A Generic Column Generation Scheme

by

Torbjörn Larsson
Athanasios Migdalas
Michael Patriksson

Division of Optimization
Department of Mathematics
Linköping Institute of Technology
S-581 83 Linköping
Sweden

April 5, 1995

Abstract

Given a non-empty, compact and convex set, and an *a priori* defined condition which each and one of the elements of the set either satisfies or not, we consider the problem of finding an element belonging to the former category. This is a generic and fundamental problem of mathematical programming which encompasses as special cases problem classes such as nonlinear programs, variational inequalities, and saddle point problems. For its solution, we present a conceptual column generation scheme, which alternates between the solution of restrictions of the original problem and an auxiliary problem which is used to augment the restricted problems. This scheme extends the class of simplicial decomposition method to a large problem class and generalizes it in the respect that its linear programming subproblem is replaced by the quite general auxiliary problem. We give a basic convergence result and establish the applicability of the conceptual scheme to the three problem classes mentioned. We also prove convergence of a truncated version of the scheme, in which the restricted and auxiliary problems are allowed to be solved approximately. As an illustration of the generality of the conceptual column generation scheme, it is used to show that the Dantzig–Wolfe decomposition method for linear programming can be viewed as a an application of a simplicial decomposition approach to the primal-dual saddle point formulation of a linear program.

Keywords: Linear Programming, Nonlinear Programming, Variational Inequality Problems, Saddle Point Problems, Column Generation, Frank–Wolfe Method, Simplicial Decomposition, Dantzig–Wolfe decomposition.

Bästa kollegor!

Nu orkar jag inte mer på egen hand! Jag uppskattar att jag har lagt ned uppemot ett par månader på denna abstrakta nonsens, så jag hoppas verkligen att det är något av intresse och att det leder till några fortsättningar. / TL

Detta bör göras/funderas över:

Kontrollera allt noga!

Är alla antaganden rimliga utifrån möjliga realiseringar?

Är alla antaganden minimala, det vill säga, är några antaganden onödiga?

Är alla steg i algoritmerna väldefinierade i alla fall som kan inträffa?

Framgår arbetets syfte och konsekvenser tillräckligt?

Nomenklatur?

Är nomenklaturen konsekvent genom hela rapporten?

Beteckningar?

Det behövs säkert exemplifieringar på diverse punkter för att underlätta för läsaren. Var och med vad? Enklast är att införa rubriken “Example” på nya stycken här och där. Kom med förslag på formuleringar av några sådana!

Det är för många “Remark”. Föreslå åtgärder!

Hur skall vi hantera standardkravet på numeriska tester? Det kanske inte kommer på denna typ av arbete?

Samt vad som helst i övrigt som man kan ha synpunkter på ....!

1 Introduction

1.1 Background and motivation

The solution strategy known as the column generation principle—and its dual counterpart, constraint generation—is one of the standard tools of mathematical programming, and has since the pioneering work on the maximal multicommodity network flow problem by Ford and Fulkersson [13] in the late 50's been developed and applied in a variety of contexts. With no doubt, the most famous column generation method is the Dantzig–Wolfe decomposition method for specially structured linear programs, in which the column generation is based on the pricing-out mechanism of the simplex method. Included in the class of column generation methods are also the simplicial decomposition type methods for nonlinear programs, in which the column generation is not based on pricing but on the solution of a subproblem obtained through a linear approximation of the nonlinear problem.

Our motivation for the work which led to the development of the generic column generation scheme was to generalize the simplicial decomposition type methods to allow nonlinear subproblems. Although the results obtained reach quite far outside our original scope, this generalization is still the primary motivation for our work and the main subject for continued research. The simplicial decomposition strategy is also the most natural background to and setting for the generic column generation scheme, and we are therefore inclined to proceed the introduction and the presentation from this angle of approach. However, we will later also consider applications in other contexts.

The origin of the class of *simplicial decomposition* methods is the well known algorithm of Frank and Wolfe [14], which is applicable to the class of nonlinear programs with pseudo-convex objectives and convex regions (e.g., Pshenichny and Danilin [36]), even though it was originally developed for convex quadratic programming. The Frank–Wolfe algorithm is an iterative feasible direction method which is most frequently applied to linearly constrained programs, and it then alternates between the solution of an auxiliary linear programming problem, or subproblem, obtained through a linear approximation of the objective function, and a line search towards the solution to the linear program.

For a convex program, the optima of the linear programs provide bounds on the optimal value of the original problem, and, furthermore, since this sequence of bounds converges to the optimal value, it may be used to monitor the progress of the solution process; this is a feature as compared to most alternative methods. Also, the Frank–Wolfe method is simple to understand and implement, and able to exploit problem structures, such as, for example, feasible sets being Cartesian products. Because of these nice features, the Frank–Wolfe method has reached some popularity, in particular within the field of traffic planning, although its convergence performance may be poor. (See, e.g., Canon and Cullum [6], Wolfe [46], and Hearn *et al.* [19] on this issue.) There has therefore been many suggestions for improvements, and among these we distinguish two important categories.

In a first category, one seeks to improve the quality of the search directions. An inherent property and severe weakness of the Frank–Wolfe method is that the search directions will eventually become arbitrarily poor, in the sense that their directional derivatives tend to zero (e.g., Wolfe [46]). This property is a direct consequence of the linearity of the direction-finding auxiliary problem, and a natural strategy for improvements is therefore the introduction of nonlinear direction-finding auxiliary problems. Examples of this strategy are the constrained Newton method (e.g., Pshenichny and Danilin [36]) and the *regularized* Frank–Wolfe method of Migdalas [29]. The latter method employs a nonlinear direction-finding auxiliary problem obtained by adding a pseudo-distance term to the usual linear objective. (The principle of using nonlinear direction-finding auxiliary problems in descent algorithms for nonlinear programs and variational inequality problems is in Patriksson [35] thoroughly analyzed within the framework of the class of *cost approximation algorithms*, which includes the constrained Newton and regularized Frank–Wolfe methods as a special case.)

In a second category of improvements of the Frank–Wolfe method, the line search is replaced by a multi-dimensional search. These so-called simplicial decomposition algorithms are founded on Carathéodory's theorem (e.g., Bazaraa *et al.* [3]), which states that any point in the convex hull of an arbitrary set in $\Re^n$ may be expressed as a convex combination of, at most, $n+1$ points in the set. For a nonlinear program over a bounded polyhedral set,

$$\min_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}),$$

any feasible solution may thus be represented as a convex combination of the extreme points of the feasible set. As applied to this program, the simplicial decomposition algorithms alternate between a so called *master problem*, which is the restriction of the nonlinear program to an inner approximation of the original feasible set defined by a restricted set of extreme points and provides a new iteration point and an upper bound on the optimal value, and the linear program of the Frank–Wolfe method, which provides an extreme point of the feasible polyhedron and a lower bound on the optimal value. Given a feasible iterate, $\mathbf{x}^{k-1}$, and $l \leq k-1$ stored distinct extreme points, $\mathbf{p}^i$, $i = 1, \ldots, l$, the master problem is given by

$$\begin{array}{lll} \min & f(\mathbf{x}^{k-1} + \sum_{i=1}^{l} \lambda_i (\mathbf{p}^i - \mathbf{x}^{k-1})) & \\ \text{s.t.} & \sum_{i=1}^{l} \lambda_i \ \leq \ 1 & \\ & \lambda_i \ \geq \ 0 & i = 1, \ldots, l. \end{array}$$

This problem generalizes the line search of the Frank–Wolfe method, and is potentially as difficult to solve as the original nonlinear program. Indeed, the master problem is generally the computationally most demanding part of the decomposition procedure, even though its particular structure and, in general, lower dimensionality, enables its efficient solution by using specialized methods. The simplicial decomposition strategy has been applied mainly to certain classes of structured linearly constrained convex programs, and it has then proven to be successful. Especially, for nonlinear network flow problems, the simplicial decomposition methods have shown to be efficient computational tools (e.g., Hearn *et al.* [19], Ahlfeld *et al.* [1], and Larsson and Patriksson [26]).

As mentioned above, an inherent property of the Frank–Wolfe method is that the directional derivatives of the search directions tend to zero, which causes the convergence to become very slow. Since the simplicial decomposition methods employ the auxiliary problem of the Frank–Wolfe method for the generation of extreme points to be included in the master problems, these methods may actually suffer from a similar weakness. In the case of simplicial decomposition, the directional derivatives of the objective function in the direction towards the new extreme point may be arbitrarily close to zero; there is therefore no guarantee that the inclusion of the extreme point in the master problem leads to any significant improvement in the objective value. As for the Frank–Wolfe method, there may thus be a potential for enhancements of simplicial decomposition methods through the introduction of nonlinear auxiliary problems.

1.2 Review of simplicial decomposition methods

In two papers, von Hohenbalken [20, 21] gives the theoretical grounds for simplicial decomposition methods for minimizing pseudo-convex functions on polytopes. His presentation is closely related to that of Holloway [22], who extends the Frank–Wolfe algorithm by means of an inner linearization (i.e., representation) of the feasible set, followed by restriction, resulting in a special case of the *inner linearization/restriction* type algorithms defined by Geoffrion [15]. As applied to a convex programs over bounded polyhedral sets, von Hohenbalken's algorithm is an instance from Holloway's class of algorithms.

Holloway shows that the extended Frank–Wolfe algorithm converges with a linear rate, and von Hohenbalken [21] shows that the convergence is finite even if extreme points with zero weights are removed from one master problem to the next. The latter convergence result allows for the use of *column dropping*, which is essential to gain computational efficiency in large-scale applications. According to Carathéodory's theorem, the number of extreme points needed in any master problem is bounded by $dim(\mathbf{X})+1$; this number is, however, often too large to be useful as a limit in practice. A stronger result is obtained by considering only the extrem points of the optimal face of the feasible set when invoking Carathéodory's theorem. Letting $\mathbf{F}^*$ denote the optimal face, it is then obvious that the number of extreme points of the feasible set needed to express an optimal solution is bounded by $dim(\mathbf{F}^*) + 1$. This fact is exploited in the *restricted simplicial* *decomposition* algorithm, devised by Hearn *et al.* [18], in which the number of stored extreme points is bounded by a number, r, supplied *a priori* by the user; the convergence becomes finite provided that r is greater then the dimension of the optimal face of the feasible set.

Sacher [40] specializes the simplicial decomposition technique to quadratic programs with, possibly, unbounded feasible sets. He uses column dropping in the same manner as von Hohenbalken. Shetty and Ben Daya [42] study the same problem class, but concentrate on the highly structured quadratic master problem and specialize the well known reduced gradient and gradient projection methods.

Also, quadratic programming methods such as the least-distance method of Wolfe [47] and the symmetric programming algorithm of van de Panne and Whinston [34] are in fact highly related to the simplicial decomposition technique. See Cottle and Djang [7], Djang [11] and Pang [32] on these issues.

As first observed by von Hohenbalken [21], it is favourable to use second-order methods for the solution of the master problem, since it has relatively few variables; he considers using quasi-Newton as well as Newton directions. Pang and Yu [33] approximate the master problem by a quadratic program, for which they apply the algorithm in [34]. In their restricted simplicial decomposition algorithm, Hearn *et al.* [19] employ the projected Newton method of Bertsekas [4], but also consider a quadratic approximation as in [33].

Finally, we also mention some extensions of the basic works described so far. Larsson and Patriksson [26] extend the simplicial decomposition strategy to take full advantage of Cartesian product structures, resulting in a *disaggregate simplicial decomposition* algorithm. (This work also includes a thorough discussion about the relationships between the column generation and simplicial decomposition solution strategies in the context of nonlinear multicommodity network flows.) Ventura and Hearn [45] extend the restricted simplicial decomposition method to convexly constrained problems, and in Lawphongpanich and Hearn [28], Smith [43], and Pang and Yu [33], simplicial decomposition algorithms are extended to the asymmetric traffic assignment problem, formulated as a variational inequality problem. The algorithm in [28] is combined with a column dropping scheme, governed by the gap function (e.g., Hearn *et al.* [17]).

1.3 Preview

We present a conceptual column generation scheme for a generic problem which encompasses as special cases a wide variety of problem classes from the field of mathematical programming. The feasible set of the generic problem required to be compact and convex, and its solution set is characterized by an *a priori* specified condition. The direction-finding auxiliary problem of the generic column generation scheme is not required to have any particular form. When applied to linearly constrained nonlinear programs, the generic column generation scheme thus allows combinations of the two strategies for enhancements of the Frank–Wolfe method discussed above. The main contribution of our work is that the generic column generation scheme provides a theoretical foundation for the introduction of multi-dimensional searches, instead of the traditional line searches, in a variety of existing solution methods for nonlinear programs, variational inequality problems, etc.

The outline of the remainder of the report is as follows. In the next section, we state the generic problem under consideration together with assumptions on its properties. We also introduce the auxiliary problem employed in the solution scheme. In the section which follow, we state the conceptual column generation algorithm and give the basic convergence theorem. In Section 4, it is shown that convergence is ensured when the algorithm is applied to nonlinear programs, variational inequalities and saddle point problems. Next, we present a version of the algorithm in which both master and auxiliary problems are solved inexactly; we also extend the basic convergence result to the truncated version of the algorithm. Thereafter, we establish that the Dantzig–Wolfe decomposition method is a special case of the conceptual scheme, by applying it to the primal-dual saddle point problem arising from the original linear program.

2 Preliminaries

Let $\mathbf{S}$ denote some non-empty, compact and convex set in $\Re^n$. Suppose that each element in the set $\mathbf{S}$ either satisfies or violates a well-defined condition, which is denoted by $\mathcal{C}(\mathbf{S})$ since it may in general depend of the set $\mathbf{S}$. This condition may for example correspond to the requirement that an element in $\mathbf{S}$ should be an optimizer of an objective function defined on $\mathbf{S}$, solve a variational inequality over $\mathbf{S}$, or satisfy a saddle point defining inequality over $\mathbf{S}$. (In the sequel, it may be convenient for the reader to think of an optimization problem.)

The generic problem defined for the set $\mathbf{S}$ is to

$(\mathbf{P_S})$

$$\textit{find an } \mathbf{x} \in \mathbf{S} \textit{ that satisfies the Condition } \mathcal{C}(\mathbf{S}).$$

The set $\mathbf{S}$ may be referred to as the *feasible set* of the problem $\mathbf{P_S}$, and the set of elements in $\mathbf{S}$ satisfying the Condition $\mathcal{C}(\mathbf{S})$, denoted $\mathbf{S}^*$, is then referred to as the *solution set*. Further, if a set $\widehat{\mathbf{S}} \subset \mathbf{S}$, i.e., an inner approximation of the feasible set, is non-empty, compact and convex, then the problem $\mathbf{P}_{\widehat{\mathbf{S}}}$ is said to be a *proper restriction* of the problem $\mathbf{P_S}$.

Now, consider the problem $\mathbf{P_X}$, where $\mathbf{X}$ is a non-empty, compact and convex set in $\Re^n$. The solution sets of the problem $\mathbf{P_X}$ and its proper restrictions $\mathbf{P}_{\widehat{\mathbf{X}}}$, $\widehat{\mathbf{X}} \subset \mathbf{X}$, are required to have the following properties.

Assumption 1: The solution set $\mathbf{X}^*$ is non-empty and closed. Further, whenever $\widehat{\mathbf{X}} \subset \mathbf{X}$ is non-empty, compact and convex, the solution set $\widehat{\mathbf{X}}^*$ is non-empty and closed.

Since the set $\mathbf{X}$ is bounded, this also holds for the solution sets of the problem $\mathbf{P}_{\mathbf{X}}$ and its proper restrictions.

We will also require the problem $\mathbf{P}_{\mathbf{X}}$ to have a continuity property. Consider any sequence $\{\mathbf{X}_k\}$ of non-empty, compact, convex and increasing subsets of $\mathbf{X}$. The sequence then has a set limit (e.g., Salinetti and Wets [41]), say $\widetilde{\mathbf{X}} \subseteq \mathbf{X}$, which is also non-empty, compact and convex. Further, let, for all k, $\mathbf{x}_k \in \mathbf{X}_k^*$. Then, since, for all k, $\mathbf{x}_k \in \mathbf{X}_k$, it directly follows (e.g., Aubin and Frankowska [2], Proposition 1.1.2) that any accumulation point of the sequence $\{\mathbf{x}_k\}$, $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$. However, we impose the following *problem continuity* property.

Assumption 2: Let the sequence $\{\mathbf{X}_k\}$ consists of non-empty, compact, convex and increasing subsets of $\mathbf{X}$, and with the non-empty, compact and convex set limit $\widetilde{\mathbf{X}} \subseteq \mathbf{X}$. Let, for all k, $\mathbf{x}_k \in \mathbf{X}_k^*$, and suppose that $\tilde{\mathbf{x}}$ is an accumulation point of the sequence $\{\mathbf{x}_k\}$. Then, $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}^*$.

Examples of familiar problem classes which fulfil this assumption will be given in Section 4.

We will further assume that we have at hand a principle for constructing an *auxiliary problem* which, for any $\mathbf{x} \in \mathbf{X}$, can be used to guide the search for a solution to the original problem $\mathbf{P}_{\mathbf{X}}$. This auxiliary problem generalizes the direction-finding linear programming problem of the simplicial decomposition schemes. The auxiliary problem constructed for some $\mathbf{x} \in \mathbf{X}$ and its solution set are denoted $\mathbf{A}(\mathbf{x})$ and $\mathbf{Y}(\mathbf{x}) \subseteq \mathbf{X}$, respectively.

Assumption 3: For any $\mathbf{x} \in \mathbf{X}$, the solution set $\mathbf{Y}(\mathbf{x}) \subseteq \mathbf{X}$ is non-empty.

We further suppose that the mapping associated with the auxiliary problem is *closed*, in the sense defined by Zangwill [48].

Assumption 4: The principle for constructing the auxiliary problem results in a point-to-set solution set mapping $\mathbf{Y} : \mathbf{X} \mapsto 2^{\mathbf{X}}$ that is closed on $\mathbf{X}$, that is, if, for all k, $\mathbf{x}_k \in \mathbf{X}$, $\{\mathbf{x}_k\} \to \tilde{\mathbf{x}}$, and if, for all k, $\mathbf{y}_k \in \mathbf{Y}(\mathbf{x}_k)$, $\{\mathbf{y}_k\} \to \tilde{\mathbf{y}}$, then $\tilde{\mathbf{y}} \in \mathbf{Y}(\tilde{\mathbf{x}})$.

We further assume that the auxiliary problem is devised so that it can be used to establish whether or not a solution to the original problem has been found, that is we assume that the auxiliary problem has the following *fixed point property*.

Assumption 5: Let $\mathbf{x} \in \mathbf{X}$. Then $\mathbf{x} \in \mathbf{Y}(\mathbf{x})$ if and only if $\mathbf{x} \in \mathbf{X}^*$.

Hence, a point in $\mathbf{X}$ should solve its resulting auxiliary problem exactly when it is also a solution to the original problem.

Further, whenever a solution is not at hand, the auxiliary problems shall have the following *set augmentation property*.

Assumption 6: Let the set $\widehat{\mathbf{X}} \subset \mathbf{X}$ be non-empty, compact and convex, and let $\widehat{\mathbf{x}}^* \in \widehat{\mathbf{X}}^*$. If $\widehat{\mathbf{x}}^* \notin \mathbf{Y}(\widehat{\mathbf{x}}^*)$, then $\mathbf{Y}(\widehat{\mathbf{x}}^*) \subseteq \mathbf{X} \setminus \widehat{\mathbf{X}}$.

Hence, when the solution to the proper restriction $\mathbf{P}_{\widehat{\mathbf{X}}}$ does not solve its resulting auxiliary problem, then, according to Assumption 5, neither does it solve the original problem $\mathbf{P}_{\mathbf{X}}$, but the auxiliary problem will in such cases supply candidate solutions to the original problem which are *strictly separated* from the feasible set of the restriction, that is, which have not already been considered; algorithmically, any solution to the auxiliary problem can thus be used to augment the set $\widehat{\mathbf{X}}$, i.e., improve the inner approximation of the original feasible set.

The reader should notice that Assumptions 5 and 6 generalize well known properties of the linear programming auxiliary problem of the simplicial decomposition algorithm as applied to linearly constrained convex programs.

3 The algorithm and its convergence

The conceptual column generation algorithm for the solution of the generic problem $\mathbf{P}_{\mathbf{X}}$ is stated as follows.

Step 0 (*Initialization*) Find an $\mathbf{x}^0 \in \mathbf{X}$, let $\mathbf{X}^0 = \{\mathbf{x}^0\}$ and $k = 0$.

Step 1 (*Termination check*) If the iterate $\mathbf{x}^k$ solves the auxiliary problem $\mathbf{A}(\mathbf{x}^k)$, then terminate with the same conclusion for the original problem $\mathbf{P}_{\mathbf{X}}$.

Step 2 (*Set augmentation*) Find $\mathbf{y}^{k+1}$ as an arbitrary solution to the auxiliary problem $\mathbf{A}(\mathbf{x}^k)$ and construct a closed and convex set $\mathbf{X}^{k+1}$ such that $\mathbf{y}^{k+1} \in \mathbf{X}^{k+1}$ and $\mathbf{X}^k \subset \mathbf{X}^{k+1} \subseteq \mathbf{X}$.

Step 3 (*Iteration*) Let $k := k + 1$.

Step 4 (*Solution of restricted problem*) Find a new iterate $\mathbf{x}^k$ as an arbitrary solution to the restricted problem $\mathbf{P}_{\mathbf{X}^k}$.

Step 5 (*Continuation*) Return to Step 1.

Remark 1 It is possible to use some additional procedure or to utilize some knowledge about the characteristics of a specific special case of the generic problem in order to replace the trivial initialization in Step 0 with an advanced start. Such a start may for example involve the solution of an approximation of the original problem in order to find a good initial iterate, or the solution of a nontrivial initial restricted problem, constructed through the generation of a number of feasible solutions which are believed to be profitable.

The augmentation of the inner approximation of the feasible set may of course be made in different ways; a proper generalization of the principle used in the simplicial decomposition methods would be to choose a $t_k \geq 1$ such that $\mathbf{p}^{k+1} = \mathbf{x}^k + t_k(\mathbf{y}^{k+1} - \mathbf{x}^k) \in \mathbf{X}$, and set $\mathbf{X}^{k+1} = conv\left(\mathbf{X}^k \cup \{\mathbf{p}^{k+1}\}\right)$, where $conv(\cdot)$ denotes the convex hull of a set.

Clearly, the restricted problem $\mathbf{P}_{\mathbf{X}^k}$ generalizes the master problem of the simplicial decomposition methods. Also, since it belongs to the same class of problems as the original one, $\mathbf{P}_{\mathbf{X}}$, it is, in principal, just as difficult to solve. However, as in simplicial decomposition schemes, this potential weakness of the proposed generic scheme may probably be suitably dealt with in actual realizations, as indicated below.

Assume that k iterations have been performed and that the current inner approximation of the feasible set is given by $\mathbf{X}^k = conv\{\mathbf{p}^1, \ldots, \mathbf{p}^k\}$, where the points $\mathbf{p}^i \in \mathbf{X}$, $i = 1, \ldots, k$, may for example have been generated through the set augmentation principle given above. Introducing the k-dimensional *simplex*

$$\boldsymbol{\Lambda}^k = \left\{ \boldsymbol{\lambda} \in \Re^k \;\middle|\; \sum_{i=1}^{k} \lambda_i \leq 1,\ \lambda_i \geq 0,\ i = 1, \ldots, k \right\}$$

and the feasible solution

$$\mathbf{x}(\boldsymbol{\lambda}) = \mathbf{x}^{k-1} + \sum_{i=1}^{k} \lambda_i (\mathbf{p}^i - \mathbf{x}^{k-1}), \qquad \boldsymbol{\lambda} \in \boldsymbol{\Lambda}^k,$$

we obtain

$$\mathbf{X}^k = \mathbf{x}(\boldsymbol{\Lambda}^k) = \left\{ \mathbf{x}(\boldsymbol{\lambda}) \;\middle|\; \boldsymbol{\lambda} \in \boldsymbol{\Lambda}^k \right\},$$

so that the restriction $\mathbf{P}_{\mathbf{X}^k}$ can preferably be stated as

$(\mathbf{P}_{\boldsymbol{\Lambda}^k})$

find a $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^k$ *such that* $\mathbf{x}(\boldsymbol{\lambda})$ *satisfies the Condition* $\mathcal{C}(\mathbf{x}(\boldsymbol{\Lambda}^k))$.

This equivalent problem may in practice be significantly less expensive to solve because of the simplicity of its feasible set. Furthermore, when solving the restricted problem it is often natural and beneficial to apply a reoptimization strategy, starting from $\mathbf{0} \in \boldsymbol{\Lambda}^k$.

Remark 2 Whenever the feasible set $\mathbf{X}$ is a Cartesian product, it is likely to be advantageous, from the point of view of computational efficiency, to employ a disaggregate representation, i.e., to approximate each of the sets in the product separately. (In Larsson and Patriksson [26], the advantage of using this strategy was established for an application of a simplicial decomposition scheme to a specially structured nonlinear multicommodity network flow problem. See also Jones *et al.* [23], where it is similarly observed that a disaggregate representation is advantageous in the application of the Dantzig–Wolfe decomposition method to linear multicommodity network flow problems.) It may in the Cartesian product case also be natural to use an auxiliary problem which separates over the individual sets; the resulting overall scheme can be interpreted as a Jacobi-type method with multi-dimensional searches. If the auxiliary problem is not separable, the column generation can be made in a cyclic decomposition (i.e., Gauss-Seidel) manner.

We now state and prove our main convergence result. Here,

$$d_{\mathbf{X}^*}(\mathbf{x}) = \min_{\mathbf{y}\in\mathbf{X}^*} \|\mathbf{y}-\mathbf{x}\|, \qquad \mathbf{x}\in\mathbf{X},$$

that is, the distance between some $\mathbf{x}\in\mathbf{X}$ and the solution set $\mathbf{X}^*$.

Theorem 1 (Basic convergence theorem) *If the sequence of iterates generated by the algorithm, $\{\mathbf{x}^k\}$, is finite, then it has been established that the last iterate solves the problem $\mathbf{P}_{\mathbf{X}}$. If the sequence of iterates is infinite, then*

$$\left\{d_{\mathbf{X}^*}(\mathbf{x}^k)\right\} \to 0.$$

Proof If the sequence of iterates is finite, the algorithm has been interrupted in Step 1, and the conclusion follows from Assumption 5. In the remainder of the proof, we thus assume that the sequence of iterates is infinite.

The sequence $\{\mathbf{X}^k\}$ consists of non-empty, compact, convex and increasing sets. Thus, it has a set limit (e.g., Salinetti and Wets [41]), say $\widetilde{\mathbf{X}}\subseteq\mathbf{X}$, which is also non-empty, compact and convex.

Let $\varepsilon\geq 0$ be such that the sequence of iterates contains an infinite subsequence, say $\{\mathbf{x}^k\}_{k\in K}$, where $K\subseteq\{0,1,2,\ldots\}$, such that $d_{\mathbf{X}^*}(\mathbf{x}^k)\geq\varepsilon$ for all $k\in K$.

Since the sequence $\{\mathbf{x}^k\}_{k\in K}$ belongs to the compact set $\mathbf{X}$, it has at least one accumulation point, say $\tilde{\mathbf{x}}\in\mathbf{X}$, which is the limit point of a convergent infinite subsequence, say $\{\mathbf{x}^k\}_{k\in\widetilde{K}}$, where $\widetilde{K}\subseteq K$. Then, from Assumption 2, $\tilde{\mathbf{x}}\in\widetilde{\mathbf{X}}^*$.

Since the sequence $\{\mathbf{y}^{k+1}\}_{k\in\widetilde{K}}$ belongs to the compact set $\mathbf{X}$, is has at least one accumulation point, say $\tilde{\mathbf{y}}\in\mathbf{X}$. From the closedness of the auxiliary problem mapping, it follows that $\tilde{\mathbf{y}}\in\mathbf{Y}(\tilde{\mathbf{x}})$. Since $\mathbf{y}^{k+1}\in\mathbf{X}^{k+1}$ for all $k\in\widetilde{K}$, it follows (e.g., Aubin and Frankowska [2], Proposition 1.1.2) that $\tilde{\mathbf{y}} \in \widetilde{\mathbf{X}}$. From Assumptions 5 and 6, one may then conclude that $\tilde{\mathbf{x}} \in \mathbf{X}^*$.

Hence, $\varepsilon = 0$, and the result of the theorem follows. □

Remark 3 If the problem $\mathbf{P}_{\mathbf{X}}$ has a finite number of solutions, then the sequence of iterates converges to one of them.

Remark 4 The boundedness requirement on the set X can be relaxed if it can in some other way be guaranteed that the sequences $\{\mathbf{x}^k\}$ and $\{\mathbf{y}^k\}$ are both bounded. Consider, as an example of this, an application to a convex nonlinear program

(**CNLP**)

$$\min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}),$$

where $\mathbf{X} \subseteq \Re^n$ is a non-empty, closed and convex, but unbounded, set, and $f : \mathbf{X} \mapsto \Re$ is continuous and convex, and suppose that the generic scheme is implemented so that the sequence $\{f(\mathbf{x}^k)\}$ of objective values is descending. Letting $L_X^f(\mathbf{x}^0) \subset \mathbf{X}$ denote the closed level set of the objective function corresponding to the initial iterate, it then follows that $\{\mathbf{x}^k\} \subseteq L_X^f(\mathbf{x}^0)$. Now, if the optimal set is bounded, then any level set of the objective function, including, in particular, the initial one, is also bounded (e.g., Rockafellar [38], Corollary 8.7.1). Hence, in such a case, the sequence of iterates becomes bounded. If, further, the auxiliary problem is designed so that the solution set $\mathbf{Y}(\mathbf{x})$ is non-empty and compact for any $\mathbf{x} \in L_X^f(\mathbf{x}^0)$ and the point-to-set solution set mapping $\mathbf{Y} : \mathbf{X} \mapsto 2^{\mathbf{X}}$ is upper semicontinuous on the compact set $L_X^f(\mathbf{x}^0)$, then the set $\mathbf{Y}(L_X^f(\mathbf{x}^0)) = \cup_{L_X^f(\mathbf{x}^0)} \mathbf{Y}(\mathbf{x})$ is also compact (e.g., Nikaido [30], Lemma 4.5). Since, for all k, $\mathbf{y}^k \in \mathbf{Y}(\mathbf{x}^k)$, and $\{\mathbf{x}^k\} \subseteq L_X^f(\mathbf{x}^0)$, it then follows that the sequence $\{\mathbf{y}^k\} \subseteq \mathbf{Y}(L_X^f(\mathbf{x}^0))$ is bounded.

Remark 5 In case the feasible set may be expressed as $\mathbf{X} = \mathbf{G} \cap \mathbf{R}$, where for example $\mathbf{G}$ is a structural ground set and the set $\mathbf{R}$ is associated with some additional requirements for overall feasibility, the application of the generic scheme may be based on an inner approximation of only the ground set, while the additional requirements are treated explicitly. Such an extended scheme make use of a (relaxed) auxiliary problem with solution set mapping $\mathbf{Y} : \mathbf{G} \cap \mathbf{R} \mapsto 2^{\mathbf{G}}$, and a (side constrained) restricted problem $\mathbf{P}_{\mathbf{G}^k \cap \mathbf{R}}$, where for example $\mathbf{G}^k = conv\{\mathbf{p}^1, \ldots, \mathbf{p}^k\}$ with $\mathbf{p}^i \in \mathbf{G}$, $i = 1, \ldots, k$, and solved as

($\mathbf{P}_{\boldsymbol{\Lambda}^k \cap \boldsymbol{\lambda}(\mathbf{R})}$)

$$\textit{find a } \boldsymbol{\lambda} \in \boldsymbol{\Lambda}^k \textit{ such that } \mathbf{x}(\boldsymbol{\lambda}) \in \mathbf{R} \textit{ and satisfies the Condition } \mathcal{C}(\mathbf{x}(\boldsymbol{\Lambda}^k) \cap \mathbf{R}),$$

where $\boldsymbol{\lambda}(\mathbf{R}) = \{\boldsymbol{\lambda} \mid \mathbf{x}(\boldsymbol{\lambda}) \in \mathbf{R}\}$. In the context of simplicial decomposition applied to nonlinear multicommodity network flows, Stefek [44] has recently developed methods of that type.

Remark 6 Convergence is guaranteed also if the solution of the auxiliary problem in the set augmentation step of the algorithm is replaced by the invocation of a closed *set augmentation mapping*, which generalizes the solution set mapping of the auxiliary problem. We have chosen to consider set augmentation through the exact solution of an auxiliary problem, since it is from a conceptual point of view a quite natural way to implement such a set augmentation mapping. Another way to implement a closed set augmentation mapping is through the approximate solution of an auxiliary problem with closed solution set mapping, using a solution method with a closed algorithmic map, so that the resulting composite mapping becomes closed.

4 Applicability

We will in this section give a simple sufficient condition for Assumption 2 to be fulfilled, and show that this condition is readily applicable to three familiar problem classes.

Lemma 1 (A sufficient condition for problem continuity) *Assume that there exists a function $\psi : 2^{\mathbf{X}} \times \mathbf{X} \mapsto \Re$ with the following properties.*

(1) (Continuity) *Let the sequence $\{\mathbf{X}_k\}$ consists of non-empty, compact, convex and increasing subsets of $\mathbf{X}$, with set limit $\widetilde{\mathbf{X}} \subseteq \mathbf{X}$. Further, consider a convergent sequence $\{\mathbf{x}_k\}$, where, for all k, $\mathbf{x}_k \in \mathbf{X}_k$, with limit $\tilde{\mathbf{x}}$. Then, $\{\psi(\mathbf{X}_k, \mathbf{x}_k)\} \to \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{x}})$.*

(2) (Merit property) *Let $\widehat{\mathbf{X}} \subseteq \mathbf{X}$ be non-empty, compact and convex. Then, $\psi(\widehat{\mathbf{X}}, \mathbf{x}) = \min_{\mathbf{y} \in \widehat{\mathbf{X}}} \psi(\widehat{\mathbf{X}}, \mathbf{y}) \Leftrightarrow \mathbf{x} \in \widehat{\mathbf{X}}^*$.*

Then Assumption 2 is fulfilled.

Proof Consider a sequence $\{\mathbf{x}_k\}$ where, for all k, $\mathbf{x}_k \in \mathbf{X}_k^*$, and let $\tilde{\mathbf{x}}$ be one of its accumulation points, which is then the limit of a convergent infinite subsequence, say $\{\mathbf{x}_k\}_{k \in K}$, where $K \subseteq \{0, 1, 2, \ldots\}$. Consider some arbitrary $\tilde{\mathbf{y}} \in \widetilde{\mathbf{X}}$, which is then (e.g., Aubin and Frankowska [2], Proposition 1.1.2) the limit of a convergent sequence $\{\mathbf{y}_k\}$, where, for all k, $\mathbf{y}_k \in \mathbf{X}_k$. From the merit property, we have that, for all k, $\psi(\mathbf{X}_k, \mathbf{y}_k) \geq \psi(\mathbf{X}_k, \mathbf{x}_k)$. Taking the limit corresponding to the infinite subsequence K, using that $\{\mathbf{X}_k\}_{k \in K} \to \widetilde{\mathbf{X}}$ (since the subsequence is infinite and $\{\mathbf{X}_k\} \to \widetilde{\mathbf{X}}$), $\{\mathbf{y}_k\}_{k \in K} \to \tilde{\mathbf{y}}$ (since $\{\mathbf{y}_k\} \to \tilde{\mathbf{y}}$), and the continuity property, we obtain that $\psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{y}}) \geq \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{x}})$. Finally, by recalling that $\tilde{\mathbf{y}} \in \widetilde{\mathbf{X}}$ is arbitrary and again invoking the merit property, the result of the lemma follows. □

We first consider the class of nonlinear programming problems of the form

(**NLP**)

$$\min_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}),$$

where $\mathbf{X} \subset \Re^n$ is a non-empty, compact and convex set, and $f : \mathbf{X} \mapsto \Re$ is continuous. As is very well known, Assumption 1 holds for this problem class, and to verify that Assumption 2 is fulfilled we invoke the lemma with the choice

$$\psi(\widehat{\mathbf{X}}, \mathbf{x}) \stackrel{\text{def}}{=} f(\mathbf{x}), \qquad \mathbf{x} \in \widehat{\mathbf{X}},\ \forall \widehat{\mathbf{X}} \subseteq \mathbf{X},$$

whose fulfillment of the continuity and merit properties is obvious.

Secondly, consider the variational inequality problem (e.g., Harker and Pang [16]) to find an $\mathbf{x}^* \in X$ such that

(**VIP**)

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

where $\mathbf{F} : X \mapsto \Re^n$ is continuous on the non-empty, compact and convex set $X \subset \Re^n$. Under these assumptions, the problem **VIP** has a non-empty and closed solution set. The lemma can in this case be invoked by choosing the function ψ as, for example, the so-called primal gap function (e.g., Larsson and Patriksson [25]), i.e.,

$$\psi(\widehat{\mathbf{X}}, \mathbf{x}) \stackrel{\text{def}}{=} \max_{\mathbf{y}\in\widehat{\mathbf{X}}} \mathbf{F}(\mathbf{x})^{\mathrm{T}}(\mathbf{x} - \mathbf{y}), \qquad \mathbf{x} \in \widehat{\mathbf{X}},\ \forall \widehat{\mathbf{X}} \subseteq \mathbf{X},$$

which can be shown to have the required continuity property. Further, it has the merit property since it may be shown that, for all $\widehat{\mathbf{X}} \subseteq \mathbf{X}$, $\psi(\widehat{\mathbf{X}}, \mathbf{x}) \geq 0$ for all $\mathbf{x} \in \widehat{\mathbf{X}}$, and $\psi(\widehat{\mathbf{X}}, \mathbf{x}) = 0$ if and only if $\mathbf{x} \in \widehat{\mathbf{X}}^*$.

Thirdly, we consider a saddle point problem (e.g., Danskin [8], Rockafellar [38], and Dem'yanov and Malozemov [9]). Given a non-empty, compact and convex Cartesian product set $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2$, where $\mathbf{X}_1 \subset \Re^{n_1}$ and $\mathbf{X}_2 \subset \Re^{n_2}$, and a continuous saddle function $L : \mathbf{X}_1 \times \mathbf{X}_2 \mapsto \Re$, i.e., a function which is convex and concave in its first and second arguments, respectively, an $\mathbf{x} = (\mathbf{x}_1^*, \mathbf{x}_2^*) \in \mathbf{X}_1 \times \mathbf{X}_2$ is sought such that

(**SPP**)

$$L(\mathbf{x}_1^*, \mathbf{x}_2) \leq L(\mathbf{x}_1^*, \mathbf{x}_2^*) \leq L(\mathbf{x}_1, \mathbf{x}_2^*), \qquad \forall \mathbf{x}_1 \in \mathbf{X}_1,\ \forall \mathbf{x}_2 \in \mathbf{X}_2.$$

In this case we may choose $\psi(\widehat{\mathbf{X}}, \mathbf{x}) \stackrel{\text{def}}{=} \psi_1(\widehat{\mathbf{X}}_1, \mathbf{x}_1) - \psi_2(\widehat{\mathbf{X}}_2, \mathbf{x}_2)$, where

$$\psi_1(\widehat{\mathbf{X}}_1, \mathbf{x}_1) \stackrel{\text{def}}{=} \max_{\mathbf{x}_2\in\widehat{\mathbf{X}}_2} L(\mathbf{x}_1, \mathbf{x}_2), \qquad \mathbf{x}_1 \in \widehat{\mathbf{X}}_1,\ \forall \widehat{\mathbf{X}}_1 \subseteq \mathbf{X}_1,$$

$$\psi_2(\widehat{\mathbf{X}}_2, \mathbf{x}_2) \stackrel{\text{def}}{=} \min_{\mathbf{x}_1 \in \widehat{\mathbf{X}}_1} L(\mathbf{x}_1, \mathbf{x}_2), \qquad \mathbf{x}_2 \in \widehat{\mathbf{X}}_2,\ \forall \widehat{\mathbf{X}}_2 \subseteq \mathbf{X}_2.$$

The function ψ is continuous in the sense of the lemma, and the merit property follows from that, for all $\widehat{\mathbf{X}} \subseteq \mathbf{X}$, $\psi(\widehat{\mathbf{X}}, \mathbf{x}) \geq 0$ for all $\mathbf{x} \in \widehat{\mathbf{X}}$, and $\psi(\widehat{\mathbf{X}}, \mathbf{x}) = 0$ if and only if $\mathbf{x} \in \widehat{\mathbf{X}}^*$.

To summarize, we have thus established that the generic column generation scheme is directly applicable to nonlinear programs, variational inequalities and saddle point problems involving continuous functions and compact and convex feasible sets.

When there exists a continuous merit function ψ, it may also be used to establish that an auxiliary problem has the fixed point and set augmentation properties, defined in Assumptions 5 and 6.

Lemma 2 (A sufficient condition for the fixed point property) *Assume that there exists a continuous merit function $\psi : 2^{\mathbf{X}} \times \mathbf{X} \mapsto \Re$ as defined in Lemma 1 with the following properties.*

(1) (Descent at non-solutions) *If $\mathbf{x} \notin \mathbf{Y}(\mathbf{x})$, then for all $\mathbf{y} \in \mathbf{Y}(\mathbf{x})$ and all sufficiently small $t \in (0, 1]$,*

$$\psi(\mathbf{X}, \mathbf{x} + t(\mathbf{y} - \mathbf{x})) < \psi(\mathbf{X}, \mathbf{x}).$$

(2) (Non-descent at solutions) *If $\mathbf{x} \in \mathbf{Y}(\mathbf{x})$, then*

$$\psi(\mathbf{X}, \mathbf{y}) \geq \psi(\mathbf{X}, \mathbf{x}), \qquad \forall \mathbf{y} \in \mathbf{X}.$$

Then Assumption 5 is fulfilled.

Proof Immediate from the merit property of the function ψ. □

Considering a differentiable nonlinear program **NLP** and choosing $\psi \equiv f$, the descent property reduces to

$$\nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) < 0,$$

i.e., the familiar descent condition for a differentiable objective function, which is the basis for the convergence analysis for Frank–Wolfe type methods and simplicial decomposition schemes.

Remark 7 From the descent property of the lemma it directly follows that if for some $\mathbf{y} \in \mathbf{Y}(\mathbf{x})$, $\psi(\mathbf{X}, \mathbf{x} + t(\mathbf{y} - \mathbf{x})) \geq \psi(\mathbf{X}, \mathbf{x})$ for arbitrarily small $t > 0$, then $\mathbf{x} \in \mathbf{Y}(\mathbf{x})$. Hence, under this assumption, a feasible solution solves its corresponding auxiliary problem if any of its solutions does not provide descent with respect to the merit function $\psi(\mathbf{X}, \cdot)$.

Lemma 3 (A sufficient condition for the set augmentation property) *Assume that there exists a continuous merit function* $\psi : 2^{\mathbf{X}} \times \mathbf{X} \mapsto \Re$ *as defined in Lemma 1 with the additional property that for all non-empty, compact and convex sets* $\widehat{\mathbf{X}} \subset \mathbf{X}$*, any point* $\widehat{\mathbf{x}}^* \in \widehat{\mathbf{X}}^*$ *such that* $\widehat{\mathbf{x}}^* \notin \mathbf{Y}(\widehat{\mathbf{x}}^*)$*, and any* $\mathbf{y} \in \mathbf{Y}(\widehat{\mathbf{x}}^*)$*, it holds that*

$$\psi\left(\widehat{\mathbf{X}}, \widehat{\mathbf{x}}^* + t(\mathbf{y} - \widehat{\mathbf{x}}^*)\right) < \psi\left(\widehat{\mathbf{X}}, \widehat{\mathbf{x}}^*\right)$$

for sufficiently small $t \in (0, 1]$*. Then Assumption 6 is fulfilled.*

Proof If Assumption 6 is not fulfilled, then there exists a non-empty, compact and convex set $\widehat{\mathbf{X}} \subset \mathbf{X}$, a point $\widehat{\mathbf{x}}^* \in \widehat{\mathbf{X}}^*$ such that $\widehat{\mathbf{x}}^* \notin \mathbf{Y}(\widehat{\mathbf{x}}^*)$, and a $\mathbf{y} \in \mathbf{Y}(\widehat{\mathbf{x}}^*)$ such that $\mathbf{y} \in \widehat{\mathbf{X}}$. Then, for all $t \in (0, 1]$, $\widehat{\mathbf{x}}^* + t(\mathbf{y} - \widehat{\mathbf{x}}^*) \in \widehat{\mathbf{X}}$, and from the merit property of the function ψ it follows that, for all $t \in (0, 1]$,

$$\psi\left(\widehat{\mathbf{X}}, \widehat{\mathbf{x}}^* + t(\mathbf{y} - \widehat{\mathbf{x}}^*)\right) \geq \psi\left(\widehat{\mathbf{X}}, \widehat{\mathbf{x}}^*\right),$$

which contradicts the given descent condition, and the result follows. □

Remark 8 In many cases, the natural strategy when designing an algorithm within the class of generic column generation schemes may be to *first* find a suitable merit function for the problem class under consideration and then devise an auxiliary problem which is related to the merit function in accordance with the above lemmas.

Even though the original problem $\mathbf{P}_{\mathbf{X}}$ does not belong to the field of traditional optimization, the availability of a continuous merit function $\psi(\mathbf{x}) = \psi(\mathbf{X}, \mathbf{x})$, $\mathbf{x} \in \mathbf{X}$, implies that the non-optimization problem $\mathbf{P}_{\mathbf{X}}$ can be reformulated into the equivalent optimization problem

$$\min_{\mathbf{x} \in \mathbf{X}} \psi(\mathbf{x}).$$

One can then apply the generic column generation scheme either directly to the problem $\mathbf{P}_{\mathbf{X}}$ or to the equivalent optimization problem. These two strategies will generally not produce the same sequence of iterates, since the problem $\mathbf{P}_{\widehat{\mathbf{X}}}$, i.e., the proper restriction to the set $\widehat{\mathbf{X}} \subset \mathbf{X}$, is generally not equivalent to the optimization problem $\min_{\mathbf{x} \in \widehat{\mathbf{X}}} \psi(\mathbf{x})$, although the original problem $\mathbf{P}_{\mathbf{X}}$ is equivalent to the optimization problem $\min_{\mathbf{x} \in \mathbf{X}} \psi(\mathbf{x})$. (An example of a case in which equivalence actually holds is the nonlinear program **NLP** with the choice $\psi \equiv f$.)

5 A truncated version of the algorithm and its convergence

In the truncated version of the generic column generation algorithm, the auxiliary and restricted problems are allowed to be solved only approximately, using iterative algorithms, denoted $\mathcal{A}_a$ and $\mathcal{A}_r$, respectively, with closed iteration maps (e.g., Zangwill [48]). The convergence analysis relies on the merit function concept.

Assumption 7: There exists a continuous merit function ψ, as defined in Lemma 1, for the problem $\mathbf{P}_{\mathbf{X}}$.

Assumption 8: As applied to an auxiliary problem $\mathbf{A}(\mathbf{x})$, where $\mathbf{x} \in \mathbf{X}$, the algorithm $\mathcal{A}_a$ has a closed iteration map and all its iterates belongs to the set $\mathbf{X}$. Further, for any proper restriction $\mathbf{P}_{\widehat{\mathbf{X}}}$, where $\widehat{\mathbf{X}} \subset \mathbf{X}$ is non-empty, compact and convex, and starting from any $\widehat{\mathbf{x}} \in \widehat{\mathbf{X}}$, each iteration of the algorithm applied to the auxiliary problem $\mathbf{A}(\widehat{\mathbf{x}})$ gives descent with respect to the merit function $\psi(\widehat{\mathbf{X}}, \cdot)$, unless a solution to the problem $\mathbf{P}_{\mathbf{X}}$ is at hand.

Assumption 9: As applied to a restriction $\mathbf{P}_{\widehat{\mathbf{X}}}$, where $\widehat{\mathbf{X}} \subseteq \mathbf{X}$ is non-empty, compact and convex, the algorithm $\mathcal{A}_r$ has a closed iteration map and all its iterates belongs to the set $\widehat{\mathbf{X}}$. Further, for any such restriction and any feasible solution to the restriction, an iteration of the algorithm gives descent with respect to the merit function $\psi(\widehat{\mathbf{X}}, \cdot)$, unless a solution to the restriction is at hand.

The truncated version of the conceptual column generation algorithm for the solution of the generic problem $\mathbf{P}_{\mathbf{X}}$ is stated as follows.

Step 0 (*Initialization*) Find an $\mathbf{x}^0 \in \mathbf{X}$, let $\mathbf{X}^0 = \{\mathbf{x}^0\}$ and $k = 0$.

Step 1 (*Termination check*) If the iterate $\mathbf{x}^k$ solves the auxiliary problem $\mathbf{A}(\mathbf{x}^k)$, then terminate with the same conclusion for the original problem $\mathbf{P}_{\mathbf{X}}$.

Step 2 (*Set augmentation*) Find $\mathbf{y}^{k+1}$ by performing one or more iterations with the algorithm $\mathcal{A}_a$ on the auxiliary problem $\mathbf{A}(\mathbf{x}^k)$, starting from $\mathbf{x}^k$, and construct a closed and convex set $\mathbf{X}^{k+1}$ such that $\mathbf{y}^{k+1} \in \mathbf{X}^{k+1}$ and $\mathbf{X}^k \subset \mathbf{X}^{k+1} \subseteq \mathbf{X}$.

Step 3 (*Iteration*) Let $k := k + 1$.

Step 4 (*Solution of restricted problem*) Find a new iterate $\mathbf{x}^k$ by performing one or more iterations with the algorithm $\mathcal{A}_r$ on the restricted problem $\mathbf{P}_{\mathbf{X}^k}$, starting from $\mathbf{x}^{k-1}$.

Step 5 (*Continuation*) Return to Step 1.

Theorem 2 (Convergence of the truncated algorithm) *If the sequence of iterates generated by the truncated algorithm, $\{\mathbf{x}^k\}$, is finite, then it has been established* *that the last iterate solves the problem* $\mathbf{P_X}$*. If the sequence of iterates is infinite, then*

$$\left\{d_{\mathbf{X}^*}(\mathbf{x}^k)\right\} \to 0.$$

Proof If the sequence of iterates is finite, the algorithm has been interrupted in Step 1, and the conclusion follows from Assumption 5. In the remainder of the proof, we thus assume that the sequence of iterates is infinite.

The sequence $\{\mathbf{X}^k\}$ consists of non-empty, compact, convex and increasing sets. Thus, it has a set limit (e.g., Salinetti and Wets [41]), say $\widetilde{\mathbf{X}} \subseteq \mathbf{X}$, which is also non-empty, compact and convex.

Since the sequence $\{\mathbf{x}^k\}$ belongs to the compact set $\mathbf{X}$, it has a non-empty and bounded set of accumulation points, say $\overline{\mathbf{X}} \subseteq \mathbf{X}$, which is closed (e.g., Rudin [39], Theorem 3.7). Since the merit function is continuous, we may find a convergent infinite subsequence, say $\{\mathbf{x}^k\}_{k\in K}$, where $K \subseteq \{0,1,2,\ldots\}$, with limit point $\mathbf{x}^K \in \arg\max_{\mathbf{x}\in\overline{\mathbf{X}}} \psi(\widetilde{\mathbf{X}},\mathbf{x})$. Further, $\mathbf{x}^K \in \widetilde{\mathbf{X}}$ (e.g., Aubin and Frankowska [2], Proposition 1.1.2).

Denote by $\mathbf{z}^{k-1} \in \mathbf{X}^k$, $k \in K$, the first iterate of the algorithm $\mathcal{A}_r$ applied to the restricted problem $\mathbf{P}_{\mathbf{X}^k}$, starting from $\mathbf{x}^{k-1} \in \mathbf{X}^k$. Since each iteration of this algorithm gives descent with respect to the merit function $\psi(\mathbf{X}^k,\cdot)$, unless a solution to the restricted problem $\mathbf{P}_{\mathbf{X}^k}$ is at hand, it follows that, for all $k \in K$, $\psi(\mathbf{X}^k,\mathbf{x}^k) \leq \psi(\mathbf{X}^k,\mathbf{z}^{k-1}) \leq \psi(\mathbf{X}^k,\mathbf{x}^{k-1})$. Let $\mathbf{x}^{K-1} \in \overline{\mathbf{X}}$ be the limit point of a convergent infinite subsequence of the sequence $\{\mathbf{x}^{k-1}\}_{k\in K}$ and let $\mathbf{z}^{K-1} \in \widetilde{\mathbf{X}}$ be an accumulation point of the corresponding subsequence of the sequence $\{\mathbf{z}^{k-1}\}_{k\in K}$. Taking the limit corresponding to this accumulation point, the continuity of the merit function yields that $\psi(\widetilde{\mathbf{X}},\mathbf{x}^K) \leq \psi(\widetilde{\mathbf{X}},\mathbf{z}^{K-1}) \leq \psi(\widetilde{\mathbf{X}},\mathbf{x}^{K-1})$.

Since $\mathbf{x}^{K-1} \in \overline{\mathbf{X}}$, the definition of $\mathbf{x}^K$ implies that $\psi(\widetilde{\mathbf{X}},\mathbf{x}^K) \geq \psi(\widetilde{\mathbf{X}},\mathbf{x}^{K-1})$, and we conclude that $\psi(\widetilde{\mathbf{X}},\mathbf{x}^K) = \psi(\widetilde{\mathbf{X}},\mathbf{z}^{K-1}) = \psi(\widetilde{\mathbf{X}},\mathbf{x}^{K-1})$. The latter equality together with the closedness and descent properties of the iteration mapping of the algorithm $\mathcal{A}_r$ gives that $\mathbf{x}^{K-1} \in \widetilde{\mathbf{X}}^*$. Then, from the relation $\psi(\widetilde{\mathbf{X}},\mathbf{x}^K) = \psi(\widetilde{\mathbf{X}},\mathbf{x}^{K-1})$, the merit property of the function $\psi(\widetilde{\mathbf{X}},\cdot)$, and the definition of $\mathbf{x}^K$, we obtain that for all $\mathbf{x} \in \widetilde{\mathbf{X}}$, $\psi(\widetilde{\mathbf{X}},\mathbf{x}) \geq \psi(\widetilde{\mathbf{X}},\mathbf{x}^K)$, and that for all $\mathbf{x} \in \overline{\mathbf{X}}$, $\psi(\widetilde{\mathbf{X}},\mathbf{x}) = \psi(\widetilde{\mathbf{X}},\mathbf{x}^K)$. Hence, $\overline{\mathbf{X}} \subseteq \widetilde{\mathbf{X}}^*$.

Now, let $\varepsilon \geq 0$ be such that there is an infinite number of iterates $\mathbf{x}^k$ with $d_{\mathbf{X}^*}(\mathbf{x}^k) \geq \varepsilon$. This infinite subsequence of iterates has some accumulation point, say $\tilde{\mathbf{x}}$, which is then the limit point of some infinite convergent sequence $\{\mathbf{x}^k\}_{k\in\widetilde{K}}$, where $\widetilde{K} \subseteq \{0,1,2,\ldots\}$. From the above we then know that $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}^*$.

For all $k \in \widetilde{K}$, let $\mathbf{v}^k \in \mathbf{X}$ denote the point obtained by performing one iteration with the algorithm $\mathcal{A}_a$ on the auxiliary problem $\mathbf{A}(\mathbf{x}^k)$, starting from $\mathbf{x}^k$. Since each iteration of the algorithm $\mathcal{A}_a$ gives descent with respect to the merit function $\psi(\mathbf{X}^k,\cdot)$, unless the current iterate is a solution to the problem $\mathbf{P_X}$, it follows that, for all $k \in \widetilde{K}$, $\psi(\mathbf{X}^k, \mathbf{y}^{k+1}) \leq \psi(\mathbf{X}^k, \mathbf{v}^k) \leq \psi(\mathbf{X}^k, \mathbf{x}^k)$. Taking the limit corresponding to an appropriate infinite subsequence, the continuity of the merit function yields that $\psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{y}}) \leq \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{v}}) \leq \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{x}})$, where $\tilde{\mathbf{v}} \in \mathbf{X}$ and $\tilde{\mathbf{y}} \in \mathbf{X}$ denote accumulation points of the sequences $\{\mathbf{v}^k\}_{k \in \widetilde{K}}$ and $\{\mathbf{y}^{k+1}\}_{k \in \widetilde{K}}$, respectively.

Since $\mathbf{y}^{k+1} \in \mathbf{X}^{k+1}$ for all $k \in \widetilde{K}$, $\tilde{\mathbf{y}} \in \widetilde{\mathbf{X}}$ holds (e.g., Aubin and Frankowska [2], Proposition 1.1.2). From the merit property of the function $\psi(\widetilde{\mathbf{X}}, \cdot)$ and the fact that $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}^*$ we conclude that $\psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{y}}) \geq \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{x}})$, and, hence, $\psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{y}}) = \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{v}}) = \psi(\widetilde{\mathbf{X}}, \tilde{\mathbf{x}})$. The second equality together with the closedness and descent properties of the iteration mapping of the algorithm $\mathcal{A}_a$ imply that $\tilde{\mathbf{x}} \in \mathbf{X}^*$.

Hence, $\varepsilon = 0$, and the result of the theorem follows. □

Remark 9 It is actually possible to relax the requirement that the algorithms employed for the solution of the auxiliary and restricted problems each is executed for at least one iteration in Step 2 and 4, respectively; convergence is also ensured under the somewhat weaker requirement that the total number of iterations made in either of these two algorithms, during the execution of the overall column generation scheme, not is finite.

6 A demonstration

As is very well known, both the simplicial decomposition algorithms, for nonlinear programs and variational inequality problems, and the classical *Dantzig–Wolfe decomposition* method, for linear programs, (e.g., Lasdon [27]), are founded on Carathéodory's theorem (e.g., Bazaraa *et al.* [3]). To demonstrate the generality of the generic column generation scheme, we will in this section establish a precise relationship between these two solution procedures by showing that the Dantzig–Wolfe decomposition method can be derived as a special case of the generic scheme. This demonstration also illustrates how special problem structures may be taken into account and exploited within the frame of the generic column generation scheme.

To this extent, consider the linear programming problem

(LP)

$$\max_{\mathbf{x} \in \mathbf{X}} \left\{ \mathbf{c}^T \mathbf{x} \;\middle|\; \mathbf{A}\mathbf{x} \leq \mathbf{b} \right\},$$

where $\mathbf{X} \in \Re^n$ is a bounded polyhedron (which is typically described by a set of linear constraints), $\mathbf{A} \in \Re^{m \times n}$, $\mathbf{b} \in \Re^m$, and $\mathbf{c} \in \Re^n$. We assume that the problem is feasible, so that its set of optimal solutions is non-empty and compact.

Introducing a vector of Lagrangean multipliers (or linear programming dual variables) $\mathbf{u} \in \Re^m_+$, the linear program is equivalent to the saddle point problem

$$\min_{\mathbf{u}\in\Re^m_+}\ \max_{\mathbf{x}\in\mathbf{X}}\ L(\mathbf{x},\mathbf{u}),$$

where

$$L(\mathbf{x},\mathbf{u}) = \mathbf{c}^T\mathbf{x} + \mathbf{b}^T\mathbf{u} - \mathbf{u}^T\mathbf{A}\mathbf{x}, \qquad \mathbf{x}\in\mathbf{X},\ \mathbf{u}\in\Re^m_+,$$

is a saddle function since it is linear in both its first and second argument.

Now, suppose that we attack this saddle point problem with the generic column generation scheme and let the auxiliary problem be defined through a linearization of the saddle function. Because of the Cartesian product structure of the feasible set $\mathbf{X}\times\Re^m_+$, each of the two sets can be treated separately. Furthermore, because of the simplicity of the dual feasible set $\Re^m_+$, it is not approximated but treated explicitly as $\{\mathbf{u} \mid \mathbf{u}\geq\mathbf{0}\}$.

Assume, for the sake of the simplicity of the presentation and with no loss of generality, that we have at hand a feasible solution to the linear program **LP**, i.e., a point $\mathbf{x}^0\in\mathbf{X}$ such that $\mathbf{A}\mathbf{x}^0\leq\mathbf{b}$. Assume further that k distinct extreme points of the set $\mathbf{X}$, denoted $\mathbf{p}^i, i=1,\ldots,k$, are known explicitly. Introducing the k-dimensional simplex

$$\boldsymbol{\Lambda}^k = \left\{\boldsymbol{\lambda}\in\Re^{k+1}\ \middle|\ \sum_{i=0}^k \lambda_i = 1,\ \lambda_i\geq 0,\ i=0,\ldots,k\right\}$$

and the primal feasible solution

$$\mathbf{x}(\boldsymbol{\lambda}) = \sum_{i=0}^k \lambda_i\mathbf{p}^i, \qquad \boldsymbol{\lambda}\in\boldsymbol{\Lambda}^k,$$

where $\mathbf{p}^0=\mathbf{x}^0$, the restricted saddle point problem is to

$$\min_{\mathbf{u}\geq\mathbf{0}}\ \max_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}^k}\ L(\mathbf{x}(\boldsymbol{\lambda}),\mathbf{u}).$$

This problem is equivalent to the linear program

$$\max_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}^k}\ \left\{\sum_{i=0}^k\left(\mathbf{c}^T\mathbf{p}^i\right)\lambda_i\ \middle|\ \sum_{i=0}^k\left(\mathbf{A}\mathbf{p}^i\right)\lambda_i\leq\mathbf{b}\right\},$$

which is recognized as the so-called *restricted master problem* of the Dantzig–Wolfe decomposition method, as applied to the problem **LP**.

Let $(\boldsymbol{\lambda}^k, \mathbf{u}^k)$ be a solution to the restricted saddle point problem, and let $\mathbf{x}^k = \mathbf{x}(\boldsymbol{\lambda}^k)$. The linearization based auxiliary problem defined at $(\mathbf{x}^k, \mathbf{u}^k)$ is to

$$\min_{\mathbf{u}\geq\mathbf{0}} \max_{\mathbf{x}\in\mathbf{X}} \; L(\mathbf{x}^k, \mathbf{u}^k) + \nabla_{\mathbf{x}} L(\mathbf{x}^k, \mathbf{u}^k)^T(\mathbf{x} - \mathbf{x}^k) + \nabla_{\mathbf{u}} L(\mathbf{x}^k, \mathbf{u}^k)^T(\mathbf{u} - \mathbf{u}^k),$$

which reduces and separates into

$$\max_{\mathbf{x}\in\mathbf{X}} \; (\mathbf{c} - \mathbf{A}^T\mathbf{u}^k)^T\mathbf{x},$$

and

$$\min_{\mathbf{u}\geq\mathbf{0}} \; (\mathbf{b} - \mathbf{A}\mathbf{x}^k)^T\mathbf{u}.$$

The former problem is recognized as the *column generation problem* (or *subproblem*) of the Dantzig–Wolfe decomposition method, as applied to the problem **LP**, and it provides a new extreme point of the set $\mathbf{X}$, $\mathbf{p}^{k+1}$. (The latter problem, which is of no interest since the dual feasible set is treated explicitly in the restricted saddle point problem, is trivial since $\mathbf{A}\mathbf{x}^k \leq \mathbf{b}$ holds.)

The Dantzig–Wolfe decomposition method is thus a special case of the generic column generation scheme, obtained by applying it to an equivalent primal-dual saddle point formulation of the original linear program and using a linearization based auxiliary problem.

We also conclude that the Dantzig–Wolfe decomposition method for linear programming can be viewed as an application of a simplicial decomposition approach to the primal-dual saddle point formulation of a linear program.

7 Discussion

A goal when deriving the generic column generation scheme has been to make it widely applicable and easily modifiable and extendible. We have therefore throughout the derivation strived at choosing the presumptions so that they are reasonable, from the point of view of our experience and intuition, and realistic, from the point of view of possible realizations, and also as weak as possible while still being able to guarantee convergence.

Clearly, the algorithm framework of the generic column generation scheme contains a large variety of possible realizations since its steps can be implemented in many ways; in an optimization context, one may for instance construct auxiliary problems with or without a pricing mechanism, which, in turn, may be linear or nonlinear (e.g., an augmented Lagrangean dualization), and one may choose between keeping some of the constraints of the original problem in the restricted problem or not.

However, it is important to observe that an extra degree of freedom in the construction of an algorithm for a specific class of problems is achieved by considering different, but equivalent, formulations of the original problem. In particular, one may in an optimization context consider primal, dual, and primal-dual formulations, respectively.

Moreover, one may also obtain different algorithms for a given problem by considering alternative equivalent formulations obtained through transformations of the original one. An example of such a transformation is the (well-known) replacement of a nonlinear objective function with an auxiliary variable and an additional constraint (made in order to create an equivalent problem with a linear objective function), which can be shown to be a means for introducing an approximation of the objective function in the restricted problem of an instance of the generic column generation scheme. (Compare with the different treatment of the objective function in a simplicial decomposition scheme and nonlinear Dantzig–Wolfe decomposition, respectively.)

8 Further research

The exploitation of the generic column generation scheme may lead to new, and hopefully also efficient, solution methods in many of the fields of mathematical programming. As concluded in Section 4, the generic scheme is directly applicable to large classes of nonlinear programming, variational inequality, and saddle point problems, but it should also be applicable to related problem classes, such as for example complementarity problems and nondifferentiable optimization problems. Because of the degree of freedom of our algorithm framework, there is with no doubt a large number of instances that are worth studying.

Within the field of nonlinear programming, a frequently employed solution principle is to alternate, in an iterative manner, between the solution of an approximate problem and a line search with respect to a merit function, which measures the degree of non-optimality of any tentative solution. Methods included in this class are the Frank–Wolfe algorithm [14], the successive linear programming approach (e.g., Bazaraa *et al.* [3]), the regularized Frank–Wolfe method [29], the method of Zoutendijk (e.g., Bazaraa *et al.* [3]), the linearization method (e.g., Pshenichnyi and Sosnovsky [37]), the constrained Newton method (e.g., Pshenichny and Danilin [36]), and the successive quadratic programming approach (e.g., Bazaraa *et al.* [3]), among others. Our main interest in continued research is to use the framework of the generic column generation scheme to enhance methods within this class through replacing their line searches with multi-dimensional searches.

Along that line of research, we are currently studying simplicial decomposition schemes with subproblems (auxiliary problems) arising from quadratic approximations of the objective functions (i.e., alternatively expressed, constrained Newton methods with multi-dimensional searches). The aim of our study is to establish whether it is computationally beneficial to utilize quadratic subproblems, which provide information of higher quality to the master (restricted) problems but are computationally more demanding than the traditional linear ones. However, this drawback can probably be circumvented, at least partially, by truncating the algorithm employed for their solution.

We are currently also studying the use of nonlinear pricing mechanisms (e.g., an augmented Lagrangean dualization) in the Dantzig–Wolfe decomposition method, which permits the subproblem to have non-extreme point solutions. (Cf., e.g., Kim and Nazaret [24], in which non-extreme subproblem solutions are generated by truncating an interior-point method, Feinberg [12], in which a special class of nonlinear price functions are used, and O'Neill and Widhelm [31], in which an augmented Lagrangian pricing is used to accelerate convergence in the extension of the Dantzig-Wolfe method to nonlinear problems.) Convergence of such a method is immediate from that of the generic scheme. Nonlinear pricing in the Dantzig–Wolfe method is of interest since it may probably enhance the method's practical performance (which is otherwise often rather poor), but also because it can be interpreted as a mathematical means to obtain an optimal utilization of scarce resources in an economic system with decentralized planning (e.g., Dirickx and Jennergren [10], and Feinberg [12]).

In order to reach a high computational efficiency in large-scale applications of specific instances of the generic column generation scheme, it may be necessary to include a column dropping facility; the development of such a facility is therefore also an obvious subject for further research regarding the generic column generation scheme itself. Column dropping facilities may be especially advantageous computationally in the realizations of the generic scheme in which the auxiliary problem constitute a high-quality approximation of the original problem (e.g., in applications in nonlinear programming, Newton-approximations), since the restricted problem will then (at least in the late iterations) have a solution which is close to the solution to the latest auxiliary problem. (See also the discussion in Section 4.16 of Patriksson [35].)

Further questions for continued research is the establishment of (non-trivial) sufficient conditions (e.g., on the properties of the original problem or the auxiliary problem, on the implementation of the steps of the algorithm, or combinations thereof) for finite convergence. In case the feasible set is polyhedral and represented by a set of linear constraints, it is also of interest to establish conditions under which the constraints that are active at a solution is identified finitely. (Cf. the results given by Burke and Moré [5] for nonlinear programming algorithms. See also Patriksson [35], Sections 4.12 and 5.13, on this issue.)

References

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

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

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

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

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

[6] M.D. Canon and C.D. Cullum, "A tight upper bound on the rate of convergence of the Frank–Wolfe algorithm", *SIAM Journal on Control* **6**, pp. 509–516 (1968).

[7] R.W. Cottle and A. Djang, "Algorithmic equivalence in quadratic programming, I: A least-distance programming problem", *Journal of Optimization Theory and Applications* **28**, pp. 275–301 (1979).

[8] J.M. Danskin, *The Theory of Max-Min*, Springer-Verlag, Berlin, 1967.

[9] V.F. Dem'yanov and V.N. Malozemov, *Introduction to Minimax*, John Wiley & Sons, New York, NY, 1974.

[10] Y.M.I. Dirickx and L.P. Jennergren, *Systems Analysis by Multilevel Methods*, John Wiley & Sons, Chichester, 1979.

[11] A. Djang, *Algorithmic Equivalence in Quadratic Programming*, Ph.D. dissertation, Department of Operation Research, Stanford University, Stanford, CA, 1980.

[12] B. Feinberg, "Coercion functions and decentralized linear programming", *Mathematics of Operations Research* **14**, pp. 177–187 (1989).

[13] L.R. Ford, Jr., and D.R. Fulkerson, "A suggested computation for maximal multi-commodity network flows", *Management Science* **5**, pp. 97–101 (1958).

[14] M. Frank and P. Wolfe, "An algorithm for quadratic programming", *Naval Research Logistics Quarterly* **3**, pp. 95–110 (1956).

[15] A.M. Geoffrion, "Elements of large scale mathematical programming", *Management Science* **16**, pp. 652–691 (1970).

[16] 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**, pp. 161–220 (1990).

[17] D.W. Hearn, S. Lawphongpanich and S. Nguyen, "Convex programming formulations of the asymmetric traffic assignment problem", *Transportation Research* **18B**, pp. 357–365 (1984).

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

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

[20] B. von Hohenbalken, "A finite algorithm to maximize certain pseudo-concave functions on polytopes", *Mathematical Programming* **9**, pp. 189–206 (1975).

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

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

[23] 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**, pp. 95–117 (1993).

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

[25] T. Larsson and M. Patriksson, "A class of gap functions for variational inequalities", Report LiTH-MAT-R-1990-18, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1990 (to appear in *Mathematical Programming*).

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

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

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

[29] A. Migdalas, "A regularization of the Frank–Wolfe method", Report LiTH-MAT-R-90-10, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1990 (to appear in *Mathematical Programming*).

[30] H. Nikaido, *Convex Structures and Economic Theory*, Academic Press, New York, NY, 1968.

[31] R.P. O'Neill and W.B. Widhelm, "Acceleration of Lagrangian column-generation algorithms by penalty function methods", *Management Science* **23**, pp. 50–58 (1976).

[32] J.-S. Pang, "An equivalence between two algorithms for quadratic programming", *Mathematical Programming* **20**, pp. 152–165 (1981).

[33] J.-S. Pang and C.-S. Yu, "Linearized simplicial decomposition methods for computing traffic equilibria on networks", *Networks* **14**, pp. 427–438 (1984).

[34] C. van de Panne and A. Whinston, "The symmetric formulation of the simplex method for quadratic programming", *Econometrica* **37**, pp. 507–527 (1969).

[35] M. Patriksson, *A Unified Framework of Descent Algorithms for Nonlinear Programs and Variational Inequalities*, Ph.D. dissertation, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1993.

[36] B.N. Pshenichny and Yu.M. Danilin, *Numerical Methods in Extremal Problems*, MIR Publishers, Moscow, 1978.

[37] B.N. Pshenichnyi and A.A. Sosnovsky, "The linearization method: Principal concepts and perspective directions", *Journal of Global Optimization* **3**, pp. 483–500 (1993).

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

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

[40] R.S. Sacher, "A decomposition algorithm for quadratic programming", *Mathematical Programming* **18**, pp. 16–30 (1980).

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

[42] C.M. Shetty and M. Ben Daya, "A decomposition procedure for convex quadratic programs", *Naval Research Logistics Quarterly* **35**, pp. 111–118 (1988).

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

[44] D. Stefek, *Extensions of Simplicial Decomposition for Solving the Multicommodity Flow Problem with Bounded Arc Flows and Convex Costs*, Ph.D. dissertation, University of Pennsylvania, 1989.

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

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

[47] P. Wolfe, "Algorithm for a least-distance programming problem", *Mathematical Programming Study* **1**, pp. 190–205 (1974).

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

Diverse rester:

Realizations/instances:

cost approximation realizations: traditional simplicial decomposition, constrained Newton-methods (e.g., Pshenichny and Danilin, 1978) with master problem (motivations: cf. Michael's thesis, and Ahlfeld et al., 1990??????), norm-regularized subproblems (yield constrained steepest descent directions as a limiting case (note: not a closed mapping), relations to the gradient projection method), proximal point methods with master problem

other realizations: subproblems according to the convex-simplex method in Zangwill (p. 162), subproblems according to a closed version of the subproblem of Zoutendijk's method (?), the feasible direction method in Zangwill p. 282, reduced gradient subproblems

Finite convergence for the case of polyhedral feasible sets if the step length in the augmentation step is always chosen maximal? Use the results for cost approximation under the Condition (a), e.g., Lemma 4.5 and Theorem 4.6.

Omformulering av standard LP till ett sadelpunktsproblem och applikation av schemat: primal tolkning?

Finns det nagot dolt avstanndsmatt i algoritmen, tex Euklidiskt avstand, som implicit resulterar i konvergensen?

something which shows the search directions may be poor in SD also, maybe Hearn et al. 1987

Gradient projection: Golstein, Levitin and Polyak, and Golstein

proximal point: Moreau 1962 and Rockafellar 1976

Jacobi: Ortega and Rheinboldt, and Feijoo and Meyer