Systems & Control Transactions

Research Article
Foundations of Computer Aided Process Design (FOCAPD 2024)
Breckenridge, Colorado, USA. July 14-18, 2024
Peer Reviewed Conference Proceeding

PSE PRESS

RiNSES^4: Rigorous Nonlinear Synthesis of Energy Systems for Seasonal Energy Supply and Storage

Yifan Wang[a], Marvin Volkmer[a], Dörthe Franzisca Hagedorn[a], Christiane Reinert[a], and Niklas von der Assen[a,*]

[a] Institute of Technical Thermodynamics, RWTH Aachen University, 52062 Aachen, Germany
* Corresponding Author: niklas.vonderassen@ltt.rwth-aachen.de.

ABSTRACT

The synthesis of energy systems necessitates simultaneous optimization of both design and operation across all components within the energy system. In real-world applications, this synthesis poses a mixed-integer nonlinear programming (MINLP) problem, considering nonlinear behaviours such as investment cost curves and part-load performance. The complexity increases further when seasonal energy storage is involved, as it requires temporal coupling of the full time series. Although numerous solution approaches exist to solve the synthesis problems simplified by linearization, methods for solving a full-scale problem are currently missing. In this work, we introduce a rigorous method, RiNSES^4, to manage the nonlinear aspects of energy system synthesis, particularly focusing on long-term time-coupling constraints. RiNSES^4 calculates the upper and lower bounds of the initial synthesis problem in two separate branches. The proposed method yields feasible solutions through upper bounds, while evaluating the solution quality via lower bounds. The solution quality is iteratively enhanced by increasing the resolution for calculating upper bounds and tightening the relaxations for computing lower bounds. Both branches work simultaneously and independently, with their outcomes compared after each iteration within each branch. The iterations continue until a predefined optimality gap is reached. We apply RiNSES^4 to design a photovoltaic and battery energy system, considering the seasonality of both energy supply and demand sides. In comparison with a state-of-the-art commercial solver, RiNSES^4 enables to solve the MINLP synthesis problem with great temporal detail and shows high potential.

Keywords: Mixed-integer nonlinear programming, time series aggregation, linearization, decomposition, relaxation

1. INTRODUCTION

Mathematical modeling and optimization can aid in identifying the optimal design and operation of energy systems, spanning from industrial to international scale. The synthesis problem of an energy system necessitates simultaneous optimization of both design and operation, across all components within the energy system [1]. At the design level, the types and sizes of energy system components are determined. At the operation level, decisions are made regarding the on/off status and load allocations for each time step. In general, the synthesis poses a mixed-integer nonlinear programming (MINLP) problem, taking into account nonlinear behaviors such as investment cost curves at the design level and part-load performance at the operation level [2]. However, solving an MINLP problem is generally challenging due to its intrinsic complexity. The complexity increases further when incorporating seasonal energy supply and storage, as it requires extensive temporal data input as well as temporal coupling of the full time series [3].

In the literature, energy system modelers adopt various approaches to address synthesis problems, aiming to achieve computationally tractable results. Kotzur et al. [4] applied time series aggregation methods to efficiently reduce the size of synthesis problems. Gabrielli et al. [5] and Kotzur et al. [6] further proposed alternative modeling approaches to reduce the complexity of synthesis problems from the temporal coupling aspects. However, such approaches only yield solutions for a simplified

version of the synthesis problem.

To tackle the synthesis problem employing the full time series, Baumgärtner et al. [7] developed the RiSES^4 method. The method is designed to solve the linear synthesis of energy systems with seasonal storage, ensuring a solution with known quality. RiSES^4 integrates time series aggregation [8] with superposition seasonal storage modeling [6] in the synthesis problem, and subsequently solve an operational optimization problem with full time series directly through commercial solvers. RiSES^4 employs a rigorous method [9] for measuring the quality of the resulting solutions. In cases where the resulting operational optimization problems remain computationally challenging, a decomposition-based method, DeLoop, proposed by Baumgärtner et al. [10], could be potentially incorporated to more effectively address the long-term operational optimization of energy systems. RiSES^4 has been applied to design an industrial energy system using a mixed-integer linear programming (MILP) formulation, as well as a national energy system using a linear programming (LP) formulation, demonstrating promising performance [7].

Nevertheless, the application of RiSES^4 is limited to linear synthesis problems. The nonlinear nature of an energy system's synthesis is neglected in advance. As a result, the solution obtained for the linearized problem might be infeasible for the initial nonlinear synthesis problem.

1.1. Contribution of this work

In this work, we propose the RiNSES^4 method, an extension of the RiSES^4 method, specifically designed to address the nonlinear aspects of energy system synthesis.

Similar to the RiSES^4 method, RiNSES^4 independently computes the upper and lower bounds of a synthesis problem, but with an MINLP formulation. For calculating the upper bounds, building upon RiSES^4, we use a linearized problem with aggregated time series to find design candidates. These design candidates are then fixed, and the initial MINLP problem is solved as an operational optimization problem to obtain a feasible solution, thereby establishing an upper bound for the initial problem. For computing the lower bounds, we relax various constraints within the synthesis problem to accelerate computation. We iteratively improve the solution quality by increasing the resolution for calculating upper bounds and tightening the relaxation for computing lower bounds. To assess the performance of the proposed method, we apply RiNSES^4 to design an energy system including photovoltaic panels and a battery for seasonal energy supply and storage.

The structure of this paper is as follows: Section 2 explicates the problem statement and the RiNSES^4 method. In Section 3, the proposed method is applied to a case study. Section 4 concludes the work.

2. METHOD

2.1. General MINLP formulation

Problem 1

$$\min_{\dot{E}_k^{\mathrm{N}},E_k^{\mathrm{N}},\dot{E}_{\tilde{e},t}^{\mathrm{buy}},\dot{E}_{\tilde{e},t}^{\mathrm{sell}},\dot{E}_{k,e,t}^{\mathrm{in}},\dot{E}_{k,e,t}^{\mathrm{out}},E_{k,e,t},\mathbf{y},\mathbf{z}} \quad TAC = \frac{1}{\mathrm{APVF}}\,CAPEX + OPEX \tag{1a}$$

with

$$CAPEX = \sum_{k\in\mathcal{K}} I_k = \sum_{k\in\mathcal{K}\setminus\mathcal{K}^{\mathrm{stor}}} I_k^{\mathrm{ref}}\left(\frac{\dot{E}_k^{\mathrm{N}}}{\dot{E}_k^{\mathrm{N,ref}}}\right)^{M_k} + \sum_{k\in\mathcal{K}^{\mathrm{stor}}} I_k^{\mathrm{ref}}\left(\frac{E_k^{\mathrm{N}}}{E_k^{\mathrm{N,ref}}}\right)^{M_k} \tag{1b}$$

$$OPEX = \sum_{k\in\mathcal{K}} c_k^m I_k + \sum_{t\in\mathcal{T}} \Delta t_t \sum_{\tilde{e}\in\mathcal{E}^{\mathrm{ext}}} \left(c_{\tilde{e},t}^{\mathrm{buy}}\dot{E}_{\tilde{e},t}^{\mathrm{buy}} - c_{\tilde{e},t}^{\mathrm{sell}}\dot{E}_{\tilde{e},t}^{\mathrm{sell}}\right) \tag{1c}$$

s.t.

$$\sum_{k\in\mathcal{K}}\left(\dot{E}_{k,e,t}^{\mathrm{out}} - \dot{E}_{k,e,t}^{\mathrm{in}}\right) + \dot{E}_{e,t}^{\mathrm{buy}} - \dot{E}_{e,t}^{\mathrm{sell}} = \dot{E}_{e,t}^{\mathrm{D}} \qquad \forall e\in\mathcal{E},\forall t\in\mathcal{T} \tag{1d}$$

$$E_{k,e,t}\left(1-\eta_{k,e}^{\mathrm{self}}\Delta t_t\right) + \Delta t_t\left(\eta_{k,e}^{\mathrm{in}}\dot{E}_{k,e,t}^{\mathrm{in}} - \frac{\dot{E}_{k,e,t}^{\mathrm{out}}}{\eta_{k,e}^{\mathrm{out}}}\right) = E_{k,e,t+1} \qquad \forall k\in\mathcal{K}^{\mathrm{sto}},\forall t\in\mathcal{T} \tag{1e}$$

$$E_{k,e,t=1} = E_{k,e,t=|\mathcal{T}|+1} \qquad \forall k\in\mathcal{K}^{\mathrm{sto}} \tag{1f}$$

$$\mathbf{g}\left(\dot{E}_k^N, E_k^N, \mathbf{y}, \mathbf{z}\right) \le 0 \qquad \forall k\in\mathcal{K},\forall e\in\mathcal{E} \tag{1g}$$

$$\mathbf{h}\left(\dot{E}_k^N, E_k^N, \dot{E}_{e,t}^{\mathrm{buy}}, \dot{E}_{e,t}^{\mathrm{sell}}, \dot{E}_{k,e,t}^{\mathrm{in}}, \dot{E}_{k,e,t}^{\mathrm{out}}, E_{k,e,t}, \mathbf{y}, \mathbf{z}\right) \le 0 \qquad \forall k\in\mathcal{K},\forall e\in\mathcal{E},\forall t\in\mathcal{T} \tag{1h}$$

$$\dot{E}_k^N, E_k^N, \dot{E}_{e,t}^{\mathrm{buy}}, \dot{E}_{e,t}^{\mathrm{sell}}, \dot{E}_{k,e,t}^{\mathrm{in}}, \dot{E}_{k,e,t}^{\mathrm{out}}, E_{k,e,t} \in \mathbb{R}^+ \qquad \forall k\in\mathcal{K},\forall e\in\mathcal{E},\forall t\in\mathcal{T} \tag{1i}$$

$$\mathbf{y}\in R^{N_y}, \mathbf{z}\in\{0,1\}^{N_z} \qquad \forall k\in\mathcal{K},\forall e\in\mathcal{E},\forall t\in\mathcal{T} \tag{1j}$$

In Problem 1, we state the generic synthesis problem of an energy system using an MINLP formulation. To start with, we assign all components inside the energy system ($k \in \mathcal{K}$) to two categories: storage components ($k \in \mathcal{K}^{\text{sto}}$) and conversion components ($k \in \mathcal{K} \backslash \mathcal{K}^{\text{stor}}$). Storage components transport products ($e \in \mathcal{E}$) from one time step to other time steps ($t \in \mathcal{T}$), while conversion components convert one product to any other product(s). Additionally, the category *"exogenous inputs"* indicates that the energy system is connected with exogenous energy supply ($\tilde{e} \in \mathcal{E}^{\text{ext}} \subseteq \mathcal{E}$) and demand ($\dot{E}^{\text{D}}_{\tilde{e},t}$).

We minimize the total annualized costs TAC that consists of the capital and operational expenditures, $CAPEX$ and $OPEX$ (Equation (1a)). The annualized present value factor is defined as $\text{APVF} = \frac{(1+\text{i})^{\text{n}}-1}{(1+\text{i})^{\text{n}} \cdot \text{i}}$ with the interest rate i and the number of periods n. $CAPEX$ is the sum of the investment costs of all energy system components, which follow the capacity power law (Equation (1b)). The investment costs I_k of the component k are defined as the reference cost I_k^{ref} multiplied by the ratio of the nominal size $\dot{E}_k^{\text{N}}$ or E_k^{N} and a reference value $\dot{E}_k^{\text{N,ref}}$ or $E_k^{\text{N,ref}}$ to the power of $M_k \leq 1$. Please note that the indicators for conversion and storage components differ in their respective units. $OPEX$ are defined as the sum of maintenance costs and external energy costs of each time step, where c_k^m, $c_{\tilde{e},t}^{\text{buy}}$ and $c_{\tilde{e},t}^{\text{sell}}$ indicate a maintenance factor, and external energy buying and selling prices, respectively (Equation (1c)). Δt_t represents the length of time step t.

Given the input time series, the designed energy system needs to fulfill the energy demands for each time step through energy conversion or by purchasing energy from external energy supplies (Equation (1d)). Here, $\dot{E}_{k,e,t}^{\text{in}}$ and $\dot{E}_{k,e,t}^{\text{out}}$ denote the input and output flows, respectively, of product e at time step t for component k. The state of charge $E_{k,e,t+1}$ of a storage component k at the time step $t+1$ depends on its' state of charge $E_{k,e,t}$ at time step t and the product input and output flow at the time step t, considering the efficiencies of self-discharge $\eta_{k,e}^{\text{self}}$, charging $\eta_{k,e}^{\text{in}}$ and discharging $\eta_{k,e}^{\text{out}}$ (Equation (1e)). The so-called cycling constraint ensures that the product is conserved (Equation (1f)). All other constraints are concisely encapsulated in Equations (1g) and (1h), including the part-load performance of each conversion component. All decision variables and their respective bounds are summarized in Equations (1i) and (1j). The surrogate vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ represent other decision variables that are not specified here, encompassing investment decisions and on/off decisions for each energy system component.

In addressing both the design and operation levels of an energy system synthesis problem, we utilize the index t to differentiate between design variables (without the index t) and operation variables (with the index t). This distinction also extends to constraints: design constraints, which are independent of the number of time steps, and operation constraints, which recur at each time step. This distinction is crucial for the method we propose in the following section.

2.2. RiNSES⁴ Method

The proposed RiNSES⁴ method solves Problem 1 with the full time series. As depicted in Figure 1, RiNSES⁴ handles the initial MINLP problem in two separate branches to compute the upper (illustrated as the green segment in Figure 1) and lower bounds (shown as the blue segment in Figure 1) of the MINLP probem,

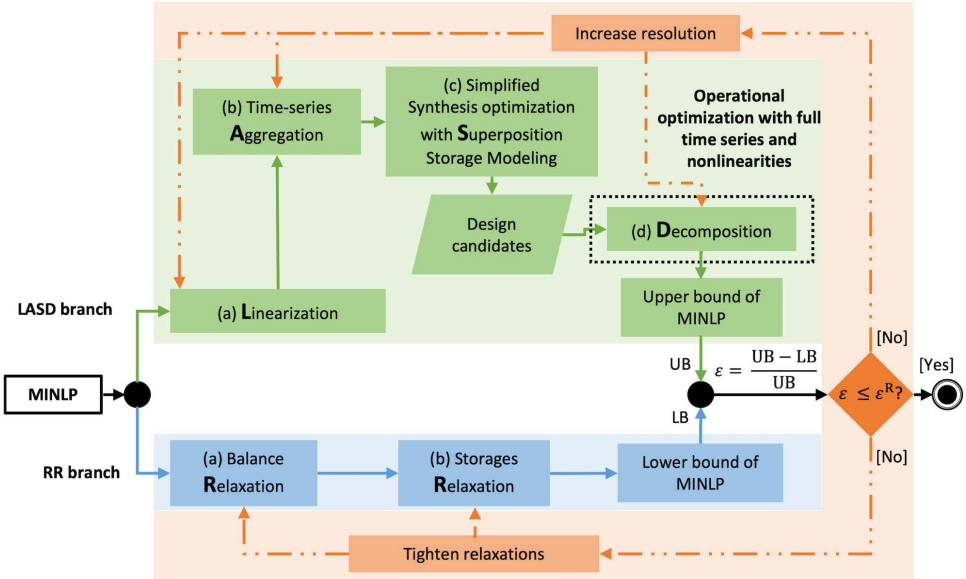

Figure 1: RiNSES⁴ - Rigorous Nonlinear Synthesis of Energy Systems for Seasonal Energy Supply and Storage.

respectively. Both branches incorporate independent iterations (represented by the red segment in Figure 1) to improve the solution quality.

The upper bounds (UBs) are feasible solutions to the MINLP Problem. To calculate an upper bound for Problem 1, we use four distinct techniques: linearizaton (L), time-series aggregation (A), superposition storage modeling (S) and decomposition (D). The tetrad of techniques is collectively referred to as the LASD branch. A detailed description of the LASD branch is provided in Section 2.2.1. The quality of the optimal solution derived from the LASD branch is evaluated using lower bounds (LBs). The lower bounds for Problem 1 are computed via a two-stage relaxation (R) process, named as the RR branch. A detailed discussion of the RR branch is provided in Section 2.2.2. After each iteration, RiNSES[4] compares the current UB and LB to calculate the current optimality gap ε, as described in detail in Section 2.2.3. We iteratively improve the solution quality by increasing the resolution for calculating UBs and tightening the relaxations for computing LBs. The two branches work simultaneously and individually from each other. The RiNSES[4] method terminates if a predefined optimality gap ε^R is reached.

2.2.1. Calculating the upper bounds in LASD branch

In the LASD branch, feasible solutions (upper bounds) of the initial synthesis, Problem 1, are calculated based on four steps, as represented in the green segment in Figure 1: We first simplify the synthesis problem in steps (a) and (b) to obtain values for the design variables in step (c). Then, we validate the feasibility of the design variables in an operational optimization problem with the full time series and all nonlinearities in step (d). To efficiently solve the resulting MINLP operational problem, a decomposition approach can be applied in step (d). If there is a feasible solution to the operational optimization problem, the solution is also a feasible solution to Problem 1 and, thus, an upper bound (UB) for Problem 1.

In step (a), we first linearize nonlinearities of the initial MINLP problem to obtain an MILP formulation. The nonlinearities include investment cost curves, as shown in Equation (1b), and nonlinear part-load performance, as outlined in Equation (1h). The linearization is based on the non-separable piecewise-linear optimization approach proposed by Vielma et al. [11], which can be readily integrated using a Python package [12]. Through linearization, we simplify the complexity of the nonlinear constraints. However, the linearization results in an increase in the number of the binary decision variables in the resulting MILP problem, potentially leading to longer computing time.

Therefore, in step (b), we further simplify the synthesis problem through time series aggregation to reduce the size of *exogenous inputs*. As explained in [8] and in [9], the initial input time series $\mathcal{T}$, consisting of N_t time steps, is divided into N_k periods $\mathcal{P}$ with N_j time steps in each period, where $N_k = N_t/N_j$. Through time series aggregation, N_k and N_j are aggregated to N_k' typical periods P' with N_j' segments in each typical period, i.e., the number of time steps is reduced to $N_k' \times N_j'$, which is significantly smaller than N_t. In this work, we employ the tsam Python package [13] for aggregating time-series data, taking advantage of its various available aggregation approaches. Kotzur et. al [4] analyzed the impact of various aggregation approaches on their optimization results, revealing that the choice of the aggregation approach had only minor impacts. In our study, we hence adopt the k-mean aggregation, a prevalent approach in energy system optimization, for time series aggregation.

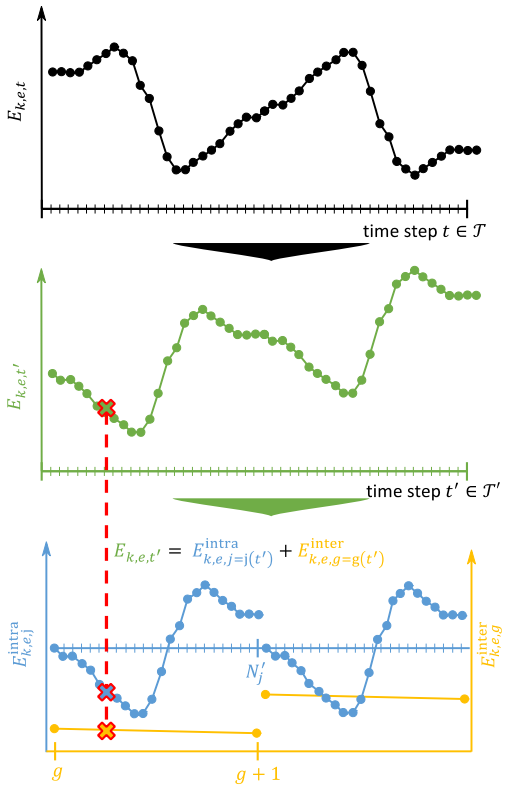

Figure 2: Overview of the superposition storage modeling. The figure is adapted from Kotzur et al. [6].

Upon completing steps (a) and (b), we derive a simplified synthesis problem in step (c). Here, to consider product transport on a seasonal scale, we adopt the superposition storage modeling approach, initially introduced by Kotzur et al. [6] and also implemented in RiSES[4] [7]. In our work, we extend the model by incorporating typical segments. Figure 2 illustrates the basic principles of this superposition modeling approach.

In step (c), the state of charge associated with the initial input time series $E_{k,e,t}$ ($t \in \mathcal{T}$), illustrated in the top

part of Figure 2, is replaced with the state of charge for the aggregated time series $E_{k,e,t'}$ ($t' \in \mathcal{T}'$), shown in the middle part of Figure 2. The superposition modeling approach decomposes $E_{k,e,t'}$ (in green) into two parts: the intra-period state of charge $E^{\text{intra}}_{k,e,j}$ ($j \in \{1, \dots, \mathrm{N}'_{\mathrm{j}}\}$, colored in blue) and inter-period state of charge $E^{\text{inter}}_{k,e,g}$ ($g \in \{1, \dots, \mathrm{N}'_{\mathrm{g}}\}$, colored in orange), depicted in the bottom part of Figure 2. $E^{\text{intra}}_{k,e,\mathrm{j}}$ describes the storage behavior within the typical periods, whereas $E^{\text{inter}}_{k,e,\mathrm{g}}$ represents the storage behavior between these periods. The relationships $j = \mathrm{j}(t')$ and $g = \mathrm{g}(t')$ denote the corresponding values of j and g associated with the time step t'.

Please note that in superposition storage modeling, the sequence of time steps is of paramount importance. Therefore, we use the symbol N'_g to denote the typical periods, rather than N'_k mentioned earlier. The key difference between N'_g and N'_k is that N'_g encompasses the sequence of occurrence of typical periods, and, thus, has the same size as the periods $\mathcal{P}$ previously described. For an in-depth explanation of superposition storage modeling approach, please refer to references [6, 7].

As a consequence, for the *exogenous inputs*, the initial full time series $\mathcal{T}$ with N_t time steps is reduced to the aggregated time series $\mathcal{T}'$ with $N'_g \times N'_j$ time steps. In the following, we refer to the linearized and aggregated synthesis problem with superposition storage modeling as Problem 2. In Problem 2, Equations (2a) to (2d) are applied as operation constraints for all storage components, replacing the constraints (1e) and (1f).

$$\frac{E^{\text{intra}}_{k,e,\mathrm{j}+1} - E^{\text{intra}}_{k,e,\mathrm{j}}}{\Delta t_j} = -\eta^{\text{self}}_{k,e} E^{\text{intra}}_{k,e,\mathrm{j}} + \Delta t_j \left(\eta^{\text{in}}_{k,e} \dot{E}^{\text{in}}_{k,e,j} - \frac{\dot{E}^{\text{out}}_{k,e,j}}{\eta^{\text{out}}_{k,e}} \right) \tag{2a}$$

$$E^{\text{inter}}_{k,e,g+1} = E^{\text{inter}}_{k,e,g} \left(1 - \eta^{\text{self}}_{k,e} \Delta t_t\right)^{N_j} + E^{\text{intra}}_{k,e,N'_j+1} \tag{2b}$$

$$E^{\text{inter}}_{k,e,1} = E^{\text{inter}}_{k,e,N'_g+1} \tag{2c}$$

$$E_{k,e,t'} = E^{\text{intra}}_{k,e,j=\mathrm{j}(t')} + E^{\text{inter}}_{k,e,g=\mathrm{g}(t')} \tag{2d}$$

Problem 2, formulated as an MILP problem with a small-scale input time series $\mathcal{T}'$, facilitates an efficient solution process in step (c). Based on the results of Problem 2, we fix the design variables of the original MINLP synthesis problem to the solution of Problem 2, receiving an MINLP operational optimization problem. The resulting operational optimization problem with the full time series is a large-scale MINLP problem (Problem 3).

In step (d), we solve the resulting Problem 3. Depending on its size, we first try to solve Problem 3 using a commercial MINLP solver. If Problem 3 cannot be solved within a predetermined computing time frame, we employ the DeLoop method, developed by Baumgärtner et al. [10]. DeLoop handles time-coupled long-term operational optimization problems via decomposition and parallel computing, and it systematically reduces the number of decomposed subproblems in an iterative manner. In the worst-case scenario, the original MINLP operational problem is tackled.

If Problem 2 provides feasible design decisions for Problem 3, where the full time series and nonlinearities at the operation level is addressed, we yield an upper bound (UB) to the initial synthesis problem by combining the design decisions of Problem 2 and the operation decisions of Problem 3. However, we cannot guarantee that the design decisions identified in Problem 2 allows feasible operation in Problem 3. If an infeasibility occurs, we iteratively repeat steps (b)-(d) with enhanced time series resolutions, until a feasible solution is found, or the original size of the input time series is applied in step (b). If no feasible solution could be found during aggregation, the resulting superposition storage modeling in step (c) are equivalent to Equations 1(e) and 1(f) in the original synthesis problem. If all attempted steps still result in infeasibility, the final move is to increase the number of breakpoints during linearization. As the number of breakpoints increases, so does the accuracy of the linearization. In the case of infinitely many breakpoints, the linearized problem closely approximates the original MINLP synthesis problem. Should this scenario arise, the LASD branch addresses the original MINLP synthesis problem.

With increasing time series resolution and accuracy of linearization, the whole LASD branch converges to the original MINLP synthesis problem in the worst-case scenario. In practice, we observed that the iterative approach consistently identified a feasible solution before necessitating the use of the full-scale time series in Problem 2. Thus, this work has not yet explored the aspect of infinite number of breakpoints. During implementation, we define a maximum number of breakpoints, for example, four. When the LASD branch reaches this maximum, it forcibly transitions to tackling the original MINLP synthesis problem.

2.2.2. Computing the lower bounds in RR branch

In the RR branch, we underestimate Problem 1 using two-stage relaxations and obtain a relaxed synthesis problem. Solving the relaxed synthesis problem provides a lower bound, which serves as the lower bound for Problem 1, as visually represented by the blue segment in Figure 1.

In stage (a), we employ the same time series aggregation methods as those in the LASD branch, which leads to typical periods P' with aggregated segments S within each typical period. In each segment, we identify the maximal and minimal external energy demands $\dot{E}^{\mathrm{D,\,max}}_{e,t}$ and $\dot{E}^{\mathrm{D,\,min}}_{e,t}$ for each energy form ($\forall e \in \mathcal{E}$). Subsequently, we relax the constraints by replacing Equation (1d) with Equations (3a) and (3b). Please note that, unlike the LASD branch, we maintain the initial size and sequence of the input time series $\mathcal{T}$ in this stage. The number of

decision variables of the synthesis problem therefore remains unchanged, only the solution space is larger.

$$\sum_{k\in\mathcal{K}}\left(\dot{E}^{\text{out}}_{k,e,t}-\dot{E}^{\text{in}}_{k,e,t}\right)+\dot{E}^{\text{buy}}_{e,t}-\dot{E}^{\text{sell}}_{e,t}\le\dot{E}^{\text{D,max}}_{e,t} \tag{3a}$$

$$\sum_{k\in\mathcal{K}}\left(\dot{E}^{\text{out}}_{k,e,t}-\dot{E}^{\text{in}}_{k,e,t}\right)+\dot{E}^{\text{buy}}_{e,t}-\dot{E}^{\text{sell}}_{e,t}\ge\dot{E}^{\text{D,min}}_{e,t} \tag{3b}$$

In stage (b), we further underestimate Problem 1 by relaxing the constraints associated with the storage components. We replace Equations (1e) and (1f) with Equations (4a) and (4b) for each time step ($\forall t\in\mathcal{T}$), which effectively decouples the constraints between two adjacent time steps. We refer to the resulting large-scale MINLP problem as Problem 4, which we solve directly using an MINLP solver. Upon solving Problem 4, its lower bound is a lower bound to the initial synthesis problem (Problem 1).

$$E_{k,e,t}\left(1-\eta^{\text{self}}_{k,e}\Delta t_t\right)+\Delta t_t\left(\eta^{\text{in}}_{k,e}\dot{E}^{\text{in}}_{k,e,t}-\frac{\dot{E}^{\text{out}}_{k,e,t}}{\eta^{\text{out}}_{k,e}}\right)\ge 0 \tag{4a}$$

$$E_{k,e,t}\left(1-\eta^{\text{self}}_{k,e}\Delta t_t\right)+\Delta t_t\left(\eta^{\text{in}}_{k,e}\dot{E}^{\text{in}}_{k,e,t}-\frac{\dot{E}^{\text{out}}_{k,e,t}}{\eta^{\text{out}}_{k,e}}\right)\le E^{N}_{k} \tag{4b}$$

In the RR branch, we expand the solution space by relaxing different constraints of Problem 1, potentially resulting in an increased computing time for Problem 4. Consequently, we employ computing time as an additional termination criterion, and adopt the lower bound of Problem 4 as the lower bound for Problem 1. If Problem 4 is infeasible, the RR branch proceeds to the next iteration.

For new iterations, we gradually revert the relaxations: starting with step (b) for each storage component, followed by step (a) for each energy balance equation. This sequential tightening of underestimation continues until, in the worst-case scenario, the original synthesis problem is fully addressed.

2.2.3. Optimality gap and Iteration

Finally, as marked by the red segment in Figure 1, we compare the best resulting lower and upper bounds obtained using Equation (5) and verify whether the predefined optimality gap ε^{R} is satisfied:

$$\varepsilon=\frac{TAC^{\text{UB}}-TAC^{\text{LB}}}{TAC^{\text{UB}}}\le\varepsilon^{\text{R}} \tag{5}$$

RiNSES[4] identifies global optimal solutions for the initial MINLP synthesis problem (Problem 1), TAC^{UB}, with a known quality TAC^{LB}. It assesses the progress by comparing the current optimality gap ε against the required optimality gap ε^{R}. RiNSES[4] solves Problem 1 in the LASD branch and the RR branch with different levels of simplification. If the achieved optimality gap ε fails to meet the required optimality gap ε^{R}, new iterations are triggered in both branches.

In the LASD branch, we enhance the time step resolution for time series aggregation in step (b) and, potentially, the number of breakpoints for linearization in step (a). Should Problem 2 or Problem 3 exhibit infeasibility, the computing process proceeds directly to the next iteration. In the RR branch, we undo the relaxations gradually, first for storage components in step (b), then for energy balance equations in step (a). Should Problem 4 encounter infeasibility, the computing process proceeds directly to the next iteration.

The iterations stop as soon as the required optimality gap ε^{R} is reached. The current best upper bound is a feasible solution to the initial nonlinear synthesis problem, Problem 1, with known quality. Since both the LASD and RR branches converge to the original MINLP problem in the worst-case scenario, the RiNSES[4] method guarantees the convergence to the initial MINLP synthesis problem (Problem 1).

3. CASE STUDY

3.1. Set-up

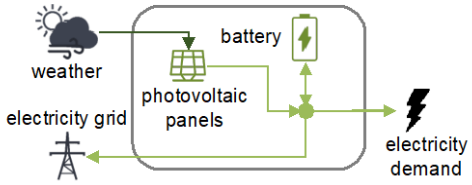

Figure 3: Structure of a PV-BAT energy system.

To evaluate the effectiveness of the proposed method, we apply RiNSES[4] to design a small energy system for seasonal energy supply and storage. We compare the method to an MINLP commercial solver. Thus, we choose a system small enough to be solvable by the MINLP commercial solver as well. As depicted in Figure 3, the energy system includes a conversion component (photovoltaic panels, PV), a storage component (battery, BAT), and three exogenous data inputs (weather, electricity selling price and demand). The renewable electricity generated by photovoltaic panels can be directly used

Table 1: Model parameters of the case study for the energy system components at the design level. The energy system components include photovoltaic panels (PVs) and a battery (BAT).

$k\in\mathcal{K}$	I^{ref}_k	$\dot{E}^{\text{N,ref}}_k$	M_k	c^m_k	$\dot{E}^{N}_k$
PVs	4264.3	1 kW_{el}	0.9592	0.01	$[100, 20000]$ in kW_{el}
	I^{ref}_k	$E^{\text{N,ref}}_k$			E^{N}_k
BAT	2116.1	1 kWh_{el}	0.8382	0.025	$[40, 600000]$ in kWh_{el}

to fulfill the electricity demand, stored in a battery, or fed into the electricity grid.

In the case study, we assume an interest rate $i = 8\%$ and an investment period n of 10 years to calculate the annualized present value factor. The model parameters and the nominal size bounds at the design level are from the literature [14] and summarized in Table 1. For details on the model at the operation level and input time series, please refer to our previous work [15]. Please note that this work adopts a 4-hour resolution, representing the average values derived from every four time steps of the input time series used in our previous research [15]. Additionally, electricity selling prices here are based on the electricity buying prices cited in literature [15] .

We set the predefined optimality gap ε^{R} to 5%. All resulting MILP problems are solved by the commercial solver Gurobi [16] with an optimality gap of 1%, whereas all MINLP problems are handled using the commercial solver BARON 22.11.3 [17] with an optimality gap of 0.2%. We model the energy system on the COMANDO platform [12] using Python 3.8. All computational tasks are carried out on Intel® Xeon® W-2155 processors with 3.3GHz and 128 GB RAM.

3.2. Results and discussion

The initial synthesis problem of the energy system contains 26,284 constraints and 17,524 decision variables. We address this synthesis problem with two solution approaches:

- Using the state-of-the-art MINLP global solver, BARON 22.11.3 [17], and
- Implementing our proposed method RiNSES^4.

Figure 4 shows the results, illustrating the computing times for the upper and lower bounds of both, RiNSES^4 (blue lines) and BARON (green lines). The black dotted line represents the global optimal solution to the MINLP problem. Additionally, as explained in Section 2, the RiNSES^4 method independently calculates the upper and lower bounds, resulting in varying computing times for each.

In our case study, RiNSES^4 finds the first feasible solution within 150 seconds, which is nearly 20 times faster than BARON (2844 seconds). In addition, the initial feasible solution of RiNSES^4 is of high quality, closely approaching the optimal solution, in contrast to that of BARON. When comparing the upper bounds with the optimal solution, the relative error of the first feasible solution of RiNSES^4 is already within the predefined optimality gap ε^{R} 5%.

In terms of the lower bounds, RiNSES^4 calculates the initial lower bound in 1444 seconds, which is twice as fast as BARON (2844 seconds). However, the initial lower bounds of both methods are significantly distant from the optimal solution. While BARON improves its lower bound in 2885 seconds, RiNSES^4 requires considerably more time, achieving a better lower bound in 6649 seconds.

The solver BARON achieves the predefined solution quality in 6518s, with an optimality gap of 2.59%, whereas RiNSES^4 needs a slightly longer computing time of 6649s to reach an optimality gap of 4.97%. This is due to the slower improvement of lower bounds in the RR branch. All in all, RiNSES^4 can find good feasible solutions to MINLP energy system synthesis problems faster than the state-of-the-art solver BARON, even if the proof of optimality takes longer for the regarded case study.

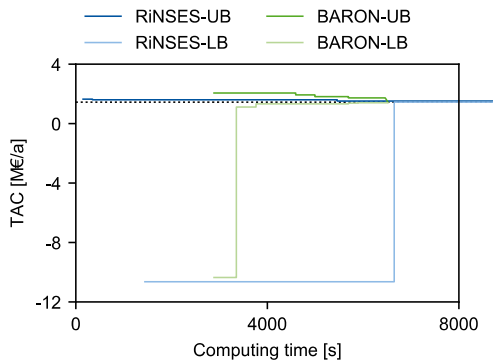

Figure 4: Lower and upper bounds of RiNSES^4 and BARON as function of the computing time for designing a PV-BAT energy system.

4. CONCLUSION

The synthesis of energy systems for seasonal energy supply and storage initially results in large-scale MINLP problems, which are computationally challenging and often simplified and reformulated as (MI)LP problems to enhance computational tractability.

This work introduces the RiNSES^4 method, specifically designed to address nonlinearity in energy system synthesis. RiNSES^4 features two separate branches to under- and overestimate the initial MINLP problem simultaneously and independently. The method provides feasible solutions by upper bounds, which involve linearization, aggregation, superposition storage modeling and decomposition. For computing lower bounds, which are crucial for assessing solution quality, two-stage relaxations are utilized.

RiNSES^4 is applied to design a photovoltaic and battery energy system, with the results evaluated in computational studies. In comparison to the commercial solver BARON, the proposed RiNSES^4 method finds the initial optimal solution very quickly, albeit with a higher optimality gap.

The RiNSES^4 method is generally applicable to two-stage, time-dependent synthesis problems with coupling decision variables and constraints in complex energy systems. However, the method's approach for computing lower bounds, which relies on two-stage relaxations,

requires further enhancement. One potential enhancement could involve integrating nonlinear relaxation into the RR branch. This integration could help to effectively underestimate the nonlinear constraints and reduce computing time by solving MILP formulations within the RR branch. Overall, the foundational design and adaptability of RiNSES[4] render it a promising tool for advancing the field of nonlinear synthesis of energy systems for seasonal energy supply and storage.

DIGITAL SUPPLEMENTARY MATERIAL

The digital supplementary material of this work can be found in a Git repository: https://git-ce.rwth-aachen.de/ltt/ptg-es4.

ACKNOWLEDGEMENTS

This study is funded by the 'Europäischer Fonds für regionale Entwicklung (EFRE)' (EFRE-0801844). The support is gratefully acknowledged.

REFERENCES

1. Lin F, Leyffer S, Munson T. A two-level approach to large mixed-integer programs with application to cogeneration in energy-efficient buildings. *Computational Optimization and Applications* 65: 1–46 (2016).
2. Goderbauer S, Bahl B, Voll P, Lübbecke ME, Bardow A, Koster AMCA. An adaptive discretization MINLP algorithm for optimal synthesis of decentralized energy supply systems. *Computers & Chemical Engineering* 95: 38–48 (2016).
3. Kotzur L, Nolting L, Hoffmann M, Groß T, Smolenko A, Priesmann J, Büsing H, Beer R, Kullmann F, Singh B, Praktiknjo A, Stolten D, Robinius M. A modeler's guide to handle complexity in energy systems optimization. *Advances in Applied Energy* 4: 100063 (2021).
4. Kotzur L, Markewitz P, Robinius M, Stolten D. Impact of different time series aggregation methods on optimal energy system design. *Computers & Chemical Engineering* 117: 474–487 (2018).
5. Gabrielli P, Gazzani M, Martelli E, Mazzotti M. Optimal design of multi-energy systems with seasonal storage. *Applied Energy* 219: 408–424 (2018).
6. Kotzur L, Markewitz P, Robinius M, Stolten D. Time series aggregation for energy system design: Modeling seasonal storage. *Applied Energy* 213: 123–135 (2018).
7. Baumgärtner NJ, Temme F, Bahl B, Hennen MR, Hollermann DE, Bardow A. RiSES4: Rigorous Synthesis of Energy Supply Systems with Seasonal Storage by relaxation and time-series aggregation to typical periods. *Proceedings of the International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS 2019)* 263–274 (2019).
8. Bahl B, Söhler T, Hennen M, Bardow A. Typical periods for two-stage synthesis by time-series aggregation with bounded error in objective function. *Frontiers in Energy Research* 5: 35 (2018).
9. Baumgärtner N, Bahl B, Hennen M, Bardow A. RiSES3: Rigorous Synthesis of Energy Supply and Storage Systems via time-series relaxation and aggregation. *Computers & Chemical Engineering* 127: 127–139 (2019).
10. Baumgärtner N, Shu D, Bahl B, Hennen M, Hollermann DE, Bardow A. DeLoop: Decomposition-based Long-term operational optimization of energy systems with time-coupling constraints. *Energy* 198: 117272 (2020).
11. Vielma JP, Ahmed S, Nemhauser G. Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions. *Operations Research* 58: 303–315 (2010).
12. Langiu M, Shu DY, Baader FJ, Hering D, Bau U, Xhonneux A, Müller D, Bardow A, Mitsos A, Dahmen M. COMANDO: A Next-Generation Open-Source Framework for Energy Systems Optimization. *Computers & Chemical Engineering* 152: 107366 (2021).
13. Hoffmann M, Kotzur L, Stolten D. The Pareto-optimal temporal aggregation of energy system models. *Applied Energy* 315: 119029 (2022).
14. Baumgärtner N, Delorme R, Hennen M, Bardow A. Design of low-carbon utility systems: Exploiting time-dependent grid emissions for climate-friendly demand-side management. *Applied Energy* 247: 755–765 (2019).
15. Wang Y, Bornemann L, Reinert C, von der Aßen N, A Power-to-Gas energy system: modeling and operational optimization for seasonal energy supply and storage. In: *33rd European Symposium on Computer Aided Process Engineering*. Ed: Elsevier (2023).
16. Gurobi Optimization LLC. *Gurobi Optimizer Reference Manual*. (2022).
17. The Optimization Firm LLC. https://minlp.com/baron-solver.