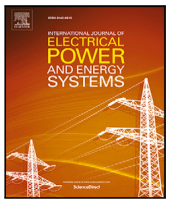
Electrical Power and Energy Systems 168 (2025) 110674

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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



Debunking the speed-fidelity trade-off: Speeding-up large-scale energy models while keeping fidelity

Diego A. Tejada-Arango [a,d,*], Juha Kiviluoma [c,e], Germán Morales-España [a,b]

[a] Energy & Materials Transition, Netherlands Organisation for Applied Scientific Research (TNO), The Netherlands
[b] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology (TUDelft), The Netherlands
[c] Design and Operation of Energy Systems, VTT Technical Research Centre of Finland Ltd., Finland
[d] Instituto de Investigación Tecnológica, Escuela Técnica Superior de Ingeniería, Universidad Pontificia Comillas, Spain
[e] Nodal-Tools Ltd., Finland

ARTICLE INFO

Keywords:
Energy sector coupling
Optimisation modelling
Energy system models
Linear programming (LP)
Computational efficiency

ABSTRACT

Energy system models are essential for planning and supporting the energy transition. However, increasing temporal, spatial, and sectoral resolutions have led to large-scale linear programming (LP) models that are often (over)simplified to remain computationally tractable—frequently at the expense of model fidelity. This paper challenges the common belief that LP formulations cannot be improved without sacrificing their accuracy. Inspired by graph theory, we propose to model energy systems using *energy assets* (vertices), as a single building-block, and *flows* to connect between them. This reduces the need for additional components such as nodes and connections. The resulting formulation is more compact, without sacrificing accuracy, and leverages the inherent graph structure of energy systems. To evaluate performance, we implemented and compared four common modelling approaches varying in their use of building blocks and flow representations. We conducted experiments using TulipaEnergyModel.jl and applied them to a multi-sector case study with varying problem sizes. Results show that our single-building-block (1BB-1F) approach reduces variables and constraints by 26% and 35%, respectively, and achieves a 1.27x average speedup in solving time without any loss in model fidelity. The speedup increases with problem size, making this approach particularly advantageous for large-scale models. Our findings demonstrate that not all LPs are equal in quality and that better reformulations can lead to substantial computational benefits. This paper also aims to raise awareness of model quality considerations in energy system optimisation and promote more efficient formulations without compromising fidelity.

1. Introduction

1.1. Motivation

Although currently available models can integrate the energy sector coupling in the models in different ways, the implications of having flexibility in the modelling choices have not been given much attention. Computational performance is affected by these modelling choices, especially in large-scale energy problems like those in regional-level studies (e.g., European case studies) for long-term expansion planning (e.g., pathway planning). As the energy modelling community is increasingly exploring new system configurations to integrate different processes, such as green hydrogen production, hybrid operation between a storage asset and a renewable asset, or small modular reactors producing both electricity and heat, there is also a need for general and flexible modelling structures that can be adapted to new configurations without compromising the model's performance. Therefore, it is important to identify the computational implications of different modelling choices in this new context.

1.2. Solving methods in energy systems modelling

Many researchers view Linear Programming (LP) as the simplest problem representation that cannot be improved without sacrificing accuracy in energy systems modelling. This is a common misconception, which can lead to the creation of large overhead model sizes. There are at least two alternatives to improve the representation without losing accuracy. First, LP models can be reformulated to leverage specific characteristics, such as sparsity, or tailored for specialised algorithms, thus enhancing computational efficiency without sacrificing accuracy. For example, specialised algorithms like the Hungarian Algorithm for

* Corresponding author at: Energy & Materials Transition, Netherlands Organisation for Applied Scientific Research (TNO), The Netherlands.
E-mail address: diego.tejadaarango@tno.nl (D.A. Tejada-Arango).

https://doi.org/10.1016/j.ijepes.2025.110674
Received 5 November 2024; Received in revised form 9 April 2025; Accepted 11 April 2025
Available online 24 April 2025

the assignment problem [1] and the Push-Relabel method for network flow problems [2] often perform better than general LP solvers. These cases suggest that researchers should explore specialised methods to improve efficiency rather than relying solely on advancements in LP solvers or hardware. Second, since researchers widely use LP models, we explore another alternative to improve the formulation that still uses LP: reformulating the problem to exploit the graph nature of energy system models. All models are not the same, and improving the quality of an LP model means that it retains fidelity while solving faster. Hence, proposing a reformulation that can result in more significant advantages as the problem size increases is especially important for large-scale energy problems.

1.3. Flexible modelling in energy system models

Multi-physics energy system models can be formulated in many ways, ranging from process-specific equations to a more generic approach where concepts such as nodes and units represent a wide variety of conversion, production, consumption, transfer, and storage processes [3]. In the process-specific approach, each process is described by its equation or set of equations. Models may have been built this way for historical reasons: at first the intention was to model a specific sector, such as the power system, and only afterwards was the model expanded to other energy sectors. An example is the early open-source model Balmorel [4], which was initially a power system model but has since been expanded to most energy sectors. This trend has also been seen in commercial options like Plexos [5]. Another example is the COMPETES-TNO model [6] that has incorporated hydrogen sector-specific constraints to a power system model. The trend has been driven by the need to model decarbonisation pathways involving several sector coupling technologies, e.g., hydrogen production by electrolysis, electric vehicles, and heat production from electricity [7]. The main drawback of the process-specific approach is that the number of possible interactions between energy system components can become very large, making the model unwieldy to maintain and expand. Besides, there are only a limited number of ways processes can be described in linear programmes (LPs) or mixed-integer linear programmes (MIPs).

Another strategy is to formulate methods between higher-level concepts like nodes, connections and units that act as building blocks (BB) to build a more general/generic energy system. Each BB can then choose an appropriate method for every particular process. This limits the number of formulations to the number of supported methods; however, it provides flexibility to the user on the modelling options. It also means that users can add new process types without writing new code, provided an appropriate method is available in the model. This approach has been used by models like the IRENA FlexTool [8] and, to an extent, SpineOpt [9].

A further perspective is that energy system models can be seen as network graphs that illustrate the connections among various energy assets in different sectors or energy carriers [10]. From this viewpoint, different types of *nodes* typically serve as the basic building blocks that link all the elements, such as producers, converters, storage, and consumers. However, the existing literature has not delved into the potential use of the natural graph structure of energy systems to link the assets directly in order to improve the computational efficiency of models.

1.4. Contribution

The main contributions of this paper are twofold:

1. In this paper, we debunk the misconception that an LP formulation cannot be further improved (sped up) without sacrificing its fidelity. We show four different approaches using different building blocks and compare the computational performance of three of them. Although all the different formulations lead to the same optimal results, they greatly differ in their computational performance, thus demonstrating that the quality of an LP model can be improved while retaining its fidelity.
2. We propose the *Energy Asset* as a single building block, allowing a more direct connection between components by leveraging the graph structure of energy systems. The proposed strategy replaces the traditional building block, nodes, units and connections with only one BB: *Energy Assets* (vertices) and use energy flows (arcs) to connect them. Thus, it inherently avoids unnecessary elements in between with their associated extra constraints and variables, while keeping the full model flexible and speeding up solving times without sacrificing its fidelity.

2. Quality of LP models

A common misconception is that an LP is the simplest representation of a problem for a desired fidelity, and that the model cannot be simplified without compromising the value of its solution [11]. That is, the only way to speed up an LP without losing accuracy is through improvements in computing power (hardware) and LP solvers (software). This belief can lead modellers to inadvertently create large overhead model sizes, assuming the model is very efficient since LP is the most simplified you can go. When the model size is still potentially problematic, the common belief is that the only other option to speed up large-scale LPs is to solve an (over)simplified, smaller model, which sacrifices its fidelity.

However, all models are not the same even if they model exactly the same problem (i.e., same model fidelity). One model can be faster than another under the same hardware and software. They differ in their theoretical model quality, and their quality can be improved so they can solve faster. Improving the quality of a model means that the model is reformulated so that it retains its fidelity while solving faster (using the same software and hardware). Crafting high-quality formulations allows us to increase the model fidelity without increasing solving times or even create higher quality models that solve faster, thus pushing the Pareto front model fidelity vs computational burden.

Although discussing model quality in LP Models is not common, model quality is a well-known concept in mixed-integer programming (MIP). The quality of an MIP model is defined by its tightness, that is, how near is its relaxed LP feasible region to that of the integer one. The tightest possible model (convex hull) can solve an MIP as an LP, greatly lowering the computational burden. However, trying to tighten an MIP formulation often implies increasing its size; hence, there is a trade-off between the tightness and compactness of an MIP.

What then defines the model quality of an LP model? The quality of an LP model is defined by its size. That is, a more compact model, i.e., fewer constraints/variables and non-zeros, has higher quality than a less compact one with the same fidelity, and hence, it is expected to solve faster. Here, we differentiate the model quality with numerical-related issues, which can also slow down solving times. That is, model quality is independent of the data used. Of course, the modeller should be careful when populating the model with data to avoid LP numerical issues, such as degeneracy, numerical stability and ill-conditioning [11].

How can the quality of an LP model be improved? How can it be lowered in size without sacrificing its fidelity? To the best of our knowledge, there are three ways to lower one of the dimensions of a problem. First, the trivial option is to remove equalities. Each equality means that a variable in that equality can be removed by replacing its equivalent in the other constraints. Although this procedure lowers the number of variables, it increases the number of non-zeros. Second, the Fourier–Motzkin procedure eliminates a set of variables, creating another model in which both models have the same solutions over the remaining variables. Fourier–Motzkin procedure could also eliminate constraints if applied to the dual formulation. However, this procedure comes at the expense of producing an often exponential number of

constraints and non-zeros, thus creating worse bottlenecks and slowing down LP-solving times. The third way of LP reformulation involves splitting dense columns into sparser ones [12]. Although this procedure increases the number of constraints and variables, it could speed up solving times when the number of non-zeros is the bottleneck.

Lowering the size of an LP model is not a trivial task since lowering one dimension comes at the expense of sacrificing another dimension, which can become a new bottleneck, potentially damaging the quality of the model instead of improving it. In this paper, we emphasise in a change of paradigm for flexible energy system modelling; we use a single building block, the energy asset, thus fully exploiting the network nature of the system. This reformulation simultaneously lowers all three dimensions of the problem: including the number of variables, constraints, and non-zeros. This higher-quality LP reformulation naturally lowers both model creation and solving times, obtaining more significant advantages as the problem size increases.

3. Energy systems modelling and building blocks for flexible modelling

Energy systems can be represented as directed graphs, where the *vertices* often denote balancing nodes, and the *edges* represent the energy flows between these components. In this paper, we denote the graph's vertices as the energy system's Building Blocks (BB). Depending on their characteristics in the energy system, these building blocks represent *Energy Assets* that can produce, consume, convert, store, or balance energy.

Current state-of-the-art energy systems models use at least two or three building blocks, such as nodes, connections and units. Energy system models often use *nodes* as an additional building block to connect energy assets and establish energy balances [13]. Some models have a storage option in their node balance [8], while others maintain a separate building block for it [14]. Moreover, some models include an extra building block to describe the *connections* between nodes [9]. Although having multiple building blocks to model an energy system seems appealing due to the flexibility to represent various configurations, it often comes with a computational cost that becomes more significant for large-scale problems.

In this paper, we highlight the advantages of using only one building block: energy assets, represented as the vertices of a graph, with the flows between the assets as the arcs. Adopting this approach makes the node concept irrelevant since we directly connect assets. This approach simplifies the graph structure of the energy system, allowing for flexible modelling options that reduce the number of variables and constraints required to model the same problem. This reduction has computational benefits in both creating the optimisation model and the time to solve it, as demonstrated in the experiments discussed in Section 5.

3.1. Generic formulation

While each approach may have more specific constraints and differing notations, we can generalise them all by using the proposed *energy asset* as the only building block. The following formulation makes comparing the formulations in the following sections more straightforward.

$$\text{minimise} \quad \sum_{a \in \mathcal{A}^{\mathrm{i}}} C^{I}_{a} \cdot i_a + \sum_{(a^{\mathrm{from}},a^{\mathrm{to}}) \in \mathcal{F}} \sum_{t \in \mathcal{T}} C^{O}_{(a^{\mathrm{from}},a^{\mathrm{to}})} \cdot f_{(a^{\mathrm{from}},a^{\mathrm{to}}),t} \tag{1}$$

$s.t.$

$$\sum_{(a^{\mathrm{from}},a) \in \mathcal{F}} f_{(a^{\mathrm{from}},a),t} - \sum_{(a,a^{\mathrm{to}}) \in \mathcal{F}} f_{(a,a^{\mathrm{to}}),t} = D_{a,t} \quad \forall a \in \mathcal{A}^{\mathrm{c}}, \forall t \in \mathcal{T} \tag{2}$$

$$\begin{aligned} s_{a,t} = {} & s_{a,t-1|t>1} + S^{0}_{a|t=1} \\ & + \eta^{\mathrm{in}}_{\mathrm{a}} \cdot \sum_{(a^{\mathrm{from}},a) \in \mathcal{F}} f_{(a^{\mathrm{from}},a),t} \\ & - \frac{1}{\eta^{\mathrm{out}}_{\mathrm{a}}} \cdot \sum_{(a,a^{\mathrm{to}}) \in \mathcal{F}} f_{(a,a^{\mathrm{to}}),t} \quad \forall a \in \mathcal{A}^{\mathrm{s}}, \forall t \in \mathcal{T} \end{aligned} \tag{3}$$

$$\eta^{\mathrm{in}}_{\mathrm{a}} \cdot \sum_{(a^{\mathrm{from}},a) \in \mathcal{F}} f_{(a^{\mathrm{from}},a),t} = \sum_{(a,a^{\mathrm{to}}) \in \mathcal{F}} f_{(a,a^{\mathrm{to}}),t} \quad \forall a \in \mathcal{A}^{\mathrm{cv}}, \forall t \in \mathcal{T} \tag{4}$$

$$\sum_{(a,a^{\mathrm{to}}) \in \mathcal{F}} f_{(a,a^{\mathrm{to}}),t} \leq \overline{P}_a \cdot (U^{0}_{a} + i_a) \quad \forall a \in \mathcal{A}, \forall t \in \mathcal{T} \tag{5}$$

$$\sum_{(a^{\mathrm{from}},a) \in \mathcal{F}} f_{(a^{\mathrm{from}},a),t} \leq \overline{P}_a \cdot (U^{0}_{a} + i_a) \quad \forall a \in \mathcal{A}^{\mathrm{s}}, \forall t \in \mathcal{T} \tag{6}$$

$$0 \leq s_{a,t} \leq \overline{S}_a \quad \forall a \in \mathcal{A}^{\mathrm{s}}, \forall t \in \mathcal{T} \tag{7}$$

$$-\overline{F}_{(a^{\mathrm{to}},a^{\mathrm{from}})} \leq f_{(a^{\mathrm{from}},a^{\mathrm{to}}),t} \leq \overline{F}_{(a^{\mathrm{from}},a^{\mathrm{to}})} \quad \forall (a^{\mathrm{from}},a^{\mathrm{to}}) \in \mathcal{F}, \forall t \in \mathcal{T} \tag{8}$$

$$i_a \leq \overline{I}_a \quad \forall a \in \mathcal{A}^{\mathrm{inv}} \tag{9}$$

$$i_a = 0 \quad \forall a \notin \mathcal{A}^{\mathrm{inv}} \tag{10}$$

where:

- $\mathcal{A}$: set of all assets with elements $a \in \mathcal{A}$
- $\mathcal{A}^{\mathrm{i}} \subseteq \mathcal{A}$: subset of investable assets
- $\mathcal{A}^{\mathrm{c}} \subseteq \mathcal{A}$: subset of consumer assets
- $\mathcal{A}^{\mathrm{s}} \subseteq \mathcal{A}$: subset of storage assets
- $\mathcal{A}^{\mathrm{cv}} \subseteq \mathcal{A}$: subset of conversion assets
- $\mathcal{F}$: set of flows between two assets with elements $(a^{\mathrm{from}}, a^{\mathrm{to}}) \in \mathcal{F}$
- $\mathcal{T}$: set of all timesteps with elements $t \in \mathcal{T}$
- i_a: investment unit variable of asset a
- $f_{(a^{\mathrm{from}},a^{\mathrm{to}}),t}$: flow variable from asset a^{from} to asset a^{to} in timestep t
- $s_{a,t}$: storage level variable of asset a in timestep t
- C^{I}_{a}: investment cost parameter of asset a
- $C^{O}_{(a^{\mathrm{from}},a^{\mathrm{to}})}$: operation cost parameter of flow between two assets
- $D_{a,t}$: demand parameter of the consumer asset a in timestep t
- S^{0}_{a}: initial storage level parameter of the storage asset a
- U^{0}_{a}: initial number of units parameter of the asset a
- $\overline{P}_a$: maximum capacity parameter of the asset a
- $\overline{F}_{(a^{\mathrm{from}},a^{\mathrm{to}})}$, $\overline{F}_{(a^{\mathrm{to}},a^{\mathrm{from}})}$: maximum flow between two assets in both directions.
- $\overline{S}_a$: maximum storage capacity parameter of the storage asset a
- $\overline{I}_a$: maximum investment limit parameter of the asset a
- $\eta^{\mathrm{in}}_{\mathrm{a}}$: efficiency parameter of flows going into the asset a
- $\eta^{\mathrm{out}}_{\mathrm{a}}$: efficiency parameter of flows going out the asset a

The objective function (1) minimises the investment cost and the operation variable cost of the flow between two assets. Eqs. (2)–(4) define the balance constraint for consumer, storage, and conversion assets. The capacity limit of the flows (5) is defined by the maximum capacity of the asset, while the maximum charging limit for storage assets is defined by (6). Eq. (7) represents the maximum storage capacity, and (8) defines the bounds for the flow variable. Eqs. (9) and (10) limit the availability of the investment variables for the assets. Annex A shows how to model a DC power flow and unit commitment in the proposed approach. In addition, the TulipaEnergyModel [15] documentation[1] shows a comprehensive overview of the proposed approach.

3.2. Illustrative example

Having flexible modelling options is becoming more relevant in energy system analysis with the appearance of new configurations, such as the hybrid configurations of storage and renewable assets. Fig. 1 shows an example of such a hybrid configuration, where storage can only be charged from solar PV. Still, the storage and solar PV assets can

[1] https://tulipaenergy.github.io/TulipaEnergyModel.jl/dev/40-formulation/.

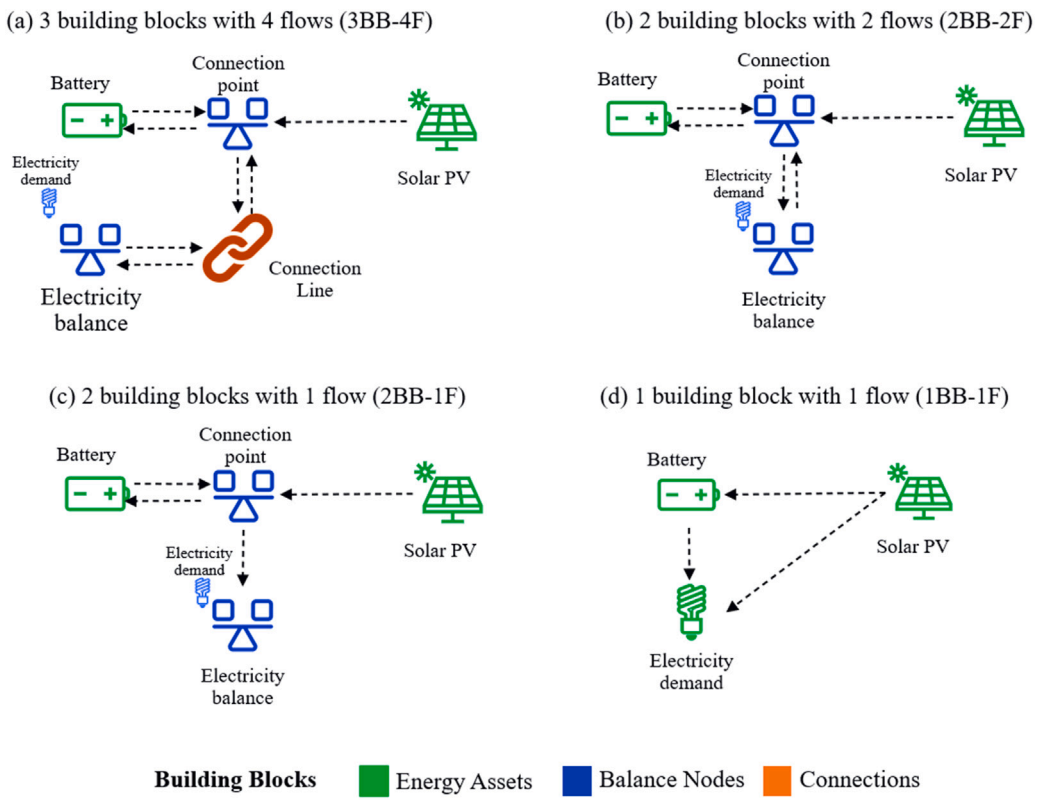


Fig. 1. Using different BB to model a hybrid solar PV with storage. In general, every arrow represents one variable, and every energy asset one constraint.

both deliver energy to the grid. We can take this use case as an example to understand the difference between various approaches, which only considers operation variables for the sake of simplicity . In this case, the storage asset can only be charged from the renewable asset and not from the grid. Fig. 1 gives an overview of each modelling approach for this example. In the following sections, we describe each approach in detail. It is worth noting that different modelling approaches represent the same situation, i.e., they are all modelling exactly the same problem, and this example is not unique. For instance, green hydrogen production is another example where a conversion asset, such as an electrolyser, can only produce hydrogen from the renewable asset; however, the renewable asset can still send energy to the network.

3.2.1. *Three building blocks with four flow variables (3BB-4F)*

This approach uses three BB: energy *assets*, balance *nodes*, and *connections* to link the nodes. In addition, the connection uses four positive variables to link the nodes, two for each node it connects to. This approach is advantageous as it can represent situations where the incoming flow might differ from the outgoing flow, which is needed to model transmission or distribution network losses or gas flows with linepack. However, modelling network constraints without losses or differences between the incoming and outgoing flow creates extra variables and constraints.

Fig. 1(a) shows the battery (bt) with a renewable (pv) example for the 2BB-4F. The connection point (cp) is an auxiliary node connecting the two assets. The electricity demand (ed) is balanced in an extra node. Moreover, the maximum capacity of the flow coming from the connection line (cl) to the connection point must be zero to avoid charging the battery from the grid (i.e., electricity demand balance node). Therefore, we need 8 variables (7 flow variables f + 1 storage level s) and 11 constraints (8 capacity limits + 2 node balances + 1 storage balance). The index t is not included in the following equations for simplicity.

- Connection point node balance constraint:

$$f_{(\mathrm{bt,cp})} + f_{(\mathrm{pv,cp})} + f_{(\mathrm{cl,cp})} = f_{(\mathrm{cp,bt})} + f_{(\mathrm{cp,cl})}$$

- Electricity demand node balance constraint:

$$f_{(\mathrm{cl,ed})} = D_{\mathrm{ed}} + f_{(\mathrm{ed,cl})}$$

- Battery storage balance constraint:

$$s_{\mathrm{bt}} = S^{0}_{\mathrm{bt}} + \eta^{\mathrm{in}}_{\mathrm{bt}} \cdot f_{(\mathrm{cp,bt})} - \frac{1}{\eta^{\mathrm{out}}_{\mathrm{bt}}} \cdot f_{(\mathrm{bt,cp})}$$

- Capacity limit constraints:

$$f_{(\mathrm{bt,cp})} \le P_{\mathrm{bt}}, \quad f_{(\mathrm{pv,cp})} \le P_{\mathrm{pv}}, \quad f_{(\mathrm{cl,cp})} \le F_{(\mathrm{cl,cp})},$$
$$f_{(\mathrm{cp,bt})} \le P_{\mathrm{bt}}, \quad f_{(\mathrm{cp,cl})} \le F_{(\mathrm{cp,cl})}, \quad f_{(\mathrm{cl,ed})} \le F_{(\mathrm{cl,ed})},$$
$$f_{(\mathrm{ed,cl})} \le F_{(\mathrm{ed,cl})}, \quad s_{\mathrm{bt}} \le S_{\mathrm{bt}}$$

- Non-negative variables:

$$f_{(\mathrm{bt,cp})}, f_{(\mathrm{pv,cp})}, f_{(\mathrm{cl,cp})}, f_{(\mathrm{cp,bt})}, f_{(\mathrm{cp,cl})}, f_{(\mathrm{cl,ed})}, f_{(\mathrm{ed,cl})}, s_{\mathrm{bt}} \ge 0$$

where:

$f_{(\mathrm{bt,cp})}$: flow from the battery to the connection point (i.e., battery discharge) limited by battery capacity P_{bt}

$f_{(\mathrm{cp,bt})}$: flow from the connection point to the battery (i.e., battery charge) limited by battery capacity P_{bt}[2]

$f_{(\mathrm{pv,cp})}$: flow from the solar pv to the connection point limited by solar PV availability P_{pv}

$f_{(\mathrm{cl,cp})}$: flow from the connection line to the connection point with a maximum capacity $F_{(\mathrm{cl,cp})}$

[2] Charging and discharging capacities may vary based on the type of storage asset; for simplicity, we consider them equal.

$f_{(cp,cl)}$: flow from the connection point to the connection line with a maximum capacity $F_{(cp,cl)}$

$f_{(cl,ed)}$: flow from the connection line to the electricity demand balance with a maximum capacity $F_{(cl,ed)}$

$f_{(ed,cl)}$: flow from the electricity demand balance to the connection line with a maximum capacity $F_{(ed,cl)}$

s_{bt}: storage level of the battery with a maximum capacity S_{bt}

D_{ed}: electricity demand input data

η_{bt}^{in} and η_{bt}^{out}: charging and discharging efficiencies of the battery

S_{bt}^{0}: initial storage level of the battery

3.2.2. Two building blocks with two flow variables (2BB-2F)

This approach uses the energy *assets* and the balance *nodes* as building blocks. In addition, it uses two positive flow variables to connect the nodes instead of the four variables used in the previous approach. The two flow variables are needed to model transmission or distribution network losses. However, gas flows with linepack cannot be modelled correctly with only two flow variables. Depending on the input data, the incoming and outgoing flow can have values simultaneously, especially when losses are not considered. In such cases, modellers typically use the net value as the transfer between the nodes. Alternatively, they may include a binary variable to prevent simultaneous incoming and outgoing flows, although this would convert the problem into a Mixed-Integer Programming problem.

Fig. 1(b) shows the battery with a renewable example for the 2BB-2F. The connection point is an auxiliary node connecting the two assets. Moreover, the maximum capacity of the flow coming from the electricity balance node to the connection point must be zero to avoid charging the battery from the grid. Therefore, we need 6 variables (5 flow variables + 1 storage level) and 9 constraints (6 capacity limits + 2 node balances + 1 storage balance).

- Connection point node balance constraint:

$$f_{(bt,cp)} + f_{(pv,cp)} + f_{(ed,cp)} = f_{(cp,bt)} + f_{(cp,ed)}$$

- Electricity demand node balance constraint:

$$f_{(cp,ed)} = D_{ed} + f_{(ed,cp)}$$

- Battery storage balance constraint:

$$s_{bt} = S_{bt}^{0} + \eta_{bt}^{in} \cdot f_{(cp,bt)} - \frac{1}{\eta_{bt}^{out}} \cdot f_{(bt,cp)}$$

- Capacity limit constraints:

$$f_{(bt,cp)} \le P_{bt}, \quad f_{(pv,cp)} \le P_{pv}, \quad f_{(ed,cp)} \le F_{(ed,cp)},$$
$$f_{(cp,bt)} \le P_{bt}, \quad f_{(cp,ed)} \le F_{(cp,ed)}, \quad s_{bt} \le S_{bt}$$

- Non-negative variables:

$$f_{(bt,cp)}, f_{(pv,cp)}, f_{(ed,cp)}, f_{(cp,bt)}, f_{(cp,ed)}, s_{bt} \ge 0$$

where the new variables are:

$f_{(cp,ed)}$: flow from the connection point to the electricity demand balance with a maximum capacity $F_{(cp,ed)}$

$f_{(ed,cp)}$: flow from the electricity demand balance to the connection point with a maximum capacity $F_{(ed,cp)}$

3.2.3. Two building blocks with one flow variable (2BB-1F)

This approach uses the energy *assets* and the balance *nodes* as building blocks. In addition, it uses a single free variable to represent the flow between nodes, which can take positive and negative values instead of two positive variables as in the previous approach. The main advantage of this method is that it eliminates the possibility of bidirectional flow between two nodes, as there is only one variable. However, it is not possible to model transmission or distribution network losses using a single free variable for the flow. Lastly, the free variable must have bounds in both directions to represent the capacity limits between the two nodes.

Fig. 1(c) shows the battery with a renewable example for this approach. The connection point is an auxiliary node connecting the two assets. Moreover, the maximum capacity of the flow in the direction from the electricity balance node to the connection point must be zero to avoid charging the battery from the grid. Therefore, we need 5 variables (4 flow variables + 1 storage level) and 9 constraints (6 capacity limits + 2 node balances + 1 storage balance).

- Connection point node balance constraint:

$$f_{(bt,cp)} + f_{(pv,cp)} = f_{(cp,bt)} + f_{(cp,ed)}$$

- Electricity demand node balance constraint:

$$f_{(cp,ed)} = D_{ed}$$

- Battery storage balance constraint:

$$s_{bt} = S_{bt}^{0} + \eta_{bt}^{in} \cdot f_{(cp,bt)} - \frac{1}{\eta_{bt}^{out}} \cdot f_{(bt,cp)}$$

- Capacity limit constraints:

$$f_{(bt,cp)} \le P_{bt}, \quad f_{(pv,cp)} \le P_{pv}, \quad f_{(cp,bt)} \le P_{(cp,bt)},$$
$$-F_{(ed,cp)} \le f_{(cp,ed)} \le F_{(cp,ed)}, \quad s_{bt} \le S_{bt}$$

- Non-negative variables:

$$f_{(bt,cp)}, f_{(pv,cp)}, f_{(cp,bt)}, s_{bt} \ge 0$$

3.2.4. One building block with one flow variable (1BB-1F)

This approach only uses the energy *assets* as building blocks. Using the graph-theory principles establishes the connection between energy assets as vertices and energy flows as edges. Connecting assets directly to each other (without any intervening nodes) can significantly reduce the number of variables and constraints required to represent the system. When transmission or distribution network losses are significant, two conversion assets can represent them—one for each direction with input–output ratios. A similar approach can be adopted for gas pressure flows. Therefore, this method enables us to easily model simple and complex situations, reducing the model size while retaining the same accuracy.

Fig. 1(d) shows the battery with a renewable example for the 1BB-1F approach. Since the assets can connect among them, the battery can directly charge from renewable, and there is no need for an extra constraint to avoid charging from the grid. Therefore, we need 4 variables (3 flow variables + 1 storage level) and 6 constraints (4 capacity limits + 1 demand balance + 1 storage balance).

- Electricity demand node balance constraint:

$$f_{(bt,ed)} + f_{(pv,ed)} = D_{ed}$$

- Battery storage balance constraint:

$$s_{bt} = S_{bt}^{0} + \eta_{bt}^{in} \cdot f_{(pv,bt)} - \frac{1}{\eta_{bt}^{out}} \cdot f_{(bt,ed)}$$

- Capacity limit constraints:

$$f_{(bt,ed)} \le P_{bt}, \quad f_{(pv,bt)} + f_{(pv,ed)} \le P_{pv},$$
$$f_{(pv,bt)} \le P_{bt}, \quad s_{bt} \le S_{bt}$$

- Non-negative variables:

$$f_{(bt,ed)}, f_{(pv,bt)}, f_{(pv,ed)}, s_{bt} \ge 0$$

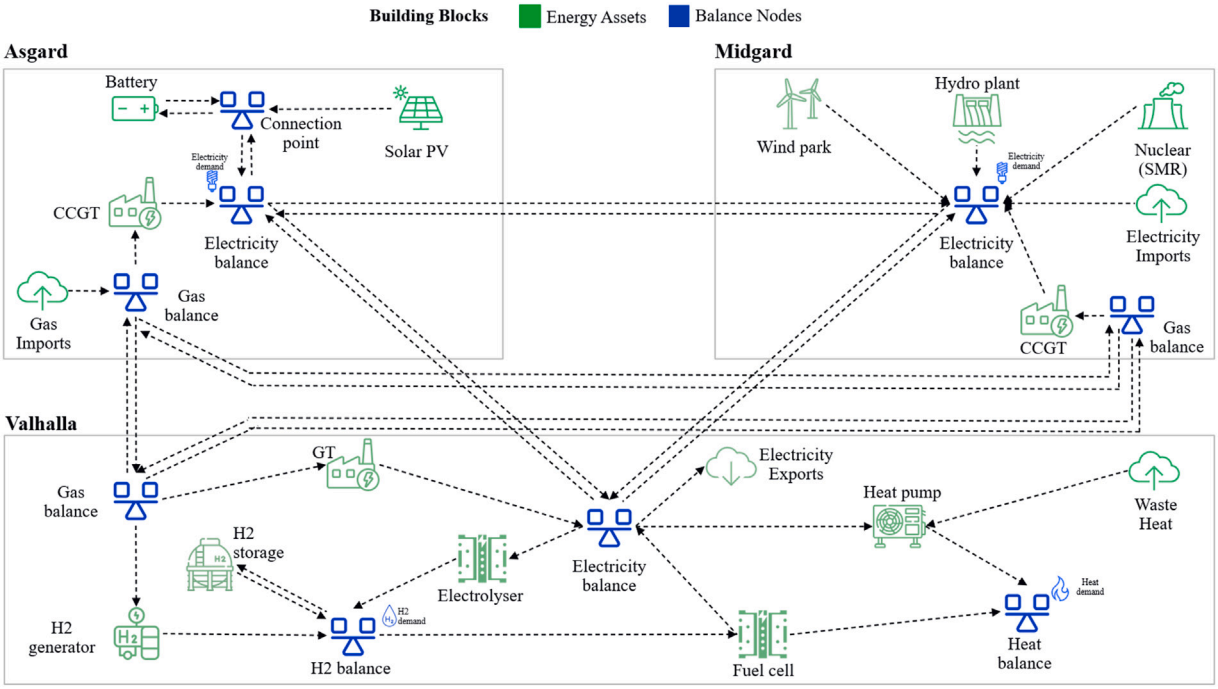


Fig. 2. Case study using 2BB-2F.

Table 1
Number of variables and constraints in each modelling approach per time step.

Modelling approach	Variables	Constraints	Non-zeros
3BB-4F	8	11	18
2BB-2F	6 (↓ 25%)	9 (↓ 18%)	16 (↓ 11%)
2BB-1F	5 (↓ 38%)	9 (↓ 18%)	13 (↓ 28%)
1BB-1F	4 (↓ 50%)	6 (↓ 45%)	9 (↓ 50%)

where the new variables are:

$f_{(\text{bt,ed})}$: flow from the battery to the electricity demand balance limited by battery capacity P_{bt}

$f_{(\text{pv,bt})}$: flow from the solar pv to the battery limited by battery capacity P_{bt} and the solar PV availability P_{pv}

$f_{(\text{pv,ed})}$: flow from the solar pv to the electricity demand balance limited by the solar availability P_{pv}

3.2.5. Summary

Table 1 summarises the number of variables and constraints per time step for each modelling approach in the example shown in Fig. 1. It also shows the reduction in variables and constraints, with the 3BB-4F approach as a reference.

Reducing the number of variables and constraints significantly benefits the time taken to build and solve an optimisation problem, as we show in Section 5. Although the solvers' presolve can eliminate unnecessary variables and constraints, we show that modellers can further improve this process by using formulations with fewer variables and constraints while representing the same energy system. Thus speeding up solving times.

4. Case study

The following sections describe the energy system optimisation model and a case study that compares different approaches. For the optimisation model, we have selected TulipaEnergyModel.jl [15] as it can model all the approaches based on the input data. The case study highlights situations where the 1BB-1F approach can leverage its flexibility to connect energy assets; nevertheless, Section 5.3 discusses when this is possible and provides some insights for energy system modellers.

4.1. Energy system optimisation model

TulipaEnergyModel.jl is an optimisation model using 1BB-1F and determines the optimal investment and operation decisions for different types of assets (e.g., producers, consumers, conversion, storage, and transport). It is developed in Julia [16] and depends mainly on the JuMP.jl [17] and Graphs.jl [18] packages. The complete description of the model, its core concepts, mathematical formulation, and tutorials are available in the GitHub documentation of the model.[3]

4.2. Case study description

In our case study, we conducted experiments on three approaches: 2BB-2F, 2BB-1F, and 1BB-1F. We omitted the 3BB-4F approach because the results in Section 5 showed it would have performed worse than the other approaches for the case study. Our focus is on illustrating the performance differences between the three selected approaches.

Fig. 2 shows an illustrative integrated energy system with three interconnected areas: Asgard, Midgard, and Valhalla. The diagram uses the 2BB-2F approach. In addition, it includes the flow and balance of electricity, heat, and gas within a mock-up energy grid to explore different possibilities for flows among energy assets using nodes. Asgard includes a combined cycle gas turbine, a solar photovoltaic installation, and a battery system. Midgard features a wind park, a hydro plant, and a small modular reactor for nuclear power generation. Valhalla focuses on hydrogen as an energy vector, with a hydrogen generator, a hydrogen storage facility, and a fuel cell. Transmission lines and gas pipelines connect the system, allowing energy transfer between areas.

Figs. 3 and 4 show the equivalent energy system for the other two approaches. In general, each arrow represents a variable, and each BB

[3] https://tulipaenergy.github.io/TulipaEnergyModel.jl/stable/.

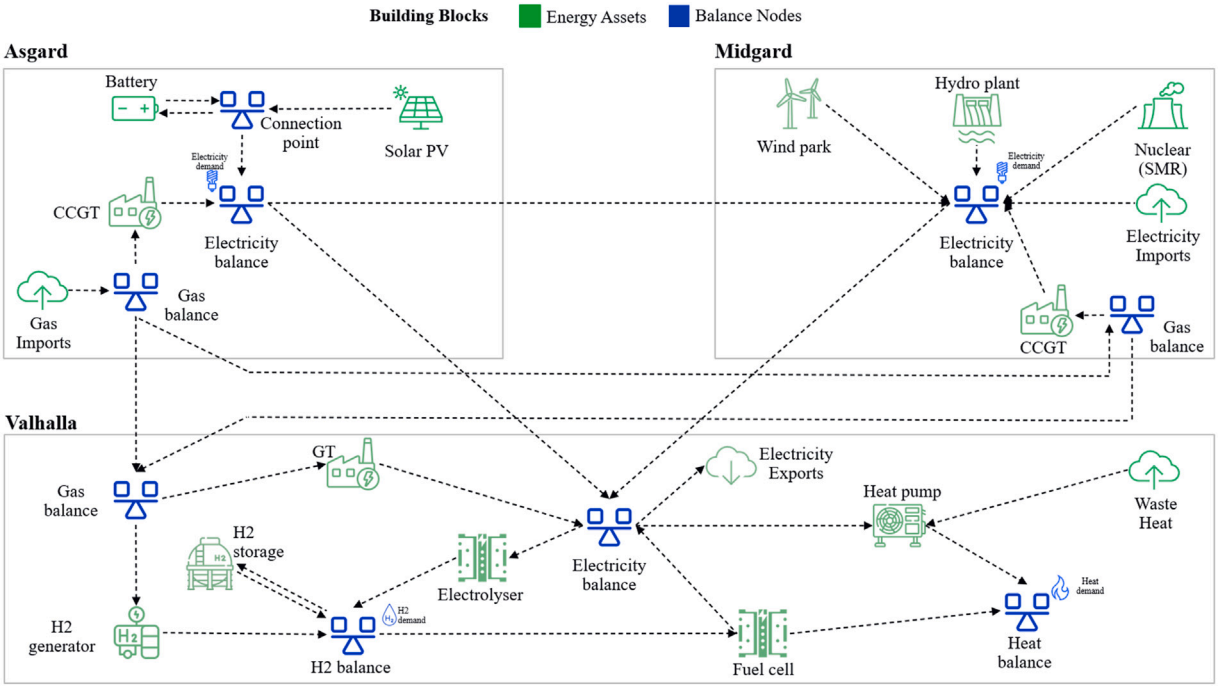


Fig. 3. Case study using 2BB-1F.

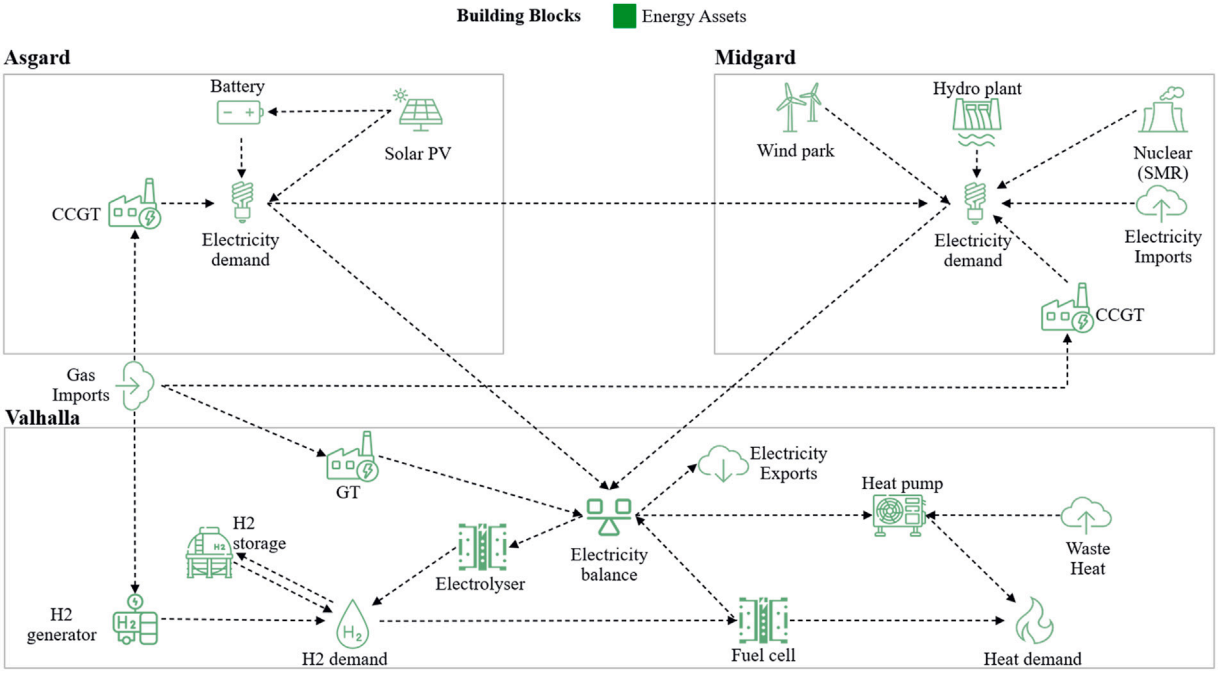


Fig. 4. Case study using 1BB-1F.

element represents a constraint. Note that the number of flows (arrows) reduces compared to the 2BB-2F approach.

To assess the impact of varying problem sizes, we developed six instances of the problem, classified from small to large-scale optimisation. The smallest instance is labelled as 1, while the largest is labelled as 6. Each instance consists of hourly time steps, with differences in the time horizon covered. For example, instance 1 spans one month, whereas instance 6 covers four years.

Section 5 analyses the impact of these reductions in the model for each approach and instance. Finally, Annex B has the link to the repository with the input data files for each approach.

5. Results

The results in this section were obtained using TulipaEnergyModel.jl version 0.6.1 and Gurobi version 11.0.0 on a 12th Gen Intel(R) Core(TM) i7-1255U 1.70 GHz processor and 16.0 GB RAM. Annex B includes the link to the repository with the files to reproduce the experiments in this paper.

Table 2 shows the objective functions and problem size for each instance, where 2BB-2F is used as a reference in the table.

We cover a wide range of model sizes, from smaller instances representing optimisation models with a few thousand variables and

Table 2
Size of the problem in each modelling approach.

Approach	Instance	Obj. func	Variables	Constraints	Nonzeros
2BB-2F	1	2.48E+08	28,908	45,696	96,318
	2	3.55E+08	173,388	274,176	578,609
	3	6.10E+08	376,692	595,680	1,257,055
	4	1.05E+09	753,372	1,191,360	2,514,096
	5	1.48E+09	1,130,052	1,787,040	3,771,168
	6	1.89E+09	1,506,732	2,382,720	5,028,256
2BB-1F	1–6	1 p.u.	↓ 14%	↓ 18%	↓ 17%
1BB-1F	1–6	1 p.u.	↓ 26%	↓ 35%	↓ 29%

constraints to larger instances with millions of variables and constraints.

The reduction in the number of variables and constraints is 14% and 18% for the 2BB-1F approach and 26% and 35% for the 1BB-1F approach. It is important to highlight that the reduction is identical for all instances since the instance is an enlarged version of the same case study. Still, the connection between assets remains unchanged in all instances. It is also worth noting that the objective function is the same for every instance, regardless of the applied approach, indicating that they all represent the same energy problem.

These results demonstrate that reducing the LP size while maintaining fidelity is possible. This finding is significant for practitioners looking to solve large-scale problems more efficiently without losing the desired level of detail.

The graph in Fig. 5 depicts the median speedup values for building and solving models using the three approaches and all the instances. The reference point used for comparison is the 2BB-2F approach. The values for the other approaches indicate the proportion of time the approach takes compared to the reference one. Reducing the number of variables and constraints has advantageous effects on building and solving models. Creating fewer variables and constraints has a significant advantage when building the model. Interestingly, there is a trend towards greater speedups as the instances increase. The surprising result is that having fewer variables and constraints also leads to speedups in the solving phase, with a slight trend towards increasing speedups as the instance size grows. Solvers can leverage these reductions to solve the model faster at each iteration with fewer variables and constraints. It is common practice among energy modellers to rely on the solver's presolve function to eliminate redundant variables and constraints. However, having a clean and concise formulation representing the same energy problem is always better, as it allows the solver to be faster in its process.

5.1. Sensitivity analysis

It is important to comment that commercial optimisation solvers, such as Gurobi, can yield different results based on the "seed" parameter used [19]. Therefore, relying on a single run is not the best procedure for comparing different modelling approaches. Furthermore, the time it takes to build the model can vary due to random variations in CPU processing, which can also lead to differences between each run. To address this, we conducted a sensitivity analysis with 30 different seeds for each approach and optimisation instance, calculating median values for each group to determine if there are any statistically significant differences. We chose the median value as it is a central measure unaffected by data outliers. Fig. 6 shows the distribution of speedups and their means with the 2BB-2F approach as a reference for all results in each instance. Similarly, Fig. 7 shows the results for solving time. Both figures suggest a difference between the 1BB-1F approach and both approaches using two building blocks, i.e., 2BB-2F and 2BB-1F. However, the 2BB-2F and 2BB-1F approaches are closer in the distribution of values. Section 5.2 provides a statistical analysis of the results to determine if there is enough statistical evidence to support these hypotheses.

Table 3
T-test results using 2BB-2F as reference.

Compared approach	Instance	t-statistic	p-value
2BB-1F	1	2.11	0.04
	2	−0.37	**0.72**
	3	0.34	**0.74**
	4	1.74	**0.09**
	5	4.47	0.00
	6	2.60	0.01
1BB-1F	1	6.68	0.00
	2	4.83	0.00
	3	4.78	0.00
	4	9.49	0.00
	5	11.48	0.00
	6	15.08	0.00

Table 4
T-test results using 2BB-1F as reference.

Compared approach	Instance	t-statistic	p-value
1BB-1F	1	6.68	0.00
	2	4.83	0.00
	3	4.78	0.00
	4	9.49	0.00
	5	11.48	0.00
	6	15.08	0.00

5.2. Statistical analysis

To determine if there is a significant difference between the modelling approaches, we use a two-sample t-test. The null hypothesis states that the average values of two related or repeated samples are identical. In contrast, the alternative hypothesis suggests that the underlying distributions of the samples have unequal means.

The variability in the results stems from the random number seed used by the solver, which leads the solver to take different solution paths. Therefore, we can assume for the t-test that the results follow a normal probability distribution, the variances of the results are equal, and each individual in the population has an equal probability of being selected in the sample.

We use 2BB-2F as the reference and compare it individually with 2BB-1F and 1BB-1F. Here, we assume a significance level of $\alpha = 0.05$. Table 3 shows the test results for each instance and compared approach. If the p-value is smaller than the threshold, we reject the null hypothesis of equal averages. Then, there is a statistically significant difference between the mean times of 2BB-2F and the others (2BB-1F and 1BB-1F).

The results presented in Table 3 indicate that 1BB-1F has a significantly different mean value than 2BB-2F in all instances. Additionally, Fig. 7 shows that the average speedup values are higher for 1BB-1F in all cases. This implies that while 2BB-2F could potentially be faster than 1BB-1F in a particular instance, there is statistical evidence that, on average, 1BB-1F is faster. As for 2BB-1F, no statistical evidence suggests it is faster than 2BB-F for half of the instances. However, it is worth noting that the larger case studies in the test, instances 5 and 6, show statistical evidence that the 2BB-1F approach is faster, on average, than the 2BB-2F.

We are left with the question of how the 1BB-1F approach compares to the 2BB-1F approach. We also use a two-sample t-test with the 2BB-1F approach as the reference to compare these approaches. The results in Table 4 indicate that the 1BB-1F approach is statistically faster for all instances, as shown in Fig. 7.

5.3. Discussion

Nowadays, energy system modellers have access to several energy models that can be used to develop case studies. Laveneziana et al. [20] present a critical review of energy planning models, comparing their capabilities and performance. However, this review does not explore

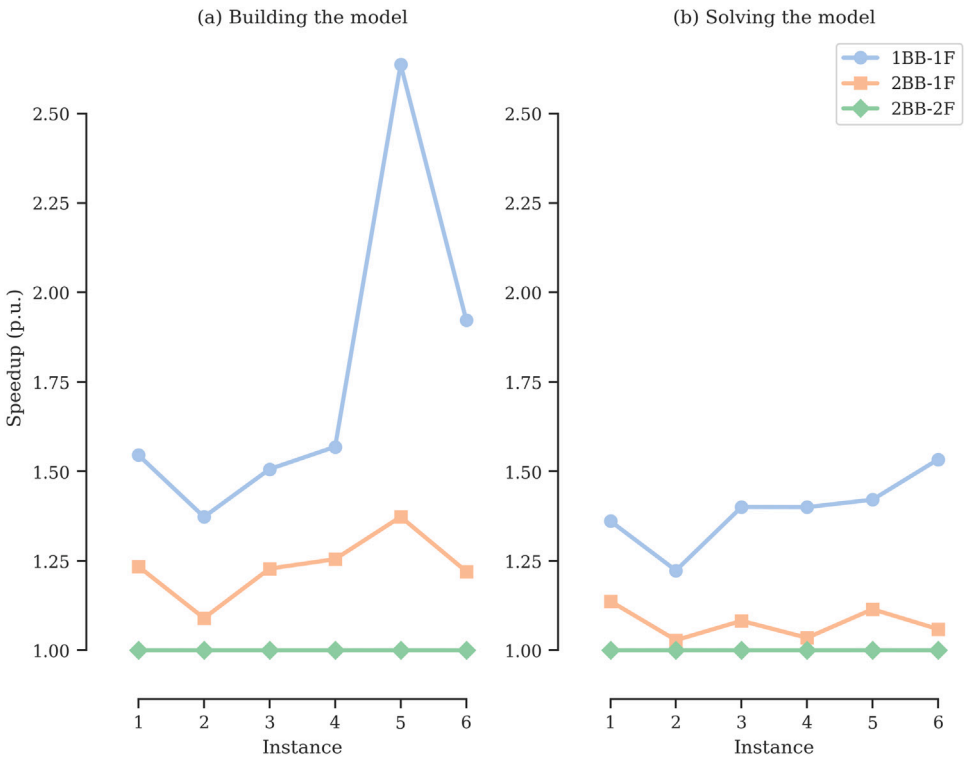


Fig. 5. Speedups comparison.

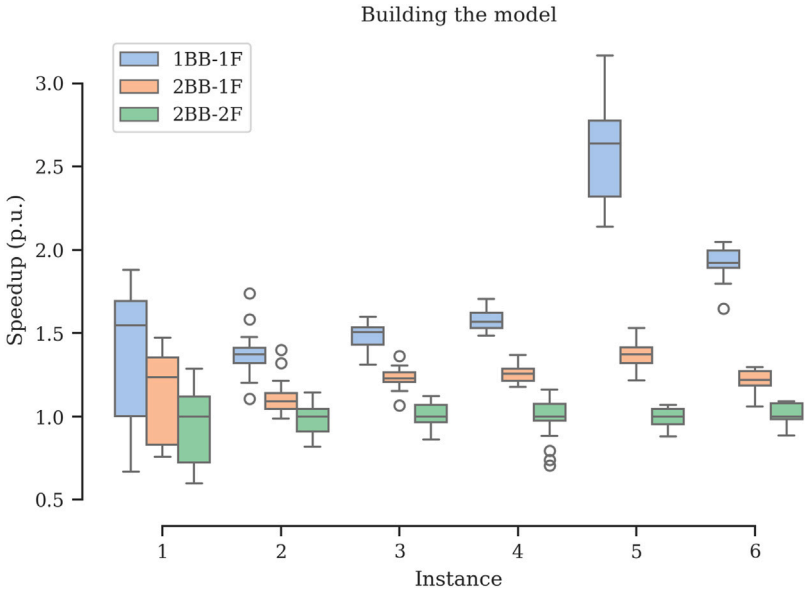


Fig. 6. Speedups distribution comparison for the time to build.

the available modelling options described in this paper. Therefore, Table 5 provides a general overview of the main available modelling approaches in a sample of these energy system models. Models focusing on power systems mainly consist of two building blocks, *nodes* and energy *assets*. In contrast, most recent models that focus on multi-sector analysis, like Calliope and SpineOpt, have three building blocks: *nodes*, energy *assets*, and *connections*. The reason behind having more building blocks is to create a flexible model that can adapt to different energy asset configurations and represent different sectors. However, a flexible model can increase the number of variables and constraints, which can come with a computation cost. The results in the illustrative example in this paper estimate the impact in terms of speedup. As a general recommendation for energy system modellers, it is better to have fewer variables, especially for large-scale case studies. In order to reduce the size of the model, instead of reformulating the model using a single building block as shown in this paper, some models provide the option to select specific methods tailored to the application. These methods also aim to reduce the number of variables and constraints. For instance, IRENA FlexTool gives users a selection of conversion and transfer methods, and some of those methods allow the use of fewer variables and constraints when the model is created, which also improves the solution process in the solver. FlexTool can represent the example problem in 1BB-1F format with one extra variable using

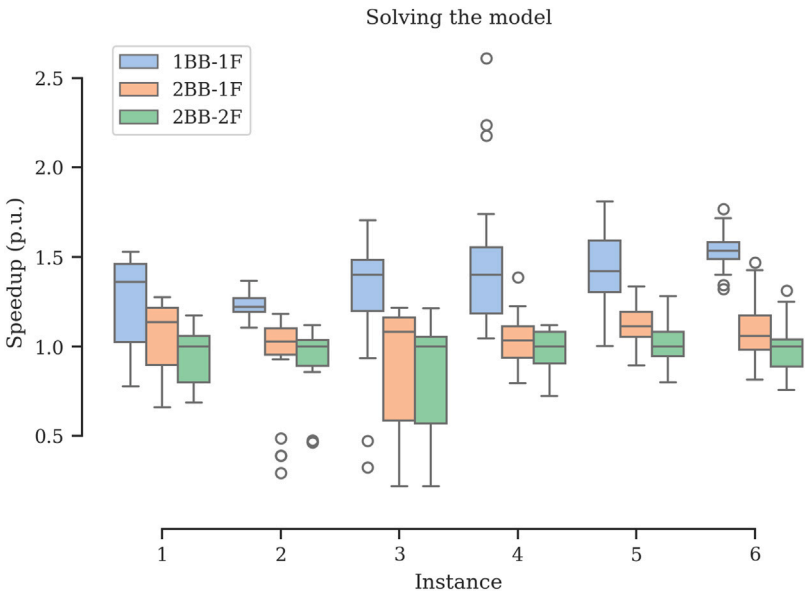


Fig. 7. Speedups distribution comparison for the total time to solve.

available methods and in full 1BB-1F format using carefully formulated user constraints, which can be entered as data.

Another example comes from PyPSA, a 2BB-1F approach model. In PyPSA, the bus serves as the fundamental building block, to which all energy assets — such as loads, generators, storage units, lines, transformers, and links — are connected. This design means that a direct connection between two energy assets is not possible, which distinguishes it from the 1BB-1F model. However, PyPSA provides the option to create custom constraints, allowing for more complex configurations, like those illustrated in the case study, while reducing the total number of constraints required. In both cases, Flextool and PyPSA, the benefit comes from choosing a modelling approach, method, or custom constraint that allows fewer variables and constraints to represent the same energy system. The advantage and strength of the proposed 1BB-1F in this paper is that it is a generic way to formulate the problem without depending on the definition of tailor-made methods or constraints that reduce model size depending on the user settings in a given case study.

The proposal for using only one building block with one flow variable (1BB-1F) is a new way of utilising the network graph structure of energy systems. While this approach has been found to offer several advantages, it also poses two primary challenges and limitations. First and foremost, it involves an extra layer of abstraction as the proposal can establish direct connections between energy assets, which may not be intuitive compared to traditional methods that use more building blocks like nodes and connections. However, this level of abstraction does not hinder the modellers from representing assets that perform the functions of nodes or connections, as evidenced in the case study where the same model, where we used TulipaEnergyModel to represent all the modelling approaches. Secondly, it is not always possible to obtain all the benefits of asset-to-asset connections (1BB-1F) in situations where constraints to model gas pressure, heat transfer, or DCOPF with losses for transmission or distribution networks are considered. Modelling these situations results in variations between an energy asset's incoming and outgoing flows, which partly reduces the potential savings on the model size of the 1BB-1F approach. In such cases, it becomes necessary to include two or four flow variables anyway to represent the situation accurately. Although the proposed 1BB-1F approach allows a direct connection between assets eliminating unnecessary elements in between, with their corresponding variables and constraints, it needs more variables to correctly represent more elaborated flows. Nevertheless, 1BB-1F helps large-scale optimisation problems that can be simplified.

Table 5
Overview of available modelling approaches in a sample of energy system models.

Model	Main focus	Year	Approach
Backbone [21]	Multi-sector	2016	2BB-1F
Balmorel [4]	Multi-sector	2018	2BB-2F
Calliope [22]	Multi-sector	2018	3BB-4F
COMPETES-TNO [23]	Power Systems	2004	2BB-2F
FlexTool [8]	Power Systems	2021	2BB-1F[a]
GenX [24]	Power Systems	2022	2BB-1F
OSEMOSYS [14]	Multi-sector	2016	2BB-2F
Plexos [5]	Multi-sector	2000	2BB-1F
PowerSystems [25]	Power Systems	2021	2BB-1F
PyPSA [13]	Multi-sector	2018	2BB-1F[a]
SpineOpt [9]	Multi-sector	2022	3BB-4F
TIMES [26]	Multi-sector	2016	2BB-2F
TulipaEnergyModel [15]	Multi-sector	2023	1BB-1F

[a] Models with extra methods or custom constraints that allow for reducing the number of variables and constraints, depending on the modelling needs.

6. Conclusion

This paper demonstrates that large-scale LP energy system models can be reformulated to reduce computational burden without compromising fidelity. We introduce a graph-based formulation using a single building block — the energy asset — and show that this approach leads to a more compact and computationally efficient LP formulation. Compared with four traditional approaches that rely on multiple building blocks and connections, our method reduces variables and constraints by 26% and 35%, respectively, and achieves an average speedup of 1.27× in solving time. These improvements scale with problem size, offering substantial benefits for large-scale, high-resolution studies.

Our findings challenge the widespread belief that LP models cannot be improved without a loss of accuracy. Even though solvers can eliminate some redundancy during presolve, we show that providing a compact, well-formulated model leads to faster building and solving times. This is particularly valuable for researchers and practitioners working on regional or multi-sector energy system models.

Beyond technical benefits, this work advocates for more conscious modelling practices. Flexible models should not only accommodate various configurations but do so efficiently. The proposed approach enables scalable, high-fidelity modelling using existing solvers (software) and computers (hardware), making it easier to explore future energy pathways with greater resolution and technical details. We hope this

research encourages the energy modelling community to reconsider assumptions about LP formulations and adopt modelling strategies that balance flexibility, fidelity, and performance.

CRediT authorship contribution statement

Diego A. Tejada-Arango: Writing – original draft, Software, Methodology, Formal analysis. **Juha Kiviluoma:** Writing – review & editing, Funding acquisition, Conceptualization. **Germán Morales-España:** Writing – review & editing, Validation, Methodology, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Grammarly to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Diego A. Tejada-Arango reports financial support was provided by Dutch Research Council. German Morales-Espana reports financial support was provided by Dutch Research Council. Diego A. Tejada-Arango reports financial support was provided by European Climate Infrastructure and Environment Executive Agency. German Morales-Espana reports financial support was provided by European Climate Infrastructure and Environment Executive Agency. Juha Kiviluoma reports financial support was provided by European Climate Infrastructure and Environment Executive Agency. If there are other authors, they 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 research received funding from the European Climate, Infrastructure and Environment Executive Agency under the European Union's HORIZON Research and Innovation Actions under grant agreement N° 101095998. In addition, the Dutch Research Council (NWO) also partially funded this research under grant number ESI.2019.008. **Disclaimer**: Views and opinions expressed are those of the author(s) only and those of the European Union or NWO. Neither the European Union nor the granting authority can be held responsible.

We would like to express our gratitude to the three anonymous reviewers for their valuable feedback, which significantly enhanced the initial version of this manuscript. Additionally, we extend our thanks to Fabian Neumann for his assistance in better understanding PyPSA's formulation and capabilities, which were essential for the comments on PyPSA in this paper.

Annex A. Extended formulation

Section 3.1 shows a basic formulation for the proposed approach. In this section, we illustrate how to include other modelling features; for instance, here we show the equations to model the DC power flow and unit commitment problem.

A.1. DC power flow

For a flow connecting assets a^{from} and a^{to}, which belongs to the set $\mathcal{F}^{\text{dc-opf}}$, the power flow constraints are as follows:

$$f_{(a^{\text{from}},a^{\text{to}}),t} = S^{\text{base}} \cdot \frac{\theta_{a^{\text{from}},t} - \theta_{a^{\text{to}},t}}{X_{(a^{\text{from}},a^{\text{to}})}} \quad \forall (a^{\text{from}}, a^{\text{to}}) \in \mathcal{F}^{\text{dc-opf}}, \forall t \in \mathcal{T} \tag{11}$$

where:

$\mathcal{F}^{\text{dc-opf}} \subseteq \mathcal{F}$: subset of flows that have DC power flow constraints
$\theta_{a,t}$: voltage angle of asset a in timestep t. Notice that this asset is playing a role of a bus in the traditional energy modelling approach.
$X_{(a^{\text{from}},a^{\text{to}})}$: per unit reactance between two assets, a^{from} and a^{to}
S^{base}: base power, e.g., 100 MVA

A.2. Unit commitment

Assets within the subset $\mathcal{A}^{\text{uc}}$ will contain the unit commitment constraints in the model. The following constraints show a basic formulation using the unit commitment variable u for the units on. However, more complex constraints with startup and shutdown variables, e.g., [27], can also be added to the model.

$$\hat{f}_{a,t} = \sum_{(a,a^{\text{to}}) \in \mathcal{F}} f_{(a,a^{\text{to}}),t} - \underline{P}_a \cdot u_{a,t} \quad \forall a \in \mathcal{A}^{\text{uc}}, \forall t \in \mathcal{T} \tag{12}$$

$$u_{a,t} \leq U_a^0 + i_a \quad \forall a \in \mathcal{A}^{\text{uc}}, \forall t \in \mathcal{T} \tag{13}$$

$$\hat{f}_{a,t} \leq \left(\overline{P}_a - \underline{P}_a\right) \cdot u_{a,t} \quad \forall a \in \mathcal{A}^{\text{uc}}, \forall t \in \mathcal{T} \tag{14}$$

$$\hat{f}_{a,t} \geq 0 \quad \forall a \in \mathcal{A}^{\text{uc}}, \forall t \in \mathcal{T} \tag{15}$$

where:

$\mathcal{A}^{\text{uc}} \subseteq \mathcal{A}$: subset of assets with unit commitment
$\hat{f}_{a,t}$: flow above the minimum operating point of asset a in timestep t
$f_{(a,a^{\text{to}}),t}$: flow variable from asset a to asset a^{to} in timestep t
$u_{a,t}$: units on variable of a in timestep t
i_a: investment unit variable of asset a
U_a^0: initial number of units parameter of the asset a
$\overline{P}_a$: maximum capacity parameter of the asset a
$\underline{P}_a$: minimum capacity parameter of the asset a

Eq. (12) defines the flow above the asset's minimum operating point, and (13) sets the limits on the variable units. Eq. (14) specifies the maximum output flow above the minimum operating point, while (15) indicates the minimum output flow above the minimum operating point.

Annex B. Supplementary material

All the code to run the experiments and the raw results in this paper are available at the following link:
https://github.com/datejada/experiments-flexible-connection
The formulation presented in this paper is a simplified version of the TulipaEnergyModel.jl. The complete formulation, including all features of the model, can be found at the following link:
https://tulipaenergy.github.io/TulipaEnergyModel.jl/stable/

Data availability

The data and code are available online in GitHub (links provided in the paper)

References

[1] Kuhn HW. The hungarian method for the assignment problem. In: 50 years of integer programming 1958-2008: from the early years to the state-of-the-art. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 29–47.

[2] Cherkassky BV, Goldberg AV. On implementing the Push—Relabel method for the maximum flow problem. Algorithmica 1997;19(4):390–410. http://dx.doi.org/10.1007/PL00009180.

[3] TNO. Energy system description language. 2020, URL: https://energytransition.gitbook.io/esdl/esdl-concepts/design-principles. [Accessed 19 March 2024].

[4] Wiese F, Bramstoft R, Koduvere H, Pizarro Alonso A, Balyk O, Kirkerud JG, Tveten ÅG, Bolkesjø TF, Münster M, Ravn H. Balmorel open source energy system model. Energy Strat Rev 2018;20:26–34. http://dx.doi.org/10.1016/j.esr.2018.01.003, URL: https://www.sciencedirect.com/science/article/pii/S2211467X18300038.

[5] Energy Exemplar. PLEXOS. 2000, URL: https://www.energyexemplar.com/plexos. [Accessed 11 March 2024].

[6] Morales-España G, Hernández-Serna R, Tejada-Arango DA, Weeda M. Impact of large-scale hydrogen electrification and retrofitting of natural gas infrastructure on the European power system. Int J Electr Power Energy Syst 2024;155:109686. http://dx.doi.org/10.1016/j.ijepes.2023.109686, URL: https://www.sciencedirect.com/science/article/pii/S0142061523007433.

[7] Gea-Bermúdez J, Jensen IG, Münster M, Koivisto M, Kirkerud JG, kuang Chen Y, Ravn H. The role of sector coupling in the green transition: A least-cost energy system development in northern-central Europe towards 2050. Appl Energy 2021;289:116685. http://dx.doi.org/10.1016/j.apenergy.2021.116685, URL: https://www.sciencedirect.com/science/article/pii/S0306261921002130.

[8] Kiviluoma J, Tupala A, Soininen A, International Renewable Energy Agency (IRENA). IRENA FlexTool. 2022, URL: https://github.com/irena-flextool/flextool.

[9] Ihlemann M, Kouveliotis-Lysikatos I, Huang J, Dillon J, O'Dwyer C, Rasku T, Marin M, Poncelet K, Kiviluoma J. SpineOpt: A flexible open-source energy system modelling framework. Energy Strat Rev 2022;43:100902. http://dx.doi.org/10.1016/j.esr.2022.100902, URL: https://www.sciencedirect.com/science/article/pii/S2211467X22000955.

[10] Markensteijn A, Romate J, Vuik C. A graph-based model framework for steady-state load flow problems of general multi-carrier energy systems. Appl Energy 2020;280:115286. http://dx.doi.org/10.1016/j.apenergy.2020.115286, URL: https://www.sciencedirect.com/science/article/pii/S0306261920307984.

[11] Klotz E, Newman AM. Practical guidelines for solving difficult linear programs. Surv Oper Res Manag Sci 2013;18(1):1–17. http://dx.doi.org/10.1016/j.sorms.2012.11.001, URL: https://www.sciencedirect.com/science/article/pii/S1876735412000189.

[12] Lustig IJ, Mulvey JM, Carpenter TJ. Formulating two-stage stochastic programs for interior point methods. Oper Res 1991;39(5):757–70. http://dx.doi.org/10.1287/opre.39.5.757, arXiv:https://doi.org/10.1287/opre.39.5.757.

[13] Brown T, Hörsch J, Schlachtberger D. PyPSA: Python for power system analysis. J Open Res Softw 2018;6(4). http://dx.doi.org/10.5334/jors.188, arXiv:1707.09913.

[14] Howells M, Rogner H, Strachan N, Heaps C, Huntington H, Kypreos S, Hughes A, Silveira S, DeCarolis J, Bazillian M, Roehrl A. OSeMOSYS: The open source energy modeling system: An introduction to its ethos, structure and development. Energy Policy 2011;39(10):5850–70. http://dx.doi.org/10.1016/j.enpol.2011.06.033, URL: https://www.sciencedirect.com/science/article/pii/S0301421511004897. Sustainability of biofuels.

[15] Tejada-Arango DA, Morales-España G, Clisby L, Wang N, Soares Siqueira A, Ali S, Soucasse L, Neustroev G. Tulipa energy model. 2023, URL: https://github.com/TulipaEnergy/TulipaEnergyModel.jl.

[16] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59(1):65–98. http://dx.doi.org/10.1137/141000671.

[17] Lubin M, Dowson O, Dias Garcia J, Huchette J, Legat B, Vielma JP. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. Math Program Comput 2023. http://dx.doi.org/10.1007/s12532-023-00239-3.

[18] Fairbanks J, Besançon M, Simon S, Hoffiman J, Eubank N, Karpinski S. JuliaGraphs/Graphs.jl: an optimized graphs package for the julia programming language. 2021, URL: https://github.com/JuliaGraphs/Graphs.jl/.

[19] Tejada-Arango DA, Lumbreras S, Sánchez-Martín P, Ramos A. Which unit-commitment formulation is best? A comparison framework. IEEE Trans Power Syst 2020;35(4):2926–36. http://dx.doi.org/10.1109/TPWRS.2019.2962024.

[20] Laveneziana L, Prussi M, Chiaramonti D. Critical review of energy planning models for the sustainable development at company level. Energy Strat Rev 2023;49:101136. http://dx.doi.org/10.1016/j.esr.2023.101136, URL: https://www.sciencedirect.com/science/article/pii/S2211467X2300086X.

[21] Helistö N, Kiviluoma J, Ikäheimo J, Rasku T, Rinne E, O'Dwyer C, Li R, Flynn D. Backbone—An adaptable energy systems modelling framework. Energies 2019;12(17). http://dx.doi.org/10.3390/en12173388, URL: https://www.mdpi.com/1996-1073/12/17/3388.

[22] Pfenninger S, Pickering B. Calliope: a multi-scale energy systems modelling framework. J Open Source Softw 2018;3(29):825. http://dx.doi.org/10.21105/joss.00825.

[23] Özdemir Ö, Hobbs BF, van Hout M, Koutstaal P. Capacity vs energy subsidies for renewables: Benefits and costs for the 2030 EU power market. Energy Policy Res Group Univ Camb 2019. URL: http://www.jstor.org/stable/resrep30402.

[24] Jenkins J, Sepulveda N, Mallapragada D, Patankar N, Schwartz A, Schwartz J, Chakrabarti S, Xu Q, Morris J, Sepulveda N. GenX. 2022, http://dx.doi.org/10.5281/zenodo.6229425, URL: https://github.com/GenXProject/GenX.

[25] Lara JD, Barrows C, Thom D, Krishnamurthy D, Callaway D. PowerSystems.jl — A power system data management package for large scale modeling. SoftwareX 2021;15:100747. http://dx.doi.org/10.1016/j.softx.2021.100747, URL: https://www.sciencedirect.com/science/article/pii/S2352711021000765.

[26] Loulou R, Goldstein G, Kanudia A, Lettila A, Remme U, Wright E, Giannakidis G, Noble K. Documentation for the TIMES model - part I. 2016, URL: https://iea-etsap.org/index.php/documentation.

[27] Gentile C, Morales-España G, Ramos A. A tight MIP formulation of the unit commitment problem with start-up and shut-down constraints. EURO J Comput Optim 2016;5(1):177–201. http://dx.doi.org/10.1007/s13675-016-0066-y, URL: https://link.springer.com/article/10.1007/s13675-016-0066-y.