Column generation algorithms for nonlinear optimization, I: Convergence analysis

Ricardo García*, Angel Marín†, and Michael Patriksson‡

October 10, 2002

Abstract

Column generation is an increasingly popular basic tool for the solution of large-scale mathematical programming problems. As problems being solved grow bigger, column generation may however become less efficient in its present form, where columns typically are not optimizing, and finding an optimal solution instead entails finding an optimal convex combination of a huge number of them. We present a class of column generation algorithms in which the columns defining the restricted master problem may be chosen to be optimizing in the limit, thereby reducing the total number of columns needed. This first paper is devoted to the convergence properties of the algorithm class, and includes global (asymptotic) convergence results for differentiable minimization, finite convergence results with respect to the optimal face and the optimal solution, and extensions of these results to variational inequality problems. An illustration of its possibilities is made on a nonlinear network flow model, contrasting its convergence characteristics to that of the restricted simplicial decomposition (RSD) algorithm.

Key words: Column generation, simplicial decomposition, inner approximation, simplices, column dropping, optimal face, non-degeneracy, pseudo-convex minimization, weak sharpness, finite convergence, pseudo-monotone$^+$ mapping, variational inequalities.

1 Introduction

1.1 Origin and use

Column generation (CG for short) is a classic and increasingly popular means to attack large-scale problems in mathematical programming. A column generation algorithm can, roughly, be said to proceed according to the following two steps, used iteratively in succession: (1) A relaxation of the original problem is constructed, based on current estimates of the optimal solution. Solving this problem provides a vector (i.e., the column), typically also a bound on the optimal value of the original problem, and the answer to an optimality test, indicating if the column identified will be beneficial to introduce into the solution, or if the process should be stopped, with the current solution being optimal. (2) A restricted master problem (RMP) is constructed, in which previously generated columns together with the new column are convex combined such that the resulting vector is feasible in the original problem and a measure of optimality is improved. The resulting solution is used to define a new column generation subproblem.

*E. U. Politécnica de Almadén, Universidad de Castilla–La Mancha, Plaza Manuel Meca, 1, 13 400 Almadén, Ciudad Real, Spain. Email: rgarcia@pol-al.uclm.es.

†E. T. S. I. Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, 28 040 Madrid, Spain. Email: amarin@dmae.upm.es.

‡Department of Mathematics, Chalmers University of Technology, S-412 96 Gothenburg, Sweden. Email: mipat@math.chalmers.se.

Depending on the problem type being addressed, the choice of relaxation, the resulting definition of the column, and the column generation and restricted master problems can be quite different.

In *linear programming* (LP), a subset of the linear constraints are Lagrangian relaxed with current dual multiplier estimates. The LP subproblem generates, as a column candidate, an extreme point in the polyhedron defined by the non-relaxed constraints; this step is reminiscent of the pricing-out step in the simplex method. The RMP optimally combines the extreme points known such that the relaxed constraints are satisfied, and the dual variables for those constraints are used to construct the next CG subproblem. For minimization problems, termination follows (like in the simplex method) when the reduced cost of the new column fails to be negative. This algorithm originates in [DaW60]; see also [Las70].

In linear *integer programming* (IP), column generation is most often based on Lagrangian relaxation, as the above described Dantzig–Wolfe decomposition algorithm. The subproblem is an integer programming problem while the RMP is formulated as a linear program; this is necessary, since column generation is here driven by Lagrange multipliers. This algorithm leads to the solution of a convexified version of the original IP, where the set defined by the non-relaxed constraints are replaced by its convex hull, the extreme points of which are identified in the CG problem. For a detailed description of the principles behind IP column generation, see Wolsey [Wol98]. In order to completely solve the original problem, column generation must be combined with an optimizing IP algorithm such as branch-and-bound (that is, branch-and-price algorithms, see [Bar+98, Wol98]).

In *nonlinear programming* (NLP), relaxation can act as linearization of nonlinear parts of the problem, as well as on the constraints as above. Nonlinear Dantzig–Wolfe decomposition ([Las70]) for convex programs extends the LP algorithm by letting the Lagrangian relaxed subproblem be nonlinear while the RMP still is an LP. In *simplicial decomposition* (SD), the relaxation is instead a linearization of the objective function. (Nonlinear constraints would be approximated by piece-wise linear functions.) The column generation problem therefore is an LP, while the objective function is retained in the RMP. The origin of SD is found in [Hol74, vHo77] for the linearly constrained case with theoretical contributions also in [HLV85, HLV87], and [HiP90, VeH93] for nonlinearly constrained problems. In SD, column generation is not based on pricing, or indeed on any dual information, but instead on the descent properties of the direction towards an extreme point of the feasible set; it is therefore more natural to associate SD with multidimensional extensions of primal descent algorithms in NLP. We however remark that SD does include classes of pricing-based column generation methods: as established in [LMP94], when applying SD to a primal–dual, saddle point, reformulation of an LP, the algorithm reduces to the Dantzig–Wolfe algorithm for that LP!

1.2 Common points among problem areas

Despite the diversity of the problem types, the column generation algorithms mentioned have some interesting points in common in both theory and practice.

The RMPs constructed in the algorithms are based on Carathéodory's Theorem (e.g., [Roc70, Thm. 17.1]), which states that a point $x \in \Re^n$ in the convex hull of any subset X of $\Re^n$ can be represented as a convex combination of at most as many elements of X as its dimension, $\mathtt{dim}\, X$ (which is defined as the dimension of its affine hull) plus one. As such, a CG algorithm induces a reformulation of the original problem in terms of an *inner representation* of the feasible set. This reformulation is stated in terms of convexity weight variables, which in general will be many more than in the original formulation. (For example, in both the Dantzig–Wolfe and SD algorithms for linearly constrained problems, there is one variable for each extreme point and direction, a number which grows exponentially with the number of variables and constraints in the original problem.)

For many applications, this problem transformation is of clear advantage. To this contributes that the inner representation of X is much simpler than its original, *outer*, representation, as it is the Cartesian product of a simplex and the non-negative orthant. Another, and perhaps even more important, fact is that an increase in dimension improves the formulation. For IP, there is a clear link between the use of many variables in a formulation and the strength of lower bounds from LP and Lagrangian relaxations (cf. [Wol98]). For structured LPs, it has been established that choosing a relaxation that leads to a further decomposition into smaller-dimensional problems, and ultimately to much larger equivalent reformulations, is of a clear computational advantage; see, for example, [JLF93]. For certain structured NLP problems, the conclusion has been similar: in the multi-commodity problem known as the traffic equilibrium model, for example, disaggregated versions of the restricted simplicial decomposition (RSD) algorithm ([LaP92, JTPR94]) are more efficient than in its aggregated form ([HLV87]).

A potential disadvantage with the current use of column generation is, however, that the column generation problems do not generate near-optimal solutions. In the Dantzig–Wolfe and SD algorithms, for example, the columns are extreme points, in the first case of a set a large portion of which are infeasible in the original problem, and in the second case none of the extreme points are optimal in general. The RMP is of course present as a remedy for this *non-coordinability* ([Las70]). What it means in terms of convergence is that in some circumstances, the number of columns needed in order to span an optimal solution is very high, and the number of main iterations must necessarily be at least as high. The efficient solution of a problem therefore demands that (a) the CG subproblem is rather easy to solve while keeping enough of the original problem's properties, and (b) the RMP can be solved efficiently even with a rather large number of variables. While the latter is usually true for LP and IP, it is not necessarily so for NLP. On the other hand, while the first may not be so easily accommodated in IP models, for LP and NLP models this is at least possible in principle through the use of penalties, regularizations, etc.

At least for mathematical programs over convex sets, we can describe an optimal solution by using much fewer columns. This is done by choosing them in the relative interior; this is achieved especially if the column generation problem is actually optimizing, an idea which we propose in this paper. Such ideas have been studied to a limited extent earlier for LP ([KiN91]) and NLP models ([LPR97, Pat98b]). The former methodology is Dantzig–Wolfe decomposition with subproblems solved with interior-point methods, while the latter is an extension of line-search based descent algorithms for NLP to *multidimensional searches*, in which the subproblems are strictly convex programs. The advantage of this scheme for NLP is that, in short, convex combining 100 extreme points is done by instead convex combining only ten points, each of which has been combined by ten extreme points. Since the column generation subproblem is more complex, whether there really is a computational advantage depends very much on the problem being solved, as evidenced in the computational tests in [LPR97]. Our proposal is more general than each of the two mentioned. There are also related attempts to improve decomposition algorithms by the use of non-extremal *cutting planes*, see, e.g., [GHV92].

In the following, we present the problem under study, and a general form of column generation algorithm.

2 A column generation method

We consider the following constrained differentiable optimization problem:

$$\underset{x \in X}{\text{minimize}}\ f(x), \qquad [\mathrm{CDP}(f, X)]$$

where $X \subseteq \Re^n$ is nonempty and convex, and $f : X \mapsto \Re^n$ is continuously differentiable on X. Its set of global minimizers is denoted $\mathrm{SOL}(f, X)$. Its first-order optimality conditions are then given by the variational inequality

$$-\nabla f(x^*) \in N_X(x^*), \qquad [\mathrm{VIP}(\nabla f, X)]$$

where

$$N_X(x) := \begin{cases} \{\, z \in \Re^n \mid z^{\mathrm{T}}(y-x) \leq 0, \quad \forall y \in X \,\}, & x \in X, \\ \emptyset, & x \notin X \end{cases} \tag{1}$$

denotes the normal cone to X at $x \in \Re^n$. Its solution set is denoted $\mathrm{SOL}(\nabla f, X)$. If f is pseudo-convex on X then $\mathrm{SOL}(f, X) = \mathrm{SOL}(\nabla f, X)$ holds.

2.1 The algorithm

We begin by stating and establishing the convergence of the general algorithm. The algorithm is described by means of (possibly point–to–set) algorithmic mappings which will be assumed to fulfill conditions similar to those utilized in the convergence analysis in [Zan69]. Consider the following conditions.

Assumption 2.1 (Closed descent algorithm). *Let $\widehat{X} \subseteq X$ be any non-empty and convex set, and $A : \widehat{X} \mapsto 2^{\widehat{X}}$ denote a point–to–set mapping on $\widehat{X}$. The mapping A then satisfies the following three conditions:*

(a) (Closedness). *The mapping A is closed on $\widehat{X}$, that is, for every $x \in \widehat{X}$,*

$$\left.\begin{array}{r} \{x^t\} \to x \\ y^t \in A(x^t),\ \{y^t\} \to y \end{array}\right\} \quad \Longrightarrow \quad y \in A(x).$$

(b) (Fixed point at solution). $x \in \mathrm{SOL}(\nabla f, \widehat{X}) \quad \Longleftrightarrow \quad x \in A(x)$.

(c) (Descent property at non-solution). *Let $x \in \widehat{X} \setminus \mathrm{SOL}(\nabla f, \widehat{X})$. Then,*

(1) $y \in A(x) \quad \Longrightarrow \quad \nabla f(x)^{\mathrm{T}}(y - x) < 0.$

(2) $y \in A(x) \quad \Longrightarrow \quad f(y) < f(x).$

In the column generation algorithm, the column generation problem is characterized by an iterative procedure, denoted by $\mathcal{A}_{\mathrm{c}}^k$ and belonging to a finite collection $\mathcal{K}_{\mathrm{c}}$. It is presumed to be of the form of A above, operating on $\widehat{X} := X$. Descent is reached in either of two ways, depending on whether the assumption (c)(1) or (c)(2) is in force. The assumption (c)(1) is associated with a procedure that provides a descent direction, and which is hence presumed to be applied once only from x. The assumption (c)(2) is associated with an algorithm that operates as a descent algorithm on the original problem. In order to rule out the uninteresting case that the original problem is solved by means of only using the column generation problem an infinite number of times starting from a given iterate $x \in X$, we presume that the number of iterations performed from x is finite.

Similarly, the restricted master problem is assumed to be solved by the use of an iterative procedure, denoted $\mathcal{A}_{\mathrm{r}}^k$ and belonging to a finite collection $\mathcal{K}_{\mathrm{r}}$. It is also presumed to be of the form of A above, but where (c)(2) is always in force, and it operates on $\widehat{X} \subset X$ being equal to the current inner approximation of X. Also this algorithm will be presumed to be applied a finite number of times; we can still solve any given RMP exactly in this way, by making the proper choice of the procedure $\mathcal{A}_{\mathrm{r}}^k$.

In Table 1, we summarize the different steps of the algorithm.

When establishing the convergence of this algorithm, we have made the choice to presume that X is bounded. We can easily find examples of algorithms in the framework which converge

Table 1: The conceptual algorithm

0. (*Initialization*): Choose an initial point $x^0 \in X$, and let $t := 0$.
1. (*Column generation problem*): Choose an algorithm $\mathcal{A}_{\mathrm{c}}^{k_t}$, $k_t \in \mathcal{K}_{\mathrm{c}}$. Apply it a finite number of iterations on $\mathrm{CDP}(f, X)$, starting from x^t; if it is of the form of the Assumption 2.1(c)(2), then do it once only. Let the resulting point be y^t.
2. (*Termination criterion*): If $x^t \in \mathrm{SOL}(\nabla f, X) \to$ Stop. Otherwise, continue.
3. (*Set augmentation*): Let $X^{t+1} \subset X$ be a non-empty, compact and convex set containing $[x^t, y^t]$.
4. (*Restricted master problem*): Choose an algorithm $\mathcal{A}_{\mathrm{r}}^{k_t}$, $k_t \in \mathcal{K}_{\mathrm{r}}$, of the form of the Assumption 2.1(c)(2). Apply it at least one iteration of the algorithm on $\mathrm{CDP}(f, X^{t+1})$, starting from x^t. Let the resulting point be x^{t+1}.
5. (*Update*): Let $t := t + 1$. Go to Step 1.

regardless of any boundedness assumptions (such as some regularization and proximal point algorithms), or with milder forms of boundedness conditions (such as the Frank–Wolfe and RSD algorithms). A good compromise has been difficult to find, however, as the algorithms and their convergence conditions are so different. What we have opted to do is instead to provide a discussion on the possibilities to strengthen the result in certain interesting cases; as the convergence proof reveals, the crucial point is to establish that the two sequences $\{x^t\}$ and $\{y^t\}$ are constructed such that they are bounded, so how this condition is ensured in different cases is the main point of the discussion.

2.2 The basic convergence result

The following global convergence result extends the one in [LMP94] for a general class of column generation methods, in that it allows for a more general definition of the restriction and a greater flexibility in the selection of algorithms in the steps 1. and 4. Its proof utilizes parts of the one in [LMP94] together with those in [Pat93b, Pat98a, Pat98b] for truncated descent algorithms with closed set-valued mappings.

Theorem 2.2 (Global convergence). *Let Assumption* 2.1 *hold, and assume that X is bounded. Then, the sequence $\{x^t\}$ of iterates converges to* $\mathrm{SOL}(\nabla f, X)$ *in the sense that*

$$\left\{ d_{\mathrm{SOL}(\nabla f, X)}(x^t) \right\} := \left\{ \underset{x \in \mathrm{SOL}(\nabla f, X)}{\operatorname{minimum}} \; \|x^t - x\|_2 \right\} \to 0.$$

Proof. If convergence is finite, then the last iterate is clearly a member of $\mathrm{SOL}(\nabla f, X)$. We assume henceforth that the sequence $\{x^t\}$ is infinite.

By construction, the sequence $\{f(x^t)\}$ descends; thus, $\{x^t\} \subset L_f^X(x^0)$, and is therefore bounded, since X is. Further, the sequence $\{X^t\}$ consists of non-empty, compact, and convex subsets of X, and is therefore also bounded. Thus, it has at least one accumulation set, $\widetilde{X} \subseteq X$ (in terms of set convergence; see [SaW79] and [RoW98, Chapter 4]), which is also non-empty (since $X^t \supseteq [x^{t-1}, y^{t-1}]$ for all t, and [RoW98, Thm. 4.18]), compact (since each set X^t is closed and $\{X^t\}$ is bounded, and [RoW98, Prop. 4.4]), and convex (since X^t is convex for all t, and [RoW98, Prop. 4.15]). The remainder of the proof concerns iterates in the subsequence defining the accumulation set $\widetilde{X}$, which (for simplicity of notation) will however not be stated explicitly.

Since the number of iterations is infinite and the sets $\mathcal{K}_c$ and $\mathcal{K}_r$ are finite, there will be at least one pair of elements in these sets, say (k_c, k_r), that appears an infinite number of times in the subsequence defining the set $\widetilde{X}$ and in the same iterations as well. We henceforth consider this further subsequence.

Since the sequence $\{x^t\}$ belongs to the compact set $L_f^X(x^0)$, it has a non-empty and bounded set of accumulation points, say $\overline{X} \subseteq L_f^X(x^0)$, which is closed (e.g., [Rud76, Thm. 3.7]). Since f is continuous, we may find a convergent infinite subsequence, say $\{x^t\}_{t\in\mathcal{T}}$, where $\mathcal{T} \subseteq \{0,1,2,\dots\}$, with limit point $x^{\mathcal{T}} \in \arg\max_{x\in\overline{X}} f(x)$. Further, $x^{\mathcal{T}} \in \widetilde{X}$ (e.g., [AuF90, Prop. 1.1.2]).

Denote by $z^{t-1} \in X^t$, $t \in \mathcal{T}$, the first iterate of the algorithm $\mathcal{A}_r^{k_r}$ applied to the restricted problem CDP(f, X^t), starting from $x^{t-1} \in X^t$. Since each iteration of this algorithm gives descent with respect to f, unless a stationary point to the RMP, CDP(f, X^t) is at hand, it follows that, for all $t \in \mathcal{T}$, $f(x^t) \le f(z^{t-1}) < f(x^{t-1})$. Let $x^{\mathcal{T}-1} \in \overline{X}$ be the limit point of a convergent infinite subsequence of the sequence $\{x^{t-1}\}_{t\in\mathcal{T}}$ and let $z^{\mathcal{T}-1} \in \widetilde{X}$ be an accumulation point of the corresponding subsequence of the sequence $\{z^{t-1}\}_{t\in\mathcal{T}}$. Taking the limit corresponding to this accumulation point, the continuity of f yields that $f(x^{\mathcal{T}}) \le f(z^{\mathcal{T}-1}) \le f(x^{\mathcal{T}-1})$.

Since $x^{\mathcal{T}-1} \in \overline{X}$, the definition of $x^{\mathcal{T}}$ implies that $f(x^{\mathcal{T}}) \ge f(x^{\mathcal{T}-1})$, and we conclude that $f(x^{\mathcal{T}}) = f(z^{\mathcal{T}-1}) = f(x^{\mathcal{T}-1})$. The latter equality together with the closedness and descent properties of the iteration mapping of the algorithm $\mathcal{A}_r^{k_r}$ at non-stationary solutions gives that $x^{\mathcal{T}-1} \in \mathrm{SOL}(\nabla f, \widetilde{X})$. Then, from the relation $f(x^{\mathcal{T}}) = f(x^{\mathcal{T}-1})$ and the definition of $x^{\mathcal{T}}$, we obtain that for all $x \in \widetilde{X}$, $f(x) \ge f(x^{\mathcal{T}})$, and that for all $x \in \overline{X}$, $f(x) = f(x^{\mathcal{T}})$. Hence, $\overline{X} \subseteq \mathrm{SOL}(\nabla f, \widetilde{X})$.

Now, let $\varepsilon \ge 0$ be such that there is an infinite number of iterates x^{t-1} with $d_{\mathrm{SOL}(\nabla f, X)}(x^{t-1}) \ge \varepsilon$. This infinite subsequence of iterates has some accumulation point, say $\widetilde{x}$, which is then the limit point of some infinite convergent sequence $\{x^{t-1}\}_{t\in\widetilde{\mathcal{T}}}$, where $\widetilde{\mathcal{T}} \subseteq \{0,1,2,\dots\}$. From the above we then know that $\widetilde{x} \in \mathrm{SOL}(\nabla f, \widetilde{X})$.

The sequence $\{y^t\} \subseteq X$, and is therefore bounded.

We first assume that the algorithm $\mathcal{A}_c^{k_c}$ is of type $(c)(1)$. Let $\widetilde{y}$ be an arbitrary accumulation point of the subsequence $\{y^{t-1}\}_{t\in\widetilde{\mathcal{T}}}$. Since $y^{t-1} \in X^t$ for all $t \in \widetilde{\mathcal{T}}$, $\widetilde{y} \in \widetilde{X}$ holds (e.g., [AuF90, Prop. 1.1.2]). Since $\widetilde{x} \in \mathrm{SOL}(\nabla f, \widetilde{X})$, we then have that $\nabla f(\widetilde{x})^{\mathrm{T}}(\widetilde{y} - \widetilde{x}) \ge 0$ holds. But if $\widetilde{x} \notin \mathrm{SOL}(\nabla f, X)$ holds, then we obtain from the closedness and descent properties of the algorithm $\mathcal{A}_c^{k_c}$ that $\nabla f(\widetilde{x})^{\mathrm{T}}(\widetilde{y} - \widetilde{x}) < 0$ holds, which yields a contradiction. Hence, $\widetilde{x} \in \mathrm{SOL}\,(\nabla f, X)$.

We next assume that the algorithm $\mathcal{A}_c^{k_c}$ is of type $(c)(2)$. Since the sequence $\{y^{t-1}\}_{t\in\widetilde{\mathcal{T}}} \subseteq X$, it has a non-empty, bounded and closed set of accumulation points, say $\widetilde{Y}$. We define herein a convergent infinite subsequence with limit point $y^{\widetilde{\mathcal{T}}-1} \in \arg\max_{y\in\widetilde{Y}} f(y)$.

For all $t \in \widetilde{\mathcal{T}}$, let $v^{t-1} \in X$ denote the point obtained by performing one iteration with the algorithm $\mathcal{A}_c^{k_c}$ on the problem CDP(f, X), starting from x^{t-1}. Since each iteration of the algorithm $\mathcal{A}_c^{k_c}$ gives descent with respect to f, unless the current iterate is in $\mathrm{SOL}(\nabla f, X)$, it follows that, for all $t \in \widetilde{\mathcal{T}}$, $f(y^{t-1}) \le f(v^{t-1}) < f(x^{t-1}) < f(y^{t-2})$, where the latter strict inequality stems from the descent properties of the algorithm applied to the previous RMP. Taking the limit corresponding to the above defined subsequence, the continuity of f yields that $f(y^{\widetilde{\mathcal{T}}-1}) \le f(\widetilde{x}) \le f(v^{\widetilde{\mathcal{T}}-1}) \le f(y^{\widetilde{\mathcal{T}}-2})$, where $y^{\widetilde{\mathcal{T}}-2}$ denotes an accumulation point of the sequence $\{y^{t-2}\}_{t\in\widetilde{\mathcal{T}}}$.

Since $y^{\widetilde{\mathcal{T}}-2} \in \widetilde{Y}$, the definition of $y^{\widetilde{\mathcal{T}}-1}$ implies that $f(y^{\widetilde{\mathcal{T}}-1}) \ge f(y^{\widetilde{\mathcal{T}}-2})$, and we conclude that $f(y^{\widetilde{\mathcal{T}}-1}) = f(v^{\widetilde{\mathcal{T}}-1}) = f(\widetilde{x}) = f(y^{\widetilde{\mathcal{T}}-2})$. The second equality together with the closedness and descent properties of the iteration mapping of the algorithm $\mathcal{A}_c^{k_c}$ at non-stationary solutions gives that $\widetilde{x} \in \mathrm{SOL}(\nabla f, X)$. [Moreover, $\widetilde{Y} \subseteq \mathrm{SOL}(\nabla f, X)$.]

Hence, $\varepsilon = 0$.

The algorithmic mapping describing the algorithm identified by extracting the choices k_{c} and k_{r} from the collections $\mathcal{K}_{\mathrm{c}}$ and $\mathcal{K}_{\mathrm{r}}$, and which defines the remaining iterates, is clearly of the form $C(x) := \{ y \in \hat{X} \mid f(y) \leq f(x) \}$, $x \in \hat{X}$, for any non-empty, compact and convex set $\hat{X} \subseteq X$. We may therefore invoke the Spacer Step Theorem (e.g., [Lue84, p. 231]), which guarantees that the result holds for the whole sequence, thanks to the properties of the mappings given by the choices k_{c} and k_{r} established above. This concludes the proof. ∎

We note that although the algorithm, the above theorem and its proof concern stationarity, a version of the algorithm which uses only Assumption 2.1(c)(2) is valid also for other forms of "optimality"; simply replace "stationary" with "optimal" in the algorithm and the theorem. This observation also helps in tying together the algorithm with the convergence theorems A–D in Zangwill [Zan69, Pages 91, 128, 241, and 244]. Theorem A, as all the other theorems, presumes nothing about what the "solution" is, and presumes that the algorithm yields descent in each step outside the solution set with respect to a continuous merit function for the problem, and generates a bounded sequence. The other theorems relax some of these conditions slightly, and among other techniques, the spacer steps used in the above proof are introduced. Our need for a special proof stems from two potential complications: the RMP is solved over a set which may vary rapidly between successive iterations, and in the column generation phase we allow for two different solution principles.

2.3 Instances of the algorithm

These algorithms are analyzed mainly for the case where X is polyhedral.

Frank–Wolfe The vector y^t is taken as any solution to the linearized problem

$$\operatorname*{minimize}_{y \in X} \ \nabla f(x)^{\mathrm{T}} y; \tag{2}$$

if, however, $\nabla f(x^t)^{\mathrm{T}} d < 0$ for some d in the recession cone of X, we let y^t be defined such that $d^t = y^t - x^t$ is a descent direction of unit length in the recession cone (such as the one obtained in the simplex method when unboundedness is detected). The boundedness of $\{x^t\}$ and $\{y^t\}$ follows from an assumption that the lower level set $L_f^X(x^0) := X \cap \{ x \in \Re^n \mid f(x) \leq f(x^0) \}$ is bounded. We have that $X^t := [x^t, y^t]$; the RMP thus is a simple line search problem. Several line search algorithms can be placed in the framework, including the exact line search mapping.

Simplicial decomposition The column generation problem is identical to that of the Frank–Wolfe algorithm. In the classic version, $X^{t+1} := \mathtt{conv}\,(X^t \cup \{y^t\})$, if no column dropping is used, or one first drops every column in X^t with zero weight in x^t, if column dropping according to the rule in [vHo77] is used. In both cases, clearly X^{t+1} is a non-empty, compact and convex subset of X, for which further $X^{t+1} \supset X^t$ holds for all t in the first case. [With reference to the above convergence proof, if the sequence of restrictions is expanding, it is guaranteed to have a unique set limit (e.g., [SaW79])].

In the RSD version ([HLV87]), we let X^{t+1} be given by the convex hull of $\{x^t\}$ and a finite number (at most $r \in \mathcal{Z}_+$) of the previously generated points y^s, $s = 1, 2, \ldots, t$, which always includes y^t. (If $r = 1$ then the Frank–Wolfe algorithm is obtained.) More on RSD follows below.

The exact solution of the RMP can be viewed as a mapping of the form $\mathcal{A}_{\mathrm{r}}^{k_t}$, which satisfies the conditions of Assumption 2.1. In [HLV87], a truncated algorithm—the exact solution of the quadratic approximation of the RMP—is proposed and analyzed. Also this mapping is closed, has the fixed-point property, and provides descent under the conditions stated in that paper.

Later, we will establish that certain properties of SD are inherited by the general algorithm. For future reference, we therefore provide a few more details on the SD algorithm.

By the Representation Theorem (e.g., [Las70, BSS93]), a vector $x \in \Re^n$ belongs to a polyhedral set X if and only if it can be represented as a convex combination of the extreme points (p_i, $i \in \mathcal{P}$) plus a non-negative linear combination of the extreme directions (d_j, $j \in \mathcal{D}$), that is, for some vectors λ and μ,

$$x = \sum_{i \in \mathcal{P}} \lambda_i p_i + \sum_{j \in \mathcal{D}} \mu_j d_j, \tag{3}$$

$$\sum_{i \in \mathcal{P}} \lambda_i = 1, \tag{4}$$

$$\lambda_i,\, \mu_j \geq 0, \qquad i \in \mathcal{P}, \quad j \in \mathcal{D}. \tag{5}$$

The result can be refined through Carathéodory's Theorem, such that only $\mathtt{dim}\, X + 1$ points are needed to describe an arbitrary feasible solution.

The classic form of the SD method was first described in [vHo77]. Given known subsets $\hat{\mathcal{P}}$ and $\hat{\mathcal{D}}$ of $\mathcal{P}$ and $\mathcal{D}$, respectively, f is minimized over the inner approximation of X which is defined when these subsets replace $\mathcal{P}$ and $\mathcal{D}$ in (3), in terms of the variables $\hat{\lambda}_i$, $i \in \hat{\mathcal{P}}$ and $\hat{\mu}_j$, $j \in \hat{\mathcal{D}}$. Notice that we use the notation $\hat{\lambda}$ and $\hat{\mu}$ to distinguish the vectors in the RMP from the (longer) vectors λ and μ in the complete master problem which is equivalent to CDP(f, X) and is defined by the system (3). Further denoting by $\hat{\Lambda}$ the set of vectors $(\hat{\lambda}, \hat{\mu})$ satisfying the restriction of the system (4)–(5) to the known subsets $\hat{\mathcal{P}}$ and $\hat{\mathcal{D}}$ and utilizing (3) to substitute x for $(\hat{\lambda}, \hat{\mu})$ [we write $x = x(\hat{\lambda}, \hat{\mu})$], the RMP may then be formulated as

$$\operatorname*{minimize}_{(\hat{\lambda}, \hat{\mu}) \in \hat{\Lambda}} f(x(\hat{\lambda}, \hat{\mu})), \tag*{[RMP$(f, \hat{\Lambda})$]}$$

Alternately, a profitable extreme point or direction of X is generated through the solution of the problem (2). If the solution x to this problem lies in the current inner approximation, then it is stationary in CDP(f, X). Otherwise, $\hat{\mathcal{P}}$ or $\hat{\mathcal{D}}$ is augmented by the new extreme point, and so on.

For problems where X is a bounded polyhedron, an improvement of SD, referred to as restricted simplicial decomposition (RSD), was devised by Hearn et al. [HLV85, HLV87]. The basis is the observation that an optimal solution x^* can be represented by an often much smaller number of extreme points than $\mathtt{dim}\, X + 1$, namely $\mathtt{dim}\, F^* + 1$, where F^* is the *optimal face* of X, that is, the face of X of the smallest dimension which contains x^*. [In the context of convex minimization, this set may be described by

$$F^* := \{\, y \in X \mid \nabla f(x^*)^{\mathrm{T}}(y - x^*) = 0 \,\},$$

a set which is spanned by the extreme points of X that solve the linear approximation (2) to CDP(f, X) defined at any optimal solution.] They devise a modification in which the number of extreme points retained is kept below a positive integer, r; when this number of extreme points has been reached, any new extreme point generated replaces the column in $\hat{\mathcal{P}}$ that received the least weight in the solution to the RMP. In order to ensure the convergence of the algorithm, the optimal solution x to the RMP must also be retained as an individual column (however not counted among the r columns).

The value of r is crucial to the performance of the algorithm. If $r \geq \mathtt{dim}\, F^* + 1$, then the number of RMP is finite, and the local rate of convergence is governed by the local convergence rate of the method chosen for the solution of the RMP; thus, a super-linear or quadratic convergence rate may be attained if a (projected) Newton method is used ([HLV87]). If $r < \mathtt{dim}\, F^* + 1$, then the algorithm is only asymptotically convergent, and the rate of convergence is the same as that of the *Frank–Wolfe* algorithm, that is, sub-linear.

The RSD algorithm has been successfully applied to large-scale, structured non-linear optimization problems, in particular mathematical programming models of non-linear network flow problems, where the column generation subproblem reduces to efficiently solvable linear network flow problems (e.g., [HLV87, LaP92]). Experience with the RSD method has shown that it makes rapid progress initially, especially when relatively large values of r are used and when second-order methods are used for the solution of the RMP, but that it slows down close to an optimal solution. It is also relatively less efficient for larger values of $\mathtt{dim}\, F^*$, since the number and size of the RMP solved within the method become larger.

The explanation for this behaviour is to be found in the construction of a *linear* column generation subproblem: the quality of the resulting search directions is known to deteriorate rapidly. (As the sequence $\{x^t\}$ tends to a stationary point, the sequence $\{\nabla f(x^t)^{\mathrm{T}} d^t\}$ of directional derivatives of the search directions $d^t := y^t - x^t$ tends to zero whereas $\{d^t\}$ does not; thus, the search directions rapidly tend to become nearly orthogonal to the gradient of f.) Similarly, the quality of the columns generated in this fashion will also deteriorate in terms of their improvement in the objective value. It is a natural conclusion that better approximations of f can be exploited in the column generation phase of SD methods; then, the columns generated would be of better quality, thus leading to larger improvements in the inner approximations of the feasible set.

Truncated algorithms for CDP Convergent (closed and descent-based) iterative algorithms for CDP can be supplied with accelerating multidimensional searches, by placing them as column generation problems of the form of (c)(2) in Assumption 2.1. The algorithm then acts as a truncated algorithm for CDP composed with (perhaps more exact) solutions of smaller-dimensional and simply constrained optimization problems.

In this case, it is natural that the algorithm for the RMP starts at y^t, not at x^t as is stated in the description of the algorithm in Table 1, since $f(y^t) < f(x^t)$. This can be accommodated in our framework as follows: redefine

$$\mathcal{A}_{\mathrm{r}}^{k_t} := \mathcal{A}_{\mathrm{r}}^{k_t} \circ \mathcal{A}_{\mathrm{c}}^{k_t} \circ \ldots \circ \mathcal{A}_{\mathrm{c}}^{k_t},$$

which maps the argument x^t into y^t through $y^t = \mathcal{A}_{\mathrm{c}}^{k_t} \circ \ldots \circ \mathcal{A}_{\mathrm{c}}^{k_t}(x^t)$, and then applies the RMP algorithm.

Two example instances are briefly mentioned. (1) *Truncated Frank–Wolfe.* Consider a line search method based on, e.g., Newton's method. Since the subproblem may be computationally expensive, a few iterations of the Frank–Wolfe algorithm on the quadratic subproblem may yield a sufficiently accurate direction given the time consumed. This has been proposed in [DeT88], and in combination with other line search methods than Newton's in [Mig94]. An application in the current framework is to use a few steps of the Frank–Wolfe algorithm on the original problem to generate a column. (2) *Truncated coordinate search.* Line search-based modifications of the Hooke and Jeeves [HoJ61] method (e.g., [BSS93, Section 8.5]) combine, in one iteration, a coordinate-wise search through each of the variables with an exploratory search along the vector between the last iterate and the result of the coordinate search. An acceleration of this methodology is to store the result of one, or possibly several, iterations of the coordinate search algorithm as a column in the proposed scheme. Convergence to stationary points is ensured if, in addition to the assumptions given so far, the objective function f has a unique minimum along each coordinate.

Nonlinear simplicial decomposition (NSD) The algorithm was proposed in [LPR97], and was motivated largely by making observations about the convergence of the SD and RSD algorithms. The algorithm is a multidimensional search extension of a large class of line search methods, which has been analyzed in slightly different forms by several researchers, notably Tseng [Tse91], Migdalas [Mig94], and in most generality by Patriksson [Pat98b]. The motivations for the algorithm are that by generating columns based on better approximations of the objective function, the sensitivity of the method to the dimension of the optimal face will be reduced, fewer columns will be needed to describe an optimal solution, resulting in fewer iterations, and enabling a smaller value of the parameter r to be chosen. Further, more efficient methods can be applied to each RMP, since they are smaller. We discuss this algorithm in some detail below, because a number of the finite convergence results will be stated in this framework.

The NSD method is obtained from the RSD method by replacing the linear column generation subproblem with the more general problem to

$$\underset{y \in X}{\text{minimize}} \ \nabla f(x)^{\mathrm{T}} y + \varphi(y, x), \qquad [\mathrm{CDP}(\varphi(\cdot, x), \nabla f, X, x)]$$

where $\varphi : X \times X \mapsto \Re$ is a continuous function of the form $\varphi(y, x)$, convex and continuously differentiable with respect to y for all $x \in X$, and with the properties that $\varphi(x, x) = 0$ and $\nabla_y \varphi(x, x) = 0^n$ for all $x \in X$. Among the possible choices for φ we mention the following, where x^t denotes an iteration point at iteration t, `diag` denotes the diagonal part of a matrix and where $\gamma > 0$:

$\varphi(y, x^t)$	Subproblem
0	Frank–Wolfe
$(1/2)(y - x^t)^{\mathrm{T}} \nabla^2 f(x^t)(y - x^t)$	Newton
$(1/2)(y - x^t)^{\mathrm{T}} [\mathtt{diag}\, \nabla^2 f(x^t)](y - x^t)$	Diag. Newton
$(\gamma/2)\|y - x^t\|^2 := (\gamma/2)(y - x^t)^{\mathrm{T}}(y - x^t)$	Projection

Under names such as partial linearization, regularized Frank–Wolfe and cost approximation, line search methods based on this subproblem, and also generalizations thereof, have been analyzed in [Tse91, Pat93a, Pat93b, Mig94, ZhM95, Pat98a, Pat98b].

Besides providing improvements over the RSD algorithm, NSD may also improve upon its line search algorithm origin; for highly non-linear problems, line search methods often become ineffective, due to very short steps being taken in the line searches. Several alternative schemes have been developed to cope with this deficiency, such as non-monotone and curve-linear searches, and trust region approaches. The framework of NSD may be another interesting alternative worthy of study.

While the solution to (2) is an extreme point of X, a solution $\hat{y}^t$ to $\mathrm{CDP}(\varphi(\cdot, x), \nabla f, X, x)$ may be in the (relative) interior of X; in order to augment the inner approximation, $X^t \subset X$, as much as possible, the NSD method does not store $\hat{y}^t$ but its extension on the (relative) boundary of X, that is,

$$y^t := x^t + \ell_t(\hat{y}^t - x^t), \qquad \text{where} \qquad \ell_t := \max\{\, \ell \mid x^t + \ell(\hat{y}^t - x^t) \in X \,\}. \tag{6}$$

Even though the finite convergence property will be lost in general (because non-extremal points will be generated, see also Example 4.17), one may expect a more rapid convergence of the NSD method than the RSD method in terms of the number of iterations needed to reach a given solution accuracy. In numerical experiments performed on large-scale non-linear network flow problems in [LPR97], it was particularly observed that the NSD method is relatively much less sensitive to the value of $\mathtt{dim}\, F^*$ than is RSD, which permits the use of a much smaller value of r in the NSD method. Further applications of NSD have been reported in transportation planning ([GaM97a, GaM97b, LuP98]), where Evans' [Eva76] algorithm has been supplied with a multidimensional search.

In the next section, we study realizations of the construction of the inner approximation of the feasible set, and establish conditions under which this set is a simplex. We then provide conditions on the problem and on the realization of the algorithm such that the active constraints at a solution (Section 4.2) or even an optimal solution itself (Section 4.3), will be attained in a finite number of iterations. The latter results are also extended to variational inequality problems in Section 4.4.

3 Properties of the RMP

As has been remarked upon in Section 2, the inner approximations of the set X employed in SD methods are polyhedral sets whose extreme points are extreme points of X. In RSD, the inner approximation is slightly redefined such that whenever a column dropping has been performed, the previous solution to the RMP is also retained as a column.

One may consider more general rules for constructing the inner approximation; cf., for example, the condition on X^{t+1} in Step 3 in the description of the conceptual algorithm in Table 1. In order to establish properties similar to those for SD methods, it is necessary to introduce further conditions on the updating of the sets X^t. In this section, we establish rules for introducing and dropping columns so as to maintain the simplicial property of the sets X^t.

3.1 Set augmentation

When determining the updating of the inner approximation from one iteration to the next, we consider two phases.

First, given the solution x^t to the current RMP, we may drop columns from the active set, for example based on their barycentric coordinates (weights) in describing x^t. If we employ a restriction strategy corresponding to that of Hearn et al. [HLV85, HLV87] based on a maximum number of columns stored, we may also drop columns that have more than an insignificant weight, while then also introducing the vector x^t as an individual column.

Second, a main difference between the simplicial decomposition method, as proposed by von Hohenbalken and successors, and the method of this paper, is that the columns are not necessarily extreme points of X. We however do assume that the columns introduced belong to the (relative) boundary of X. This corresponds to utilizing, if necessary, the rule (6).

In the following, we use $\mathcal{P}_s^t$ to denote the set of columns generated in Step 1 of the CG algorithm, and retained at iteration t; further, $\mathcal{P}_x^t$ is either an empty set or it contains one column which corresponds to the result of Step 4 in the previous iteration.

Table 2 summarizes the various rules considered in the introduction of new and deletion of old columns; they are realizations of Step 3 in the conceptual algorithm of Table 1. (The corresponding initializations necessary are also included.)

The column dropping rule 3.1.a is applied in the original work on SD ([Hol74, vHo77]), as well as in the later developments in [HLV85, HLV87, LPR97]. The rule 3.1.b is to be used when the RMP is only solved inexactly.

Definition 3.1 (ε-stationarity). The vector $x \in X$ is an *ε-stationary* solution to CDP(f, X) if

$$\nabla f(x)^{\mathrm{T}}(y - x) \geq -\varepsilon, \qquad y \in X, \tag{8}$$

holds. □

Proposition 3.2 *Let $\bar{x}^t$ be an ε-stationary solution to the RMP at iteration t with $\bar{x}^t = \sum_i^m \beta_i p_i$, where $\sum_i^m \beta_i = 1$, $\beta_i \geq 0$, $p_i \in \mathcal{P}^t$ for all $i \in \{1, \dots, m\}$, and $m = |\mathcal{P}^t|$. Then for any $j \in \{1, \dots, m\}$,*

$$\nabla f(\bar{x}^t)^{\mathrm{T}}(p_j - \bar{x}^t) \geq \varepsilon_1^t > 0 \qquad \Longrightarrow \qquad \beta_j \leq \frac{\varepsilon}{\varepsilon + \varepsilon_1^t}. \tag{9}$$

Proof. Let $z = \bar{x}^t + \frac{\beta_j}{(1-\beta_j)}(\bar{x}^t - p_j) = \sum_{i \neq j}^m \frac{\beta_i}{(1-\beta_j)} p_i$. The element z belongs to X^t because it is a convex combination of points of $\mathcal{P}^t \subset X^t$ and X^t is a convex set.

Using the property of ε-stationarity of $\bar{x}^t$ over X^t,

$$-\varepsilon \leq \nabla f(\bar{x}^t)^{\mathrm{T}}(z - \bar{x}^t) = -\frac{\beta_j}{(1 - \beta_j)} \nabla f(\bar{x}^t)^{\mathrm{T}}(p_j - \bar{x}^t)$$

Table 2: The set augmentation phase

---

0. (*Initialization*): Choose an initial point $x^0 \in X$, let $t := 0$, $\mathcal{P}_s^0 = \emptyset$, $\mathcal{P}_x^0 = \{x^0\}$, $\mathcal{P}^0 = \mathcal{P}_s^0 \cup \mathcal{P}_x^0$ and $X^0 = \texttt{conv}(\mathcal{P}^0)$. Further, let r be a positive integer, and let $\Re_{++} \supset \{\varepsilon_1^t\} \to 0$.

3.1 (*Column dropping rules*): Let $x^t = \sum_{i=1}^m \beta_i p_i$, where $m = |\mathcal{P}^t|$ and $p_i \in \mathcal{P}^t$.

3.1.a (Exact solution of RMP). Discard all elements p_i with weight $\beta_i = 0$.

3.1.b (Truncated solution of RMP). Discard all elements p_i satisfying

$$\nabla f(x^t)^{\mathrm{T}}(p_i - x^t) \geq \varepsilon_1^t > 0. \tag{7}$$

3.2 (*Extension to the relative boundary of* X): Let $\hat{y}^t$ be the vector generated, and let y^t be defined by (6).

3.3 (*Set augmentation rules*):

3.3.a (Simplicial decomposition scheme). $\mathcal{P}^{t+1} = \mathcal{P}^t \cup \{y^t\}$. Set $X^{t+1} = \texttt{conv}(\mathcal{P}^{t+1})$.

3.3.b (Restricted simplicial decomposition scheme). If $|\mathcal{P}_s^t| < r$, then set $\mathcal{P}_s^{t+1} = \mathcal{P}_s^t \cup \{y^t\}$, and $\mathcal{P}_x^{t+1} = \mathcal{P}_x^t$; otherwise, replace the element of $\mathcal{P}_s^t$ with minimal weight in the expression of x^t with y^t to obtain $\mathcal{P}_s^{t+1}$, and let $\mathcal{P}_x^{t+1} = \{x^t\}$. Finally, set $\mathcal{P}^{t+1} = \mathcal{P}_s^{t+1} \cup \mathcal{P}_x^{t+1}$ and $X^{t+1} = \texttt{conv}(\mathcal{P}^{t+1})$.

---

holds. Further, by hypothesis, we obtain that $-\frac{\beta_j}{(1-\beta_j)}\nabla f(\bar{x}^t)^{\mathrm{T}}(p_j - \bar{x}^t) \leq -\frac{\beta_j}{(1-\beta_j)}\varepsilon_1^t$. Combining these inequalities then yields the desired result. □

Remark 3.3 (Equivalence of two column dropping rules). This result implies that if the RMP is solved exactly (i.e., $\varepsilon = 0$ holds), then $\beta_j = 0$ in (9), whence the two rules coincide. (See also [HLV85, Lemma 1] for a similar result in the context of the RSD method.) In general, however, the rule 3.1.a implies a more aggressive column dropping than does rule 3.1.b. □

3.2 Inner approximations are simplices

From here on, we further presume that f *is pseudo-convex* on X.

According to [HLV85, Theorem 3], the inner approximations $X^t = \texttt{conv}(\mathcal{P}^t)$ utilized in the SD and RSD algorithms are simplices, under the hypothesis that the RMP are solved exactly. We now establish, based on similar arguments, that this property also holds in the method proposed, even without an assumption that X is polyhedral. We then need the following definition and properties of a simplex, taken from Rockafellar [Roc70].

Definition 3.4 (Simplex). Let $\{z_0, z_1, \ldots, z_m\}$ be $m + 1$ distinct points in $\Re^n$ with $m \leq n$ where the vectors $z_1 - z_0, z_2 - z_0, \ldots, z_m - z_0$ are linearly independent. Then, the set $C = \texttt{conv}(z_0, z_1, \ldots, z_m)$, the convex hull of $\{z_0, z_1, \ldots, z_m\}$, is an *m-simplex* in $\Re^n$. In addition, since C is always contained in a manifold of dimension m, C is said to have dimension m, or $\texttt{dim}\ (C) = m$. □

Proposition 3.5 (Properties of simplices).

(a) *If x is an element of an m-simplex, C, then x can be uniquely expressed as a convex* *combination of points, $z_0, z_1, \dots, z_m$, defining C, i.e.,*

$$x = \sum_{i=0}^{m} \beta_i z_i, \qquad \sum_{i=0}^{m} \beta_i = 1, \qquad \beta_i \geq 0, \qquad i = 0, 1, \dots, m,$$

and $\beta_0, \beta_1, \dots, \beta_m$ are unique.

(b) *If x is an element of an m-simplex, C, and the convexity weight, β_i, for some $i = 0, 1, \dots, m$ is positive in the (unique) expression of x as a convex combination of $z_0, z_1, \dots, z_m$, then the set $\texttt{conv}\,(z_0, z_1, \dots, z_{i-1}, x, z_{i+1}, \dots, z_m)$ is an m-simplex.*

(c) *If $\texttt{conv}\,(z_0, z_1, \dots, z_m)$ is an m-simplex, then $\texttt{conv}\,(z_0, z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m)$, for some $i = 0, 1, \dots, m$, is an $(m-1)$-simplex.* □

The main result of this section follows.

Theorem 3.6 (The inner approximation is a simplex). *Assume that the RMP are solved exactly. Then, the set X^t is a simplex for all t.*

Proof. We show by induction that X^t is a simplex at the start of Step 3.3. It follows that X^t is a simplex in every step of the algorithm.

When $t = 0$, $X^t = \{x^0\}$; therefore, X^0 is a 0-simplex. Assume now that X^t is a simplex for $t \geq 0$. The elements with zero weight have been discarded at the beginning of Step 3.3; therefore, the remaining elements in $\mathcal{P}^t$ must have positive weight. By the induction hypothesis and Proposition 3.5.c the points not eliminated define a simplex. Assume without loss of generality that at the beginning of Step 3.3, $\mathcal{P}^t = \{p_0, p_1, \dots, p_m\}$, and that, by assumption, $\mathcal{P}^t$ defines an m-simplex. We denote the convex hull of this set by $\bar{X}^t$.

The element x^t is expressed as

$$x^t = \sum_{i=0}^{m} \beta_i p_i, \qquad \text{with } \beta_i > 0 \text{ and } p_i \in \mathcal{P}^t.$$

It follows that $x^t \in \texttt{rint}\,(\bar{X}^t)$. We will prove that if x^t is not an optimal solution to [CDP(f,X)] then $\texttt{conv}\,(\bar{X}^t \cup \{y^t\})$ is a simplex, where y^t is the column added in iteration $t+1$. First, however, we note that x^t is also an optimal solution to the problem of minimizing f over $\texttt{aff}\,(\bar{X}^t) \cap X$, where $\texttt{aff}\,(\bar{X}^t)$ is the affine hull of $\bar{X}^t$, since no constraint of the form $\beta_i \geq 0$ is binding, so

$$\nabla f(x^t)^{\mathrm{T}}(y - x^t) \geq 0, \qquad y \in \texttt{aff}\,(\bar{X}^t) \cap X, \tag{10}$$

which we proceed to establish.

Let y be an arbitrary element of $\texttt{aff}\,(\bar{X}^t) \cap X$. If $y \in \bar{X}^t \subset X^t$ then the point y satisfies the inequality in (10) because x^t solves the RMP over X^t. Otherwise, $y \in \texttt{aff}\,(\bar{X}^t) - \bar{X}^t$. Using the fact that x^t is in the relative interior of $\bar{X}^t$, there exists a unique element, z, in the set $[x^t, y] \cap \texttt{rfro}\,(\bar{X}^t)$, where $\texttt{rfro}\,(\bar{X}^t)$ is the relative boundary of $\bar{X}^t$. This point satisfies $y = x^t + \lambda(z - x^t)$ for some $\lambda > 1$. By the optimality of x^t over X^t and the fact that $z \in X^t$, we obtain that $\nabla f(x^t)^{\mathrm{T}}(z - x^t) \geq 0$, whence it follows that $\nabla f(x^t)^{\mathrm{T}}(y - x^t) = \lambda[\nabla f(x^t)^{\mathrm{T}}(z - x^t)] \geq 0$. This completes the proof of (10).

If x^t solves [CDP(f,X)] then the algorithm terminates without generating X^{t+1}. Otherwise, by Assumption 2.1.c and the use of the rule (6), it follows that the column y^t generated in Step 3.2 satisfies $\nabla f(x^t)^{\mathrm{T}}(y^t - x^t) < 0$. This relation, together with the optimality of x^t over $\texttt{aff}\,(\bar{X}^t) \cap X$, implies that $y^t \notin \texttt{aff}\,(\bar{X}^t)$. As $\bar{X}^t$ is an m-simplex by the induction hypothesis, $\texttt{conv}\,(\bar{X}^t \cup \{y^t\})$ is therefore an $(m+1)$-simplex.

In the case that $m = |\mathcal{P}_s^t| < r$ holds, that set is produced by Step 3.3.a and Step 3.3.b. The only other case to consider is the use of Step 3.3.b in the case when $m = |\mathcal{P}_s^t| = r$

holds. We will then assume without loss of generality that $\mathcal{P}_s^t = \{p_0, \ldots, p_{m-1}\}$, and let $\mathcal{P}_x^t = \{x'\}$. By assumption, $\mathcal{P}^t$ defines an m-simplex. In this case $X^{t+1} = \mathtt{conv}(\mathcal{P}^{t+1})$ where $\mathcal{P}^{t+1} = \{p_0, \ldots, p_{i-1}, p_{i+1}, \ldots, p_{m-1}, x^t, y^t\}$, for some i. This set defines an m-simplex because $\mathtt{conv}(p_0, p_1, \ldots, p_{m-1}, x', y^t)$ is an $(m+1)$-simplex by the above, by Proposition 3.5.b $(p_0, p_1, \ldots, p_{m-1}, x^t, y^t)$ is an $(m+1)$-simplex, and $\mathtt{conv}(p_0, \ldots, p_{i-1}, p_{i+1}, \ldots, p_{m-1}, x^t, y^t) =: X^{t+1}$ is an m-simplex by Proposition 3.5.c. Thus, in either case, $\mathtt{conv}(\mathcal{P}^{t+1})$ is a simplex. This completes the proof. □

4 Finiteness convergence properties of the column generation algorithm

This section offers some insight into the finite convergence properties of the column generation algorithm proposed. The investigation is divided in two parts. First, we establish conditions on the problem and on the algorithm so that the optimal face will be attained in a finite number of iterations. When X is polyhedral, this result implies the finite identification of the active constraints at the limit point. Second, we study the stronger property of finitely attaining an optimal solution, under the condition of weak sharpness of the set $\mathrm{SOL}(f, X)$. We finally extend these results to variational inequality problems.

4.1 Facial geometry and non-degeneracy

We begin with some elementary properties of faces of convex sets.

Definition 4.1 (Face). Let X be a convex set in $\Re^n$. A convex set F is a *face* of X if the endpoints of any closed line segment in X whose relative interior intersects F are contained in F. Thus, if x and y are in X and $\lambda x + (1-\lambda) y$ lies in F for some $0 < \lambda < 1$, then x and y must also belong to F. □

The following two results appear in [Roc70, Theorems 18.1–2].

Theorem 4.2 *Let F be a face of the convex set X. If Ω is a convex subset of X so that $\mathtt{rint}\,\Omega$ meets F, then $\Omega \subset F$.* □

A corollary to this result is that if the relative interiors of two faces F_1 and F_2 have a non-empty intersection then they are equal. The following result complements the above by establishing that each point in a convex set belongs to the relative interior of a unique face.

Theorem 4.3 *The collection of all relative interiors of faces of the convex set X is a partition of X.* □

We will use the notation $F(x)$ to denote the unique face F of X for which $x \in \mathtt{rint}\,F$. Note that this is the minimal face containing the point x. We will subsequently characterize these minimal faces.

Definition 4.4 (The k-tangent cone $K_X(x)$). A vector v is said to be *k-tangent* to the set X at the point x in X if for some $\varepsilon > 0$, $x + tv \in X$ holds for all $t \in (-\varepsilon, \varepsilon)$. The set of all k-tangent vectors v at x is a cone, which we denote by $K_X(x)$. □

For any cone C, let $\mathtt{lin}\,C$ denote the *lineality* of C, the largest subspace contained in C, that is, $\mathtt{lin}\,C = C \cap (-C)$.

Lemma 4.5 (Characterization of $F(x)$). *Let $x \in X$. It holds that $F(x) = (x + \texttt{lin}\, K_X(x)) \cap X$.*

Proof. It is obvious that $(x + \texttt{lin}\, K_X(x)) \cap X$ is a face of X satisfying $x \in \texttt{rint}\,((x + \texttt{lin}\, K_X(x)) \cap X) \cap \texttt{rint}\, F(x)$. Using Theorem 4.3 these faces are identical. □

Recall next the definition (1) of the normal cone $N_X(x)$ to the set X at x. We note that if F is a face of X, then the normal cone is independent of $x \in \texttt{rint}\, F$, whence we may write $N_X(F)$. The *tangent cone* to X at x, $T_X(x)$, is the polar cone of $N_X(x)$. The face of X exposed by the vector $d \in \Re^n$ (the *exposed face*) is the set

$$E_X(d) = \arg\max_{y \in X} d^{\mathrm{T}} y.$$

For general convex sets, $x \in E_X(d)$ holds if and only if $d \in N_X(x)$ (e.g., [BuM94]), whereas for polyhedral sets, the exposed face is independent of the choice of $d \in \texttt{rint}\, N_X(x)$. Further, every face of a polyhedral set is exposed by some vector d. These results however fail to hold for general convex sets. In the analysis of identification properties of the CG algorithm, we will focus on faces of X which enjoy stronger properties than faces in general.

Definition 4.6 (Quasi-polyhedral face). A face F of X is *quasi-polyhedral* if

$$\texttt{aff}\, F = x + \texttt{lin}\, T_X(x), \qquad x \in \texttt{rint}\, F. \tag{11}$$

□

The relative interior of a quasi-polyhedral face is equivalent to an open facet, as defined in [Dun87]. Every face of a polyhedral set X is quasi-polyhedral, but this is not true for a general convex set, as the example in [BuM88] shows. Further, a quasi-polyhedral face need not to be a polyhedral set, and vice versa. Quasi-polyhedral faces are however exposed by any vector in $\texttt{rint}\, N_X(F)$, and have several other properties in common with faces of polyhedral sets. See [BuM88, BuM94] for further properties of quasi-polyhedral faces.

We now turn to study the optimal face of X. The following definition extends the classic one for problems with unique solutions (e.g., [Wol70]).

Definition 4.7 (Optimal face). The *optimal face* of [CDP(f, X)] is

$$F^* = \bigcap_{x^* \in \mathrm{SOL}(f,X)} F_{x^*},$$

where $F_{x^*} = \{\, y \in X \mid \nabla f(x^*)^{\mathrm{T}}(y - x^*) = 0 \,\}$. □

The set F^* is a face because it is the intersection of a collection of faces. It is elementary to show that whenever f is pseudo-convex, $F^* \supset \mathrm{SOL}(f, X)$ holds. Further note that for any stationary point x^*, the face F_{x^*} is the exposed face $E_X(-\nabla f(x^*))$.

In the case where f is convex, we recall that the value of ∇f is constant on $\mathrm{SOL}(f, X)$, by a result of Burke and Ferris [BuF91]. Therefore, in this case, $F^* = F_{x^*} = E_X(-\nabla f(x^*))$ for any $x^* \in \mathrm{SOL}(f, X)$, simplifying the above definition.

Under the following regularity condition on an optimal solution, the finite identification of the optimal face has been demonstrated for several algorithms (e.g., [Dun87, BuM88, Pat98b]):

Definition 4.8 (Non-degenerate solution). An optimal solution x^* to [CDP(f, X)] is *non-degenerate* if

$$-\nabla f(x^*) \in \texttt{rint}\, N_X(x^*) \tag{12}$$

holds. □

We note that this regularity condition does not depend on the representation of the set X. When X is described explicitly with constraints, then the condition is weaker than strict complementarity ([BuM88]). Before establishing the finite identification result for the CG algorithm, we introduce two other regularity conditions that have been used in the literature, and relate them to each other.

Definition 4.9 (Conditions of regularity). (1) (Geometric stability, [MaD89]). An optimal solution x^* is *geometrically stable* if

$$\nabla f(x^*)^{\mathrm{T}}(x - x^*) = 0 \qquad \Longrightarrow \qquad x \in F^*. \tag{13}$$

(2) (Geometric regularity, [DuM89]). The optimal face F^* is *geometrically regular* if

$$\mathrm{SOL}(f, X) \ \subset \mathtt{rint}\, F^*, \tag{14}$$

and the set $\mathrm{SOL}(f, X)$ is non-degenerate in the sense of Definition 4.8. □

A sufficient condition for geometric stability is the convexity of f on X, as remarked above.

The notions of geometric stability and regularity are equivalent when X is a bounded polyhedral set (see [DuM89, Corollary 2.4]). The following result extends this characterization to the case of general convex sets under a non-degeneracy assumption. (The constraint qualification (CQ) of Guignard [Gui69] utilized in the result implies that $N_X(x)$ is a polyhedral cone for every x, and is satisfied automatically for polyhedral sets X.)

Theorem 4.10 (Relations among conditions of regularity). *Assume that Guignard's CQ holds. Further, assume that* $\mathrm{SOL}(f, X)$ *is a set of non-degenerate optimal solutions. Consider the following three statements.*

(a) F^* *is geometrically regular.*

(b) F^* *is a quasi-polyhedral face, and* $F^* = F(x^*)$ *holds for all* $x^* \in \mathrm{SOL}(f, X)$.

(c) *Every* $x^* \in \mathrm{SOL}(f, X)$ *is geometrically stable.*

It then holds that (a) $\Longrightarrow$ (b) $\Longrightarrow$ (c).

Proof. [(a) $\Longrightarrow$ (b)]. The following relationship holds: $x^* \in (\mathtt{rint}\, F^*) \cap (\mathtt{rint}\, F(x^*))$ for all $x^* \in \mathrm{SOL}(f, X)$. By Theorem 4.3, $F^* = F(x^*)$ holds for all $x^* \in \mathrm{SOL}(f, X)$.

We now prove that F^* is a quasi-polyhedral face. As $x^* \in \mathtt{rint}\, F^*$, we demonstrate that $F^* = (x^* + \mathtt{lin}\,(T_X(x^*)) \cap X$. We begin by showing that $F^* \subset (x^* + \mathtt{lin}\,(T_X(x^*)) \cap X$. Using Lemma 4.5, we obtain that $F^* = (x^* + K_X(x^*)) \cap X$. Moreover, $K_X(x^*) \subset T_X(x^*)$ and $\mathtt{lin}\, K_X(x) = K_X(x)$, which establishes the inclusion. Conversely, let $x^* + v \in (x^* + \mathtt{lin}\,(T_X(x^*)) \cap X$. Using Lemma 2.7 in [BuM88], it follows that $v \in N_X^{\perp}(x^*)$, where $\perp$ denotes the orthogonal complement. On the other hand, $N_X^{\perp}(x^*) = N_X^{\perp}(z)$ holds for all $z \in \mathtt{rint}\, F^*$ (see [BuM88, Theorem 2.3]). This implies that $v \in N_X^{\perp}(y^*)$ for every $y^* \in \mathrm{SOL}(f, X)$. As $-\nabla f(y^*) \in N_X(y^*)$, $\nabla f(y^*)^{\mathrm{T}} v = 0$ holds for every $y^* \in \mathrm{SOL}(f, X)$. This relationship establishes that $\nabla f(y^*)^{\mathrm{T}}(x^* + v - y^*) = \nabla f(y^*)^{\mathrm{T}} v + \nabla f(y^*)^{\mathrm{T}}(x^* - y^*) = 0$, and $x^* + v \in F_{y^*}$ for all $y^* \in \mathrm{SOL}(f, X)$. By the definition of F^*, we obtain that $x^* + v \in F^*$.

[(b) $\Longrightarrow$ (c)]. Let $x^* \in \mathrm{SOL}(f, X)$. We prove that if $\nabla f(x^*)^{\mathrm{T}}(z - x^*) = 0$, for $z \in X$, then $z \in F^*$. As $N_X(x^*)$ is a polyhedral cone and $-\nabla f(x^*) \in N_X(x^*)$, a set of vectors and scalars exists so that $-\nabla f(x^*) = \sum \lambda_i v_i$, where $\lambda_i \geq 0$. The point x^* is non-degenerate; using Lemma 3.2 of [BuM88] these coefficients must therefore be positive. The relationship $0 = -\nabla f(x^*)^{\mathrm{T}}(z - x^*) = \sum \lambda_i v_i^{\mathrm{T}}(z - x^*)$ implies that $v_i^{\mathrm{T}}(z - x^*) = 0$ for all i, and hence that $(z - x^*) \in N_X^{\perp}(x^*)$. Using Lemma 2.7 of [BuM88], $N_X^{\perp}(x^*) = \mathtt{lin}\, T_X(x^*)$. By hypothesis,

$F^* = (x^* + \mathtt{lin}(T_X(x^*)) \cap X$ and $z = x^* + (z - x^*) \in (x^* + \mathtt{lin}(T_X(x^*)) \cap X = F^*$. This completes the proof. $\square$

The result can not be strengthened to an equivalence among all three: (c) $\Longrightarrow$ (a) may fail for non-polyhedral sets.

4.2 Finite identification of the optimal face

The identification results to follow will be established under the following assumption on the construction and solution of the sequence of RMP:

Assumption 4.11 (Conditions on the RMP). *Either one of the following conditions hold.*

(1) $r \geq \mathtt{dim}\, F^* + 1$, *and the RMP are solved exactly.*

(2) $r = \infty$, *and the RMP are solved such that x^t is ε^t-optimal with $\{\varepsilon^t\} \downarrow 0$.*

Theorem 4.12 (Identification results). *Let $\{x^t\}$ and $\{\hat{y}^t\}$ be the sequence of iterates and columns generated by the algorithm, and assume that $\{x^t\}$ converges to $x^* \in \mathrm{SOL}(f, X)$.*

(a) *Assume further that the RMP are solved exactly. If a positive integer τ_1 exists such that $x^t \in \mathtt{rint}\, F^*$ for every $t \geq \tau_1$, then there exists a positive integer τ_2 such that*

$$\hat{y}^t \in F^*, \qquad t \geq \tau_2.$$

(b) *Let Assumption 4.11 hold, and assume that F^* is geometrically regular. If a positive integer τ_1 exists such that $\hat{y}^t \in F^*$ for every $t \geq \tau_1$, then there exists a positive integer τ_2 such that*

$$x^t \in \mathtt{rint}\, F^*, \qquad t \geq \tau_2.$$

Proof. (a) Let $t \geq \tau_1$, so that $x^t \in \mathtt{rint}\, F^*$. First we show that the column y^{t+1} generated by the rule (6) belongs to the optimal face F^*. Since in each iteration the RMP is solved exactly, $x^{t+1} \in X^{t+1} - X^t$ holds, and hence $x^{t+1} = \lambda y^{t+1} + (1 - \lambda) z$, where $z \in X^t$ and $0 < \lambda \leq 1$. If $\lambda = 1$ then the result follows trivially. Otherwise, using the fact that F^* is a face, and that $x^{t+1} \in (z, y^{t+1}) \cap \mathtt{rint}\, F^*$, we have that $[z, y^{t+1}] \subset F^*$, and hence y^{t+1} belongs to the optimal face. Now we show that also $\hat{y}^{t+1}$ belongs to F^*. Using that $x^{t+1} \in \mathtt{rint}\, F^* \subset F^*$, it follows that $[x^{t+1}, y^{t+1}] \subset F^*$. Since $\hat{y}^{t+1} \in [x^{t+1}, y^{t+1}]$ holds, the result follows.

(b) Let $t \geq \tau_1$, so that $\hat{y}^t \in F^*$. If $\hat{y}^t \in \mathtt{rint}\, F^*$, then since F^* is a face of X, also y^t belongs to F^*. Otherwise, $\hat{y}^t \in \mathtt{rfro}\, F^*$, whence the rule (6) produces $y^t = \hat{y}^t \in F^*$. This guarantees that $\{y^t\}_{t \geq \tau_1} \subset F^*$.

We next prove that if there exists an element z that is never discarded from the set $\mathcal{P}^t$ for $t \geq \tau$, then z is in the optimal face. This is obviously possible if and only if z does not satisfy the column dropping rule in any iteration $t \geq \tau$. Let $t \geq \tau$, and let the solution to the RMP at iteration t be expressed by

$$x^{t+1} = \beta_z^t z + \sum_{i=1}^{n_t} \beta_i^t p_i, \quad 0 < \beta_z^t,\ 0 \leq \beta_i^t,\ i = 1, \ldots, n_t \text{ and } \beta_z^t + \sum_{i=1}^{n_t} \beta_i^t = 1,\ p_i \in \mathcal{P}^t.$$

If the RMP is solved inexactly, then the fact that z does not satisfy the column dropping rule implies that $\nabla f(x^{t+1})^{\mathrm{T}}(z - x^{t+1}) < \varepsilon_1^{t+1}$. Using the continuity of $\nabla f(x)$, and taking the limit of the inequality, we obtain that

$$\nabla f(x^*)^{\mathrm{T}}(z - x^*) = \lim_{t \to \infty} \nabla f(x^{t+1})^{\mathrm{T}}(z - x^{t+1}) \leq \lim_{t \to \infty} \varepsilon_1^t = 0.$$

By the optimality of x^*, we obtain that $\nabla f(x^*)^{\mathrm{T}}(z - x^*) \geq 0$, which implies that $\nabla f(x^*)^{\mathrm{T}}(z - x^*) = 0$. If however the RMP is solved exactly, then $\nabla f(x^{t+1})^{\mathrm{T}}(z - x^{t+1}) = 0$ must hold, because otherwise it would have to be positive by the optimality of x^{t+1}, which in turn would imply by Proposition 3.2 that $\beta_z^t = 0$; this however contradicts our assumption that $\beta_z^t > 0$. In the limit of the above equality, then,

$$\nabla f(x^*)^{\mathrm{T}}(z - x^*) = \lim_{t \to \infty} \nabla f(x^{t+1})^{\mathrm{T}}(z - x^{t+1}) = 0.$$

Since x^* is geometrically stable, in either case we have established that $z \in F^*$.

This also proves that any element of the set $\cup_{t > \tau} \mathcal{P}^t$ which is not in the optimal face must be eliminated in some iteration. We first consider the case where $r = \infty$ holds. By the above, $y^t \in F^*$ for $t \geq \tau_1$. Hence, by the construction of the inner approximation, there exists an integer τ_2 such that $\mathcal{P}^t \subset F^*$, for $t \geq \tau_2$. We therefore obtain that $x^t \in X^t = \texttt{conv}(\mathcal{P}^t) \subset F^*$, $t \geq \tau_2$. The fact that the weights are positive then implies that actually $x^t \in \texttt{rint}\, F^*$.

In the case where $r < \infty$, it may be that there are iterations t in which an element x^t is introduced into $\mathcal{P}^t$. We however establish that this is impossible when $r \geq \texttt{dim}\, F^* + 1$ and the RMP are solved exactly. The conclusion is then the same as for the case when $r = \infty$.

Using the previous result, $\mathcal{P}_s^t \subset F^*$ for all $t \geq \tau_2$. This implies that $\texttt{dim}(\texttt{conv}(\mathcal{P}_s^t)) \leq \texttt{dim}\, F^*$. For future reference, let $\texttt{dim}(\texttt{conv}(\mathcal{P}_s^t)) = m$. As the RMP are solved exactly, X^t is a simplex by Theorem 3.6 for any $t \geq 0$, so $\texttt{conv}(\mathcal{P}_s^t)$ is an m-simplex by Proposition 3.5.c. Consider the column y^t generated for some iteration $t \geq \tau_2$. Since we assume that $x^t \notin \mathrm{SOL}(f, X)$, according to the proof of Theorem 3.6, $\texttt{conv}(\mathcal{P}_s^t \cup \{y^t\})$ is then an $(m+1)$-simplex; further, since $t \geq \tau_2$, $\mathcal{P}_s^t \cup \{y^t\} \subset F^*$ holds. It then follows that

$$|\mathcal{P}_s^t| = \texttt{dim}(\texttt{conv}(\mathcal{P}_s^t)) + 1 = \texttt{dim}(\texttt{conv}(\mathcal{P}_s^t \cup \{y^t\})) \leq \texttt{dim}\, F^* \leq r - 1,$$

which implies that $|\mathcal{P}_s^t| < r$. This, in turn, implies that $\mathcal{P}_x^{t+1} = \mathcal{P}_x^t$ holds for all $t \geq \tau_2$ (cf. Step 3.3.b). This completes the proof. □

When X is defined by constraints of the form $g_i(x) \leq 0$, $i = 1, \ldots, m$, and each function g_i is strictly convex, then the optimal face is a singleton. The result (b) then states that convergence is actually finite if the columns finitely lie in the optimal face.

We finally state a sufficient condition for $\hat{y}^t \in F^*$ to hold for every $t \geq \tau_1$. To this end, we introduce the following concept.

Definition 4.13 (Projected gradient). Let $x \in X$. The *projected gradient* at x is

$$\nabla^X f(x) := \arg \min_{\nu \in T_X(x)} \|\nabla f(x) + \nu\|. \tag{15}$$

□

Hence, the projected gradient at x equals $P_{T_X(x)}[-\nabla f(x)]$, where $P_S[\cdot]$ is the Euclidean projection mapping onto a convex set S. Note that by their definitions, $-\nabla f(x) \in N_X(x)$ holds if and only if $P_{T_X(x)}[-\nabla f(x)] = 0^n$. The following result shows that algorithms that force the projected gradient to zero characterizes those that identify the optimal face in a finite number of iterations. (In the application to polyhedral sets, we assume that the linear constraints all are inequalities, and let $\mathcal{I}(x)$ and λ_i^* denote the subset of the constraints that are active at x and their Lagrange multipliers, respectively.)

Theorem 4.14 (Identification characterization, [BuM88, BuM94]). *Assume that $\{z^t\} \subset X$ converges to $x^* \in \mathrm{SOL}(f, X)$.*

(a) *Assume further that* X *is a polyhedral set. Then, there exists an integer* τ *such that*

$$\{\nabla^X f(z^t)\} \to 0^n$$
$$\Longleftrightarrow$$
$$z^t \in E_X[-\nabla f(x^*)], \qquad t \geq \tau$$
$$\Longleftrightarrow$$
$$\mathcal{I}(z^t) = \{\, i \in \mathcal{I}(x^*) \mid \lambda_i^* > 0 \,\}, \qquad t \geq \tau.$$

(b) *Assume that* x^* *is non-degenerate. Further, assume that* $x^* \in \texttt{rint}\, F^*$ *holds, where the face* F^* *of* X *is quasi-polyhedral. Then, there exists an integer* τ *such that*

$$\{\nabla^X f(z^t)\} \to 0^n$$
$$\Longleftrightarrow$$
$$z^t \in \texttt{rint}\, F^*, \qquad t \geq \tau.$$

Assume further that X *is polyhedral. Then, to the above equivalence can be added the following:*

$$\mathcal{I}(z^t) = \mathcal{I}(x^*), \qquad t \geq \tau.$$

The immediate application of this result to our algorithm follows.

Theorem 4.15 (Finite identification of the optimal face). *Assume that Guignard's CQ holds. Further, assume that* $\mathrm{SOL}(f, X)$ *is a set of non-degenerate solutions. Let Assumption 4.11 hold, and further assume that* F^* *is geometrically regular. If the sequence* $\{\hat{y}^t\}$ *is such that* $\{\nabla^X f(\hat{y}^t)\} \to 0^n$ *holds, then there exists a positive integer* τ *such that* $x^t \in \texttt{rint}\, F^*$ *holds for every* $t \geq \tau$.

Proof. The result follows immediately by applying Theorems 4.10, 4.12.b and 4.14.b. □

Algorithms which force the projected gradient to zero include the gradient projection and sequential quadratic programming algorithms ([BuM88]). Patriksson [Pat98b, Theorem 7.11 and Instance 7.19] establishes the more general result that the corresponding sequence generated from the use of the subproblem $\mathrm{CDP}(\varphi(\cdot, x), \nabla f, X, x)$ defined in Section 2, forces the projected gradient to zero, under the additional assumption that $\varphi(\cdot, x)$ is strictly convex:

Theorem 4.16 (Finite identification of the optimal face). *Consider an arbitrary sequence* $\{z^t\} \subset X$. *Let* $\{y^t\}$ *be the corresponding sequence of solutions to the problem* $\mathrm{CDP}(\varphi(\cdot, z^t), \nabla f, X, z^t)$. *Then,*

$$\{y^t - z^t\} \to 0^n \qquad \Longrightarrow \qquad \{\nabla^X f(y^t)\} \to 0^n.$$

In particular, if the sequence $\{z^t\}$ *converges to an optimal solution to* $\mathrm{CDP}(f, X)$ *and* $\varphi(\cdot, z)$ *is strictly convex for every* $z \in X$, *then* $\{y^t - z^t\} \to 0^n$ *holds.* □

The immediate application of this result is of course to the NSD algorithm, which hence can be established to finitely attain the optimal face.

4.3 Finite identification of an optimal solution

The finite convergence property of the SD and RSD algorithms are based on the finiteness of the number of candidate columns that need to be generated in order to span the optimal face, by the finiteness of the number of extreme points of a polyhedron. This property is in general lost in the CG algorithm, due to the nonlinear character of the feasible set and/or the column generation problem. An example illustrates this fact.

EXAMPLE 4.17 (Asymptotic convergence of the CG algorithm). Consider the following instance of CDP(f, X):

$$\begin{aligned} \text{minimize} \quad & f(x_1, x_2) := \left(x_1 - \frac{1}{2}\right)^2 + x_2, \\ \text{subject to} \quad & -2x_1 - x_2 \leq -1, \\ & 2x_1 - x_2 \leq 1, \\ & x_2 \leq 1. \end{aligned}$$

Let the columns be constructed as follows. For a given feasible x, $y^{\mathrm{T}} = (-1/2+\sqrt{1+f(x)/2}, -2+2\sqrt{1+f(x)/2})$, the result of which is used in the construction of the inner approximation. For any feasible $x \neq x^* = (\frac{1}{2}, 0)^{\mathrm{T}}$, $f(y) = \frac{1}{2}f(x) < f(x)$ holds. Clearly, then, the conditions for the asymptotic convergence of the algorithm towards the unique solution x^* are satisfied. If, for some restriction X^t of the feasible set X, $x^* \in X^t$ holds, then x^* is an extreme point of X^t because x^* is an extreme point of X and $X^t \subset X$. We assume that the rule used in the set augmentation is 3.3.a. It then follows that $x^* \in X^t$ if and only if $y^t = x^*$. We will establish by induction that $x^* \notin X^t$ for any t, whence convergence must be asymptotic. For $t = 0$, assume that $X^0 = \{x^0\} \neq \{x^*\}$. We assume that $x^* \notin X^t$ for some $t \geq 0$. Using that the RMP is solved exactly, so $x^{t+1} \neq x^*$, it follows that $f(x^{t+1}) > 0$, and further that $f(y^{t+1}) > 0$ holds. But this implies that $y^{t+1} \neq x^*$, and using the previous argument $x^* \notin X^{t+1}$. This completes the argument. □

In order to establish the finite convergence of the CG algorithm we impose a property on the optimal solution set SOL(f, X) which is stronger than non-degeneracy and the regularity conditions given in Theorem 4.10. As we shall see, it will imply that the number of columns needed to span the optimal face is finite—in fact, the optimal face *equals* the optimal solution set—whence the result of Theorem 4.15 implies that convergence is finite.

The regularity condition we will employ is the following.

DEFINITION 4.18 (Weak sharp minimum, [Pol87]). The set SOL(f, X) is a set of *weak sharp minima* if for some $\alpha > 0$,

$$f(x) - f\left(P_{\mathrm{SOL}(f,X)}(x)\right) \geq \alpha \|x - P_{\mathrm{SOL}(f,X)}(x)\|, \qquad x \in X. \tag{16}$$

Polyak [Pol87] established that the gradient projection algorithm is finitely convergent under the weak sharp property. Burke and Ferris [BuF93] extended this result to characterize the algorithms for convex programs which finitely attain an optimal solution, while also extending the characterization in Theorem 4.14.b of those algorithms which finitely attain the optimal face:

THEOREM 4.19 (Finite convergence characterization, [BuF93, Theorem 4.7]). *Assume that f is convex and that* SOL(f, X) *is a set of weak sharp minima for* CDP(f, X). *Assume that $\{z^t\} \subset X$ converges to* SOL(f, X). *Then, there exists an integer τ such that*

$$\{\nabla^X f(z^t)\} \to 0^n$$
$$\Longleftrightarrow$$
$$z^t \in \mathrm{SOL}(f, X), \qquad t \geq \tau.$$

We utilize this theorem as follows.

THEOREM 4.20 (Finite identification of an optimal solution). *Assume that f is convex and that* $\mathrm{SOL}(f,X)$ *is a set of weak sharp minima for* $\mathrm{CDP}(f,X)$.

(a) [The NSD algorithm]. *Suppose that the sequence $\{x^t\}$ is the generated by the NSD algorithm, and that it converges to an optimal solution to* $\mathrm{CDP}(f,X)$. *Suppose further that $\varphi(\cdot,z)$ is strictly convex for every $z \in X$. Then, there exists an integer τ such that $x^t \in$* $\mathrm{SOL}(f,X)$, *for all $t \geq \tau$.*

(b) [The general algorithm]. *Suppose that* $\mathrm{SOL}(f,X)$ *is a regular face. Let Assumption 4.11 hold. If the sequence $\{\hat{y}^t\}$ is such that $\{\nabla^X f(\hat{y}^t)\} \to 0^n$ holds, then there exists a positive integer τ such that $x^t \in \mathtt{rint}\,\mathrm{SOL}(f,X)$ holds for every $t \geq \tau$.*

PROOF. (a) Combine Theorems 4.16 and 4.19.

(b) By the convexity of f, Theorem 4.1 of [BuF93] shows that the optimal face F^* equals the optimal solution set $\mathrm{SOL}(f,X)$, which furthermore is the face exposed by the vector $-\nabla f(x^*)$ for any $x^* \in \mathrm{SOL}(f,X)$. By assumption, $\mathrm{SOL}(f,X)$ is a geometrically regular face.

By hypothesis, $\{\nabla^X f(\hat{y}^t)\} \to 0^n$ holds. Under the weak sharpness and regularity assumptions, Theorem 4.19 then establishes that there exists an integer τ_1 such that $\hat{y}^t \in \mathrm{SOL}(f,X)$ for every $t \geq \tau_1$.

Theorem 4.12.b then implies the existence of an integer τ_2 such that $x^t \in \mathtt{rint}\,(\mathrm{SOL}(f,X))$ for all $t \geq \tau_2$. □

4.4 Finiteness properties extended to variational inequalities

4.4.1 Introduction

Consider the *variational inequality problem* of finding $x^* \in X$ such that

$$M(x^*)^{\mathrm{T}}(x - x^*) \geq 0, \qquad x \in X. \qquad [\mathrm{VIP}(M,X)]$$

where $M : X \mapsto \Re^n$ is continuous on X. Whenever $M = \nabla f$, $\mathrm{VIP}(M,X)$ constitutes the first-order optimality conditions of $\mathrm{CDP}(f,X)$. General properties of variational inequalities are, for example, found in [HaP90].

We establish in this final section that the finiteness properties of the CG algorithm are preserved when considering this more general problem. It is a nontrivial problem to establish even asymptotic convergence for classic descent algorithms when extended to VIP; counterexamples exist for the convergence of an extension of the Frank–Wolfe algorithm to VIP (e.g., [Ham84]), so a straightforward extension of RSD is *not* a convergent algorithm! (Without column dropping, SD *does* converge for VIP; there is also a convergence result in [LaH84] for a special RSD algorithm in which however no column dropping can be performed after a finite number of iterations.) So before moving on to establishing finite convergence, we first need to establish that there are instances of the general algorithm that have asymptotic convergence. To this end, we will cite results from [Pat98b, Sections 6.2.1, 9.4.1, 9.4.2] on what effectively is an extension of the NSD algorithm to VIP.

Assume that M is strongly monotone, in C^1 and Lipschitz continuous on X. Further, let the function φ (cf. Section 2.3 on the NSD algorithm) further be strictly convex in y for each fixed $x \in X$ and in C^1 on $X \times X$. Let $\alpha > 0$. We define the merit function

$$\psi_\alpha(x) := \underset{y \in X}{\operatorname{maximum}} \{ M(x)^{\mathrm{T}}(x - y) - (1/\alpha)\varphi(y,x) \}. \tag{17}$$

The function $\psi_\alpha : X \mapsto \Re^n$ clearly is a merit function for $\mathrm{VIP}(M, X)$, since $\mathrm{SOL}(M, X) = \arg\min_{x \in X} \psi_\alpha(x)$, and, further, $\psi_\alpha(x) = 0$ on $\mathrm{SOL}(M, X)$. It is furthermore in C^1 on X, by the strict convexity assumption on $\varphi(\cdot, x)$. Further, we define $\theta(x, \alpha) := -(1/\alpha)[\varphi(y, x) + \nabla_x \varphi(y, x)^{\mathrm{T}}(y - x)]$, where y is the vector that defines the value of $\psi_\alpha(x)$ [that is, which solves $\mathrm{CDP}((1/\alpha)\varphi(\cdot, x), M, X, x)$].

The algorithm of Table 3 is a special case of that in [Pat98b, Section 6.2.1].

TABLE 3: A descent CG algorithm for VIP

0. (*Initialization*): Choose an initial point $x^0 \in X$, let $\alpha_0 > 0$, $\Delta\alpha > 0$, and $\gamma \in (0, 1)$. Set $t := 0$.
1. (*Column generation*): Find a vector y^t that solves $\mathrm{CDP}(\varphi(\cdot, x^t), M, X, x^t)$.
2. (*Termination criterion*): If x^t solves $\mathrm{CDP}(\varphi(\cdot, x^t), M, X, x^t) \to$ Stop [$x^t \in \mathrm{SOL}(M, X)$]. Otherwise, continue.
3. (*Restricted master problem or null step*): If $\psi_{\alpha_t}(x^t) \leq \theta(x^t, \alpha_t)/(1 - \gamma)$, then let $\alpha_{t+1} := \alpha_t + \Delta\alpha$ and $x^{t+1} := x^t$; otherwise, let $\alpha_{t+1} := \alpha_t$, and let x^{t+1} be an arbitrary point in any closed and convex subset of X that contains the line segment $[x^t, y^t]$, and which also satisfies $\psi_{\alpha_t}(x^{t+1}) \leq \psi_{\alpha_t}(z^t)$ for some $z^{t+1} := x^t + \ell_t d^t$ satisfying the Armijo Rule.
4. (*Termination criterion*): If x^t is acceptable $\to$ Stop. Otherwise, go to Step 1 with $t := t+1$.

For this algorithm, we have the following result, combining [Pat98b, Theorem 6.15, Corollary 6.17, and Theorem 9.17]:

THEOREM 4.21 (Asymptotic convergence of a CG algorithm for VIP). *In the algorithm of Table 3, there exists a finite integer τ such that $\alpha_t = \bar{\alpha} > 0$ for all $t \geq \tau$. Therefore, after a finite number of iterations, the algorithm is a closed descent algorithm for* $\mathrm{VIP}(M, X)$, *to whose unique solution the sequence $\{x^t\}$ converges, and $\{\psi_{\alpha_t}(x^t)\} \to 0$.* □

This result establishes that a large class of closed CG algorithms, among which is the class of NSD algorithms, based on the monotonic decrease of a merit function for the variational inequality has asymptotic convergence. As an example instance, choosing $\varphi(y, x) := (1/2)(y - x)^{\mathrm{T}} Q (y - x)$ for some symmetric and positive definite matrix $Q \in \Re^{n \times n}$ reduces the above algorithm to a general multidimensional version of Fukushima's [Fuk92] gap minimization algorithm. Corollary 4.45 of [Pat98b] establishes an upper bound on $\bar{\alpha}$ to be $\|Q\|/(2m_M)$, where m_M is the modulus of strong monotonicity of M.

In order to establish the conclusions of the two main finiteness results, Theorems 4.15 and 4.20, also for applications to VIP, we will follow their proofs and discuss what needs to be changed or specialized. In our analysis, every definition of Section 3 and 4 which includes ∇f is generalized to VIP by replacing it with the mapping M. (Definition 4.18 will below be extended to this more general case by first considering an equivalent restatement.)

4.4.2 Finite identification of the optimal face

We first seek to reach the conclusion of Theorem 4.15. To this end, note first that every condition of the theorem is either kept as is or extended through the identification mentioned above. (The condition $\{\nabla^X f(\hat{y}^t)\} \to 0^n$ is replaced by $\{P_{T_X(\hat{y}^t)}[-M(\hat{y}^t)]\} \to 0^n$.)

Turning to the analysis of the proof of Theorem 4.15, Theorem 4.10 is immediately extended to the present situation. Second, noting that Proposition 3.2 also extends immediately, we can trace the proof of Theorem 4.12.b until we reach the stage where Theorem 3.6 is invoked.

Stopping to analyze this result in detail, we note first that if we specialize the result to the algorithm class defined in Table 3 (except of course for the crude column dropping rule present there, and the crude solution of the RMP to be replaced by the extension of Assumption 4.11 to VIP), then for each t, unless a solution to VIP(M, X) is at hand, $M(x^t)^{\mathrm{T}}(y^t - x^t) < 0$ must hold, since the merit function then is strictly positive and the assumptions on φ imply that $\varphi(y^t, x^t) \geq 0$. Tracing the proof of Theorem 3.6, we see that with ∇f replaced by M, (10) is established. Further, the above, together with (10), ensures that $y^t \notin \mathtt{aff}\,(\bar{X}^t)$, as desired. With this simple change, we can now reach the conclusion that the inner approximations are simplices also in the context of VIP(M, X). The remaining result to be studied, Theorem 4.14.b, was already in Patriksson [Pat98b, Corollary 7.10] established to immediately extend to the case of the VIP. This concludes the analysis of Theorem 4.15 in this more general setting.

We summarize the above development: the finite identification of the optimal face is ensured under the identical conditions for the cases of CDP and VIP, as long as the CG algorithm is based on NSD. We also note that for this particular result, no convexity (respectively, monotonicity) assumption on f (respectively, M) is necessary.

4.4.3 Finite identification of an optimal solution

In order to reach the conclusion of Theorem 4.20 in the setting of VIP(M, X), we begin by extending the concept of weak sharpness of the set SOL(f, X). Patriksson [Pat98b, Section 7.1.4] used an equivalent definition of weak sharpness stated in [BuF93] for the case of a convex function f, and used its extension to the VIP as a definition of weak sharpness of the set SOL(M, X), as follows: for any $x^* \in$ SOL(M, X),

$$-M(x^*) \in \mathtt{int} \bigcap_{x \in \mathrm{SOL}(M,X)} [T_X(x) \cap N_{\mathrm{SOL}(M,X)}(x)]^\circ. \tag{18}$$

Later, Marcotte and Zhu [MaZ98] used this definition to establish an extension of Theorem 4.19. Before turning to this result, we first, however, introduce a further assumption on the mapping M. Recall that Theorem 4.19 relies heavily on the invariance of ∇f on SOL(f, X) in the convex case. In order to extend the theorem to the case of VIP, Marcotte and Zhu [MaZ98, Theorem 4.3] establish that M is invariant on SOL(M, X), as desired, when M is pseudo-monotone$^+$ on X, that is, when M is pseudo-monotone on X and

$$\left.\begin{array}{l} F(y)^{\mathrm{T}}(x-y) \geq 0 \\ F(x)^{\mathrm{T}}(x-y) = 0 \end{array}\right\} \Longrightarrow F(x) = F(y), \qquad x, y \in X.$$

We next state an extension of Theorem 4.19.

Theorem 4.22 (Finite convergence characterization, [MaZ98, Theorem 5.2]). *Assume that M is pseudo-monotone$^+$ and that SOL(M, X) is a set of weak sharp solutions to VIP(M, X). Assume that $\{z^t\} \subset X$ converges to SOL(M, X). Then, there exists an integer τ such that*

$$\begin{gathered} \{P_{T_X(z^t)}[-M(z^t)]\} \to 0^n \\ \Longleftrightarrow \\ z^t \in \mathrm{SOL}(M, X), \qquad t \geq \tau. \end{gathered}$$

We are now ready to extend Theorem 4.20. The new conditions having been stated already, we turn to the proof. First, we replace [BuF93, Theorem 4.1] with [MaZ98, Theorem 4.3]. Next, we replace Theorem 4.19 with Theorem 4.22, concluding that the columns generated are optimal after a finite number of steps. Further, Theorem 4.16 was extended to VIP in [Pat98b, Corollary 7.12]. Finally, Theorem 4.12.b has been declared valid for the case of the VIP already above. This concludes the analysis of the theorem, which we hence have shown is valid also for the case of VIP, again, when the algorithm is based on NSD.

We note finally that Theorem 4.20 was established under a convexity assumption on f, whereas M was here assumed to be pseudo-monotone$^+$. When $M = \nabla f$, the latter is actually a milder assumption (take $f(x) := -x^2 - x$ and $X := [0, 1]$). It follows that we can replace convexity by pseudo-convexity$^+$ in Theorem 4.20. (Consequently, Theorem 4.19 can be thus extended also.)

5 A numerical example

We illustrate the performance of the new class of algorithms with the study of a medium-scale example. (The forthcoming paper [GMP02] addresses a more ambitious computational study.)

The problem considered is a single-commodity nonlinear network flow problem with separable link costs. In order to assess the efficiency and convergence properties of the new class of algorithms, instances of it were coded, and compared with simplicial decomposition. We coded the following algorithms:

SD Simplicial decomposition;

FW Frank–Wolfe;

N An instance of NSD (cf. Section 2.3) in which the new column is defined by the subproblem of Newton's method, the quadratic programming problem being solved inexactly by using 3 iterations of the Frank–Wolfe algorithm;

P An instance of NSD in which the new column is defined by the subproblem of the Goldstein–Levitin–Polyak gradient projection algorithm ([Gol64, LeP66]), the quadratic programming problem being solved inexactly by using 5 iterations of the Frank–Wolfe algorithm;

SD/SD An instance of the new algorithm in which 7 iterations of simplicial decomposition on the original problem is used to define the new column;

SD/FW An instance of the new algorithm in which 10 iterations of the Frank–Wolfe algorithm on the original problem is used to define the new column.

In all these cases, a prolongation according to (6) is performed from the respective subproblem solution to define the column stored. Further, each restricted master problem is solved using 5 iterations of the projected Newton method ([Ber82]).

The nonlinear single-commodity network flow problem (NSNFP) is defined by a directed graph $(\mathcal{N}, \mathcal{A})$ with n nodes and m links. For each node $i \in \mathcal{N}$ a scalar s_i is given, where s_i is the source or sink flow (imbalance) at node i, and for each link $(i, j) \in \mathcal{A}$ a convex and continuously differentiable function $f_{ij} : \Re_+ \mapsto \Re$ is given, as well as a flow x_{ij}, which is furthermore subject to an upper bound $u_{ij} > 0$. The nonlinear single-commodity network flow problem with separable link cost functions is

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad f(x) := \sum_{(i,j)\in\mathcal{A}} f_{ij}(x_{ij}), \\ &\text{subject to} \quad \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = s_i, \qquad i \in \mathcal{N}, \qquad\qquad \text{[NSNFP]} \\ &\qquad\qquad\quad 0 \le x_{ij} \le u_{ij}, \qquad (i,j) \in \mathcal{A}. \end{aligned}$$

The test network is shown in Figure 1. It consists of 22 nodes and 120 links. The link cost functions are $f_{ij}(x_{ij}) := x_{ij} \ln x_{ij}$ for all $(i, j) \in \mathcal{A}$. The parameters s_i are $s_O := 100$,

$s_D := -100$, and $s_i := 0$ for all $i \in \{1, \ldots, 20\}$. The link capacities are $u_{ij} := \infty$, that is, the example is a uncapacitated problem. The feasible region then is a polyhedron defined by 100 extreme points. The optimal solution is in the relative interior of the polyhedron and the optimal value of the problem is $f(x^*) = 200 \ln(10)$. This value is used to calculate the relative error of the result provided by the algorithms.

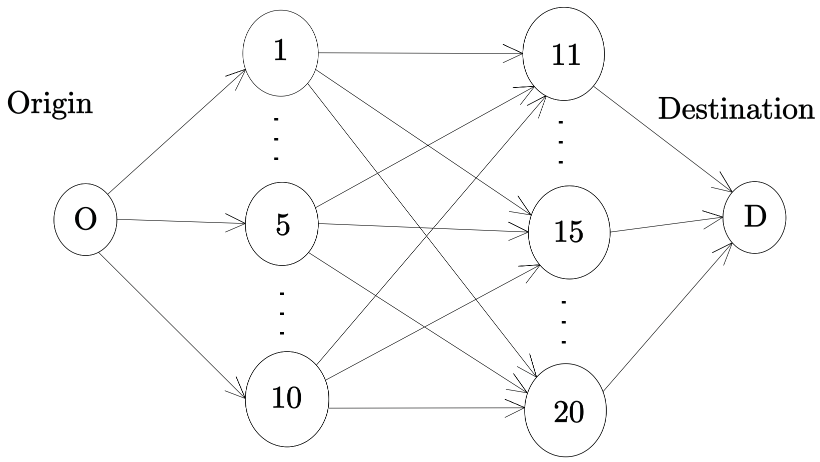

FIGURE 1: A traffic network.

The algorithms were coded in the Fortran Visual Workbench Programming language, and executed on a PC with 384 Megabytes of RAM and a 400 MHz clock processor. The NSNFP was solved with predefined tolerances, measured by the relative errors 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}; the relative error is defined by $(f(x) - f(x^*))/f(x^*)$ for any given feasible flow x.

TABLE 4: Computational results

Accuracy		SD	FW	N	P	SD/SD	SD/FW
	CPU	5.66	0.61	5.87	0.39	0.66	0.44
10^{-1}	Iter.	64	69	63	16	10	9
	Columns	64	1	63	14	10	9
	CPU	17.80	0.88	17.63	0.66	1.04	0.49
10^{-2}	Iter.	98	102	96	22	15	11
	Columns	98	1	96	20	15	11
	CPU	20.38	1.54	20.21	0.88	1.10	0.60
10^{-3}	Iter.	103	168	101	26	16	13
	Columns	100	1	98	24	16	13
	CPU	23.01	4.67	20.70	1.76	1.43	0.88
10^{-4}	Iter.	108	486	102	38	20	17
	Columns	100	1	98	36	20	17
	CPU	25.65	37.13	21.75	3.41	2.69	1.43
10^{-5}	Iter.	113	4221	104	51	32	25
	Columns	100	1	100	49	32	25

The computational results are shown in Table 4. For each tolerance used, the first row gives the CPU time in seconds, the second one the number of main iterations (or, restricted master problems), and the last row gives the number of columns stored in the last RMP. To compute the total number of extreme points generated during the execution of an algorithm (a number which is equal to the number of linear subproblems solved in the Frank–Wolfe algorithm, which are shortest path calculations), one multiplies the number of iterations by the number of extreme points generated in each iteration. For example, if in SD/SD 32 main iterations have been used, and in each of them 7 extreme points are generated, then the total number of columns generated will be 224.

The conclusion is that the algorithms SD/FW and SD/SD are much more efficient than SD for solving this problem instance. The new algorithms solve a lower number of restricted master problems, using a smaller number of variables. SD solves 113 RMPs, and the last ones have a size of 100 variables. In these last iterations, the RMPs coincide with the original problem, in contrast to SD/FW, which uses only 25 RMPs, with a maximum size of 25 columns in defining a solution with a similar quality.

The number of extreme points generated in each algorithm depends on the accuracy required. For example, the SD and the SD/SD algorithms, for the accuracies of 10^{-1}, 10^{-2}, and 10^{-3}, generate almost the same number of extreme points: 64/90, 98/105 and 103/102, respectively, so the computational cost in the subproblem phases is roughly the same. The SD/FW algorithm uses somewhat more extreme points (250), but the greater computational cost in the column generation problem is compensated by a large reduction in the computational cost in the RMP.

In relation with the NSD algorithms N and P, the improvement can be explained by the fact that the Frank-Wolfe algorithm is applied directly on the original problem and not on a quadratic approximation. The behaviour of the algorithm N during the first iterations is due to the ill-conditioning of the Hessian matrix.

Acknowledgement

The authors would like to thank an anonymous referee for pointing out a mistake in a proof, and for suggesting several improvements in the presentation.

References

[AuF90] J.-P. Aubin and H. Frankowska, *Set-Valued Analysis*, Birkhäuser, Boston, MA, 1990.

[Bar+98] C. Barnhart, E. L. Johnson, G. L. Nemhauser, G. L. Savelsberg, and P. H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Operations Research, 46 (1998), pp. 316–329.

[BSS93] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, New York, NY, second ed., 1993.

[Ber82] D. P. Bertsekas, *Projected Newton methods optimization problems with simple constraints*, SIAM Journal on Control and Optimization, 20 (1982), pp. 221–246.

[BuF91] J. V. Burke and M. C. Ferris, *Characterization of solution sets of convex programs*, Operations Research Letters, 10 (1991), pp. 57–60.

[BuF93] J. V. Burke and M. C. Ferris, *Weak sharp minima in mathematical programming*, SIAM Journal on Control and Optimization, 31 (1993), pp. 1340–1359.

[BuM88] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM Journal on Numerical Analysis, 25 (1988), pp. 1197–1211.

[BuM94] J. V. Burke and J. J. Moré, *Exposing constraints*, SIAM Journal on Optimization, 4 (1994), pp. 573–595.

[DaW60] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operations Research, 8 (1960), pp. 101–111.

[DeT88] R. S. Dembo and U. Tulowitzki, *Computing equilibria on large multicommodity networks: An application of truncated quadratic programming algorithms*, Networks, 18 (1988), pp. 273–284.

[Dun87] J.C. Dunn, *On the convergence of projected gradient processes to singular critical points*, Journal of Optimization Theory and Applications, 55 (1987), pp. 203–216.

[DuM89] J.P. Dussault and P. Marcotte, *Conditions de régularité géometrique pour les inéquations variationnelles*, Recherche opérationelle, 23 (1989), pp. 1–16.

[Eva76] S. P. Evans, *Derivation and analysis of some models for combining trip distribution and assignment*, Transportation Research, 10 (1976), pp. 37–57.

[Fuk92] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Mathematical Programming, 53 (1992), pp. 99–110.

[GaM97a] R. García and A. Marín, *Urban multimodal interchanges (Macro Vision)*, conference presentation at the EURO XV–INFORMS XXXIV Joint International Meeting, Barcelona, July 14–17, 1997.

[GaM97b] R. García and A. Marín, *Using RSD within partial linearization methods*, conference presentation at the 16th International Symposium on Mathematical Programming, Lausanne, August 24–29, 1997.

[GMP02] R. García, A. Marín, and M. Patriksson, *Column generation algorithms for nonlinear optimization, II: Numerical investigations*, report, Department of Mathematics, Chalmers University of Technology, Gothenburg, Sweden, 2001 (to appear).

[GHV92] J.-L. Goffin, A. Haurie, and J.-P. Vial, *Decomposition and nondifferentiable optimization with the projective algorithm*. Management Science, 37 (1992), pp. 284–302.

[Gol64] A. A. Goldstein, *Convex programming in Hilbert space*, Bulletin of the American Mathematical Society, 70 (1964), pp. 709–710.

[Gui69] M. Guignard, *Generalized Kuhn–Tucker conditions for mathematical programming problems in a Banach space*, SIAM Journal on Control, 7 (1969), pp. 232–241.

[Ham84] J. H. Hammond, *Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms*, PhD thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[HaP90] P. T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Mathematical Programming, 48 (1990), pp. 161–220.

[HLV85] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura, *Finiteness in restricted simplicial decomposition*, Operations Research Letters, 4 (1985), pp. 125–130.

[HLV87] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura, *Restricted simplicial decomposition: Computation and extensions*, Mathematical Programming Study, 31 (1987), pp. 99–118.

[HiP90] J. E. Higgins and E. Polak, *Minimizing pseudoconvex functions on convex compact sets*, Journal of Optimization Theory and Applications, 65 (1990), pp. 1–27.

[Hol74] C. A. Holloway, *An extension of the Frank and Wolfe method of feasible directions*, Mathematical Programming, 6 (1974), pp. 14–27.

[HoJ61] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the Association for Computer Machinery, 8 (1961), pp. 212–229.

[JTPR94] R. Jayakrishnan, W. K. Tsai, J. N. Prashker, and S. Rajadhyaksha, *Faster path-based algorithm for traffic assignment*, Transportation Research Record, 1443 (1994), pp. 75–83.

[JLF93] K. L. Jones, I. J. Lustig, J. M. Farvolden, and W. B. Powell, *Multicommodity network flows: The impact of formulation on decomposition*, Mathematical Programming, 62 (1993), pp. 95–117.

[KiN91] K. Kim and J. L. Nazareth, *The decomposition principle and algorithms for linear programming*, Linear Algebra and Its Applications, 152 (1991), pp. 119–133.

[LMP94] T. Larsson, A. Migdalas, and M. Patriksson, *A generic column generation scheme*, Report LiTH-MAT-R-94-18, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1994. Revised version, July 1999.

[LaP92] T. Larsson and M. Patriksson, *Simplicial decomposition with disaggregated representation for the traffic assignment problem*, Transportation Science, 26 (1992), pp. 4–17.

[LPR97] T. Larsson, M. Patriksson, and C. Rydergren, *Applications of simplicial decomposition with nonlinear column generation to nonlinear network flows*, in Network Optimization, P. M. Pardalos, W. W. Hager, and D. W. Hearn, eds., Springer-Verlag, Berlin, 1997, pp. 346–373.

[Las70] L. S. Lasdon, *Optimization Theory for Large Systems*, Macmillan, New York, NY, 1970.

[LaH84] S. Lawphongpanich and D. W. Hearn, *Simplicial decomposition of the asymmetric traffic assignment problem*, Transportation Research, 18B (1984), pp. 123–133.

[LeP66] E. S. Levitin and B. T. Polyak, *Constrained minimization methods*, USSR Computational Mathematics and Mathematical Physics, 6 (1966), pp. 1–50.

[Lue84] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, second ed., 1984.

[LuP98] J. T. Lundgren and M. Patriksson, *An algorithm for the combined distribution and assignment model*, in Transportation Networks: Recent Methodological Advances, Selected Proceedings of the 4th EURO Transportation Meeting, Newcastle University, Newcastle, UK, September 9–11, 1996, M. G. H. Bell (ed.), Pergamon Press, Amsterdam, 1998, pp. 239–253.

[MaD89] P. Marcotte and J.-P. Dussault, *A sequential linear programming algorithm for solving monotone variational inequalities*, SIAM Journal on Control and Optimization, 27 (1989), pp. 1260–1278.

[MaZ98] P. Marcotte and D. Zhu, *Weak sharp solutions of variational inequalities*, SIAM Journal on Optimization, 9 (1998), pp. 179–189.

[Mig94] A. Migdalas, *A regularization of the Frank–Wolfe method and unification of certain nonlinear programming methods*, Mathematical Programming, 65 (1994), pp. 331–345.

[Pat93a] M. Patriksson, *Partial linearization methods in nonlinear programming*, Journal of Optimization Theory and Applications, 78 (1993), pp. 227–246.

[Pat93b] M. Patriksson, *A unified description of iterative algorithms for traffic equilibria*, European Journal of Operational Research, 71 (1993), pp. 154–176.

[Pat98a] M. Patriksson, *Cost approximation: A unified framework of descent algorithms for nonlinear programs*, SIAM Journal on Optimization, 8 (1998), pp. 561–582.

[Pat98b] M. Patriksson, *Nonlinear Programming and Variational Inequality Problems—A Unified Approach*, Kluwer Academic Publishers, Utrecht, The Netherlands, 1998.

[Pol87] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, NY, 1987.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[RoW98] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

[Rud76] W. Rudin, *Principles of Mathematical Analysis*, Third Edition, McGraw-Hill, Auckland, 1976.

[SaW79] G. Salinetti and R.-B. Wets, *On the convergence of sequences of convex sets in finite dimensions*, SIAM Review, 21 (1979), pp. 18–33.

[Tse91] P. Tseng, *Decomposition algorithm for convex differentiable minimization*, Journal of Optimization Theory and Applications, 70 (1991), pp. 109–135.

[VeH93] J. A. Ventura and D. W. Hearn, *Restricted simplicial decomposition for convex constrained problems*, Mathematical Programming, 59 (1993), pp. 71–85.

[vHo77] B. Von Hohenbalken, *Simplicial decomposition in nonlinear programming algorithms*, Mathematical Programming, 13 (1977), pp. 49–68.

[Wol70] P. Wolfe, *Convergence theory in nonlinear programming*, in Integer and Nonlinear Programming, J. Abadie (ed.), North-Holland, New York, NY, 1970, pp. 1–36.

[Wol98] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, New York, NY, 1998.

[Zan69] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[ZhM95] D. L. Zhu and P. Marcotte, *Coupling the auxiliary problem principle with descent methods of pseudoconvex programming*, European Journal of Operational Research, 83 (1995), pp. 670–685.