SIMPLICIAL DECOMPOSITION ALGORITHMS

Simplicial decomposition (SD) is a class of methods for solving continuous problems in mathematical programming with convex feasible sets. There are two main characteristics of the methods in this class: (i) an approximation of the original problem is constructed and solved, wherein the original feasible set is replaced by a polyhedral subset thereof, that is, an *inner approximation* of it which is spanned by a finite set of feasible solutions; and (ii) this inner approximation is improved (that is, enlarged) by generating a vector (or, column) in the feasible set through the solution of another approximation of the original problem wherein the original cost function is approximated (often by a linear function). As such, the class of SD methods may be placed within the framework of *column generation* methods. Another characteristic of an SD method however is that the sequence of solutions to the inner approximated problems tends to a solution to the original problem in such a way that the cost function (or, some merit function) strictly monotonically approaches its optimal value. Therefore, the class of SD methods also falls within the framework of iterative descent (or, ascent) algorithms for continuous mathematical programs.

We consider, for the most part, the solution of the differentiable optimization problem

$$\text{minimize} \quad f(x), \tag{1a}$$

$$\text{subject to} \quad x \in X, \tag{1b}$$

where $f : X \mapsto \Re$ is *pseudo-convex* on X [that is, for any $x, y \in X$, $\nabla f(x)^{\mathrm{T}}(y - x) \geq 0$ implies $f(x) \leq f(y)$], and where $X := \{\, x \in \Re^n \mid Ax = b; \quad x \geq 0^n \,\}$ is a non-empty polyhedral set.

The derivation of the method rests on two classical results on the representation of convex sets and of points in such sets. The first result is the *Representation Theorem* (e.g., [2, 19]), which states that (i) the set of extreme points p^i, $i \in \mathcal{P}$, of the polyhedral set X is non-empty and finite; (ii) the set of extreme directions d^i, $i \in \mathcal{D}$, is empty if and only if X is bounded, and if X is not bounded, then it is non-empty and finite; finally, and most importantly, (iii) a vector $x \in \Re^n$ belongs to X if and only if it can be represented as a convex combination of the extreme points plus a non-negative linear combination of the extreme directions, that is, for some vectors λ and μ,

$$x = \sum_{i \in \mathcal{P}} \lambda_i p^i + \sum_{i \in \mathcal{D}} \mu_i d^i, \tag{2a}$$

$$\sum_{i \in \mathcal{P}} \lambda_i = 1, \tag{2b}$$

$$\lambda_i \geq 0, \qquad i \in \mathcal{P}, \tag{2c}$$

$$\mu_i \geq 0, \qquad i \in \mathcal{D}. \tag{2d}$$

Thus, in principle, the polyhedral set X can be given an inner representation in terms of extreme points and directions, and the problem (1) can be cast in the variables λ_i and μ_i instead of in x. The advantage of making this problem transformation is that the inner representation of X is much simpler than its original, *outer*, representation in terms of linear equalities and inequalities; disregarding the definitional constraints (2a), the set described by (2) is the Cartesian product of a *simplex* and the non-negative orthant, an optimization over which often can be made with little more effort than for an unconstrained problem (e.g., [3, 4]). Furthermore, the inner representation may also be useful for interpreting the result of an optimization, since the extreme points and directions may have further significance; in applications to network flows, for example, the Representation Theorem states that a link flow $x \in X$, where the polyhedral set X describes the flow conservation and non-negativity constraints for the network flow, can equivalently be represented by (or, decomposed into) the sum of flows on routes and in cycles (e.g., [1, Thm. 3.5]). The

---

inner approximation
column generation
pseudo-convex → *pseudo-convex function*
Representation Theorem
simplex

latter also explains the origin of the term *simplicial decomposition*: the variable transformation decomposes a feasible solution into the sum of variables that (in the bounded case) forms a simplex.

The disadvantage of the transformation is that since the number of extreme points and directions of a polyhedral set grows exponentially with its dimension, the transformation introduces an impractically large number of variables. The practical use of simplicial decomposition then hinges on the second basic result in the representation of convex sets, *Carathéodory's Theorem* (e.g., [27, Thm. 17.1]). This result states that a point x 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. (This number is not larger than $n+1$.) Although Carathéodory's Theorem is not stated in terms of extreme points and directions, its natural application in the context of simplicial decomposition is that, in the case of a bounded polyhedral set, for example, any feasible point can be described as the convex combination of extreme points of the set, the total number of which need never exceed the dimension of the polyhedron plus one. (This result obviously refines the Representation Theorem.)

The classical form of the simplicial decomposition method was first described by von Hohenbalken [32] (see however the end of the article for some earlier references to similar algorithms) for the problem (1). The algorithm alternates between the solution of two problems. 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 (2), in terms of the variables $\hat{\lambda}_i$, $i \in \hat{\mathcal{P}}$ and $\hat{\mu}_i$, $i \in \hat{\mathcal{D}}$. [We will denote this problem the *restricted master problem* (*RMP*); it is also sometimes referred to as the *coordination* step.] 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 (1) and is defined by the system (2). Further denoting by $\hat{\Lambda}$ the set of vectors $(\hat{\lambda}, \hat{\mu})$ satisfying the restriction of the system (2b)–(2c) to the known subsets $\hat{\mathcal{P}}$ and $\hat{\mathcal{D}}$ and utilizing (2a) to substitute x for $(\hat{\lambda}, \hat{\mu})$ [we write $x = x(\hat{\lambda}, \hat{\mu})$], the RMP may then be formulated as

$$\text{minimize} \quad f(x(\hat{\lambda}, \hat{\mu})), \tag{3a}$$

$$\text{subject to} \quad (\hat{\lambda}, \hat{\mu}) \in \hat{\Lambda}. \tag{3b}$$

Alternately, a profitable extreme point or direction of X is generated through the solution of an approximation of (1), in which f is replaced by its first-order, linear approximation, $y \mapsto f(x) + \nabla f(x)^{\mathrm{T}}(y - x)$, defined at the solution, x, to the RMP (3), that is, by the problem

$$\text{minimize} \quad \nabla f(x)^{\mathrm{T}} y, \tag{4a}$$

$$\text{subject to} \quad y \in X; \tag{4b}$$

this approximate problem is a linear programming problem, which in general is much easier to solve than the original one. (This is called the *column generation subproblem*, and corresponds to the *decomposition* step in some descriptions of column generation methods.) If the solution to this problem lies within the current inner approximation, then the conclusion is that the current solution, x, is optimal in (1), since, then, $\nabla f(x)^{\mathrm{T}}(y-x) \geq 0$ must hold for all $y \in X$. Otherwise, $\hat{\mathcal{P}}$ or $\hat{\mathcal{D}}$ is augmented by a new element, the resulting inner approximation is improved (that is, enlarged), and the solution to the new RMP has a strictly lower objective value than the previous one; the latter result follows since the strict inequality $\nabla f(x)^{\mathrm{T}} d < 0$ holds (that is, d defines a direction of descent with respect to f at x), where d denotes either the direction $d := y - x$ towards the new extreme point y or an

simplicial decomposition
Carathéodory's Theorem
restricted master problem
coordination
column generation subproblem
decomposition

extreme direction. The iteration is then repeated with the solution of a new column generation subproblem defined at the solution to the RMP. In the method of [32], Carathéodory's Theorem is utilized in the validation of a *column dropping rule*, according to which any extreme point or direction whose weight in the expression of the solution x to the RMP is zero is removed; thanks to the finiteness of $\mathcal{P}$ and $\mathcal{D}$ and the strictly decreasing values of f, the convergence of the SD algorithm in the number of RMP is finite. (In the case of non-polyhedral sets, it was observed in [12] that von Hohenbalken's original procedure does not necessarily converge. Their remedy is the introduction of a safe-guarding step which ensures that each iteration reduces the objective function value at least as much as an exact line search would produce in the Frank–Wolfe algorithm. With this modification, the algorithm converges over general compact and convex sets, however, of course, not finitely.)

In the case of a convex function f, a finite termination criterion is automatically supplied, on the one hand by the upper bound on the optimal objective value of (1) that is defined by the solution to the RMP, and on the other hand by the lower bound that is supplied by the solution to the column generation subproblem; in fact, letting x be an arbitrary feasible solution to (1), and x^* be any optimal solution to (1), we obtain from the optimality of x^* and by the convexity of f that

$$\begin{aligned} f(x) \geq f(x^*) &\geq f(x) + \nabla f(x)^{\mathrm{T}}(x^* - x) \\ &\geq f(x) + \min_{y \in X} \{\nabla f(x)^{\mathrm{T}}(y - x)\}; \end{aligned} \qquad (5)$$

note that the problem defined in (5) is precisely the column generation subproblem (4) defined at x. A finite termination criterion for the solution of the RMP can be defined by the analogous lower bound. In this case, due to the simple form of the constraints defined by (2b)–(2d), the lower bound is available directly from the value of the gradient of f with respect to $(\hat{\lambda}, \hat{\mu})$ at $x = x(\hat{\lambda}, \hat{\mu})$. Indeed, the lower bound is either [analogously to (5)] given by the current objective value plus

$$\begin{aligned} &\min_{i \in \hat{\mathcal{P}}} \left\{ \frac{\partial f(x(\hat{\lambda}, \hat{\mu}))}{\partial \hat{\lambda}_i} \right\} \\ &- \nabla_{\hat{\lambda}} f(x(\hat{\lambda}, \hat{\mu}))^{\mathrm{T}} \hat{\lambda} - \nabla_{\hat{\mu}} f(x(\hat{\lambda}, \hat{\mu}))^{\mathrm{T}} \hat{\mu} \end{aligned}$$

or it is minus infinity (if $\nabla f(x)^{\mathrm{T}} d^i < 0$ holds for some $i \in \hat{\mathcal{D}}$).

Assume now that the function f is *strictly pseudo-convex* (that is, for any $x, y \in X$ with $x \neq y$, $\nabla f(x)^{\mathrm{T}}(y - x) \geq 0$ implies that $f(x) < f(y)$ holds), so that the optimal solution x^* is unique, and for simplicity we also assume that X is bounded. For such problems, an improvement over the original scheme was devised by Hearn et al. [10, 11]. The basis for the improvement is the observation that a particular feasible solution, such as the optimal one, can be represented as the convex combination of an often much smaller number of extreme points than $\mathtt{dim}\, X + 1$, as implied by Carathéodory's Theorem; in fact, the highest number of extreme points needed to describe the optimal solution x^* is $\mathtt{dim}\, F^* + 1$, where F^* is the *optimal face* of X, that is, the face of X of the smallest dimension which contains x^*. [In the present context, 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 (4) to (1) defined at the optimal solution.] Based on this observation, they devise a modification of the original scheme, 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). They show that the modified algorithm is finitely convergent in the number of RMP, provided that

optimal face

$r \geq \texttt{dim}\, F^* + 1$. Referred to as restricted simplicial decomposition (RSD), the scheme is shown below, for the case of a bounded set X.

PROCEDURE RSD (r)
 (Init): $x^0 \in X$, $p^x := x^0$, $\hat{\mathcal{P}} = \emptyset$, $t := 0$.
 (Sub): Solve (4) defined at $x^t \Rightarrow$
  p^{i_t}, $i_t \in \mathcal{P}$.
 (Augment): $i_t \in \hat{\mathcal{P}} \Rightarrow x^t$ is optimal.
  $|\hat{\mathcal{P}}| = r \Rightarrow$ replace an element of $\hat{\mathcal{P}}$ by i_t.
  $|\hat{\mathcal{P}}| < r \Rightarrow \hat{\mathcal{P}} := \hat{\mathcal{P}} \cup \{i_t\}$.
 (Master): x^{t+1} minimizes f over the
  convex hull of p^x and p^i, $i \in \hat{\mathcal{P}}$.
 (Update): Let $p^x := x^{t+1}$, $t := t + 1$.
  Go to the Subproblem.
END

The value of r is crucial to the performance of the algorithm. If $r \geq \texttt{dim}\, F^* + 1$, then since the number of RMP is finite, 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 ([11]). If $r < \texttt{dim}\, F^* + 1$, however, then the algorithm is only asymptotically convergent, and the rate of convergence is the same as that of the *Frank–Wolfe* algorithm (or, *conditional gradient* method) [which is actually obtained as a special case of RSD when $r := 1$], that is, the convergence rate is sub-linear. Since the threshold value for finite convergence cannot be estimated from the original data and thus is unknown a priori, the proper value of r must in general be based on computational experience.

In column generation methods for linear programs, such as *Dantzig–Wolfe decomposition* ([6]), the columns are generated through the *pricing* operation of the *simplex method* in linear programming, which utilizes an estimate of the dual optimal solution. Its extension to non-linear programming, *non-linear Dantzig–Wolfe decomposition* (e.g., [19]), also utilizes the pricing operation in the construction of the column generation subproblem, which in both cases is equivalent to the result of performing a Lagrangean relaxation of the original problem using the current estimate of the vector of Lagrange multipliers; while the column generation subproblem is non-linear in the latter algorithm, the RMP are in both cases linear programs. In contrast, the class of SD algorithms are column generation methods where the columns are generated through the solution of the primal, linearized problem (4), and which thus does not utilize dual information. However, it is established in [15] that Dantzig–Wolfe decomposition is in fact a special case of simplicial decomposition, when the latter is applied to a primal–dual (saddle point) reformulation of the linear program. Also for linearly constrained non-linear programs of the form (1), simplicial decomposition may be based on the *pricing-out* of a subset of the linear constraints. Identifying a subset of the constraints defining X as complicating, these may be priced-out (that is, Lagrangean relaxed) in the column generation subproblem, and instead included in the master problem, just as in Dantzig–Wolfe decomposition for linear and non-linear programming problems. Such methods have been devised in [21, 29]. It should be noted, however, that just as in the original (primal) SD method, the column generation subproblems in these methods are based on the *linearization* of the original objective function, and are therefore linear programs, and their RMP are non-linear; this is precisely the opposite to the case of non-linear Dantzig–Wolfe decomposition.

The RSD algorithm has been successfully applied to large-scale, structured non-linear optimization problems, in particular mathematical

Frank–Wolfe
conditional gradient
Dantzig–Wolfe decomposition
pricing
simplex method
non-linear Dantzig–Wolfe decomposition
pricing-out
linearization

programming models of various non-linear network flow problems, where the column generation subproblem reduces to efficiently solvable linear network flow problems (e.g., [11, 16, 24]).

Other special structures in the feasible set X may also be taken into account efficiently in the construction of an SD method. For example, assume that the set X is a Cartesian product of polyhedral sets X_k in smaller dimensions $\Re^{n_k}$, with $k \in \mathcal{K}$ and $\sum_{k \in \mathcal{K}} n_k = n$. (In network flow problems, k could denote a commodity of goods to be transported or a pair of origin and destination in an urban transportation network.) Noting that the linear column generation subproblem decomposes into $|\mathcal{K}|$ independent linear column generation problems, it is possible to store extreme points and directions of the individual (smaller-dimensional) sets X_k rather than extreme points and directions of X. The RMP of such a *disaggregate simplicial decomposition* (*DSD*) method ([16]) would then have variables of the form λ_{ik}, $i \in \mathcal{P}_k$, $k \in \mathcal{K}$, and likewise for μ_{ik}, and $|\mathcal{K}|$ convexity constraints (2b) instead of only one as in the SD method. The total number of extreme points and directions is much less in the disaggregated representation [$\sum_{k \in \mathcal{K}} \{\mathcal{P}_k + \mathcal{D}_k\}$ in the disaggregated case, and $\prod_{k \in \mathcal{K}} \{\mathcal{P}_k + \mathcal{D}_k\}$ in the aggregated case ([25])]. On the other hand, according to Carathéodory's Theorem, the total number of columns needed to express an optimal solution is in this case bounded above by $\sum_{k \in \mathcal{K}} (\mathtt{dim}\, F_k^* + 1)$, which may be a much higher number than $\mathtt{dim}\, F^* + 1$. This result notwithstanding, it has been observed in applications of Dantzig–Wolfe decomposition to linear multi-commodity network flow problems ([14]) that a disaggregated representation of the solution (in this case, as commodity route flows instead of as aggregated link flows) speeds up the convergence of the method. The same conclusion has been drawn from applications to non-linear multi-commodity network flow problems ([16]) of the DSD algorithm.

Experience with the RSD method has shown that it makes rapid progress initially, quickly reaching a near-optimal solution, 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 $\mathtt{dim}\, F^*$.

The explanation for this behaviour is to be found in the construction of the column generation subproblem, which utilizes first-order approximations of f. The column generation subproblem of RSD is the same as that of the Frank–Wolfe (FW) method mentioned earlier, the quality of whose search directions are known to deteriorate rapidly. The reason is that as the sequence $\{x^t\}$ tends to an optimal solution, 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 orthogonal to the gradient of f, and the result of the deteriorating descent property is a decreasing step length in the line search of this algorithm.

We then make the observation that the RSD method is similar to FW, since the same descent direction-generating subproblem is used, and that the only difference between FW and RSD lies in the updating phase, the latter algorithm using a multi-dimensional search (when $r \geq 2$) rather than a one-dimensional search, which one may interpret as a devise for reducing the zig-zagging effect inherent in the FW algorithm. It is a natural conclusion from this discussion that better approximations of f could be exploited in the column generation phase of SD methods, since then the columns generated would be of better quality, thus leading to larger improvements in the inner approximations of the feasible set. [The counterpart in line search methods for (1) is that improved approximations of f in a direction-finding subproblem yield better search directions.]

An extension of the RSD algorithm was made by Larsson, Patriksson and Rydergren [17] based on this observation. The motivation behind the *non-linear simplicial decomposition* (*NSD*) method is that by generating columns based

disaggregate simplicial decomposition
non-linear simplicial decomposition

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.

The NSD method is obtained from the RSD method by replacing the linear column generation subproblem (4) with

$$\min_{y \in X} \{\nabla f(x^t)^{\mathrm{T}} y + \varphi(y, x^t)\}, \tag{6}$$

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 property that $\nabla_y \varphi(x, x) = 0^n$ for all $x \in X$. Among the possible choices for φ we mention the following, where `diag` denotes the diagonal part of the matrix and where $\gamma_t > 0$:

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

Even though the finite convergence property will be lost (because non-extremal points will be generated), one may expect a more rapid convergence of the NSD method than the RSD method, both in terms of the number of iterations needed to reach a given solution accuracy and in terms of the required solution time, provided however that the non-linear column generation subproblems can be efficiently solved, at least approximately. In numerical experiments performed on large-scale non-linear network flow problems, such conclusions were indeed made. 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.

Convergence results for SD methods allow for both the column generation subproblem and the RMP to be solved inexactly, thus facilitating its practical use. In [11], convergence is established for the RSD method, wherein the RMP is solved using one iteration of Newton's method only. Further developments along these lines are found in [15, 8] for a general class of SD methods and in [26, Chap. 9] for the NSD method. The convergence results established in the latter reference not only validate inexact computations but also quite arbitrary rules both for defining and for dropping columns.

SD algorithms have been extended to handle non-linear constraints as well. In [31], the column generation subproblem is made linear by approximating the non-linear constraint functions by piece-wise linear functions, reminiscent to the Topkis–Veinott scheme ([30]). The NSD method of [17, 26] applies to general convex sets X directly. A combination of sequential quadratic programming (SQP) and NSD is devised in [26, Chap. 9]. There, in the column generation subproblem one replaces the non-linear constraints with linear approximations, and dual information about these constraints is included in the objective function. SD methods for non-linearly constrained problems are believed to be efficient, if the non-linearity is mild, as concluded from the computational results of [31]; however, few applications of SD methods to non-linearly constrained problems have been reported.

The class of SD methods has furthermore been extended to the solution of the general *variational inequality problem* (*VIP*) of finding $x^* \in X$ such that

$$F(x^*)^{\mathrm{T}}(x - x^*) \geq 0, \qquad \forall x \in X, \tag{7}$$

where $F : X \mapsto \Re^n$ is a continuous and monotone mapping. If the mapping F satisfies $F \equiv \nabla f$, that is, it is the gradient of f, then the VIP defines the first-order optimality conditions for x^* in (1), and the SD method for the VIP becomes that for (1). If this is not the case, then F replaces ∇f in the column generation subproblem (4), and the RMP is defined as the restriction of the variational inequality problem (7) to the currently known inner approximation of X. Further, in this case there is no objective function (or, *merit function*) immediately available for monitoring the convergence of the SD

variational inequality problem
merit function

method. Column dropping rules in SD methods for the VIP must however be based on the improvement of the method in terms of some merit function, without which the method may cycle. (This is evidenced by the non-convergence of the FW method applied to the VIP; see [9].) Lawphongpanich and Hearn [20] utilize the *primal gap function*,

$$\psi(x) := \max_{y \in X} F(x)^{\mathrm{T}}(x-y),$$

which is zero at solutions to VIP and positive elsewhere in X, to guide the dropping of columns. A related merit function is used in [28]. The NSD and NSD/SQP methods are extended to VIP in [26, Chap. 9], there using the merit function

$$\psi(x) := \max_{y \in X} \{F(x)^{\mathrm{T}}(x-y) - \varphi(y,x)\}.$$

In contrast to the case of the problem (1), the sequence of solutions to the RMP in SD methods applied to (7) does not necessarily yield a monotonically decreasing sequence of values of any merit function for the VIP unless very restrictive assumptions are made on the original data, whence the theoretical properties of SD methods for VIP are less strong; for example, a consequence of the property just mentioned is that the finite convergence result for the RSD method cannot be transferred to the VIP. The solution methods that have been considered for the RMP for the VIP are generalizations of those used for the RMP of (1); the most popular ones are projection algorithms, due to a large degree to the simple form of the constraints of the RMP; see, e.g., [23] for numerical investigations of different algorithmic approaches to the RMP.

Larsson et al. [18] develop an SD scheme for *non-differentiable, convex optimization*. There, the gradient $\nabla f(x)$ is replaced by the set of subgradients, the *sub-differential*, $\partial f(x)$, defined by

$$\partial f(x) = \{\, \xi_f \in \Re^n \mid f(y) \geq f(x) + \xi_f^{\mathrm{T}}(y-x), \quad \forall y \in \Re^n \,\}.$$

It is shown that the utilization of an arbitrary subgradient $\xi_f \in \partial f(x)$ in place of the gradient in the column generation subproblem may lead to the termination of the algorithm at a non-optimal solution, since not all subgradients define descent directions. A modification of the SD scheme is therefore made, wherein the subgradients evaluated in the course of solving the RMP are averaged with weights proportional to the step lengths used in the solution method for RMP; this vector of averaged (or, ergodic) subgradients is then shown to yield an improved inner approximation. An alternative, and probably computationally much more efficient, means to define a linear column generation subproblem with the properties required is through the generation of (approximately) shortest ε-subgradients at the solutions to the RMP.

The first simplicial decomposition type methods evolved from the experience of the poor convergence of the FW method (e.g., [5, 13, 22]). The perhaps first thorough theoretical investigation of SD methods is due to Holloway [13]. Its close relationships to column generation methods however make it difficult to trace its earliest history; for example, already the first urban transportation planning studies in the 1950's applied heuristics resembling the DSD algorithm (see [25] for an overview of the history of such methods). A related class of methods also exist for least-distance and other problems in quadratic programming (e.g., [7]).

Further surveys on simplicial decomposition methods, their history and their relationships to column generation, are found in [16, 25, 26].

non-differentiable, convex optimization

References

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B.: *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

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

[3] Bertsekas, D. P.: *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, San Diego, CA, 1982.

[4] Bertsekas, D. P.: *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[5] Cantor, D. G., and Gerla, M.: 'Optimal routing in a packet-switched computer network', *IEEE Transactions on Computers* **c-23** (1974), 1062–1069.

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

[7] Djang, A.: *Algorithmic equivalence in quadratic programming*, PhD thesis, Department of Operations Research, Stanford University, Stanford, CA, 1980.

[8] García, R., Marín, A., and Patriksson, M.: A class of column generation/simplicial decomposition algorithms in convex differentiable optimization, Tech. rep., Department of Mathematics, Chalmers University of Technology, Gothenburg, Sweden, 1999.

[9] Hammond, J. H.: *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.

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

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

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

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

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

[15] Larsson, T., Migdalas, A., and Patriksson, M.: A generic column generation scheme, Report LiTH-MAT-R-94-18, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1994.

[16] Larsson, T., and Patriksson, M.: 'Simplicial decomposition with disaggregated representation for the traffic assignment problem', *Transportation Science* **26** (1992), 4–17.

[17] Larsson, T., Patriksson, M., and Rydergren, C.: 'Applications of simplicial decomposition with nonlinear column generation to nonlinear network flows', in P. M. Pardalos, W. W. Hager, and D. W. Hearn (eds.): *Network Optimization*, Vol. 450 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, 1997, pp. 346–373.

[18] Larsson, T., Patriksson, M., and Strömberg, A.-B.: Ergodic convergence in subgradient optimization, Report, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1997.

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

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

[21] Marín, A.: 'Restricted simplicial decomposition with side constraints', *Networks* **26** (1995), 199–215.

[22] Meyer, G. G. L.: 'Accelerated Frank–Wolfe algorithms', *SIAM Journal on Control* **12** (1974), 655–663.

[23] Montero, L.: *A simplicial decomposition approach for solving the variational inequality formulation of the general traffic assignment problem for large scale networks*, PhD thesis, Departament d'Estadística i Investigació Operativa, Facultat d'Informàtica, Universitat Politècnica de Catalunya, Barcelona, Spain, 1991.

[24] Mulvey, J. M., Zenios, S. A., and Ahlfeld, D. P.: 'Simplicial decomposition for convex generalized networks', *Journal of Information & Optimization Sciences* **11** (1990), 359–387.

[25] Patriksson, M.: *The Traffic Assignment Problem—Models and Methods*, Topics in Transportation. VSP BV, Utrecht, The Netherlands, 1994.

[26] Patriksson, M.: *Nonlinear Programming and Variational Inequality Problems—A Unified Approach*, Applied Optimization Series. Kluwer Academic Publishers, Dordrect, The Netherlands, 1998.

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

[28] Smith, M. J.: 'An algorithm for solving asymmetric equilibrium problems with a continuous cost-flow function', *Transportation Research* **17B** (1983), 365–371.

[29] Stefek, D.: *Extensions of simplicial decomposition for solving the multicommodity flow problem with bounded arc flows and convex costs*, PhD thesis, University of Pennsylvania, 1989.

[30] Topkis, D. M., and Veinott, A. F.: 'On the convergence of some feasible direction algorithms for nonlinear programming', *SIAM Journal on Control* **5** (1967), 268–279.

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

[32] von Hohenbalken, B.: 'Simplicial decomposition in nonlinear programming algorithms', *Mathematical Programming* **13** (1977), 49–68.

Michael Patriksson
Department of Mathematics
Chalmers University of Technology
S-412 96 Göteborg (Gothenburg)
Sweden
E-mail address: mipat@math.chalmers.se

AMS 1991 Subject Classification: 90C06, 90C25, 90C35.

Key words and phrases: inner approximation, Representation Theorem, Carathéodory's Theorem, simplex, linearization, pseudo-convex function, decomposition/coordination, column generation, restricted master problem, Dantzig–Wolfe decomposition, pricing-out, column dropping, optimal face, Frank–Wolfe, disaggregated representation, variational inequality, merit function, non-differentiable convex program.