

# Column Generation based Alternating Direction Methods for solving MINLPs

Ivo Nowak

HAW Hamburg, Department of Mechanical Engineering

Berliner Tor 21, D-20099 Hamburg

ivo.nowak@haw-hamburg.de

December 2015

## Abstract

Traditional decomposition based branch-and-bound algorithms, like branch-and-price, can be very efficient if the duality gap is not too large. However, if this is not the case, the branch-and-bound tree may grow rapidly, preventing the method to find a good solution.

In this paper, we present a new decomposition algorithm, called ADGO (Alternating Direction Global Optimization algorithm), for globally solving quasi-separable nonconvex MINLPs, which is *not* based on the branch-and-bound approach. The basic idea of ADGO is to restrict the feasible set by an upper bound of the objective function and to check via a (column generation based) globally convergent alternating direction method if the resulting MINLP is feasible or not.

Convergence of ADGO to a global solution is shown by using the fact that the duality gap of a general nonconvex projection problem is zero (in contrast to the Lagrangian dual of a general nonconvex program). Furthermore, we describe how ADGO can be accelerated by an inexact sub-problem solver, and discuss modifications to solve large-scale quasi-separable network and black-box optimization problems. Since solving the sub-problems of ADGO is not much more difficult than solving traditional pricing problems, it might be that the computational cost of ADGO is similar to a traditional column generation method.

**Keywords.** global optimization, alternating direction method, column generation, branch-and-cut, mixed-integer nonlinear programming, network optimization, surrogate optimization

## 1 Introduction

Most exact (convex and nonconvex) MINLP algorithms are based on the branch-and-bound approach and variants like branch-cut-and-price [DL10] or branch-decompose-and-cut [RG06], see [BV14, BL12] for an overview of MINLP-solvers. A main difficulty of this approach is a possibly rapidly growing branch-and-bound tree, which makes it difficult to solve large-scale models in reasonable time.

For structured nonconvex optimization problems it is possible to generate inner- or outer-approximations using traditional decomposition methods, like Lagrangian decomposition, column generation (or Dantzig-Wolfe decomposition), cutting plane methods and the Frank-Wolfe algorithm by convexifying Lagrangian sub-problems. The resulting approximation error is called duality gap. If the duality gap is not too large, the generated approximation can be used to solve huge optimization problems with several hundred millions of variables [BLR<sup>+</sup>13, Now14]. However, for many MINLPs the duality gap is not small, and in this case traditional decomposition methods may be not efficient [Now05].

Another decomposition method is the Alternating Direction Method (ADM), which solves alternately a (coupled) QP master problem and (decoupled) MINLP sub-problems. Originally, ADMs were developed for finite element problems [GM76] and are based on Uzawas algorithm [Uza58]. A review of ADMs including a convergence proof for convex problems is given in [BPC<sup>+</sup>11]. An ADM for solving MIPs is presented in [IMSS14]. Recently, a globally convergent ADM for solving quasi-separable (nonconvex) NLPs was proposed [HFD14], called ALADIN (Augmented Lagrangian based Alternating Direction Inexact Newton method). The algorithm computes iteratively approximate solutions of a nonconvex projection problem via a dual line search. Global convergence of this method is shown by the fact that the duality gap of the projection problem is zero (in contrast to the Lagrangian dual of a general nonconvex program).

Similarly to ADMs, Feasibility Pumps (FPs) for MIPs or MINLPs solve alternately a master problem and integer or nonlinear sub-problems. The first FP was proposed in [FGL05] for MIPs. FPs for solving MINLPs are presented in [AFLL12]. Current FPs find often quickly a feasible solution, however, the quality may be not always good [Ber14].

**The new solution approach.** Motivated by the approach of [HFD14] and by the excellent performance of column generation methods for solving huge network optimization problems [BLR<sup>+</sup>13, Now14], we combine in this paper column generation and an ADM for globally solving nonconvex MINLPs.

The basic idea of the new method, called Alternating Direction target-oriented Global Optimization algorithm (ADGO), is to add an upper bound constraint on the objective function to the feasible set and to check via a (column generation based) globally convergent ADM if the resulting MINLP is feasible or not. Since ADGO is a proximal optimization method, which is not based on the branch-and-bound approach, the generation of a (possibly huge) branch-and-bound tree is avoided. Furthermore, it is possible to use fast inexact MINLP-solvers for solving the sub-problems, and the sub-problems can be solved in parallel. Since processors with many cores are available, many sub-problems can be solved simultaneously. Since solving the sub-problems of ADGO is not much more difficult than solving traditional pricing problems, we expect that the computational cost of ADGO is similar to a traditional column generation method.

**MINLP formulation.** We consider in this paper a general *quasi-separable* (or *block-separable*) MINLP of the form:

$$\begin{aligned} \min \quad & c(x) \\ & Ax \leq b \\ & g_j(x) \leq 0, \quad j \in J \\ & x \in [\underline{x}, \bar{x}], \quad x_i \in \{0, 1\}, \quad i \in I^{\text{int}} \end{aligned} \tag{1.1}$$

where  $c(x) := \langle c, x \rangle$  is a linear objective function and the matrix  $A \in \mathbb{R}^{m \times n}$  and the vector  $b \in \mathbb{R}^m$  specify the linear *coupling constraints*. The (possibly nonconvex) functions  $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$  with  $j \in J$  specify the nonlinear constraints of the problem and fulfill

$$g_j(x) = g_j(x_{I_k}) \quad \text{for } j \in J_k, \quad k \in K, \tag{1.2}$$

for some disjoint index sets  $I_k \subset [n]$  and  $J_k \subset J$ , where  $x_{I_k} := (x_i)_{i \in I_k}$  denotes the subvector of  $x$ . The restriction to inequality constraints is only for notational simplicity.

The vectors  $\underline{x}, \bar{x} \in \bar{\mathbb{R}}^n$  determine the lower and upper bounds on the variables ( $\bar{\mathbb{R}} := \cup\{\pm\infty\}$ ) and  $I^{\text{int}} \subseteq [n]$  denotes the set of variables with integrality requirement. Here and in the following, we denote by  $[\underline{x}, \bar{x}] := \{x \in \mathbb{R}^n : x_i \leq \bar{x}_i, x_i \geq \underline{x}_i, i \in [n]\}$  the box for the variables.

From (1.2) it follows that (1.1) is quasi-separable regarding the variable-blocks  $\{I_k\}_{k \in K}$ . Note that a general sparse MINLP can be reformulated as a quasi-separable optimization problem by adding new variables and copy-constraints, see, e.g., [Now05].

**Outline of the paper** Section 2 describes a simplified version of ALADIN (Augmented Lagrangian based Alternating Direction Inexact Newton method) for solving quasi-separable nonconvex NLPs. Section 3 contains the main contributions of this paper. We present in section 3.1 a globally convergent ADM with an infeasibility check and in section 3.2 the new solver, called ADGO, for computing global solutions of nonconvex MINLPs. In section 3.3 we describe an exact global optimization approach, called ADOA (Alternating Direction Outer Approximation), which is based on ADGO using a branch-and-cut solver for solving the MINLP sub-problems in an extended space. ADOA uses a dual cutting plane based start heuristic and a predictor corrector approach for accelerating ADGO using both an exact and an inexact sub-problem solver. Furthermore, we discuss in section 3.4 modifications to solve large-scale quasi-separable network and black-box optimization problems. We finish with some conclusions and possible next steps in section 4.

## 2 A globally convergent decomposition based ADM for nonconvex NLPs

Recently, in [HFD14] an ADM for quasi-separable nonconvex NLPs, called ALADIN, was proposed, which uses a dual line search for making the method globally convergent. We describe now ALADIN in a simplified form.

Consider the following *weakly coupled* re-formulation of an NLP:

$$\begin{aligned} \min \quad & c(y) \\ & h(x, y) = 0, \\ & y \in G, x \in P \end{aligned} \tag{2.1}$$

where

$$\begin{aligned} P &:= \{x \in [\underline{x}, \bar{x}] : Ax \leq b\} \\ G &:= \prod_{k \in K} G_k \quad \text{with} \quad G_k := \{y \in [\underline{x}_{I_k}, \bar{x}_{I_k}] : g_j(y) \leq 0, j \in J_k\} \end{aligned}$$

specify the linear and nonlinear restrictions. The linear coupling constraints are defined by

$$h(x, y) := (h_k(x_{I_k}, y_{I_k}))_{k \in K} \quad \text{with} \quad h_k(x_{I_k}, y_{I_k}) := A_k(y_{I_k} - x_{I_k}), \quad k \in K, \tag{2.2}$$

where  $Ax = \sum_{k \in K} A_k x_{I_k}$ .

ADMs compute a sequence of approximate solutions  $(x^i, y^i) \in P \times G$  of (2.1) by alternately solving the *NLP sub-problems*

$$\min_y \{c(y) + (\lambda^{i-1})^T h(x^{i-1}, y) + \frac{\rho}{2} \|x^{i-1} - y\|_{\Sigma}^2 : y \in G\} \tag{2.3}$$

where  $\Sigma$  is a positive definite scaling matrix, and the *QP master-problem*

$$\min_x \{(\lambda^{i-1})^T h(x, y^i) + \frac{\rho}{2} \|x - y^i\|_{\Sigma}^2 : x \in P\} \tag{2.4}$$

Both problems can be solved efficiently, since (2.4) is a convex QP and (2.3) is a quasi-separable NLP, which decomposes into the following low-dimensional sub-problems, which can be solved in parallel

$$\min \{c_k(y) + (\lambda^{i-1})^T h_k(x_{I_k}^{i-1}, y) + \frac{\rho}{2} \|x_{I_k}^{i-1} - y\|_{\Sigma_k}^2 : y \in G_k\} \tag{2.5}$$

with  $c_k(x) := \langle c_{I_k}, x \rangle$ .

Consider for a fixed trial point  $x^{i-1} \in P$  the dual problem

$$\max_{\lambda \in \mathbb{R}_+^m} D_G(\lambda, x^{i-1}) \quad (2.6)$$

with

$$D_G(\lambda, x^{i-1}) := \min_{y \in G} c(y) + \lambda^T h(x^{i-1}, y) + \frac{\rho}{2} \|x^{i-1} - y\|_\Sigma^2. \quad (2.7)$$

In [HFD14] (Lemma 1) it is shown:

**Lemma 2.1.** *Assume that  $\rho$  is sufficiently large, the functions  $g_j$  are twice continuously differentiable with the second order derivative being bounded on the feasible set and the solution  $x^+$  of the following projection problem is a regular KKT-point:*

$$\begin{aligned} \min \quad & c(y) + \frac{\rho}{2} \|x^{i-1} - y\|_\Sigma^2 \\ & Ay \leq b \\ & y \in G \end{aligned} \quad (2.8)$$

If  $\|x^{i-1} - x^+\|$  is sufficiently small, then the duality gap of (2.6) is zero, and (2.6) is equivalent to (2.8).

Notice that the solution of (2.8) is a projection of  $x^{i-1}$  onto  $P \cap G$ . The proof of Lemma 2.1 is based on locally eliminating the active inequality constraints using an implicit function theorem, and showing that the resulting problem is strictly convex.

In order to make the ADM (globally) convergent, the authors of [HFD14] propose to update  $\lambda^i$  by performing the following dual line search for increasing (2.7) at an iteration point  $x^{i-1} \in P$

$$\max_{\alpha \in [0,1]} D_G(\lambda(\alpha), x^{i-1}) \quad (2.9)$$

where  $\lambda(\alpha) := \lambda^{i-1} + \alpha(\lambda^+ - \lambda^{i-1})$  and  $\lambda^+$  is the dual solution of (2.4).

Notice that traditional ADMs update  $\lambda^i$  by a subgradient step,  $\lambda^i = \lambda^{i-1} + Ax^{i-1} - b$ , and are in general not globally convergent, see [HFD14] for an example.

The dual line search (2.9) is performed whenever the solutions of (2.4) and (2.3) do not significantly reduce an exact L1-penalty function of (1.1) defined by

$$\Phi(x) := c(x) + \bar{\lambda} \|Ax - b\|_{+,1} + \bar{\mu} \sum_{k \in K, j \in J_k} |g_{kj}(x_{I_k})|_+ \quad (2.10)$$

where  $\bar{\lambda} > 0$  and  $\bar{\mu} > 0$  are sufficiently large upper bounds of the dual variables  $\lambda_j$  and  $\mu_j$  of the coupling constraints  $Ax \leq b$  and of the inequality constraints  $g_j(x) \leq 0$ , respectively.

Significant reduction of  $\Phi(x^{i-1})$  at an iteration point  $x^+$  is checked by:

$$\Phi(x^{i-1}) - \Phi(x^+) \geq \frac{\rho}{2} \|x^+ - x^{i-1}\|_2^2 + \bar{\lambda} \|Ax^+ - b\|_{+,1} \quad (2.11)$$

A simplified version of ALADIN is shown in Algorithm 1. The algorithm uses the globalization strategy of [Han77, NW06], described in Algorithm 2, which performs the dual line search (2.9), whenever (2.11) is not fulfilled, i.e.  $\Phi$  is not reduced significantly.

The following result is proved in [HFD14] (6.6):

**Lemma 2.2.** *Let  $x^{i-1} \in P$  be a trial point and  $x^+$  be the solution of the projection problem (2.8). Then  $x^+$  fulfills the descent criterion (2.11).*

---

**Algorithm 1** Globally convergent Alternating Direction NLP solver

---

```
1: function AD-NLP( $x^0, \lambda^0$ )
2:    $i \leftarrow 0$ 
3:   repeat
4:      $i \leftarrow i + 1$ 
5:      $y^i \leftarrow$  solution of (2.3)           # project  $x^{i-1}$  onto  $G$  regarding  $\lambda^{i-1}$ 
6:      $(x^i, \lambda^i) \leftarrow \text{Proj}(y^i, \lambda^{i-1}, x^{i-1})$    # project  $y^i$  onto  $P$  regarding  $\lambda^{i-1}$ 
7:   until  $\|Ay^i - b\|_{+,1} < \epsilon$ 
8:   return  $y^i$ 
```

---

---

**Algorithm 2** Projection of  $y^i$  onto  $P$  and update of  $\lambda^i$  by dual line search

---

```
1: function PROJ( $y^i, \lambda^{i-1}, x^{i-1}$ )
2:    $(x^+, \lambda^+) \leftarrow$  primal/dual solution of (2.4)   # project  $y^i$  onto  $P$  regarding  $\lambda^{i-1}$ 
3:   if (2.11) then                                         #  $\Phi(x^{i-1}) - \Phi(x^+)$  is large enough
4:      $x^i \leftarrow x^+$  and  $\lambda^i \leftarrow \lambda^+$ 
5:   else
6:     if (2.11) is fulfilled for  $x^+ = y^i$  then           #  $\Phi(x^{i-1}) - \Phi(y^i)$  is large enough
7:        $x^i \leftarrow y^i$  and  $\lambda^i \leftarrow \lambda^{i-1}$ 
8:     else
9:        $\alpha \leftarrow$  solution of (2.9)                   # dual line search for updating  $\lambda^i$ 
10:       $\lambda^i \leftarrow (1 - \alpha)\lambda^{i-1} + \alpha\lambda^+$ 
11:       $x^i \leftarrow x^{i-1}$ 
12:   return  $(x^i, \lambda^i)$ 
```

---

From this it follows that if the dual line search (2.9) is performed sufficiently many times, the descent criterion (2.11) is fulfilled, and the current trial point  $x^{i-1}$  is accepted. This is used to prove the following theorem:

**Theorem 2.1.** *Let problem (1.1) be feasible and bounded from below such that a minimum exists. Assume that the assumptions of Lemma 2.1 are satisfied. Let  $x^0 \in [\underline{x}, \bar{x}]$  and  $\lambda^0 \in \mathbb{R}_+^m$ . Then Algorithm 1 terminates after a finite number of iterations.*

Proof. Algorithm 1 is a special case of ALADIN (see section 5 of [HFD14]). Since the assumptions of Theorem 2.1 and Theorem 2 of [HFD14] are equal and because global convergence of ALADIN is proved in Theorem 2, [HFD14], the statement is proved.  $\square$ .

**Remark 2.1.** *Since Algorithm 1 stops if  $\|Ay^i - b\|_{+,1} < \epsilon$ , a solution  $y^i$  is  $\epsilon$ -feasible regarding the coupling constraints, i.e. it is contained in the  $\epsilon$ -feasible set of (1.1) defined by*

$$\Omega_\epsilon := \{x \in G : \|Ax - b\|_{+,1} \leq \epsilon\} \quad (2.12)$$

*Let  $v^*$  be the optimal value of (1.1) and  $v_\epsilon^* := \min_{x \in \Omega_\epsilon} c(x)$ . Then  $v_\epsilon^* \leq c(y^i) \leq v^*$ .*

**Remark 2.2.** *As proposed in [HFD14], it is possible to accelerate Algorithm 1 by refining the (simple) polyhedral outer-approximation  $P$  in (2.4) by adding quadratic approximations of the nonlinear constraints  $g_j$  at  $x^{i-1}$  to  $P$ .*

**Remark 2.3.** *Notice that a dual problem of (2.1) with zero duality gap can be defined using the augmented Lagrangian dual function*

$$D_{\text{aug}}(\lambda) := \min\{c(y) + \lambda^T h(x, y) + \frac{\rho}{2} \|h(x, y)\|_2^2 : x \in P, y \in G\} \quad (2.13)$$

*where  $\lambda$  are the multipliers of the linear coupling constraints  $h(x, y) = 0$ . If  $\rho$  is sufficiently large, then maximizing  $D_{\text{aug}}(\lambda)$  solves (2.1) and the duality gap is zero [Roc74]. However, in*

contrast to the nonconvex projection problem (2.8), the augmented Lagrangian problem (2.13) is not quasi-separable and therefore difficult to solve.

### 3 A column generation based ADM for globally solving nonconvex MINLPs

We present now a new algorithm for globally solving the nonconvex MINLP (1.1). The basic idea of the algorithm is to add a target constraint

$$c(x) \leq \sigma \quad (3.1)$$

to the feasible set of (1.1) and to check via a (column generation based) globally convergent ADM if the resulting MINLP is feasible or not.

Similarly as (2.1), the method is based on the following reformulation of the nonconvex MINLP (1.1):

$$\begin{aligned} \min \quad & \frac{1}{2}(c(x) + c(y)) \\ & h(x, y) = 0, \\ & y \in G, x \in P \end{aligned} \quad (3.2)$$

where  $P$  and  $h(x, y)$  are defined as in (2.1) and  $G = \prod_{k \in K} G_k$  with

$$G_k := \{y \in [\underline{x}_{I_k}, \bar{x}_{I_k}] : g_j(y) \leq 0, j \in J_k, i \in I_k^{\text{int}}\}. \quad (3.3)$$

#### 3.1 A column generation based ADM for nonconvex MINLP

In this section we describe a globally convergent ADM for computing solution candidates of (1.1) and for checking if the target constraint (3.1) is valid or not. The method is similar to Algorithm 1, but since it is called several times with different target values  $\sigma$ , it uses Column Generation (CG) to compute approximate solutions of the projection problem (2.8), instead of the dual line search of Algorithm 2. This makes it possible to perform an efficient warm start by keeping sample points of previous iterations. If no new points are generated by CG, a local improvement step is performed, in order to reduce  $\|x^{i-1} - x^+\|$ .

In order to check if (3.1) is valid and to make the sets  $G$  and  $P$  more similar,  $P$  is replaced in (3.2) by

$$P_\sigma := \{x \in [\underline{x}, \bar{x}] : \bar{A}x \leq \bar{b}, C_k x_{I_k} \leq d_k, k \in K\} \quad (3.4)$$

where  $\bar{A}x \leq \bar{b}$  is equal to  $\{Ax \leq b \wedge c(x) \leq \sigma\}$  and  $C_k x \leq d_k$  defines an *outer-approximation* of  $G_k$ , i.e.

$$\{x \in \mathbb{R}^{n_k} : C_k x \leq d_k\} \supseteq G_k, \quad (3.5)$$

See section 3.3.2 for how an outer-approximation of  $G_k$  can be computed using a dual cutting plane algorithm.

The new ADM solves alternately a MNLP-sub-problem and a QP-master-problem, respectively:

$$\min_y \quad \{c(y) + (\lambda^{i-1})^T h(x^{i-1}, y) + \rho \|x^{i-1} - y\|_\Sigma^2 : y \in G\} \quad (3.6)$$

$$\min_x \quad \{c(x) + (\lambda^{i-1})^T h(x, y^i) + \rho \|h(x, y^i)\|_2^2 : x \in P_\sigma\} \quad (3.7)$$

Problem (3.6) decompose into the following  $|K|$  sub-problems:

$$y_k^i = \operatorname{argmin}_{y \in G_k} c_k(y) + \rho \cdot d_k(y, x_{I_k}^{i-1}) + (\lambda^{i-1})^T \bar{A}_k y \quad (3.8)$$

where  $d_k(y, x) := \|y - x\|_{\Sigma_k}^2$  is the L2-distance function and  $\lambda$  denotes the dual multiplier of the constraint  $h(x, y) = 0$ .

### 3.1.1 Solving the projection problem by column generation

Similar as Algorithm 1, the new ADM projects trial points  $x^{i-1}$  onto  $P_\sigma \cap G$  by approximately solving the (quasi-separable) MINLP-projection problem

$$\begin{aligned} \min \quad & c(y) + \rho \|y - x^{i-1}\|_\Sigma^2 \\ & \bar{A}y \leq \bar{b} \\ & y \in G \end{aligned} \tag{3.9}$$

Consider the dual problem

$$\max_{\lambda \in \mathbb{R}^m} D_G(\lambda, x^{i-1}) \tag{3.10}$$

where the dual function  $D_G$  is defined in (2.7) with  $G$  as in (3.3).

**Observation 1.** *A MINLP can be formulated as a NLP by replacing binary constraints  $x_i \in \{0, 1\}$  by (concave) quadratic constraints  $x_i(1 - x_i) \leq 0$  and  $x_i \in [0, 1]$ . Hence, assuming that the assumptions of Lemma 2.1 are satisfied and  $P_\sigma \cap G \neq \emptyset$ , it follows from Lemma 2.1 that the dual problem (3.10) has a zero duality gap in a neighborhood of  $x^+$ , and is therefore equivalent to (3.9).*

We show, how the projection problem (3.9) can be solved by column generation. Consider the following reformulation of (3.9) as a quasi-separable MINLP with a linear objective function:

$$\begin{aligned} \min \quad & c(y) + \rho \cdot \sum_{k \in K} r_k \\ & \bar{A}y \leq \bar{b} \\ & y_{I_k} \in G_k, d_k(y_{I_k}, x_{I_k}^{i-1}) \leq r_k, k \in K \end{aligned} \tag{3.11}$$

where the distance function  $d_k$  is defined as in (3.8).

Let  $S_k \subset G_k$  be a sample set consisting of previously generated trial points (columns). Then an inner approximation of (3.11) is given by

$$\begin{aligned} \min \quad & c(y) + \rho \cdot \sum_{k \in K} r_k \\ & \bar{A}y \leq \bar{b} \\ & (y_{I_k}, r_k) \in \text{conv}(\bar{S}_k), k \in K \end{aligned} \tag{3.12}$$

where

$$\bar{S}_k := \{(s, d_k(s, x_{I_k}^{i-1})) : s \in S_k\}.$$

A *pricing problem* to (3.12) is defined by

$$\min\{\bar{c}_k(y, r, \lambda^+) : y \in G_k, d_k(y, x_{I_k}^{i-1}) \leq r\} \tag{3.13}$$

where  $\bar{c}_k(y, r, \lambda) := c_k(y) + \rho \cdot r + \lambda^T \bar{A}_k y$  is the *reduced cost* of a point  $(y, r) \in G_k \times \mathbb{R}$ . For a sufficiently large  $\rho$  this problem is equivalent to the sub-problem (3.8).

From the definition of (3.13) it follows for solutions  $(y_{I_k}^+, r_k^+)$  of (3.13) with  $k \in K$ , that

$$D_G(\lambda^+, x^{i-1}) = c(y^+) + (\lambda^+)^T (\bar{A}y^+ - \bar{b}) + \rho \cdot \sum_{k \in K} r_k^+$$

A (simplified) column generation procedure for increasing the dual function  $D_G(\lambda, x^{i-1})$  is defined by repeating the following steps

1.  $(x^+, \lambda^+) \leftarrow$  primal and dual solution of (3.12)
2.  $S_k \leftarrow S_k \cup \{ \text{solutions of (3.13)} \}$  for  $k \in K$

**Observation 2.** *Since this column generation procedure is equivalent to the dual cutting plane method, it follows that  $\lambda^+$  converges to a solution of (3.10). If  $P_\sigma \cap G \neq \emptyset$  and the assumptions of Lemma 2.1 are satisfied,  $x^+$  converges towards a solution of the projection problem (3.9), because the duality gap of (3.10) is zero.*

**Remark 3.1.** *Notice that instead of solving (3.13) in step 2, it is sufficient for the convergence of the above column generation procedure to add points  $y_k^+ \in G_k$  in step 3, with negative reduced cost*

$$\bar{c}_k(y_k^+, \rho \cdot d_k(y_k^+, x_{I_k}^{i-1}), \lambda^+) < 0 \quad (3.14)$$

### 3.1.2 A globally convergent CG-based alternating direction algorithm

In order to consider integrality restrictions, the exact L1-penalty function of (2.10) is extended in the following way:

$$\Phi(x) := c(x) + \bar{\lambda} \|\bar{A}x - \bar{b}\|_{+,1} + \bar{\mu} \sum_{j \in J} |g_j(x)|_+ + \bar{\gamma} \sum_{i \in I^{\text{int}}} (1 - x_i) \cdot x_i \quad (3.15)$$

where  $\bar{\lambda}, \bar{\mu} > 0$  and  $\bar{\gamma} > 0$  are sufficiently large. Sufficient reduction of  $\Phi(x^{i-1})$  at an iteration point  $x^+$  is checked by:

$$\Phi(x^{i-1}) - \Phi(x^+) \geq \frac{\rho}{2} \|x^+ - x^{i-1}\|_2^2 + \bar{\lambda} \|\bar{A}x^+ - \bar{b}\|_{+,1} \quad (3.16)$$

**Observation 3.** *Since MINLPs can be formulated equivalently as an NLP by replacing a binary constraint  $x_i \in \{0, 1\}$  by the quadratic equality constraint  $(1 - x_i) \cdot x_i = 0$ , the result of Lemma 2.2 is also valid for the penalty function (3.15): If  $x^+$  is a solution of the projection problem (3.9), then  $x^+$  fulfills the descent criterion (3.16).*

Similarly as Algorithm 1, a globally convergent ADM for nonconvex MINLPs is shown in Algorithm 3. It uses the CG-based globalization strategy of Algorithm 4. In contrast to Algorithm 1, it stops if no new points are generated, i.e. the sample set is not changing.

ADCG is a proximal descent method, which computes a solution in the neighborhood of the starting point (because in each iteration a descent point in the neighborhood of the current trial point is computed by approximately solving a projection problem).

---

#### Algorithm 3 Globally convergent Alternating Direction CG based MINLP algorithm

---

```

1: function ADCG( $x^0, \lambda^0, S, \sigma$ )
2:    $i \leftarrow 0$ 
3:   repeat
4:      $i \leftarrow i + 1$ 
5:      $y_{I_k}^i \leftarrow$  solution of (3.8) for  $k \in K$            # project  $x^{i-1}$  onto  $G$  regarding  $\lambda^{i-1}$ 
6:      $S_k \leftarrow S_k \cup \{y_{I_k}^i\}$  for  $k \in K$ 
7:      $(x^i, \lambda^i, S) \leftarrow$  CGProj( $y^i, \lambda^{i-1}, x^{i-1}, S, \sigma$ )   # project  $y^i$  onto  $P_\sigma$  regarding  $\lambda^{i-1}$ 
8:   until  $\|x^i - y^i\|_1 \leq \epsilon$  or  $S$  is not changed
9:   return  $(x^i, y^i, \lambda^i, S)$ 

```

---

**Theorem 3.1.** *Assume that the assumptions of Theorem 2.1 are satisfied. Then for a given  $\epsilon > 0$  Algorithm 3 terminates after finitely many steps with a point  $(x, y) \in P_\sigma \times G$ . If  $\|x - y\|_1 > \epsilon$ , then either  $\sigma$  is smaller than the optimal value of (1.1) or the starting point  $x^0$  is not sufficiently close to a local optimizer. If  $\|x - y\|_1 < \epsilon$ ,  $y \in \Omega_\epsilon$  and  $c(y)$  is an upper bound on the  $\epsilon$ -optimal value  $v_\epsilon^*$  defined in (2.12).*



---

**Algorithm 4** Column Generation based projection of  $y^i$  onto  $P_\sigma$  and update of  $\lambda^i$ 


---

```

1: function CGPROJ( $y^i, \lambda^{i-1}, x^{i-1}, S, \sigma$ )
2:    $(x^+, \lambda^+) \leftarrow$  primal/dual solution of (3.7)      # project  $y^i$  onto  $P_\sigma$  regarding  $\lambda^{i-1}$ 
3:   if (3.16) then                                         #  $\Phi(x^{i-1}) - \Phi(x^+)$  is large enough
4:      $x^i \leftarrow x^+$  and  $\lambda^i \leftarrow \lambda^+$ 
5:   else
6:     if (3.16) is fulfilled for  $x^+ = y^i$  then           #  $\Phi(x^{i-1}) - \Phi(y^i)$  is large enough
7:        $x^i \leftarrow y^i$  and  $\lambda^i \leftarrow \lambda^{i-1}$ 
8:     else
9:       repeat                                             # column generation for reducing  $\text{dist}(x^+, P_\sigma \cap G)$ 
10:         $S \leftarrow S \cup \{ \text{solutions of (3.14) for } k \in K \}$  # price probl. at  $(x^{i-1}, \lambda^+)$ 
11:         $(x^+, \lambda^+) \leftarrow$  primal/dual solution of (3.12) # QP master problem at  $x^{i-1}$ 
12:      until  $S$  is not changed or (3.16)                  #  $\Phi(x^{i-1}) - \Phi(x^+)$  is large enough
13:       $x^i \leftarrow x^+$  and  $\lambda^i \leftarrow \lambda^+$ 
14:   return  $(x^i, \lambda^i, S)$ 

```

---

Proof. We adapt the global convergence proof of ALADIN in Theorem 2, [HFD14]. The proof shows that the penalty function  $\Phi$  is reduced significantly in each iteration of the algorithm. It consists of two parts.

Part 1: Assume that steps 9-13 of Algorithm 4 are performed for an infinite number of iterations.

Because of Observation 2,  $\lambda^+$  converges to a local solution of the dual problem (3.10). If the starting point  $x^0$  is sufficiently close to a local optimizer, (3.10) has a zero duality gap and the points  $x^+$  converge to a solution  $\hat{x}$  of (3.9). In this case  $\hat{x}$  fulfills the descent condition (3.16), because of Observation 3. This is a contradiction.

Part 2: If Algorithm 3 does not terminate after a finite number of steps either Step 4 or Step 7 are applied infinitely often (because of part 1 steps 9-13 is not applied infinitely often). From [HFD14] it follows, that whenever Step 4 or 7 is applied, the progress difference  $\Phi(x) - \Phi(x^+)$  is bounded from below by strictly positive constant, and that  $\Phi(x)$  is bounded from below. This is a contradiction.

Consequently, Algorithm 3 must terminate after a finite number of steps with a solution  $(x, y)$  fulfilling the statement of the theorem. □

## 3.2 A target-oriented global optimization algorithm

In this section we present a target-oriented alternating direction algorithm for computing global solutions of nonconvex MINLPs (which is not based on the branch-and-bound approach). Furthermore, it uses an improvement step, if CG does not generate new columns. If no improvement is possible and the target constraint is still unfeasible,  $P_\sigma \cap G = \emptyset$  and  $\sigma$  is a lower bound on the optimal value of (1.1).

Algorithm 5 describes the new MINLP solver. It uses a procedure SEARCH to reduce the penalty function  $\Phi$  at a trial point  $x^i$ . The method updates iteratively an interval  $[\underline{v}^i, \bar{v}^i]$  containing the optimal value  $v_\epsilon^*$ , until the diameter of the interval is smaller than a given tolerance. In each iteration, trial points  $(x^i, y^i)$  are computed using Algorithm 3 regarding a target value  $\sigma$  of the constraint (3.1).

If  $\|y^i - x^i\|_1 \leq \epsilon$ , then  $y^i$  is  $\epsilon$ -feasible and  $c(y^i) \geq v_\epsilon^*$ . In this case, the upper bound is set to  $\bar{v}^i \leftarrow c(y^i)$  and the target value is updated by  $\sigma \leftarrow (1 - \delta)\bar{v}^i + \delta\underline{v}^i$ , where  $\delta \in (0, 1)$  is an estimate for the relative duality gap of (1.1).

---

**Algorithm 5** Alternating Direction target-oriented Global Optimization algorithm

---

```

1: function ADGO( $S, P$ )
2:    $(x^0, \lambda^0) \leftarrow$  primal/dual solution of (3.7)      # solution of the outer-approximation
3:    $i \leftarrow 0, \underline{v}^0 \leftarrow c(x^0), \bar{v}^0 \leftarrow \infty, \sigma \leftarrow \infty$       # set initial lower bound
4:   repeat
5:      $i \leftarrow i + 1, stop \leftarrow \text{false}$ 
6:     repeat
7:        $(x^i, y^i, \lambda^i, S) \leftarrow$  ADCG( $x^{i-1}, \lambda^{i-1}, S, \sigma$ )      # compute trial point in  $P_\sigma$ 
8:       if  $S$  is not changed then
9:          $x^+ \leftarrow$  SEARCH( $x^{i-1}$ )      # improvement step
10:        if  $\Phi(x^+) < \Phi(x^{i-1})$  then
11:           $x^{i-1} \leftarrow x^+$ 
12:        else:  $stop \leftarrow \text{true}$ 
13:      until  $\|y^i - x^i\|_1 \leq \epsilon$  or  $stop = \text{true}$ 
14:      if  $\|y^i - x^i\|_1 \leq \epsilon$  then      # check if trial point is  $\epsilon$ -feasible
15:         $\bar{v}^i \leftarrow c(y^i), \underline{v}^i \leftarrow \underline{v}^{i-1}$       # decrease upper bound
16:         $\sigma \leftarrow (1 - \delta)\bar{v}^i + \delta\underline{v}^i$       # update target value
17:         $y^* \leftarrow y^i$ 
18:      else
19:         $\bar{v}^i \leftarrow \bar{v}^{i-1}, \underline{v}^i \leftarrow \sigma$       # increase lower bound
20:         $\sigma \leftarrow \bar{v}^i - \epsilon$       # update target value
21:      until  $\bar{v}^i - \underline{v}^i \leq \epsilon$ 
22:      return  $y^*$ 

```

---

Otherwise, if  $\|y^i - x^i\|_1 > \epsilon$ , it follows from Theorem 3.1 that  $\sigma$  is a lower bound of  $v_\epsilon^*$ . In this case, the lower bound is set to  $\underline{v}^i \leftarrow \sigma$ , and the trial point  $x^i$  is projected onto the feasible set by calling Algorithm 3 with a new target value  $\sigma \leftarrow \bar{v}^i - \epsilon$ . If the new trial-point  $y^{i+1}$  is not  $\epsilon$ -feasible, the algorithm stops, since  $\bar{v}^{i+1} - \underline{v}^{i+1} \leq \epsilon$ .

**Theorem 3.2.** *Assume that the Assumptions of Theorem 2.1 are fulfilled and the MINLP sub-problems (3.8) and (3.13) are solved to global optimality. Furthermore, assume that SEARCH is able to reduce the penalty function  $\Phi$  at a trial point  $x^i$ , if  $x^i$  is not a global minimizer. Then Algorithm 5 terminates in finitely many steps with a global  $\epsilon$ -minimizer of  $y^* \in \Omega_\epsilon$  of (1.1).*

Proof. From the construction it follows that  $v_\epsilon^* \in [\underline{v}^i, \bar{v}^i]$  defined in (2.12). Furthermore, the diameter of  $[\underline{v}^i, \bar{v}^i]$  is reduced in each iteration by a value which is bounded from above from zero.

Hence,  $\bar{v}^i - \underline{v}^i \leq \epsilon$  after finitely many iterations, and  $c(y^*) - v_\epsilon \leq \epsilon$

□

Notice that in [FGL05] a similar target-oriented strategy is proposed for the MIP Feasibility Pump (FP). After each successful run of the FP, a primal bound constraint  $c(x) \leq (1 - \delta)c(x^{i-1}) + \delta\underline{v}$  is added to the MIP, with  $\delta \in (0, 1)$ ,  $\underline{v}$  is the optimal value of an outer-approximation, and  $x^{i-1}$  is the solution from the previous run.

### 3.3 An exact global optimization algorithm using a branch-and-cut solver for the MINLP sub-problems

In this section we describe an exact global optimization algorithm, called ADOA (Alternating Direction Outer Approximation), which is based on ADGO, see Algorithm 5, using polyhedral outer-approximations for solving the MINLP sub-problems in an extended space. An overview of the components of this algorithm is shown in Figure 3.1.

ADOA consists of the following two phases:

1. In the first phase the input data  $(S, P)$  of algorithm ADGO is computed using the dual cutting plane based start heuristic StartHeu described in Algorithm 6. The cuts of the polyhedral outer-approximation  $P$  are calculated by solving LP pricing problems using the sub-solver SubLP, and the sample set  $S$  is generated using a MINLP heuristic, called SubHeu.
2. In the second phase, ADGO computes an exact global solution of the given MINLP by calling the predictor-corrector algorithm PredCorr, described in Algorithm 7, for computing solution candidates or checking if the feasible set is empty. Algorithm PredCorr computes first (possibly inexact) solution candidates using ADCG, described in Algorithm 3, where the MINLP sub-problems are solved by SubHeu. Then the solution candidates are corrected using an exact branch-and-cut solver, called SubBC.

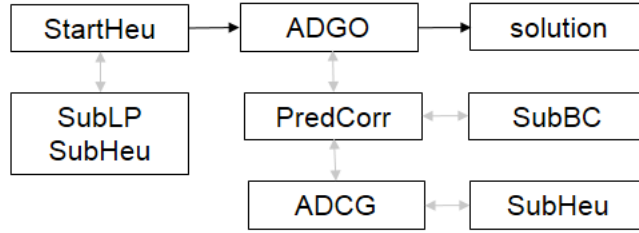


Figure 3.1: Components of the exact global optimization algorithm ADOA

### 3.3.1 Computing exact and inexact solutions of MINLP sub-problems

We describe outer-approximation based exact and inexact solver for solving the MINLP sub-problems (3.8) and (3.13) defined by

$$\min Q_k^i(y) : y \in G_k \quad (3.17)$$

where the quadratic objective function is defined by

$$Q_k^i(y) := c_k(y) + (\lambda^+)^T A_k y + \rho \cdot d_k(y, x_{I_k}^{i-1}),$$

$d_k$  is defined as in (3.8) and

$$G_k = \{y \in [\underline{x}_{I_k}, \bar{x}_{I_k}] : g_j(y) \leq 0, j \in J_k, y_i \in \{0, 1\}, i \in I_k^{\text{int}}\}.$$

Assuming that the nonlinear constraint functions  $g_j$  describing the feasible set  $G_k$  of (3.17) are given in factorable form, an outer-approximation of  $G_k$  can be computed by reformulating  $G_k$  in an extended space by

$$G_k = \{x \in \mathbb{R}^n : (x, x^{\text{ext}}) \in G_k^{\text{ext}}\} \quad (3.18)$$

where  $G_k^{\text{ext}}$  is defined by separable constraint functions, i.e. by a sum of univariate functions, which can be generated using *expression trees* [SP99].

The extended formulation (3.18) makes it possible to generate a polyhedral outer-approximation

$$\hat{G}_k^{\text{ext}} \supset G_k^{\text{ext}} \quad (3.19)$$

using linear underestimators of univariate functions.

Software packages for computing both the reformation (3.20) and a polyhedral outer-approximation (3.19) from a given MINLP are, e.g., SCIP/MINLP [Vig12] and ROSE (Reformulation/Optimization Software Engine) [LCS10].

Denote by  $s_k^*$  the minimizer of the objective function  $Q_k^i$  over the sample set  $S_k \subset G_k$ , and by  $y_k^* \in G_k$  a local minimzer of (3.17) defined by fixing the integer values of  $\hat{s}_k$  and minimizing  $Q_k^i$  regarding the continuous variables over  $G_k$  using an NLP solver starting from  $s_k^*$ .

Then the feasible set  $G_k$  of (3.17) can be reduced by adding a quadratic constraint

$$\tilde{G}_k := \{y \in G_k : Q_k^i(y) \leq Q_k^i(y_k^*)\} \subset G_k.$$

**Definition 1.** We denote by *SubBC* and by *SubHEU* a branch-and-cut algorithm and a MINLP-heuristic, respectively, for minimizing  $Q_k^i$  over  $\tilde{G}_k$  using a polyhedral outer-approximation, as defined in (3.19).

See [BV14] for an overview of branch-and-cut solvers for MINLP. Examples for outer-approximation based MINLP-heuristics are described in [Vig12, Ber14] or in [AFLL12] (MINLP feasibility pumps).

**Observation 4.** Since the distance of the trial points  $x^{i-1}$  to the sample set  $S$  decreases during iterations of Algorithm 5, the accuracy of the polyhedral outer-approximations and the solution quality of *SubHeu* increases during iterations of Algorithm 5.

### 3.3.2 Dual cutting plane based start heuristic for computing initial outer- and inner-approximations

In this section we present a dual cutting plane method, described in Algorithm 6, for initializing the polyhedral outer-approximation  $P$  defined in (3.5) and an initial inner approximation  $S \subset G$ . The method solves the following LP-outer-approximation of the MINLP (1.1) using (CG)

$$\min c(x) : Ax \leq b, (x_{I_k}, x^{\text{ext}}) \in G_k^{\text{ext}}, k \in K \quad (3.20)$$

---

**Algorithm 6** Dual cutting plane based start heuristic

---

```

1: function STARTHEU
2:   Compute outer-approximations  $\hat{G}_k^{\text{ext}} \supset G_k^{\text{ext}}, k \in K$ 
3:    $S \leftarrow \emptyset$ 
4:   repeat
5:      $(\lambda^+, x^+) \leftarrow$  dual/primal solution of (3.21)           # linear master problem
6:     for  $k \in K$  do
7:       # Update  $P$  by adding cuts via solutions of LP-pricing problems at  $\lambda^+$ 
8:        $(v_k^+, s_{I_k}^+) \leftarrow$  optimal value/solution of (3.22)     # using SubLP
9:       Add cut  $c_k(x_{I_k}) + (\lambda^+)^T A_k x_{I_k} \geq v_k^+$  to  $P$      # update  $P$ 
10:      # Update  $S$  by projecting  $s_k^+$  onto  $G_k$  regarding  $\lambda^+$ 
11:       $S_k^+ \leftarrow$  solutions of (3.23) in the neighborhood of  $s_k^+$  # using SubHeu
12:       $S_k \leftarrow S_k \cup S_k^+$                                      # update  $S$ 
13:   until  $c_k(s^+) + (\lambda^+)^T A_k s^+ \geq -\epsilon$  for all  $k \in K$ 
14:   return  $(S, P)$ 

```

---

In each iteration of StartHeu the following sub-problems are solved:

- the LP master problem

$$\min c(x) : Ax \leq b, x \in P \quad (3.21)$$

- the LP pricing sub-problems regarding  $\hat{G}_k^{\text{ext}}$ :

$$\min c_k(s) + (\lambda^+)^T A_k s : (s, s^{\text{ext}}) \in \hat{G}_k^{\text{ext}} \quad (3.22)$$

- and the MINLP Lagrangian sub-problems regarding  $G_k$

$$\min c_k(y) + (\lambda^+)^T A_k y : y \in G_k \quad (3.23)$$

### 3.3.3 Accelerating ADGO by an inexact outer-approximation based sub-solver

In each iteration of Algorithm 5 several ADM sub-problems (3.8) and pricing sub-problems (3.13) have to be solved to global optimality. In principle, these MINLP sub-problems can be solved using the exact branch-and-cut solver SubBC. However, this can be time-consuming. Moreover, because the quadratic objective function of the MINLP-sub-problems depends on the last trial point, cuts which are generated in one iteration must not be valid in the next iteration. This makes the development of a fast branch-and-cut based sub-solver using warm-start difficult.

Therefore, we propose to use the fast predictor-corrector variant of Algorithm 3 shown in Algorithm 7. It computes first an inexact solution using an inexact sub-solver SubHEU, and corrects it using an exact branch-and-cut sub-solver SubBC.

---

#### Algorithm 7 Predictor-corrector ADCG solver

---

```

1: function PREDCORR( $x, \lambda, S, \sigma$ )
2:   repeat
3:     # predict: compute a near feasible trial point or check  $P_\sigma \cap G = \emptyset$ 
4:      $(x, y, \lambda, S) \leftarrow$  ADCG( $x, \lambda, S, \sigma$ ) # using SubHEU
5:     # correct: solve the pricing problems regarding  $\lambda$  and  $x$  exactly
6:      $S_k^+ \leftarrow$  solutions of (3.13) for  $k \in K$  # using SubBC
7:      $S_k \leftarrow S_k \cup S_k^+$  for  $k \in K$ 
8:   until  $S_k^+ = \emptyset$  for all  $k \in K$  or  $D_G(\lambda, x) > d_G(x)$  # w.r.t the best solution of (3.13)
9:   return  $(x, y, \lambda, S)$ 

```

---

**Proposition 3.1.** *Algorithm 7 computes the same solution as Algorithm 3 using SubBC as a MINLP sub-solver.*

*Proof.* The statement can be proved in the same way as Theorem 3.2. Assume that steps 3-7 in Algorithm 7 are performed infinitely many times without changing the trial point  $x$ . Then  $\lambda$  converges to a local solution of the dual problem (3.10) at  $x$ . It follows that, either the objective of (3.10) tends to infinity, if (3.2) is unfeasible, or  $x$  fulfills the descent condition (3.16). In the first case the algorithm stops, and in the second case  $x$  is changed. This is a contradiction.

From the proof of Theorem 3.2 it follows, that  $x$  cannot be changed infinitely many times. Hence, Algorithm 7 stops after finitely many iterations with the same solution as the exact version of Algorithm 3. □

## 3.4 Variants

We show that Algorithm 5 can also be used for solving network and black-box optimization problems.

### 3.4.1 Solving network optimization problems

Let  $\mathcal{N}_k$  be a network defined by arcs  $(i, j) \in A_k$ ,  $k \in K$ . Define the set of all paths of  $\mathcal{N}_k$  by

$$\mathcal{P}(\mathcal{N}_k) := \{x = (x_{i,j})_{(i,j) \in A_k} : x_{i,j} \in \{0, 1\}, (i, j) \in A_k, x \text{ defines a path in } \mathcal{N}_k\}$$

The set of path of  $\mathcal{N}_k$  fulfilling resource constraints is defined by

$$G_k = \{x \in \mathcal{P}(\mathcal{N}_k) : g_j(x) \leq 0, j \in J_k\}$$

where  $g_j(x)$  is a linear or nonlinear resource constraint.

A simplified arc-based formulation of a network optimization problem is given by

$$\begin{aligned} \min \quad & c(x) \\ & Ax \leq b \\ & x_{I_k} \in G_k, k \in K \end{aligned} \tag{3.24}$$

An example for a network optimization problem is a crew roster or pairing problem, where  $k$  represents a crew member or a group of crew members and  $x_{I_k}$  represents a roster or a pairing consisting of duties and transports (typically for one month).

Using column generation it is possible to compute solutions of crew scheduling instances with more than 700 million arc-variables by solving pricing problems of (3.24) defined by

$$\min\{\bar{c}_k(y, \lambda) : y \in G_k\} \tag{3.25}$$

where  $\bar{c}_k(y, \lambda) := c_k(y) + \lambda^T A_k y$  is the *reduced cost*, see e.g. [Now08, BLR<sup>+</sup>13].

The pricing problem is typically solved by a dynamic programming based constrained shortest path solver, see e.g. [ENS08]. Because of the huge problem size, the search space of (3.25) is dynamically reduced [Now08].

Furthermore, in order to avoid many branching operations, *Rapid Branching* is used for solving the integer master problem. The idea of this method is to find a near-optimal solution of (3.24) by successively solving a *perturbed master problem* and generating new columns, in order to move its solution towards the feasible set [BLR<sup>+</sup>13].

Heuristics for solving huge integer master problems, like Rapid Branching, may work well if the duality gap is small, however, they do not provide lower bounds of the objective value, and are not able to rigorously improve a solution.

In contrast, it is possible to calculate lower bounds using Algorithm 5, and the trial points  $y^i$  converge towards a global optimizer, provided that the sub-problems are solved exactly. Similar as in ADOA, an initial sampling set  $S$  of columns and a fractional solution  $\hat{x} \in \text{conv}(S)$  of the restricted master problem can be computed using traditional column generation. Then Algorithm 5 is used to move  $\hat{x}$  towards the feasible set of (3.24), where new columns are generated by solving the quadratic constrained shortest path sub-problems (3.8) and (3.13) (using dynamic programming).

In order to compute an outer-approximation  $P$  of the feasible set (3.24) using Algorithm 6, a polyhedral outer-approximation  $\hat{G} \supseteq G$  has to be defined. This can be done, e.g., by defining  $\hat{G}$  by linear resource constraints, i.e.

$$\hat{G}_k := \{x \in \mathcal{P}(\mathcal{N}_k) : g_j(x) \leq 0, j \in J_k^{\text{lin}}\}$$

where  $J_k^{\text{lin}} \subseteq J_k$  denotes the index set of linear resource constraints.

### 3.4.2 Solving black-box optimization problems

A quasi-separable simulation-based black-box problem, like a multidisciplinary optimization (MDO) problem, is defined as in (1.1), where some or all nonlinear constraint functions  $g_j, j \in J$ , are black-box functions, i.e.  $g_j(x)$  can only be evaluated for  $x \in [\underline{x}, \bar{x}]$ , but no algebraic description of  $g_j$  is available.

In [TEPR06] an ADM for solving quasi-separable black-box problem is proposed. Algorithm 5 can also be used to solve such problems, if a Surrogate Optimization Algorithm (SOA) is used for solving the sub-problems (3.8) and (3.13), see e.g. [EBS15, FSK08]. SOA computes sample points by solving pricing problems regarding search directions generated by Algorithm 5, and generates simultaneously other sample points for fitting surrogate models  $\check{g}_j$  of the nonlinear constraint functions  $g_j$ .

An initial sample set for initializing Algorithm 5 can be generated by traditional sampling methods, like the Latin Hypercube Method. Similar as in Algorithm 7, first the surrogate model is solved using Algorithm 5 (where  $g_j$  is replaced by its surrogate model  $\check{g}_j$ ), and then the surrogate models of the sub-problems are refined by performing expensive function evaluations of  $g_j$  at new sample points.

## 4 Final Remarks

We presented a novel target-oriented decomposition algorithm for solving quasi-separable nonconvex MINLPs, called ADGO, by combining column generation and a globally convergent alternating direction method. Moreover, we presented the exact global optimization algorithm ADOA, which is based on ADGO using both a branch-and-cut solver and a heuristic for solving the MINLP sub-problems.

ADGO moves an initial solution of a convex relaxation successively towards the feasible set by approximately solving the dual of a nonconvex projection problem using column generation. The projection problem can be solved efficiently, because it is quasi-separable and has a zero duality gap. The ADM and pricing sub-problems can be solved in parallel, and a lower bound of the optimal value can be computed by adding a target constraint and checking if the restricted feasible set is empty.

Solving the sub-problems of ADGO is not much more difficult than solving traditional pricing problems, since the only difference is that a convex quadratic objective function is minimized instead of a linear objective function.

Provided that an efficient method SEARCH is available, which is able to reduce the penalty function  $\Phi$  at a trial point, we expect that the computational cost of ADGO is similar to a traditional column generation method.

In order to test the performance of the new solvers ADGO and ADOA, it is planned to implement them using SCIP/MINLP, and to present numerical results for nonconvex MINLPs in a subsequent paper. It would also be interesting to apply the new approach to network and black-box optimization problems, as discussed in section 3.4.

**Acknowledgements** I thank Stefan Vigerske for valuable comments on a preliminary version of the paper.

## References

[AFLL12] C. D' Ambrosio, A. Frangioni, L. Liberti, and A. Lodi. A storm of feasibility pumps for nonconvex minlp. *Mathematical Programming*, 136(2):375–402, 2012.

- [Ber14] T. Berthold. *Heuristic algorithms in global MINLP solvers*. PhD thesis, Technical University Berlin, 2014.
- [BL12] S. Burer and A. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17 (2):97–106, 2012.
- [BLR<sup>+</sup>13] R. Borndörfer, A. Löbel, M. Reuther, T. Schlechte, and S. Weider. Rapid branching. *Public Transport*, 5:3–23, 2013.
- [BPC<sup>+</sup>11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.
- [BV14] M. R. Bussieck and S. Vigerske. *MINLP Solver Software*. [www.math.hu-berlin.de/~stefan/minlpsoft.pdf](http://www.math.hu-berlin.de/~stefan/minlpsoft.pdf), 2014.
- [DL10] J. Desrosiers and M. Lübbecke. Branch-price-and-cut algorithms. In J. Cochran, L. Cox, P. Keskinocak, J. Kharoufeh, and J. Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010.
- [EBS15] D. Eriksson, D. Bindel, and C. Shoemaker. *Surrogate Optimization Toolbox (pySOT)*. [github.com/dme65/pySOT](https://github.com/dme65/pySOT), 2015.
- [ENS08] F. Engineer, G. Nemhauser, and M. Savelsbergh. Shortest path based column generation on large networks with many resource constraints. Technical report, Georgia Tech, 2008.
- [FGL05] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Math. Progr.*, 104(1):91–104, 2005.
- [FSK08] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.
- [GM76] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications*, 2:17–40, 1976.
- [Han77] S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.
- [HFD14] Boris Houska, Janick Frasch, and Moritz Diehl. *An Augmented Lagrangian based Algorithm for Distributed Non-Convex Optimization*. [www.optimization-online.org/DB\\_HTML/2014/07/4427.html](http://www.optimization-online.org/DB_HTML/2014/07/4427.html), 2014.
- [LCS10] L. Liberti, S. Caferi, and D. Savourey. The reformulation-optimization software engine. In K. Fukuda, J. Hoeven, M. Joswig, and N. Takayama, editors, *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 303–314. Springer Berlin Heidelberg, 2010.
- [LMSS14] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. *Solving Power-Constrained Gas Transportation Problems using an MIP-based Alternating Direction Method*. [www.optimization-online.org/DB\\_HTML/2014/11/4660.html](http://www.optimization-online.org/DB_HTML/2014/11/4660.html), 2014.
- [Now05] I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Birkhäuser, 2005.
- [Now08] I. Nowak. *A Dynamic Reduce and Generate Approach for Airline Crew Scheduling*. [www.gerad.ca/colloques/ColumnGeneration2008/slides/IvoNowak.pdf](http://www.gerad.ca/colloques/ColumnGeneration2008/slides/IvoNowak.pdf), 2008. GERAD International Workshop on Column Generation, Aussois.



- [Now14] I. Nowak. *Parallel Decomposition Methods for Nonconvex Optimization - Recent Advances and New Directions*, 2014. Proceedings of MAGO.
- [NW06] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.
- [RG06] T. Ralphs and M. Galati. Decomposition and dynamic cut generation in integer linear programming. *Mathematical Programming*, 106(2):261–285, 2006.
- [Roc74] R. T. Rockafellar. Augmented Lagrangian multiplier functions and duality in nonconvex programming. *SIAM J. Control*, 12(2):268–285, 1974.
- [SP99] E. M. B. Smith and C. C. Pantelides. A Symbolic Reformulation/Spatial Branch and Bound Algorithm for the Global Optimization of nonconvex MINLPs. *Comp. Chem. Eng.*, 23:457–478, 1999.
- [TEPR06] S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda. An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Struct Multidisc Optim*, 31:176–189, 2006.
- [Uza58] H. Uzawa. Iterative methods for concave programming. *Studies in Linear and Nonlinear Programming*, pages 154–165. 1958.
- [Vig12] S. Vigerske. *Decomposition in Multistage Stochastic Programming and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt-Universität zu Berlin, 2012.