A stabilised Benders decomposition with adaptive oracles applied to investment planning of multi-region power systems with short-term and long-term uncertainty

Hongyu Zhang\textsuperscript{a,*}, Nicolò Mazzi\textsuperscript{c}, Ken McKinnon\textsuperscript{b}, Rodrigo Garcia Nava\textsuperscript{b}, Asgeir Tomasgard\textsuperscript{a}

\textsuperscript{a}Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Høgskoleringen 1, 7491, Trondheim, Norway  
\textsuperscript{b}School of Mathematics, University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, United Kingdom  
\textsuperscript{c}aHead-Research, Corso Svizzera 185, Torino, 10149, Italy

Abstract

Benders decomposition with adaptive oracles was proposed to solve large-scale optimisation problems with a column bounded block-diagonal structure, where subproblems differ on the right-hand side and cost coefficients. Adaptive Benders reduces computational effort significantly by iteratively building inexact cutting planes and valid upper and lower bounds. However, Adaptive Benders and standard Benders may suffer severe oscillation when solving a multi-region investment planning problem. Therefore, we propose stabilising Adaptive Benders with the level set method and adaptively selecting the subproblems to solve per iteration for more accurate information. Furthermore, we propose a dynamic level set method to improve the robustness of stabilised Adaptive Benders by adjusting the level set per iteration. We compare stabilised Adaptive Benders with the unstabilised versions of Adaptive Benders with one subproblem solved per iteration and standard Benders on a multi-region long-term power system investment planning problem with short-term and long-term uncertainty. The problem is formulated as multi-horizon stochastic programming. Four algorithms were implemented to solve linear programming with up to 1 billion variables and 4.5 billion constraints. The computational results show that: a) for a 1.00% convergence tolerance, the proposed stabilised method is up to 113.7 times faster than standard Benders and 2.14 times faster than unstabilised Adaptive Benders; b) for a 0.10% convergence tolerance, the proposed stabilised method is up to 45.5 times faster than standard Benders and unstabilised Adaptive Benders cannot solve the largest instance to convergence tolerance due to severe oscillation and c) dynamic level set method makes stabilisation more robust.

Keywords: Large scale optimisation, Multi-stage stochastic programming, Multi-horizon stochastic programming, Stabilised Benders decomposition with adaptive oracles, Level set method
1. Introduction

Power system infrastructure planning is crucial during the energy transition towards zero emission by 2050. Optimisation models are widely used for the investment and operational planning of systems. To gain enough environmental and economic insights from such models, sometimes a large-scale problem needs to be modelled, such as (Li et al., 2022; Zhang et al., 2022b). An investment planning problem can involve many technologies and regions and span over decades with multiple investment periods (Conejo et al., 2016). This can lead to a large-scale optimisation problem that is intractable. Furthermore, investment planning of a power system often faces uncertainty from two time horizons (Kaut et al., 2014; Lara et al., 2020): a) the uncertainty from the operational time horizon, such as the availability of renewable energy. The operational uncertainty becomes even more crucial for a system with higher penetration of intermittent renewable energy, and b) the uncertainty from the strategic time horizon, e.g., CO$_2$ tax and CO$_2$ budget. Stochastic programming is often used to model uncertainty. However, including uncertainty from both time horizons using multi-stage stochastic programming may lead to a large scenario tree and an intractable model. Most studies on power system investment planning in a multi-horizon framework only consider short-term uncertainty, such as (Backe et al., 2022), partly because of the tractability of the problem. However, short-term and long-term uncertainty can play a decisive role in investment planning. Although there are examples including short-term and long-term uncertainty in a multi-horizon model (Hellemo et al., 2013), the computational difficulty is not sufficiently addressed. Therefore, we aim to address the computational difficulties of long-term planning problems with short-term and long-term uncertainty. One possible way is to reduce the problem size by using a different modelling approach called multi-horizon stochastic programming (Kaut et al., 2014). Although multi-horizon stochastic programming can reduce the scenario tree significantly, it is essentially multi-stage stochastic programming once both short-term and long-term uncertainty is added and can be intractable when the problem gets large. Another way is to develop an algorithm that can efficiently solve a class of large-scale optimisation problems, such as progressive hedging type method (Munoz & Watson, 2015). Although some decomposition algorithms have been proposed to tackle the computational difficulty and claimed to be capable of solving problems with short-term and long-term uncertainty (Downward et al., 2020), the algorithms were only demonstrated to solve a problem with only short-term (Munoz et al., 2016) or long-term uncertainty (Singh et al., 2009). Therefore, this paper proposes an algorithm for solving such problems efficiently and demonstrates the algorithm for a long-term investment planning problem with short-term and long-term uncertainty.

In this paper, we propose an algorithm to efficiently solve large-scale optimisation problems that exhibit a column bounded block-diagonal structure, where Subproblems (SPs) differ on the right-hand side and cost coefficients. Such problems can be formulated as a full Master Problem (MP) (1),

$$\min_{x \in \mathcal{X}} f(x) + \sum_{i \in \mathcal{I}} \pi_i g(x_i, c_i),$$

(1)
where \( f(x) = \sum_{i \in I} \pi_i c_i x_i \) and the function \( g(x_i, c_i) \) is the optimal solution of the linear programming SP,
\[
g(x_i, c_i) := \min_{y_i \in Y} \{ c_i^\top C y_i | Ay_i \leq B x_i \}. \tag{2}
\]
The set of decision nodes is given by \( I \). The \( x_i \) are subvectors of \( x \). The \( y_i \) is the decision variables of SP \( i \) that is in the convex set \( Y \). The \( \pi_i \) are non-negative constants. The coefficient matrices \( A, B, \) and \( C \) are the same in SPs, and \( x_i \) and \( c_i \) are independent of short-term uncertainty. The decisions made in \( f(x) \) are passed to SPs as right-hand side parameters. Multi-stage stochastic programming problems can be formulated as (1) and (2), but the algorithm can be applied to any optimisation problems with the same structure.

Solving this problem directly can be computationally expensive. However, when \( g(x_i, c_i) \) is convex and decreasing w.r.t. \( x_i \), and concave and increasing w.r.t. \( c_i \), one can exploit these properties to efficiently solve the problem (Mazzi et al., 2020). (Mazzi et al., 2020) proposed two inexact oracles that approximate \( g(x_i, c_i) \) from below and above adaptively, and by using these, one can avoid solving all SPs every iteration to reduce the computational cost compared with standard Benders decomposition. The method is called Adaptive Benders. Like other Benders-type decomposition, Adaptive Benders suffers from oscillation and results in slower performance. The performance of the Adaptive Benders method was tested on a UK power system planning problem (Mazzi et al., 2020). However, it is a single-region investment planning problem, and we found that the algorithm becomes slower after introducing more regions into the problem. This issue needs to be addressed because a power system investment planning problem normally involves multiple regions connected via transmission lines (Gacitua et al., 2018). Therefore, this paper develops a stabilised Benders decomposition with adaptive oracles. We call the improved method stabilised Adaptive Benders in the rest of the paper. The stabilised Adaptive Benders consist of a level set method stabilisation and a mechanism that dynamically selects the SPs to solve at every iteration.

We use the algorithm to solve an investment planning problem with short-term and long-term uncertainty formulated as multi-horizon stochastic programming. In such a problem, \( x \) represents investment decisions with corresponding investment cost \( f(x) \). The investments affect a set \( I \) of investment periods, and \( x_i \) is the subvector of \( x \) that represents the investments that affect period \( i \), \( c_i \) specifies the operational costs, \( y_i \) defines the operational decisions at period \( i \), and \( g(x_i, c_i) \) gives the optimal operational cost. The \( \pi_i \) is the probability associated with decision node \( i \).

The contributions of this paper are: (1) we develop a level set stabilised Adaptive Benders decomposition to address the oscillation issue and analyse the tuning of parameters; and (2) we test the proposed method on a multi-horizon stochastic programming model with short-term and long-term uncertainty with up to 1 billion variables and 4.5 billion constraints. The results show that it is up to 113.7 times faster than standard Benders and 2.14 times faster than the Adaptive Benders for a 1.00% convergence, and up to 45.5 times faster than standard Benders for a 0.10% convergence and the unstabilised Adaptive Benders cannot solve the largest instance to convergence tolerance; and (3) dynamic level set stabilisation increases the robustness of the proposed method and can be up to 33.5 times faster for 1.00% convergence and 25.4 times faster for 0.10% convergence.
compared with standard level set method stabilisation with poor parameter choices.

The outline of the paper is as follows: Section 2 introduces the background knowledge regarding stochastic programming, multi-horizon modelling approach, Benders decomposition and stabilisation. Section 3 introduces the level set stabilisation method. Section 4 gives the problem description. Section 5 presents the model for the case study. Section 6 states the computational results and numerical analysis. Section 7 discusses the implications of the method and results and summaries the limitations of the research. Section 8 concludes the paper and suggests further research.

2. Literature review

This paper proposes a Benders-type algorithm to solve large-scale optimisation problems. In the following, we present the background knowledge of stochastic programming, multi-horizon modelling approach, standard Benders decomposition, Adaptive Benders decomposition, and level set method stabilisation.

2.1. Stochastic programming

Stochastic programming is the part of mathematical programming and operations research that studies how to incorporate uncertainty into decision problems (King & Wallace, 2012). It is one of the most popular methods of dealing with uncertainties in energy system planning (Birge & Louveaux, 2011). The electricity system in regulated markets is a well-developed area for using stochastic programming in energy (Wallace & Fleten, 2003; Powell & Meisel, 2016). However, stochastic programming is also exploited in natural gas systems (Fodstad et al., 2016), offshore oil and gas infrastructure planning (Gupta & Grossmann, 2014), and hydrogen network (Galan et al., 2019).

Two-stage stochastic programming (Boffino et al., 2019), multi-stage stochastic programming (Pereira & Pinto, 1991), stochastic mixed-integer programming (Salo et al., 2022; Lara et al., 2020; Munoz et al., 2016), and stochastic nonlinear programming (Li, 2021) are all used in energy system research. In (Lara et al., 2020), a multi-stage stochastic mixed-integer programming formulation was developed to optimise electricity infrastructure planning over multiple years. In order to solve a large-scale model, they decomposed and solved the problem using parallelised stochastic dual dynamic integer programming.

2.2. Multi-horizon stochastic programming

In traditional multi-stage stochastic programming, uncertainty from operational and strategic time horizons can lead to a large scenario tree, thus, an intractable planning model. The multi-horizon modelling approach was proposed as an alternative formulation that reduces the model size significantly (Kaut et al., 2014). One can have a much smaller model by disconnecting operational nodes between successive planning stages and embedding them into their respective strategic nodes. The resulted model is called multi-horizon stochastic programming. However, the multi-horizon formulation is an approximation to multi-stage stochastic programming unless two requirements
are met (Kaut et al., 2014): a) strategic and operational uncertainties are independent, and the strategic decisions must not depend on any particular operational decisions; and b) the operational decisions in the last operational period in a stage do not affect the system operation in the first operational period in the next stage. An illustration of multi-horizon stochastic programming with short-term and long-term uncertainty is shown in Figure 1.

2.3. Benders decomposition

Benders decomposition was firstly developed in (Benders, 1962) and has been successfully applied to a wide range of difficult optimisation problems (Rahmaniani et al., 2017). Benders decomposition exploits the block diagonal structure of (1) and creates outer linearisation. This method has been extended in stochastic programming to take care of feasibility questions and is known as the L-shaped method (Slyke, 1969). In Benders decomposition, a sequence of approximations is solved, and two types of constraints are added after each solve: feasibility cuts (enforcing the feasibility of (1)) and optimality cuts (linear approximations to (1) on its domain of finiteness) (Birge & Louveaux, 2011).

In standard Benders decomposition, a relaxation of the MP is solved. At iteration $j$, the Relaxed Master Problem (RMP) is

$$\min_{x \in \mathcal{X}, \beta} f(x) + \sum_{i \in I} \pi_i \beta_i$$  \hspace{1cm} (3a)

s.t. $\beta_i \geq \theta + \lambda^T(x_i - x)$, \hspace{1cm} $(x, \theta, \lambda) \in F_{i(j-1)}, i \in I$,  \hspace{1cm} (3b)

where $F_{i(j-1)}$ is the set of cuts associated with SP $i$ up to iteration $j - 1$. To perform Benders
Algorithm 1 Standard Benders

1: choose $\epsilon$ (convergence tolerance), $\beta$ (initial lower bound for $\beta_i$), $U_0^* := M$ (initial upper bound), set $j := 0$, $F_0 := \{(\beta_0, 0, 0)\}$ for each $i \in \mathcal{I}$;

2: repeat
3: \quad set $j := j + 1$;
4: \quad solve RMP and obtain $\beta_{ij}$ and $x_{j,RMP}^*$; set $L_j^* := f(x_{j,RMP}^*) + \sum_{i \in \mathcal{I}} \pi_i \beta_{ij}$;
5: \quad for $i \in \mathcal{I}$ do
6: \quad \quad solve $\text{SP}_i$ at $(x_{j,RMP}^*, c_i)$ and obtain $\theta_{ij}$ and $\lambda_{ij}$;
7: \quad end for
8: \quad set $U_j^* := \min(U_{j-1}^*, f(x_{j,RMP}^*) + \sum_{i \in \mathcal{I}} \pi_i \theta_{ij})$;
9: \quad for $i \in \mathcal{I}$ do
10: \quad \quad set $F_{ij} := F_{i(j-1)} \cup \{(x_{j,RMP}^*, \theta_{ij}, \lambda_{ij})\}$;
11: \quad end for
12: until $U_j^* - L_j^* \leq \epsilon$.

decomposition, we firstly solve the RMP to obtain optimal solution $x_j$. Then we pass a subvector of $x_j$, $x_{ij}$, to the SP $i$ and call an oracle that gives the optimal value of the SP, $\theta_{ij}$, and a subgradient, $\lambda_{ij}$, w.r.t $x_{ij}$. Finally, a new cutting plane is added to $F_{i(j-1)}$ which gives $F_{ij} := F_{i(j-1)} \cup \{x_{ij}, \theta_{ij}, \lambda_{ij}\}$
The standard Benders decomposition is presented in Algorithm 1.

2.4. Benders decomposition with adaptive oracles

An investment planning problem formulated as (1) and (2) can easily get intractable once we have a large number of decision nodes that are caused by a long planning horizon or inclusion of multiple uncertainties. In stochastic programming, it refers to the curse of dimensionality that arises from the number of nodes in a scenario tree as the number of scenarios increases (Powell, 2011). Benders-type algorithms iteratively approximate the SP cost function through a set of cutting planes. However, the acquisition of the cutting planes needs all SPs to be solved at every iteration. Thus, Benders decomposition may get slow severely when there are many SPs. Therefore, research on making Benders decomposition more efficient was conducted (Skar et al., 2014; Zakeri et al., 2000; Baena et al., 2020). One approach is to exploit the SP structure to avoid solving all SPs but still get a valid cutting plane each iteration.

In (Mazzi et al., 2020), two adaptive oracles were proposed in order to approximate the unsolved SPs objective function using the solution from the solved ones, which improves the efficiency significantly. One adaptive oracle generates inexact but valid cutting planes, and the other adaptive oracle gives a valid upper bound of the actual optimal value.

At iteration $j$, the RMP is

$$\min_{x \in X, \beta} f(x) + \sum_{i \in \mathcal{I}} \pi_i \beta_i$$

s.t. $\beta_i \geq \theta + \Delta^T (x_i - x)$, \hspace{1cm} $(x, \theta, \Delta) \in F_{i(j-1)}, i \in \mathcal{I},$ \hspace{1cm} (4a)

where $F_{i(j-1)}$ is the supporting hyperplane from solution of SP $i$ at iteration $j$. At iteration $j$, the
lower bound is denoted as \( L_j^* = f(x) + \sum_{i \in I} \beta_i \) and the upper bound is \( U_j^* = f(x) + \sum_{i \in I} \pi_i \theta_i \). \( \beta, \Delta \) are obtained by solving the lower bound oracle and \( \theta \) is obtained by computing the upper bound oracle. Their algorithm requires SP to be convex and decreasing w.r.t. \( x_i \) and concave and increasing w.r.t. \( c_i \). In order to apply the algorithm, the SPs need to be always feasible, which can be achieved by introducing penalty terms in SPs.

3. Level set method stabilisation

In this paper, we stabilise the algorithm in (Mazzi et al., 2020) using the level set method. The level set method was introduced in (Lemarechal et al., 1995). It was then used to regularise standard Benders decomposition (Fabian, 2000).

We now present the stabilisation step and its coordination with Adaptive Benders. At each iteration \( j \), the Level Method Problem (LMP) for stabilisation can be formulated as

\[
\begin{align*}
\min_{x \in X, \beta} & \quad \|x - x_{j-1}\|^2 \\
\text{s.t.} & \quad \beta_i \geq \theta + \Delta^T (x_i - x), \quad (x, \theta, \Delta) \in F_i(j-1), i \in I \\
f(x) + \sum_{i \in I} \pi_i \beta_i \leq L_j^* + \gamma \Delta_j. 
\end{align*}
\]

Constraint (5c) is the level set, \( L_j^* + \gamma \Delta_j \) is the target that is denoted as \( T_j \), and \( \Delta_j = U_{j-1}^* - L_j^* \). The stabilisation factor, \( \gamma \), is interpreted as the ratio of the achieved improvement to the predicted improvement between successive iterations. The lower bound is denoted by \( L_j^* \) is the lower bound. By introducing LMP, we restrict expected improvement between iterations, thus restricting the distance moved between iterations. LMP is essentially a RMP but with an objective that minimises distance and an extra constraint for the level set. A graphical interpretation of the level method stabilised Adaptive Benders decomposition is presented in Figure 2.

Figure 2 illustrates how the algorithm works for iteration \( j \). At the beginning of iteration \( j \), we have the cuts that have been added in all previous iterations, the upper bound \( U_{j-1}^* \) and the lower approximation of the function value (blue dot). The black dot represents the function value at that point which is unknown unless all SPs are solved exactly. Based on the cuts, we solve the RMP and get a lower bound \( L_j^* \) (blue square). If there is no stabilisation, we would move to the point \( x_j^{RMP} \). When there is stabilisation, the moving area is restricted by the target \( T_j \), and we move to the closest point to \( x_j^{LMP} \) that is below the target. At point \( x_j^{LMP} \), we evaluate one or more SPs and get a new upper bound \( U_j^* \) (red dot) and add cuts.

3.1. Stabilised Benders decomposition with adaptive oracles algorithm

In this section, we present the stabilised Benders decomposition with the adaptive oracles algorithm, shown in Algorithm 2.

Although an interesting feature of the level method compared with other bundle-type methods is that \( \gamma \) is fixed (Zverovich et al., 2012), it may be beneficial to adjust it based on the progress.
Therefore, in addition to using fixed stabilisation factor $\gamma$, we also explore adjusting the stabilisation factor for a potentially more robust algorithm. There are several ways to adjust stabilisation dynamically, and we choose a method analogous to what is used to adjust trust regions. The trust region method uses a local approximation of the function to be minimised and optimised within the trust region. The trust region size is updated throughout the iterations. In the trust region method, one adjusts the trust region according to the ratio of the actual decrease to the predicted decrease (Fletcher, 2000). Inspired by (Fletcher, 2000), we adjust the level set based on the ratio of the actual improvement to the expected improvement. At iteration $j$, we define the ratio

$$r := \frac{L_{j-1}^{LBO} - L_j^{LBO}}{L_{j-1}^{LBO} - T_j},$$

Figure 2: An illustrative example from iteration $j-1$ to iteration $j$. 

(a) The beginning of iteration $j$

(b) Solve the RMP and get a lower bound

(c) Set the target and solve the LMP

(d) Get an upper bound and add a cut
Algorithm 2 Level set method stabilised Benders decomposition with adaptive oracles

1: choose $\epsilon$ (convergence tolerance), $\gamma$ (stabilisation factor), $\beta$ (initial lower bound $\beta_i$), $U_0^* := M$ (initial upper bound);
2: set $j := 0$, $F_0 := \{ (\beta_0, 0, 0) \}$ for each $i \in \mathcal{I}$;
3: solve $SP$ at the special point $(\bar{x}, \bar{c})$ and obtain $\theta$, $\lambda$ and $\phi$; set $S := \{(\bar{x}, \bar{c}, \theta, \lambda, \phi)\}$;
4: repeat
5:  set $j := j + 1$;
6:  solve $RMP$ and obtain $\beta_{ij}$ and $x_{ij}^{RMP}$; set $L_j^* := f(x_{ij}^{RMP}) + \sum_{i \in \mathcal{I}} \pi_i \beta_{ij}$;
7:  set $x_{ij}^{Ref} := x_{ij}^{RMP}$ (when $j = 1$);
8:  set $LMP$ target: $L_j^* + \gamma(U_{j-1}^* - L_j^*)$;
9:  solve $LMP$ and obtain $x_{ij}^{LMP};$
10: for $i \in \mathcal{I}$ do
11:     | call adaptive oracles at $(x_{ij}^{LMP}, c_i)$ and obtain $\theta_{ij}$, $\bar{\theta}_{ij}$, $\bar{\phi}_{ij}$ and $\Lambda_{ij}$;
12: end for
13: set $n := 0$;
14: repeat
15:  set $n := n + 1$;
16:  choose $SP_i$ in $\mathcal{I}$ that has the largest gap;
17:  solve $SP_i$ at $(x_{ij}^{LMP}, c_i)$ exactly and obtain $\theta_{ij}$, $\lambda_{ij}$, $\phi_{ij}$;
18:  set $S := S \cup \{(x_{ij}^{LMP}, c_i, \theta_{ij}, \lambda_{ij}, \phi_{ij})\}$;
19: for $i \in \mathcal{I}$ do
20:     | set $F_{ij} := F_{i(j-1)} \cup \{(x_{ij}^{LMP}, \theta_{ij}, \Lambda_{ij})\}$;
21: end for
22: for $i \in \mathcal{I}$ do
23:     | call adaptive oracles at $(x_{ij}^{LMP}, c_i)$ and obtain $\theta_{ij}$, $\bar{\theta}_{ij}$, $\bar{\phi}_{ij}$ and $\Lambda_{ij}$;
24: end for
25: set $L_j^{LBO} := f(x_{ij}^{LMP}) + \sum_{i \in \mathcal{I}} \pi_i \theta_{ij}$;
26: set $U_j^{UBO} := f(x_{ij}^{LMP}) + \sum_{i \in \mathcal{I}} \pi_i \bar{\theta}_{ij}$;
27: until $U_j^{UBO} - L_j^{LBO} \leq U_{j-1}^* - L_{j-1}^*$ or $n > |\mathcal{I}|$ or $L_j^{LBO} \geq U_{j-1}^*$;
28: set $U_j^* := \min(U_{j-1}^*, U_j^{UBO})$; $x_{ij}^{Ref} := x_{ij}^{LMP}$;
29: until $U_j^* - L_j^* \leq \epsilon$.

where $I_j^A = L_j^{LBO} - L_j^{LBO}$ is the actual improvement from iteration $j - 1$ to $j$, and $I_j^P = L_j^{LBO} - T_j$ is the predicted improvement from iteration $j - 1$ to $j$. Then we update $\gamma$,

if $I_j^A > 0$ and $I_j^P > 0$ then
    if $r \leq P$ then
        | $\gamma = 1 - \omega(1 - \gamma)$
    else if $P < r < P$ then
        | $\gamma = \gamma$
    else
        | $\gamma = \omega \gamma$
    end if
else $I_j^A \leq 0$ or $I_j^P \leq 0$

Inexact information, $\gamma = \gamma$

end if

where $0 \leq \omega \leq 1$ is a constant that increases or decreases $\gamma$, $\overline{P}$ and $\underline{P}$ are constants that determine what actions to take on $\gamma$. Unlike standard Benders that knows the exact value of the SPs, stabilised Adaptive Benders only knows lower and upper bounds on the objective values. By comparing the lower and upper bounds with the exact values of the SPs, we find that the lower bound oracle gives a much closer and more stable approximation. Therefore, we use $L^{LBO}$ instead of $U^{UBO}$ when defining the ratio $r$. Furthermore, a bad approximation from the lower bound oracle at the current point or a bad approximation from the upper bound oracle from previous points can lead to a negative $I_j^P$. Therefore, we choose to do nothing once we find that $I_j^P$ is negative. For $I_j^A$, there are two possibilities for it to be negative: 1) bad approximation from the lower bound oracle or the upper bound oracle, and 2) going to a bad point. A bad point, in this case, means the $L^{LBO}$ at current iteration is higher than the $L^{LBO}$ at the previous iteration. If the information is exact and $I_j^A$ is negative, one may reject the point, go back to the best point seen so far, and try again with a higher $\gamma$. However, in the case of inexact information, it may not be sensible to reject a point based on a bad approximation. In the computational study in this paper, fixed stabilisation is mainly used, but dynamic stabilisation is also tested.

4. Problem description, modelling strategies and modelling assumptions

The proposed power planning problem is designed to choose the optimal investment strategy and operating scheduling for a power system to achieve emission targets. In this section, we present the temporal and geographical representations of the problem and the modelling assumptions.

The problem under consideration aims to make optimal investment and operational decisions for the UK power system that satisfies the emission reduction goal under a) short-term uncertainty, including renewable energy availability and load profile; and b) long-term uncertainty, including CO\textsubscript{2} budget, CO\textsubscript{2} tax, and long-term power demand.

For the investment planning, we consider: (a) thermal generators (Coal-fired plant, OCGT, CCGT, Diesel, and nuclear plants); (b) generators with Carbon Capture and Storage (CCS) (Coal-fired plant with CCS); (c) renewable generators (offshore wind, onshore wind and solar PV); (d) electric storage (PHES and lithium); and (e) transmission lines. The capital expenditures and fixed operational costs are assumed to be known. The problem is to determine: (a) the capacities of technologies and (b) operational strategies that include scheduling of generators, storage and approximate power flow among regions to meet the power demand with minimum overall investment, operational and environmental costs.

4.1. Modelling strategies and assumptions

In this section, we present the modelling strategies and assumptions we use in the stochastic long-term multi-region multi-period investment planning problem.
4.1.1. Scenario generation

For short-term uncertainty, we select some time intervals with a half-hourly resolution in four seasons of a year and scale them up to represent an operational scenario. For long-term uncertainty, each independent uncertain parameter has \( n \) possible outcomes in the next stage, which is linked to additional \( n \) possible outcomes in the following stage. The realisations in one stage are assigned with an equal probability. We use a reasonably simple scenario generation routine because scenario generation is not the scope of the paper, and we refer the readers to (King & Wallace, 2012; Fairbrother et al., 2022) for more advanced scenario generation approaches.

4.1.2. Geographical representation of the problem

The problem potentially consists of many regions and results in a large model. Therefore, we aggregate regions into representative ones to reduce the number of locations. The generators and storage units in one region with the same characteristics are aggregated into clusters. In such a way, the model does not invest in a specific unit but in that type of device, and a linear investment model may be sufficient in this case.

4.1.3. Modelling assumptions

We assume that: a) a linear cost model for each technology because we deal with an aggregated system and the fixed part of the investment cost can be evened out and lead to a linear programming master problem; b) the Kirchhoff voltage law is omitted, and c) no loss in the transmission lines.

5. Mathematical model

This section presents the mathematical model for the power system investment planning and operational problem. The problem is decomposed by having an investment planning master problem and an operational SP. The complete nomenclature of the model can be found in Appendix A. We use the conventions that calligraphic capitalised Roman letters denote sets, upper case Roman and lower case Greek letters denote parameters, and lower case Roman letters denote variables. The indices are subscripts, and name extensions are superscripts. The same lead symbol represents the same type of thing. The names of variables, parameters, sets and indices are single symbols.

5.1. Investment planning model

\[
\begin{align*}
\min \ c^{INV} + \kappa \sum_{i \in \mathcal{I}} \pi_i c^{OPE}(x_i, c_i) \\
\text{s.t.} & \quad c^{INV} = \sum_{i \in \mathcal{I}} \delta_i^{I_0} \pi_i^{I_0} \sum_{p \in \mathcal{P}} C_{pi}^{Inv} x_{pi}^{Inst} + \kappa \sum_{i \in \mathcal{I}} \delta_i^{I_0} \pi_i^{I_0} \sum_{p \in \mathcal{P}} C_{pi}^{Fix} x_{pi}^{Acc} \\
x_{pi}^{Acc} &= x_{pi}^{Hist} + \sum_{i \in \mathcal{I} \cap \{i \in \mathcal{I} : |i-i_0| \leq H_p^i\}} x_{pi}^{Inst}, \quad p \in \mathcal{P}, i \in \mathcal{I} \\
x_{pi}^{Acc} &\leq x_{pi}^{Max}, \quad p \in \mathcal{P}, i \in \mathcal{I}
\end{align*}
\]
\[ x_i = \left( \left\{ x_{pi}^{\text{Acc}}, p \in P \right\}, \mu_i^{DP}, \mu_i^{E} \right), \quad i \in I \]  
\[ c_i = \left( C_i^{CO2} \right), \quad i \in I \]  
\[ x_{pi}^{\text{Inst}}, x_{pi}^{\text{Acc}} \in \mathbb{R}_0^+ \].  

The total cost for investment planning, Equation \((7a)\), consists of actual discounted investment costs and discounted fixed operating and maintenance costs \(c^{\text{INV}}\), as well as the expected operational cost of the system over the time horizon \(\kappa \sum_{i \in I} \pi_i c^{\text{OPE}}(x_i, c_i)\). Here, \(\kappa\) is a scaling factor that depends on the time step between two successive investment nodes. Constraint \((7c)\) states that the accumulated capacity of a technology \(x_{pi}^{\text{Acc}}\) in an operational node equals the sum of the historical capacity \(X_{pi}^{\text{Hist}}\) and newly invested capacities \(x_{pi}^{\text{Inst}}\) in its ancestor investment nodes \(I_i\) that are in their lifetimes. The parameter \(X_p^{\text{Max}}\) denotes the maximum accumulated capacity of technologies. Constraint \((7e)\) collects all right hand side coefficients that will be passed to the SP \((8)\) into vector \(x_i\). And constraint \((7f)\) collects all the cost coefficients into vector \(c_i\).

### 5.2. Operational model

We now compute the operational cost \(c^{\text{OPE}}(x_i, c_i)\) at one operational node \(i\) by solving SP \((8)\) given the decisions \(x_i\) and \(c_i\) made in the master problem \((7)\). Note that we omit index \(i\) in the operational model for ease of notation.

\[
\text{min} \quad \sum_{t \in T} \pi_t H_t \left( \sum_{g \in G} C^G p_{gt}^G + \sum_{s \in S} C^S p_{st}^S + \sum_{z \in Z} C^{\text{Shed}} p_{zt}^{\text{Shed}} \right) 
\]
\[
\text{s.t.} \quad p_{gt}^G \leq p_{gt}^{\text{Acc}}^G, \quad g \in G, t \in T \tag{8a}
\]
\[
- \mu_{l}^{\text{Acc}} \leq p_{lt}^L \leq \mu_{l}^{\text{Acc}}, \quad l \in L, t \in T \tag{8b}
\]
\[
p_{st}^{\text{SE}^+} \leq p_{st}^{\text{AccSE}}, \quad s \in S, t \in T \tag{8c}
\]
\[
p_{st}^{\text{SE}^-} \leq p_{st}^{\text{AccSE}}, \quad s \in S, t \in T \tag{8d}
\]
\[
q_{st}^{\text{SE}} \leq \gamma_s q_{st}^{\text{AccSE}}, \quad s \in S, t \in T \tag{8e}
\]
\[
- \kappa p_{gt}^{\text{AccG}} \leq p_{gt}^G - p_{g(t-1)}^G \leq \kappa p_{gt}^{\text{AccG}}, \quad g \in G, n \in N, t \in T \tag{8f}
\]
\[
\sum_{g \in G_g} p_{gt}^G + \sum_{l \in L_{g}^{\text{in}}} p_{lt}^L + \sum_{s \in S_s} p_{st}^{\text{SE}^-} + \sum_{r \in R_s} R_{zt} p_r^{\text{AccR}} + p_{zt}^{\text{ShedP}} = \mu_{zt}^{DP} \tag{8g}
\]
\[
q_{(t+1)}^{\text{SE}} = q_{st}^{\text{SE}} + H_t(q_s^{\text{SE}^+} - p_{st}^{\text{SE}^-}), \quad s \in S, n \in N, t \in T \tag{8h}
\]
\[
\sum_{t \in T} \sum_{g \in G} \pi_t H_g p_{gt}^G \leq \mu_{zt}^{E}, \quad z \in Z, t \in T \tag{8i}
\]
\[
p_{lt}^L \in \mathbb{R}_0, \tag{8j}
\]
\[
p_{gt}^G, p_{zt}^{\text{AccG}}, p_{zt}^{\text{ShedP}}, p_{st}^{\text{SE}^+}, p_{st}^{\text{SE}^-}, p_{st}^{\text{AccSE}}, q_{st}^{\text{SE}}, p_r^{\text{AccR}}, p_{zt}^{\text{ShedP}} \in \mathbb{R}_0^+. \tag{8k}
\]
The operational cost function $c^{OP}(x, c)$ includes total operating costs of all generators and storage facilities $C^G G + C^S S^P + \text{load shedding costs } C^{Shed} p^{Shed}$. The parameters $C^G G$ and $C^S S$ include the variable operational cost of generators and storage. For thermal generators, $C^G G$ also includes the fuel cost and the CO$_2$ tax charged on the emissions of generators. Constraint (8b) ensures that generators are within their capacity limits. Constraint (8c) shows that the power flow $p^L$ is within the transmission capacity $p^{AccL}$. Constraints (8d) and (8e) dictate that the power charged $p^{SE+}$ and the discharging power $p^{SE-}$ of a storage facility should be within the capacity, respectively. Constraint (8f) limits the energy storage level $q^{SE}$ to be within the capacity $q^{AccSE}$. Constraint (8g) captures how fast thermal generators can ramp up or ramp down their power output, respectively. The parameters $\alpha^G G$ is the maximum ramp rate of thermal generators. The power nodal balance, Constraint (8h), ensures that in one operational period $t$, the sum of total power generation of thermal generators $p^G G t$, power discharged from all the electricity storage $p^{SE-} st$, renewable generation $R^R R^C R^R t$, power transmitted to this region, and load shed $p^{Shed} st$ equals the sum of power demand $P^{DP} st$ power transmitted to other regions, and power generation shed $p^{GShed} st$. The parameter $R^R R^R t$ is the capacity factor of a renewable unit that is a fraction of the nameplate capacity $p^{AccR}$. The subset of a technology in region $z$ is represented by $R_z := \{ r \in R : r \text{ is available in region } z \}$, where $R$ can be replaced by other sets of technologies. Constraint (8i) states that the state of charge $q^{SE} st$ in period $t + 1$ depends on the previous state of charge $q^{SE} st$, the charged power $p^{SE+} st$ and discharged power $p^{SE-} st$. The parameter $( q^{SE} st)$ represent the charging efficiency. Constraint (8j) restricts the total emission. The parameter $H_t$ is the length of the period $t$. The parameter $\mu^E$ is the CO$_2$ budget. The symbol $E^G G$ is the emission factor per unit of power generated. The capacities $p^{AccG} G, p^{AccL} l, p^{AccSE} s$, scaling factor of demand $\mu^{DP} DP$ and CO$_2$ budget $\mu^E$ are passed from the master problem (7) via vector $x_i$ and CO$_2$ tax that is included in cost coefficient $C^G G$ is passed from master problem (7) via vector $c_i$.

6. Results

This section firstly uses small illustrative cases to show how stabilisation helps solve multi-region investment planning problems. Then we demonstrate the proposed algorithm on larger instances and present the computational results.

6.1. Illustrative cases

We use three cases to show the value of stabilisation in a multi-region investment planning problem. A summary of the four cases is presented in Table 1. To simplify the visualisation of the results, we consider only two types of generation, OCGT and Diesel, and a one-time investment planning problem is solved. In this case, there is no difference between Adaptive Benders and standard Benders because there is only one SP.

Figure 3 - Figure 5 show how solutions are explored until convergence. In each figure, the darkest blue point represents the initial solution, the lightest blue point is the optimal solution, and the arrows indicate the order of points explored. For the stabilised versions, the stabilisation factor
Table 1: Summary of the illustrative cases.

| Description                                                                 |
|----------------------------------------------------------------------------|
| Case A | single region, two technologies to invest (OCGT and Diesel)                |
| Case B | two unconnected regions with sizes 60% and 40% of case A                  |
| Case C | Case B with a transmission line with 0 initial capacity                    |

is fixed to 0.2. In all cases, there is degeneracy in the dimension of the total amounts of the two generation types. From Case B and Case C, we find that there is degeneracy in the dimension of regions. In the two region cases, there is a CO$_2$ constraint that restricts the total emissions from both regions. In Case B, where to put the capacities becomes relevant. From Figure 4, we see that without stabilisation, the algorithm struggles to balance the capacities of the two technologies and starts jumping to points with different proportions of the two technologies many times until it
finds the optimal solution. In Figure 4, we see that the stabilised approach is clearer about which direction to explore and make small movements towards the optimal instead of sampling points wildly. The number of iterations is doubled without stabilisation. For a more realistic problem with more technologies and regions and a more complicated network topology, the value of stabilisation reveals further, as is shown in Section 6.3.

![Figure 5: Comparative results of Case C.](image)

(a) Case C (stabilised Benders, 13 iters)  (b) Case C (unstabilised Benders, 27 iters)

In Case C, two regions are initially disconnected, but a line can be invested to connect them. However, there should be no line invested because the two regions are proportional to each other and making investments in the local generation is optimal. By observing the solution proposed by RMP in the unstabilised version, we notice that RMP does not realise that and makes an investment in the line in some iterations before finding the optimum, and this leads to more iterations compared with the stabilised version.

6.2. Case study

We test the stabilised Benders algorithm with adaptive oracles on the stochastic investment planning of the UK power system. We use the model presented in Section 5 to investigate the computational issues. The network topology is shown in Figure 6. We implemented the algorithm and model in Julia 1.7.3 using JuMP (Dunning et al., 2017) and solved with Gurobi 9.5.1 (Gurobi Optimization, LLC, 2022). We ran the code on nodes of a computer cluster with a 2x 3.6GHz 8 core Intel Xeon Gold 6244 CPU and 384 GB of RAM, running on CentOS Linux 7.9.2009. Some data was taken from (Mazzi et al., 2020). The Julia code and data for the case study have been made publicly available at (Zhang et al., 2022a).

6.3. Computational results

This section presents the computational results of the proposed stabilised Adaptive Benders. We compare the performance of stabilised Adaptive Benders against the unstabilised versions of Adaptive Benders with one SP solved per iteration (Mazzi et al., 2020) and the standard Benders.
We use the model presented in Section 5 to solve a 5-region UK power system planning to make the benchmark. The long-term uncertainties include CO₂ price and power demand. The short-term uncertainties include wind and solar capacity factors and load profiles. The summary of cases and their problem sizes are shown in Table 2. In Case 0–3, there are four short-term operational scenarios, each consisting of 4380 operational periods. Case 0 has no long-term uncertainty. Case 1 has one long-term uncertainty, CO₂ budget. Case 2 has CO₂ budget and long-term demand uncertainty. And Case 3 has CO₂ budget, long-term demand, and CO₂ tax as long-term uncertainty.

From Table 3, we can see that (a) stabilised Adaptive Benders is up to 113.7 times faster than standard Benders for a 1.00% convergence tolerance and 45.5 times faster than standard Benders for a 0.10% convergence tolerance, (b) Adaptive Benders gets slower when converging to a tighter tolerance and (c) compared with Adaptive Benders, stabilised Adaptive Benders is up to 2.14 times faster for a 1.00% convergence tolerance, and Adaptive Benders cannot solve the largest instance to 0.10% due to severe oscillation. Therefore, for Case 3, we report the performance of unstabilised

---

**Table 2: Overview of the cases used in the computational study.**

| Operational periods per short-term scenario | Short-term scenarios | Long-term scenarios | Number of decision nodes | Problem size (undecomposed) |
|--------------------------------------------|----------------------|---------------------|--------------------------|-----------------------------|
|                                            | Present              | In 5 years          | In 10 years              | Variables | Constraints | Nonzeros |
| Case 0                                     | 4380                 | 4                   | 1                        | 1         | 1          | 3         | 2.7 × 10⁶ | 7.7 × 10⁶ | 1.9 × 10⁷ |
| Case 1                                     | 4380                 | 4                   | 9                        | 1         | 3          | 9         | 1.6 × 10⁷ | 4.6 × 10⁷ | 1.1 × 10⁸ |
| Case 2                                     | 4380                 | 4                   | 81                       | 1         | 9          | 81        | 1.2 × 10⁸ | 3.5 × 10⁸ | 8.4 × 10⁸ |
| Case 3                                     | 4380                 | 4                   | 729                      | 1         | 27         | 729       | 1.0 × 10⁹ | 4.5 × 10⁹ | 1.1 × 10¹⁰ |

*: the model cannot be loaded into the solver due to its size.
3.1. Improving the robustness

The stabilisation factor $\gamma$ significantly impacts the performance. A very small $\gamma$ leads to loose stabilisation and makes stabilisation less effective, whereas a very large $\gamma$ leads to tight stabilisation and may hinder the exploitation of the solution space. We test the performance using different $\gamma$ from 0.025 to 0.9 and present the results in Table 4. We find that a stabilisation factor less than 0.2 generally performs well. By checking the average performance over four cases, we find that $\gamma$ equals...
0.025 give the best performance. However, for different cases, the $\gamma$ that yields better performance varies. Furthermore, a rule of thumb for setting a fixed stabilisation factor may be to set it less than or equal to 0.5.

Table 5: Results for stabilised Benders decomposition with adjusted level sets (speed up: the time spent using fixed $\gamma$ divided by time spent using dynamic stabilisation).

| Case   | $\gamma$ | $\omega$ | $\mathcal{P}$ | $\mathcal{T}$ | Iters/Evals | Time (s) |
|--------|----------|----------|---------------|---------------|-------------|----------|
|        |          |          | $\epsilon = 1.00\%$ | $\epsilon = 0.10\%$ | $\epsilon = 1.00\%$ | Speed up | $\epsilon = 0.10\%$ | Speed up |
| Case 0 | 0.025    | 0.5      | 0.1           | 0.9           | 20/30       | 35/48    | 843          | 0.9  | 1368      | 1.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 17/27       | 28/44    | 702          | 1.1  | 1233      | 1.1   |
|        | 0.100    | 0.5      | 0.1           | 0.9           | 14/19       | 31/38    | 526          | 1.2  | 1081      | 0.9   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 13/18       | 23/31    | 484          | 1.3  | 862       | 1.2   |
|        | 0.500    | 0.5      | 0.1           | 0.9           | 17/24       | 27/37    | 753          | 1.3  | 1212      | 1.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 20/27       | 28/38    | 761          | 1.2  | 1112      | 1.1   |
|        | 0.900    | 0.5      | 0.1           | 0.9           | 31/39       | 53/43    | 1123         | 2.4  | 1554      | 2.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 33/41       | 43/55    | 1248         | 2.2  | 1655      | 1.9   |
| Case 1 | 0.025    | 0.5      | 0.1           | 0.9           | 22/31       | 35/49    | 882          | 0.9  | 1425      | 0.8   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 17/27       | 28/43    | 732          | 1.1  | 1238      | 0.9   |
|        | 0.100    | 0.5      | 0.1           | 0.9           | 14/27       | 31/57    | 753          | 2.4  | 1651      | 1.4   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 18/28       | 26/40    | 734          | 2.5  | 1157      | 2.1   |
|        | 0.500    | 0.5      | 0.1           | 0.9           | 22/42       | 31/54    | 1182         | 1.1  | 1599      | 1.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 22/45       | 34/45    | 1254         | 1.0  | 2054      | 0.8   |
|        | 0.900    | 0.5      | 0.1           | 0.9           | 15/20       | 30/44    | 627          | 13.0 | 1342      | 6.5   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 26/41       | 35/61    | 1277         | 6.4  | 1901      | 4.6   |
| Case 2 | 0.025    | 0.5      | 0.1           | 0.9           | 38/68       | 51/110   | 1979         | 0.8  | 3224      | 1.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 18/61       | 41/152   | 1601         | 1.0  | 4461      | 0.7   |
|        | 0.100    | 0.5      | 0.1           | 0.9           | 28/67       | 36/94    | 1775         | 1.0  | 2720      | 1.1   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 29/67       | 39/99    | 1931         | 0.9  | 2858      | 1.1   |
|        | 0.500    | 0.5      | 0.1           | 0.9           | 39/90       | 46/108   | 2533         | 1.7  | 3191      | 2.3   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 39/102      | 49/141   | 3124         | 1.3  | 4261      | 1.7   |
|        | 0.900    | 0.5      | 0.1           | 0.9           | 48/95       | 76/194   | 2703         | 8.0  | 5839      | 5.7   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 50/107      | 65/164   | 3377         | 6.4  | 5086      | 6.5   |
| Case 3 | 0.025    | 0.5      | 0.1           | 0.9           | 49/217      | 84/418   | 6573         | 0.7  | 14067     | 1.2   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 28/173      | 66/399   | 5486         | 0.8  | 15195     | 1.1   |
|        | 0.100    | 0.5      | 0.1           | 0.9           | 24/111      | 87/433   | 3306         | 2.0  | 14860     | 1.0   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 44/143      | 126/606  | 4538         | 1.5  | 21276     | 0.7   |
|        | 0.500    | 0.5      | 0.1           | 0.9           | 52/182      | 140/566  | 5837         | 7.5  | 24892     | 2.9   |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 61/210      | 123/494  | 6901         | 6.4  | 17332     | 4.2   |
|        | 0.900    | 0.5      | 0.1           | 0.9           | 58/194      | 98/425   | 5758         | 33.5 | 14838     | 25.4  |
|        | 0.9      | 0.1      | 0.1           | 0.9           | 76/178      | 133/444  | 6042         | 31.9 | 15820     | 23.8  |
| Average |            |          |          |               | 31/80       | 56/174   | 2417         |      | 6011      |      |
| Average (fixed $\gamma$)* | | | | | 225/338 | 307/550 | 84749 | 145321 | |
| Standard deviation | | | | | 16/64 | 35/183 | 2052 | 6854 | |
| Standard deviation (fixed $\gamma$)* | | | | | 342/424 | 458/656 | 102821 | 190074 | |

*: consider the runs when $\gamma = 0.025, 0.1, 0.5$ and 0.9

We see that different fixed $\gamma$ can lead to a noticeable difference in performance. It is aligned with Remark 5 in (Zverovich et al., 2012). Unlike (Zverovich et al., 2012) who decided to use $\gamma = 0.5$ to set the level set, we test the approach presented in Section 3.1 to adjust $\gamma$ and then the level set that may make the stabilisation more robust and independent of the choice of the stabilisation
We test extensively the dynamic stabilisation scheme on Case 0-3. The results for different cases are shown in Table 5. By comparing results from Table 5 and their fixed $\gamma$ counterparts in Table 4, we can see that in almost all cases adjusting the level set outperforms the cases using a fixed $\gamma$. Although sometimes dynamic stabilisation is slightly slower than standard stabilisation, most of the time it is better. Furthermore, dynamic stabilisation makes the level set method stabilisation more robust in terms of the choice of initial $\gamma$. It is particularly valuable because one may need extensive tests to find the $\gamma$ that yields the best performance for the problems to be solved. However, as we see in Table 4, different problems may have different best $\gamma$. Therefore, a dynamic stabilisation that makes the performance less dependent on the choice of initial $\gamma$ may make it easier to get a satisfying performance if one chooses a bad initial $\gamma$ because the dynamic adjustment will help correct $\gamma$ to a sensible value while solving the problem.

6.4. Power system analysis

In this section, we present the results of the 5-region UK power system planning problem. We analyse the investment decisions, expected costs, and the Value of the Stochastic Solution (VSS).

The investment decisions in the first stage are presented in Table 6. There are no investments in technologies except onshore wind in the first investment stage. We notice that the transmission lines are expanded in the later investment nodes. Therefore, for the first investment stage, only investment in onshore wind is presented. The onshore wind is mainly invested in Scotland, North England and South England in the first investment stage. When considering only short-term uncertainty, we can see that in Case 0, a total of 90.85 GW of onshore wind is invested, 28% of which is in North England. Compared with Case 1, around 3.8 GW less capacity is installed in Case 0. When considering uncertainty in both long-term demand and CO$_2$ budget, we can see a 3.56 GW investment in onshore wind in Scotland, compared with 14.76 GW in Case 1 and 1.42 GW in Case 3.

| Case  | Scotland | North England | Midlands & Wales | East England | South England | UK Total |
|-------|----------|---------------|------------------|--------------|--------------|----------|
| Case 0| 0.00     | 25.70         | 0.00             | 0.00         | 65.15        | 90.85    |
| Case 1| 14.76    | 28.25         | 0.00             | 0.00         | 51.68        | 94.69    |
| Case 2| 3.56     | 23.22         | 0.00             | 0.00         | 59.19        | 85.97    |
| Case 3| 1.42     | 23.60         | 0.00             | 0.00         | 69.16        | 94.18    |

Table 6: Investment in onshore wind in the first investment stage.

Table 7 shows the optimal costs and the VSS for considering long-term uncertainties. We can see that there is up to £7,702 million VSS when considering uncertainty, including CO$_2$ budget and long-term demand. The VSS is £2,904 million when considering only CO$_2$ budget as an uncertainty parameter. When considering long-term uncertainty, including CO$_2$ budget, CO$_2$ tax and long-term demand, the VSS is 4.4% of the optimal cost. This shows the value of including long-term uncertainty in a long-term planning problem and solving a large model.
Table 7: Optimal costs and VSS.

|                | Case 0 | Case 1 | Case 2 | Case 3 |
|----------------|--------|--------|--------|--------|
| Optimal cost (mn £) | 174099 | 174276 | 174871 | 174785 |
| VSS (long-term uncertainty, mn £) | - | 2904 | 7325 | 7702 |

7. Discussion

In this paper, we propose a method to address the computational difficulty of a multi-stage stochastic programming problem with short-term and long-term uncertainty that is formulated using a multi-horizon stochastic programming approach. Similar studies on developing a method to solve such type of problems can be found in (Zakeri et al., 2000; Downward et al., 2020). Compared with their approach, we exploit the properties of the SP and stabilise the algorithm with the level set method and adaptively select SPs to solve exactly per iteration for better approximation, which shows significant performance improvement. The method can be generally applied to solve any problem that is formulated in (1) and (2).

We demonstrate our proposed method on a multi-region UK power system planning problem. To the authors’ knowledge, this is the first study that presents a multi-horizon formulation of a multi-region power system planning problem with short-term and long-term uncertainty and proposes a method to solve such a problem efficiently. Compared with a similar problem for long-term investment planning such as (Backe et al., 2022) that only considers short-term uncertainty, this paper firstly introduces both long-term and short-term uncertainty in a power system planning problem using a multi-horizon framework.

We notice and analyse the oscillation of the Benders-type decomposition method for multi-region investment planning problems. The level set method stabilisation approach was used to stabilise Benders. Compared with the existing literature that studied the level set method, we integrate it with the inexact oracles and show that it significantly improves computational performance. In addition, similar studies normally set the level set in an ad hoc way (Zverovich et al., 2012; Ruszczyński & Świątanowski, 1997). Moreover, we test to adjust the target based on a proposed measurement. For the test instance, adjusting the level set can usually yield better or equivalent performance. However, the parameters that yield the best performance may be case-dependent.

Although the stabilisation is useful, the stabilisation problem can potentially be a large QP and slow to solve. One possible approach to stabilise the problem efficiently is to utilise the built-in method, analytic centre (Gondzio et al., 1996) in a commercial solver like Gurobi to solve a feasibility problem to avoid solving a QP. We test utilising the analytic centre of Gurobi to potentially avoid solving a QP LMP. However, the results show that proper stabilisation may still be the better option, even for large problems.

We demonstrate the method for solving large-scale linear programming. However, the method can be applied to solve mixed-integer linear programming problems without modification as long as the integer variables are in the MP. In such a case, the stabilisation problem becomes a mixed-integer QP problem which may be slow to solve. Some other stabilisation techniques, such as local
branching (Baena et al., 2020) may be an alternative.

Although this paper presents a general method to solve a class of large-scale optimisation problems very efficiently, there are some limitations. Firstly, we need the same coefficient matrices in all nodes to utilise the inexact oracles. This may be limited when different operational scenarios are preferred. However, the inexact oracles can be generalised to apply to some groups of nodes with the same matrices and work in a problem with different matrices in different nodes. However, having different scenarios in different nodes may lead to lower stability of a stochastic problem. Secondly, although multi-horizon formulation significantly reduces the problem size, it may be limited when long-term storage, such as pumped hydro storage, is in the system. However, an easy fix to this issue is to collect information on the storage level at the end of one stage in the master problem and pass it to the next stage. Thirdly, in the case study, we only demonstrate the proposed method to solve the UK power system planning problem and show significant performance improvement, but more problems may be solved using the proposed method to gain more insights into the performance of the algorithm.

8. Conclusions and future work

In this paper, we proposed stabilised Benders decomposition with adaptive oracles to solve long-term multi-region investment planning problems with short-term and long-term uncertainty. We applied the algorithm to solve a multi-region UK power system investment planning problem towards 2035. We formulated such a problem using a multi-horizon stochastic programming approach. The test instances have up to 1 billion variables and 4.5 billion constraints. The computational results show that: a) for a 1.00% convergence tolerance, the proposed stabilised method is up to 113.7 times faster than standard Benders decomposition and 2.14 times faster than Adaptive Benders decomposition without stabilisation; b) for a 0.10% convergence tolerance, the proposed stabilised method is up to 45.5 times faster than standard Benders decomposition and the unstabilised Adaptive Benders decomposition cannot solve the largest instance to the convergence tolerance due to severe oscillation and c) dynamic level set method increases the robustness of the stabilisation.

Although the proposed method reduced the computational effort significantly and was used to solve multi-horizon stochastic programming with short-term and long-term uncertainty, we notice that for a very large problem with many decision nodes, the reduced master problem and the stabilisation problem may take longer to solve. Therefore, in future, techniques including node aggregation and cuts selection and deletion may be needed to improve the performance. In addition, although multi-horizon formulation reduces the problem size significantly, the model size may be reduced further by adjusting the scenario tree, e.g., removing the scenarios that do not make a difference while solving the problem.

CRediT author statement

Hongyu Zhang: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Visualisation, Data curation, Writing - original draft, Writing - review & editing.
Nicolò Mazzi: Conceptualisation, Methodology, Data curation, Software, Writing - review & editing. Ken McKinnon: Conceptualisation, Methodology, Supervision, Writing - review & editing, Funding acquisition. Rodrigo Garcia Nava: Conceptualisation, Methodology, Software, Writing - review & editing. Asgeir Tomasgard: Conceptualisation, Supervision, Writing - review & editing, Funding acquisition.

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.

Acknowledgements

This work was supported by the Research Council of Norway through PETROSENTER LowEmission [project code 296207]; and the Engineering and Physical Sciences Research Council (EPSRC) through the CESI project [EP/P001173/1].

Appendix A. Appendix

Investment planning model sets

$\mathcal{P}$ set of technologies
$\mathcal{I}$ set of operational nodes, indexed by $i$
$\mathcal{I}_0$ set of investment nodes, indexed by $i_0$
$\mathcal{I}_i$ set of investment nodes $i$ ($i \in \mathcal{I}_0$) ancestor to operational node $i$ ($i \in \mathcal{I}$)

Operational model sets

$\mathcal{N}$ set of time slices
$\mathcal{T}$ set of hours in all time slices
$\mathcal{L}$ set of transmission lines
$\mathcal{G}$ set of thermal generators
$\mathcal{S}$ set of electricity storage
$\mathcal{R}$ set of renewable generations

Investment planning model parameters

$C_{pi}^{Inv}$ unitary investment cost of device $p$ in investment node $i$ ($p \in \mathcal{P}, i \in \mathcal{I}_0$) [£/MW]
$C_{pi}^{Fix}$ unitary fix operational and maintenance cost of device $p$ ($p \in \mathcal{P}$) [£/MW]
$X_{pi}^{Hist}$ historical capacity of device $p$ ($p \in \mathcal{P}$) [MW]
$X_{pi}^{Max}$ maximum installed capacity of device $p$ ($p \in \mathcal{P}$) [MW]
$\kappa$ scaling effect depending on the number of operation years between investment nodes
$\delta_{i_0}^{I} / \delta_{i}^{I}$ discount factor of investment node $i$ ($i_0 \in \mathcal{I}_0$)/ operational node $i$ ($i \in \mathcal{I}$)
$\pi_{i_0}^{I} / \pi_{i}^{I}$ probability of investment node $i$ ($i_0 \in \mathcal{I}_0$)/ operational node $i$ ($i \in \mathcal{I}$)
$H_{pi}$ life time of technology $p$ ($p \in \mathcal{P}$)
$x_i$ right hand side coefficients of the operational subproblem
$c_i$ cost coefficients of the operational subproblem
µ_{i}^{E} \quad \text{CO}_2 \text{ budget at operational node } i \ (i \in I) \\
µ_{i}^{DP} \quad \text{scaling factor on power demand at operational node } i \ (i \in I) \\
C_{i}^{CO_2} \quad \text{CO}_2 \text{ emission price at operational node } i \ (i \in I) \\

**Operational model parameters**

µ_{i}^{E} \quad \text{yearly CO}_2 \text{ emission limit (tonne)} \\
π_{t} \quad \text{probability of operation period } t \ (t \in \mathcal{T}) \\
H_{t} \quad \text{number of hour(s) in one operational period } t \\
α_{g}^{G} \quad \text{maximum ramp rate of gas turbines } (g \in \mathcal{G}) \ [\text{MW/MW}] \\
R_{r}^{R} \quad \text{capacity factor of renewable unit } r \text{ in period } t \ (r \in \mathcal{R}, t \in \mathcal{T}) \\
η_{s}^{SE} \quad \text{efficiency of electricity store } s \ (s \in \mathcal{S}) \\
γ_{s}^{SE} \quad \text{power ratio of electricity store } s \ (s \in \mathcal{S}) \ [\text{MWh/MW}] \\
E_{g}^{G} \quad \text{emission factor of gas turbine } g \ (g \in \mathcal{G}) \ [\text{tonne/MWh}] \\
C_{g}^{G}/C_{s}^{SE} \quad \text{total operational cost of a generator } g/ \text{ a storage facility } s \ (g \in \mathcal{G}, s \in \mathcal{S}) \ [£/MW] \\
C_{\text{ShedP}} \quad \text{power load shed penalty cost} \ [£/MW] \\
P_{zt}^{DP} \quad \text{power demand at region } z \text{ period } t \ (z \in \mathcal{Z}, t \in \mathcal{T}) \ [\text{MW}] \\

**Investment planning model variables**

x_{pi}^{Acc} \quad \text{accumulated capacity of device } p \text{ in operational node } i \ (p \in \mathcal{P}, i \in I) \ [\text{MW}] \\
x_{pi}^{Inst} \quad \text{newly invested capacity of device } p \text{ in investment node } i_{0} \ (p \in \mathcal{P}, i \in I_{0}) \ [\text{MW}] \\
c_{i}^{INV} \quad \text{total expected investment cost} \ (£) \\

**Operational model variables**

p_{g}^{AccG} \quad \text{accumulated capacity of gas turbine } g \ (g \in \mathcal{G}) \ [\text{MW}] \\
p_{r}^{AccR} \quad \text{accumulated capacity of renewable unit } r \ (r \in \mathcal{R}) \ [\text{MW}] \\
p_{s}^{AccSE} \quad \text{accumulated charging/discharging capacity of electricity store } s \ (s \in \mathcal{S}) \ [\text{MW}] \\
p_{g}^{G} \quad \text{power generation of gas turbine } g \text{ in period } t \ (g \in \mathcal{G}, t \in \mathcal{T}) \ [\text{MW}] \\
p_{s}^{SE+}/p_{s}^{SE-} \quad \text{charge/discharge power of electricity store } s \text{ in period } t \ (s \in \mathcal{S}, t \in \mathcal{T}) \ [\text{MW}] \\
p_{lt}^{L} \quad \text{power flow in line } l \text{ in period } t \ (l \in \mathcal{L}, t \in \mathcal{T}) \ [\text{MW}] \\
p_{zt}^{G\text{ShedP}} \quad \text{generation shed at } z \text{ in period } t \ (z \in \mathcal{Z}, t \in \mathcal{T}) \ [\text{MW}] \\
q_{st}^{SE} \quad \text{energy level of electricity store } s \text{ at the start of period } t \ (s \in \mathcal{S}, t \in \mathcal{T}) \ [\text{MWh}] \\
p_{zt}^{\text{ShedP}} \quad \text{load shed at } z \text{ in period } t \ (z \in \mathcal{Z}, t \in \mathcal{T}) \ [\text{MW}] \\

**Function**

c_{i}^{OPE}(\cdot, \cdot) \quad \text{operational cost at node at operational node } i \ (i \in I) \ (£)
References

Backe, S., Skar, C., del Granado, P. C., Turgut, O., & Tomasgard, A. (2022). EMPIRE: An open-source model based on multi-horizon programming for energy transition analyses. *SoftwareX, 17*, 100877. doi:10.1016/j.softx.2021.100877.

Baena, D., Castro, J., & Frangioni, A. (2020). Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy. *Management Science, 66*, 3051–3068. doi:10.1287/mnsc.2019.3341.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik, 4*, 238–252. doi:10.1007/BF01386316.

Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Science & Business Media. doi:10.1007/978-1-4614-0237-4.

Boffino, L., Conejo, A. J., Sioshansi, R., & Oggioni, G. (2019). A two-stage stochastic optimization planning framework to decarbonize deeply electric power systems. *Energy Economics, 84*, 104457. doi:10.1016/j.eneco.2019.07.017.

Conejo, A. J., Baringo Morales, L., Kazempour, S. J., & Siddiqui, A. S. (2016). *Investment in Electricity Generation and Transmission*. Springer Cham. doi:10.1007/978-3-319-29501-5.

Downward, A., Baucke, R., & Philpott, A. B. (2020). JuDGE.jl: a Julia package for optimizing capacity expansion. Technical Report Engineering Science, University of Auckland. URL: http://www.optimization-online.org/DB_FILE/2020/11/8086.pdf [accessed July 2022].

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A Modeling Language for Mathematical Optimization. *SIAM Review, 59*, 295–320. doi:10.1137/15M1020575.

Fabian, C. I. (2000). Bundle-type Methods for Inexact Data. *Central European Journal of Operations Research, 8*, 35–55. URL: https://web.cs.elte.hu/~fabian/inexact_bundle.pdf. [accessed July 2022].

Fairbrother, J., Turner, A., & Wallace, S. W. (2022). Problem-driven scenario generation: an analytical approach for stochastic programs with tail risk measure. *Mathematical Programming, 191*, 141–182. doi:10.1007/s10107-019-01451-7.

Fletcher, R. (2000). *Practical methods of optimization*. John Wiley & Sons. doi:10.1002/9781118723203.

Fodstad, M., Egging, R., Midthun, K., & Tomasgard, A. (2016). Stochastic modeling of natural gas infrastructure development in Europe under demand uncertainty. *Energy Journal, 37*, 5–32. doi:10.5547/01956574.37.SI3.mfod.

Gacitua, L., Gallegos, P., Henriquez-Auba, R., Lorca, Negrete-Pincetic, M., Olives, D., Valenzuela, A., & Wenzel, G. (2018). A comprehensive review on expansion planning: Models and tools for energy policy analysis. *Renewable and Sustainable Energy Reviews, 98*, 346–360. doi:10.1016/j.rser.2018.08.043.

Galan, A., de Prada, C., Gutierrez, G., Sarabia, D., Grossmann, I. E., & Gonzalez, R. (2019). Implementation of RTO in a large hydrogen network considering uncertainty. *Optimization and Engineering, 20*, 1161–1190. doi:10.1007/s11081-019-09444-3.

Gondzio, J., du Merle, O., Sarkissian, R., & Vial, J. P. (1996). ACCPM - a library for convex optimization based on an analytic center cutting plane method. *European Journal of Operational Research, 94*, 206–211. doi:10.1016/0377-2217(96)00169-5.

Gupta, V., & Grossmann, I. E. (2014). Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. *Journal of Petroleum Science and Engineering, 124*, 180–197. doi:10.1016/j.petrol.2014.10.006.

Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual. URL: https://www.gurobi.com [accessed July 2022].

Hellemo, L., Midthun, K., Tomasgard, A., & Werner, A. (2013). Multi-stage stochastic programming for natural gas infrastructure design with a production perspective. In *Stochastic Programming: Applications in Finance, Energy, Planning and Logistics* (pp. 259–288). World Scientific. doi:10.1142/9789814407519_0010.

Kaut, M., Midthun, K. T., Werner, A. S., Tomasgard, A., Hellemo, L., & Fodstad, M. (2014). Multi-horizon stochastic programming. *Computational Management Science, 11*, 179–193. doi:10.1007/s10287-013-0182-6.

King, A., & Wallace, S. (2012). *Modeling with Stochastic Programming*. Springer Science & Business Media. doi:10.1007/978-0-387-87817-1.
Lara, C. L., Siirola, J. D., & Grossmann, I. E. (2020). Electric power infrastructure planning under uncertainty: stochastic dual dynamic integer programming (SDDiP) and parallelization scheme. *Optimization and Engineering, 21*, 1243–1281. doi:10.1007/s11081-019-09471-0.

Lemarechal, C., Nemirovskii, A., & Nesterov, Y. (1995). New variants of bundle methods. *Mathematical Programming, Series B, 69*, 111–147. doi:10.1007/BF01585555.

Li, C. (2021). *Algorithms for Stochastic Mixed-integer Nonlinear Programming and Long Term Optimization of Electric Power Systems*. Ph.D. thesis Carnegie Mellon University. URL: https://kilthub.cmu.edu/articles/thesis/Algorithms_for_Stochastic_Mixed-integer_Nonlinear_Programming_and_Long_Term_Optimization_of_Electric_Power_Systems/19146431 [accessed September 2022.

Li, C., Conejo, A. J., Liu, P., Omell, B. P., Siirola, J. D., & Grossmann, I. E. (2022). Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems. *European Journal of Operational Research, 297*, 1071–1082. doi:10.1016/j.ejor.2021.06.024.

Mazzi, N., Grothey, A., McKinnon, K., & Sugishita, N. (2020). Benders decomposition with adaptive oracles for large scale optimization. *Mathematical Programming, Series B, 69*, 111–147. doi:10.1007/BF01585555.

Munoz, F. D., Hobbs, B. F., & Watson, J. P. (2016). New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints. *European Journal of Operational Research, 248*, 888–898. doi:10.1016/j.ejor.2015.07.057.

Munoz, F. D., & Watson, J. P. (2015). A scalable solution framework for stochastic transmission and generation planning problems. *Computational Management Science, 12*, 491–518. doi:10.1007/s10287-015-0229-y.

Pereira, M. V., & Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming, 52*, 359–375. doi:doi.org/10.1007/BF01582895.

Powell, W. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons. doi:10.1002/9781118029176.

Powell, W. B., & Meisel, S. (2016). Tutorial on stochastic optimization in energy—part i: Modeling and policies. *IEEE Transactions on Power Systems, 31*, 1459–1467. doi:10.1109/TPWR.S.2015.2424974.

Rahmanian, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research, 259*, 801–817. doi:10.1016/j.ejor.2016.12.005.

Ruszczyński, A., & Świȩtanowski, A. (1997). Accelerating the regularized decomposition method for two stage stochastic linear problems. *European Journal of Operational Research, 101*, 328–342. doi:10.1016/S0377-2217(96)00401-8.

Salo, A., Andelmin, J., & Oliveira, F. (2022). Decision programming for mixed-integer multi-stage optimization under uncertainty. *European Journal of Operational Research, 299*, 550–565. doi:10.1016/j.ejor.2021.12.013.

Singh, K. J., Philpott, A. B., & Kevin Wood, R. (2009). Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems. *Operations Research, 57*, 1271–1286. doi:10.1287/opre.1080.0678.

Skar, C., Doorman, G., & Tomasgard, A. (2014). Large-scale power system planning using enhanced Benders decomposition. *Proceedings - 2014 Power Systems Computation Conference, PSCC 2014*, . doi:10.1109/PSCC.2014.7038297.

Slyke, R., R M Van; Wets (1969). Optimal Control and Stochastic Programming. *SIAM journal on applied mathematics, 17*, 638–663. doi:10.1137/0117061.

Wallace, S. W., & Fleten, S. E. (2003). Stochastic Programming Models in Energy. *Handbooks in Operations Research and Management Science, 10*, 637–677. doi:10.1016/S0927-0507(03)10010-2.

Zakeri, G., Philpott, A., & Ryan, D. (2000). Inexact Cuts in Benders Decomposition. *SIAM Journal on Optimization, 10*, 643–657. doi:10.1137/S1052623497318700.

Zhang, H., McKinnon, K., Mazzi, N., Nava, R. G., & Tomasgard, A. (2022a). Stabilised adaptive Benders. doi:10.5281/zenodo.6818491.

Zhang, H., Tomasgard, A., Knudsen, B. R., Svendsen, H. G., Bakker, S. J., & Grossmann, I. E. (2022b). Modelling and analysis of offshore energy hubs. *Energy* (p. 125219). doi:10.1016/j.energy.2022.125219.
