

Larsson, Torbjörn; Quttineh, Nils-Hassan

Article

One-parametric analysis of column-oriented linear programs

Operations Research Perspectives

Provided in Cooperation with:

Elsevier

Suggested Citation: Larsson, Torbjörn; Quttineh, Nils-Hassan (2023) : One-parametric analysis of column-oriented linear programs, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 10, pp. 1-6,
<https://doi.org/10.1016/j.orp.2023.100278>

This Version is available at:

<https://hdl.handle.net/10419/325764>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

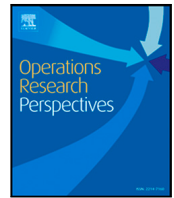
Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



One-parametric analysis of column-oriented linear programs

Torbjörn Larsson, Nils-Hassan Quttineh*

Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden

ARTICLE INFO

MSC:
90C05
90C31
90C10
90C29

Keywords:

Linear programming
Parametric optimization
Column generation
Dantzig–Wolfe decomposition
Bi-objective discrete optimization

ABSTRACT

A linear optimization problem which is amenable to column generation and contains a single parameter in the objective function is considered. We extend and adapt the standard linear programming column generation scheme to effectively and efficiently solve this problem for all values of the parameter. As a potential application we consider bi-objective discrete optimization and describe how the one-parametric column generation scheme can be used to form an outer approximation of the Pareto frontier for such a problem.

1. Derivation

The field of sensitivity and parametric analyses in linear programming is well established since several decades. Such analyses are however typically performed for explicitly given linear programs. In contrast, we make a parametric analysis with respect to a single parameter in the objective function of a linear program that is not explicitly known and can be solved only by means of a column generation procedure. We give a precise algorithm for such an analysis; to the best of our knowledge, there is no previous method for doing this. The parametric analysis is inspired by results within the field of fractional optimization.

In the following the reader is assumed to be well acquainted with linear programming and the column generation principle, and is otherwise referred to, for example, [1,2].

Let $\lambda \in \mathbb{R}_+$, $c_j(\lambda) = c_j + \lambda \Delta c_j$, with $c_j \in \mathbb{R}$ and $\Delta c_j \in \mathbb{R}$, $j = 1, \dots, N$. Let further $A_j \in \mathbb{R}^m$, $j = 1, \dots, N$, and $b \in \mathbb{R}^m$, and consider the one-parametric linear optimization problem

$$z^*(\lambda) = \min \sum_{j=1}^N c_j(\lambda) x_j \quad (1a)$$

$$\text{s.t. } \sum_{j=1}^N A_j x_j = b, \quad (1b)$$

$$x_j \geq 0, \quad j = 1, \dots, N, \quad (1c)$$

which is assumed to have a non-empty and bounded set of feasible solutions. If the feasible set has P extreme points, denoted by x^p ,

$p = 1, \dots, P$, we obtain that

$$z^*(\lambda) = \min_{p=1, \dots, P} \left\{ \sum_{j=1}^N c_j x_j^p + \lambda \sum_{j=1}^N \Delta c_j x_j^p \right\},$$

which reveals that the function $z^* : \mathbb{R}_+ \mapsto \mathbb{R}$ is the pointwise minimum of a finite number of linear functions. Hence it is finite, piecewise linear and concave on its domain; clearly, each piece corresponds to a certain extreme point being optimal and the number of breakpoints is finite.

Assume further that N is huge, but that the set of columns, $A = \{A_1, \dots, A_N\}$, and their corresponding cost coefficients, $c(a)$ and $\Delta c(a)$, $a \in A$, with $c(A_j) = c_j$ and $\Delta c(A_j) = \Delta c_j$, $j = 1, \dots, N$, can be characterized in a compact way. Further, for any value of $\lambda \in \mathbb{R}_+$ and any constant vector $y \in \mathbb{R}^m$ it shall be possible to solve the column generation problem

$$\min_{a \in A} \{c(a) + \lambda \Delta c(a) - y^T a\} \quad (2)$$

without making an explicit enumeration of the columns, but by instead solving some application-specific optimization problem by a tailored algorithm. This means that the one-parametric linear optimization problem is, for a fixed value of λ , amenable to a column generation approach. The set A is then typically described by linear constraints and integrality restrictions, while $c(a)$ and $\Delta c(a)$, $a \in A$, are costs associated with the integer points satisfying the linear constraints.

Remark 1. A one-parametric optimization model of the form (1) may for example arise from a linear bi-objective column-oriented model

* Corresponding author.

E-mail address: nils-hassan.quttineh@liu.se (N.-H. Quttineh).

which is approached by a weighted-sum scalarization. Let $c_1, c_2 \in \mathbb{R}^N$ be two cost vectors, let $x \in \mathbb{R}^N$ be the variable vector, and consider the objective to simultaneously minimize $c_1^T x$ and $c_2^T x$. With $w \in [0, 1]$, the scalarized single objective then becomes $wc_1^T x + (1-w)c_2^T x = c_2^T x + w(c_1 - c_2)^T x$, which is of the type (1a). Due to convexity, the solution of the scalarized linear optimization problem for all $w \in [0, 1]$ will provide the entire Pareto frontier of the bi-objective problem.

A precise algorithm for finding the function z^* is missing in the literature. A naive way would be to sample values of λ and use straightforward column generation for each of them (and exploiting reoptimization facilities). This way would however not guarantee that all breakpoints are found (if the sampling is too coarse) and it can also be unnecessarily computationally demanding (if the sampling is finer than needed). Our goal is to enable an effective and efficient calculation of the breakpoints of z^* , even though the number of columns in the one-parametric linear optimization problem is huge, by deriving an extension of a standard column generation scheme.

Let $\lambda \in \mathbb{R}_+$ and assume that column generation is used to find a basic feasible solution (BFS) that is verified to be optimal, that is, dual feasible, and let $B(\lambda)$ be its basis matrix. Let $c_{B(\lambda)}$ and $\Delta c_{B(\lambda)}$ be the vectors of coefficients c_j and Δc_j , respectively, for the optimal basic variables, and let $c_{B(\lambda)} = c_{B(\lambda)} + \lambda \Delta c_{B(\lambda)}$. The dual feasible solution then is $y(\lambda)^T = c_{B(\lambda)}^T B(\lambda)^{-1}$ and the reduced costs are $\bar{c}_j(\lambda) = c_j(\lambda) - y(\lambda)^T A_j \geq 0, j = 1, \dots, N$.

We now consider a fixed value $\bar{\lambda} \in \mathbb{R}_{++}$ and the problem of finding the largest value of λ such that a BFS that is verified to be optimal at $\bar{\lambda}$ remains verified optimal, that is, dual feasible since primal feasibility is clearly not affected by λ . For this purpose we let $\delta = \lambda - \bar{\lambda} \in \mathbb{R}_+$. The following result, which is easily verified, is instrumental for the analysis.

Proposition 2. *Given a dual feasible BFS at $\bar{\lambda} \in \mathbb{R}_{++}$. In this basis it then holds that $\bar{c}_j(\bar{\lambda} + \delta) = \bar{c}_j(\bar{\lambda}) + \delta \Delta \bar{c}_j(\bar{\lambda}), j = 1, \dots, N$, where $\Delta \bar{c}_j(\bar{\lambda}) = \Delta c_j - \Delta y(\bar{\lambda})^T A_j$ with $\Delta y(\bar{\lambda})^T = \Delta c_{B(\bar{\lambda})}^T B(\bar{\lambda})^{-1}$.*

Notice that, by construction, $\bar{c}_j(\bar{\lambda} + \delta) = 0$ holds for all basic variables and for all $\delta \in \mathbb{R}_+$. The largest value of δ that maintains dual feasibility is

$$\bar{\delta} = \max\{\delta \in \mathbb{R}_+ \mid \bar{c}_j(\bar{\lambda}) + \delta \Delta \bar{c}_j(\bar{\lambda}) \geq 0, j = 1, \dots, N\}. \quad (3)$$

If N were of moderate size, and problem (1) were known explicitly, then $\bar{\delta}$ could be calculated in the straightforward way

$$\bar{\delta} = \min_{j=1, \dots, N} \left\{ -\frac{\bar{c}_j(\bar{\lambda})}{\Delta \bar{c}_j(\bar{\lambda})} \mid \Delta \bar{c}_j(\bar{\lambda}) < 0 \right\}.$$

But because N is huge, $\bar{\delta}$ can clearly not be calculated like this.

We instead find $\bar{\delta}$ by studying the function $h : \mathbb{R}_+ \mapsto \mathbb{R}$ with

$$h(\delta) = \min_{j=1, \dots, N} \{ \bar{c}_j(\bar{\lambda}) + \delta \Delta \bar{c}_j(\bar{\lambda}) \}. \quad (4)$$

This function is clearly finite, piecewise linear and concave on its domain, and has a finite number of breakpoints. Since $\bar{c}_j(\bar{\lambda}) + \delta \Delta \bar{c}_j(\bar{\lambda}) = 0$ holds for all basic variables and all $\delta \in \mathbb{R}_+$, it holds that $h(0) = 0$ and that $h(\delta) \leq 0$ for all $\delta \in \mathbb{R}_+$.

Let h'_{\min} denote the most negative slope of this function. Obviously,

$$h'_{\min} = \min_{j=1, \dots, N} \Delta \bar{c}_j(\bar{\lambda}). \quad (5)$$

If $h'_{\min} \not\leq 0$, then $\bar{c}_j(\bar{\lambda}) + \delta \Delta \bar{c}_j(\bar{\lambda}) \geq 0$ holds for all $j = 1, \dots, N$ and all $\delta \in \mathbb{R}_+$. Hence, $h'_{\min} \not\leq 0$ means that $\bar{\delta} = +\infty$, that dual feasibility holds for all $\delta \in \mathbb{R}_+$, and that the optimal BFS at $\bar{\lambda}$ remains optimal for all $\lambda > \bar{\lambda}$.

If $h'_{\min} < 0$, then $\bar{\delta}$ is finite and from the definition of the function h follows that $\bar{\delta} = \max\{\delta \in \mathbb{R}_+ \mid h(\delta) = 0\}$. As is well known, if the optimal BFS is non-degenerate, then the condition $\bar{c}_j(\bar{\lambda} + \delta) \geq 0$,

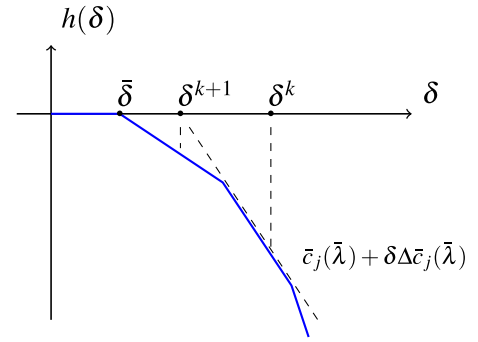


Fig. 1. Illustration of the function h and Newton's method.

$j = 1, \dots, N$, is both necessary and sufficient for maintained optimality, but in case the solution is degenerate, it is only sufficient. In the case of non-degeneracy, the value $\bar{\lambda} + \bar{\delta}$ yields the smallest breakpoint of z^* that is larger than $\bar{\lambda}$, but in the case of degeneracy this value may just yield a degenerate change of basis, which leaves the primal feasible solution unchanged. In the latter case, $\bar{\lambda} + \bar{\delta}$ is not a breakpoint of z^* , but just indicates a transition to a new primal and dual feasible basis for one and the same extreme point.

In order to find the finite $\bar{\delta}$, we suggest using Newton's method on the piecewise linear equation $h(\delta) = 0$. Let $\delta^k \in \mathbb{R}_{++}$ be an iterate and

$$j(\delta^k) \in \arg \min_{j=1, \dots, N} \{ \bar{c}_j(\bar{\lambda}) + \delta^k \Delta \bar{c}_j(\bar{\lambda}) \},$$

giving $h(\delta^k) = \bar{c}_{j(\delta^k)}(\bar{\lambda}) + \delta^k \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})$. Suppose that $h(\delta^k) < 0$. Since δ^k is on a linear piece of h with slope $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})$, a Newton update gives the next iterate

$$\delta^{k+1} = \delta^k - \frac{h(\delta^k)}{\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})} = \delta^k - \frac{\bar{c}_{j(\delta^k)}(\bar{\lambda}) + \delta^k \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})}{\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})} = -\frac{\bar{c}_{j(\delta^k)}(\bar{\lambda})}{\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})}. \quad (6)$$

The function h and the progress of Newton's method for finding its zero are illustrated in Fig. 1.

Proposition 3. *Given any initial iterate $\delta^0 \in \mathbb{R}_{++}$ with $h(\delta^0) < 0$, Newton's method on the piecewise linear equation $h(\delta) = 0$ generates an iteration sequence $\delta^k \in \mathbb{R}_+, k = 0, 1, 2, \dots$, that is monotonically decreasing and finitely converges to the value $\bar{\delta}$, as defined in (3).*

Proof.

Throughout the proof it is assumed that $h(\delta^k) < 0$.

By using $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) \in \partial h(\delta^k)$ in the subgradient-defining inequality for h at δ^k we obtain that $h(\delta) \leq h(\delta^k) + \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})(\delta - \delta^k)$ holds for all $\delta \in \mathbb{R}_+$. The choice $\delta = 0$ gives that $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) \leq h(\delta^k)/\delta^k < 0$. From the update (6) then follows that $\delta^{k+1} < \delta^k$. Since $\bar{c}_{j(\delta^k)}(\bar{\lambda}) \geq 0$, it also follows from (6) that $\delta^{k+1} \geq 0$.

To prove finite convergence we show that the method will in each iteration exclude at least one linear piece of h from further consideration in the search for its zero. We first consider the case that δ^k is not a breakpoint of h and again use that $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) \in \partial h(\delta^k)$. Assume that $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) \in \partial h(\delta^{k+1})$. Using the subgradient-defining inequality for h at δ^k and δ^{k+1} , for the choices $\delta = \delta^{k+1}$ and $\delta = \delta^k$, respectively, gives that $h(\delta^{k+1}) = h(\delta^k) + \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})(\delta^{k+1} - \delta^k)$. By inserting the expression for δ^{k+1} according to (6), it follows that $h(\delta^{k+1}) = 0$, so that the method terminates. Hence, if $h(\delta^{k+1}) < 0$, then $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) \notin \partial h(\delta^{k+1})$, implying that δ^{k+1} is not on the piece of h where δ^k is located. Further, since the iterates are monotonically decreasing, no subsequent iterate can be located on that piece of h . In the case that δ^k is a breakpoint of h , we may instead conclude that no subsequent iterate can be located on the piece of h which has its lower endpoint at δ^k , by again using that the iterates are monotonically decreasing.

From the update (6) follows that $h(\delta^k) = \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})(\delta^k - \delta^{k+1})$, and therefore $h(\delta) \leq h(\delta^k) + \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})(\delta - \delta^k) = \Delta \bar{c}_{j(\delta^k)}(\bar{\lambda})(\delta - \delta^{k+1})$ holds

for all $\delta \in \mathbb{R}_+$. Suppose that $h(\delta) = 0$ holds for some $\delta \in \mathbb{R}_+$. Since $\Delta \bar{c}_{j(\delta^k)}(\bar{\lambda}) < 0$, it must then hold that $\delta \leq \delta^{k+1}$. Hence, if $h(\delta^{k+1}) = 0$ holds, then $\delta \leq \delta^{k+1}$ holds for all other zeros of h , and therefore $\delta^{k+1} = \bar{\delta}$. \square

The construction of the function h and the use of Newton's method to find its zero $\bar{\delta}$ are inspired by results within fractional optimization, see e.g. [3,4].

From the assumption that the linear program is amenable to column generation, it follows that the calculation of the value of the function h in (4) can be replaced by the column generation problem

$$a(\delta) \in \arg \min_{a \in A} \left\{ c(a) + (\bar{\lambda} + \delta)\Delta c(a) - (y(\bar{\lambda}) + \delta \Delta y(\bar{\lambda}))^T a \right\}, \quad (7)$$

for the application under consideration, and the calculation

$$h(\delta) = c(a(\delta)) + (\bar{\lambda} + \delta)\Delta c(a(\delta)) - (y(\bar{\lambda}) + \delta \Delta y(\bar{\lambda}))^T a(\delta). \quad (8)$$

Similarly, the calculation (5) amounts to solving the column generation problem $\min_{a \in A} \{ \Delta c(a) - \Delta y(\bar{\lambda})^T a \}$.

We are now in the position to state a procedure for solving the one-parametric linear optimization problem by column generation. The breakpoints between optimal bases are denoted by $\lambda_0, \lambda_1, \lambda_2, \dots$, where we define $\lambda_0 = 0$ (although it is not likely to be a proper breakpoint). In order to make the transition from one dual feasible basis to the next, a very small step $\varepsilon > 0$ is taken beyond each breakpoint $\lambda_p, p = 0, 1, 2, \dots$. In case it is determined that $\bar{\delta}$ is finite, Newton's method is initialized at a δ^0 such that $h(\delta^0) < 0$ holds. (Such a δ^0 can for example be found by starting with a tentative value and then, if needed, doubling the value until the condition holds.) The complete parametric procedure is summarized in Algorithm 1. The algorithm will clearly find all the breakpoints of the function z^* . Notice that some of the breakpoints λ_p may not be actual breakpoints of z^* , but may instead correspond to a change of dual feasible basis at a degenerate extreme point.

Algorithm 1 One-Parametric Column Generation

input: a one-parametric column-oriented linear program
output: breakpoints $\lambda_0, \lambda_1, \lambda_2, \dots$, and an optimal basis for each interval
let $p = 0$ and $\lambda_0 = 0$
loop
 let $\bar{\lambda} = \lambda_p + \varepsilon$
 apply column generation to solve the linear program for $\lambda = \bar{\lambda}$
 record the optimal basis found
 apply column generation to find $h'_{\min} = \min_{a \in A} \{ \Delta c(a) - \Delta y(\bar{\lambda})^T a \}$
 if $h'_{\min} < 0$ **then**
 let $k = 0$ and use the column generation problem (7) and the expression (8) to find a δ^0 such that $h(\delta^0) < 0$
 while $h(\delta^k) < 0$ **do**
 let $\delta^{k+1} = -\frac{c(a(\delta^k)) + \bar{\lambda}\Delta c(a(\delta^k)) - y(\bar{\lambda})^T a(\delta^k)}{\Delta c(a(\delta^k)) - \Delta y(\bar{\lambda})^T a(\delta^k)}$
 solve the column generation problem (7) for $\delta = \delta^{k+1}$
 use the expression (8) for $\delta = \delta^{k+1}$ to calculate $h(\delta^{k+1})$
 let $k = k + 1$
 end while
 let $\lambda_{p+1} = \bar{\lambda} + \delta^k$
 let $p = p + 1$
 else
 terminate
 end if
end loop

In column generation approaches, all columns that are generated are commonly accumulated in the restricted master problem. When the parametric solution procedure terminates, this problem will then clearly contain all columns needed to find verified optimal solutions for all $\lambda \in \mathbb{R}_+$.

Remark 4. In case the parameter values of interest are bounded from above, like in the application discussed in Remark 1, the algorithm can be simplified. Suppose that the range of interest is $\lambda \in [0, L]$, where $L \in \mathbb{R}_{++}$. Then the calculations of h'_{\min} and δ^0 can be replaced by the calculation of $h(L - \bar{\lambda})$. If $h(L - \bar{\lambda}) \not< 0$ holds, then the algorithm terminates, and otherwise we let $\delta^0 = L - \bar{\lambda}$.

Remark 5. If the original one-parametric linear optimization problem includes columns from several categories, such that several column generation problems are obtained, then the analysis of the function h separates over these column categories, and the value $\bar{\delta}$ is determined as the smallest of the values obtained from the separate categories.

Since Dantzig–Wolfe decomposition is simply an application of the general column generation principle, our approach is directly applicable to one-parametric linear optimization problems with complicating side constraints, and in particular to one-parametric block-angular linear programs.

The work found in the literature that is closest to ours is by Moradi et al. [5]; it presents a column generation, Dantzig–Wolfe type method for finding the Pareto frontier for the bi-objective multi-commodity minimum cost network flow problem. This problem is a special case of a block-angular linear program and it should be straightforward to extend the method in [5] to any such linear program. This method uses the bi-objective simplex method, fractional linear programming and the Charnes–Cooper transformation of a fractional linear program into a standard linear programming problem; hence, it relies fully on linear programming techniques.

In a straightforward application of the Dantzig–Wolfe decomposition principle to a single-objective multi-commodity minimum cost network flow problem, the column generation problems are single-commodity minimum cost flow problem, but in the method in [5] the column generation problems are instead one-parametric side constrained single-commodity minimum cost flow problems; hence, the favorable column generation structure from the application of Dantzig–Wolfe decomposition is not inherited to the method in [5].

In contrast to the method in [5], our parametric procedure can be applied in situations where the column generation is a discrete (or mixed-integer) optimization problem. Further, our procedure preserves structure, in the sense that it employs only the column generation problem for the application at hand. An application of our procedure to the bi-objective multi-commodity minimum cost flow problem studied in [5], after making a scalarization of the two objectives, would therefore be much simpler than the method in [5].

2. Application to bi-objective discrete optimization

We here investigate how a one-parametric column-oriented linear optimization problem and the procedure stated above can be used to effectively and efficiently calculate a strong bounding frontier (that is, an outer approximation) of the Pareto frontier of a bi-objective discrete optimization problem with complicating side constraints. The bounding frontier is based on a scalarization and convexification of the problem, and typically it is tighter than the bounding frontier obtained from a linear programming relaxation of the bi-objective problem.

Let the vectors $c_1, c_2 \in \mathbb{R}^n$, the vector $b \in \mathbb{R}^m$, and the matrix $A \in \mathbb{R}^{m \times n}$. Further, let the set $X \subset \mathbb{R}_+^n$ be non-empty and finite, and consider the bi-objective discrete optimization problem

$$\begin{aligned} \min (z_1, z_2) &= (c_1^T x, c_2^T x) \\ \text{s.t. } Ax &= b, \\ x &\in X, \end{aligned}$$

which is assumed to have a feasible solution. It is presumed that it is possible to efficiently minimize a linear objective over the set X , such that it is the bi-objective nature of the problem and the explicit linear

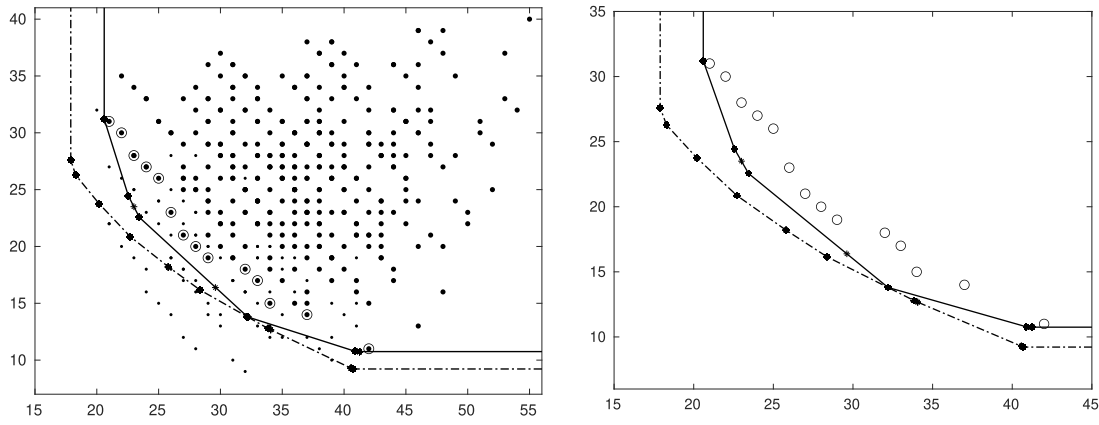


Fig. 2. To the left, the objective outcomes of the points in the set X (small dots) and of the feasible solutions (large dots), the Pareto optimal outcomes (circles), the bounding frontier obtained from the linear programming relaxation (dash-dotted), the bounding frontier obtained from the one-parametric column generation (solid), and the breakpoints of the two frontiers (asterisks). To the right, a magnification of the Pareto optimal outcomes and the two frontiers.

side constraints that make the problem challenging. Let Z be the set of feasible objective outcomes, that is,

$$Z = \left\{ (c_1^T x, c_2^T x) \mid Ax = b \text{ and } x \in X \right\}.$$

Let $P \subset \mathbb{R}^n$ be the set of Pareto optimal solutions and let $Z^* \subset \mathbb{R}^2$ be the set of Pareto optimal outcomes, that is, $Z^* = \left\{ (c_1^T x, c_2^T x) \mid x \in P \right\}$. Clearly, the sets Z , P , and Z^* are finite.

Consider the function $z^* : [0, 1] \mapsto \mathbb{R}$ defined by the scalarized problem

$$\begin{aligned} z^*(w) = \min \quad & wz_1 + (1-w)z_2 = \min [wc_1 + (1-w)c_2]^T x \\ \text{s.t.} \quad & (z_1, z_2) \in Z \quad \text{s.t.} \quad Ax = b, \\ & \quad \quad \quad \quad \quad \quad \quad \quad x \in X. \end{aligned}$$

Since Z is finite, z^* is clearly finite, piecewise linear and concave on its domain, with each piece corresponding to some $x \in X$. From the expression for z^* follows that $wz_1 + (1-w)z_2 \geq z^*(w)$ holds for all $(z_1, z_2) \in Z$. Since $Z^* \subseteq Z$, we may conclude that

$$Z^* \subset \left\{ (z_1, z_2) \mid wz_1 + (1-w)z_2 \geq z^*(w) \right\}, \quad w \in [0, 1].$$

Hence,

$$Z^* \subset \bigcap_{w \in [0,1]} \left\{ (z_1, z_2) \mid wz_1 + (1-w)z_2 \geq z^*(w) \right\}. \tag{9}$$

We next consider the function $z_c^* : [0, 1] \mapsto \mathbb{R}$ defined by the convexified scalarized problem

$$\begin{aligned} z_c^*(w) = \min \quad & [wc_1 + (1-w)c_2]^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \in \text{conv}(X). \end{aligned}$$

Since this is a linear program with a bounded feasible set, the function z_c^* is finite, piecewise linear and concave on its domain, with each piece corresponding to an extreme point of the feasible set. Define

$$Z_c = \bigcap_{w \in [0,1]} \left\{ (z_1, z_2) \mid wz_1 + (1-w)z_2 \geq z_c^*(w) \right\}.$$

The set can alternatively be expressed as

$$Z_c = \left\{ (c_1^T x, c_2^T x) \mid Ax = b \text{ and } x \in \text{conv}(X) \right\} + \mathbb{R}_+^2,$$

where the first term is a convex, polyhedral, and bounded set, which leads to the following result.

Proposition 6. *The set Z_c is convex and polyhedral.*

Using that $z_c^*(w) \leq z^*(w)$ holds for all $w \in [0, 1]$ and the inclusion (9), it follows that $Z_c \supset Z^*$. Let Z_c^* be the set of Pareto optimal outcomes within the set Z_c . Clearly, $Z_c^* = \text{bd}(Z_c)$. We have now arrived at the main result.

Proposition 7. *Each member of the set Z_c^* is non-dominated by each member of the set Z^* .*

We conclude that set Z_c^* constitute a bounding frontier for the Pareto frontier of the given bi-objective discrete optimization problem. Further, this frontier is the boundary of a polyhedral convex set.

Remark 8. The bounding property of the set Z_c^* is a close relative to the bounding property of a Lagrangian dual problem for a single-objective discrete optimization problem with complicating side constraints, which are Lagrangian relaxed.

Assuming that the set X consists of N points, denoted by $x^j, j = 1, \dots, N$, and letting $\mu_j, j = 1, \dots, N$, be convexity weights for these points, the convexified scalarized problem can be expressed as

$$\begin{aligned} z_c^*(w) = \min \quad & \sum_{j=1}^N [wc_1 + (1-w)c_2]^T x^j \mu_j \\ \text{s.t.} \quad & \sum_{j=1}^N Ax^j \mu_j = b, \\ & \sum_{j=1}^N \mu_j = 1, \\ & \mu_j \geq 0, \quad j = 1, \dots, N. \end{aligned}$$

This is a one-parametric column-oriented linear optimization problem that can be approached by Algorithm 1. Considering a fixed value of $w \in [0, 1]$ and denoting a dual solution with $(y^T, v)^T \in \mathbb{R}^{m+1}$, the column generation problem (2) here becomes

$$\min_{x \in X} \left\{ [wc_1 + (1-w)c_2 - A^T y]^T x - v \right\}, \tag{10}$$

that is, the minimization of a linear objective over the set X . (Recall that we assume that this minimization can be readily done.)

If the set X is described by integrality restrictions on variables and linear constraints, say $X = \left\{ x \in \mathbb{Z}_+^n \mid Dx \geq e \right\}$, then the Pareto frontier can alternatively be bounded by the linear programming frontier $Z_{LP}^* = \text{bd}(Z_{LP})$ where

$$Z_{LP} = \bigcap_{w \in [0,1]} \left\{ (z_1, z_2) \mid wz_1 + (1-w)z_2 \geq z_{LP}^*(w) \right\}$$

with

$$\begin{aligned} z_{LP}^*(w) = \min \quad & [wc_1 + (1-w)c_2]^T x \\ \text{s.t.} \quad & Ax = b, \\ & Dx \geq e, \\ & x \geq 0. \end{aligned}$$

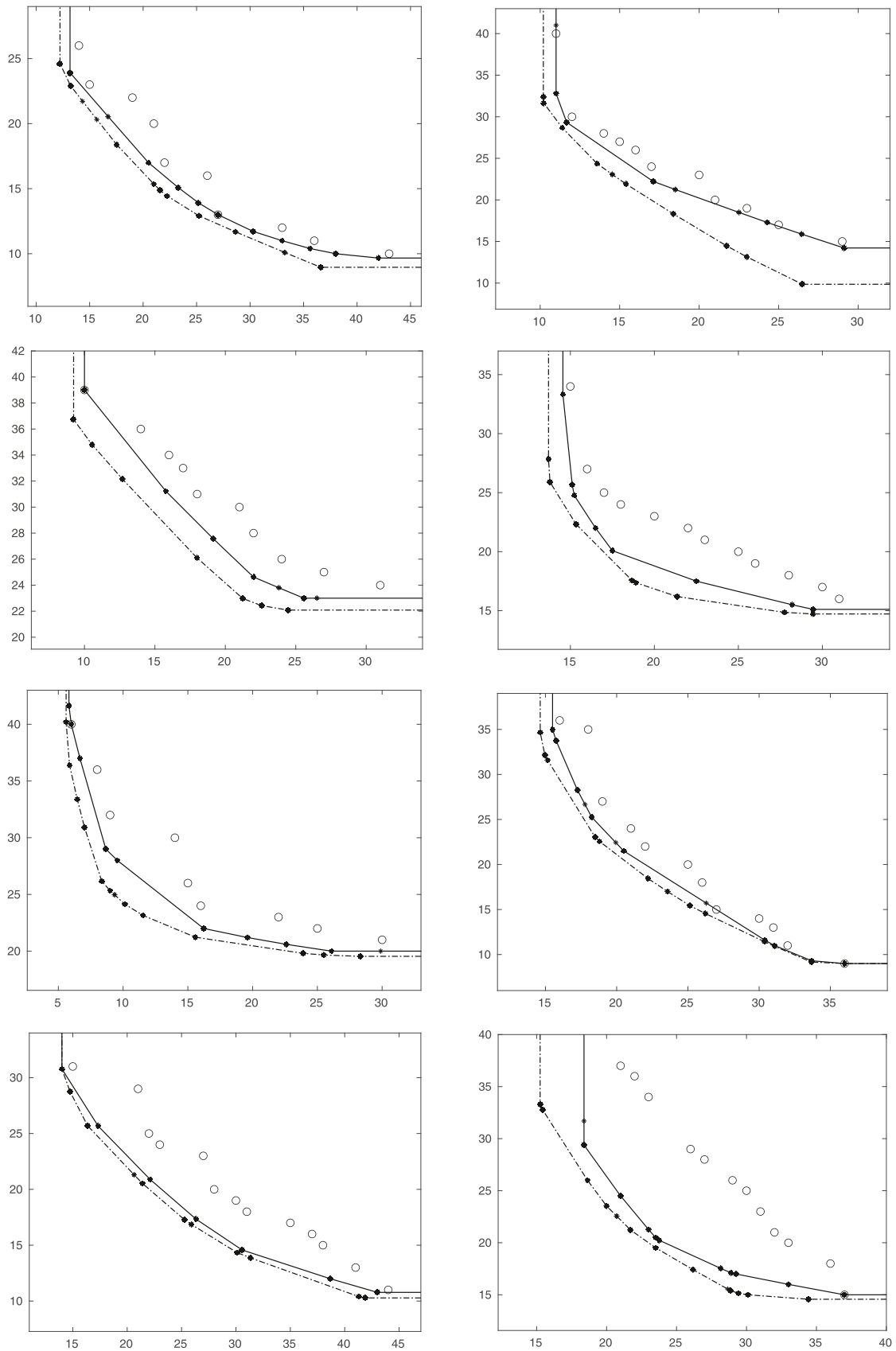


Fig. 3. Pareto optimal outcomes and the two frontiers for eight examples.

Since $\text{conv}\left(\left\{x \in \mathbb{Z}_+^n \mid Dx \geq e\right\}\right) \subseteq \left\{x \in \mathbb{R}_+^n \mid Dx \geq e\right\}$, it holds that $Z_c \subseteq Z_{LP}$. If the set $\left\{x \in \mathbb{R}_+^n \mid Dx \geq e\right\}$ has the integrality property (see e.g. [2]), then the column generation problem (10) can be solved as a linear program, $Z_c = Z_{LP}$ holds, and hence the bounding frontier Z_c^* coincides with that obtained from the linear programming relaxation of the scalarized bi-objective problem. Otherwise the column generation (10) is a genuine integer program and the bounding frontier Z_c^* will in general be tighter than that from the linear programming relaxation.

Example 9. To illustrate the bounding of the Pareto frontier, we use a small numerical zero–one example.

$$\begin{aligned} \min \quad & (z_1 = 7x_1 + 9x_2 + 4x_3 + 8x_4 + x_5 + 9x_6 + 7x_7 + 6x_8 + 2x_9 + 2x_{10}, \\ & z_2 = x_1 + x_2 + 9x_3 + 2x_4 + 8x_5 + x_6 + 3x_7 + 5x_8 + 7x_9 + 3x_{10}) \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 7x_3 + 5x_4 + 9x_5 + 4x_6 + 4x_7 + 2x_8 + 3x_9 + 2x_{10} \geq 26, \\ & x_1 + 9x_2 + 7x_3 + 5x_4 + 2x_5 + 7x_6 + 5x_7 + 8x_8 + 7x_9 + x_{10} \geq 28, \quad (11a) \\ & x \in \{0, 1\}^{10} \quad (11b) \end{aligned}$$

The set X is here defined by (11a) and (11b). It contains 457 points, and 337 out of these are feasible. These give 232 distinct objective outcomes, out of which 14 are Pareto optimal. Fig. 2 shows to the left the objective outcomes of the points in the set X and of the feasible solutions, the Pareto optimal outcomes, and the bounding frontiers obtained from the linear programming relaxation and from the one-parametric column generation. To the right is a magnification of only the Pareto optimal outcomes and the two frontiers. Column generation is here made by solving a knapsack problem, which does not have the integrality property, and therefore the column generation provides a tighter bounding frontier. This frontier has eight breakpoints, each corresponding to a knapsack solution.

Fig. 3 shows results for eight more problems of the same structure as in Example 9. The problem data are randomly generated and the results illustrate that the strengths of the bounding frontiers of course vary between problem instances. As can be seen, the one-parametric column generation frontier is sometimes noticeable stronger than the linear programming relaxation frontier, while it is sometimes only slightly stronger. Further, the frontiers can be close to the Pareto optimal outcomes or farther away.

An obvious potential application of the above described bounding of the Pareto frontier is in situations when it is prohibitively computationally expensive to calculate the Pareto frontier, but it is possible to use meta-heuristics (for example population-based) for finding a set of feasible solutions and resulting objective outcomes that are mutually non-dominated. The bounding frontier can then be used to assess the quality of the set of mutually non-dominated solutions.

3. Summary

We have showed how to effectively and efficiently perform a one-parametric analysis of column-oriented linear programs. This topic has, quite surprisingly, not been studied previously. The proposed scheme relies on the standard column generation problem for the linear program at hand, and Newton’s method for finding the largest zero of a univariate function. The latter technique is inspired by results from fractional optimization.

To exemplify how a one-parametric column-oriented linear program may arise, we consider a scalarized strong reformulation of a bi-objective discrete optimization problem. The outcome of this reformulation and the one-parametric analysis is a strong bounding of the Pareto frontier. An interesting topic for future research is to apply this approach to some class of computationally challenging bi-objective discrete optimization problems, for example for assessing the quality of a set of mutually non-dominated solutions found by a population-based meta-heuristic.

CRedit authorship contribution statement

Torbjörn Larsson: Conceptualization, Formal analysis, Visualization, Writing – original draft, Writing – review & editing. **Nils-Hassan Quttineh:** Conceptualization, Formal analysis, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data for the numerical example is given in the paper.

References

- [1] Murty KG. Linear programming. New York, NY: John Wiley & Sons; 1983.
- [2] Wolsey LA. Integer programming. New York, NY: John Wiley & Sons; 1998.
- [3] Radzik T. Fractional combinatorial optimization. In: Du D-Z, Pardalos PM, editors. Handbook of combinatorial optimization, vol. 1. Kluwer Academic Publishers; 1998, p. 429–78.
- [4] Borrero JS, Gillen C, Prokopyev OA. Fractional 0-1 programming: applications and algorithms. J Global Optim 2017;69(1):255–82.
- [5] Moradi S, Raith A, Ehrgott M. A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem. European J Oper Res 2015;244(2):369–78.