1041, Storrs, CT 06269
1 Introduction

Linear programming textbooks often end their discussion of the simplex method by referring to a post-optimality analysis that can easily be performed by means of the final simplex tableau. However, it is then mentioned that caution has to be taken when either primal or dual solution is degenerate. In that case, usual post-optimality results do no longer hold. For example, when the primal solution is degenerate, it is possible that the change in the optimal solution when the right hand side changes, is only valid in a zero range, which is rather useless from a practical perspective. Another unique phenomenon that occurs for degenerate problems is the complementarity effect between resources. For example, if in the degenerate case resources $a$ and $b$ have shadow prices of $p(a) \geq 0$ and $p(b) \geq 0$ respectively, then it is possible that acquiring $a$ and $b$ in a certain proportion, say in the amounts $r_a$ of $a$ and $r_b$ of $b$ yields for the value of the bundle: $p_r(a,b) > r_a p(a) + r_b p(b)$, so the value of the bundle $p_r(a,b)$ is worth more than the weighted sum of the values of resources $a$ and $b$ separately (for non-degenerate problems equality always holds). At first glance, a complementarity effect seems rather surprising for a problem that consists of all linear functionals. Jansen et al. [17] describe the problems with commercial LP packages and outline three approaches how to deal with the difficulties caused by degeneracy.

Research on post-optimality analysis using simplex tableaus goes back more than twenty years [3, 6, 18], but the most recent stream uses interior point methods to obtain post-optimality results in the presence of non-uniqueness and degeneracy. Given an optimal interior point solution, an optimal partition can be identified [4] which can then be used for sensitivity analysis in the presence of degeneracy. Adler and Monteiro [1] find all breakpoints of the parametric objective function when the perturbation vector $r$ is kept constant. To compute every breakpoint, an Oracle LP problem with an auxiliary column needs to be solved. Yildirim and Todd [23] describe the computation of the optimal solution to a perturbed system in one interior point iteration. As in the latter paper, this paper primarily focuses on the optimal change vector, i.e., the direction in which the optimal solution changes as the resource availabilities are varied, but we use an extreme point method to arrive at those conclusions. If the optimal solution and the optimal change vector are both unique, the interior and extreme point methods of course give the same optimal change vector.

Many papers on post-optimality analysis (e.g., [2, 3, 6, 17, 11, 13, 14, 18] focus on how the optimal value of the LP changes when the right hand side changes, and construct methods...
on how to find the breakpoints and the rate of change (shadow price) of a piecewise linear optimal value function. The purpose there is to find the shadow prices $u_r$ of some resource (combination) $r$ and the breakpoints $\tau$ such that the optimal value takes the expression $v^*_r(t) = wy^*_r(t) = wy^* + tu_r \forall t \in (0, \tau]$, where $wy^*$ is the value of the original optimal solution. Our focus of attention, however, is on how the change in certain resource availabilities affects the optimal solution, for the following reasons. The primary driver of this research is the appearance of market-based decomposition-like algorithms [7, 16] where different iterations require the solution of perturbed problems where resource availabilities are varied and where the knowledge of the optimal change vector is needed to compute ask and bid prices. In addition, the proof of finite convergence of such algorithms hinges upon the generation of extreme points of the perturbed system at each step, hence our use of extreme point methods rather than interior point methods. Another reason for our interest in the optimal change vector rather than shadow prices of the resource(s), is practical. When linear programming is used to solve planning problems (e.g., production planning, etc.) managers may sometimes be more interested in knowing how the optimal solution changes as a function of a right hand side change, because many times data for the latter has been forecast and is subject to change before the plan is executed. For example, managers might be more interested in knowing how their production plan would change when demand forecasts for the different products change (or when resource availabilities display some variance) rather than the effect of such change on the objective function, in order to anticipate the change in production plan and take the necessary precautions (e.g., not take a machine down for maintenance if it’s likely to become a bottleneck when forecasts change).

In the rest of the paper, when we mention “degeneracy”, we mean “primal degeneracy”. That is, we do not study the effects of dual degeneracy and multiple primal optima. In case we have multiple primal optimal solutions, our method just gives one optimal change vector from one particular original extreme point out of many. In this setting we can also obtain an equivalence between the optimal basis approach and optimal partition approach when computing the valid range for a shadow price. Under the assumptions stated above we answer the following questions:

1. Find an optimal change vector for a given change in resource(s).

2. In what range does the optimal change vector remain constant?

3. What are the “efficient” proportions for a change in resource availability, i.e., in what
proportion should one change the availability of several resources such that no portion of the acquired resources becomes unused in the new optimal solution?

4. If we consider the polyhedron that represents the “fixed” constraints, i.e. the ones where there is no change in resource availability, in what face of this polyhedron does the optimal change vector lie?

Using Convex Analysis [20] Akgül [2] derived shadow prices when the primal optimal solution is degenerate and generalized the results for the case in which resource availabilities are allowed to change simultaneously and in any arbitrary proportion. Since the analysis is done in dual space, it is not immediately clear how to get the optimal change vector that is associated with this shadow price, neither is it shown whether this proportion of resources is “efficient.”

As in [2, 21] we make a distinction between a dual price and a shadow price: the former is not always unique, the latter is always unique. The shadow price for a resource bundle \( r \) is the rate of change of the optimal value when the resource availabilities are increased by a small amount \( \varepsilon r \) (\( \varepsilon > 0 \)). Usually, one concentrates on \( r = e_i \), i.e., the availability of only one resource changes at a time.

Gal [8, 10] uses a degeneracy graph—a graph constructed from basic representations of the optimal vertex—to perform sensitivity analysis in case \( r = e_i \). He also discusses how degeneracy graphs can be used for parametric linear programming under (primal) degeneracy [10]. Because of the close connection between parametric programming and sensitivity analysis, results derived for the former can generally be applied to the latter and vice versa. In fact, our approach can be viewed as using parametric programming of the right hand side, but our derivation is novel since it is based on geometric arguments, rather than degeneracy graphs [10] or convex analysis [2]. Greeenberg’s [13] concept of compatible bases to perform sensitivity analysis for \( r = e_i \) is closely related to the present work. This paper extends the results in [13] by deriving necessary and sufficient conditions for a compatible basis and by showing that such conditions imply that a compatible basis can be found by solving an LP problem. This LP problem is a subproblem of the original LP and can be solved starting with an optimal tableau using a dual algorithm. Solving an LP is—in general—considered to be more efficient than enumerating bases when the problem is of anything else than trivial size. Methods using degeneracy graphs essentially enumerate bases in a lexicographic order [11, p. 4-10], which might become inefficient when the number of possible bases becomes
large.

The first part of the paper introduces the underlying theory. The notation and approach closely follow the notation in [6] and [12]. As in [6], our method primarily uses the primal representation of the linear program (as opposed to [2] where the results are obtained via the dual). The second part of the paper discusses an efficient implementation of this theory using a tableau representation with a dual algorithm. Our goal is to better understand the geometry of post-optimality analysis, to present an efficient computational procedure for answering such questions and to show the connection between the geometric results and algebraic computations.

2 Determining the Optimal Change Vector

2.1 Problem Statement

The original linear program of interest is the canonical linear programming problem:

$$\min \, wy$$

subject to:

$$Ay = \begin{bmatrix} A & -I_n \end{bmatrix} y \leq \begin{bmatrix} r^0 & 0 \end{bmatrix} = b$$

with $w \in \mathbb{R}^{1 \times n}$, $y \in \mathbb{R}^{n \times 1}$, $r^0 \in \mathbb{R}^{m \times 1}$, $b \in \mathbb{R}^{(m+n) \times 1}$, $A \in \mathbb{R}^{m \times n}$, $I_n$ the identity matrix of order $n$ and $A \in \mathbb{R}^{(m+n) \times n}$. The matrix $A$ contains the resource constraints ($r^0$ are the original resource availabilities), and the matrix $A$ (vector $b$) contains the resource constraints (resource availabilities) as well as the nonnegativity constraints, and will be referred to later as the complete constraint matrix (resp. complete right hand side).

We assume that (LP) has been solved by an extreme–point algorithm and $y^*$ is an optimal extreme point solution to (LP). Conceptually, all post-optimality questions with respect to a small change in resource availabilities $tr$ with $r \in \mathbb{R}^{m \times 1}$, $t \in \mathbb{R}$, can be answered by solving the problem:

$$\min \, wy$$

subject to:

$$Ay = \begin{bmatrix} A & -I_n \end{bmatrix} y \leq \begin{bmatrix} r^0 + tr & 0 \end{bmatrix}$$
for a “small” $t$. The optimal value and solution of problem $(LP_r(t))$ is a function of $t$ and is parametrized by $r$. Writing $y^*_r(t)$ as the optimal solution to problem $(LP_r(t))$, we want to find a vector $p^*_r \in \mathbb{R}^{n \times 1}$, which we call the optimal change vector, for the linear expression of the optimal solution:

$$y^*_r(t) = y^* + tp^*_r$$

Once the parametric optimal solution is determined, the shadow price of $r$, $u_r$, can be easily computed by the vector multiplication $u_r = wp^*_r$, and we also compute an upper bound $\Delta_r > 0$, for which the optimal change vector remains valid. Hence, the questions we try to answer are to determine

- $p^*_r$: the optimal change vector for an arbitrary $r$; and
- $\Delta_r$: the range for which the expression $y^*_r(t) = y^* + tp^*_r$ stays valid $\forall t \in (0, \Delta_r]$.

In order to exclude trivial cases, we assume that for the resource bundle $r$, $r_i = 0$ when constraint $i$ has slack at $y^*$, and only focus on the “bottleneck” resources, i.e., the binding constraints at $y^*$. In addition, we would like to know whether resource bundle $r$ is an “efficient” bundle.

**Definition:** $r$ is an efficient resource bundle iff there does not exist $r' \leq r$, $r' \neq r$ with $u_{r'} \leq u_r$.

(Note that because our LP problem is a minimization problem with “$\leq$” constraints, we have $u_r \leq 0$ for $r > 0$, so all feasible solutions to the dual of (LP) are non-positive.) In other words, a resource bundle $r$ is efficient if and only if there is no other bundle $r'$ that uses less resources and is as valuable as bundle $r$. Knowing whether a resource bundle is efficient is important from a managerial perspective. As was pointed out earlier, degenerate problems may give rise to a complementarity effect between resources, i.e., the value of their sum is higher than the sum of their values. So, if we decide to acquire or dispose of resources in a bundle rather than one at a time, a manager would be interested in knowing whether these proportions are the minimal ones, i.e., that there does not exist a resource bundle that is worth as much, but uses less resources.

Next, we show how the computation of $p^*_r$ amounts to solving an $n$–dimensional LP problem, but with generally far fewer constraints than the original LP problem. For this
new LP problem we will use the currently available information from the optimal solution of the original problem. The computation of $\Delta_r$ becomes easy after $p_r^*$ has been determined.

We let the matrix $A^- \in \mathbb{R}^{m^- \times n}$ be the submatrix of the complete constraint matrix $A$ of (LP) consisting of all $m^-$ constraints with zero slack at $y^*$. In case the primal solution is non-degenerate, we have $n = m^-$, so from now on we assume that $n + \sigma = m^- > n$ ($\sigma$ is called the degree of degeneracy [10]). In the spirit of Best [6] the matrix $A^-$ is further partitioned into the matrices $B$ and $D^-$: $A^- = \begin{bmatrix} B \\ D^- \end{bmatrix}$ with $B \in \mathbb{R}^{n \times n}$, a matrix of $n$ linearly independent rows for which $wB^{-1} \leq 0$, and $D^- \in \mathbb{R}^{\sigma \times n}$. Similarly, the vector $b_0^- \in \mathbb{R}^{m^- \times 1}$ contains the portion of the original right hand side $b_0$ corresponding to the constraints in $A^-$, and likewise $b_r^-$ is partitioned as $b_r^- = \begin{bmatrix} f_0^- \\ g_0^- \end{bmatrix}$ with $f_0^- \in \mathbb{R}^{n \times 1}$ and $g_0^- \in \mathbb{R}^{\sigma \times 1}$. In the same way, $b_r^-$ contains the portion of $b_r$, the parametric right hand side (the change in resource availabilities) corresponding to $A^-$ and is further partitioned as $b_r^- = \begin{bmatrix} f_r^- \\ g_r^- \end{bmatrix}$. The matrix $D^<$ contains all rows of $A$ with positive slack at $y^*$ and the vector $g^<$ is the portion of the right hand side $b_0$ corresponding to $D^<$ (by assumption $b_r$ and $r$ corresponding to $D^<$ is zero).

$B$ and $f_0$ represent the optimal extreme point $y^* = B^{-1}f_0$, and we denote by $P \triangleq -B^{-1}$, which we call the conjugate primal basis \[^1\] [12] for the primal space $\mathbb{R}^n$. We also assume—by using complementary slackness—that the matrix $P$ yields a dual feasible solution, which implies that (see [6] p. 347) $wP \geq 0$.

The $m^-$ rows of the matrix $A^-$ can be related to the complete constraint matrix $A$ by introducing the one-to-one function $\pi(\cdot) : \{1, 2, \ldots, m^-\} \rightarrow \{1, 2, \ldots, m + n\}$. The $i$-th row in $A^-$ then equals row $\pi(i)$ in $A = \begin{bmatrix} A \\ -I_n \end{bmatrix}$. The partitioning of the constraint matrix and right hand side and their relationship with the original data of the problem is illustrated in Figure 1.

If we express the matrix $P$ as an array of column vectors $P = [p^1, \ldots, p^n]$, then $\{p^1, \ldots, p^n\}$ is a basis for $\mathbb{R}^n$ and consequently the unknown $p_r^*$ can be expressed with respect to the basis $\{p^1, \ldots, p^n\}$ as:

$$p_r^* = P\alpha_r^*$$

for some column vector $\alpha_r^* \in \mathbb{R}^{n \times 1}$ and we can write $y_r^*(t) = y^* + tP\alpha_r^*$. In order to determine

\[^1\] The non-singular matrix $B$ and hence $P$ exist, since we assumed that $y^*$ was an extreme point solution, i.e., it lies on the intersection of $n$ linearly independent constraints.
Figure 1: The relation between the partitioned matrices and vectors.

If we can rewrite \( \text{LP}_r(t) \) by substituting \( y \) with \( \alpha \), and by taking \( y^* \) as a starting solution, yielding:

\[
\min_{\alpha} \ (wy^* + t\ wP\alpha) = wy^* + t\ \min_{\alpha} \ wP\alpha
\]

subject to:

\[
t\ AP\alpha = \begin{bmatrix} AP \\ -P \end{bmatrix} \alpha \leq \begin{bmatrix} (r^0 - Ay^*) + tr \\ y^* \end{bmatrix}
\]

The above problem has the same complete constraint matrix as (LP), i.e., the first set of constraints are the \( m \) resource constraints plus a second set of \( n \) “translated” non-negativity constraints for \( y \). However, for \( t \) sufficiently small, the constraints that have slack at \( y^* \) will still have slack at \( y^*_r(t) \), so we only need to concentrate on the \( m^= \) binding constraints at \( y^* \), i.e., only the contraints in \( A^= \), and we will ignore the constraints with positive slack until we compute \( \Delta_r \). For the constraints binding at \( y^* \), the portion of the right hand side not involving \( t \) is zero, and since \( t > 0 \), we can always re-scale the problem by dividing the right hand side of those binding constraints by \( t \). This new problem has only \( m^= \) constraints, does...
not depend on \( t \), and can be written as:

\[
\min \ wP\alpha \quad \text{(SP\(_r\))}
\]

subject to:

\[
(A^= P) \alpha = \begin{bmatrix} B \\ D^= \end{bmatrix} P\alpha \leq \begin{bmatrix} f_r \\ g_r^= \end{bmatrix} = b_r^=.
\]

We remind that \( BP = -I_n \) and after substituting \( \alpha' = \alpha + f_r \), and translating the right hand side by the lower bounds, we get an LP problem with \( n \) decision variables and \( \sigma \) constraints:

\[
\min (wP)\alpha' \quad \text{(SP'\(_r\))}
\]

subject to:

\[
-I_n \alpha' \leq 0 \\
(D^= P)\alpha' \leq g_r^= + (D^= P)f_r
\]

This explains why we have stated the problem (LP\(_r(t)\)) in terms of the direction vectors \( \{p^j\} \) rather than in terms of \( \Delta y_j = \pm e_j \) (as in a periodic re-start): this would require solving an (LP) with \( m^= > \sigma \) constraints from scratch, without taking advantage of the information of the optimal solution of the original LP problem presently available. In addition, the present formulation will allow us to get additional insight in the geometrical properties of the optimal change vector. From our last formulation, it also becomes apparent why sensitivity analysis for non-degenerate problems is trivial: the LP problem (SP'\(_r\)) has no constraints.

The above problem can be shown to be the dual of the problem of Akgül [2], by which the shadow price \( u_r \) for a resource bundle \( r \) is obtained. We are solving this problem in the original primal space rather than in the dual space—as Akgül does—to allow us to obtain the optimal change vector \( p^*_r \) directly.

Problem (SP'\(_r\)) gives the necessary and sufficient conditions for finding the optimal change vector for the present solution \( y^* \). Solving (SP'\(_r\)) is expected to be more efficient than basis enumeration as there can be many possible bases at a degenerate vertex [5, 19].
2.2 Post-Optimality Results

Since we assumed that we have an optimal solution \( y^* = -Pf \) to the original problem (LP) with \( wP > 0 \), we have a dual feasible solution, and hence the solution \( \alpha' = 0 \) is dual feasible for (SP'\(_r\)) as well, making problem (SP'\(_r\)) an ideal candidate to be solved by the dual algorithm to restore primal feasibility. Restoring primal feasibility to (SP'\(_r\)) corresponds to finding an unblocked optimal change vector. Because a dual feasible solution exists, (SP'\(_r\)) cannot be unbounded, (SP'\(_r\)) can either have (i) a feasible solution; or (ii) an infeasible solution. Case (ii) means that for every \( t > 0 \) problem (LP\(_r\)(t)) becomes infeasible, so there does not exist a non-null change vector and the shadow price \( u_r = \infty \). Because of the construction of (SP'\(_r\)) and its apparent equivalence with (LP\(_r\)(t)), the proof of the next proposition is omitted.

**Proposition 1** Let \( \alpha'^* \) be the optimal solution to (SP'\(_r\)) with \( \alpha^* = \alpha'^* - f \). Then:

\[
p^*_r = P\alpha^*
\]

The shadow price for \( r \) is then computed as \( u_r = wP^*_r \). For calculating the maximum range of the optimal change vector, we need to calculate the maximum movement in the primal space in direction \( p^*_r \) such that a constraint that was not binding at \( y^* \), now becomes binding. These constraints are part of the matrix \( D^< \) (see Figure 1), and its right hand side is expressed as \( g^< \).

**Proposition 2** If (SP'\(_r\)) has a feasible solution, the range for \( t \) where \( y^*_r(t) = y^* + tp^*_r \) is given by:

\[
\Delta_r = \min_{i=1,\ldots,m-\sigma} \left\{ \frac{(g^<_i - d^<_iy^*)}{d^<_ip^*_r} \mid d^<_ip^*_r > 0 \right\}
\]

We set \( \Delta_r = \infty \) when \( D^<p^*_r \leq 0 \).

**Proof**

The quantity \( (g^<_i - d^<_iy^*) \) is the amount of positive slack of constraint \( i \) in \( D^< \) at \( y^* \). Since \( y^*_r(t) = y^* + tp^*_r \), the slack for constraint \( i \) decreases from \( (g^<_i - d^<_iy^*) \) to \( (g^<_i - d^<_iy^*) - td^<_ip^*_r \) when \( d^<_ip^*_r > 0 \), so it is clear that the maximum movement that does not violate any constraint is given by \( \Delta_r \). If \( d^<_ip^*_r \leq 0 \) the slack of constraint \( i \) does not decrease, so these rows can be ignored. By construction of \( p^*_r \), \( A^= tp^*_r \leq tr \), so constraints binding at \( y^* \) can be
Note that \( \Delta_r > 0 \) as required, since \( (g_i^\leq - d_i^\geq y^*) > 0 \) and \( d_i^\geq p^*_r < \infty \) since \( p^*_r \) is bounded. The bound we have computed here is a bound on the length of the optimal change vector \( p^*_r \), not the breakpoint of the optimal value function \( v^*_r(t) = wy^*_r(t) \) such as calculated in [1, 14]. Since \( \Delta_r \) is computed from a compatible basis (see [13]), it cannot be the correct breakpoint for the shadow price. Adler and Monteiro [1] proved that the optimal partition remains valid as long as \( u_r \) does not change and Greenberg [14, Example on p. 437] gives an example where the correct breakpoint for the shadow price can only be computed using the optimal partition. However, since we are using a pivoting method (i.e. bases and extreme points) and require our parametric solution to remain an extreme point of the perturbed system, the proof of proposition 2 shows that \( \Delta_r \) is the best possible bound on the direction of change \( p^*_r \) in this case. Strictly speaking, we should write \( \Delta_r(p^*_r) \) in case \( p^*_r \) is not unique. When \( p^*_r \) is not unique, there might be another optimal change vector \( p'_r \) and it might happen that \( \Delta_r(p'_r) > \Delta_r(p^*_r) \). As was mentioned in the introduction we ignore multiple (primal) optimal solutions or multiple optimal change vectors, or—in the presence of multiplicity—our method will just find one optimal change vector \( p^*_r \) from one specific optimal solution \( y^* \).

The interval for \( u_r \) may be larger than \( \Delta_r \) when computed from the optimal partition [14], but \( \Delta_r \) is the best possible bound for \( p^*_r \). Additional pivots and transitions to neighboring vertices may be required to obtain a new direction vector \( p^*_r \), while \( u_r \) remains invariant. However, the following result gives a special case in which the optimal partition and the compatible basis approach give the same result, i.e. when the true breakpoint for the value function \( v^*_r(t) \) is exactly \( \Delta_r \).

**Proposition 3** If \( \alpha'^* \) is the unique optimal solution to (SP’), then \( \Delta_r \) is a breakpoint of the value function \( v^*_r(t) \).

**Proof**
First we note that if \( \alpha'^* \) is the unique optimal solution to (SP’), then it is implied that \( y^* \) is the unique optimal solution to (LP) for the following reason. If \( y^* \) is not unique, then there must exist another optimal solution \( y^{**} = y^* + \lambda v \) for some scalar \( \lambda > 0 \) and vector \( v \), with \( wv = 0 \). This implies that if \( y'^*_r(t) = y^* + tp^* = y^* + tP(\alpha'^* - f) \) is the optimal parametric solution to (LP, (t)), then there exist an \( \varepsilon > 0 \) such that \( y'^*_r(t) + \varepsilon v \) is also feasible.
because of convexity of the feasible region of \((LP_r(t))\) and optimal since \(wv = 0\). Then, there also exists an optimal change vector \(P(\alpha'^* - f) + \varepsilon v\) contradicting the uniqueness of \(\alpha'^*\), hence \(y^*\) must be unique.

What remains to be proven is that \(v^*_r(t) \neq wy^* + tu^*_r\) when \(t > \Delta_r\) in case \(\alpha'^*\) is unique. Let \(y^1_r \triangleq y^* + \Delta_r p^*_r\) and let \(y^1_r(t)\) be the optimal solution to \((LP_r(t))\) for \(t > \Delta_r\). In case \((LP_r(t))\) becomes infeasible for \(t > \Delta_r\), we set \(u_r = \infty\) and we are done. Otherwise, \(y^1_r(t)\) can be expressed as \(y^1_r(t) = y^1_r(t) = y^1_r + tp^1\) for some optimal change vector \(p^1\). Obviously \(p^1 \neq \lambda p^*_r\) for any \(\lambda > 0\), since otherwise the constraints \(i\) for which \(\Delta_r = \{\frac{(y^*_r - d^*_i - u^*_r)}{d^*_i - p^*_r} | d^*_i p^*_r > 0\}\) in Proposition 2 would be violated. Now suppose that \(\Delta_r\) is not the maximal range and thus \(v^*_r(t) = wy^* + tu^*_r\) for a \(t > \Delta_r\). Then, by convexity, there exists a point \(y^*_r(t + t_1) = y^* + tp^*_r + t_1 p^*_r\) with \(t + t_1 < \Delta_r\) with \(y^*_r(t + t_1)\) feasible and optimal, contradicting the uniqueness of \(\alpha'^*\).

Q.E.D.

Next we turn our attention to the case whether a given \(r\) is an efficient bundle or not.

**Proposition 4** Bundle \(r\) is an efficient bundle iff \(A^- p^*_r = r\).

**Proof**

Since \(p^*_r\) does not violate any binding constraints, we have that \(A^- p^*_r \leq r\). If there is a resource constraint \(a_i\) which is a row of \(A^-\), and \(a_i p^*_r < r_i\), then any resource bundle \(r'\) with \(r_i - a_i p^*_r \leq r'_i < r_i\) will still be feasible in \((SP'_r)\) and have \(u_{r'} = u_r\) and \(r' \leq r\) with \(r' \neq r\), which contradicts the fact that \(r\) is an efficient bundle.

Q.E.D.

**Corollary 5** If bundle \(r\) is not an efficient bundle, then \(r' = r - A^- p^*_r\) is efficient.

The last corollary states that the resource proportions in an efficient bundle can just be read off from the tableau, in the column represented by \(p^*_r\). This corollary has an application in the market-based decomposition algorithms such as in Fan et al. [7]. In such algorithms, one needs to compute “resource bundles” as well as a price for the bundle. The LP problems to be solved are highly degenerate and a method is needed to derive an efficient bundle such that one only acquires exactly enough resources without creating waste in any of the resources.

The following result focuses on the geometry of \(p^*_r\). Define the polyhedron \(\mathcal{P}(S), S \subseteq \{1, \ldots, m + n\}\) as the polyhedron that is generated by a subset constraints of the original
Define \( \Gamma \triangleq \{ i \mid r_i = 0 \} \), i.e., all constraints with fixed right hand side in the perturbed problem, and similarly we have \( \Gamma \triangleq \{ i \mid r_i \neq 0 \} \). The next results show in what face of \( \mathcal{P}(\Gamma^0) \) \( p^*_r \) lies. We will need the following lemma.

**Lemma 6** For a sufficiently small constant \( \bar{t} \), \( y^*_r(\bar{t}) \) is an extreme point solution of \( (LP_r(\bar{t})) \) iff \( \alpha'^* \) is an extreme point solution of \( (SP'_r) \).

**Proof**
1. \( y^*_r(\bar{t}) \) is an extreme point of \( (LP_r(\bar{t})) \) \( \implies \) \( \alpha'^* \) is an extreme point solution of \( (SP'_r) \).
   Suppose \( \alpha'^* \) is not an extreme point, but \( y^*_r(\bar{t}) \) is. But then, there exist \( \alpha^1 \) and \( \alpha^2 \), \( \alpha'^* = \frac{1}{2}(\alpha^1 + \alpha^2) \) and \( p^*_r = P(\frac{1}{2}\alpha^1 + \frac{1}{2}\alpha^2 + f) \). By construction of \( (SP'_r) \) it means that both \( y^1(\bar{t}) = y^* + \bar{t}P(\alpha^1 + f) \) and \( y^2(\bar{t}) = y^* + \bar{t}P(\alpha^2 + f) \) are feasible solutions with \( y^*_r(\bar{t}) = \frac{1}{2}(y^1(\bar{t}) + y^2(\bar{t})) \) contradicting the fact that \( y^*_r(\bar{t}) \) is an extreme point.

2. \( y^*_r(\bar{t}) \) is an extreme point of \( (LP_r(\bar{t})) \) \( \iff \) \( \alpha'^* \) is an extreme point solution of \( (SP'_r) \).
   Suppose that \( \alpha'^* \) is an extreme point, but \( y^*_r(\bar{t}) \) is not. Then, there exist \( y^1(\bar{t}) \) and \( y^2(\bar{t}) \) such that \( y^1(\bar{t}) = \frac{1}{2}(y^1(\bar{t}) + y^2(\bar{t})) \) which are both feasible and so there exist \( p^1_r = y^1(\bar{t}) - y^*_r \) and \( p^2_r = y^2(\bar{t}) - y^*_r \), and thus \( p^1_r = P\alpha^1 \) and \( p^2_r = P\alpha^2 \) yield feasible solutions to \( (SP'_r) \). This contradicts the fact that \( \alpha'^* \) is an extreme point solution.

\[ \text{Q.E.D.} \]

**Proposition 7** If \( \alpha'^* = \alpha^* + f \) is an extreme point solution of \( (SP'_r) \), then \( p^*_r \) lies on a face of \( \mathcal{P}(\Gamma^0) \) of dimension less than or equal to \( |\Gamma| \).

**Proof**
From lemma 6 we know that if \( \alpha' \) is an extreme point of \( (SP'_r) \) then \( y^*_r(\Delta_r) \) is an optimal extreme point solution to \( (LP_r(\Delta_r)) \). So, \( y^*_r(\Delta_r) \) lies on the intersection of at least \( n \) hyperplanes at most \( |\Gamma| \) of which have \( r_i \neq 0 \), and thus at least \( n - |\Gamma| \) of the \( |\Gamma^0| \) constraints binding at \( y^*_r \) (with \( r_i = 0 \)) are still binding at \( y^*_r(\Delta_r) \). Since \( p^*_r = y^*_r(\Delta_r) - y^* \), these same constraints are binding along the direction \( y^* + tp^*_r \), so \( p^*_r \) lies in a face of dimension less than or equal to \( |\Gamma| \).

\[ \text{Q.E.D.} \]

If \( \alpha'^* \) is not an extreme point, then \( y^*_r(\Delta_r) \) is not necessarily an extreme point of \( (LP_r(\Delta_r)) \) and the result does not hold.
In the following part, we concentrate on the special case in which \( r = e_k \), i.e., the effect of the change of availability of only one resource is examined. The first result is a special case of the preceding proposition.

**Corollary 8** If \( \alpha^* \) is an extreme point solution of \((SP_{e_k}')\), then \( p^*_r \) is an edge of the polyhedron \( \mathcal{P}(\Gamma^0) \).

**Corollary 9** If \( \alpha^* \) is an extreme point solution of \((SP_{e_k}')\), then there exists a representation \( P^* \) such that \( p^*_r = p^k \), with \( p^k \) a column of some conjugate basis \( P^* \).

**Proof**

Since \( p^*_r \) is an extreme ray according to corollary 8, there exist \( n - 1 \) linearly independent hyperplanes \( a_i, i = 1, \ldots, n, i \neq k \) from \( \mathcal{A}^= \) that intersect along the half-line \( y^0 + \lambda p^* \), so \( a_i p^* = 0 \) for \( i \neq k \). In addition, since \( a_k p^*_r = -1 \), \( a_k \) is linearly independent from those \( n - 1 \) hyperplanes \( a_i \) and \( B^* = [a_1 \cdots a_n]^T \) is a basis, with \( -B^{-1} = P^* \) and \( p^*_r \) is the \( k \)th column of \( P^* \). Q.E.D.

The previous corollary states that there exist a basis representation for \( y^* \) for which \( p^*_r \) is a column of the conjugate basis \( P^* \). In that case, the solution to \((SP_r)\) with the conjugate basis \( P^* \) \((SP_r)\) is simply \( \alpha = e_k \). Other algorithms, such as the ones in [3] and [18] identify this column in the simplex tableau by finding this basic representation of \( y^* \). This is clearly a special case of our result, but such column does generally not exist in the more general case of a resource bundle, i.e., where \( r \neq \pm e_k \).

### 2.3 Implementation and Illustrative Example

#### 2.3.1 Example and Computation

The labeling convention we use in our tableau is the following. The \( m \) constraints of the original constraint matrix are labeled 1, \ldots, \( m \), the \( n \) nonnegativity constraints are labeled \( m + 1, \ldots, m + n \), consistent with the row indices of the complete constraint matrix \( \mathcal{A} \).

The column labels of the tableau carry the indices of the constraints in the matrix \( \mathcal{A} \) that make up the rows of the matrix \( \mathcal{B} \), or—equivalently—the columns of \( P \). The variables \( \alpha' \) from our problem \((SP_r')\) carry the same labels \( \pi(\cdot) \) as the columns of \( P \). The row indices carry the other constraints: first come the constraints in \( D^= \), followed by the constraints in \( D^- \). We will add one additional data structure: the column that will carry the information
about $-DP^*_r$. The resulting layout is displayed in Figure 2 and is consistent with a tableau representation for the simplex method.

<table>
<thead>
<tr>
<th>Labels $\pi(\cdot)$ from Rows in $D^=$</th>
<th>Labels $\pi(\cdot)$ from Rows in $D^&lt;$</th>
<th>RHS column $g_0 - Dy^*$</th>
<th>New column $-Dp^*_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D^P$</td>
<td>$D^P$</td>
<td>$g^= - D^= y^*$</td>
<td>$g^= - D^= p^*_r$</td>
</tr>
<tr>
<td>$D^P$</td>
<td>$D^P$</td>
<td>$g^&lt; - D^&lt; y^*$</td>
<td>$-D^&lt; p^*_r$</td>
</tr>
<tr>
<td>OBJ</td>
<td>$w^P$</td>
<td>$-wy^*$</td>
<td>$-wp^*_r$</td>
</tr>
</tbody>
</table>

Figure 2: Layout of the tableau.

As a first example, we take the one from [3], also used in [2] with $r = [1 \ 2]^T$ [2, p. 429]:

$$\text{min } -3y_1 - y_2$$

subject to:

$$y_1 + y_2 \leq 1$$
$$2y_1 + y_2 \leq 2$$
$$-y_1 \leq 0$$
$$-y_2 \leq 0$$

Table 1 contains the optimal tableau for this problem ($y^* = [1 \ 0]$), and the new column $-Dp^*_r$ is generated as follows. First, we map $r$ into $f_r = [1 \ 0]^T$ and $g^= [2 \ 0]^T$. Then, using simple algebra it can be shown that the new column can be computed as $-Dp^*_r = g^r + DPf_r$. The matrix $B$ that represents $y^*$ contains rows 1 and 4 from the complete constraint matrix, and thus we label the columns of $P$ as 1 and 4 also. The variables $\alpha'$ also carry the same labels as the columns in $P$, and our starting solution thus has $\alpha' = [0 \ 0]^T$. Without inverting $B$, we can get the matrix $P$ from the present tableau using the formulas in [12] to obtain

$$P = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}.$$

The fact that this column has the opposite sign can be explained by the fact that this column actually represents $(LP_r(t))$, but with constraint set $Ay - tr \leq r^h$ as well as the right hand side of $(SP'_r)$. So, either $\alpha'$ or $p_r$ needs to reverse signs if we insist on the same representation for both problems.
From now on, we treat column $-Dp^*_r$ as the new right hand side, and we “restore” primal feasibility only with respect to the rows that have zero right hand side (column labeled RHS). Other rows can be ignored and are shaded; in this example only the first row (labeled 2) belongs to $D^=\$, so problem (SP'$_r$) has only one constraint. The current tableau is in optimal state (there are no primal infeasibilities in column $-Dp^*_r$ in the degenerate rows) with $\alpha'^* = 0$ and thus $\alpha^* = [0 0]^T - [1 0]^T$, so $p^*_r = -p^1$, $u_r = -3$ (from the row labeled ‘OBJ’) and $\Delta_r = 1$ (from the row labeled ‘3’).

### 2.3.2 Two-fold interpretation of the new column

Because problems (LP$_r$(t)) and (SP'$_r$) have the same representation, the new column in the tableau has a two-fold interpretation. We decided to label it $-Dp^*_r$ since it contains all information about the optimal direction of change $p^*_r$ projected onto the constraints in $D$ and thus $-Dp^*_r$ measures what effect a unit movement in direction of $p^*_r$ has on the constraints in $D$ ($D^= \text{ as well as } D^<\$). E.g., table 1 shows that a unit movement in direction $p^*_r$ will leave the slack of constraint 2 invariant, but will decrease the slack of constraint 3 ($-y_1 \leq 0$ by one unit. However, we could also have labeled the new column $\alpha'$, because the new column contains the optimal weights of $p^*_r = P(\alpha'^* + f)$ (with $P$ the original representation of $y^*$, i.e., the result of solving (LP) originally). This is the interpretation we used throughout the paper: the new column contains the values for the optimization problem (SP'$_r$) (in this interpretation only the $\sigma$ rows from $D^= \text{—the ones with zeroes in column RHS—are relevant, the other entries in the column can be ignored}$). Viewed as the representation of problem (SP'$_r$), the tableau contains not only the optimal weights $\alpha'^*$, but the slacks in resource constraints that will be obtained by a unit step in direction $p^*_r$. So, viewed as the representation of (SP'$_r$), the tableau allows to identify an efficient resource bundle $r$ in a straight-forward manner. Looking at the first column (resource 1), the efficient resource bundle can easily be read from the tableau as $r = [1 2]$ (the proportion for resource 2 can be read from the tableau in its corresponding row) with $u[1 2] = -3$ (in this case we also have

<table>
<thead>
<tr>
<th>Label</th>
<th>1</th>
<th>4</th>
<th>RHS</th>
<th>$-Dp^*_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>OBJ:</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-3</td>
</tr>
</tbody>
</table>

Table 1: Optimal tableau for (LP) with $-D^<p^*_r$ initialized.
Now, in order to get a less trivial example, suppose $r = [-1 -3]$, (so $f_r = [-1 0]^T$ and $g_r = [-3 0]^T$) then after starting with the same representation $B$ of $y^*$ as in table 1 (i.e., rows 1 and 4) and after executing one pivot, we get the optimal tableau in table 2. From this tableau, we can read that $\alpha^{\ast\ast} = [0.5 0]$ and thus $\alpha^{\ast} = [1.5 0]$. Since there is slack of 0.5 for resource constraint 1, it means that $r$ is not an efficient bundle, but $r' = [-1.5 -3]^T$ is. We also get $u_r = 4.5$, $p^{\ast}_r = [-\frac{3}{2} 0]$ and $\Delta_r = \frac{2}{3}$.

In order to illustrate the complementarity effect mentioned in the introduction, we suppose that the objective function from the first example has been changed to $w = [-3 -2]$. Now if we look at the three changes in resource availabilities: $r^1 = [1 0]^T$, $r^2 = [0 1]^T$ and $r^{12} = r^1 + 2r^2 = [1 2]^T$, we get the following results: $u_{r^1} = -1$, $u_{r^2} = 0$ and $u_{r^{12}} = -3 < u_{r^1} + 2u_{r^2}$. So, acquiring the resources in the right proportion $[1 2]^T$ is worth more than the sum of the values of obtaining them separately.

Figure 3 illustrates Proposition 7. For $\Gamma = \{1\}$ and $\Gamma^0 = \{2, 3, 4\}$, $\mathcal{P}(\Gamma^0)$ has two edges $p^1$ and $p^2$, and the proposition states that $p^{\ast}_r$ lies in a face of dimension 1. By inspection, it can be seen that $p^{\ast}_r = p^1$ for $r = [-1 0]^T$. For $\Gamma = \{2\}$, $\mathcal{P}(\Gamma^0)$ also has edges $p^1$ and $p^2$, and for $r = [0 -1]^T$ we get $p^{\ast}_r = p^1$, but for the objective function $[-3 -2]$, we would get $p^{\ast}_r = p^2$. For $\Gamma = \{1, 2\}$, $\mathcal{P}(\Gamma^0)$ is given by the halfspace $y_2 \geq 0$ and by proposition 7, $p^{\ast}_r$ can lie anywhere in this space. In higher dimensions, the results of proposition 7 become less trivial: in 2–space there are at most two edges from any extreme point for any polyhedron.

### Table 2: Optimal tableau for resource bundle $r = [-1 -3]^T$.

<table>
<thead>
<tr>
<th>Label</th>
<th>2</th>
<th>4</th>
<th>RHS</th>
<th>$-D^{&lt;}p^{\ast}_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.5</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>-1.5</td>
</tr>
<tr>
<td>OBJ:</td>
<td>1.5</td>
<td>0.5</td>
<td>3</td>
<td>-4.5</td>
</tr>
</tbody>
</table>

3 Conclusion

We presented geometrical results and a computational procedure to obtain post-optimality results relating to the optimal change vector for degenerate linear programs. Under the
assumption that the primal optimal solution and the optimal change vector are unique, this method computes the breakpoints for the optimal value function just as an interior point method using optimal partitions.

As was shown, the optimal change vector can be computed utilizing the information from the optimal tableau. The new method uses a dual simplex algorithm to compute the optimal change vector and should hence be more efficient than the historical methods [3, 11, 13, 18] that rely upon basis enumeration.

In addition, we generalized these results for the case in which the availability of multiple resources changes at the same time. We also introduced the concept of an efficient resource bundle which maintains the bottleneck activities when acquiring or disposing of some certain resources. This concept can be used by managers to determine in what proportions they should acquire or dispose of resources such that no slack is created.

The computations are easily performed with the addition of just one $m$–dimensional vector, solely by executing pivot operations. Furthermore, we characterized the face that contains the direction of change of the optimal solution. The calculations required to identify the direction of change of the optimal solution led to a tableau representation that could
be interpreted in two ways. From one viewpoint, the tableau determines how the change in optimal solution will affect the resource constraints, as the resource availabilities are changed, from another viewpoint this tableau expresses the direction of change of the optimal solution as a linear combination of the vectors used to represent the optimal solution of the original problem.

Using duality, the proposed method can be used to find ranges on the cost vector such that primal optimal solution remains constant with the optimal value being a simple linear function of the change in cost vector. The method also seems to be directly applicable to the parametric programming problem with respect to the right hand side, which is one possible topic for future research. Using the relationships between primal and dual degeneracy and multiplicity [22] future research will investigate the extension of the present results to the more general case when dual degeneracy is present also.
References


