Mathematical Programming
https://doi.org/10.1007/s10107-021-01740-0

FULL LENGTH PAPER

Series B

Non-convex nested Benders decomposition

Christian Füllner[1] · Steffen Rebennack[1]

Received: 2 April 2020 / Accepted: 8 November 2021

Abstract

We propose a new decomposition method to solve multistage non-convex mixed-integer (stochastic) nonlinear programming problems (MINLPs). We call this algorithm *non-convex nested Benders decomposition* (NC-NBD). NC-NBD is based on solving dynamically improved mixed-integer linear outer approximations of the MINLP, obtained by piecewise linear relaxations of nonlinear functions. Those MILPs are solved to global optimality using an enhancement of nested Benders decomposition, in which regularization, dynamically refined binary approximations of the state variables and Lagrangian cut techniques are combined to generate Lipschitz continuous non-convex approximations of the value functions. Those approximations are then used to decide whether the approximating MILP has to be dynamically refined and in order to compute feasible solutions for the original MINLP. We prove that NC-NBD converges to an ε-optimal solution in a finite number of steps. We provide promising computational results for some unit commitment problems of moderate size.

Keywords Nested Benders decomposition · Mixed-integer nonlinear programming (MINLP) · Global optimization · Non-convexities · Non-convex value functions

Mathematics Subject Classification 90C26 · 90C11 · 49M27

1 Introduction

We propose a new decomposition method to solve multistage non-convex mixed-integer (stochastic) nonlinear programming problems (MINLPs), *i.e.*, optimization problems modeling a sequential decision making process. Continuous and integer

✉ Christian Füllner
christian.fuellner@kit.edu

Steffen Rebennack
steffen.rebennack@kit.edu

[1] Institute for Operations Research (IOR), Stochastic Optimization (SOP), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Published online: 07 January 2022

decision variables and possibly non-convex objective functions and constraints are allowed for any of the T stages.

If multistage (stochastic) problems are too large to be solved by off-the-shelf solvers, then tailored solution techniques are required. One example are decomposition algorithms making use of the specific sequential and block-diagonal structure of the constraints. The problems are decomposed into a large number of smaller but coupled subproblems which are solved iteratively. One of the most common decomposition methods is Benders decomposition, introduced by Benders [6] for linear programs. Since then, it has been enhanced to several more general cases, such as convex problems (*generalized Benders decomposition* (GBD) [19]), two-stage stochastic linear problems (*L-shaped method* [49]) and multistage (stochastic) linear problems (*nested Benders decomposition* (NBD) [8]). To mitigate the curse-of-dimensionality related to NBD in the stochastic case, Pereira and Pinto introduced its sampling-based variant *stochastic dual dynamic programming* (SDDP) [35], which was followed by various extensions [23,37].

The basic principle of NBD is to use the dynamic programming formulation of a given multistage problem. For each stage $t \in \{1, \dots, T\}$, a parametric subproblem is considered. This subproblem contains only those constraints, variables and parts of the objective function related to this specific stage, plus a *value function* determining the optimal value of all following stages for a given stage t solution. Since the value functions are not known in advance, they are iteratively approximated with linear cutting-planes. However, this approach requires the value functions to be convex. Therefore, most decomposition methods for multistage problems cover linear programs, as their value functions are guaranteed to be piecewise linear and convex.

However, in many applications, also integer variables or non-linearities occur naturally. In such case, the value functions are no longer convex and may also no longer be continuous. Therefore, the classical Benders approach fails, as it is impossible to construct a tight convex polyhedral approximation [47].

Thus, more sophisticated approaches have been developed to use Benders-type decomposition methods for non-convex MINLPs, mostly for the two-stage case. Li et al. propose an extension of GBD to the non-convex case for two-stage stochastic MINLPs with functions separable in integer and continuous variables [29,30]. In [28], a branch-and-cut framework is presented, where in each node Lagrangian and generalized Benders cuts are constructed. Related methods are proposed in [26,33]. All these methods have not been generalized to the multistage case yet.

To handle non-convexities in multistage problems, a common idea is to use convex relaxations of the value function, *e.g.*, by relaxing the integrality constraints for MILPs or by convexifying nonlinear terms in a static manner. Dynamically convexifying the non-convex value functions using Lagrangian relaxation techniques allows for a polyhedral approximation by Lagrangian cuts [10,45,46]. None of these discussed approaches can guarantee to compute an optimal solution for non-convex multistage problems, though.

Only recently, some substantial progress has been made in generalizing the Benders decomposition idea to multistage problems with non-convex value functions directly. In [36], step functions are used, instead of cutting-planes, to approximate the value functions, presuming their monotonicity.

For the mixed-integer linear case, the *stochastic dual dynamic integer programming* (SDDiP) approach is proposed [55]. SDDiP is an enhancement of NBD and SDDP which allows the solution of multistage (stochastic) MILPs in case of binary state variables. The method is based on generating special Lagrangian cuts, which reproduce the lower convex envelope of the value function. As the latter is piecewise linear and exact at binary state variables, strong duality is ensured and the problem is solved to global optimality in a finite number of iterations. SDDiP is applied to multistage unit commitment in [54]. It is also applied to a problem containing non-convex functions in context of hydro power scheduling by using a static binary expansion of the state variables and a Big-M reformulation [22].

As long as the value functions are assured to be Lipschitz continuous and some recourse property is satisfied, the requirement of binary state variables can be dropped, as is shown by the Stochastic Lipschitz Dynamic programming (SLDP) method in [1]. Here, two types of non-convex Lipschitz continuous cuts are introduced: reverse-norm cuts and augmented Lagrangian cuts.

In [52], Zhang and Sun present a new framework to solve multistage non-convex stochastic MINLPs, generalizing both SDDiP and SLDP. Similarly to [1], nonlinear *generalized conjugacy cuts* are constructed by solving augmented dual problems. Moreover, as Lipschitz continuity is not assured for the value functions, a Lipschitz continuous regularized value function is considered within the decomposition method.

In this article, we propose a new method to solve multistage non-convex MINLPs to proven global optimality, which we refer to as *non-convex nested Benders decomposition* (NC-NBD). The method combines piecewise linear relaxations, regularization, binary approximation and the SDDiP Lagrangian cuts in a unique and dynamic fashion. Its basic idea is to solve a MINLP by iteratively improved MILP outer approximations, which in turn are solved using a NBD-based decomposition scheme similar to that in [52]. The binary and piecewise linear approximations are dynamically refined.

In particular, the original MINLP is outer approximated by MILPs, which are iteratively improved in an outer loop. Those MILPs are obtained by piecewise linear approximations of all occuring nonlinear functions, which is an established method in global optimization [50]. In general, using MILP relaxations is a common approach to global optimization solvers [27,32,53].

In an inner loop, the multistage MILPs are solved to approximate optimality in finitely many steps. This is achieved using a NBD-based decomposition method. In a forward pass through the stages, trial solutions for the dynamic programming equations are determined. As Lipschitz continuity of the value functions is not guaranteed, this is done solving a regularized forward pass problem, as proposed in [52]. For a sufficiently large, but finite parameter, the regularization is exact [14,52], so that still the desired MILP is solved.

In a backward pass through the stages, nonlinear non-convex cuts are constructed to approximate the non-convex value functions of the MILP. To this end, we make use of a binary approximation of the state variables in the subproblems. As proven in [55], for MILPs with binary state variables we obtain (sufficiently) tight cuts by solving Lagrangian dual problems. The constructed linear cuts are then projected back to the original state space, yielding a nonlinear, non-convex, but Lipschitz continuous approximation of the value functions. The binary approximation is refined dynamically within the inner loop if required. By careful construction, all existing cuts remain valid even with such refinements.

Once the MILP approximation is solved to approximate optimality, the cut approximation of the value functions is used in the outer loop to determine bounds for the optimal value of the original MINLP. If the bounds are sufficiently close, the algorithm terminates with an ε-optimal solution. Otherwise, the piecewise linear approximations are refined, and thus the approximating MILP is tightened. Again, by careful construction it is ensured that all previously generated cuts remain valid.

To our best knowledge, the above concepts have not been combined in this dynamic way to solve multistage non-convex MINLPs yet. In that regard, our work also differs significantly from the aforementioned solution techniques.

Our proposed decomposition scheme uses the same regularization technique and similar convergence ideas as in [52]. However, a fundamental difference is that we only apply this technique to solve MILP outer approximations of the original MINLP. This has the advantage that in our framework MINLPs have to be solved only occasionally. In contrast, in [52], MINLPs are assumed to be solved by some oracle in each iteration and cuts are generated directly for the MINLP, which is computationally challenging. Moreover, contrary to our approach, the method in [52] does not require recourse assumptions, but in return it only allows for state variables in the objective function.

In contrast to SDDiP [55] and SLDP [1], we solve MINLPs, and thus consider a larger solution framework with an inner and an outer loop. However, even the inner loop, in which MILPs are solved, differs from both approaches.

To solve MILPs with non-binary state variables using SDDiP, it is proposed to apply a *static* binary approximation [22,55]. This way, the original MILP is replaced by an approximating problem with only binary state variables. It can be shown that for a sufficiently small approximation precision, *i.e.*, an sufficiently large number of binary variables, an ε-optimal solution of an MILP can be determined with this approach under some recourse assumption [55]. However, for a given problem at hand, it is not necessarily clear in advance how this precision has to be chosen, as knowledge on a problem-specific Lipschitz constant is required. This becomes even more challenging in our framework, where an MINLP is iteratively approximated by MILPs, for which the required precision may change. On the contrary, within NC-NBD the binary approximation is refined dynamically if required.

More crucially, in NC-NBD the binary approximation is applied temporarily only to derive cuts in the backward pass. These cuts are then projected back to the original state space. This construction has a few key advantages: Firstly, it is ensured that cuts remain valid even if the binary precision is refined later on. Secondly, the original state variables remain continuous and are not limited to values which can be exactly represented by the binary approximation. This, in turn, ensures that the true MILPs are solved in the inner loop. Consequently, the generated cuts are valid for the value functions of these MILPs and, due to their relaxation property, also the original MINLP. Analogously, the obtained lower bounds are valid for the corresponding optimal values. Importantly, this is not true for SDDiP with static binary approximation, where the state space is permanently modified and only approximations of the true MILPs are solved in the inner loop. In our approach to solve MINLPs, it is crucial to determine guaranteed valid cuts for the value functions in both loops. Therefore, SDDiP cannot be used effectively in this setting.

Our cut generation approach also differs from that in SLDP [1] (and also [52]), where augmented Lagrangian problems are solved to determine nonlinear cuts. While our method comes at the cost of introducing additional (binary) variables and constraints compared to those approaches, *e.g.*, for the cut projection, we avoid solving dual problems containing nonlinear penalization in the objective. Such penalization may be disadvantageous as it prevents decomposition of the primal problems which are solved in the solution process of the dual problem. Additionally, in contrast to SLDP [1], we do not assume continuously complete recourse, but only the weaker complete recourse, as we circumvent the requirement of Lipschitz continuity of the true value functions by regularization.

The main contributions of this paper are as follows:

(1) We present the *non-convex nested Benders decomposition* (NC-NBD) method to globally solve general multistage non-convex MINLPs. The method combines piecewise linear relaxations, regularization, binary approximation and cutting-planes techniques in a unique way. In contrast to existing approaches, all approximations are improved dynamically where and when it is reasonable. To our knowledge, this is the first decomposition method for general multistage non-convex MINLPs.

(2) A crucial requirement using dynamic refinements is to ensure that all previously determined cuts remain valid within the refinement process and have not to be generated from scratch. We ensure this by a special cut projection and careful choice of the MILP relaxations.

(3) We prove that the proposed NC-NBD method converges to an ε-optimal solution of P in a finite number of steps under some mild assumptions.

(4) We provide first computational results of applying NC-NBD to moderate-sized instances of a unit commitment problem to illustrate its efficacy.

To enhance readability, we focus our discussions solely on deterministic MINLPs. However, the presented NC-NBD idea can also be applied to stochastic programs with stagewise independent and finite random variables.

The remainder of the paper is organized as follows. We present the considered problem formulation and assumptions in Sect. 2. Then, we introduce the NC-NBD with its different steps in Sect. 3, before presenting convergence results in Sect. 4. Afterwards, we provide computational results for instances of a simple unit commitment problem in Sect. 5. We conclude with Sect. 6.

2 Problem formulation

We consider the following multistage non-convex MINLP problems

$$(\boldsymbol{P}) \quad v := \min_{x_1,\dots,x_T,y_1,\dots,y_T} \quad \sum_{t=1}^{T} f_t(x_t, y_t)$$
$$\text{s.t.} \quad (x_t, y_t) \in M_t(x_{t-1}) \quad \forall t = 1, \dots, T.$$

Here $t = 1, \dots, T$ denotes the different stages with the final stage $T \in \mathbb{N}$. For each stage t, the decision variables can be separated into mixed-integer state variables $x_t \in \mathbb{R}_+^{n_t^1} \times \mathbb{Z}_+^{n_t^2}$ and local variables $y_t \in \mathbb{R}^{n_t^3} \times \mathbb{Z}^{n_t^4}$, with $x_0 = 0$. We define $n_t := n_t^1 + n_t^2$ as the number of state variables. The sets $M_t(x_{t-1})$ appearing in the constraints for each stage t are defined by

$$M_t(x_{t-1}) := \{(x_t, y_t) \in X_t \times Y_t \ : \ g_t(x_{t-1}, x_t, y_t) \leq 0, \ h_t(x_{t-1}, x_t, y_t) = 0\}.$$

X_t and Y_t denote box constraints; $X_0 := \{0\}$. As such, X_t and Y_t are compact sets for all stage-t variables. All functions $f_t : X_t \times Y_t \to \mathbb{R}$, $g_t : X_{t-1} \times X_t \times Y_t \to \mathbb{R}^{m_t^1}$ and $h_t : X_{t-1} \times X_t \times Y_t \to \mathbb{R}^{m_t^2}$ are well-defined on their domains.

To exploit its multistage structure, we solve $(\boldsymbol{P})$ by some extension of NBD. NBD makes use of the dynamic programming formulation of $(\boldsymbol{P})$, where each stage-t subproblem, $t = 1, \dots, T$, can be denoted by

$$\begin{aligned}(\boldsymbol{P_t(x_{t-1})}) \quad Q_t(x_{t-1}) := \min_{x_t, y_t, z_t} \quad & f_t(x_t, y_t) + Q_{t+1}(x_t) \\ \text{s.t.} \quad & (z_t, x_t, y_t) \in M_t \\ & z_t = x_{t-1},\end{aligned}$$

with the value function $Q_t(\cdot)$ of stage t and $Q_{T+1}(\cdot) \equiv 0$. Note that x_t links different stages, *i.e.*, x_t is a decision variable for $(\boldsymbol{P_t(x_{t-1})})$ and a parameter for $(\boldsymbol{P_{t+1}(x_t)})$. For the first stage, we obtain that $Q_1(x_0) = v$ with $x_0 \equiv 0$. Importantly, subproblem $(\boldsymbol{P_t(x_{t-1})})$ is enhanced by introducing local copies z_t of the state variables x_{t-1} and the *copy constraints* $z_t = x_{t-1}$. Those copy constraints will prove crucial for the cut generation later on. Taking into account the local copies, we define

$$M_t := \{(z_t, x_t, y_t) : z_t \in X_{t-1}, (x_t, y_t) \in M_t(z_t)\}.$$

As the subproblems $(\boldsymbol{P_t(x_{t-1})})$ are non-convex MINLPs, the value functions $Q_t(\cdot)$ may be non-continuous and non-convex, two detrimental properties for Benders decomposition approaches. To ensure that the value functions $Q_t(\cdot)$ are at least lower semicontinuous (l.sc.), we make the following technical assumptions:

(A1). For all $t = 1, \dots, T$,

(a) the functions f_t are Lipschitz continuous on $X_t \times Y_t$,
(b) the functions g_t and h_t are continuous on $X_{t-1} \times X_t \times Y_t$.

(A2) (Complete recourse). For any stage t and any $\bar{x}_{t-1} \in X_{t-1}$, there exists some $(z_t, x_t, y_t) \in X_{t-1} \times X_t \times Y_t$ which is feasible for $(\boldsymbol{P_t(\bar{x}_{t-1})})$.

As all variables are box-constrained, the feasible set $M_t(x_{t-1})$ of $(\boldsymbol{P_t(x_{t-1})})$ is bounded. With assumption **(A1)** and the recourse assumption **(A2)**, all subproblems $(\boldsymbol{P_t(x_{t-1})})$ are feasible and bounded. Analogously, $(\boldsymbol{P})$ is feasible with finite optimal value v. Note that under assumption **(A2)** we can restrict to generating *optimality cuts* in NC-NBD without the need to introduce *Benders feasibility cuts*.

We obtain our required l.sc. property of the value functions $Q_t(\cdot)$.

Lemma 2.1 *Under assumptions **(A1)** and **(A2)** the value functions $Q_t(\cdot)$ are l.sc. for all $t = 1, \ldots, T$.*

Proof Fixing all integer variables, the l.sc. follows from Exercise 1.19 in [41]. As X_t and Y_t are bounded, only finitely many different values can be attained by the integer variables. The minimum of finitely many l.sc. functions is l.sc. □

In the next section, we introduce the NC-NBD method, which combines regularization, piecewise linear approximations, binary expansion and special cutting-plane techniques in a unique way to solve $(\boldsymbol{P})$.

3 Non-convex nested Benders decomposition

3.1 The NC-NBD principle

The basic idea of the NC-NBD algorithm is to employ that MILP problems can be solved exactly by enhancements of NBD under certain assumptions and that MINLPs can be outer approximated by MILPs iteratively. Thus, the method consists of two main components. The first component is an inner loop which is used to determine an approximately optimal solution of some MILP outer approximation $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ of problem $(\boldsymbol{P})$. This approximation is determined by piecewise linear relaxations of nonlinear functions in $(\boldsymbol{P})$. The second component is an outer loop which refines this outer approximation iteratively (indexed by ℓ) to improve the approximation of the optimal value v of $(\boldsymbol{P})$. The NC-NBD is summarized in Algorithm 1 and illustrated in Fig. 1.

The inner loop follows the general principle of NBD to solve $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$. It consists of a *forward* and a *backward pass* through the stages $t = 1, \ldots, T$ in each iteration i. In the forward pass, the stage-t subproblem $(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}^{\boldsymbol{\ell}}(\boldsymbol{x}_{\boldsymbol{t-1}}))$ is approximated in two different ways: The value function $\widehat{Q}_{t+1}(\cdot)$ of the following stage is replaced by some outer approximation $\mathfrak{Q}_{t+1}^{\ell i}(\cdot)$. Moreover, a regularization is added to ensure Lipschitz continuity of the corresponding value functions. Thus, regularized subproblems $(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}^{\boldsymbol{R,\ell i}}(\boldsymbol{x}_{\boldsymbol{t-1}}))$ are solved, as proposed in [52], yielding trial solutions $\widehat{x}_{t-1}^{\ell i}$ and an upper bound $\overline{\widehat{v}}^{\ell i}$ for $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$.

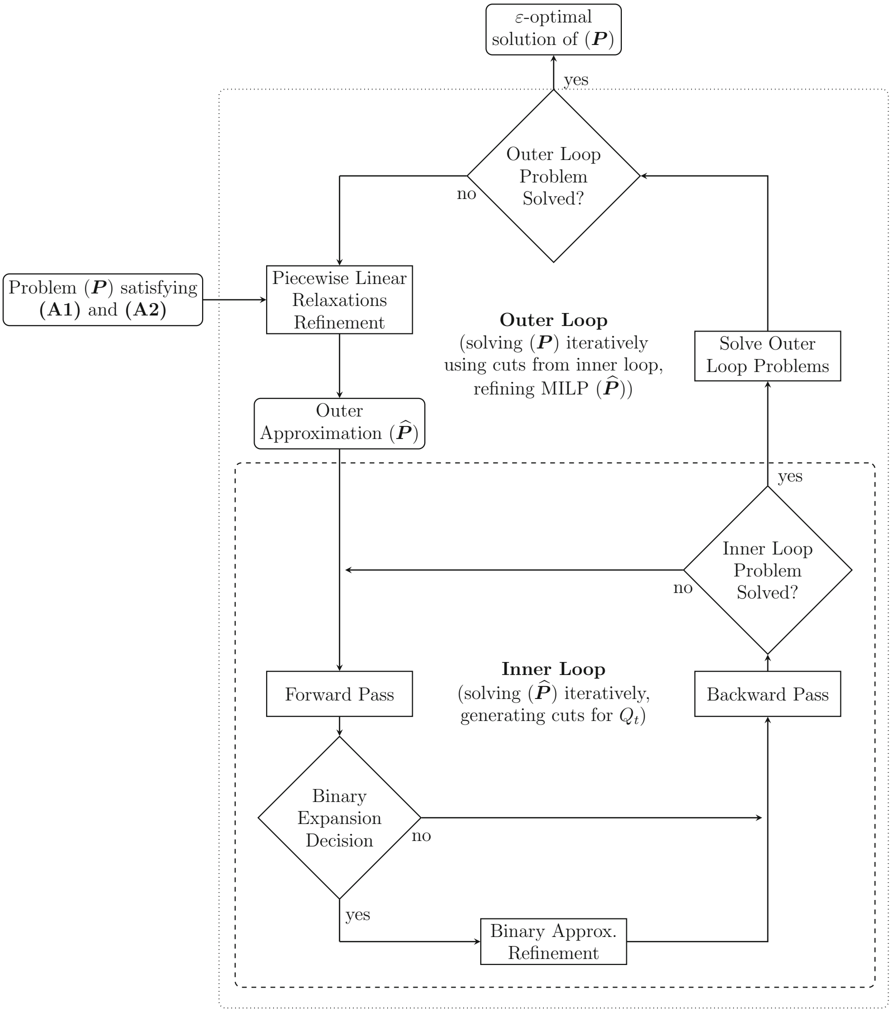
In the backward pass, the approximations $\mathfrak{Q}_{t+1}^{\ell i}(\cdot)$ of $\widehat{Q}_{t+1}(\cdot)$ are improved iteratively by constructing additional cuts. As the value functions are possibly non-convex, those cuts are nonlinear. Importantly, cuts for $\widehat{Q}_{t+1}(\cdot)$ are also valid for $Q_{t+1}(\cdot)$, as the first is an outer approximation of the latter.

In the literature, different ways are proposed to obtain nonlinear optimality cuts and to ensure that the inner loop converges to the optimal value $\widehat{v}^{\ell}$ of $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$. One method is to generate reverse-norm cuts [1]. However, this only works if the value functions themselves are Lipschitz continuous which is not guaranteed in our setting. Another, more general method is to solve some augmented Lagrangian dual problem, as proposed in [1,52].

We propose a third and new method, based on the SDDiP technique [55]. We utilize that we can generate sufficiently tight cuts by solving a Lagrangian dual in a lifted space, where all state variables are binary. Thus, we (temporarily) approximate the

Algorithm 1 NC-NBD

```
Input: Problem $(\boldsymbol{P})$ satisfying (A1), (A2), tolerances $\varepsilon > \widehat{\varepsilon} > 0$, scalar $K_t$ for all $t$, initial bin. approx.
    precision $\beta_t \in (0,1)^{K_{t-1}}$, upper bounds $\overline{v}^0 = +\infty$, lower bound $\underline{\widehat{v}}^0$, initial $\mathfrak{Q}_t^0(\cdot)$ for all $t$ and triangulations
    $\mathcal{T}_\gamma^0$ for all $\gamma \in \Gamma$, $\ell \leftarrow 0$.
while $\overline{v}^\ell - \underline{\widehat{v}}^\ell > \varepsilon$ do
    Set $\ell \leftarrow \ell + 1, i \leftarrow 1$. Set $\mathfrak{Q}_t^{\ell,1}(\cdot) \leftarrow \mathfrak{Q}_t^{\ell-1}(\cdot), \underline{\widehat{v}}^{\ell,0} \leftarrow \underline{\widehat{v}}^{\ell-1}, \overline{\widehat{v}}^{\ell,0} \leftarrow \overline{\widehat{v}}^{\ell-1}$.
    ▷ PIECEWISE LINEAR RELAXATION REFINEMENT
    Refine the piecewise linear approx. of all $\gamma \in \Gamma$ to obtain $\mathcal{T}_\gamma^\ell$ by longest-edge
    bisection of the simplex corresponding to $\big((\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)\big)_{t=1,\dots,T}$.
    Determine an outer approximation $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ of $(\boldsymbol{P})$ by using a MILP model and
    appropriate shifts for the piecewise linear approximations.
    ▷ INNER LOOP
    Solve subproblem $(\boldsymbol{\widehat{P}_1^{R,\ell i}(x_0^{\ell i}, \mathfrak{Q}_2^{\ell i})})$. Store the optimal point $(\widehat{z}_1^{\ell i}, \widehat{x}_1^{\ell i}, \widehat{y}_1^{\ell i})$.
    while $\overline{\widehat{v}}^{\ell i} - \underline{\widehat{v}}^{\ell i} > \widehat{\varepsilon}$ do
        ▷ FORWARD PASS
        for stages $t = 2, ..., T$ do
            Solve subproblem $(\boldsymbol{\widehat{P}_t^{R,\ell i}(\widehat{x}_{t-1}^{\ell i}, \mathfrak{Q}_{t+1}^{\ell i})})$ satisfying (A3). Store the optimal
            point $(\widehat{z}_t^{\ell i}, \widehat{x}_t^{\ell i}, \widehat{y}_t^{\ell i})$.
            $\overline{\widehat{v}}^{\ell,i} = \min\left\{\overline{\widehat{v}}^{\ell,i-1}, \sum_{t=1}^T \left(\widehat{f}_t(\widehat{x}_t^{\ell i}, \widehat{y}_t^{\ell i}) + \sigma_t \|\widehat{x}_{t-1}^{\ell i} - \widehat{z}_t^{\ell i}\|\right)\right\}$.
        end for
        ▷ BINARY APPROXIMATION REFINEMENT
        if Forward Pass solution in $i$ equals that in $i - 1$ then
            Set $K_{tj} \leftarrow K_{tj} + 1$ for all $t$ and $j$. Set $\beta_{tj} = U_j \left(\sum_{k=1}^{K_{tj}} 2^{k-1}\right)^{-1}$.
        end if
        ▷ BACKWARD PASS
        for stages $t = T, ..., 2$ do
            Determine the best binary approx. $\widehat{x}_{\mathbb{B},t-1}^i = B_{t-1}\lambda_{t-1}^i$ of the state $\widehat{x}_{t-1}^{\ell i}$.
            Solve subproblem $(\boldsymbol{D}_{\mathbb{B}t}^{\boldsymbol{\ell i}}(\boldsymbol{\lambda}_{t-1}^{i}, \boldsymbol{\mathfrak{Q}}_{t+1}^{\ell,i+1}))$. Store the optimal multiplier $\pi_t^{\ell i}$ and
             the corresponding optimal value $c_t^{\ell i}$ of the Lagrangian dual function.
            Construct the cut $\phi_{\mathbb{B}t}^{\ell i}(\lambda_{t-1}) = c_t^{\ell i} + (\pi_t^{\ell i})^\top \lambda_{t-1}$ in the binary state space.
            Model the optimal value function $\phi_t^{\ell i}$ of projecting $\phi_{\mathbb{B}t}^{\ell i}$ to the original space
             by MILP constraints using the KKT conditions.
            Set $\mathfrak{Q}_t^{\ell,i+1}(x_t) = \max\{\mathfrak{Q}_t^{\ell i}(x_t), \phi_t^{\ell i}(x_t)\}$.
        end for
        ▷ FIRST STAGE UPDATE
        Solve subproblem $(\boldsymbol{\widehat{P}_1^{R,\ell i}(x_0^{\ell i}, \mathfrak{Q}_2^{\ell,i+1})})$. Store the optimal point $(\widehat{z}_1^{\ell i}, \widehat{x}_1^{\ell i}, \widehat{y}_1^{\ell i})$.
        Update $\underline{\widehat{v}}^{\ell,i}$ to the optimal value $\underline{\widehat{Q}}_1^{\ell i}(0, \mathfrak{Q}_2^{\ell,i+1})$.
        $i \leftarrow i + 1$.
    end while
    Set $\underline{\widehat{v}}^\ell \leftarrow \underline{\widehat{v}}^{\ell i}, \overline{\widehat{v}}^\ell \leftarrow \overline{\widehat{v}}^{\ell i}, \mathfrak{Q}_t^\ell(\cdot) \leftarrow \mathfrak{Q}_t^{\ell i}(\cdot)$ for all $t = 2, ..., T$.
    ▷ OUTER LOOP PROBLEMS
    for stages $t = 1, \dots, T$ do
        Solve subproblem $(\boldsymbol{P_t^\ell(x_{t-1}^\ell, \mathfrak{Q}_{t+1}^\ell)})$. Store the optimal solution $(z_t^\ell, x_t^\ell, y_t^\ell)$.
    end for
    $\overline{v}^\ell = \min\left\{\overline{v}^{\ell-1}, \sum_{t=1}^T f_t(x_t^\ell, y_t^\ell)\right\}$.
end while
Output: $\varepsilon$-optimal solution $\big((z_t^\ell, x_t^\ell, y_t^\ell)\big)_{t=1,\dots,T}$ of $(\boldsymbol{P})$.
```



ε-optimal solution of $(\boldsymbol{P})$
yes
Outer Loop Problem Solved?
no
Problem $(\boldsymbol{P})$ satisfying $\mathbf{(A1)}$ and $\mathbf{(A2)}$
Piecewise Linear Relaxations Refinement
Outer Loop
(solving $(\boldsymbol{P})$ iteratively using cuts from inner loop, refining MILP $(\widehat{\boldsymbol{P}})$)
Solve Outer Loop Problems
Outer Approximation $(\widehat{\boldsymbol{P}})$
yes
Inner Loop Problem Solved?
no
Forward Pass
Inner Loop
(solving $(\widehat{\boldsymbol{P}})$ iteratively, generating cuts for Q_t)
Backward Pass
Binary Expansion Decision
no
yes
Binary Approx. Refinement

Fig. 1 Conceptual overview of NC-NBD

state variables with binary ones, construct cuts in the binary space and then project those cuts back to the original space. As we show, these projections can be modeled by mixed-integer linear constraints in the original space. By careful construction, these cuts remain valid even if the binary approximation is refined in later iterations.

In this way, we circumvent solving an augmented Lagrangian dual, which may be even more expensive than solving the classical Lagrangian dual, as with the additional nonlinear term in the objective, the primal problems lose their decomposability. In return, we require more (binary) variables and constraints in the Lagrangian duals and for an MILP representation of our cuts than the approach in [1].

In principle, the MILPs as they occur in the inner loop could also be solved by using SDDiP with a static binary approximation of the state variables [55]. As discussed in

Sect. 1, this approach has some properties which prevent an efficient integration into our algorithmic framework, though.

As we show in the next section, for a sufficiently fine binary approximation, the obtained cuts in the NC-NBD provide a sufficiently good approximation at the trial solutions $\widehat{x}_{t-1}^{\ell i}$. Additionally, the cut approximations $\mathfrak{Q}_t^{\ell}(\cdot)$ are generated in such a way that they are Lipschitz continuous. This is sufficient to ensure convergence to a globally optimal solution of $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$.

At the end of the backward pass, a lower bound $\widehat{\underline{v}}^{\ell i}$ is determined. If $\widehat{\overline{v}}^{\ell i}$ and $\widehat{\underline{v}}^{\ell i}$ are sufficiently close to each other, an approximate globally minimal point $\big((\widehat{z}_t^{\ell}, \widehat{x}_t^{\ell}, \widehat{y}_t^{\ell})\big)_{t=1,\dots,T}$ of $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ has been identified and the inner loop is left. Otherwise, further cuts have to be constructed or the binary approximation has to be refined. We discuss this decision in more detail in Sect. 3.3.6.

Once the inner loop is left, subproblems $(\boldsymbol{P_t(x_{t-1}, \mathfrak{Q}_{t+1}^{\ell})})$ are solved to determine trial points x_{t-1}^{ℓ} and an upper bound $\overline{v}^{\ell}$ to v for the original problem $(\boldsymbol{P})$. If this upper bound is sufficiently close to $\widehat{\underline{v}}^{\ell}$, the solution $\big((z_t^{\ell}, x_t^{\ell}, y_t^{\ell})\big)_{t=1,\dots,T}$ is approximately optimal for problem $(\boldsymbol{P})$. If not, the MILP relaxation $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell+1}})$ is created by refining $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ in the neighborhood of $\big((\widehat{z}_t^{\ell}, \widehat{x}_t^{\ell}, \widehat{y}_t^{\ell})\big)_{t=1,\dots,T}$ and a new inner loop is started.

As for the inner loop, it is crucial that with these refinements in the outer loop all previously generated cuts remain valid. Otherwise, the cut approximation $\mathfrak{Q}_{t+1}^{\ell}(\cdot)$ would have to be built from scratch, counteracting the idea of a dynamic solution framework. In the following subsections, we show how such persistent validity can be achieved by careful design. Note that, even though we make use of the same regularization idea, our framework with nested loops and dynamic refinements also differs from the method presented in [52].

We explain the different steps of NC-NBD in more detail in the following subsections, before we discuss convergence results in Sect. 4. As long as the index ℓ is not needed for the discussions of the inner loop, we omit it for notational convenience. Moreover, we note that several of the considered subproblems require the introduction of additional decision variables, *e.g.*, for piecewise linear approximation or cut projection. For reasons of clarity and comprehensibility, by the terms *optimal point* or *optimal solution* we refer to the projection of their actual optimal points to the space $X_{t-1} \times X_t \times Y_t$, which we are interested in.

3.2 Piecewise linear relaxations

In the outer loop of NC-NBD, all nonlinear functions $\gamma \in \Gamma$ in problem $(\boldsymbol{P})$ are approximated by some piecewise linear functions. This is achieved by determining a triangulation of their domain, which in our box-constrained setting is always possible. Then, the piecewise linear functions can be defined on the simplices of this triangulation using the function values of γ at their vertices. For a thorough discussion and state-of-the-art approaches to construct piecewise linear approximations and triangulations, see [18,39,40].

The piecewise linear approximations can then be reformulated as mixed-integer linear constraints using auxiliary continuous and binary variables. In the literature, several modeling techniques have been proposed, such as the convex combination model, the incremental model and some logarithmic variants [4,18,38,51]. Later on, we draw on refinement and convergence ideas from [9], which work for several of these models, such as the generalized incremental model [9] or the disaggregated logarithmic convex combination model [51].

By shifting the approximations appropriately, it can be ensured that the obtained MILP $(\widehat{\boldsymbol{P}}^{\boldsymbol{j}})$ is indeed a relaxation of the original problem $(\boldsymbol{P})$ [18]. Alternatively, one can construct piecewise linear underestimators and overestimators, yielding tubes for nonlinear equations [25].

Applying the piecewise linear approximations to problem $(\boldsymbol{P})$, we obtain the MILP outer approximation with copy constraints

$$(\widehat{\boldsymbol{P}}) \quad \widehat{v} := \min_{\substack{x_1,\dots,x_T,y_1,\dots,y_T \\ z_1,\dots,z_T}} \quad \sum_{t=1}^{T} \widehat{f}_t(x_t, y_t)$$

$$\text{s.t.} \quad (z_t, x_t, y_t) \in \widehat{M}_t \qquad \forall t = 1, \dots, T$$

$$z_t = x_{t-1} \qquad \forall t = 1, \dots, T.$$

For reasons of clarity, we denote the piecewise linear relaxations of $f_t(\cdot)$, $g_t(\cdot)$ and $h_t(\cdot)$ by $\widehat{f}_t(\cdot)$, $\widehat{g}_t(\cdot)$ and $\widehat{h}_t(\cdot)$, although they are modeled using auxiliary constraints and variables. The set $\widehat{M}_t$ is defined by replacing the functions $g_t(\cdot)$ and $h_t(\cdot)$ in M_t or $M_t(x_{t-1})$, respectively, with $\widehat{g}_t(\cdot)$ and $\widehat{h}_t(\cdot)$.

The dynamic programming equations for $t = 1, \dots, T$ are given by

$$(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{t-1}})) \quad \widehat{Q}_t(x_{t-1}) := \min_{z_t, x_t, y_t} \quad \widehat{f}_t(x_t, y_t) + \widehat{Q}_{t+1}(x_t)$$

$$\text{s.t.} \quad (z_t, x_t, y_t) \in \widehat{M}_t$$

$$z_t = x_{t-1}.$$

For the MILP subproblems $(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}(\cdot))$, we obtain the following properties.

Lemma 3.1 *Under assumption (A2), subproblem* $(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}(\cdot))$ *has complete recourse and the value function* $\widehat{Q}_t(\cdot)$ *is l.sc. for all* $t = 1, \dots, T$.

The complete recourse follows from the complete recourse of $(\boldsymbol{P}_{\boldsymbol{t}}(\cdot))$ by construction. The l.sc. then follows from Theorem 3.1 in [31].

3.3 The inner loop

In the inner loop of NC-NBD, the MILP subproblems $(\widehat{\boldsymbol{P}}_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{t-1}}))$ are considered. As stated before, we omit the index ℓ for its discussion.

The copy constraints are crucial for all problems solved in the inner loop. In the forward pass, to ensure Lipschitz continuity, we consider regularized subproblems. The regularization is based on relaxing and penalizing the copy constraints. In the backward pass, to generate cuts, a special Lagrangian dual subproblem is solved based on dualizing the copy constraints. This is effective, since combined with a binary expansion of the state variables, the copy constraints yield a local convexification [55].

3.3.1 Regularization

Lipschitz continuity of the value functions is difficult to ensure in the general nonconvex case. However, as shown recently in [52], for l.sc. value functions, it is possible to determine some underestimating Lipschitz continuous function by enhancing the original subproblem with an appropriate penalty function ψ_t. In contrast to the more general regularization approach in [52], we require only so-called *sharp* penalty functions $\psi_t(x_{t-1}) = \|x_{t-1}\|$ to regularize the subproblems ($\boldsymbol{\widehat{P}_t(x_{t-1})}$), for some norm $\|\cdot\|$.

Definition 3.2 (Regularized subproblem and value function) Let $\sigma_t > 0$ for $t = 2, \dots T$, $\sigma_1 = 0$ and define

$$(\boldsymbol{\widehat{P}_t^R(x_{t-1})}) \quad \widehat{Q}_t^R(x_{t-1}) := \min_{z_t, x_t, y_t} \quad \widehat{f}_t(x_t, y_t) + \sigma_t \|x_{t-1} - z_t\| + \widehat{Q}_{t+1}^R(x_t)$$
$$\text{s.t.} \quad (z_t, x_t, y_t) \in \widehat{M}_t.$$

($\boldsymbol{\widehat{P}_t^R}$) is called *regularized subproblem* and $\widehat{Q}_t^R(\cdot)$ *regularized value function.*

By recursion, this approach yields the regularized optimal value $\widehat{v}^R := \widehat{Q}_1^R(x_0)$ for the first stage. Lemma 3.1 implies that under assumption **(A2)**, the function $\widehat{Q}_t(\cdot)$ is l.sc. Then, the regularized value function $\widehat{Q}_t^R(\cdot)$ has the following properties.

Lemma 3.3 (Proposition 2 in [52]) *For all $t = 1, \dots, T$ we have:*

(a) $\widehat{Q}_t^R(x_{t-1}) \leq \widehat{Q}_t(x_{t-1})$ *for all $x_{t-1} \in X_{t-1}$,*
(b) *Under assumptions **(A1)** and **(A2)**, the regularized value function $\widehat{Q}_t^R(\cdot)$ is Lipschitz continuous on X_{t-1}.*

As also stated in [52], using sharp penalty functions as in Definition 3.2, the penalization is exact for sufficiently large (but finite) $\sigma_t > 0$. For such σ_t, the problems ($\boldsymbol{\widehat{P}}$) and ($\boldsymbol{\widehat{P}^R}$) have the same optimal points and $\widehat{v}^R = \widehat{v}$. This result goes back to [14], in which augmented Lagrangian problems are analyzed for MILPs. It is shown that using sharp penalty functions and a sufficiently large augmenting parameter, strong duality holds. As this result holds for any value of the dual multipliers, it is also valid for the regularized subproblems.

Lemma 3.4 (Proposition 8 in [14]) *Using sharp penalty functions ψ_t, there exist some $\bar{\sigma}_t > 0$ such that the penalty reformulation in ($\boldsymbol{\widehat{P}_t^R(x_{t-1})}$) is exact for all $\sigma_t > \bar{\sigma}_t$.*

Lemma 3.4 indicates that using the regularized subproblems within our decomposition method NC-NBD, we obtain convergence to $\widehat{v}$ in the inner loop. To exploit this, we take the following assumption:

(A3). All $\sigma_t > 0$ are chosen sufficiently large such that Lemma 3.4 is satisfied.

If **(A3)** is not satisfied, σ_t has to be increased gradually in the course of the NC-NBD method to ensure convergence.

3.3.2 Forward pass

In the forward pass of the inner loop we solve approximations of the regularized subproblems ($\boldsymbol{\widehat{P}_t^R(x_{t-1})}$).

For iteration i, the stage-t forward pass problem is defined as follows

$$\begin{aligned}
&(\boldsymbol{\widehat{P}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)})\\
&\underline{\widehat{Q}}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i) := \min_{z_t, x_t, y_t} \quad \widehat{f}_t(x_t, y_t) + \sigma_t \|\widehat{x}_{t-1}^i - z_t\| + \mathfrak{Q}_{t+1}^i(x_t)\\
&\qquad\qquad\qquad\qquad\quad \text{s.t.} \quad (z_t, x_t, y_t) \in \widehat{M}_t,
\end{aligned}$$

for the trial state variable $\widehat{x}_{t-1}^i$, with $\widehat{x}_0^i \equiv 0$. Function $\mathfrak{Q}_{t+1}^i(\cdot)$, in some sense, approximates the value functions $\underline{\widehat{Q}}_{t+1}^{R,i}(\cdot, \mathfrak{Q}_{t+2}^i)$ and $\underline{\widehat{Q}}_{t+1}^i(\cdot, \mathfrak{Q}_{t+2}^i)$. This approximation is constructed in the backward pass, see Sect. 3.3.4. As those value functions are non-convex, the cut approximation $\mathfrak{Q}_{t+1}^i(\cdot)$ is required to be nonlinear and non-convex. However, as we show later, it can be expressed with mixed-integer linear constraints by lifting the problems to a higher dimension. Therefore, in addition to x_t, y_t and z_t, the forward pass problem contains further decision variables, which are hidden in $\mathfrak{Q}_{t+1}^i(\cdot)$ and the piecewise linear relaxations $\widehat{f}_t$, $\widehat{g}_t$ and $\widehat{h}_t$.

Note that expressing $\mathfrak{Q}_{t+1}^i(\cdot)$ by mixed-integer linear constraints with bounded integer variables, the same reasoning as in Lemma 3.1 can be applied to show that $\underline{\widehat{Q}}_t^i(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)$ is l.sc. and therefore, $\underline{\widehat{Q}}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)$ is Lipschitz continuous.

Even with a mixed-integer linear representation of $\mathfrak{Q}_{t+1}^i(\cdot)$, subproblem ($\boldsymbol{\widehat{P}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)}$) is a MINLP due to the regularization. For $\|\cdot\|_1$ or $\|\cdot\|_\infty$, it can be modeled by MILP constraints using standard reformulation techniques for absolute values, though.

The optimal point $(\widehat{z}_t^i, \widehat{x}_t^i, \widehat{y}_t^i)$ of each subproblem ($\boldsymbol{\widehat{P}_t^{R,i}(\widehat{x}_{t-1}^i)}$) is stored and $\widehat{x}_t^i$ is passed to the following stage. Since $\left((\widehat{z}_t^i, \widehat{x}_t^i, \widehat{y}_t^i)\right)_{t=1,\dots,T}$ satisfies all constraints of ($\boldsymbol{\widehat{P}^R}$), after all stages have been considered, an upper bound $\overline{\widehat{v}}$ on the optimal value $\widehat{v}^R$ of the regularized problem can be determined by

$$\overline{\widehat{v}}^i = \min\left\{\overline{\widehat{v}}^{i-1}, \sum_{t=1}^T \left(\widehat{f}_t(\widehat{x}_t^i, \widehat{y}_t^i) + \sigma_t \|\widehat{x}_{t-1}^i - \widehat{z}_t^i\|\right)\right\}.$$

With assumption **(A3)** and Lemma 3.4, this is also an upper bound to $\widehat{v}$.

3.3.3 Backward pass–Part 1: binary approximation

The aim of the backward pass of an inner loop iteration i is twofold: Firstly, a lower bound $\underline{\widehat{v}}^i$ on $\widehat{v}$ is determined. Secondly, cuts for $Q_t(\cdot)$ are derived to improve and update the current approximation $\mathfrak{Q}_t^i(\cdot)$.

As mentioned before, we use a dynamically refined binary approximation of the state variables and then apply cutting-plane techniques from the SDDiP algorithm [55]. This approximation is based on static binary expansion [21].

Binary expansion can be applied component-wise to some vector x_t. Some integer component $x_{tj} \in \left\{0, ..., U_j\right\}$ can be exactly and uniquely expressed as

$$x_{tj} = \sum_{k=1}^{K_{tj}} 2^{k-1} \lambda_{tkj}$$

with variables $\lambda_{tkj} \in \{0, 1\}$ and $K_{tj} = \lfloor \log_2 U_j \rfloor + 1$. Some continuous component $x_{tj} \in [0, U_j]$ can be expressed by discretizing the interval with precision $\beta_{tj} \in (0, 1)$. We then have

$$x_{tj} = \sum_{k=1}^{K_{tj}} 2^{k-1} \beta_{tj} \lambda_{tkj} + r_{tj}$$

with $K_{tj} = \lfloor \log_2 \left(\frac{U_j}{\beta_{tj}} \right) \rfloor + 1$ and some error $r_{tj} \in \left[-\frac{\beta_{tj}}{2}, \frac{\beta_{tj}}{2} \right]$.

For vector x_t, this yields $K_t = \sum_{j=1}^{n_t} K_{tj}$ number of binary variables. Defining an $(n_t \times K_t)$-matrix B_t containing all the coefficients of the binary expansion and collecting all binary variables in one large vector $\lambda_t \in \mathbb{B}^{K_t}$, the binary expansion then can be written compactly as $x_t = B_t \lambda_t + r_t$.

Based on this definition, to generate cuts, for each stage t and iteration i, a binary approximation of $\widehat{x}^i_{t-1}$ is used, *i.e.*, it is replaced by $B_{t-1} \lambda^i_{t-1}$. Note that the approximation is not necessarily exact for continuous components of $\widehat{x}^i_{t-1}$. Therefore, the cuts are not necessarily constructed at the *trial point* $\widehat{x}^i_{t-1}$ but at the deviating *anchor point* $\widehat{x}^i_{\mathbb{B},t-1} := B_{t-1} \lambda^i_{t-1}$.

In the backward pass, we start from the following subproblem, where due to the binary approximation of the state variables, we also adapt the copy constraint to $\lambda^i_{t-1} = \mathfrak{z}_t$ with variables $\mathfrak{z}_t \in [0, 1]^{K_{t-1}}$.

$$\left(\boldsymbol{\widehat{P}^i_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1})}\right) \quad \underline{\widehat{Q}}^i_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1}) := \begin{array}{rl} \min\limits_{\substack{x_t, y_t, \\ \mathfrak{z}_t, z_t}} & \widehat{f}_t(x_t, y_t) + \mathfrak{Q}^{i+1}_{t+1}(x_t) \\ \text{s.t.} & (z_t, x_t, y_t) \in \widehat{M}_t \\ & z_t = B_{t-1} \mathfrak{z}_t \\ & \mathfrak{z}_t \in [0, 1]^{K_{t-1}} \\ & \mathfrak{z}_t = \lambda^i_{t-1}. \end{array}$$

Remark 3.5 Subproblem $\left(\boldsymbol{\widehat{P}^i_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1})}\right)$ is equivalent to subproblem $\left(\boldsymbol{\widehat{P}^i_t(\widehat{x}^i_{\mathbb{B},t-1}, \mathfrak{Q}^{i+1}_{t+1})}\right)$ because $z_t = B_{t-1} \mathfrak{z}_t = B_{t-1} \lambda^i_{t-1} = \widehat{x}^i_{\mathbb{B},t-1}$.

Asymptotically, *i.e.*, for an infinitely fine binary approximation, the anchor point converges to the actual trial point.

Lemma 3.6 *We have* $\lim_{\beta_{t-1}\to 0} \widehat{x}^i_{\mathbb{B}t-1} = \widehat{x}^i_{t-1}$.

With Lemma 3.6, asymptotically, the cuts are constructed at $\widehat{x}^i_{t-1}$. While this is not directly useful in practice, since it requires an infinite number of binary variables, it also implies that for componentwise sufficiently small $\beta_{t-1} \in (0, 1)$, the cuts are constructed very close to $\widehat{x}^i_{t-1}$. As NC-NBD constructs Lipschitz continuous cuts, this guarantees a sufficiently good approximation of the value function at $\widehat{x}^i_{t-1}$, as we show in Sect. 4.

Importantly, in our framework the binary approximation is only applied temporarily to derive cuts, while the state variables x_{t-1} in the forward pass remain continuous. In other words, the anchor points determine where cuts can be *constructed*, but do not limit where they can be *evaluated*. This is a crucial difference to applying a *static* binary expansion, as suggested in the original SDDiP work to solve MILPs with continuous state variables [55].

Moreover, let us emphasize again that applying such static approximation is not appropriate in our inner loop, as the obtained lower bounds are not guaranteed to be valid for $\widehat{v}$ or v. Similarly, the obtained cuts are not guaranteed to be valid for $\widehat{Q}_t(\cdot)$ or $Q_t(\cdot)$, and therefore cannot be re-used within the outer loop. Our proposed inner loop method does not share these issues. We follow a *dynamic* approach where the binary precision is dynamically refined if required and, as we show later, all cuts remain valid with later refinements.

3.3.4 Backward pass–Part 2: cut generation

As proposed in [55], the copy constraint is dualized to generate cuts. Applied to our context, the following Lagrangian dual subproblem has to be solved

$$(\boldsymbol{D}^i_{\mathbb{B}t}(\boldsymbol{\lambda}^i_{t-1}, \boldsymbol{\mathfrak{Q}}^{i+1}_{t+1})) \quad \max_{\|\pi_t\|_* \le l_t} \quad \mathcal{L}^i_{\mathbb{B}t}(\pi_t, \mathfrak{Q}^{i+1}_{t+1}) + \pi_t^\top \lambda^i_{t-1},$$

where $\mathcal{L}^i_{\mathbb{B}t}(\cdot)$ denotes the Lagrangian function for π_t defined by

$$\begin{aligned} \mathcal{L}^i_{\mathbb{B}t}(\pi^i_t, \mathfrak{Q}^{i+1}_{t+1}) := \min_{x_t, y_t, \mathfrak{z}_t, z_t} \quad & \widehat{f}_t(x_t, y_t) + \mathfrak{Q}^{i+1}_{t+1}(x_t) - \pi_t^\top \mathfrak{z}_t \\ \text{s.t.} \quad & (z_t, x_t, y_t) \in \widehat{M}_t \\ & z_t = B_{t-1}\mathfrak{z}_t \\ & \mathfrak{z}_t \in [0, 1]^{K_{t-1}} \end{aligned}$$

and $\|\cdot\|_*$ denotes the dual norm to the norm used in the regularized forward pass problems $(\widehat{\boldsymbol{P}}^{R,i}_t(\widehat{\boldsymbol{x}}^i_{t-1}, \boldsymbol{\mathfrak{Q}}^i_{t+1}))$.

A linear (optimality) cut in binary space $\{0, 1\}^{K_{t-1}}$ is then given by

$$\phi_{\mathbb{B}t}(\lambda_{t-1}) := \underbrace{\mathcal{L}^i_{\mathbb{B}t}(\pi^i_t, \mathfrak{Q}^{i+1}_{t+1})}_{=:c^i_t} + (\pi^i_t)^\top \lambda_{t-1}, \tag{1}$$

where π_t^i is an optimal solution of the Lagrangain dual subproblem $(\boldsymbol{D}^{i}_{\mathbb{B}t}(\boldsymbol{\lambda}^{i}_{t-1}, \boldsymbol{\mathfrak{Q}}^{i+1}_{t+1}))$. Those Lagrangian cuts are introduced in [55] and identified to be finite, valid and tight in the SDDiP setting. In our setting, we obtain the following validity result.

Lemma 3.7 *Let $\widehat{Q}_{\mathbb{B}t}(\cdot)$ denote the MILP value function of stage t with additional binary approximations. Then,*

(a) *for all $\lambda_{t-1} \in [0,1]^{K_{t-1}}$*

$$\widehat{Q}_{\mathbb{B}t}(\lambda_{t-1}) \geq \phi_{\mathbb{B}t}(\lambda_{t-1}),$$

(b) *for all x_{t-1}*

$$\widehat{Q}_t(x_{t-1}) \geq \phi_{\mathbb{B}t}(\lambda_{t-1})$$

for any $\lambda_{t-1} \in [0,1]^{K_{t-1}}$, such that $x_{t-1} = B_{t-1}\lambda_{t-1}$.

Lemma 3.7 a) follows directly from the validity proof for the SDDiP cuts, which does also hold for $\lambda_{t-1} \in [0,1]^{K_{t-1}}$ instead of $\lambda_{t-1} \in \{0,1\}^{K_{t-1}}$ (see Theorem 3 in [55]). Part b) then follows using similar arguments as in Remark 3.5. Hence, $\phi_{\mathbb{B}t}$ is, in fact, a valid cut in $[0,1]^{K_{t-1}}$. This enables us to obtain valid under-approximations also for those points, which are not exactly approximated by the current binary expansion. As it refers to an outer approximation, $\widehat{Q}_t(\cdot)$ underestimates the original MINLP value function $Q_t(\cdot)$. Thus, the obtained cuts are valid for $Q_t(\cdot)$ as well.

Contrary to [55], but following [52], we bound the dual variable π_t in the Lagrangian dual subproblem. Therefore, tightness for $\underline{\widehat{Q}}^{i}_{\mathbb{B}t}(\cdot, \mathfrak{Q}^{i+1}_{t+1})$ is not guaranteed. However, the cuts are at least guaranteed to overestimate the value function $\underline{\widehat{Q}}^{R,i}_{\mathbb{B}t}(\cdot, \mathfrak{Q}^{i+1}_{t+1})$ at λ^i_{t-1}. This value function is obtained by regularizing $\underline{\widehat{Q}}^{i}_{\mathbb{B}t}(\cdot, \mathfrak{Q}^{i+1}_{t+1})$ in the binary space using the same norm as in the forward pass problem. By careful choice of the regularization factor, then, also the regularized value function $\underline{\widehat{Q}}^{R,i}_{t}(\cdot, \mathfrak{Q}^{i+1}_{t+1})$ in the original space is overestimated at $x^i_{\mathbb{B},t-1}$. This result is formalized in the following lemma.

Lemma 3.8 *Assume that we use $\|\cdot\|_1$ for regularization and its dual norm $\|\cdot\|_\infty$ for bounding the dual multipliers. Then, as long as $l_t \geq \sigma_t \|B_{t-1}\|$, where the latter denotes the induced matrix norm of B_{t-1}, we have*

$$\phi_{\mathbb{B}t}(\lambda^i_{t-1}) \geq \underline{\widehat{Q}}^{R,i}_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1}) \geq \underline{\widehat{Q}}^{R,i}_{t}(x^i_{\mathbb{B},t-1}, \mathfrak{Q}^{i+1}_{t+1}).$$

Proof See Appendix A. □

Remark 3.9 The induced matrix norm $\|B_{t-1}\|$ depends on the chosen precision of the binary approximation. It can be bounded from above independent of the precision, *e.g.*, $\|B_{t-1}\|_1 \leq U_{t-1,\max}$ with $U_{t-1,\max}$ the largest component of the upper bounds in X_{t-1}.

3.3.5 Backward pass–Part 3: cut projection

Solving the forward pass problems ($\boldsymbol{\widehat{P}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)}$) and the backward pass dual problems ($\boldsymbol{D_{\mathbb{B}t}^i(\lambda_{t-1}^i, \mathfrak{Q}_{t+1}^{i+1})}$) requires expressing the cut approximation $\mathfrak{Q}_{t+1}^i(\cdot)$ in the original state variables x_t. Recall that the computed cut $\phi_{\mathbb{B},t+1}(\cdot)$ is a function of $[0,1]^{K_t}$.

According to Lemma 3.7 a), the obtained cuts $\phi_{\mathbb{B},t+1}(\cdot)$ are not only valid for all binary points, but for all values in $[0,1]^{K_t}$. Allowing for $\lambda_t \in [0,1]^{K_t}$ in the binary approximation, there exist infinitely many combinations of λ_t to exactly describe some point $x_t \in X_t$, though. Therefore, following from Lemma 3.7 b), one cut in binary space entails infinitely many underestimators of $Q_{t+1}(\cdot)$ at x_t in the original space X_t. Including infinitely many inequalities in $\mathfrak{Q}_{t+1}(\cdot)$ is computationally infeasible. Instead, we consider the pointwise maximum of the projection of the cuts to X_t. That way, only the best underestimation for each point x_t is taken into account. In doing so, we obtain a nonlinear, *i.e.*, piecewise linear, cut in the original state space. For simplicity, in the following, by *cut projection* we always mean the pointwise maximum of the actual projection.

The projection of some cut $\phi_{\mathbb{B},t+1}(\cdot)$ to X_t can be described as the value function

$$\phi_{t+1}(x_t) := \max_{\lambda_t} \left\{ c_{t+1} + (\pi_{t+1})^\top \lambda_t : B_t \lambda_t = x_t, \lambda_t \leq e, \lambda_t \geq 0 \right\} \tag{2}$$

of a linear program where e denotes a vector of ones of dimension K_t. The dual problem to (2) yields

$$\phi_{t+1}^D(x_t) := \min_{\eta_t, \mu_t} \left\{ c_{t+1} + x_t^\top \eta_t + e^\top \mu_t \ : \ B_t^\top \eta_t + I \mu_t \geq \pi_{t+1}, \mu_t \geq 0 \right\}. \tag{3}$$

Note that the dual feasible region does not depend on x_t and has a finite number of extreme points. Therefore, the cut projection is piecewise linear and concave.

As problem (2) is feasible and bounded for any $x_t \in X_t$, this also holds for the dual problem (3). Therefore, in a dual optimal solution, η_t and μ_t are bounded. Note that this bound may change with the binary approximation precision β_t, though, and that, if we would generate tight cuts for $\underline{\widehat{Q}}_{t+1}^i(\cdot, \mathfrak{Q}_{t+2}^{i+1})$, those cuts may become infinitely steep close to discontinuities. However, as we can bound π_t in the Lagrangian dual subproblem independent of β_t, see Remark 3.9, and thus construct cuts which at least overestimate the regularized value function $\underline{\widehat{Q}}_{t+1}^{R,i}(\cdot, \mathfrak{Q}_{t+2}^{i+1})$ at the anchor point $x_{\mathbb{B},t}^i$, such cases should be ruled out.

We formalize this by assuming the existence of a global bound for η_t.

(A4). There exists some $\rho_t > 0$, such that for all $t = 1, \ldots, T$, any binary precision β_t and any x_t, the optimal dual variable η_t in problem (3) can be bounded, *i.e.*, $\|\eta_t\| \leq \rho_t$.

For example, if we obtain cuts which are, in fact, tight for $\underline{\widehat{Q}}_{t+1}^{R,i}(\cdot, \mathfrak{Q}_{t+2}^{i+1})$ at $x_{\mathbb{B},t}^i$ and consider only basic solutions in the Lagrangian dual, the gradient of the cuts is bounded by σ_{t+1}. With Assumption **(A4)** it follows that the linear cuts $\phi_{\mathbb{B},t+1}(\cdot)$

derived in the binary space yield a nonlinear, but Lipschitz continuous projection $\phi_{t+1}(\cdot)$ in the original space.

To express this projection by mixed-integer linear constraints, we use the KKT conditions to problems (2) and (3). To emphasize that these conditions are considered for the projection of one specific cut r (the index denoting the r-th cut constructed), we index all occurring variables and coefficients by r.

$$
\begin{aligned}
-\pi_{t+1}^r - \nu_t^r + \mu_t^r + (B_t^r)^\top \eta_t^r &= 0 && (4)\\
B_t^r \lambda_t^r - x_t &= 0 && (5)\\
\lambda_t^r &\geq 0 && (6)\\
\lambda_t^r - e &\leq 0 && (7)\\
\nu_t^r, \mu_t^r &\geq 0 && (8)\\
-(\nu_t^r)^\top \lambda_t^r &= 0 && (9)\\
(\mu_t^r)^\top (\lambda_t^r - e) &= 0. && (10)
\end{aligned}
$$

The complementary slackness constraints (9) and (10) are nonlinear, but componentwise can be expressed linearly using a Big-$\mathcal{M}$ formulation (alternatively, SOS-1 constraints may be used):

$$
\begin{aligned}
&\lambda_{tk}^r \leq \mathcal{M}_{1k}\omega_{tk}^r, \quad \nu_{tk}^r \leq \mathcal{M}_{2k}(1-\omega_{tk}^r), \quad \omega_{tk}^r \in \{0,1\} && (11)\\
&\lambda_{tk}^r - 1 \geq \mathcal{M}_{3k}u_{tk}^r, \quad \mu_{tk}^r \leq \mathcal{M}_{4k}(1-u_{tk}^r), \quad u_{tk}^r \in \{0,1\} && (12)
\end{aligned}
$$

For all components k, we can choose $\mathcal{M}_{1k} = 1$ and $\mathcal{M}_{3k} = -1$ due to $\lambda_{tk} \in [0, 1]$. Moreover, using **(A4)**, we are able to obtain explicit choices for $\mathcal{M}_{2k}$ and $\mathcal{M}_{4k}$ as well.

Lemma 3.10 *Under **(A4)**, there exist explicit, finite bounds for ν_{tk}^r and μ_{tk}^r.*

Proof See Appendix B. □

The cut approximation $\mathfrak{Q}_{t+1}^{i+1}(\cdot)$ is then defined as the maximum of all cuts $\phi_{\mathbb{B},t+1}^r = c_{t+1}^r + (\pi_{t+1}^r)^\top \lambda_t^r$ where the variable λ_t^r satisfies the linearized KKT conditions (4)–(8) and (11)–(12) for the r-th cut. With Assumption **(A4)**, it is Lipschitz continuous.

Lemma 3.11 *The cut approximation $\mathfrak{Q}_{t+1}(\cdot)$ is Lipschitz continuous in X_t with Lipschitz constant ρ_t.*

The cut projection requires to introduce the variables $\lambda_t^r, \nu_t^r, \mu_t^r, w_t^r, u_t^r, \eta_t^r$ and constraints (4)–(8) and (11)–(12) for each cut r. In particular, each cut is associated with a variable $\lambda_t^r \in [0, 1]^{K_t^r}$ where K_t^r corresponds to the number of binary variables at the time of the cut's generation. This increases the problem size considerably, as the number of variables and constraints to be added per cut is in $\mathcal{O}\big(n_t \log\big(\frac{1}{\beta_t}\big)\big)$. In return, it ensures that cuts do not have to be generated from scratch after each refinement.

3.3.6 Stopping and refining

At the end of the backward pass, a lower bound $\underline{\widehat{v}}^i$ is determined by solving the first-stage subproblem $(\widehat{\boldsymbol{P}}_{\mathbf{1}}^{\boldsymbol{i}}(\mathbf{0}, \boldsymbol{\mathfrak{Q}}_{\mathbf{2}}^{\boldsymbol{i+1}}))$. Here, no Lagrangian dual is solved, since no cuts have to be derived. The lower bound is non-decreasing because the cut approximation is only improved.

If the updated bounds are sufficiently close to each other, *i.e.*, if

$$\overline{\widehat{v}}^i - \underline{\widehat{v}}^i \leq \widehat{\varepsilon}$$

for some predefined tolerance $\widehat{\varepsilon} > 0$, an approximately optimal point of problem $(\widehat{\boldsymbol{P}})$ has been determined. We show in the following section that this is the case after finitely many iterations i.

If the gap between the bounds does not meet the stopping criteria yet, two cases are possible: In the first case, the algorithm has not determined the best possible approximation for the given binary approximation precision, yet. New cuts have been determined in iteration i such that the lower bound $\underline{\widehat{v}}^i$ has been updated, and the forward solution will change in iteration $i + 1$ as the previous one is cut away.

In the second case, despite not meeting the stopping criterion, the forward solution does not change at the beginning of iteration $i + 1$. This case is related to the binary approximation. It can occur if the binary approximation is too coarse and therefore, for all t, the determined cuts at $\widehat{x}_{\mathbb{B}t}^i$ do not improve the approximation at $\widehat{x}_t^i$. Moreover, it can occur if in subsequent iterations the same cuts are constructed, since $\widehat{x}_{B,t-1}^i = \widehat{x}_{B,t-1}^{i+1}$. Finally, it can also occur if all possible cuts have been generated: For a fixed binary approximation, there exist only finitely many points $\widehat{x}_{\mathbb{B}t}$. If we restrict the Lagrangian dual subproblem to basic solutions, then only finitely many different cuts can be determined [55].

In the second case, at the beginning of the backward pass of iteration i, the binary approximation is refined. The refinement is computed by increasing K_{tj} by +1 for all components j and all stages t with

$$\beta_{tj} = \frac{U_j}{\sum_{k=1}^{K_{tj}} 2^{k-1}}.$$

For simplicity, we refine in Algorithm 1 all stages and components equally by +1. Note that each refinement requires the introduction of an additional vector λ_t, as described in the previous subsection.

As all previously generated cuts have been projected to the original space X_t, they remain valid and have not to be recomputed when refining the binary approximation. This is computationally important.

3.4 The outer loop

3.4.1 The outer loop problem

Once the inner loop is left, we set $\underline{\widehat{v}}^{\ell} := \underline{\widehat{v}}^{\ell i}, \overline{\widehat{v}}^{\ell} := \overline{\widehat{v}}^{\ell i}$ and $\mathfrak{Q}_t^{\ell}(\cdot) := \mathfrak{Q}_t^{\ell i}(\cdot)$ for all $t = 2, ..., T$. Note that $\overline{\widehat{v}}^{\ell}$ is not guaranteed to be a valid upper bound for v because $\widehat{v}^{\ell} \leq v$. Moreover, we set $\big((\widehat{z}_t^{\ell}, \widehat{x}_t^{\ell}, \widehat{y}_t^{\ell})\big)_{t=1,...,T} := \big((\widehat{z}_t^{\ell i}, \widehat{x}_t^{\ell i}, \widehat{y}_t^{\ell i})\big)_{t=1,...,T}$.

To approximate the optimal value v of $(\boldsymbol{P})$, we solve subproblems

$$(\boldsymbol{P_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell})}) \quad \underline{Q}_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell}) := \begin{aligned}[t] \min_{z_t, x_t, y_t} \quad & f_t(x_t, y_t) + \mathfrak{Q}_{t+1}^{\ell}(x_t) \\ \text{s.t.} \quad & (z_t, x_t, y_t) \in M_t \\ & z_t = x_{t-1}^{\ell} \end{aligned}$$

in a forward manner for $t = 1, \ldots, T$ with $x_0^{\ell} \equiv 0$ and $x_t^{\ell} := x_t$, where x_t is an optimal solution of $(\boldsymbol{P_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell})})$ for t. Here, we exploit that the cut approximation $\mathfrak{Q}_t^{\ell}(\cdot)$, constructed in the inner loop, is valid for $Q_t(\cdot)$ by design as well. By solving these subproblems, we obtain a feasible solution $\big((z_t^{\ell}, x_t^{\ell}, y_t^{\ell})\big)_{t=1,...,T}$ for $(\boldsymbol{P})$ and we can determine a valid upper bound for v as $\overline{v}^{\ell} = \min\left\{\overline{v}^{\ell-1}, \sum_{t=1}^{T} f_t(x_t^{\ell}, y_t^{\ell})\right\}$.

The subproblems $(\boldsymbol{P_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell})})$ are non-convex MINLP problems. This means that in order to solve the original non-convex problem $(\boldsymbol{P})$, easier, but still non-convex subproblems have to be solved to optimality for each stage t in each outer loop iteration ℓ. This might be a hard challenge by itself. We make the following assumption for the remainder of this article:

(A5). An oracle exists that is able to solve subproblems $(\boldsymbol{P_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell})})$ to global optimality.

In case that no such global optimization algorithm is available, one can solve appropriate inner approximations of $(\boldsymbol{P_t^{\ell}(x_{t-1}^{\ell}, \mathfrak{Q}_{t+1}^{\ell})})$, which are improved in the course of the algorithm.

If $\overline{v}^{\ell} - \underline{\widehat{v}}^{\ell} \leq \varepsilon$, then NC-NBD terminates and $\big((z_t^{\ell}, x_t^{\ell}, y_t^{\ell})\big)_{t=1,...,T}$ is an ε-optimal solution for $(\boldsymbol{P})$. Otherwise, the cut approximations $\mathfrak{Q}_{t+1}^{\ell}(\cdot)$ are not sufficiently good underestimators for the true value functions, even though they give a good approximation of $\widehat{Q}_t^{\ell}(\cdot)$. This implies that the piecewise linear relaxations have to be improved. Instead of refining them everywhere, they are refined dynamically where it is promising, *i.e.*, in a neighborhood of the approximate optimal solution $\big((\widehat{z}_t^{\ell}, \widehat{x}_t^{\ell}, \widehat{y}_t^{\ell})\big)_{t=1,...,T}$ of $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$. In refining the piecewise linear relaxations in its neighborhood, the current solution can be excluded in the next inner loop and the lower bound $\underline{\widehat{v}}^{\ell}$ improves.

Remark 3.12 Instead of $\underline{\widehat{v}}^{\ell}$, an even better lower bound for v is given by the optimal value of the first stage subproblem $(\boldsymbol{P_1^{\ell}(x_0^{\ell}, \mathfrak{Q}_2^{\ell})})$.

3.4.2 Refining the piecewise linear relaxations

The refinement consists of two steps: (1) the piecewise linear approximations are refined and (2) the corresponding MILP $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ is updated – in such a way that the new MILP $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell+1}})$ again yields a relaxation of $(\boldsymbol{P})$.

Different strategies are possible to achieve this. For a thorough overview, we refer to [18]. In the following, we make use of a specific adaptive refinement scheme for triangulations from [9] for any nonlinear function $\gamma_t \in \Gamma_t$. The given piecewise linear approximation at iteration ℓ is defined by a triangulation $\mathcal{T}$ of $X_{t-1} \times X_t \times Y_t$ (or a subspace) and the corresponding function values of γ_t. Instead of refining this triangulation everywhere now, the main idea is to only refine it in a neighborhood of $\big((\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)\big)_{t=1,\ldots,T}$. Therefore, first, the simplex in $\mathcal{T}$ containing this point is identified. It is then divided by a longest-edge bisection, yielding a refined triangulation, for which a new MILP model can be set up. As proven in [9], this refinement strategy has some favorable properties with respect to convergence, see Sect. 4.2.

It is important that the obtained relaxation $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell+1}})$ is tighter than $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ so that the corresponding value functions improve monotonically. This is required to ensure that previously generated cuts remain valid in later iterations. For concave functions, this is always satisfied using the presented refinement strategy. For other functions, *e.g.*, convex ones, a more careful determination of the relaxation is required or the MILP models for earlier relaxations have to be kept instead of being replaced. For our theoretical results, it is sufficient that such monotonically improving relaxations can always be determined.

After refining the piecewise linear relaxations, a new iteration $\ell + 1$ is started, beginning with the inner loop.

4 Convergence results

In this section, we prove the convergence of the NC-NBD algorithm. We start proving the convergence of the inner loop to an optimal solution of $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ based on some results on the binary refinements. Afterwards, we prove that the outer loop converges to an optimal solution of the original problem $(\boldsymbol{P})$.

4.1 Convergence of the inner loop

As explained in Sect. 3.3.3, within NC-NBD the cuts are not generated at the trial points $\widehat{x}_{t-1}^i$, but instead at anchor points $\widehat{x}_{\mathbb{B},t-1}^i := B_{t-1}\lambda_{t-1}^i$. This means that the generated cuts, and with that also the cut approximations $\mathfrak{Q}_t(\cdot)$, implicitly depend on the binary approximation precision β_t.

However, Lemma 3.6 implies that $\widehat{x}_{t-1}^i$ and $\widehat{x}_{\mathbb{B},t-1}^i$ should become equal asymptotically in the refinements of the binary approximations. Therefore, asymptotically, the cuts are guaranteed to overestimate $\underline{\widehat{Q}}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^{i+1})$ and, due to their Lipschitz continuity, for some sufficiently small precision, they are at least $\varepsilon_{\mathbb{B}t}$-close. This, in turn, leads to convergence of the inner loop, as we formalize and prove below.

Prior to this, let us introduce two useful ideas. Firstly, using the Lipschitz continuity results from Lemma 3.3, page 12 and Lemma 3.11, we can bound the cut approximation error in $\widehat{x}_{t-1}^i$ as follows:

Lemma 4.1 *With Assumption **(A4)**, for any iteration i and stage t it follows*

$$\mathfrak{Q}_t^{i+1}(\widehat{x}_{t-1}^i) - \underline{Q}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^{i+1}) \geq -(L_t^R + \rho_t)\|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\|.$$

Proof See Appendix C. □

Secondly, for any stage t and any fixed binary approximation, if we restrict to basic solutions in the Lagrangian duals, only finitely many different realizations of cut approximations $\mathfrak{Q}_t(\cdot)$ can be generated. Thus, after a finite number of iterations, the binary approximation is refined. Assuming that the inner loop does not terminate for $\widehat{\varepsilon} = 0$, we can then observe infinitely many such refinements. Hence, with $j \to \infty$, we also get $\beta_t \to 0$ for all $t = 1, \ldots, T$.

Now, we address convergence of the inner loop of NC-NBD to an optimal solution of $(\widehat{\boldsymbol{P}})$. First, we provide a preliminary result, which can be proven by backward induction using Lemmas 3.11 and 4.1.

Lemma 4.2 *Suppose that the inner loop does not terminate for $\widehat{\varepsilon} = 0$. Then, the infinite sequence of forward pass trial solutions $(\widehat{x}^i)_{i\in\mathbb{N}}$ possesses some cluster point $\widehat{x}^*$ with a corresponding convergent subsequence $(\widehat{x}^{i_j})_{j\in\mathbb{N}}$. This subsequence satisfies*

$$\lim_{j\to\infty} \mathfrak{Q}_t^{i_j}(\widehat{x}_{t-1}^{i_j}) \geq \widehat{Q}_t^R(\widehat{x}_{t-1}^*). \tag{13}$$

Proof See Appendix D. □

Using this result, convergence can be proven.

Theorem 4.3 *Suppose that the inner loop does not terminate for $\widehat{\varepsilon} = 0$. Then, the sequence $(\underline{\widehat{v}}^i)_{i\in\mathbb{N}}$ of lower bounds determined by the algorithm converges to $\widehat{v}$ and every cluster point of the sequence of feasible forward pass solutions generated by the inner loop is an optimal solution of $(\widehat{\boldsymbol{P}})$.*

Note that with a similar argument it can be shown that the inner loop terminates as soon as $\mathfrak{Q}_t^i(\widehat{x}_{t-1}^i) \geq \widehat{Q}_t^R(\widehat{x}_{t-1}^i)$ for all $t = 2, ..., T$.

Considering that the inner loop is integrated into an outer loop improving the MILP approximations of $(\boldsymbol{P})$, infinite convergence is not directly useful. Moreover, infinitely many binary refinements are not computationally feasible. However, we can deduce that an approximately optimal solution of $(\widehat{\boldsymbol{P}})$ is determined in a finite number of iterations.

Corollary 4.4 *For any stopping tolerance $\widehat{\varepsilon} > 0$, the inner loop stops in a finite number of iterations with an $\widehat{\varepsilon}$-optimal solution of $(\widehat{\boldsymbol{P}})$.*

4.2 Convergence of the outer loop

We start our convergence analysis of the outer loop with a feasibility result for the solutions determined in the inner loop, which follows from the convergence results in [9]. The main idea is that, as the domain is bounded for all functions $\gamma \in \Gamma$, using a longest-edge bisection, after a finite number of steps, all considered simplices become sufficiently small (since in the worst case all simplices have been refined sufficiently often).

Lemma 4.5 ([9]) *Using longest-edge bisection for the piecewise linear relaxation refinements within NC-NBD yields optimal solutions* $\big((\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)\big)_{t=1,\dots,T}$ *for* $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ *in the inner loop, which*

(a) *are approximately feasible for* $(\boldsymbol{P})$ *after a finite number of steps* ℓ,
(b) *become feasible for* $(\boldsymbol{P})$ *asymptotically in the number of refinements* ℓ.

Next we show that with decreasing the feasibility error also the deviation in the optimal value between $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ and $(\boldsymbol{P})$ can be controlled.

As a preliminary result, we obtain that for sufficiently small feasibility tolerances $\widehat{\varepsilon}_\gamma$ for all $\gamma \in \Gamma$, there exists a neighborhood of the optimal solution $\widehat{\mathrm{x}}^\ell := \big((\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)\big)_{t=1,\dots,T}$ of problem $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ containing a feasible point $\widetilde{\mathrm{x}}^\ell := \big((\widetilde{z}_t^\ell, \widetilde{x}_t^\ell, \widetilde{y}_t^\ell)\big)_{t=1,\dots,T}$ of $(\boldsymbol{P})$. This follows primarily from the continuity of all functions in $(\boldsymbol{P})$.

Lemma 4.6 *For any* $\delta > 0$, *there exists some* $\hat{\ell} \in \mathbb{N}$ *such that for all* $\ell \geq \hat{\ell}$ *there exists some feasible point* $\widetilde{\mathrm{x}}^\ell$ *of* $(\boldsymbol{P})$ *with*
$\|\widetilde{\mathrm{x}}^\ell - \widehat{\mathrm{x}}^\ell\|_2 \leq \delta$.

Applying Lemma 4.6 yields the following result with respect to the deviation in the optimal value between $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ and $(\boldsymbol{P})$.

Theorem 4.7 *There exists some* $\hat{\hat{\ell}} \in \mathbb{N}$ *such that for all* $\ell \geq \hat{\hat{\ell}}$ *we have*

$$0 \leq v - \widehat{v}^\ell \leq \varepsilon.$$

Proof The proof makes use of the Lipschitz continuity of f_t, Lemma 4.5 and Lemma 4.6 to bound $v - \widehat{v}^\ell$ from above by $L_f \delta + \sum_{t=1}^T \widehat{\varepsilon}_{f_t}$ (with $\widehat{\varepsilon}_{f_t}$ deduced from $\widehat{\varepsilon}_\gamma$ with $\gamma = f_t$). The assertion then follows with $\varepsilon := L_f \delta + \sum_{t=1}^T \widehat{\varepsilon}_{f_t}$. For a detailed proof see Appendix F. □

We obtain the central convergence result for NC-NBD:

Theorem 4.8 *NC-NBD has the following convergence properties:*

(a) *Assume that for all* ℓ *the MILP* $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ *is solved to global optimality in a finite number of steps. Then, if NC-NBD does not terminate with* $\varepsilon = 0$, *the sequence of lower bounds* $(\underline{\widehat{v}}^\ell)_{\ell \in \mathbb{N}}$ *converges to* v *and the outer loop solutions converge to an optimal solution of* $(\boldsymbol{P})$.

(b) *Let* $\varepsilon = \widehat{\varepsilon} > 0$. *Then, if NC-NBD does not terminate, it converges to an* $\widehat{\varepsilon}$*-optimal solution of* $(\boldsymbol{P})$.

(c) *For any* $\varepsilon > \widehat{\varepsilon} > 0$*, NC-NBD terminates with an* ε*-optimal solution of* $(\boldsymbol{P})$ *after a finite number of steps.*

Proof See Appendix G. □

5 Computational results

We illustrate the adequacy of using the NC-NBD to solve multistage non-convex MINLPs by applying it to moderate-sized instances of a unit commitment problem. NC-NBD is implemented in Julia-1.5.3 [7] based on the `SDDP.jl` package [12], which provides an existing implementation for SDDP. More implementation details are presented in Appendix H.

The considered unit commitment problem is formally described in detail in Appendix I. Importantly, the considered problem contains binary state variables, but also continuous state variables, such that a binary approximation of the state variables is required in the backward pass of NC-NBD. Additionally, all instances contain a nonlinear function in the objective. In the *base instances*, we consider a concave quadratic emission cost curve in the objective. In the *valve-point instances*, additionally, we consider a non-convex fuel cost curve with a sinusoidal term. In both cases, we analyze instances with 2 to 36 stages and 3 to 10 generators, resulting in 6 to 20 state variables. More details on our parameter settings and the complete test results for all instances are presented in Appendix I.

The results show that NC-NBD succeeds to solve multistage non-convex MINLPs with a moderate number of stages and state variables to (approximate) global optimality. It converges to the globally minimal point for each of the instances and, considering our 1% tolerance, terminates with valid upper and lower bounds for v.

For the base instances, we observe long computation times of several minutes compared to state-of-the-art solvers for MINLPs, which solve the problems in a few seconds, though. We address some of the reasons and possible solutions for this behavior at the end of this section. As the results for problems with a small number of state variables, but many stages look most promising, for our valve-point instance tests we focus on such instances.

For these instances, the sinusoidal terms in the objective exclude many existing general purpose solvers from application. A sample of the obtained results is presented in Table 1, for the complete ones, see Appendix I. The results show that NC-NBD is less efficient than existing solvers for problems with few stages, but becomes competitive with a larger number of stages. Especially for the instances with 36 stages, conventional solvers have difficulties closing the optimality gap while NC-NBD manages to solve the instances in reasonable time.

These results confirm that NC-NBD should be best suited for multistage problems with a large number of stages, but a relatively small number of state variables, as the obtained subproblems remain sufficiently small even for a larger number of iterations, while general purpose solvers may start to struggle due to the combination of many

Table 1 Solution times in sec. for valve-point instances with three different demand time series

T	# of Gen.	Demand	LINDOGlobal	Couenne	NC-NBD
10	3	v1	201	29	1074
16	3	v1	135	273	760
24	3	v1	–	4427	1408
24	3	v2	1562	647	771
24	4	v1	–	349	1220
24	4	v2	2603	–	4191
36	4	v1	–	–	6816
36	4	v2	2733	7501	3646
36	4	v3	–	–	3497

stages and nonlinear terms. Therefore, NC-NBD may also be useful for stochastic programs where the deterministic equivalent becomes computationally infeasible for monolith approaches. To investigate this is left for future research.

While some of the test results look promising, we still see substantial potential for improvement. This should also help to make NC-NBD more efficient and competitive for problems with a larger number of state variables. It is a known drawback of SDDiP [55], which is inherited by NC-NBD, that existing methods to solve the Lagrangian dual problems may take extremely long to converge. To some extent, this could possibly be mitigated by additionally using different cut types such as strengthened Benders cuts [55], thus, only constructing tight cuts every few iterations. Yet, developing more efficient solution methods is an important open research question.

Additionally, with each projected cut, the considered subproblems become considerably larger. While we implemented a simple cut selection scheme to reduce the subproblem size, more sophisticated approaches may be required to keep the subproblems tractable for applications with many state variables.

Finally, so far, we assume that the outer loop MINLPs are solved to global optimality **(A5)** directly. In a more efficient implementation of NC-NBD, these subproblems should be approximated as well.

6 Conclusion

We propose the non-convex nested Benders decomposition (NC-NBD) method to solve multistage non-convex MINLPs. The method is based on combining piecewise linear relaxations, regularization, binary approximation and cutting-plane techniques in a unique and dynamic way. We are able to prove that NC-NBD is guaranteed to compute an ε-optimal solution for the original MINLP in a finite number of steps. Computational results for some moderate-sized instances of a unit commitment problem demonstrate its applicability to multistage problems.

We require all constraints to be continuous and the objective function to be Lipschitz continuous, which are common assumptions in nonlinear optimization. We also assume complete recourse for the multistage problem. Moreover, the regularization factors are assumed to be sufficiently large to ensure exact penalization in the regularized subproblems. If this is not the case, the factors can be increased gradually within NC-NBD.

In contrast to previous approaches to solve multistage non-convex problems, we do not require the value functions to be monotonic in the state variables [36] and allow the state variables to enter not only the objective function, but also the constraints. The latter avoids the assumption of oracles to handle indicator functions [52].

In NC-NBD, we combine dynamic binary approximation of the state variables, cutting-plane techniques tailor-made for binary state variables and a projection from the binary to the original space. This way, we are able to obtain non-convex, piecewise linear cuts to approximate the non-convex value functions of multistage MILPs. Using some additional regularization, this is even possible if those value functions are not (Lipschitz) continuous. Together with piecewise linear relaxations, this yields non-convex underestimators for the non-convex value functions of MINLPs. All approximations are refined dynamically and, by careful design, it is ensured that all cuts remain valid even with such refinements.

The proposed method can be enhanced to solve stochastic MINLPs as well. In particular, a sampling-based approach like in SDDP could be used. In such case some adaptions have to be made with respect to the refinement criteria (forward solutions may remain unchanged for several iterations until the *right* scenarios are sampled) or the convergence checks, though.

While the presented version of NC-NBD already uses approximations which are dynamically refined, different strategies may be even more dynamic and efficient in practice. For instance, the piecewise linear relaxations could be refined dynamically in the inner loop as well.

The main drawback of NC-NBD is that the considered subproblems can become severely large, since for binary approximation, for piecewise linear approximations and for cut projection, a high number of additional variables and constraints may have to be introduced. This can become problematic, especially, if a very high binary expansion precision is required to approximate the value functions sufficiently good in the forward solutions. Recent results show that the number of binary variables K required grows linearly with the dimension n_t of state variables and logarithmically with the inverse of the binary precision β_t [55].

Therefore, in its current form, NC-NBD is best applicable to multistage MINLPs which are too large to solve in their extensive form, but for which each subproblem is sufficiently small and contains only a few nonlinear functions.

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)–445857709. The authors are grateful to the anonymous reviewers and the editor for their thoughtful and substantial remarks on earlier versions of this manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

A Proof of Lemma 3.8

Proof We start proving the second inequality. We have

$$\begin{aligned}
\underline{\widehat{Q}}_t^{R,i}(x^i_{\mathbb{B}t-1}, \mathfrak{Q}^{i+1}_{t+1}) &= \min_{z_t \in X_{t-1}} \underline{\widehat{Q}}_t^{i}(z_t, \mathfrak{Q}^{i+1}_{t+1}) + \sigma_t \|x^i_{\mathbb{B}t-1} - z_t\| \\
&= \min_{\mathfrak{z}_t \in [0,1]^{K_{t-1}}} \underline{\widehat{Q}}_t^{i}(B_{t-1}\mathfrak{z}_t, \mathfrak{Q}^{i+1}_{t+1}) + \sigma_t \|B_{t-1}(\lambda^i_{t-1} - \mathfrak{z}_t)\| \\
&\leq \min_{\mathfrak{z}_t \in [0,1]^{K_{t-1}}} \underline{\widehat{Q}}_t^{i}(B_{t-1}\mathfrak{z}_t, \mathfrak{Q}^{i+1}_{t+1}) + \sigma_t \|B_{t-1}\| \|\lambda^i_{t-1} - \mathfrak{z}_t\| \\
&= \min_{\mathfrak{z}_t \in [0,1]^{K_{t-1}}} \underline{\widehat{Q}}_{\mathbb{B}t}^{i}(\mathfrak{z}_t, \mathfrak{Q}^{i+1}_{t+1}) + \sigma_t \|B_{t-1}\| \|\lambda^i_{t-1} - \mathfrak{z}_t\| \\
&= \underline{\widehat{Q}}_{\mathbb{B}t}^{R,i}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1}).
\end{aligned}$$

The inequality follows from $\|B_{t-1}\|$ being the induced matrix norm to the used vector norm. The last equality is obtained by choosing the same norm and $\alpha_t := \sigma_t \|B_{t-1}\|$ as regularization factor in ($\boldsymbol{\widehat{P}^{R,i}_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1})}$).

To show the first inequality, we construct a dual vector componentwise by

$$\widehat{\pi}_{tj} := \begin{cases} l_t, \text{ if } \lambda^i_{t-1,j} = 1 \\ -l_t, \text{ if } \lambda^i_{t-1,j} = 0. \end{cases}$$

This vector is feasible, as it satisfies $\|\widehat{\pi}_t\|_\infty \leq l_t$. By feasibility and by definition of the Lagrangian dual ($\boldsymbol{D^i_{\mathbb{B}t}(\lambda^i_{t-1}, \mathfrak{Q}^{i+1}_{t+1})}$) it follows

$$\phi_{\mathbb{B}t}(\lambda^i_{t-1}) \geq \min_{\mathfrak{z}_t \in [0,1]^{K_{t-1}}} \underline{\widehat{Q}}_{\mathbb{B}t}^{i}(\mathfrak{z}_t, \mathfrak{Q}^{i+1}_{t+1}) + \widehat{\pi}_t^\top (\lambda^i_{t-1} - \mathfrak{z}_t). \tag{14}$$

Moreover, by construction we have $\widehat{\pi}_{tj}(\lambda^i_{t-1,j} - \mathfrak{z}_{tj}) = l_t |\lambda^i_{t-1,j} - \mathfrak{z}_{tj}|$ in each component j. Inserting this into (14) and choosing $l_t \geq \alpha_t$, we obtain the first inequality.

B Proof of Lemma 3.10

Proof Consider line (4) in the KKT conditions. By rearranging and taking norms on both sides, we obtain

$$\|\nu_t - \mu_t\| = \|\pi_{t+1} + B_t^\top \eta_t\| \leq \|\pi_{t+1}\| + \|B_t^\top \eta_t\| \leq \|\pi_{t+1}\| + \|B_t^\top\| \|\eta_t\|. \tag{15}$$

The inequalities follow with the triangle inequality and with the compatibility of the matrix norm.

We can bound all three norms in (15) individually. In the Lagrangian dual, we have $\|\pi_{t+1}\| \le \sigma_{t+1}\|B_t\|$. With Remark 3.9, we can bound $\|B_t^\top\|$ and with Assumption **(A4)**, we have $\|\eta_t\| \le \rho_t$.

For example, using the ∞-norm, we obtain

$$\|\nu_t - \mu_t\| \le \|\pi_{t+1}\|_\infty + \|B_t^\top\|_\infty \|\eta_t\|_\infty \le \sigma_{t+1}\|B_t\|_1 + \|B_t\|_1 \rho_t \le U_{t,\max}(\sigma_{t+1} + \rho_t).$$

By the equivalence of norms, we obtain similar bounds using other norms. This means that every entry of $\nu_t - \mu_t$ is bounded by this constant. Moreover, since in each component only ν_t or μ_t can be non-zero, this also implies that the components of ν_t and μ_t are bounded by this constant. □

C Proof of Lemma 4.1

Proof From the Lipschitz continuity of $\underline{Q}_t^{R,i}$ we have

$$\underline{Q}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^{i+1}) - \underline{Q}_t^{R,i}(\widehat{x}_{\mathbb{B},t-1}^i, \mathfrak{Q}_{t+1}^{i+1}) \le L_t^R \|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\|. \tag{16}$$

Analogously, using Assumption **(A4)** and Lemma 3.11, for the cut approximation we obtain

$$\mathfrak{Q}_t^{i+1}(\widehat{x}_{\mathbb{B},t-1}^i) - \mathfrak{Q}_t^{i+1}(\widehat{x}_{t-1}^i) \le \rho_t \|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\|. \tag{17}$$

Starting with (17) it follows

$$\begin{aligned}
\mathfrak{Q}_t^{i+1}(\widehat{x}_{t-1}^i) &\ge \mathfrak{Q}_t^{i+1}(\widehat{x}_{\mathbb{B},t-1}^i) - \rho_t \|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\| \\
&\ge \phi_{\mathbb{B}t}^{i+1}(\lambda_{t-1}^i) - \rho_t \|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\| \\
&\ge \widehat{\underline{Q}}_t^{R,i}(\widehat{x}_{\mathbb{B},t-1}^i, \mathfrak{Q}_{t+1}^{i+1}) - \rho_t \|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\| \\
&\ge \underline{Q}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^{i+1}) - (L_t^R + \rho_t)\|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\|.
\end{aligned}$$

The second inequality follows from the definition of $\mathfrak{Q}_t^{i+1}(\cdot)$. The third inequality follows from Lemma 3.8 and the last one is obtained using (16). □

D Proof of Lemma 4.2

Proof The structure of the proof is inspired by the proof for Lemma 4 in [1].

As the inner loop does not terminate and X is compact, there exists an infinite sequence of forward pass trial solutions $(\widehat{x}^i)_{i\in\mathbb{N}}$ with cluster points. Let $\widehat{x}^* \in X$ be such cluster point and $(\widehat{x}^{i_j})_{j\in\mathbb{N}}$ a subsequence of $(\widehat{x}^i)_{i\in\mathbb{N}}$ with $\lim_{j\to\infty} \widehat{x}^{i_j} = \widehat{x}^*$.

We show that $\lim_j \mathfrak{Q}_t^{i_j}(\widehat{x}_{t-1}^{i_j}) \ge \widehat{Q}_t^R(\widehat{x}_{t-1}^*)$ holds by backward induction.

For $t = T + 1$, this relation is trivially true, since no stage follows after T. Now, assume it already holds for stage $t + 1$, *i.e.*,

$$\lim_j \mathfrak{Q}^{i_j}_{t+1}(\widehat{x}^{i_j}_t) \geq \widehat{Q}^R_{t+1}(\widehat{x}^*_t).$$

We consider two subsequent indices in the subsequence $(\widehat{x}^{i_j})_{j\in\mathbb{N}}$.

$$\mathfrak{Q}^{i_j+1}_t(\widehat{x}^{i_j}_{t-1}) \geq \mathfrak{Q}^{i_{j-1}+1}_t(\widehat{x}^{i_j}_{t-1}) \geq \mathfrak{Q}^{i_{j-1}+1}_t(\widehat{x}^{i_{j-1}}_{t-1}) - \rho_t \|\widehat{x}^{i_j}_{t-1} - \widehat{x}^{i_{j-1}}_{t-1}\|,$$

where the first inequality follows from the monotonicity of $\mathfrak{Q}_t(\cdot)$ in i and the second inequality uses Lemma 3.11.

By adding zero, we obtain

$$\begin{aligned}\mathfrak{Q}^{i_j+1}_t(\widehat{x}^{i_j}_{t-1}) \geq\ & \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) + \mathfrak{Q}^{i_{j-1}+1}_t(\widehat{x}^{i_{j-1}}_{t-1}) \\ & - \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) - \rho_t \|\widehat{x}^{i_j}_{t-1} - \widehat{x}^{i_{j-1}}_{t-1}\|.\end{aligned}$$

We can now also use the monotonicity of $\underline{\widehat{Q}}^R_t(\cdot)$ in i and apply Lemma 4.1 to obtain

$$\begin{aligned}\mathfrak{Q}^{i_j+1}_t(\widehat{x}^{i_j}_{t-1}) &\geq \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) + \mathfrak{Q}^{i_{j-1}+1}_t(\widehat{x}^{i_{j-1}}_{t-1}) \\ &\quad - \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}+1}_{t+1}) - \rho_t \|\widehat{x}^{i_j}_{t-1} - \widehat{x}^{i_{j-1}}_{t-1}\| \\ &\geq \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) \\ &\quad - (L^R_t + \rho_t)\|\widehat{x}^{i}_{t-1} - \widehat{x}^{i}_{\mathbb{B},t-1}\| - \rho_t \|\widehat{x}^{i_j}_{t-1} - \widehat{x}^{i_{j-1}}_{t-1}\|\end{aligned} \tag{18}$$

Moreover, we expand

$$\underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) = \widehat{f}_t(\widehat{x}^{i_{j-1}}_t, \widehat{y}^{i_{j-1}}_t) + \sigma_t \|\widehat{x}^{i_{j-1}}_{t-1} - \widehat{z}^{i_{j-1}}_t\| + \mathfrak{Q}^{i_{j-1}}_{t+1}(\widehat{x}^{i_{j-1}}_t) \tag{19}$$

and

$$\widehat{Q}^R_t(\widehat{x}^{i_{j-1}}_{t-1}) = \widehat{f}_t(\tilde{x}_t, \tilde{y}_t) + \sigma_t \|\widehat{x}^{i_{j-1}}_{t-1} - \tilde{z}_t\| + \widehat{Q}^R_{t+1}(\tilde{x}_t) \tag{20}$$

with corresponding optimal points $(\widehat{z}^{i_{j-1}}_t, \widehat{x}^{i_{j-1}}_t, \widehat{y}^{i_{j-1}}_t)$ and $(\tilde{z}_t, \tilde{x}_t, \tilde{y}_t)$.

Then with (20) it follows

$$\widehat{Q}^R_t(\widehat{x}^{i_{j-1}}_{t-1}) \leq \widehat{f}_t(\widehat{x}^{i_{j-1}}_t, \widehat{y}^{i_{j-1}}_t) + \sigma_t \|\widehat{x}^{i_{j-1}}_{t-1} - \widehat{z}^{i_{j-1}}_t\| + \widehat{Q}^R_{t+1}(\widehat{x}^{i_{j-1}}_t)$$

as the solution from (19) is feasible. Thus,

$$\widehat{Q}^R_t(\widehat{x}^{i_{j-1}}_{t-1}) - \underline{\widehat{Q}}^{R,i_{j-1}}_t(\widehat{x}^{i_{j-1}}_{t-1}, \mathfrak{Q}^{i_{j-1}}_{t+1}) \leq \widehat{Q}^R_{t+1}(\widehat{x}^{i_{j-1}}_t) - \mathfrak{Q}^{i_{j-1}}_{t+1}(\widehat{x}^{i_{j-1}}_t).$$

Rearranging yields

$$\underline{\widehat{Q}}_t^{R,i_{j-1}}(\widehat{x}_{t-1}^{i_{j-1}}, \mathfrak{Q}_{t+1}^{i_{j-1}}) \geq \widehat{Q}_t^R(\widehat{x}_{t-1}^{i_{j-1}}) + \mathfrak{Q}_{t+1}^{i_{j-1}}(\widehat{x}_t^{i_{j-1}}) - \widehat{Q}_{t+1}^R(\widehat{x}_t^{i_{j-1}}). \tag{21}$$

With inserting (21) in (18), we obtain

$$\begin{aligned}\mathfrak{Q}_t^{i_j+1}(\widehat{x}_{t-1}^{i_j}) \geq\ & \underbrace{\widehat{Q}_t^R(\widehat{x}_{t-1}^{i_{j-1}})}_{(*)} + \underbrace{\mathfrak{Q}_{t+1}^{i_{j-1}}(\widehat{x}_t^{i_{j-1}}) - \widehat{Q}_{t+1}^R(\widehat{x}_t^{i_{j-1}})}_{(\#)} \\ & - \underbrace{(L_t^R + \rho_t)\|\widehat{x}_{t-1}^i - \widehat{x}_{\mathbb{B},t-1}^i\|}_{(+)} - \underbrace{\rho_t\|\widehat{x}_{t-1}^{i_j} - \widehat{x}_{t-1}^{i_{j-1}}\|}_{(-)}\end{aligned}$$

We take limits on both sides. $(*)$ converges to $\widehat{Q}_t^R(\widehat{x}_{t-1}^*)$, since the function is continuous. (#) becomes greater than or equal to zero by the induction hypothesis. $(+)$ tends to zero with Lemma 3.6 since with j to ∞, the binary precision β_t goes to 0. $(-)$ tends to zero as $\widehat{x}_{t-1}^{i_j}$ and $\widehat{x}_{t-1}^{i_{j-1}}$ both converge to $\widehat{x}_{t-1}^*$.

Thus, the induction is proven for t. As this result holds for any cluster point of $(\widehat{x}^i)_{i\in\mathbb{N}}$, the assertion follows. □

E Proof of Theorem 4.3

Proof Consider the first stage optimal value $\widehat{v}^{FP,i}$ of the forward pass. By recursion we obtain

$$\widehat{v}^{FP,i} = \overline{\widehat{v}}^i + \sum_{t=2}^{T} \mathfrak{Q}_t(\widehat{x}_{t-1}^i) - \underline{\widehat{Q}}_t^{R,i}(\widehat{x}_{t-1}^i, \mathfrak{Q}_{t+1}^i)$$

and hence

$$\widehat{v}^{FP,i} \geq \overline{\widehat{v}}^i + \sum_{t=2}^{T} \mathfrak{Q}_t(\widehat{x}_{t-1}^i) - \widehat{Q}_t^R(\widehat{x}_{t-1}^i) \geq \widehat{v}^R + \sum_{t=2}^{T} \mathfrak{Q}_t(\widehat{x}_{t-1}^i) - \widehat{Q}_t^R(\widehat{x}_{t-1}^i). \tag{22}$$

As in the proof of Lemma 4.2, let $(\widehat{x}^{i_j})_{j\in\mathbb{N}}$ denote a convergent subsequence of $(\widehat{x}^i)_{i\in\mathbb{N}}$, with $\lim_j \widehat{x}^{i_j} = \widehat{x}^*$. Applying (22) to this subsequence and taking limits on both sides, yields

$$\begin{aligned}\lim_j \widehat{v}^{FP,i_j} &\geq \widehat{v}^R + \lim_j \left(\sum_{t=2}^{T} \mathfrak{Q}_t(\widehat{x}_{t-1}^{i_j}) - \widehat{Q}_t^R(\widehat{x}_{t-1}^{i_j})\right) \\ &\geq \widehat{v}^R + \sum_{t=2}^{T} \underbrace{\left(\widehat{Q}_t^R(\widehat{x}_{t-1}^*) - \widehat{Q}_t^R(\widehat{x}_{t-1}^*)\right)}_{=0}.\end{aligned}$$

The second inequality here stems from Equation (13). Using Lemma 3.4 and Assumption **(A3)**, yields $\lim_j \widehat{v}^{FP,i_j} \geq \widehat{v}$.

As $\widehat{v}^{FP,i_j}$ is also a lower bound to $\widehat{v}$, we have $\lim_j \widehat{v}^{FP,i_j} \leq \widehat{v}$. Thus, $\lim_j \widehat{v}^{FP,i_j} = \widehat{v}$. Since this is true for any cluster point $\widehat{x}^*$ of $(\widehat{x}^i)_{i\in\mathbb{N}}$, the inner loop converges to the optimal value $\widehat{v}$. With a similar reasoning it follows that every such cluster point is an optimal point of $(\widehat{P})$. □

F Proof of Theorem 4.7

Proof Let $\mathrm{x}^* := \big((z_t^*, x_t^*, y_t^*)\big)_{t=1,\dots,T}$ be an optimal point of $(\boldsymbol{P})$ and let $\big((\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)\big)_{t=1,\dots,T}$ be an optimal point of its outer approximation $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$. Then, we have

$$v - \widehat{v}^\ell = \sum_{t=1}^{T} f_t(x_t^*, y_t^*) - \widehat{f}_t^\ell(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \tag{23}$$

As $(\widehat{\boldsymbol{P}}^{\boldsymbol{\ell}})$ is a relaxation of $(\boldsymbol{P})$ this expression is clearly non-negative for all ℓ. Moreover, analogously to the feasibility result in Lemma 4.5 for sufficiently large ℓ, for all t we have

$$0 \leq f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell) - \widehat{f}_t^\ell(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \leq \varepsilon_{f_t}. \tag{24}$$

We distinguish two cases: First, let $\sum_{t=1}^T f_t(x_t^*, y_t^*) \leq \sum_{t=1}^T f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell)$, *e.g.*, because $(\widehat{z}_t^\ell, \widehat{x}_t^\ell, \widehat{y}_t^\ell)$ is feasible for $(\boldsymbol{P})$. Then, inserting this into (23) and using (24) it directly follows $v - \widehat{v}^\ell \leq \sum_{t=1}^T \varepsilon_{f_t}$.

Now let $\sum_{t=1}^T f_t(x_t^*, y_t^*) > \sum_{t=1}^T f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell)$. With Lemma 4.6, for any $\delta > 0$, there exists some $\widehat{\ell} \in \mathbb{N}$ such that for all $\ell \geq \widehat{\ell}$ there exists some feasible point $\widetilde{\mathrm{x}}^\ell := \big((\widetilde{z}_t^\ell, \widetilde{x}_t^\ell, \widetilde{y}_t^\ell)\big)_{t=1,\dots,T}$ of $(\boldsymbol{P})$ with $\|\widetilde{\mathrm{x}}^\ell - \widehat{\mathrm{x}}^\ell\|_2 \leq \delta$.

Clearly, $\sum_{t=1}^T f_t(x_t^*, y_t^*) \leq \sum_{t=1}^T f_t(\widetilde{x}_t^\ell, \widetilde{y}_t^\ell)$. Therefore,

$$0 \leq \sum_{t=1}^{T} f_t(x_t^*, y_t^*) - \sum_{t=1}^{T} f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \leq \sum_{t=1}^{T} \Big(f_t(\widetilde{x}_t^\ell, \widetilde{y}_t^\ell) - f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \Big). \tag{25}$$

With Assumption **(A1)** f_t is Lipschitz continuous with some constant $L_{f_t} > 0$. Thus, $\sum_{t=1}^T f_t$ is Lipschitz continuous with constant $L_f := \sum_{t=1}^T L_{f_t}$ and (25) can be bounded from above by $L_f \delta$.

We can write the right-hand side of (23) as

$$\sum_{t=1}^{T} \Big(f_t(x_t^*, y_t^*) - f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \Big) + \sum_{t=1}^{T} \Big(f_t(\widehat{x}_t^\ell, \widehat{y}_t^\ell) - \widehat{f}_t^\ell(\widehat{x}_t^\ell, \widehat{y}_t^\ell) \Big).$$

Then, with (24) and the previous result it follows that $v - \widehat{v}^\ell \leq L_f \delta + \sum_{t=1}^T \varepsilon_{f_t}$.

Choosing $\varepsilon := L_f \delta + \sum_{t=1}^{T} \varepsilon_{f_t}$ proves the assertion. □

G Proof of Theorem 4.8

Proof (a) From Theorem 4.7 it follows that if NC-NBD does not terminate for $\varepsilon = 0$, infinitely many piecewise linear relaxation refinements occur and $\widehat{\overline{v}}^{\ell}$ converges to v. Using the premise, we have $\widehat{\overline{v}}^{\ell} = \widehat{\underline{v}}^{\ell}$. Therefore, $\widehat{\underline{v}}^{\ell}$ converges to v. This also implies that the cut approximations $\mathfrak{Q}_{t+1}^{\ell}(\cdot)$ become tight at x_t^{ℓ} asymptotically. Thus, the solutions $\big((z_t^{\ell}, x_t^{\ell}, y_t^{\ell})\big)_{t=1,\dots,T}$ converge to an optimal solution for $(\boldsymbol{P})$.

(b) For sufficiently large ℓ, as in the proof of Theorem 4.7, we have

$$\sum_{t=1}^{T} f_t(\widehat{x}_t^{\ell}, \widehat{y}_t^{\ell}) \leq \sum_{t=1}^{T} \widehat{f}_t^{\ell}(\widehat{x}_t^{\ell}, \widehat{y}_t^{\ell}) + \sum_{t=1}^{T} \varepsilon_{f_t}.$$

Using this, and the termination of the inner loop, it follows

$$\sum_{t=1}^{T} f_t(\widehat{x}_t^{\ell}, \widehat{y}_t^{\ell}) \leq \overline{\widehat{v}}^{\ell} + \sum_{t=1}^{T} \varepsilon_{f_t} \leq \underline{\widehat{v}}^{\ell} + \widehat{\varepsilon} + \sum_{t=1}^{T} \varepsilon_{f_t} \leq v + \widehat{\varepsilon} + \sum_{t=1}^{T} \varepsilon_{f_t}.$$

For ℓ approaching infinity, $\widehat{x}^{\ell}$ becomes feasible. Thus,

$$v \leq \lim_{\ell} \overline{v}^{\ell} \leq \lim_{\ell} \sum_{t=1}^{T} f_t(\widehat{x}_t^{\ell}, \widehat{y}_t^{\ell}) \leq \lim_{\ell} v + \widehat{\varepsilon} + \sum_{t=1}^{T} \varepsilon_{f_t} = v + \widehat{\varepsilon}.$$

Since, $\overline{v}^{\ell}$ is bounded from above and non-increasing, the limit exists. This proves the assertion.

(c) This follows directly from b).

□

H Implementation details

The NC-NBD method is implemented in Julia-1.5.3 [7] using the `JuMP.jl` package [13] for optimization. The implementation is mainly derived from package `SDDP.jl` [12], which is enhanced by extensions specific to NC-NBD. To model piecewise linear approximations of multidimensional functions, we draw on `Delaunay.jl` [43] to determine triangulations. All MILP subproblems are solved with CPLEX and all MINLP subproblems are solved with appropriate MINLP solvers, both accessed using `GAMS.jl` [17]. The Lagrangian duals are solved using Kelley's cutting-plane method or a Level Bundle method as implemented in `SDDiP.jl` [24]. To reduce the size of the considered subproblems, a very basic Level cut selection technique is used based on `SDDP.jl`. In our case, however, not only the previously visited trial points, but also the anchor points are used to determine dominated cuts. Our code is available on GitHub [15].

All computations are performed on a machine with Intel(R) Xeon(R) E5-1630 v4 CPU and 128 GB RAM. All benchmark runs using state-of-the-art solvers are executed in GAMS 32.1.0.

I Unit commitment problem formulation and results

We consider a unit commitment problem with thermal generators based on [2] for some first tests of NC-NBD. To formulate this problem, we define the following elements:
Sets:

- $\mathcal{G}$: set of thermal generators

Data:

- p_g^f: price of fuel for generator g [EUR/MWh]
- $p_g^{\overline{s}}$: price of start up for generator g [EUR]
- $p_g^{\underline{s}}$: price of shut down for generator g [EUR]
- p_d: price for not meeting demand or load shedding [EUR/MWh]
- p_e: tax on emissions from generators [EUR/kg]
- d_t: demand at hour t [MWh]
- $\overline{c}_g$: maximum hourly generation of generator g [MWh]
- $\underline{c}_g$: minimum hourly generation of generator g [MWh]
- $\overline{r}_g$: ramp-up rate of generator g [MWh/h]
- $\underline{r}_g$: ramp-up rate of generator g [MWh/h]
- a_g, b_g, c_g: coefficients of the emission cost curve
- $v_g^a, v_g^b, v_g^c, v_g^d, v_g^e$: coefficients of the fuel cost curve

Decision Variables:

- x_{gt} : electricity production from generator g at time t [MWh]
- y_{gt} : binary variable modeling commitment of generator g at time t
- $\overline{y}_{gt}$: binary variable modeling start-up of generator g at time t
- $\underline{y}_{gt}$: binary variable modeling shut-down of generator g at time t
- $\underline{d}_t, \overline{d}_t$: variables modeling demand slack at time t

The objective is to minimize the total costs of electricity generation, which consists of different cost components. For all instances, the objective function is nonlinear due to a concave quadratic function modeling emission costs with $a_g < 0$ for all $g \in \mathcal{G}$ [11].

Additionally, we consider two different types of fuel cost function. In the first case (*base instances*), the fuel cost function is linear

$$c_{gt}^f(x_{gt}, y_{gt}) = p_g^f x_{gt}, \tag{26}$$

with a static fuel price p_g^f. In the second case (*valve-point instances*), we consider the more sophisticated cost function

$$c^f_{gt}(x_{gt}, y_{gt}) = v^a_g x^2_{gt} + v^b_g x_{gt} + v^c_g y_{gt} + v^d_g \big| \sin(v^e_g(\underline{c}_g - x_{gt}))\big|, \tag{27}$$

including a convex quadratic term and a sinusoidal term, modeling the so-called valve point effect of steam turbines [34].

Then, the unit commitment model reads

$$(\boldsymbol{P}_{UC}) \quad \min \quad \sum_{g\in\mathcal{G}}\sum_{t=1}^{T} c^f_{gt}(x_{gt}, y_{gt}) + \sum_{g\in\mathcal{G}}\sum_{t=1}^{T} p_g^{\bar{s}} \overline{y}_{gt} + \sum_{g\in\mathcal{G}}\sum_{t=1}^{T} p_g^{\underline{s}} \underline{y}_{gt} \tag{28}$$

$$+ \sum_{g\in\mathcal{G}}\sum_{t=1}^{T} p^e_g \left(a_g x^2_{gt} + b_g x_{gt} + c_g y_{gt} \right) + \sum_{t=1}^{T} p_d(\overline{d}_t + \underline{d}_t) \tag{29}$$

$$\text{s.t.} \quad \sum_{g\in\mathcal{G}} x_{gt} + \overline{d}_t - \underline{d}_t = d_t, \quad \forall t = 1,\dots,T \tag{30}$$

$$x_{gt} \leq \overline{c}_g y_{gt}, \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{31}$$

$$x_{gt} \geq \underline{c}_g y_{gt}, \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{32}$$

$$x_{gt} - x_{g,t-1} \leq \overline{r}_g y_{g,t-1} + \underline{c}_g(1 - y_{g,t-1}), \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{33}$$

$$x_{g,t-1} - x_{g,t} \leq \underline{r}_g y_{gt} + \underline{c}_g(1 - y_{gt}), \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{34}$$

$$\overline{y}_{gt} \geq y_{gt} - y_{g,t-1}, \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{35}$$

$$\underline{y}_{gt} \geq y_{g,t-1} - y_{gt}, \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{36}$$

$$x_{gt} \geq 0, \quad \forall g \in \mathcal{G}, t = 1,\dots,T \tag{37}$$

$$\underline{d}_t, \overline{d}_t \geq 0, \quad \forall t = 1,\dots,T \tag{38}$$

$$y_{gt}, \overline{y}_{gt}, \underline{y}_{gt} \in \{0,1\}, \quad \forall g \in \mathcal{G}, t = 1,\dots,T. \tag{39}$$

In this model, both, the continuous generation variables x_{gt} and the commitment variables y_{gt}, for all $g \in G$, $t = 1,\dots,T$, act as state variables. For the states at time $t = 0$, we use some fixed inputs. (30) denotes the balance between generation and demand. It also contains slack variables for unmet or overfulfilled demand, which are penalized in the objective. By this construction, Assumption **(A2)** is satisfied. (31)–(32) denote limits to the generator output, while (33)–(34) define ramping constraints. Those ramping constraints require x_{gt} to be a state variable. (33)–(34) are required to model start-up and shut-down costs.

For our input data, we draw on unit commitment instances created by Frangioni and published in the OR-Library [3]. The data is enhanced by own assumptions, as it does not cover all inputs in our problem formulation. We consider a number T of stages between 2 and 36 and a number G of generators between 3 and 10.

We solve all instances using NC-NBD with a relative optimality tolerance of 1% for the outer loop. All MILP subproblems are solved exactly while the outer loop MINLPs

Table 2 Bounds obtained for base instances in 100 EUR

T	G	BARON v	NC-NBD UB	NC-NBD LB	NC-NBD Time (s)
2	5	587.4	587.4	586.5	8
2	10	1266.1	1266.4	1262.3	93
3	5	849.8	849.8	849.3	17
3	10	1830.5	1830.5	1816.3	1502
4	5	1110.2	1110.2	1106.8	25
5	5	1374.7	1374.7	1372.6	202
10	3	2593.6	2598.3	2587.5	325
24	3	7037.5	7037.5	7037.5	674

are solved to an optimality tolerance of 10^{-3}. We use the lower bound provided by the MINLP solver to ensure that still a valid lower bound is obtained in the outer loop. The Lagrangian duals are solved with an optimality tolerance of 10^{-4}. In case that σ_t is not chosen large enough for some stage t from the beginning, it is increased iteratively within the solution procedure once identified. For the base instances, we use BARON [42,48] to solve the outer loop subproblem, for the valve-point instances we draw on LINDOGlobal [44], as BARON does not support sinusoidal functions. For the same reason, ANTIGONE [32], SCIP [16] and Gurobi [20] cannot be applied to the valve-point instances, so that we refer to LINDOGlobal and Couenne [5] as benchmarks.

The obtained upper bounds (UB) and lower bounds (LB) for the base instances are summarized in Table 2 and compared with the optimal point obtained by BARON.

All test instances can be solved by benchmark solvers in a few seconds, thus, outperforming NC-NBD. Still, these results can be regarded as a proof of concept for applying NC-NBD to multistage non-convex MINLPs, as in each case, the globally minimal point is succesfully approximated.

For the valve-point instances, we consider a larger number of stages, but only 3 or 4 generators, *i.e.*, 6 or 8 state variables, to focus on cases, in which NC-NBD looks most promising. For cases with many stages, we test differently scaled demand time series, as this seems to have a considerable effect on solution times. All instances are solved with a maximum solution time of two hours. The results are summarized in Table 3. If a solver does not terminate within the time limit, this is indicated by "-".

For a small number of stages, NC-NBD takes significantly more time than conventional solvers. With a larger number of stages, this difference vanishes, though. For 36 stages, NC-NBD manages to solve all considered instances within less than two hours, while LINDOGlobal and Couenne show more variance in computation time. For one instance, when terminated after two hours, they still show a 5% optimality gap.

Table 3 Results obtained for valve-point instances in 100 EUR

T	G	Demand		LINDOGlobal	Couenne	NC-NBD
10	3	v1	UB	3187.8	3178.0	3185.8
			LB	3166.7	3161.2	3158.6
			Time (s)	201	29	1074
16	3	v1	UB	5602.4	5584.1	5584.1
			LB	5553.0	5584.1	5548.3
			Time (s)	135	273	760
24	3	v1	UB	8593.4	8554.8	8554.8
			LB	8446.3	8545.0	8498.6
			Time (s)	–	4427	1408
24	3	v2	UB	9518.0	9508.8	9509.7
			LB	9428.5	9414.9	9432.4
			Time (s)	1562	647	771
24	4	v1	UB	9977.2	9965.1	9964.8
			LB	9848.3	9865.5	9867.5
			Time (s)	–	349	1220
24	4	v2	UB	8557.9	8587.1	8572.3
			LB	8473.1	8310.9	8492.2
			Time (s)	2603	–	4191
36	4	v1	UB	12,654.3	12,662.4	12,638.8
			LB	11,960.6	11,981.0	12,547.8
			Time (s)	–	–	6816
36	4	v2	UB	10,742.7	10,737.5	10,737.5
			LB	10,651.7	10,737.5	10,630.4
			Time (s)	2733	7501	3646
36	4	v3	UB	14,013.3	13,981.5	13,978.7
			LB	13,678.2	13,767.1	13,879.1
			Time (s)	–	–	3497

References

1. Ahmed, S., Cabral, F. G., Freitas Paulo da Costa, B.: Stochastic Lipschitz dynamic programming. Math. Programm. (2020)
2. Bacci, T., Frangioni, A., Gentile, C., Tavlaridis-Gyparakis, K.: New MINLP formulations for the unit commitment problems with ramping constraints. Preprint, http://www.optimization-online.org/DB_FILE/2019/10/7426.pdf (2019)
3. Beasley, J.E.: OR-library: distributing test problems by electronic mail. J. Oper. Res. Soc. **41**(11), 1069 (1990)
4. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. Acta Numer. **22**, 1–131 (2013)
5. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex minlp. Optim. Methods Softw. **24**, 597–634 (2009)
6. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numer. Math. **4**(1), 238–252 (1962)

7. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. SIAM Rev. **59**(1), 65–98 (2017)
8. Birge, J.R.: Decomposition and partitioning methods for multistage stochastic linear programs. Oper. Res. **33**(5), 989–1007 (1985)
9. Burlacu, R., Geißler, B., Schewe, L.: Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. Optim. Methods Softw. **35**(1), 37–64 (2020)
10. Cerisola, S., Latorre, J.M., Ramos, A.: Stochastic dual dynamic programming applied to nonconvex hydrothermal models. Eur. J. Oper. Res. **218**(3), 687–697 (2012)
11. Van Dinter, J., Rebennack, S., Kallrath, J., Denholm, P., Newman, A.: The unit commitment model with concave emissions costs: a hybrid benders' decomposition with nonconvex master problems. Ann. Oper. Res. **210**(1), 361–386 (2013)
12. Dowson, O., Kapelevich, L.: SDDP.jl: a Julia package for stochastic dual dynamic programming. INFORMS J. Comput. (2020)
13. Dunning, I., Huchette, J., Lubin, M.: JuMP: a modeling language for mathematical optimization. SIAM Rev. **59**(2), 295–320 (2017)
14. Feizollahi, M.J., Ahmed, S., Sun, A.: Exact augmented Lagrangian duality for mixed integer linear programming. Math. Program. **161**(1–2), 365–387 (2017)
15. Füllner, C.: NCNBD.jl. Code released on GitHub https://github.com/ChrisFuelOR/NCNBD.jl (2021)
16. Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.-K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S. J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M. E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D., Witzig, J.: The SCIP optimization suite 7.0. Technical report, Optimization Online (2020)
17. GAMS Software GmbH. GAMS.jl. Code released on GitHub https://github.com/GAMS-dev/gams.jl (2020)
18. Geißler, B.: Towards globally optimal solutions for MINLPs by discretization techniques with applications in gas network optimization. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2011)
19. Geoffrion, A.M.: Generalized Benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972)
20. LLC Gurobi Optimization. Gurobi optimization reference manual (2021)
21. Glover, F.: Improved linear integer programming formulations of nonlinear integer problems. Manage. Sci. **22**(4), 455–460 (1975)
22. Hjelmeland, M.N., Zou, J., Helseth, A., Ahmed, S.: Nonconvex medium-term hydropower scheduling by stochastic dual dynamic integer programming. IEEE Trans. Sustain. Energy **10**(1) (2019)
23. Infanger, G., Morton, D.P.: Cut sharing for multistage stochastic linear programs with interstage dependency. Math. Program. **75**(2), 241–256 (1996)
24. Kapelevich, L.: SDDiP.jl. Code released on GitHub https://github.com/lkapelevich/SDDiP.jl (2018)
25. Kallrath, J., Rebennack, S.: Computing area-tight piecewise linear overestimators, underestimators and tubes for univariate functions. In: Optimization in Science and Engineering, pp. 273–292. Springer (2014)
26. Kannan, R.: Algorithms, analysis and software for the global optimization of two-stage stochastic programs. PhD thesis, Massachusetts Institute of Technology (2018)
27. Kilinç, M.R., Sahinidis, N.V.: Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with baron. Optim. Methods Softw. **33**(3), 540–562 (2018)
28. Li, C., Grossmann, I.E.: A generalized Benders decomposition-based branch and cut algorithm for two-stage stochastic programs with nonconvex constraints and mixed-binary first and second stage variables. J. Global Optim. **75**(2), 247–272 (2019)
29. Li, X., Chen, Y., Barton, P.I.: Nonconvex generalized Benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. Ind. Eng. Chem. Res. **51**(21), 7287–7299 (2012)
30. Li, X., Tomasgard, A., Barton, P.I.: Nonconvex generalized Benders decomposition for stochastic separable mixed-integer nonlinear programs. J. Optim. Theory Appl. **151**, 425–454 (2011)
31. Meyer, R.R.: Integer and mixed-integer programming models: general properties. J. Optim. Theory Appl. **3**(4) (1975)
32. Misener, R., Floudas, C.A.: Antigone: algorithms for continuous/integer global optimization of nonlinear equations. J. Global Optim. **59**(2–3), 503–526 (2014)

33. Ogbe, E., Li, X.: A joint decomposition method for global optimization of multiscenario nonconvex mixed-integer nonlinear programs. J. Global Optim. **75**, 595–629 (2018)
34. Pedroso, J. P., Kubo, M., Viana, A.: Unit commitment with valve-point loading effect. Technical report, Universidade do Porto (2014)
35. Pereira, M.V.F., Pinto, L.M.V.G.: Multi-stage stochastic optimization applied to energy planning. Math. Program. **52**(1–3), 359–375 (1991)
36. Philpott, A.B., Wahid, F., Bonnans, F.: MIDAS: a mixed integer dynamic approximation scheme. Math. Program. (2019)
37. Rebennack, S.: Combining sampling-based and scenario-based nested Benders decomposition methods: application to stochastic dual dynamic programming. Math. Program. **156**(1–2), 343–389 (2016)
38. Rebennack, S.: Computing tight bounds via piecewise linear functions through the example of circle cutting problems. Math. Methods Oper. Res. **84**(1), 3–57 (2016)
39. Rebennack, S., Kallrath, J.: Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems. J. Optim. Theory Appl. **167**(2), 617–643 (2015)
40. Rebennack, S., Krasko, V.: Piecewise linear function fitting via mixed-integer linear programming. INFORMS J. Comput. (2020)
41. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis. Springer, Berlin (2009)
42. Sahinidis, N.V.: BARON 17.8.9: global optimization of mixed-integer nonlinear programs, User's Manual (2017)
43. Schnetter, E.: Delaunay.jl. Code released on GitHub https://github.com/eschnett/Delaunay.jl (2020)
44. Schrage, L.: *LindoSystems: LindoAPI* (2004)
45. Steeger, G., Lohmann, T., Rebennack, S.: Strategic bidding for a price-maker hydroelectric producer: stochastic dual dynamic programming and Lagrangian relaxation. IISE Trans. **47**, 1–14 (2018)
46. Steeger, G., Rebennack, S.: Dynamic convexification within nested Benders decomposition using Lagrangian relaxation: an application to the strategic bidding problem. Eur. J. Oper. Res. **257**(2), 669–686 (2017)
47. Tawarmalani, M., Sahinidis, N.: Convex extensions and envelopes of lower semi-continuous functions. Math. Program. **93**, 247–263 (2002)
48. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**, 225–249 (2005)
49. van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. J. SIAM Appl. Math. **17**(4), 638–663 (1969)
50. Vielma, J.P., Ahmed, S., Nemhauser, G.: Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions. Oper. Res. **58**(2), 303–315 (2010)
51. Vielma, J.P., Nemhauser, G.L.: Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Math. Program. **128**(1–2), 49–72 (2011)
52. Zhang, S., Sun, X. A.: Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. (2021). Available at https://arxiv.org/abs/1912.13278. Accessed 29 Nov 2021
53. Zhou, K., Kilinç, M.R., Chen, X., Sahinidis, N.V.: An efficient strategy for the activation of mip relaxations in a multicore global minlp solver. J. Global Optim. **70**(3), 497–516 (2018)
54. Zou, J., Ahmed, S., Sun, X.: Multistage stochastic unit commitment using stochastic dual dynamic integer programming. IEEE Trans. Power Syst. **34**(3) (2019)
55. Zou, J., Ahmed, S., Sun, X.A.: Stochastic dual dynamic integer programming. Math. Program. **175**(1–2), 461–502 (2019)