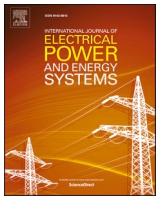
Electrical Power and Energy Systems 145 (2023) 108689

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes



Accelerated dual dynamic integer programming applied to short-term power generation scheduling

Kenny Vinente dos Santos [a,b,*], Bruno Colonetti [b], Erlon Cristian Finardi [b,c], Victor M. Zavala [d]

[a] Department of Oil and Gas, Federal University of Amazonas, 6200 Gen. Rodrigo, Octavio Av., Coroado I, Manaus 69080-900, Amazonas, Brazil
[b] Department of Electrical and Electronic Engineering, Federal University of Santa Catarina, Trindade, Florianopolis 88040-900, Santa Catarina, Brazil
[c] INESC P&D Brasil, Santos, 11055-300 Sao Paulo, Brazil
[d] Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1415 Engineering Dr, Madison, WI 53706, USA

ARTICLE INFO

Keywords:
Dual dynamic integer programming
Short-term generation scheduling
Mixed-integer linear programming

ABSTRACT

The short-term generation scheduling (STGS) problem defines which units must operate and how much power they must deliver to satisfy the system demand over a planning horizon of up to two weeks. The problem is typically formulated as a large-scale mixed-integer linear programming problem, where off-the-shelf commercial solvers generally struggle to efficiently solve realistic instances of the STGS, mainly due to the large-scale of these models. Thus, decomposition approaches that break the model into smaller instances that are more easily handled are attractive alternatives to directly employing these solvers. This paper proposes a dual dynamic integer programming (DDiP) framework for solving the STGS problem efficiently. As in the standard DDiP approach, we use a nested Benders decomposition over the time horizon but introduce multiperiod stages and overlap strategies to accelerate the method. Simulations performed on the IEEE-118 system show that the proposed approach is significantly faster than standard DDiP and can deliver near-optimal solutions.

1. Introduction

The short-term generation scheduling (STGS) problem aims to decide which generating units will be active and how much power they will produce for a day-ahead operation over a one-week planning horizon. The STGS solution is used for several systems operation purposes, especially as a reference point for real-time operation and determining the day-ahead spot prices. Despite being well-studied and utilized worldwide, the STGS still poses a considerable challenge because it needs to be solved rapidly. Specifically, the STGS problem is a large-scale, nonlinear, and nonconvex optimization problem. The STGS is typically simplified and formulated as a mixed-integer linear programming (MILP) problem, but even this reformulation can be intractable to modern, off-the-shelf optimization solvers.

In large-scale centrally dispatched hydrothermal systems like the Brazilian one, hundreds of thousands of variables and constraints are necessary to formulate the STGS as a MILP. Modern solvers struggle to find high-quality solutions for this problem in a reasonable time without the support of decomposition methods. The literature offers a variety of decomposition techniques that are tailored to the STGS problem. These strategies are commonly categorized as primal and dual strategies, where the first focuses on complicating variables to break the original problem, and the latter focuses on complicating constraints.

Dual strategies aim to decompose the problem into small subproblems that are easy to solve, normally finding a set of dual variables that maximizes the dual function. One bottleneck is that we do not obtain a feasible solution, demanding a primal recovery phase. Despite this fact, several works use dual strategies in the context of the STGS problem. For example, Lagrangian Relaxation is employed in [1,2].

Primal strategies do not need to perform the recovery phase, but they typically deal with more difficult subproblems than dual approaches and may present some difficulties in the convergence. In particular, Benders decomposition (BD) [3] is a popular method used for solving deterministic STGS problems [4-6], where the original problem is decomposed into a master problem (MP) and a subproblem (SP), and often requires a large number of iterations. Several acceleration techniques have been proposed to improve the convergence of the Benders method [7-9].

For the stochastic STGS problem, BD has also been applied [10,11]. Although a multistage decomposition can be employed, the problem is decomposed in a two-stage model. Stochastic dual dynamic programming (SDDP) [12,13] is a generalization of the BD approach that has been extensively used for solving problems where the resulting SPs are

* Corresponding author.
E-mail address: kennyvinente@ufam.edu.br (K.V. Santos).

https://doi.org/10.1016/j.ijepes.2022.108689
Received 25 May 2022; Received in revised form 16 August 2022; Accepted 20 September 2022
Available online 11 October 2022

Nomenclature

Indices

t	index of periods (hours).
g	index of thermal plants.
h	index of hydro plants.
b	index of buses.
l	index of transmission lines.
ref	index of reference bus.

Sets

B	buses.
B_b	buses connected to bus b.
G	thermal plant plants.
H	hydro plants.
L	transmission lines.
T	periods.
U_b	hydro and thermal plant plants connected to bus b.
Ω_h	hydro plants upstream of hydro h.

Variables

ph_{ht}	power output of hydro h in period t (p.u. of MW).
pl_{lt}	active power in transmission line l in period t (p.u. of MW).
pt_{gt}	power output of thermal plant g in period t (p.u. of MW).
qht	turbine discharge of hydro h in period t (m^3/s).
s_{ht}	spillage of hydro h in period t (m^3/s).
v_{ht}	reservoir volume of hydro h at period t (hm^3).
w_{gt}	binary variable indicating the shutdown of thermal plant g in period t.
x_{gt}	binary variable indicating the on/off status of thermal plant g in period t.
y_{ht}	binary variable indicating the on/off status of hydro h in period t.
z_{gt}	binary variable indicating the startup of thermal plant g in period t.
θ_{bt}	voltage angle of bus b in period t (radians).

Parameters

$\mathbf{a}_{mhk}$	hydro production function coefficient m of the of hydro h and hyperplane k.
$\mathbf{C0}_g$	unitary variable cost of thermal plant g ($/p.u. of MW).
$\mathbf{C1}_g$	fixed cost of thermal plant g ($).
$\mathbf{C2}_g$	startup cost of thermal plant g ($).
$\mathbf{C3}_g$	shutdown cost of thermal plant g ($).
$\mathbf{DT}_g$	minimum downtime of thermal plant g (h).
$\mathbf{K}$	constant that converts water flow (m^3/s) to volume (hm^3).
$\mathbf{M0}_{h,i}$	angular coefficient of reservoir h concerning the i-th linear equation of future cost function ($/$hm^3$).
$\mathbf{M1}_i$	independent term of the i-th linear equation of future cost function ($).
$\mathbf{N1}_h$	number of hyperplanes used to represent the hydro production function of hydro h.
$\mathbf{NF}$	number of linear equations used to represent the future cost function.
$\mathbf{NG}$	number of thermal plants.
$\mathbf{NH}$	number of hydro plants.
$\mathbf{NT}$	number of hours of the planning horizon.
$\mathbf{P}_{bt}$	demand of active power on bus b in period t (p.u. of MW).
$\mathbf{RD}_g$	maximum ramp-down of thermal plant g (p.u. of MW).
$\mathbf{RU}_g$	maximum ramp-up of thermal plant g (p.u. of MW).
$\mathbf{SR}_t$	system reserve in period t (p.u. of MW).
$\mathbf{UT}_g$	minimum uptime of thermal plant g (h).
$\mathbf{x}_{ab}$	reactance of the transmission line, which connects buses a and b (p.u.).
$\mathbf{Y}_{ht}$	incremental inflow of hydro h in period t (m^3/s).
τ_{ih}	water traveling time between hydro plants i and h (h).

linear programming (LP) problems. A variant of SDDP for solving mixed-integer SPs, known as stochastic dual dynamic integer programming (SDDiP), has been recently used for solving STGS problems [14,15] and has shown satisfactory performance.

In this work, we propose a variant of the nested Benders decomposition (NBD) strategy, also known as dual dynamic integer programming (DDiP) [16], to solve deterministic STGS problems. The DDiP temporal decomposition employs forward steps for obtaining upper bounds on the optimal value and backward steps that improve Piecewise Linear (PWL) models of recourse functions used to update lower bounds. It is important to note that only a few works in the literature have applied DDiP to realistic, large-scale power systems. Additionally, we note that standard DDiP implementation might fail to deliver a high-quality solution in a reasonable time due to drawbacks inherent to cutting-plane methods and the lack of the finite convergence property (due to the presence of mixed-integer stage variables).

As the first contribution of our work, we explore different multi-period aggregation strategies and analyze their effect on DDiP convergence (in terms of solution time and optimality gap). The key observation is the DDiP flexibility, which enables each subproblem to have its particular characteristics, such as different time-length stages, as long as the information from the original problem is not removed. Compared to a classical DDiP that defines a stage as a single period, the aggregation approach (i) reduces the number of cost-functions that need to be approximated, and (ii) the multiperiod stages lead to tight approximations for the cost-to-go functions with a reduced number of iterations. The block representation can also more naturally capture the multiple timescales in the STGS problem (e.g., ramp and minimum uptime and downtime equations).

The second contribution of our work is the introduction of overlapping decomposition schemes, recently used in optimal control [17]. This approach decomposes the planning horizon into a set of overlapping subdomains. Convergence is achieved by exchanging information in the overlapping region that couples neighboring partitions. This paradigm is adapted here to the context of BD, combined with a multiperiod stage that incorporates a specific set of constraints of the right neighboring partition. The key observation is that the overlapping partition allows subproblems to preserve some information on the problem structure, which helps accelerate convergence. To our knowledge, this is the first time this decomposition has been employed in BD. Our results confirm that the multiperiod and overlap strategies can identify high-quality solutions faster than standard DDiP in almost all cases simulated, showing the potential of the approaches used in this work.

The paper is organized as follows: Section 2 presents the STGS model, assuming a tight-pool market framework with centralized power generation dispatch. The DDiP algorithm and its improvements in this work are presented in Section 3. Experiments and results are presented in Section 4, and conclusions are presented in Section 5.

2. Short-term generation scheduling model

The STGS model considered in this work comprises a planning horizon with NT periods of one hour, and the network is modeled via a classical DC power flow model.

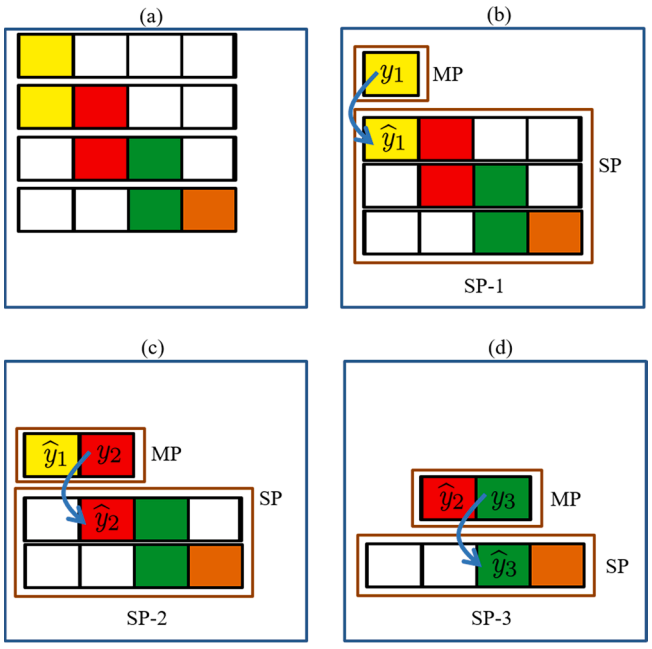


Fig. 1. Example of Nested Benders Decomposition, where (a) is the original problem. For this case, three SPs are obtained, where (b) is the SP-1, (c) is the SP-2, and (d) is the SP-3.

$$\min \sum_{t=1}^{\mathbf{NT}} \sum_{g=1}^{\mathbf{NG}} \left(\mathbf{C0}_g pt_{gt} + \mathbf{C1}_g x_{gt} + \mathbf{C2}_g z_{gt} + \mathbf{C3}_g w_{gt}\right) + \alpha$$

$$\text{s.t.}: z_{gt} - w_{gt} = x_{gt} - x_{g,t-1},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1b}$$

$$\sum_{i=\max(t-\mathbf{UT}_g+1,1)}^{t} z_{gt} \leqslant x_{gt},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1c}$$

$$\sum_{i=\max(t-\mathbf{DT}_g+1,1)}^{t} w_{gi} \leqslant 1 - x_{gt},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1d}$$

$$pt_{gt} - pt_{g,t-1} \leqslant \mathbf{RU}_g x_{g,t-1} + \underline{\mathbf{pt}}_g z_{gt},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1e}$$

$$pt_{g,t-1} - pt_{gt} \leqslant \mathbf{RD}_g x_{gt} + \underline{\mathbf{pt}}_g w_{gt},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1f}$$

$$\underline{\mathbf{pt}}_g x_{gt} \leqslant pt_{gt} \leqslant \overline{\mathbf{pt}}_g x_{gt},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1g}$$

$$ph_{ht} \leqslant \mathbf{a}_{0hk} v_{ht} y_{ht} + \mathbf{a}_{1hk} q_{ht} + \mathbf{a}_{2hk} s_{ht} + \mathbf{a}_{3hk} y_{ht},$$

$$\forall h \in \mathrm{H}, \forall t \in \mathrm{T}, k = 1, \ldots, \mathbf{N1}_h \tag{1h}$$

$$\underline{\mathbf{q}}_h y_{ht} \leqslant q_{ht} \leqslant \overline{\mathbf{q}}_h y_{ht},\ \forall h \in \mathrm{H}, \forall t \in \mathrm{T} \tag{1i}$$

$$v_{ht} = v_{h,t-1} - \mathbf{K}\left(q_{ht} + s_{ht} - \sum_{\forall i \in \Omega_h} \left(q_{i,t-\tau_{ih}} + s_{i,t-\tau_{ih}}\right) - \mathbf{Y}_{ht}\right), \quad \forall h \in \mathrm{H}, \forall t \in \mathrm{T} \tag{1j}$$

$$\alpha \geqslant \mathbf{M1}_i - \sum_{\forall h \in \Psi} \mathbf{M0}_{hi} v_{h,\mathbf{NT}},\ \forall i = 1, \ldots, \mathbf{NF} \tag{1k}$$

$$\sum_{\forall g \in \mathrm{U_b}} pt_{gt} + \sum_{\forall h \in \mathrm{U_b}} ph_{ht} - \sum_{\forall l \in \mathrm{B_b}} pl_{lt} = \mathbf{P}_{bt}, \quad \forall \mathrm{b} \in \mathrm{B}, \forall \mathrm{t} \in \mathrm{T} \tag{1l}$$

$$pl_{lt} = (\theta_{at} - \theta_{bt})/\mathbf{X}_{ab},\ \forall l : (a,b) \in \mathrm{L}, \forall t \in \mathrm{T} \tag{1m}$$

$$\sum_{g=1}^{\mathbf{NG}} (\overline{\mathbf{pt}}_g x_{gt} - pt_{gt}) x_{gt} + \sum_{h=1}^{\mathbf{NH}} (\overline{\mathbf{ph}}_h y_{ht} - ph_{ht}) \geqslant \mathbf{SR}_t, \quad \forall t \in \mathrm{T} \tag{1n}$$

$$\underline{\mathbf{v}}_h \leqslant v_{ht} \leqslant \overline{\mathbf{v}}_h;\ 0 \leqslant s_{ht} \leqslant \overline{\mathbf{s}}_h;\ 0 \leqslant ph_{ht} \leqslant \overline{\mathbf{ph}}_h, \quad \forall h \in \mathrm{H}, \forall t \in \mathrm{T} \tag{1o}$$

$$\underline{\mathbf{pl}}_l \leqslant pl_{lt} \leqslant \overline{\mathbf{pl}}_l,\ \forall l : (a,b) \in \mathrm{L}, \forall t \in \mathrm{T} \tag{1p}$$

$$\underline{\theta}_b \leqslant \theta_{bt} \leqslant \overline{\theta}_b,\ \forall b \in \mathrm{B}, \forall t \in \mathrm{T} \tag{1q}$$

$$x_{gt}, z_{gt}, w_{gt} \in \{0,1\},\ \forall g \in \mathrm{G}, \forall t \in \mathrm{T} \tag{1r}$$

$$y_{ht} \in \{0,1\},\ \forall h \in \mathrm{H}, \forall t \in \mathrm{T} \tag{1s}$$

$$\theta_{ref,t} = 0,\ \forall t \in \mathrm{T}. \tag{1t}$$

The objective function comprises the thermal generation costs, which account for the startup, nominal operation, and shutdown terms, and the variable α is related to the future cost function (FCF) [13,18]. The FCF represents the expected operational cost of the system as a function of the water stored in the reservoirs at the end of the planning horizon. The constraints related to the thermal plant unit commitment (TUC) are represented by (1b)-(1g). Equation (1h) represents the hydro production function (HPF) for a plant-based model [19], where the non-convexities inherent to the HPF are dealt with a convex hull approach, which is a reasonable approximation, especially in the operative zones domain [20]. Equation (1j) is the water balance (WB) constraint, and (1k) is related to the PWL model for the FCF. Equations (1l)-(1m) are related to the DC network power flow model, where the transmission line l, which connects buses a and b, has reactance $\mathbf{x}_{ab}$. Equation (1n) is the system reserve constraint. Equations (1o)-(1t) impose limits on variables, where the notation $\overline{\mathbf{x}}$, $\underline{\mathbf{x}}$ is the upper and lower limit of variable x, respectively.

3. Dual dynamic integer programming

Certain optimization problems have a particular structure, such as Fig. 1(a), which allows applying the BD in different nested schemes. This particular approach is the Nested BD (NBD) and was first proposed to solve a multistage stochastic linear problem in [21]. Each pair of adjacent stages is considered a particular SP. As an example, Fig. 1(b)-(d) presents a case of the NBD applied to the problem represented in Fig. 1 (a). The original problem is decomposed into an MP and an SP, where the decisions variables of the MP are sent to the SP (represented by the variable with an over hat symbol).

BD and NBD require convex SPs to generate valid cuts; therefore, these approaches do not allow sets of integer variables. To overcome this issue in MILP problems, DDiP generalizes the NBD with these particular sets. To illustrate the DDiP approach, consider the following MILP problem in a compact representation.

$$\min \sum_{t=1}^{T} \left(d_t^{\mathrm{T}} y_t + c_t^{\mathrm{T}} x_t\right) \tag{2a}$$

$$\text{s.t. } A_t y_t = b_t,\ \forall t \in \{1, \ldots, T\} \tag{2b}$$

$$D_t y_t + B_t x_t = g_t,\ \forall t \in \{1, \ldots, T\} \tag{2c}$$

$$F_t y_t + E_t y_{t-1} = h_t,\ \forall t \in \{2, \ldots, T\} \tag{2d}$$

$$y_t \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2},\ \forall t \in \{1, \ldots, T\} \tag{2e}$$

$$x_t \in \mathbb{Z}^{m_1} \times \mathbb{R}^{m_2},\ \forall t \in \{1, \ldots, T\} \tag{2f}$$

where $n = n_1 + n_2$, $m = m_1 + m_2$ are the dimensions of vectors y_t and x_t, respectively, $\forall t$ belonging to the set of stages {1,2,…,T} (the

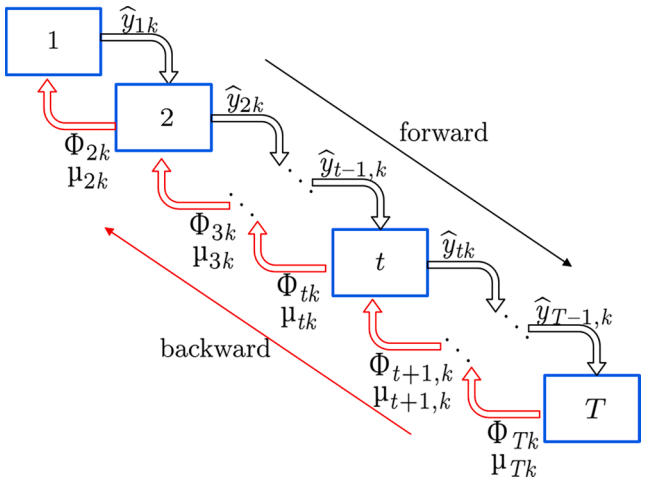


Fig. 2. An overview of the DDiP scheme at iteration k.

dimensions of $x_t y_t$ are the same just for simplicity). In this problem, x_t are local variables (non-linking), and y_t are linking variables, and both can contain continuous and discrete components, $A_t D_t B_t F_t E_t$ are matrices and $d_t c_t b_t g_t h_t$ are vectors on the compatible dimensions. Problem (2) can be cast as dynamic programming [16]. In this case, for each stage t, there is an optimization problem that minimizes the sum of the current stage cost and a cost-to-go function, which contains the cost of the future stages. Thus, introducing auxiliary variables $z_t \in \mathrm{R}^n$ to represent the previous stage variable y_{t-1}, the cost-to-go function for stage t, $Q_t(x_t y_t y_{t-1})$, is given by:

$$Q_t(\mathrm{x_t}, \mathrm{y_t}, \mathrm{y_{t-1}}) = \min \; d_t^{\mathrm{T}} y_t + c_t^{\mathrm{T}} x_t + Q_{t+1}(y_t) \tag{3a}$$

$$\text{s.t. } A_t y_t = b_t \tag{3b}$$

$$D_t y_t + B_t x_t = g_t \tag{3c}$$

$$F_t y_t + E_t y_{t-1} = h_t \tag{3d}$$

$$z_t = y_{t-1} \tag{3e}$$

$$y_t \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \tag{3f}$$

$$x_t \in \mathbb{Z}^{m_1} \times \mathbb{R}^{m_2} \tag{3g}$$

$$z_t \in \mathbb{R}^n, \tag{3h}$$

with $Q_{T+1} = 0$. The DDiP scheme can now be applied to the problem (3). The proposed scheme approximates the cost-to-function Q_t by a function Φ_t modeled by PWL equations obtained in each iteration. Thus, to obtain these equations, the cost-to-go functions must be convex; therefore, an LP relaxation of stage SPs is imposed. Using an iterative scheme based on the forward and backward passes, the DDiP aims to improve ϕ_t iteratively. At iteration k, the forward pass solves a sequence of SPs formulated by Equations (4a)-(4i) for each stage t, following the order $t = 1,2,\ldots,T$, as presented in Fig. 2 (highlighted by the black arrow).

$$\Phi_{\mathrm{tk}}\left(x_t, y_t, y_{t-1}^*\right) = \min \; d_t^{\mathrm{T}} y_t + c_t^{\mathrm{T}} x_t + \phi_{t+1} \tag{4a}$$

$$\text{s.t. } A_t y_t = b_t \tag{4b}$$

$$D_t y_t + B_t x_t = g_t \tag{4c}$$

$$F_t y_t + E_t z_t = h_t \tag{4d}$$

$$z_t = y_{t-1,k}^* : \left(\mu_{tk}\right) \tag{4e}$$

$$\phi_{t+1} \geqslant \Phi_{t+1,l} + \left(\mu_{t+1,l}\right)^{\mathrm{T}} \left(y_t - y_{tl}^*\right),\ l = 0, \ldots, k-1 \tag{4f}$$

$$y_t \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \tag{4g}$$

$$x_t \in \mathbb{Z}^{m_1} \times \mathbb{R}^{m_2} \tag{4h}$$

$$z_t \in \mathbb{R}^n, \tag{4i}$$

where φ_{t+1} is a PWL approximation of Φ_{t+1}, μ is a dual variable (only valid for the continuous relaxation of the problem) associated with constraint (4e), $y^*_{t-1,k}$ is the decision obtained for y in the previous stage $t-1$ at iteration k, and Equation (4f) contains the collection of Benders cuts obtained from previous iterations, used to approximate Φ_{t+1}. The forward pass yields an upper bound and a feasible solution to the original problem. After finishing the forward, the backward pass is performed. In the backward, a sequence of relaxed versions of the problem (4) for each t is solved, aiming to update the cost-to-go function Φ_t, and obtain the dual variables μ_t. The order in which the SPs are solved is the reverse of the forward pass, i.e., $t = \{T,\ldots,2,1\}$, as shown in Fig. 2 (red arrows). The backward pass yields a lower bound to the optimal value of the original problem.

To summarize, the standard DDiP scheme is:

Algorithm 1: The standard DDiP scheme

Input: A MILP problem of type (2)
Output: Solution of MILP, y^*,x^* and $f^* = d^T y^* + c^T x^*$
Initial Setup: start with $\Phi_{T+1,0} = \mu_{T+1,0} = 0$. Set the iteration counter $k = 0$, a tolerance ε and an initial upper bound UB
for $k = 0,1,\ldots$ **do**
  The Forward Pass: for $t = 1,\ldots,T$, solve the MILP problem (4), obtaining y_t, x_t. Compute the current upper bound (ub_k), using:
    $ub_k = \sum_{t=1}^{T} d_t^{\mathrm{T}} y_{t,k} + c_t^{\mathrm{T}} x_{t,k}$ (5)
  if $UB < ub_k$ **then**
    $UB = ub_k$
  end
  The Backward Pass: for $t = T,\ldots,1$, solve a relaxation of problem (4), obtaining $\Phi_{t,k}$, $\mu_{t,k}$. For each stage t, a Benders cut is constructed from the dual variables $\mu_{t,k}$ and inserted in the subsequent backward problem referring to $t-1$ period in the form of Equation (4f). At the $t = 1$ period, we evaluate the current lower bound (lb_k), using Equation (6):
    $lb_k = \Phi_{1,k}$ (6)
  if $gap = (UB - lb_k)/UB \leq \varepsilon$ **then**
    Stop (converged)
  end
end
return y^*, x^*, f^* if converged.

A key advantage of DDiP is that it decomposes the original problem into more easily SPs. A disadvantage is that DDiP does not have guaranteed finite convergence (due to the presence of integer state variables). However, experimental results show that the optimality gap can be relatively small, and the solution is mostly near-optimal. Also, the BD requires that all decisions obtained in the MP must be valid in the SP; otherwise, the SP will be infeasible. Thus, to avoid that, usually, feasibility cuts are created. However, since the use of feasibility cuts can increase the computational effort drastically, in this work, slack variables are added in constraints (4b)-(4d) with a high penalty value in the objective function. Adding slack variables makes every solution in the MP feasible, as the price to obtain a positive value for some of these variables.

4. Improvements in DDiP

Different approaches have been developed to improve the performance of BD algorithms when applied to MILP problems. However, in the context of DDiP, only a few works have been reported. Seminal works that explore BD algorithms with integer state variables are [14,15,22]. In [23], different cuts classes, e.g., Benders, Lagrangian, strengthened Benders, and integer optimality cuts, are explored. In [24],

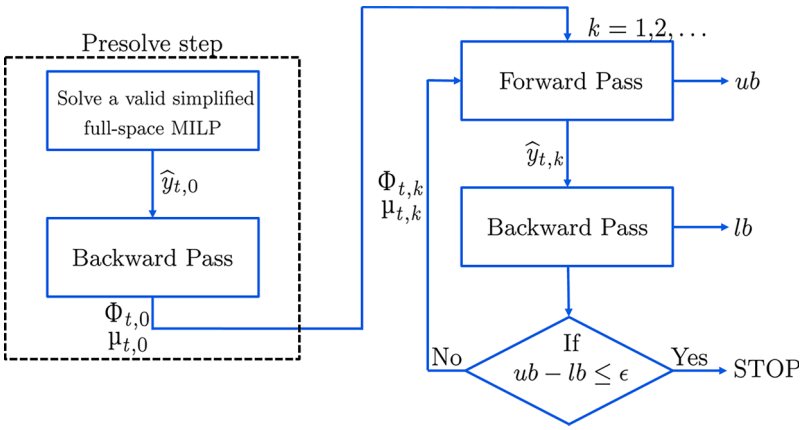


Fig. 3. DDiP with a pre-solve step (adapted from [24]).

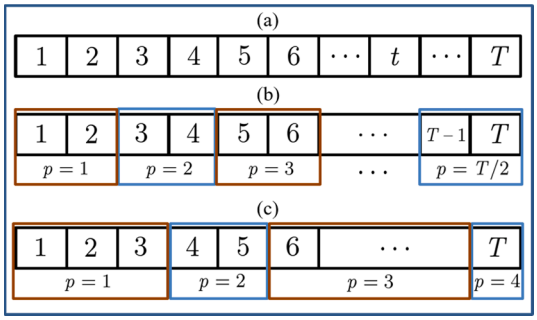


Fig. 4. Some multi-period stages for a MILP: (a) stages with one period, (b) stages with two periods, and (c) stages with non-uniform size.

a comparison of the first three types of cuts in a power system planning problem is explored. The authors show that, in general, although Benders cuts are less tight than others, the convergence of the DDiP with Benders cuts has superior performance due to the less computational effort required.

To improve the DDiP, we use an acceleration scheme to initialize the DDiP with a warm-start cost-to-go function since this has been proven to work efficiently in large-scale problems [24]. In addition, a multi-period stage scheme is proposed, with aggregation of multiple sequential single-periods in one SP. This strategy allows for solving fewer SPs and obtaining high-quality cuts in each stage; however, the computational effort to solve each SP increases. Finally, a particular overlap strategy is performed to find better solutions in each stage of the forward step. Details of these improvements are presented in the remainder of this section.

4.1. Pre-solve strategy for DDiP

The proposed DDiP algorithm has interesting properties, particularly temporal decomposition, suitable for STGS problems. However, since it is an NBD strategy, the DDiP has the same issues as the BD, such as oscillating in the upper bound and poor-quality cuts, especially in early iterations [25]. To improve the DDiP, [24] proposed an acceleration strategy that aims to initiate the algorithm with a warm-start, using an initial cost-to-go function obtained by solving an auxiliary problem. This paper will perform a similar acceleration technique denominated pre-solve step, as presented below.

The first step is to solve an auxiliary problem related to the original MILP problem (called full-space MILP, using the same notation as [24]) and use its solution to generate cuts, as long as the cuts are valid before applying the DDiP scheme. This auxiliary problem can be an aggregated version [24], a version of the original MILP problem where sequential periods are aggregated into one equivalent period, or a relaxed version of the full-space MILP. The aggregation level must be chosen carefully to generate valid cuts in the first option. Likewise, a lower level of aggregation can lead to a problem that can take almost the same time to solve as the original problem. On the other hand, a high level of aggregation can lead to a problem that can be solved quickly, as the price to obtain weaker cuts.

As an alternative for aggregation, we can solve a relaxed problem that has to be chosen in a way that can not take much computational effort. In particular, the LP relaxation of the original problem can be applied, or a MILP without a subset of hard constraints. The solution $y^*_{t,0}$ obtained in this step is used to generate cuts in a pre-backward pass, obtaining initial cuts related to variables $\varphi_{t,0}$, $\mu_{t,0}$. In this work, we used the LP relaxation of the original problem in the pre-solve step. An overview of the DDiP with this pre-solve step is presented in Fig. 3.

4.2. Multi-period strategy and overlap

Based on the main ideas presented in [26], we propose a multi-period aggregation scheme by aggregating sequential single-periods as one single SP. The main objective is to speed up the DDiP convergence by reducing the number of SPs. However, since the size of SPs will increase in this approach, the number of periods that compound the stage must be chosen carefully. Ideally, this number should not significantly impact the SP's computational performance. Since the definition of aggregation only implies that multiple adjacent periods must be used to create a multi-period stage p, different multi-period stages for the same problem can be generated, as shown in Fig. 4.

In Fig. 4, case (a) is the most frequent decomposition scheme used in standard DDiP (i.e., each SP refers to one period). Without loss of generality, we can consider that each stage contains two periods, i.e., the first block includes the first two periods and so on, resulting in the multi-period stage presented in (b). Since the stages do not need to be uniform, we can decompose the problem like (c). The multi-period stages performed will directly impact the performance of the DDiP in terms of computational effort and duality gap. In cases with uniform size aggregation, a factor K is defined as the number of periods in stage p. For instance, $K = 1$ for case (a), $K = 2$ is the case (b), and $K = T$ means the original problem without decomposition, i.e., one unique stage containing all periods.

Considering that in the forward step, the current stage's decisions do not precisely consider the constraints and costs associated with the future stages, the solutions obtained in the initial iterations of the DDiP are usually of low quality or even infeasible. The infeasibility issue can be avoided by using feasibility cuts or slack variables with high penalty costs, where the last strategy is used in this work. On the other hand, an additional strategy must be employed to avoid low-quality solutions. In

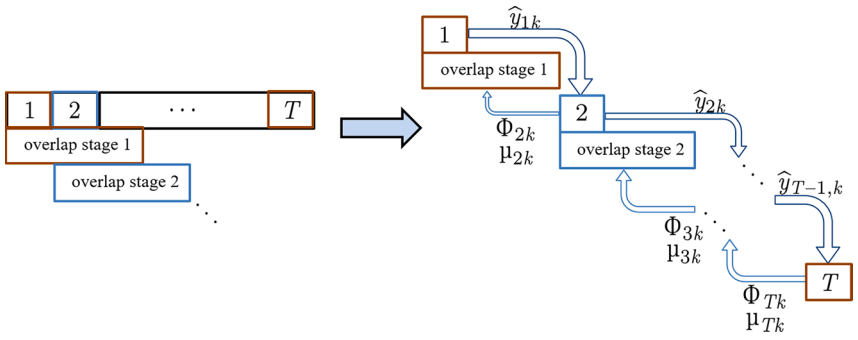


Fig. 5. Illustration of overlap decomposition scheme with no multi-period strategy.

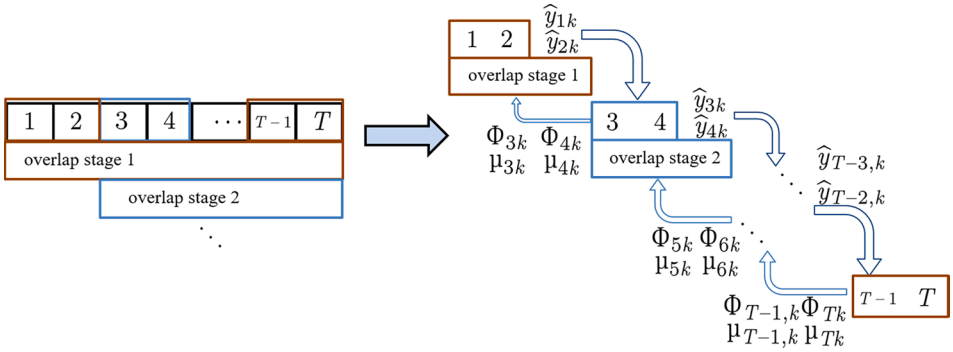


Fig. 6. Illustration of overlap decomposition scheme with multi-period strategy.

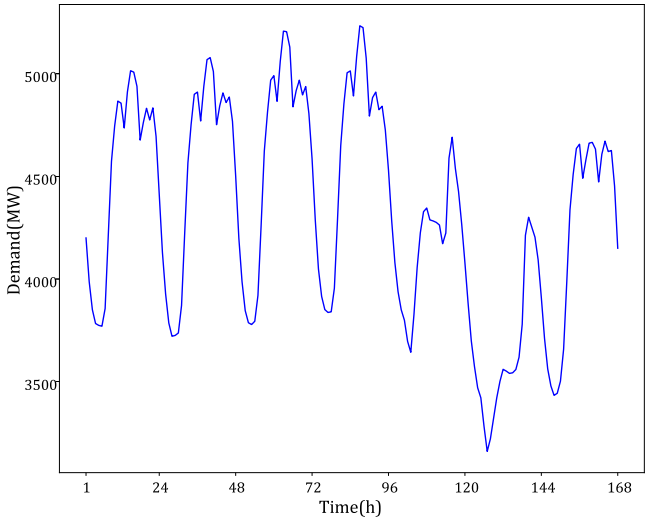


Fig. 7. Demand curve of modified IEEE-118 system.

Table 1
Number of variables and constraints in STGS problem.

Variables	Binary Variables	Constraints
196,297	23,010	366,985

this work, we employ a strategy based on the overlap Schwarz decomposition [17], aiming to improve the overall convergence of the DDiP scheme.

The overlapping strategy includes some variables and constraints of subsequent SPs in the current SP. The aim is to avoid potential nonzero slack variables obtained in the current SP due to the lack of precise information about the next SPs. Choosing which variables and constraints to be included must be carefully performed because the overlapping strategy increases the size of the optimization problem of each SP. An example of a DDiP case with no aggregation of periods and overlap can be visualized in Fig. 5. For this example, each SP contains constraints and variables of the current period and the next two SPs, except the two last SPs. The last SP does not contain overlap, and the last but one SP contains overlap constraints of the next SP, which is the last SP. A second example, presented in Fig. 6, is a case where each stage contains two periods, and the overlap contains constraints and variables of all subsequent SPs. This case demonstrates that combining the overlapping and multi-period strategies in the DDiP strategy is possible.

5. Computational experiments

This section presents computational experiments performed on the STGS problem using the DDiP scheme framework proposed in this paper.

5.1. Data and initial setup

The STGS problem is performed on a modified version of the IEEE-118 system. This system contains 118 buses, 186 transmission lines, 40 thermal plants, and 15 hydros extracted from the Brazilian power system. The average demand is 4,330 MW, and the total demand curve for 168 h is presented in Fig. 7. Also, the total installed power for hydro generation is around 64% of the system, and the remaining 36% belongs to thermal plant generation. Other details, such as initial volumes on reservoirs, inflows, water travel times, and hydropower production function can be found at https://github.com/kennyvinente/scucdata [19]. All experiments are performed on a machine with a Ryzen 9 3900X 12-core processor, 16 GB RAM, and Windows 10. The DDiP is implemented in Julia with the algebraic language JuMP [27]. Gurobi [28] is used to solve the MILP problems. The DDiP is performed considering a planning horizon of 168 h (periods).

For comparison, the STGS without any decomposition strategy is also

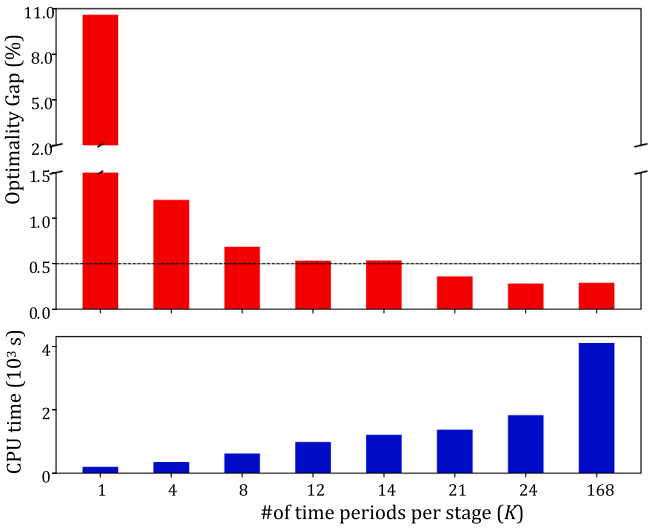


Fig. 8. Optimality gap and CPU time for s-DDiP.

Table 2
Summary of results obtained for s-DDiP.

K	Optimality gap (%)		Final result		
	2 min	10 min	CPU time (sec.)	Optimality gap (%)	Iterations
1	12.67	–	197.96	10.59	25
4	3.31	–	347.57	1.20	25
8	29.03	0.68	619.62	0.68	25
12	22.00	1.88	985.23	0.53	25
14	100	9.30	1211.35	0.53	25
21	100	3.86	1338.93	0.38	15
24	100	11.24	1831.94	0.28	13
168	–	–	4110.00	0.29	–

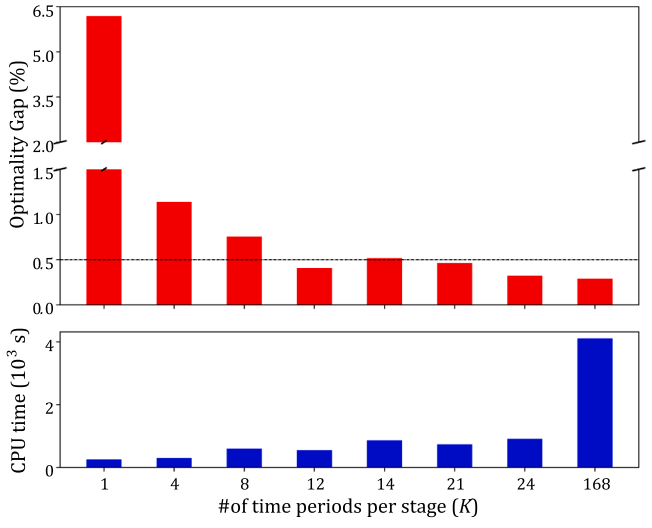


Fig. 9. Optimality gap and CPU time for a-DDiP.

Table 3
Summary of results obtained for a-DDiP.

K	Optimality gap (%)		Final result		
	2 min	10 min	CPU time (sec.)	Optimality gap (%)	Iterations
1	7.85	–	256.45	6.18	25
4	1.41	–	299.91	1.14	25
8	6.85	0.75	601.72	0.75	25
12	100	0.76	703.45	0.40	19
14	100	0.62	863.50	0.51	25
21	100	0.95	739.66	0.46	9
24	100	0.93	913.05	0.28	9
168	–	–	4110.00	0.29	–

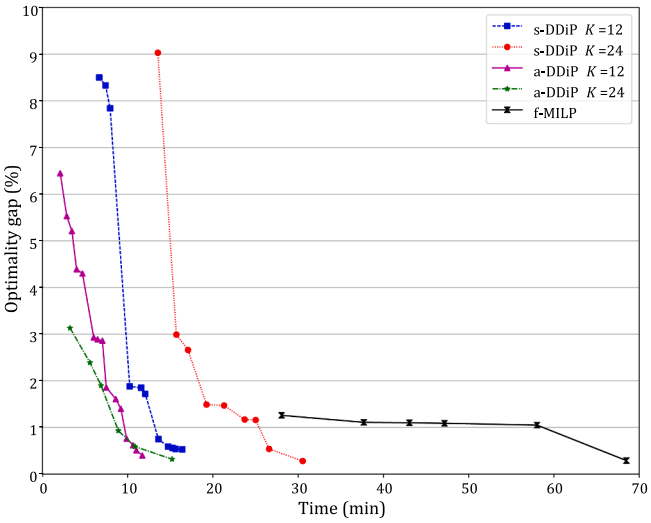


Fig. 10. DDiP performance for cases with different values of K.

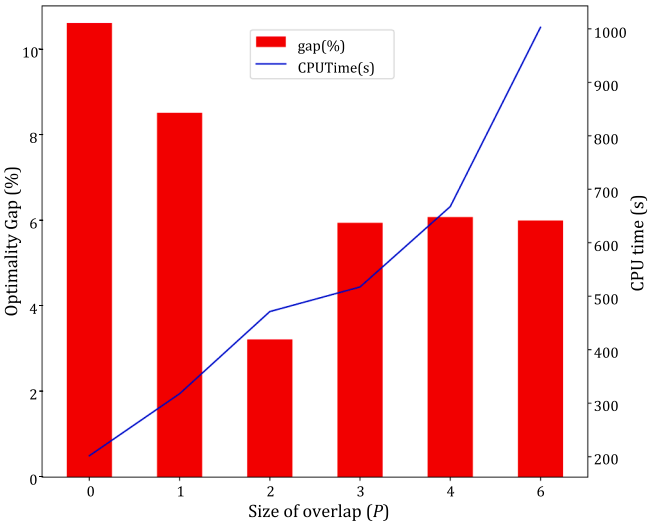


Fig. 11. Optimality gap and CPU time for s-DDiP.

solved and compared the performance in terms of computational effort and solution with several schemes of the DDiP approach. All SPs in DDiP have a limit runtime of 900 s and a 0.1% optimality gap. The DDiP presents a limit of 25 iterations and $\varepsilon = 0.5\%$. For illustrative purposes, Table 1 gives an overview of the size of the STGS problem when represented as a single MILP.

In the results, the DDiP implementation is called s-DDiP. In turn, DDiP with the pre-solve step is termed as a-DDIP, where the pre-solve strategy is performed by solving the LP relaxation of the original problem. The original MILP without any decomposition strategy is labeled f-MILP. Concerning the multi-period strategy, different values for parameter K were tested (we consider only strategies with uniform sizes), aiming to find the value that gives the best overall performance. Additionally, to implement the overlapping strategy (o-DDiP), it is necessary to analyze the STGS problem to find which constraints are

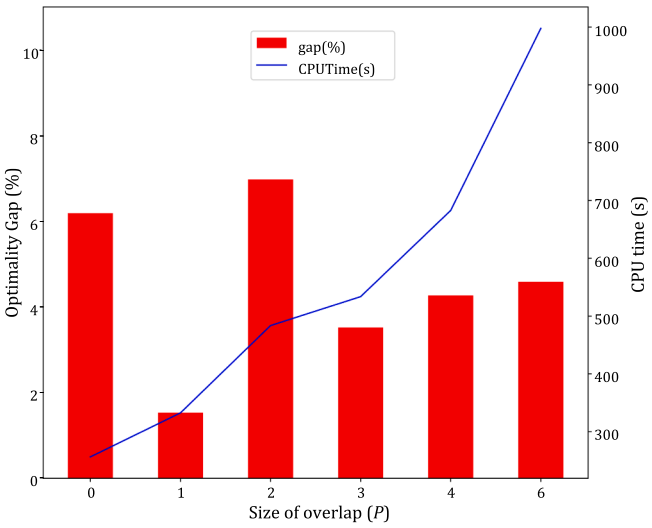


Fig. 12. Optimality gap and CPU time for a-DDiP.

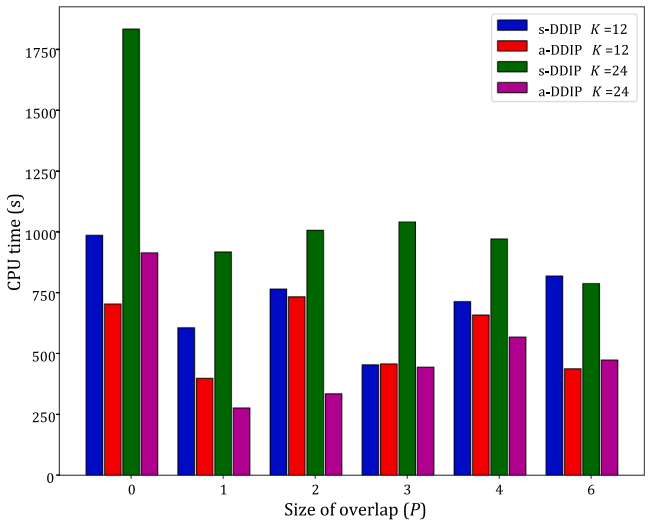


Fig. 13. Impact of overlap constraints in the DDiP for cases $K = 12$ and $K = 24$.

Table 4
Summary of results obtained for s-DDiP with $K = 1$.

P	Optimality gap (%)		Final result		
	2 min	10 min	CPU time (sec.)	Optimality gap (%)	Iterations
0	12.67	–	197.96	10.59	25
1	16.91	–	317.69	8.50	25
2	18.42	–	471.28	3.21	25
3	13.23	–	517.31	5.93	25
4	19.95	8.08	667.62	6.06	25
6	67.91	10.67	1002.81	5.98	25

Table 5
Summary of results obtained for a-DDiP with $K = 1$.

P	Optimality gap (%)		Final result		
	2 min	10 min	CPU time (sec.)	Optimality gap (%)	Iterations
0	7.85	–	256.45	6.18	25
1	11.91	–	332.58	1.52	25
2	9.58	–	483.60	6.97	25
3	11.27	–	533.66	3.51	25
4	11.44	5.16	682.82	4.26	25
6	23.35	11.97	997.70	4.58	25

more interesting to insert.

Some subsets of constraints are interesting to include in the overlap because they contain key information about the problem. In experiments, the water balance (1j), network (1l)-(1m), HPF (1h), and TUC constraints (1b)-(1g) are included in the overlapping strategy. To support this assumption, consider the water balance equation, which gives stored water value on the reservoirs at the end of the planning horizon for a specific hydro h, and supposes that the s-DDiP is used to solve the STGS problem. In the first stage (and in the subsequent others until the final stage T), the current decision on the volume of hydro h does not contain precise information about the value of water stored. Therefore, any feasible decision is valid, usually forcing the reservoir to a low-level volume. Since this is not an optimal decision due to the water use in the next stages, it is important to include this information in some way in the current stage. Thus, the water balance equation concerning the p subsequent stages, denominated p_{WB}, is included in the current stage in the overlapping strategy.

Similarly, the inclusion of network constraints of the subsequent SPs

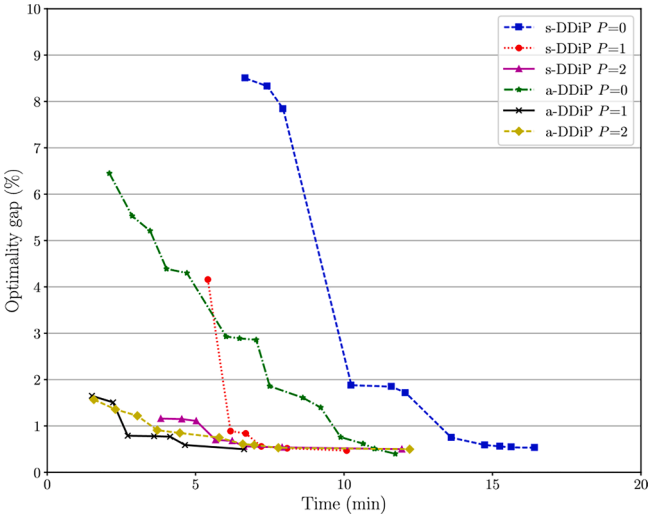


Fig. 14. DDiP performance for cases with different values of P and $K = 12$.

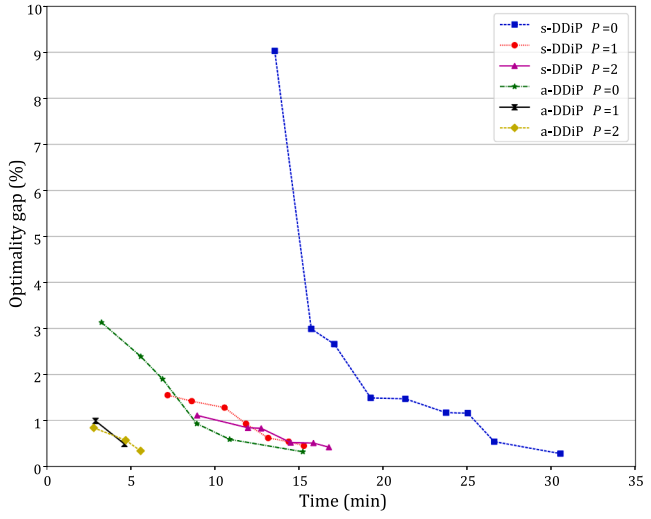


Fig. 15. DDiP performance for cases with different values of P and $K = 24$.

Table 6
Summary of results obtained of DDiP for different P values with $K = 12$ and $K = 24$.

Case	K	P	Optimality gap (%)	CPU time (s)	K	P	Optimality gap (%)	CPU time (s)
s-DDiP	12	0	0.53	985.23	24	0	0.28	1831.94
		1	0.47	605.28		1	0.45	916.62
		2	0.50	764.07		2	0.42	1006.11
		3	0.47	453.67		3	0.40	1039.53
		4	0.49	712.82		4	0.49	970.46
		6	0.50	818.41		6	0.46	786.88
a-DDiP	12	0	0.40	703.45	24	0	0.28	913.05
		1	0.50	397.94		1	0.49	276.58
		2	0.49	732.55		2	0.34	334.34
		3	0.40	457.23		3	0.32	444.09
		4	0.49	658.00		4	0.46	567.32
		6	0.48	437.30		6	0.48	472.85

Table 7
Summary of results obtained of DDiP for case with $\mathbf{Y}^{+}$.

Case	K	P	Optimality gap (%)	Time (s)	K	P	Optimality gap (%)	Time (s)
s-DDiP	12	0	0.55	819.02	24	0	0.47	1586.49
		1	0.47	353.37		1	0.45	885.38
		2	0.51	751.97		2	0.41	983.44
a-DDiP	12	0	0.53	616.63	24	0	0.49	464.70
		1	0.47	332.72		1	0.43	378.14
		2	0.38	518.71		2	0.45	375.11
f-MILP			–				1.07	1800
							0.92	3600
							0.36	5079

Table 8
Summary of results obtained of DDiP for case with $\mathbf{Y}^{-}$.

Case	K	P	Optimality gap (%)	Time (s)	K	P	Optimality gap (%)	CPU time (s)
s-DDiP	12	0	0.52	698.45	24	0	0.40	1341.42
		1	0.50	498.89		1	0.43	1012.26
		2	0.49	765.99		2	0.36	1026.43
a-DDiP	12	0	0.38	694.72	24	0	0.31	485.39
		1	0.49	341.71		1	0.38	299.16
		2	0.46	417.88		2	0.44	424.82
f-MILP			–				0.92	1800
							0.65	3600
							0.26	5849

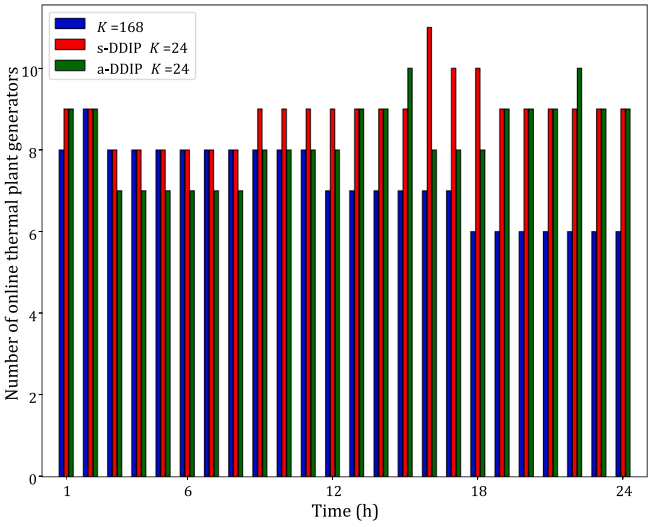


Fig. 16. Number of active thermal plants for the first 24 h of the planning horizon, considering cases $K = 168$, s-DDIP, a-DDIP with $K = 24$,$P = 0$.

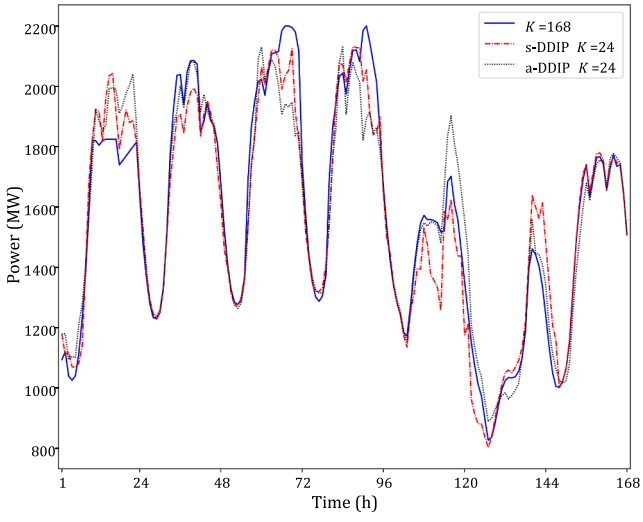


Fig. 17. Thermal plant generation for cases $K = 168$, s-DDIP, a-DDIP with $K = 24$,$P = 0$.

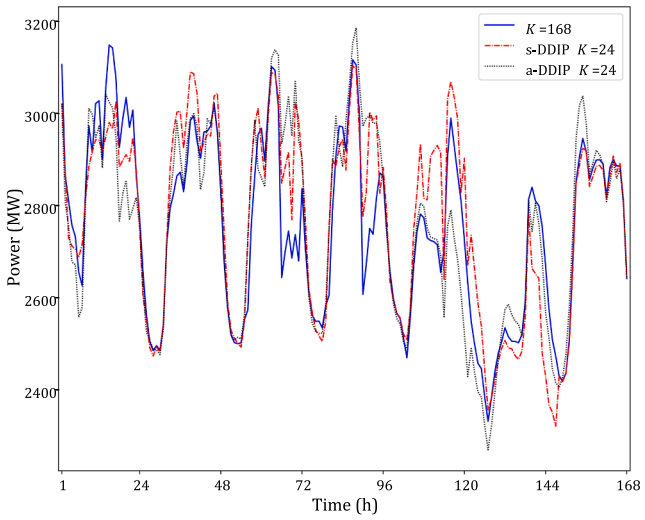


Fig. 18. Hydro generation for cases $K = 168$, s-DDIP, a-DDIP with $K = 24$,$P = 0$.

Table 9
Summation of all slack variables for simulations with different sizes of stages.

Case	*K*	*SV*
s-DDiP	1	1.50
	4	0.91
	8	1.07
	12	0.92
	14	0.90
	21	1.96
	24	1.83
a-DDiP	1	1.82
	4	1.19
	8	1.77
	12	2.00
	14	0.69
	21	2.24
	24	0.92
f-MILP	–	0.78

can increase the quality of the solution in the current stage. For example, consider the case without overlapping. Since the current problem does not have information about the demand profile for the next stages, the solutions obtained in this stage could be to turn off some thermal plant units that must be online in the next stages due to an increase in demand. The inclusion of the network constraints is expected to minimize this issue. Considering that the network constraints can increase the computational effort dramatically, and taking into account that we are using the DC network model, we represent a hydrothermal economic dispatch (ED) model of the p subsequent stages (denominated p_{DC}) for the overlapping constraints:

$$\sum_{\forall g \in G} pt_{gt} + \sum_{\forall h \in H} ph_{ht} = \sum_{\forall b \in B} \mathbf{P}_{bt}. \tag{7}$$

Considering the network constraints are included for p_{DC} subsequent stages, it is also interesting to insert the HPF and TUC constraints for the same subsequent stages. First, consider p_{TUC} (p_{HPF}) the p subsequent stages related to the TUC (HPF) constraints. It is important to note that each subset of constraints does not need to include the same subsequent stages, e.g., $p_{WB} \neq p_{DC}$. However, for simplification, in all simulations the overlapping constraints will be applied to the same subsequent stages, i.e., $p_{WB} = p_{DC} = p_{TUC} = p_{HPF} = P$. Also, the HPF constraints included in the overlap are inserted in the full version, i.e., without simplification or relaxation. The set of TUC constraints is included in the overlapping strategy via linear relaxation of (1r), which means that we are removing the integrality constraint of each binary variable.

5.2. Influence of the size of stages

A comparison of different values for the K parameter used in the DDiP applied to the STGS problem is presented in this section. The values of K chosen are 1, 4, 8, 12, 14, 21, 24 and 168 ($K = 168$ is the original MILP problem solved without decomposition, so-called f-MILP), and experiments for s-DDiP and a-DDiP (with linear relaxation in the pre-solve step) are performed. The results for the s-DDiP are presented in Fig. 8 and summarized in Table 2.

As can be seen, there is a trade-off between a small optimality gap and the CPU time required. For small values of K, the DDiP runs relatively fast if compared with simulations for large values of K, but the optimality gap does not reach the tolerance $\varepsilon = 0.5\%$. On the other hand, as the value of K increases, the optimality gap reaches the tolerance; however, with the increase in the CPU time. With 25 iterations, cases with $K = 1,4,8,12$, and 14 do not reach the optimality gap. In particular, cases $K = 12$ and 14 almost reach the 0.5% tolerance. For the remaining cases, the optimality gap was reached will less CPU when compared to f-MILP. For the a-DDiP case, the results are presented in Fig. 9 and summarized in Table 3.

The case $K = 12$ presents the best trade-off between the optimality gap and CPU time. Also, except for $K = 8$, in the final result, all simulations presented improvements in the CPU time or optimality gap compared with s-DDiP cases. Since case a-DDiP executes a pre-solve step, it is expected to increase the computational time at the beginning of the DDiP iterations. As seen for $K = 12$, the DDiP did not complete the first iteration in two minutes. On the other hand, the improvements are significant compared to Tables 2 and 3. The performance of the DDiP can be visualized in Fig. 10, where we observe that the DDiP can achieve solutions with a 1 % optimality gap faster than the problem solved without a decomposition strategy. Analyzing these experiments, case $K = 12$ reaches good performances in general, being this case used for the rest of the computational experiments of this paper.

Table 10
Summation of all slack variables for simulations with different overlap sizes.

Case	***K***	***P***	***SV***	**Case**	***K***	***P***	***SV***	**Case**	***K***	***P***	***SV***
s-DDiP	1	0	1.50	**s-DDiP**	12	0	0.92	**s-DDiP**	24	0	1.83
		1	0.78			1	0.83			1	0.82
		2	0.82			2	0.82			2	0.43
		3	1.58			3	1.39			3	0.40
		4	0.38			4	0.22			4	0.22
		6	0.36			6	0.25			6	0.21
a-DDiP	1	0	1.82	**a-DDiP**	12	0	2.00	**a-DDiP**	24	0	0.92
		1	0.77			1	1.58			1	1.43
		2	1.61			2	1.56			2	1.48
		3	1.28			3	1.01			3	1.40
		4	0.61			4	0.88			4	0.32
		6	0.56			6	0.71			6	0.17

5.3. Impact of overlapping constraints

The overlapping strategy offers new options to improve the DDiP framework, although there are many possibilities for using it. In this work, only the size of overlapping constraints is analyzed, and the set of constraints will be the same as in Section 5.1. Due to the number of combinations between the size of stages and the size of overlap constraints (*K, P*, respectively), and chosen based on the results from the previous section, only cases $K = 1$,12 and 24 are used to assess the impact of the strategy. The values of P chosen are 1, 2, 3, 4, and 6. The initial experiments will consider case $K = 1$ since it presents the worst results in the optimality gap. The results for cases s-DDiP and a-DDiP are presented in Figs. 11-12 and Tables 4-5.

As can be seen, the inclusion of overlapping constraints improves the optimality gap at the price of increasing the computational effort, which is expected since there is an increase in each SP involved in the DDiP. For the case s-DDiP, $P = 2$ achieved the lowest optimality gap, and for case a-DDiP, $P = 1$ returns the best optimality gap, due to the limit of iterations. For the remaining experiments concerning the overlap, the results are presented only in the computational effort once the tolerance of the optimality gap is always achieved. The results for s-DDiP and a-DDiP considering cases $K = 12$ and 24 are presented in Figs. 13-15 and summarized in Table 6.

In general, the overlapping constraints significantly impact the overall performance of DDiP. Particularly, for case $K = 24$ with $P = 1$ reduces the computational effort by approximately 70%, showing the potential of this approach.

Finally, to consolidate the results presented, we performed other experiments by changing the original inflows $\mathbf{Y}$. Specifically, we created two other computational instances, where the first is referred as $\mathbf{Y}^{+}$, and the inflows used are $1.6 \times \mathbf{Y}$. The second one is referred to as $\mathbf{Y}^{-} = 0.4 \times \mathbf{Y}$. For these new cases, we performed the s-DDiP and a-DDiP with $K = \{12,24\}$, and $P = \{0,1,2\}$. The results obtained are presented in Tables 7-8.

Although the DDiP did not reach the 0.5% tolerance in some cases, the CPU time is significantly lower when compared with the f-MILP. Therefore, these results show the potential of the strategies presented in this paper when applied to STGS problems.

5.4. Assess the primal solutions

This section compares the primal solutions obtained for the different experiments involving the DDiP strategy for simulations with inflows equal to $\mathbf{Y}$. Only the first 24 hours are used for this analysis since a rolling horizon strategy in the Brazilian STGS problem is performed. In other words, despite running a 168 h STGS problem, the independent system operator is interested only in the decisions obtained for the first 24 hours (day-ahead dispatch). The number of active thermal plant plants for the first 24 hours with $K = 24$, $P = 0$, and $K = 168$ is presented in Fig. 16. As can be seen, there are differences between the number of online thermal plants, especially in the last hours.

The total thermal plant generation for simulations with $K = 24$, $P = 0$, and $K = 168$ is presented in Fig. 17. Despite the differences in the total amount of thermal generation between each case, it is possible to observe the same generation patterns (generation curves) in all simulations.

Concerning the total hydro generation, the results from simulations with K = 24, P = 0, and K = 168 are presented in Fig. 18. Like the thermal generation, the hydro generation presents some differences between each simulation, but the generation patterns are similar.

These results make it possible to observe differences in the generation obtained from these experiments and the thermal plant's on/off status. Although this is expected, due to the optimality gap imposed, the power dispatches generally follow the same pattern, and the differences are not significantly high. As mentioned, we use slack variables instead of feasibility cuts. Tables 9-10 present the summation of all slack variables (*SV*) obtained from each simulation to clarify the effects of the slack variables. These variables are included in TUC, WB, and power balance (PB) constraints. The penalty term (P_0) associated with slack variables in TUC and PB constraints is ten times greater than the most expensive unitary variable cost. In contrast, the penalty term in WB constraints is $P_0 \times 10^6$.

Although the values of the slack variables are not zero for all simulations (including the original problem solved without a decomposition strategy), the values of the slack variables are low enough and have a negligible impact on the solution quality. Therefore, we conclude that DDiP is a promising strategy that can be used to solve STGS problems.

5.5. Conclusions and future work

In this paper, an STGS problem is solved using a temporal decomposition strategy provided by DDiP. The classical DDiP is based on nested Benders decomposition, and a pre-solve step is initially applied. A study on the size of stages is presented, aiming to find the best trade-off between the computational effort involved in solving each SP and the convergence of DDiP. Next, a novel contribution constituted on overlap constraints is presented, and criteria for choosing the type of constraints that will belong to the overlap for the STGS problem are discussed. The experiments are performed on a modified version of the IEEE-118 bus system to validate the different types of DDiP strategies. The results show that the DDiP can obtain solutions with an optimality gap of less than 0.5% way faster than solving the STGS without decomposition, showing the potential of the DDiP, especially on large-size problems. The impact of period aggregation is significant and shows the potential approach in other optimization problems with a similar mathematical structure. For the problem under analysis, there is an optimal value of the aggregation parameter that leads to the minimal computational effort. Regarding the overlap constraints, our results show that this approach is promising and can be applied to several other problems employing the BD strategy. A natural extension of this study is to evaluate different types of pre-solve strategies, specifically how to use the decomposition scheme of the DDiP to generate initial cuts. Another extension is to use this strategy with off-the-shelf solvers, including free mixed-integer optimization solvers.

CRediT authorship contribution statement

Kenny Vinente dos Santos: Conceptualization, Methodology, Visualization, Writing – original draft. **Bruno Colonetti:** Conceptualization, Methodology, Writing – review & editing. **Erlon Cristian Finardi:** Conceptualization, Visualization, Investigation, Supervision, Writing – review & editing, Project administration, Funding acquisition. **Victor M. Zavala:** Conceptualization, Visualization, Investigation, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil) and Norte Energia, via R&D project registered with the PD-07427-0318/2018 code in ANEEL (Agência Nacional de Energia Elétrica - Brazil), for their financial support.

References

[1] Finardi EC, Silva EL. Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. IEEE Trans Power Syst 2006;21(2):835–44.

[2] Borthetti A, Frangioni A, Lacalandra F, Nucci CA. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. IEEE Trans Power Syst 2003;18(1):313–23.
[3] Geoffrion AM. Generalized benders decomposition. J Optim Theory Appl 1972;10 (4):237–60.
[4] Fu M, Shahidehpour M, Li Z. Security-constrained unit commitment with ac constraints*. IEEE Trans Power Syst 2005;20(3):1538–50.
[5] Amjady N, Reza Ansari M. Hydrothermal unit commitment with ac constraints by a new solution method based on benders decomposition. Energy Convers Manage 2013;65:57–65.
[6] Alemany J, Magnago F. Benders decomposition applied to security constrained unit commitment: Initialization of the algorithm. Int J Electr Power Energy Syst 2015; 66:53–66.
[7] Wu L, Shahidehpour M. Accelerating the benders decomposition for network-constrained unit commitment problems. Energy Syst 2010;1(3):339–76.
[8] Beheshti Asl N, MirHassani SA. Accelerating benders decomposition: Multiple cuts via multiple solutions. J Comb Optim 2019;37(3):806–26.
[9] Zaourar S, Malick J. Quadratic stabilization of benders decomposition, hal-01181273 (2014).
[10] Ansari MR, Amjady N, Vatani B. Stochastic security-constrained hydrothermal unit commitment considering uncertainty of load forecast, inflows to reservoirs and unavailability of units by a new hybrid decomposition strategy. IET Gener Transm Distrib 2014;8(12):1900–15.
[11] Colonetti B, Finardi EC. Stochastic hydrothermal unit commitment models via stabilized benders decomposition. Electr Eng 2021;103(4):2197–211.
[12] Pereira MVF, Pinto LMVG. Stochastic optimization of a multireservoir hydroelectric system: A decomposition approach. Water Resour Res 1985;21(6): 779–92.
[13] Pereira MVF, Pinto LMVG. Multistage stochastic optimization applied to energy planning. Math Program 1991;52(1–3):359–75.
[14] Zou J, Ahmed S, Sun XA. Multistage stochastic unit commitment using stochastic dual dynamic integer programming. IEEE Trans Power Syst 2019;34(3):1814–23.
[15] Lara CL, Siirola JD, Grossman IE. Eletric power infrasctructure planning under uncertainty: Stochastic dual dynamic integer programming (sddip) and parallelization scheme. Optim Eng 2020;21(4):1243–81.
[16] Kumar R, Wenzel MJ, ElBsat MN, Risbeck MJ, Drees KH, Zavala VM. Dual dynamic integer programming for multi-scale mixedinteger mpc, arXiv:2007.10149 [math], online; accessed 24 August 2021 (2020).
[17] Na S, Shin S, Anitescu M, Zavala VM. On the convergence of overlapping schwarz decomposition for nonlinear optimal control, arXiv:2005.06674 [cs, math], online; accessed 24 August 2021 (2021).
[18] Pedrini R, Finardi EC. Long-term generation scheduling: A tutorial on the practical aspects of the problem solution. J Control Autom Electr Syst 2022;33:806–21.
[19] Santos KV, Finardi EC. Piecewise linear approximations fo hydropower production function applied on the hydrothermal unit commitment problem. Int J Electr Power Energy Syst 2022;135:107464.
[20] Diniz AL, Maceira MEP. A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects. IEEE Trans on Power Systems 2008;23(3):1298–308.
[21] Birge JR. Decomposition and partitioning methods for multistage stochastic linear programs. Oper Res 1985;33(5):989–1007.
[22] Shahidehpour M, Ding T, Ming Q, Huang C, Wang Z, Du P. Multiperiod active distribution network planning using multi-stage stochastic programming and nested decomposition by sddip. IEEE Trans on Power Systems 2021;36(3): 2281–92.
[23] Zou J, Ahmed S, Sun XA. Stochastic dual dynamic integer programming. Math Program 2019;175(1–2):461–502.
[24] Lara CL, Mallapragada DS, Papageorgiou DJ, Grossman IE. Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm. Eur J Oper Res 2018;271(3):1037–54.
[25] Rahmaniani R, Crainic TG, Gendreau M, Rei W. The benders decomposition algorithm: A literature review. Eur J Oper Res 2016;259(3):801–17.
[26] Santos TN, Diniz AL. A new multiperiod stage definition for the multistage benders decomposition approach applied to hydrothermal scheduling. IEEE Trans Power Syst 2009;24(3):1383–92.
[27] Dunning I, Huchette J, Lubin M. Jump: A modeling language for mathematical optimization. SIAM Rev 2017;59(2):295–320.
[28] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual (2022). URL https://www.gurobi.com.