Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle

Nelson Maculan
Michel Minoux
Gérard Plateau

A $O(n)$ algorithm for projecting a vector on the intersection of a hyperplane and R^n_+

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 31, nº 1 (1997), p. 7-16.

<http://www.numdam.org/item?id=RO_1997__31_1_7_0>

Recherche opérationnelle/Operations Research
(vol. 31, n° 1, 1997, pp. 7-16)

A $O(n)$ ALGORITHM FOR PROJECTING A VECTOR ON THE INTERSECTION OF A HYPERPLANE AND R_+^n (*)

by Nelson MACULAN (1), Michel MINOUX (2) and Gérard PLATEAU (3)

Abstract. – *We present a $O(n)$ time algorithm for the projection of a vector on the intersection of a hyperplane and R_+^n. A linear-time median-finding algorithm is used to determine the median of the components of the vector to be projected. This extends a previous result of Maculan and Paula Jr. concerning the projection on the n-dimensional simplex.*

Keywords: Projection of a vector, computational linear algebra, subgradient methods.

Résumé. – *Cet article présente un algorithme de complexité linéaire pour déterminer la projection d'un vecteur de $\mathbb{R}^n$ sur l'intersection d'un hyperplan et de l'orthant positif. Un algorithme de recherche de médiane en temps linéaire est utilisé pour calculer la médiane des composantes du vecteur à projeter. Ceci généralise donc un résultat antérieur de Maculan et Paula Jr pour le cas de la projection sur le simplexe de $\mathbb{R}^n$.*

Mots clés : Projection d'un vecteur, algorithmes d'algèbre linéaire, méthodes de sous-gradient.

1. INTRODUCTION

Given a vector $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)^T \in R^n$ we would like to compute its projection on $X = \{x \in R^n \mid a^T x = b,\ x \geq 0\}$ where $a = (a_1, a_2, \ldots, a_n)^T \in R^n$ and $b \in R$ are given, and $a \neq 0$.

We suppose $X \neq \phi$, that is, if $b > 0$ (*respectively* $b < 0$) there exists j such that $a_j > 0$ (*respectively* $a_j < 0$).

This paper extends the results presented in [4] for the special case where X is the n-dimensional simplex, *i.e.,* for $b = 1$ and $a = (1, 1, \ldots, 1)^T$.

(*) Received November 1993.

(1) Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação, P.O. Box 68511, Rio de Janeiro, RJ 21945-970, Brazil. e-mail: maculan@cos.ufrj.br

(2) Laboratoire MASI, Université Pierre-et-Marie-Curie, 4, place Jussieu, 75252 Paris Cedex 05, France. e-mail: minoux@masi.ibp.fr

(3) LIPN, URA CNRS 1507 Institut Galilée, Université Paris-XIII, avenue Jean-Baptiste-Clément, 93430 Villetaneuse, France. e-mail: gerard.plateau@ura1507.univ-paris13.fr

Recherche opérationnelle/Operations Research, 0399-0559/97/01/$ 7.00

A different approach was proposed in [2] but without detailed complexity analysis.

In section 2 we formulate this projection as a quadratic programming problem and using the Karush-Kuhn-Tucker optimality conditions we state some results which will be used in section 3, where we present a $O(n)$ time complexity algorithm. We conclude in section 4.

2. DEFINITIONS AND RESULTS

2.1. Problem Formulation

The projection of $\overline{x}$ on X can be stated as the following quadratic programming problem:

$$(P): \quad \text{minimize } \frac{1}{2} \sum_{j=1}^{n} (x_j - \overline{x}_j)^2 \tag{1}$$

subject to

$$\sum_{j=1}^{n} a_j x_j = b \tag{2}$$

$$x_j \geq 0, \quad j = 1, \ldots, n. \tag{3}$$

Associating the dual variables $\alpha \in R$ and $w_j \in R_+$, $j = 1, \ldots, n$, with the constraints (2) and (3) respectively, we write the Lagrangean of (P) as follows:

$$\mathcal{L}(x, \alpha, w) = \frac{1}{2} \sum_{j=1}^{n} (x_j - \overline{x}_j)^2 + \alpha \left(\sum_{j=1}^{n} a_j x_j - b \right) - \sum_{j=1}^{n} w_j x_j.$$

It is easy to verify that

$$\frac{\partial \mathcal{L}}{\partial x_j} = x_j - \overline{x}_j + \alpha a_j - w_j.$$

The Karush-Kuhn-Tucker optimality conditions (*see* e.g. [3]) are necessary and sufficient for (P) and can be stated as:

$$x_j - \overline{x}_j + \alpha a_j - w_j = 0, \qquad j = 1, \ldots, n \tag{4}$$

$$x_j w_j = 0, \qquad j = 1, \ldots, n \tag{5}$$

$$x_j \geq 0, \qquad j = 1, \ldots, n \tag{6}$$

$$w_j \geq 0, \qquad j = 1, \ldots, n \tag{7}$$

$$\sum_{j=1}^{n} a_j x_j = b. \tag{8}$$

We can take x_j and w_j as a function of α, satisfying (4) to (7) in the following way:

$$x_j(\alpha) = \max\{\overline{x}_j - \alpha\, a_j,\, 0\},$$

$$w_j(\alpha) = \max\{\alpha\, a_j - \overline{x}_j,\, 0\}.$$

We define $\varphi(\alpha) = \sum_{j=1}^{n} a_j\, x_j(\alpha)$, $\alpha \in R$ as a real function. It is easy to check that φ is a piecewise linear function.

We would like to compute α^* such that $\varphi(\alpha^*) = b$, then the corresponding $x(\alpha^*)$ is the desired projection of $\overline{x}$ on X.

2.2. Properties of φ

The following results are easy to prove.

PROPERTY 1: $x_j(\alpha^*) = \max\{\overline{x}_j,\, 0\}$ *for all j such that* $a_j = 0$.

Let us denote $J = \{j \mid a_j \neq 0,\ j = 1, 2, \ldots, n\}$ and consider for $\alpha \in R$, $\varphi^+(\alpha) = \sum_{j \in J_+} a_j\, x_j(\alpha)$, and $\varphi^-(\alpha) = \sum_{j \in J_-} a_j\, x_j(\alpha)$, where $J_+ = \{j \in J \mid a_j > 0\}$ and $J_- = \{j \in J \mid a_j < 0\}$.

PROPERTY 2 (*see* figure 1): (i) φ^+ *is a convex decreasing piecewise linear function with breakpoints* $\alpha_j = \frac{\overline{x}_j}{a_j}$, $\forall j \in J_+$; (ii) φ^- *is a concave decreasing piecewise linear function with breakpoints* $\alpha_j = \frac{\overline{x}_j}{a_j}$, $\forall j \in J_-$; *and* (iii) $\varphi = \varphi^+ + \varphi^-$ *is a decreasing piecewise linear function with breakpoints* $\alpha_j = \frac{\overline{x}_j}{a_j}$, $\forall j \in J$.

Let us denote for $\alpha \in R$:

$$U^+(\alpha) = \left\{ j \in J_+ \mid \frac{\overline{x}_j}{a_j} > \alpha \right\}$$

and

$$L^-(\alpha) = \left\{ j \in J_- \mid \frac{\overline{x}_j}{a_j} < \alpha \right\},$$

it is clear that

$$\varphi^+(\alpha) = \sum_{j \in U^+(\alpha)} a_j\, (\overline{x}_j - \alpha\, a_j)$$

and

$$\varphi^-(\alpha) = \sum_{j \in L^-(\alpha)} a_j\, (\overline{x}_j - \alpha\, a_j).$$

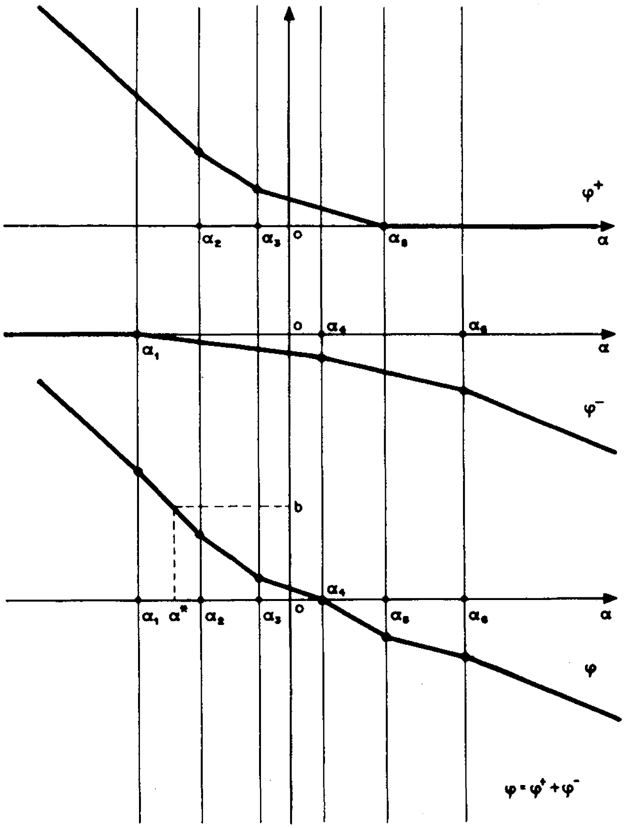

Figure 1.

This implies the following updating property of function φ:

PROPERTY 3: *Given $\varphi^+(\alpha)$ and $\varphi^-(\alpha)$ computed for a real number α, the computation of $\varphi(\alpha')$ for a distinct real number α' may be achieved in time complexity:*

$$O(|L^-(\alpha') - L^-(\alpha)| + \min\{|U^+(\alpha) - U^+(\alpha')|, |U^+(\alpha')|\}) \text{ if } \alpha' > \alpha$$

or

$$O(|U^+(\alpha') - U^+(\alpha)| + \min\{|L^-(\alpha) - L^-(\alpha')|, |L^-(\alpha')|\}) \quad \text{if } \alpha' < \alpha$$

where $|(\cdot)|$ denote the cardinality of $(\cdot)$.

Proof: The result is only checked for $\alpha' > \alpha$ (similar proof for the other case). As we assume to have computed $\varphi^+(\alpha)$ and $\varphi^-(\alpha)$, the following values are known:

$$p^+(\alpha) = \sum_{j \in U^+(\alpha)} a_j \bar{x}_j, \qquad q^+(\alpha) = \sum_{j \in U^+(\alpha)} a_j^2,$$

$$p^-(\alpha) = \sum_{j \in L^-(\alpha)} a_j \bar{x}_j, \qquad q^-(\alpha) = \sum_{j \in L^-(\alpha)} a_j^2$$

where

$$\varphi^+(\alpha) = p^+(\alpha) - \alpha q^+(\alpha) \qquad \text{and} \qquad \varphi^-(\alpha) = p^-(\alpha) - \alpha\, q^-(\alpha).$$

Therefore, the result is proved by noting that:

$$p^+(\alpha') = \begin{cases} p^+(\alpha) - \displaystyle\sum_{j \in U^+(\alpha) - U^+(\alpha')} a_j \bar{x}_j \\ \text{or} \\ \displaystyle\sum_{j \in U^+(\alpha')} a_j \bar{x}_j \end{cases}$$

$$q^+(\alpha') = \begin{cases} q^+(\alpha) - \displaystyle\sum_{j \in U^+(\alpha) - U^+(\alpha')} a_j^2 \\ \text{or} \\ \displaystyle\sum_{j \in U^+(\alpha')} a_j^2 \end{cases}$$

$$p^-(\alpha') = p^-(\alpha) + \sum_{j \in L^-(\alpha') - L^-(\alpha)} a_j \bar{x}_j$$

and

$$q^-(\alpha') = q^-(\alpha) + \sum_{j \in L^-(\alpha') - L^-(\alpha)} a_j^2$$

thus

$$\varphi(\alpha') = \varphi^+(\alpha') + \varphi^-(\alpha') = [p^+(\alpha') + p^-(\alpha')] - \alpha' [q^+(\alpha') + q^-(\alpha')]. \quad \square$$

Without loss of generality we will assume $b \geq 0$ in the next sections (if $b < 0$, equality (2) has to be multiplied by -1).

Contrary to the special case of projection on the n-dimensional simplex, the function φ may not be strictly decreasing as we show in the following example.

Given $X = \{x = (x_1, x_2)^T \mid 3x_1 - x_2 = b \geq 0\}$ and $\overline{x} = (-6, -1)^T$, we would like to find the projection of $\overline{x}$ on X. We compute the two breakpoints $\alpha_1 = \frac{-6}{3} = -2$ and $\alpha_2 = \frac{-1}{-1} = 1$; and we have $\varphi^+(\alpha) = 3\max\{-6 - 3\alpha, 0\}$ and $\varphi^-(\alpha) = -\max\{-1 + \alpha, 0\}$, that is,

$$\varphi^+(\alpha) = \begin{cases} -18 - 9\alpha & \text{if } \alpha \leq -2 \\ 0 & \text{if } \alpha \geq -2 \end{cases}$$

and

$$\varphi^-(\alpha) = \begin{cases} 0 & \text{if } \alpha \leq 1 \\ 1 - \alpha & \text{if } \alpha \geq 1 \end{cases}$$

then

$$\varphi(\alpha) = \varphi^+(\alpha) + \varphi^-(\alpha) = \begin{cases} -18 - 9\alpha & \text{if } \alpha \leq -2 \\ 0 & \text{if } -2 \leq \alpha \leq 1 \\ 1 - \alpha & \text{if } \alpha \geq 1 \end{cases}$$

This is illustrated in figure 2.

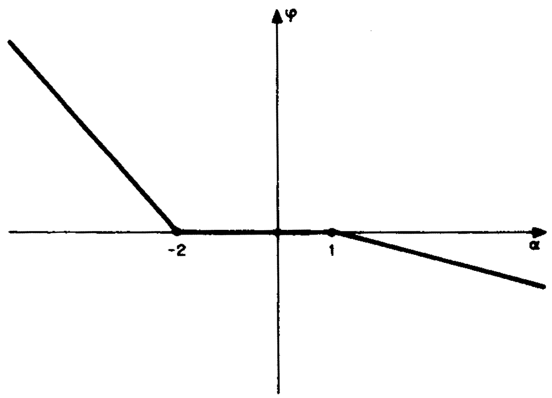

Figure 2.

For $b = 0$, to compute α^* such that $\varphi(\alpha^*) = 0$, we will not have unicity for α^*, because all $\alpha^* \in [-2, 1]$ solve $\varphi(\alpha^*) = 0$, in this case $x_1(\alpha^*) = x_2(\alpha^*) = 0$.

2.3. A $O(n \log n)$ Algorithm

Thanks to the properties of φ, a classical procedure for finding α^* such that $\varphi(\alpha^*) = b \geq 0$ may be the following one.

ALGORITHM VEPROJECT

Sort $\alpha_j = \frac{\overline{x}_j}{a_j}$, $j \in J$ in increasing order {the data are assumed to be renumbered, so that $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_{|J|}$};

if $b > \varphi(\alpha_1)$ *then* find $\alpha^* \in]-\infty, \alpha_1[$ {case 1};

else

find $k \in \{1, \ldots, |J| - 1\}$ such that $\varphi(\alpha_{k+1}) \leq b \leq \varphi(\alpha_k)$;

find $\alpha^* \in [\alpha_k, \alpha_{k+1}]$ {case 2};

endif

THEOREM 1: *Algorithm VEPROJECT solves* (P) *in* $O(n \log n)$ *time complexity.*

Proof: We note that α^* is obtained by using a linear extrapolation (case 1) or by a linear interpolation (case 2), and by observing properties 2 and 3 for finding k in the second case, the if-statement requires a $O(|J|)$ time complexity.

Thus, due to the $O(|J| \log |J|)$ time complexity for the initial ordering of the α_j, $j \in J$, the **veproject** algorithm solves (P) in $O(n \log n)$ time ($|J| \leq n$ and $x(\alpha^*)$ is computed in $O(n)$ time). □

Remark: With the assumption $b \geq 0$, $\varphi(\alpha_{|J|})$ never exceeds b since, for all $\alpha \geq \alpha_{|J|}$ we have $\varphi^+(\alpha) = 0$ and $\varphi^-(\alpha) \leq 0$; then, for all $\alpha \geq \alpha_{|J|}$, $\varphi(\alpha) \leq 0$.

3. THE LINEAR TIME ALGORITHM LIVEPROJECT

We present a linear-time complexity algorithm –denoted by LIVE-PROJECT– whose goal is to search the projection $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T$ of a given point $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)^T$ on $X = \{x \in R^n \,|\, a^T x = b,\ x \geq 0\}$ where $a = (a_1, a_2, \ldots, a_n)^T \in R_*^n$ and $b \in R_+$ are given. For each set of indices denoted by J, let us denote $J_- = \{j \in J \,|\, a_j < 0\}$, $J_+ = \{j \in J \,|\, a_j > 0\}$ and $J_0 = \{j \in J \,|\, a_j = 0\}$.

ALGORITHM LIVEPROJECT

$J := \{1, 2, \ldots, n\}$;

if $b > 0$ and $J_+ = \phi$ *then* return {the problem has no solution};

else

for each $j \in J_0$ do $x_j^* := \max\{\overline{x}_j, 0\}$ *endfor;*

for each $j \in J_+ \cup J_-$ *do* $\alpha_j := \frac{\overline{x}_j}{a_j}$ *endfor;*

if max $\{\alpha_j \mid j \in J_+\} <$ min $\{a_j \mid j \in J_-\}$ *then*
for each $j \in J_+ \cup J_-$ *do* $x_j^* := 0$ *endfor;*
return (x^*);
else
$p^+ := 0;\ q^+ := 0;\ p^- := 0;\ q^- := 0;$
$\sigma := 0$ {value of φ for the current value of $\alpha \in R+$, *i.e.*,
$\sigma = \varphi(\alpha) = (p^+ + p^-) - \alpha(q^+ + q^-)$ where $p^+ - \alpha q^+ = \varphi^+(\alpha)$
and $p^- - \alpha q^- = \varphi^-(\alpha)\}$;
$S :=$ list of α_j for all $j \in J_+ \cup J_-$;
while $|S| \geq 3$ and $\sigma \neq b$ *do*
$\alpha_m :=$ median of the list S; $\{\alpha_m$ is obtained in time $O(|S|)\}$
$L := \{j \in J_+ \cup J_- \mid \alpha_j < \alpha_m\}$;
$E := \{j \in J_+ \cup J_- \mid \alpha_j = \alpha_m\}$;
$U := \{j \in J_+ \cup J_- \mid \alpha_j > \alpha_m\}$;
$p_1^+ := \sum_{j \in U \cap J_+} a_j \overline{x}_j;\ q_1^+ := \sum_{j \in U \cap J_+} a_j^2;$
$p_1^- := \sum_{j \in L \cap J_-} a_j \overline{x}_j;\ q_1^- := \sum_{j \in U \cap J_-} a_j^2;$
$\sigma := (p^+ + p_1^+ + q^+ + q_1^+) - \alpha_m (p^- + p_1^- + q^- + q_1^-);$
if $\sigma > b$ *then*
$p^- := p^- + p_1^- + \sum_{j \in E \cap J_-} a_j \overline{x}_j;$
$q^- := q^- + q_1^- + \sum_{j \in E \cap J_-} a_j^2;$
$J := U \cup \{m\};$
$S :=$ list of α_j for all $j \in J$;
else
if $\sigma < b$ *then*
$p^+ := p^+ + p_1^+ + \sum_{j \in E \cap J_+} a_j \overline{x}_j;$
$q^+ := q^+ + q_1^+ + \sum_{j \in E \cap J_+} a_j^2;$
$J := L \cup \{m\};$
$S :=$ list of α_j for all $j \in J$;
else $\{\sigma = b\}$
$p^+ := p^+ + p_1^+;\ q^+ := q^+ + q_1^+;$
$p^- := p^- + p_1^-;\ q^- := q^- + q_1^-;$
endif
endif

endwhile;

if $\sigma = b$ *then* $\alpha^* := \alpha_m$

else

$\{\alpha_m \in S$, and when $|S| = 2$ the list S contains two distinct elements$\}$

$\alpha^* := \frac{p^+ + p^- - b}{q^+ + q^-}$

endif;

for each $j \in \{1, 2, \ldots, n\}$ such that $a_j \neq 0$ *do*

$x_j^* := \max\{\bar{x}_j - \alpha^* a_j, 0\}$

endfor;

return (x^*)

endif

endif

THEOREM 2: *The LIVEPROJECT algorithm has linear-time complexity* $O(n)$.

Proof: The proof given in [4] for the projection on the n-dimensional simplex easily extends the more general problem studied here. We briefly sketch the proof for completeness. The number of iterations of LIVEPROJECT algorithm is bounded by $\lceil \log_2 n \rceil$. For each iteration k –denoting S_k the generic list S– the search for the median of the list S_k is achieved by a $O(|S_k|)$ time complexity algorithm ([1]). The time complexity of the remaining operations (partition of S_k, updating the function φ) is obviously $O(|S_k|)$. In particular, these operations remove at least $\left\lfloor \frac{|S_k|}{2} \right\rfloor$ elements from the current set of indices J.

Note that $\left\lfloor \frac{|S_k|}{2} \right\rfloor \geq \left\lfloor \frac{|S_k|}{3} \right\rfloor$ provided that $|S_k| \geq 2$ (such a coarse lower bounding is enough to get the result). Therefore at iteration $k + 1$ we have $|S_{k+1}| \leq |S_k| - \frac{|S_k|}{3} = \frac{2}{3}|S_k|$.

By noting that $|S_1| = n$, and x^* is computed in $O(n)$ time complexity, the global time complexity of LIVEPROJECT algorithm is

$$O\left(\sum_{k=1}^{\lceil \log_2 n \rceil} |S_k|\right) = O\left(\sum_{k=1}^{\lceil \log_2 n \rceil} \left(\frac{2}{3}\right)^{k-1} n\right) = O(n). \quad \square$$

4. CONCLUSIONS

The algorithm proposed here to find the projection of a vector on a hyperplane in R^n_+ can be used to solve subproblems arising in some iterative optimization methods, for example, to find the solution, through subgradient techniques, of some dual problems obtained by Lagrangean relaxation in combinatorial optimization.

REFERENCES

1. A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms. *Addison-Wesley Publishing Company,* 1974.
2. C. Michelot, A Finite Algorithm for Finding the Projection of a Point onto the Canonical Simplex of R^n. *Journal of Optimization Theory and Applications,* 50(1), 1986, pp. 195-200.
3. M. Minoux, Mathematical Programing. *John Wiley and Sons*, 1986.
4. N. Maculan and G. G. de Paula Jr., A Linear-Time Median-Finding Algorithm for Projecting a Vector on the Simplex of R^n. *Operations Research Letters,* 8 (4), 1989, pp. 219-222.