Scheduling Tasks Towards Energy Autarky: Benefits and Computational Costs of Flexibility

Robert Bredereck
Institut für Informatik, TU Clausthal, Germany

Till Fluschnik
Department of Computer Science, Algorithm Engineering Group,
Humboldt-Universität zu Berlin, Germany

Klaus Heeger
Department of Industrial Engineering and Management,
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

We study the *autarky* problem: given an energy forecast, a battery, and a set of energy-consuming jobs with time windows, decide whether all jobs can be scheduled without requiring external energy. We analyze the problem through the lens of job flexibility, defined as the number of time steps at which a job may be scheduled. We show that the problem is NP-hard already for flexibility two, even in restricted settings. On the positive side, we identify settings in which the problem is polynomial-time solvable, even for large flexibilities. Moreover, we obtain fixed-parameter tractability for combined parameters involving flexibility, such as the number of jobs. In contrast, we establish W-hardness when parameterized by maximum flexibility alone, even in a restricted setting. To complement our theoretical results, we formulate an integer linear program (ILP) that computes the minimum required external energy and evaluate it experimentally on instances derived from real-world energy-consumption and radiation data. The experiments indicate that increased job flexibility substantially reduces the need for external energy at moderate computational cost.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Theory of computation → Design and analysis of algorithms; Social and professional topics → Sustainability

Keywords and phrases computational sustainability, NP-hardness, parameterized complexity and algorithms, integer linear programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.10

Related Version *Full Version*: `https://arxiv.org/abs/2607.03260` [7]

Supplementary Material *Software (Code and data)*:
`https://github.com/buhtig-tf/Scheduling-Tasks-towards-Energy-Autarky` [6]
archived at `swh:1:dir:ec658833659409634e2c49460594460b33fb79bc`

Funding *Till Fluschnik*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project “Parameterized Algorithmics in Computational Sustainability (PACS)” – FL 1247/1-1, 522475669.

Acknowledgements We gratefully acknowledge Andreas Reinhardt and Mazen Bouchur (both TU Clausthal, Energy Informatics group) for their continuous guidance throughout the project regarding details and specifics about energy systems.

1 Introduction

Renewable, weather-dependent resources such as solar and wind become increasingly important for energy production. This, in turn, increases the importance of reliable forecasts. Given a reliable forecast of resource availability, energy-consuming jobs (abstracting tasks or

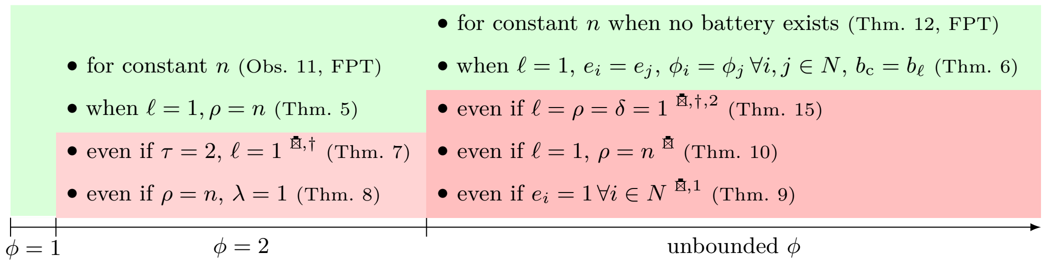


Figure 1 Overview of our results for ABS with $\lambda = \min\{\lambda_{\text{in}}, \lambda_{\text{out}}\}$ (green: P-time; light/darker-red: weak/strong NP-hardness). Some results hold even if : there is no battery; †: the forecast is the same for all time steps. For some restrictions we also have ^{1}: W[1]-h. wrt. $\rho+\delta$; ^{2}: W[1]-h. wrt. τ. Results for tractability propagate to the left, for intractability to the right.

devices, e.g., a kettle or a PC) can be scheduled accordingly to directly consume available power. This applies both at a local level (e.g., solar panels on private properties) and at a global level (e.g., offshore wind farms for industrial use). In this work, we study the following problem: given an energy forecast, a battery, and a set of energy-consuming jobs, how can the jobs be scheduled so as to minimize the required external energy – and, in particular, to decide whether no external energy is required at all. The difficulty stems from the interaction between cumulative energy constraints over time and execution-window constraints, which together create long-range dependencies between scheduling decisions.

Our problem is closely linked to the evaluation of a household's degree of *autarky*. Several approaches to this already exist. What is novel in our approach is that we optimally solve the underlying scheduling problem arising from the fact that jobs often allow some flexibility in their execution. In particular, we show that while flexibility makes the problem computationally hard, it can yield significant energy savings in practice.

The application domain ranges from households to quarters (i.e., neighborhoods) and industrial settings, which changes the view on the problem parameters. For a household, we expect relatively few jobs; for quarters, somewhat more; and for industry, potentially many. Conversely, households and quarters may involve diverse jobs, while industrial jobs may be more homogeneous. Moreover, households and quarters may be less flexible, while industry may allow more flexibility in execution times, focusing primarily on whether a job is completed at all. Hence, to fundamentally understand the complexity of the problem, we perform a classic computational and parameterized complexity analysis.

Our Contributions. We present an elaborate mathematical model and introduce the energy autarky problem AUTARKY BY SCHEDULING (ABS). We conduct an extensive computational and parameterized complexity analysis (see Figures 1 and 2; Figure 3 organizes our results; Sections 2 and 3 survey our notations) and show that ABS becomes NP-hard already when every job has flexibility exactly two, even under further restrictions, such as each job having unit length. Moreover, we show that ABS is fixed-parameter tractable regarding the combined parameter flexibility and the order of the largest connected component of the job graph, which intuitively captures the dependency between jobs. In contrast, we show that ABS is W[1]-hard when parameterized by the number of time steps, and hence by flexibility, even if all jobs have length one and the same release date and deadline. Given the intractability of ABS, we formulate an exact algorithm in form of an integer linear program (ILP). With the ILP at hand, we run experiments on real-world data (household energy consumption and radiation profiles), artificially combined into a total of 48048 instances, all of which are solved

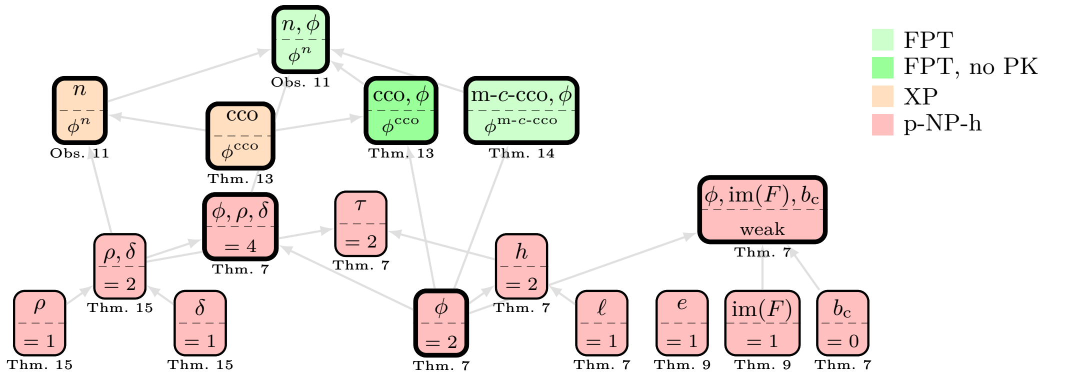


Figure 2 Hasse diagram of our parameters with running times of algorithms or hardness specifics in the lower half of each cell. When a parameter p points to a parameter p', then there is a function f such that $p \leq f(p')$ for all instances. Parameters are combined additively (e.g., p, p' refers to $p + p'$).

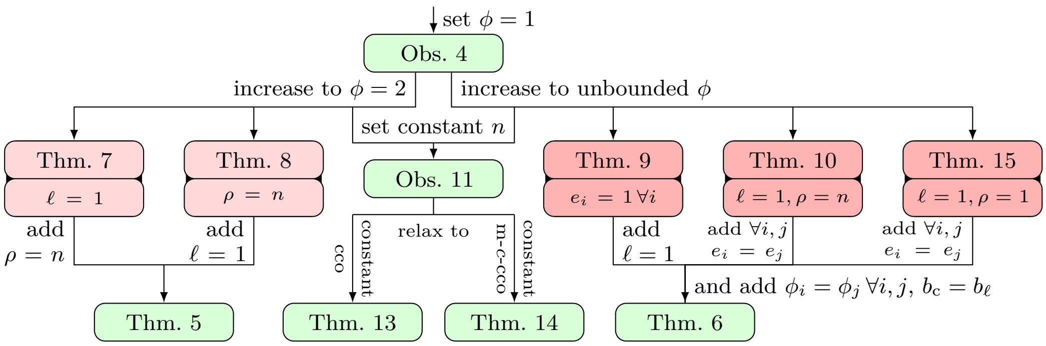


Figure 3 Organization chart of our structural results. Green boxes correspond to polynomial-time solvability, light-red boxes to weak NP-hardness, and (darker) red boxes to strong NP-hardness.

using the ILP. Our results show how job flexibility relates to the minimum required external energy: In a nutshell, larger flexibility can save a significant amount of energy at a moderate average increase of runtime. The experimental evaluation is intended to complement the theoretical analysis by illustrating the practical impact of flexibility, rather than to serve as a benchmarking study.

Given the algorithmic and experimental focus of the presentation and the space constraints, we defer all hardness proofs and provide only proof sketches for our algorithmic results. The omitted proofs and details on results marked with ($\star$) can be found in the full version [7].

2 Preliminaries

We denote by $\mathbb{N}$ and $\mathbb{N}_0$ the natural numbers ex- and including zero, respectively. We denote by $\mathbb{Q}_+$ and $\mathbb{Q}_{\geq 0}$ the set of all positive and non-negative numbers from the rational numbers $\mathbb{Q}$, respectively. For a function f, we denote by $\text{Im}(f)$ the image of f, and by $\text{im}(F) = |\text{Im}(F)|$.

We distinguish between weak and strong NP-hardness in the standard sense. Weak NP-hardness arises from numerical parameters encoded in binary and does not preclude pseudo-polynomial-time algorithms, whereas strong NP-hardness rules out such algorithms unless P = NP. This distinction is relevant in our setting since several hardness results are obtained via reductions from Partition, Bin Packing, or variants with unary encodings.

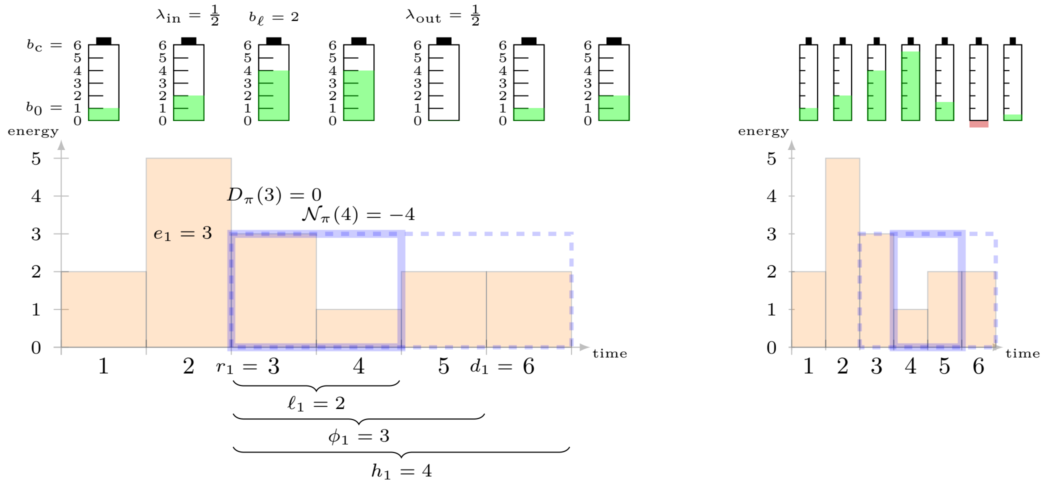


Figure 4 Illustration to Section 3. A simple instance with $\tau = 6$, one job $J_1 = (r_1, d_1, \ell_1, e_1) = (3, 6, 2, 3)$ (blue), battery $B = (b_0, b_c, b_\ell, \lambda_{\text{in}}, \lambda_{\text{out}}) = (1, 6, 2, \frac{1}{2}, \frac{1}{2})$ (top), and orange forecast F. Scheduling J_1 at times 3 and 4, respectively, leads to a (left) feasible schedule π and a (right) infeasible one.

We use basic terminology from parameterized algorithms and complexity [10]. A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}_0$ over a fixed finite-sized alphabet Σ. L is in XP if there are computable functions f, g only depending on p such that every instance $I = (x, p)$ can be decided for L in $f(p) \cdot |I|^{g(p)}$ time. L is fixed-parameter tractable (FPT) if $g \equiv c$ is a constant, i.e., every instance $I = (x, p)$ can be decided for L in $f(p) \cdot |I|^c$ time; we also say that L is in FPT. If L is W[1]-hard, then it is presumably not in FPT. Such hardness is shown via parameterized reductions, basically translating the parameter of the first problem into the parameter of the second problem. A kernelization is a polynomial-time algorithm that transforms each instance $I = (x, p)$ into a decision-equivalent instance $I' = (x', p')$ such that $|I'| \leq f(p)$ for some computable function f only depending on p; if f is a polynomial, then we call it a polynomial kernelization (PK).

3 Model, Problem Definition, and Parameters

Our Model. For a time window $T = \{1, \dots, \tau\}$ of τ time steps, we have an energy forecast $F \colon T \to \mathbb{Q}$ (see Figure 4, accompanying this section). Additionally, we have a set $\mathcal{J}$ of n non-preemptive jobs, where for each $i \in N := \{1, \dots, n\}$ we have a job $J_i = (r_i, d_i, \ell_i, e_i)$ with

$$r_i \in T\colon \text{release date},\quad d_i \in T\colon \text{deadline},\quad \ell_i \in T\colon \text{job length},\quad e_i \in \mathbb{Q}_+\colon \text{energy per time}.$$

▶ Remark 1. Real-world tasks may exhibit time-varying power consumption profiles. In our model, however, each job is assumed to have a constant energy per time over its duration. This abstraction can be interpreted either as replacing the original profile by its mean power (thus preserving total energy) or as a conservative modeling choice that assumes the profile's maximum power throughout the job (thus ensuring feasibility under fluctuations). ◁

Also, we have a *battery* $B = (b_0, b_c, b_\ell, \lambda_{\text{in}}, \lambda_{\text{out}})$ where

$$b_0 \in \mathbb{Q}_{\geq 0}\colon \text{initial level},\quad b_c \in \mathbb{Q}_{\geq 0}\colon \text{capacity},\quad b_\ell \in \mathbb{Q}_{\geq 0}\colon \text{maximum loading speed},$$
$$\lambda_{\text{in}}, \lambda_{\text{out}} \in (0, 1]\colon \text{effective in- and output efficiencies}.$$

Let $(F, \mathcal{J}, B)$ be an instance. A *schedule* $\pi\colon \mathcal{J} \to T$ is an assignment of jobs to starting times such that $\pi(J_i) \geq r_i$ and $\pi(J_i) + \ell_i - 1 \leq d_i$. A job J_j is *active* at time $t \in T$ if $t \in [\pi(J_j), \pi(J_j) + \ell_j - 1]$. We denote by $\widehat{\pi}(t)$ the index set of the jobs active at time t. Let

$$D_\pi(t) = F(t) - E_\pi(t), \text{ where } E_\pi(t) \coloneqq \sum\nolimits_{i \in \widehat{\pi}(t)} e_i, \tag{1}$$

denote the *net* energy at time $t \in T$ given schedule π. Let $\max^0(x) \coloneqq \max\{0, x\}$, $\min^0(x) \coloneqq \min\{0, x\}$, and

$$\mathcal{N}_\pi(t) = \lambda_{\text{in}} \cdot \max^0(D_\pi(t)) + \lambda_{\text{out}}^{-1} \cdot \min^0(D_\pi(t)) \tag{2}$$

denote the *effective* net energy relevant to the battery, that is, transformed by λ_{in} when charged (i.e., $D_\pi(t) > 0$) and by $\lambda_{\text{out}}^{-1}$ when discharged (i.e., $D_\pi(t) < 0$). We denote the battery state at time t given schedule π by $B_\pi(t)$, where $B_\pi(1) = b_0$. The battery state in the current time step is composed of the battery level and the effective net energy of the previous time step, coupled with the maximum loading speed b_ℓ, and upper bounded by the battery's capacity. Formally, for all $t \in \{2, \ldots, \tau + 1\}$ we have

$$B_\pi(t) = \min\left\{b_{\text{c}}, B_\pi(t-1) + \min\{b_\ell, \mathcal{N}_\pi(t-1)\}\right\}. \tag{3}$$

▶ Remark 2. We neglect discharge-rate limitations of the battery. For modern residential battery systems, the maximum discharge power (of several kW, cf. [38]) exceeds the demand of household appliances (cf. [29, Table 5]) and is therefore rarely binding in practice. ◁

A schedule $\pi\colon \mathcal{J} \to T$ is *feasible* if $B_\pi(t) \geq 0$ for all $t \in \{1, \ldots, \tau + 1\}$. Note that we include the auxiliary time step $\tau + 1$ to verify the effective net energy of the last time step τ.

Problem Definition. Our central decision problem is defined as follows.

▶ **Problem 1** (Autarky by Scheduling (AbS)). Given a forecast F, a set $\mathcal{J}$ of jobs, and a battery B, the question is whether there is a feasible schedule.

Given a schedule π, one can compute each of (1), (2), and (3) in polynomial time, thus:

▶ **Observation 3.** AbS *is contained in NP.*

We say that the battery has no losses if $\lambda_{\text{in}} = \lambda_{\text{out}} = 1$. When we say that there is no battery, then we assume a battery with no losses and with $b_{\text{c}} = 0$ (note that feasibility in this case is equivalently defined over $D_\pi(t) \geq 0$ for all $t \in T$).

Further Parameters. The parameters defined below will be central in the subsequent complexity analysis. We denote by $\rho = |R|$ and $\delta = |D|$ the sizes of the sets of unique release times $R \coloneqq \bigcup_{i \in N}\{r_i\}$ and deadlines $D \coloneqq \bigcup_{i \in N}\{d_i\}$, resp. For a job J_i, $i \in N$, we define the *horizon* by $h_i \coloneqq d_i - r_i + 1$, the *seat* $S_i = \{r_i, \ldots, d_i - \ell_i + 1\}$, and, as the size of the seat,

the *flexibility* by $\phi_i \coloneqq h_i - \ell_i + 1$.

Flexibility captures the number of different start times a job can take; $\phi_i = 1$ corresponds to a job with no scheduling flexibility. Note that for all $i \in N$, $\ell_i \leq h_i$ and $\phi_i \leq h_i$, but ℓ_i and ϕ_i are incomparable. Recall that in this work, we focus on the flexibility in our analysis. We drop the subscripts to refer to the maximum over all values, e.g., $\phi \coloneqq \max_{i \in N} \phi_i$ denotes the maximum flexibility over all jobs.

Let $G(\mathcal{J}) = (V, E)$ be the (undirected interval) graph with vertex set $V = \{v_i \mid J_i \in \mathcal{J}\}$ and edge set $E = \{\{v_i, v_j\} \mid [r_i, d_i] \cap [r_j, d_j] \neq \emptyset\}$. We call G the *job graph*. Intuitively, edges in the job graph represent potential overlaps in time windows, which may induce dependencies for scheduling. We consider connected components in G, i.e., inclusion-wise maximal vertex subsets such that in such a subset, every two vertices are reachable from each other via a sequence of consecutively adjacent edges. We denote by $\text{cco}(G)$ the order of the largest connected component of G. A *modulator to* c-cco with constant value c is a vertex set $W \subseteq V$ such that the modified graph $G - W$, i.e., when removing W and all edges with an endpoint in W from G, has cco of size at most c. By m-c-cco we denote the smallest size of a *modulator to* c-cco. The associated problem of computing m-c-cco is also known as c-Component Order Connectivity [25], which is already NP-hard for every $c \geq 1$ [27].

4 Related Work

As usual for problems in computational sustainability, related work stems from two perspectives: from a computational perspective (here: scheduling and parameterized complexity), and from a sustainable perspective (here: demand-response and energy autarky).

Scheduling and Complexity. By now, many scheduling problems have been studied from a parameterized complexity perspective [34], revealing several interesting open problems [33]. Jobs whose costs co-depend on a resource – as in our setting – are comparatively rare. More common are budgets on the total number of jobs [37], rejection costs [18], or weights tied to completion times [3]. A related outsourcing-motivated model considers weighted jobs that partially depend on an external resource which itself incurs a cost [8]; however, jobs have unit length, cannot overlap, and may be overdue. ILP approaches to scheduling are by now well established, including systematic algorithmic investigations [22].

Models without batteries include settings where energy is constantly renewed or globally limited and activities are subject to precedence constraints [9], as well as more general resource-constrained scheduling frameworks studied from a complexity-theoretical viewpoint [12]. Other battery-free models assume no release dates (i.e., all jobs are available at time step 1) together with a common deadline and a no-overlap constraint [5]; in contrast to our work, several resources may be available and jobs may require more than one simultaneously. Further related is single-machine scheduling with processing times and energy demands, where recharging itself may consume time steps, and analyze objectives such as (weighted) completion time or late jobs [48]. Only few works combine energy harvesting with a battery: some restrict to unit-length, weighted jobs without parallel execution and forbid harvesting while processing, aiming to maximize the total weight of feasible jobs, and provide weak NP-hardness, polynomial-time, and approximation results for special cases [43]; related heuristic approaches have also been explored experimentally [24]. An interesting extension to our model is so-called battery care, where the battery must recharge to a minimum level before reuse and may need to be discharged to a prescribed level before the next recharge [14].

Demand-Response. Demand-response (DR) research studies coordination in systems with distributed renewable generation such as photovoltaic (PV) units and batteries across residential [19, 28], hotel [47], community [42], and industrial [32, 39] settings, predominantly aiming at reducing electricity costs, peak demand, or grid imports. In residential PV-battery systems, Lezama et al. [28] formulate an MILP model and solve their problem via evolutionary algorithms; optimality is not guaranteed due to the stochastic approach. Runtimes of one to three minutes are reported. Similarly, Hua et al. [19] introduce an energy consumption scheduler for interruptible and non-interruptible time-flexible appliances. Energy reduction for unscheduled against scheduled scenarios are compared, but only for a fixed flexibility setting. Neither varying degrees of flexibility nor runtimes are analyzed. At larger scales, Wamalwa and Ishimwe [47] propose an MINLP model for a PV-battery-powered hotel building to re-schedule flexible loads (e.g., washing machines, dishwashers, electric stoves) taking the end-user appliance rescheduling inconvenience into account. For energy communities, Sangare et al. [42] present an MILP model that is solved optimally for small and heuristically (potentially suboptimally) for larger instances. Their Type B loads resemble our jobs. Across these PV-battery DR works [19, 28, 42, 47], flexibility is modeled as a property of loads but not systematically varied or analyzed with respect to feasibility or algorithmic behavior. None classifies the associated decision problems in terms of weak or strong NP-hardness, parameterized complexity, or tractability boundaries. In contrast, our work treats flexibility as a central structural parameter. Rather than focusing on solver performance for fixed formulations, we characterize the complexity landscape of autarkic scheduling and analyze how increasing flexibility influences feasibility and computational difficulty.

Finally, we point out that there is a broader notion of "energy flexibility" in the DR literature [16, 31, 44], where flexibility is defined as the ability of an energy network to act in response to external signals, e.g., by temporal shifting of consumption or supply adaptation. To the best of knowledge, these works focusing on "energy flexibility" are more conceptual and neither provide complexity-theoretic analyses nor empirically study how varying flexibility levels affect external energy reductions or runtimes of solution algorithms.

Energy Autarky and PVs. Combining photovoltaic systems (PVs) and batteries on household level is studied in the context of cost-minimization [17], indicator-based self-consumption [4], emergency power supply facing blackouts [46], combination with heat pumps and thermal storage [26], or autarky through decentralized batteries [40]. However, to the best of our knowledge, none of these studies investigate the use of flexible job scheduling to achieve household or community energy autarky.

5 Polynomial-Time Solvable Cases

If every job's flexibility is one and thus no choice is left, then AbS is trivial.

▶ **Observation 4.** AbS *is linear-time solvable if* $\phi = 1$.

We will see that AbS becomes NP-hard already for $\phi = 2$. Thus, tractability persists only under further restrictions, such as the following.

▶ **Theorem 5** ($\star$)**.** AbS *is polynomial-time solvable if* $\phi = 2$*,* $\rho = n$*, and* $\ell = 1$*.*

As we will see, a flexibility of two is crucial here, since AbS becomes strongly NP-hard for unbounded flexibilities even when $\rho = n$ and $\ell = 1$ (Theorem 10). Note that the setup of Theorem 5 implies that all jobs are weakly-ordered by their starting times. We will see that flexibility two and all jobs being weakly-ordered is not enough for tractability (Theorem 8).

Proof sketch. We use dynamic programming: For each job J_j we have two table entries, corresponding to whether J_j is scheduled on r_j or $r_j + 1$, that store the battery state for the next job. Since all release dates are distinct, we can now sweep "from left to right", with at most two jobs' horizons overlapping at any time. Thus, a table entry's update only requires the battery state for the current job, stored for the previous job's two cases. ◀

Finally, we show that if all jobs are equal except for their release times, and the battery can fully recharge in one time step, then the problem becomes polynomial-time solvable.

▶ **Theorem 6** (⋆). ABS *is polynomial-time solvable if* $\ell = 1$*,* $e_i = e_j$ *and* $\phi_i = \phi_j$ *for all* $i, j \in N$*, and* $b_\ell = b_c$*.*

Proof sketch. We run the following greedy algorithm with initially $\mathcal{J}^* = \emptyset$. For $t = 1, \ldots, \tau$ in ascending order, first add all jobs with release date t to $\mathcal{J}^*$ and then, in ascending order of their deadlines, try to schedule at time step t as many jobs as the forecast and battery at t allow. Intuitively, the algorithm is correct in this case for the following reasons. On the one hand, the jobs are sufficiently similar that they can be reordered within a schedule. On the other hand, there is no incentive to delay a job since the battery can recharge from 0 to 100% in one time step. Interestingly, our algorithm as such is incorrect if we drop this requirement of $b_c = b_\ell$ as shown by the example in Figure 11 in the appendix. ◀

6 Hardness Results

We next show that a flexibility of two, as opposed to one, already makes the problem computationally hard. Herein, we observe a hierarchy: for a flexibility of two, we obtain weak NP-hardness, and for larger (unbounded) flexibilities, we obtain strong NP-hardness.

6.1 Weak Hardness for Flexibility Two

We show that ABS is weakly NP-hard even for flexibility two. We give two hardness reductions, both from the well-known (weakly) NP-hard [21] PARTITION problem, where, given a multiset $X = \{x_1, \ldots, x_n\}$ of numbers from $\mathbb{N}$, one asks whether there is a subset X_1 of X such that $\sum_{x_i \in X_1} x_i = \sum_{x_j \in X \setminus X_1} x_j$. We first show that ABS is already hard for two time steps. Intuitively, we can model the numbers in any PARTITION instance with the jobs' energies and the task to partition with flexibility two and length one for each job facing only two time steps overall.

▶ **Theorem 7** (⋆). ABS *without battery is weakly NP-hard even if* $\phi = \tau = 2$*,* $\ell = 1$*,* $\rho = \delta = 1$*, and* $F(1) = F(2)$*.*

Even if all jobs are weakly ordered by their starting times, ABS is computationally challenging.

▶ **Theorem 8** (⋆). ABS *is weakly NP-hard even if* $\phi = 2$*,* $\rho = n$*, and* $\lambda_{\text{in}} = \lambda_{\text{out}} = 1$*.*

Note that $\phi = 2$ and $\rho = n$ imply that all jobs are weakly ordered by their starting times. Recall that by Theorem 5, we know that when additionally $\ell = 1$, ABS is polynomial-time solvable. Interestingly, this is the only reduction in our work that makes use of the battery.

6.2 Strong Hardness for Larger Flexibilities

The distinct reductions behind Theorems 9, 10, 15, and 16 use the same well-known NP- and W[1]-hard [13, 20] problem UNARY BIN PACKING, where, given a multiset $X = \{x_1, \ldots, x_n\}$ of numbers from $\mathbb{N}$ and two integers $k, b \in \mathbb{N}$, all numbers encoded in unary, one asks whether there are k pairwise disjoint subsets $X_1, \ldots, X_k$ of X such that $X = \bigcup_{i=1}^{k} X_i$ and for each $i \in \{1, \ldots, k\}$ it holds that $\sum_{x \in X_i} x \leq b$. Similar for each reduction is that, intuitively, each bin is represented as some distinct time interval.

▶ **Theorem 9** (⋆). ABS *without battery is strongly NP-hard even if* $e_i = 1$ *for all* $i \in N$ *and* $\text{Im}(F) = \{1\}$*. In this case, the problem is also W[1]-hard when parameterized by* $\rho + \delta$*.*

Note that Theorem 9's setup implies that only one job can be scheduled at any time step. Next we show that for unit lengths and distinct release dates, hardness remains.

▶ **Theorem 10** (⋆). ABS *without battery is strongly NP-hard, even if* $\ell = 1$, $\rho = n$ *and* $\mathrm{im}(F) = 2$.

Recall that when adding a flexibility of two to the restrictions of Theorem 10, we arrive in a polynomial-time solvable case (Theorem 5).

7 Parameterized Complexity

7.1 FPT Results

In Section 6, we showed that ABS is NP-hard for constant values of almost all natural parameters, except for the number n of jobs. In fact, for any constant number of jobs, the problem becomes polynomial-time solvable since we can guess for each job its starting time.

▶ **Observation 11** (⋆). *Let I be any instance of* ABS *with n jobs and maximum flexibility ϕ. In $O(\phi^n \cdot \mathrm{poly}(|I|))$ time, we can either correctly report that I is a* no-*instance or output a feasible schedule that maximizes the battery state when the last job finishes.*

Observation 11 leads to two natural next questions: (Q1) Is ABS in FPT when parameterized by n alone? (Q2) What (substantially) smaller parameters than n still lead to FPT, when combined with ϕ, and XP? Regarding (Q1), so far, we only know that this is the case for constant flexibility and if there is no battery.

▶ **Theorem 12** (⋆). ABS *without battery is FPT when parameterized by n.*

Proof sketch. We guess the (weak) order of jobs regarding their starting times and then iteratively along this order schedule each job as early as possible. This is correct since intuitively, the absence of a battery makes delays useless. Recall that ABS with battery is NP-hard (Theorem 8) even when such a weak order is given. ◀

Regarding (Q2), we identify two (incomparable) parameters for which this is the case. The first one is the size cco of the largest connected component of job graph G, and the second one is the size m-c-cco of a smallest modulator to c-cco in G.

▶ **Theorem 13** (⋆). ABS *is in FPT when parameterized by* cco *combined with the flexibility and in XP when parameterized by* cco *alone.*

Proof sketch. The idea is to solve each instance induced by a connected component from left to right via the algorithm from Observation 11, maximizing the battery level when the last job of the component finishes. Intuitively, this is correct since the only influence between components is the battery state; thus, we pass on a maximally charged battery. ◀

With Theorem 13 at hand, we obtain the following for a modulator to c-cco.

▶ **Theorem 14** (⋆). *Let I be an instance of* ABS, *G its job graph, and $W \subseteq V$ be a modulator to c-cco. Then, I is solvable in $O(\phi^{|W|+c} \cdot \mathrm{poly}(|I|))$ time.*

Theorem 14 yields that ABS is in FPT when parameterized by the size m-c-cco of a smallest modulator to c-cco combined with the flexibility ϕ and in XP when parameterized by m-c-cco alone. A well-known modulator yielding constant-size connected components is a vertex cover, i.e., a modulator to 1-cco. Since we can compute a minimum-size vertex cover in FPT-time (folklore), we have that ABS is FPT when parameterized by the flexibility combined with vertex cover size of the job graph.

7.2 W-hardness and Kernelization Lower Bounds Results

Achieving FPT for ϕ even in quite restricting settings or improving Theorem 13 to polynomial kernelization seems unlikely.

▶ **Theorem 15** ($\star$). ABS *without battery is strongly NP-hard and W[1]-hard when parameterized by* τ*, and hence also when parameterized by* ϕ*, even if* $\ell = 1$*,* $\rho = \delta = 1$*, and* $\mathrm{im}(F) = 1$*.*

▶ **Theorem 16** ($\star$). *Unless NP* $\subseteq$ *coNP/poly,* ABS *without battery admits no problem kernel of size polynomial in* $\phi + \mathrm{cco}$*, even if* $\ell = 1$*.*

8 ILP and Experiments

In this section, we present our experimental evaluation. Since our theoretical analysis revealed that ABS is intractable already in quite restricted settings, we employ integer linear programming (ILP) as solver. Exploiting the power of ILPs, we even solve a generalization of ABS, where an external power source $X\colon T \to \mathbb{Q}_{\geq 0}$ exists. We include the external power source by replacing (1) with $D_\pi(t) = F(t) + X(t) - E_\pi(t)$ for all $t \in T$.

▶ **Problem 2** (OPT-AUTARKY BY SCHEDULING (OPT-ABS)). **Given** a forecast F, a set $\mathcal{J}$ of jobs, and a battery B, the **task** is to find a schedule π and external energy X making π feasible such that $\sum_{t\in T} X(t)$ is minimized.

The minimum sum of external energy can be understood as distance-to-autarky measure. Note that ABS is OPT-ABS where no external power source is available ($X(t) = 0$ for all t). Thus, ABS reduces to OPT-ABS and hence the latter generalizes the former.

8.1 ILP Model

Consider the following ILP, called ABS-ILP, that minimizes the sum of external power:

$$\min \sum\nolimits_{t\in T} X_t, \quad X_t \in \mathbb{Q}_{\geq 0}\, \forall t \in T. \tag{4}$$

First, we model the job scheduling. For each $j \in N$, let $T_j \coloneqq \{r_j, \dots, d_j\}$ and we have:

$$x_{j,t} \in \{0,1\}\ \forall\, t \in T_j, \quad z_{j,t} \in \{0,1\}\ \forall\, t \in S_j \tag{5}$$

$$1 = \sum\nolimits_{t\in S_j} z_{j,t}, \qquad x_{j,t} = \sum\nolimits_{s\in S_j : s\leq t\leq s+\ell_j-1} z_{j,s} \quad \forall\, t \in T_j \tag{6}$$

Note that $z_{j,t}$ models whether job J_j starts at time step $t \in S_j$ and $x_{j,t}$ models whether job J_j is active at time step $t \in T_j$. Herein, (6) ensures that each job starts and the active times of a job form a consecutive sequence starting in accordance with $z_{j,t}$.

Next, we model the battery including the external power. For all $t \in T_{+1} \coloneqq \{2, \dots, \tau+1\}$, we have the following:

$$0 \leq B(t) \leq b_c, \qquad B(1) = b_0, \qquad B(\tau+1) \geq b_0 \tag{7}$$

$$E^+_{t-1} \leq F(t-1) + X_{t-1} + E^-_{t-1} - \sum\nolimits_{j\in N\colon (t-1)\in T_j} x_{j,t-1} \cdot e_j \tag{8}$$

$$0 \leq E^+_{t-1} \leq y_{t-1} \cdot b_\ell/\lambda_{\mathrm{in}}, \qquad y_{t-1} \in \{0,1\} \tag{9}$$

$$0 \leq E^-_{t-1} \leq (1 - y_{t-1}) \cdot b_{\mathrm{c}} \cdot \lambda_{\mathrm{out}}, \tag{10}$$

$$B(t) = B(t-1) + \lambda_{\mathrm{in}} \cdot E^+_{t-1} - \lambda^{-1}_{\mathrm{out}} \cdot E^-_{t-1} \tag{11}$$

Note that we additionally require in (7) that at each end of the scheduling horizon, the battery's level is at least b_0. This makes the initial battery level b_0 meaningful when computing on consecutive days (cf. [36, Sec. 2.3]). Moreover, our ILP formulation deviates from a formulation of (2) and (3) that directly implements min and max functions and enforces the fully available excess to go into the battery. We model the max-min constellation of (2) through lower bounds of zero and the binary variables y_{t-1}; the nested min in Equation (3) we model by upper bounding the battery sizes by b_c, the excess by $b_\ell/\lambda_{\text{in}}$, and the demand by $b_c \cdot \lambda_{\text{out}}$. Finally, note that in (8), the excess E^+_{t-1} is only upper bounded. This allows that the battery is possibly loaded with less than the actual available excess and only as much as required to reach an overall minimum of external energy. Indeed, we will obtain several ILP solutions making use of this (see Figure 12 in the appendix). While these formulation tricks are quite simple, we observe a significant effect: we tested a more direct formulation which failed to compute solutions with larger flexibilities in reasonable time.

Finally, note that we have a 1-to-1 correspondence between a solution for OPT-AbS, where we additionally require (7), and an optimal solution for AbS-ILP via $\pi(J_j) = t \iff z_{j,t} = 1$.

▶ Remark 17. Our ILP allows easily for many reasonable modifications. If one is interested in minimizing the cost of the energy, one can set the goal function to $\min \sum_{t\in T} w_t \cdot X_t$, where w_t is the energy's price in time step t. When two jobs i and j must be disjoint, we add $x_{i,t} + x_{j,t} \le 1$ for all $t \in T_i \cap T_j$. When a job i has to start or end before another job j, we can add $\sum_{t\in T_j} t \cdot z_{i,t} < \sum_{t\in T_j} t \cdot z_{j,t}$ or $\ell_i + \sum_{t\in T_j} t \cdot z_{i,t} \le \sum_{t\in T_j} t \cdot z_{j,t}$, respectively. When each job may have several, mutually disjoint horizons in which it is allowed to be scheduled, then consider their union for (6). Our data includes no information about any such further requirements addressed above. Hence, these constraints are omitted. ◁

8.2 Experiments

In this section, we describe the data, setup, and results of our experiments. The goal of our experiments is threefold. First, we quantify the practical impact of flexibility on reducing external energy under realistic solar radiation forecasts. Second, we compare small-scale balcony PV systems with residential rooftop systems to understand how system scale influences achievable autarky. Third, we investigate which structural properties of instances – such as job density and forecast irregularity – drive the computational difficulty of solving OPT-AbS in practice.

8.2.1 Data

Solar Radiation Data. To simulate our forecast, we use data from the Institute for Electrical Information Technology of the Technical University Clausthal, Germany, for radiation (in Watt per square meter [W m^{-2}]) measured for each minute [min] for every year from 2016 to 2022. The measurement is at a fixed point in Clausthal-Zellerfeld, Germany.

Appliance-level Power Consumption Data. We use data [1, 2] in which for two households A and B in Germany, several devices, such as kettle, coffee maker, fridge, or oven, were monitored over a certain time period at a temporal resolution of a few seconds. We only use the data for household A since household B's data is too sparse. We set a threshold of 6 W for a device to be active, and two active points in time that are at most 120 seconds apart belong to the same job. Each job then has a start and end time (understood as release date and deadline at flexibility 0), given at minute resolution. The difference between them defines the job's length. The job's energy requirement is defined as the maximum observed power (in W) over its execution interval (recall that this ensures the job to be executable even if energy demand fluctuates). There is no information about the flexibility of any job in the data.

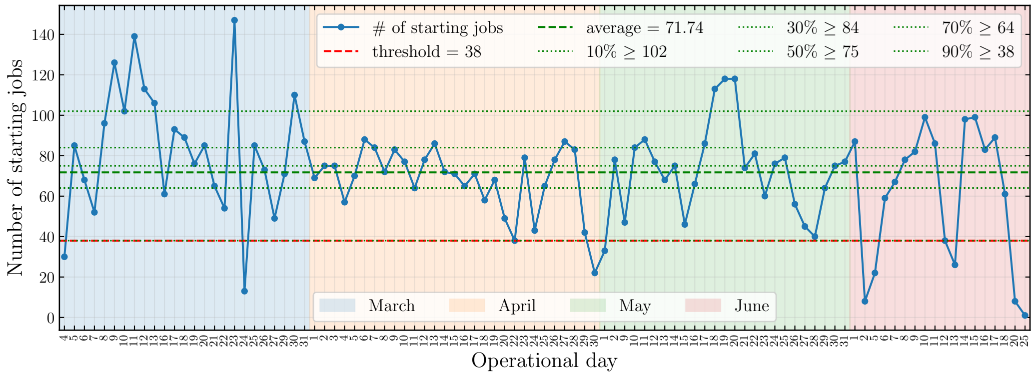


Figure 5 Overview of the number of jobs starting per operational day (calendar date) from March to June. The x-axis shows the day of the month (e.g., 2 in the green area is May 2). Percentiles of daily job counts are included; for instance, on 70% of the operational days, at least 64 jobs start.

8.2.2 Setup

PVs. For PVs, we assume a threshold of $10\,\mathrm{W\,m^{-2}}$, i.e., below which we set an entry to zero. We assume a pessimistic-realistic efficiency of 20% [15, 45]. We consider two scenarios, called `RA` and `BS`. `RA` (`BS`) corresponds to a rooftop (balcony) PV, where we assume $48\,\mathrm{m^2}$ ($3\,\mathrm{m^2}$) of PV area [30, 41] directed south. This allows us to compare plug-in balcony systems with residential rooftop systems [23]. So, given the radiation data and the sun's position per time step, the above described transformations are performed to compute the actual forecast.

Jobs. We computed the minimum external energy for an *operational day*, where an operational day starts at 4 am and ends at 3:59 am on the next day (at minute resolution, leading to 1440 time steps). We selected the 90% (88) of all available days between March and June (see Figure 5 for an overview) with the highest number of jobs to cut off small instances, leading to each instance having at least 38 jobs (the average of all days is about 72 jobs). For this set of jobs, the average daily mean job length is 39.17 min and the average daily median job length is 6.88 min. We added flexibility scenarios next to the 0% scenario given by the data. These are specified through a parameter $\hat{\phi} = 0.25 \cdot x$, $x \in \{0, 1, \dots, 12\}$, corresponding to a symmetric extension of up to 300% of the original time window length. Concretely, for $\hat{\phi} = 1$ and $\hat{\phi} = 3$, a job whose length equals the average daily mean job length obtains additional time windows of 19.58 min and 58.75 min, respectively, before and after its original release date. For a job whose length equals the average daily median job length, the corresponding additional time windows are 3.44 min and 10.32 min, respectively. Formally, if r_j, d_j, and ℓ_j are the release date, deadline, and length of J_j obtained from the data (recall that $\ell_j = d_j - r_j + 1$ here), then $r'_j = \max\{1, r_j - \hat{\phi} \cdot \ell_j/2\}$ and $d'_j = \min\{\tau, d_j + \hat{\phi} \cdot \ell_j/2\}$ (here, $\tau = 1440$, corresponding to the last time step for the operational day). We took the forecast of every year from 2016 to 2022, and combined them with the jobs recorded on the respective days.

Battery. The battery is measured in Watt-minutes [W min]. We assume a battery with capacity 60 kW min (1 kW h), commonly considered for balcony-sized PVs [4], with a loading speed of 3 hours from 0% to 100% and an initial battery load of 10%. Further, we assume that the effective in- and output efficiencies are equal ($\lambda := \lambda_{\text{in}} = \lambda_{\text{out}}$) and consider three values $\lambda \in \{0.9, 0.94, 0.98\}$ around the commonly assumed value 0.95 [46]. Since we focus on the effect of forecast variability and flexibility, we fix the battery size to isolate these factors.

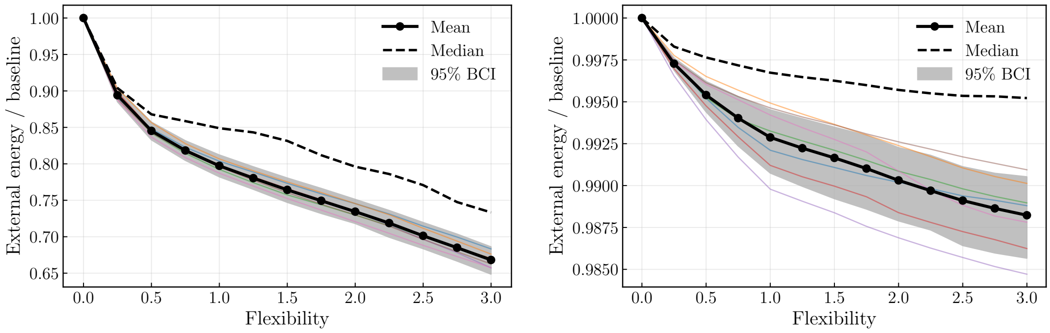


Figure 6 Flexibility versus aggregated mean and median external energy reduction over all years 2016–2022 and $\lambda = 0.94$ with 95% BCI for (left) RA and (right) BS. Each year's mean is shown in a different color (used for illustration only).

8.2.3 Results

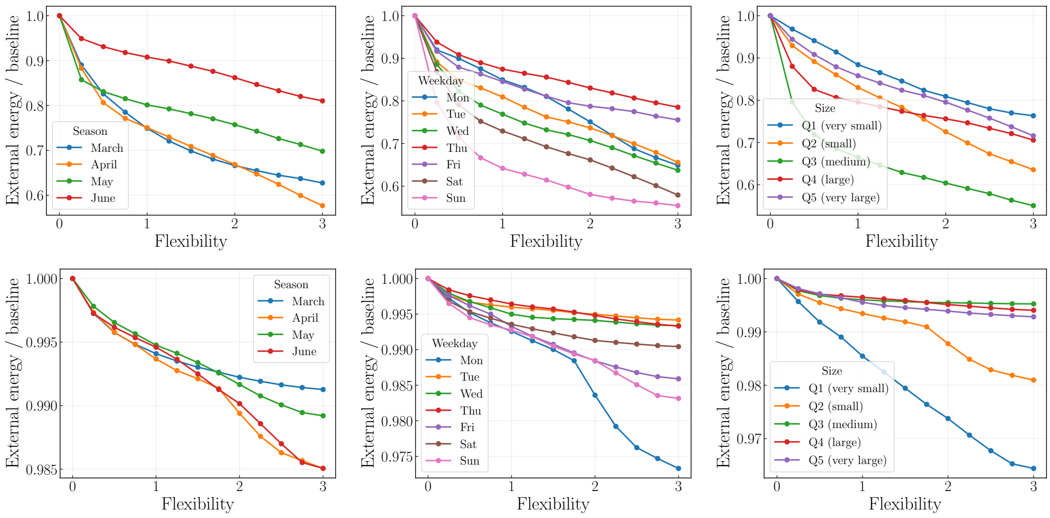
We used Python 3.10 and Gurobi 12.0 (Python interface) to compute solutions for ABS-ILP.[1] We compared RA and BS, each available day from 2016–2022, the thirteen flexibility values, and the three effectivities of our battery. In total, we solved 24024 instances for each of RA and BS. Figure 12 (appendix) shows optimal solutions for four example instances. When applicable, we discuss the results for $\lambda = 0.94$ in greater detail.

Energy Reduction. In Figure 6, we show the external energy reduction against the bottom line of flexibility zero when increasing the flexibility, aggregated over all years 2016–2022 for $\lambda = 0.94$. Our results show that for both RA and BS, a higher flexibility allows higher energy savings on average. With our maximum flexibility of 3, these are relatively small for BS with a reduction of 1.18% on average (median: 0.48%), yet quite large for RA with about 33.18% on average (median: 26.66%). Notably, the improvements are larger in the flexibility interval $(0, 1]$ and are then almost linear in $(1, 3]$. While the standard deviations are quite high (cf. Figure 13 in the appendix), the 95% percentile nonparametric bootstrap confidence interval [11] with 2000 replications (BCI for short) is quite narrow, in particular for RA ($\hat{\phi} = 3$: $[0.65, 0.69]$). Regarding the battery effectivity, we observe two orthogonal trends. With increasing effectivity, the reduction for $\hat{\phi} = 3$ slightly decreases for BS ($\lambda = 0.9$: 1.55%; $\lambda = 0.98$: 0.83%) but slightly increases for RA ($\lambda = 0.9$: 32.8%; $\lambda = 0.98$: 33.51%). Overall, flexibility yields substantial benefits for rooftop systems (with small-space batteries), but only marginal improvements for small balcony systems.

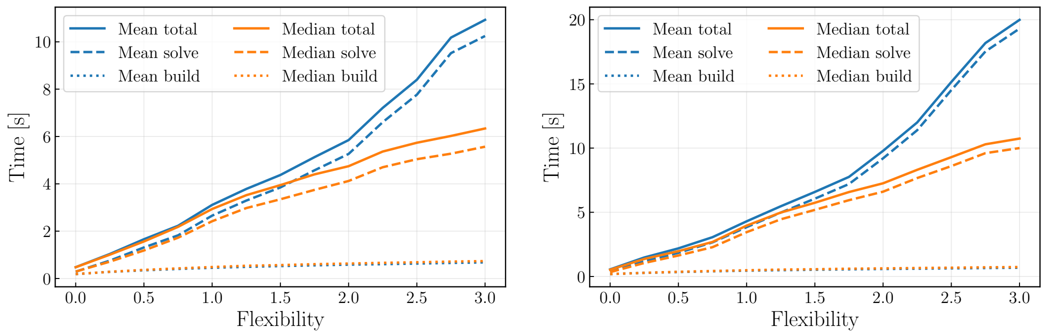
We additionally investigated which additional factors the energy reduction along flexibility may co-depend (see Figures 7 and 13 for year 2022). We checked for a seasonal correlation (by day of year), a weekday correlation, or correlation with the number of jobs. For none of these factors, we could identify a clear correlation.

Running Times. The runtime increases with the flexibility almost linearly (see Figure 8), up to 10.928 s (6.338 s) and 19.989 s (10.742 s) on average (median) at $\hat{\phi} = 3$ for RA and BS, respectively. Note that there are few yet severe outliers when the flexibility is at least 2 (see Figure 9): Here, in the range of 100–120 jobs, it may take up to 3 minutes on average to

[1] Run on Intel® Xeon® Silver 4310 CPU@2.10GHz (12 cores), 125GB RAM, Ubuntu 22.04.3 LTS (x86_64).



■ **Figure 7** External energy reduction per flexibility for all instances aggregated in the year 2022 with $\lambda = 0.94$ for (top) RA and (bottom) BS. Instances are aggregated by month (left), weekday (middle), and five percentiles of the number of jobs (right).



■ **Figure 8** Flexibility versus mean and median runtimes for RA (left) and BS (right). Here, 'build' and 'solve' refer to the runtimes of constructing and solving the ILP, respectively.

compute solutions for $\hat{\phi} = 3$. We also detected that the runtime increases with increasing number of scheduled jobs (see Figure 10); this effect was even more significant for BS (with an r-value of 0.4480, as compared to 0.3723 for RA). In contrast, we observed a weak positive correlation (r-value of 0.1871) for RA showing that the runtime increases with increasing *forecast irregularity*. We measure the forecast irregularity of an operational day by first normalizing the forecast by a clear-sky forecast, and then taking the sum of the squared differences. Interestingly, our hardness results rely on structurally simple forecasts. In contrast, the empirical runtimes increase with forecast irregularity, indicating that temporal variability is a practically relevant complexity driver. We point out that for BS, we did not detect such a correlation with statistical significance. Finally, we identified no correlation between the runtime and the battery effectivity for either RA or BS.

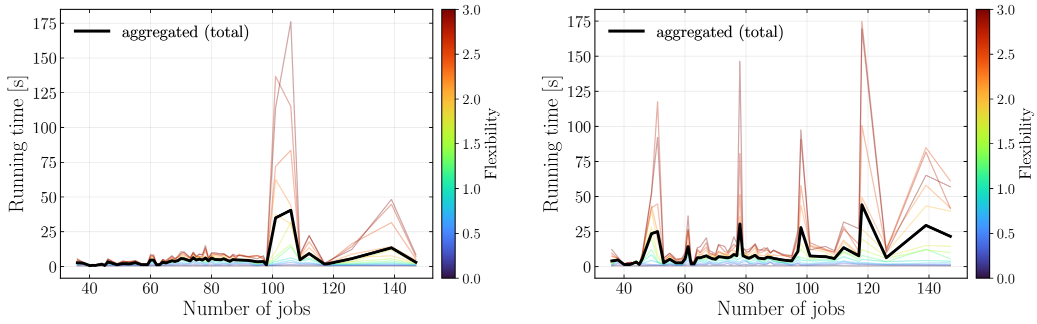


Figure 9 Running times versus number of jobs versus flexibility, with `RA` (left) and `BS` (right).

9 Conclusion

We introduced and analyzed a novel scheduling problem motivated by energy autarky, combining parameterized complexity theory with data-driven evaluation. Our results reveal a sharp contrast between worst-case computational hardness and substantial empirical gains achievable through flexibility. Overall, our experiments confirm that flexibility substantially reduces external energy in larger PV systems, already for relatively small flexibility values, while also revealing how forecast irregularity and job density may influence computational effort. For more detailed discussions on the theoretical and experimental parts, as well as a concluding big picture, see below.

Computational and Parameterized Complexity. Our most intriguing open question is whether the weak versus strong NP-hardness boundary is substantial. In this context, combining ϕ with the number of distinct job energies remains open. Further open questions regard whether ABS is FPT when parameterized by the number n of jobs alone. We know that guessing only the job order is insufficient (Theorem 8), and that a battery is required for hardness (Theorem 12). For our FPT results, e.g., for the parameter $\phi + n$, we wonder whether polynomial-sized problem kernels exist. Finally, future work may uncover additional polynomial-time solvable cases by adapting the algorithms from Theorems 5 and 6.

Experimental Evaluation. Assuming symmetric flexibility on both ends of a job's time window may be unrealistic in practice, but we are unaware of data connecting jobs with their typical flexibilities. Thus, we seek additional and structurally diverse datasets to validate and extend our analysis. Such data could also enhance our understanding of which additional factors significantly impact the energy reduction and runtimes. In our experiments, we considered 24 h time windows starting at 4 am. For most households, 4 am is a reasonable cutoff for defining operational days with respect to executed tasks. Future work may investigate the robustness of our results under shifted, shorter, or longer time windows. As to the battery, in our experiments we assumed that the in- and output effectivity is equal. While this is practically a reasonable assumption [46], it would be interesting to investigate how the results change with larger gaps between the two effectivities. Moreover, studying how charge-dependent efficiencies affect the outcomes, compared with the constant-efficiency setting considered here, is a well-motivated research direction. For future work, we plan to analyze the effect on the energy reductions of varying PV orientations [35] as well as battery loading speeds and sizes. Regarding battery sizes, we have preliminary results that indicate that doubling the battery size to 2 kW h leads to a marginally higher mean runtime and with

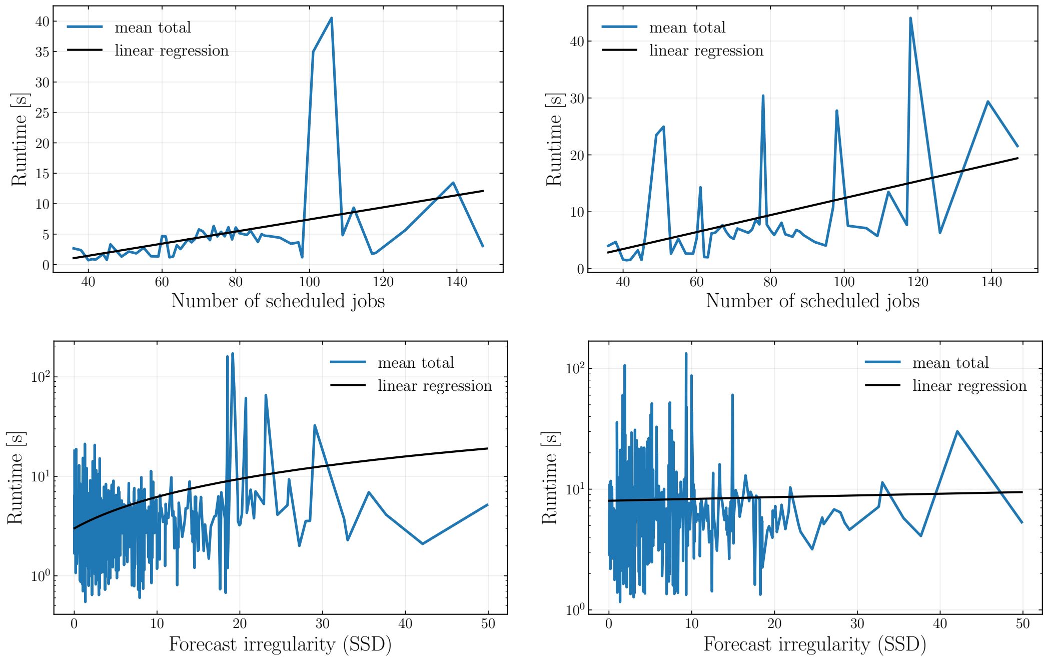


Figure 10 Running times versus (top) number of jobs and (bottom) forecast irregularity (y-log scale), with `RA` (left) and `BS` (right).

average reductions that are slightly higher for `RA` and lower for `BS`. Moreover, with a huge battery size of 6 kW h the average reduction diminishes since mostly the required external energy is little to none.

Big Picture. Our work demonstrates that flexibility-aware energy scheduling forms a rich algorithmic problem at the intersection of sustainability and complexity theory. We point out that the model, and hence our ILP, applies not only to households and PVs, but also to small neighborhoods, wind, or other power sources; it is not even restricted to electrical energy. While discharge-rate limits were assumed to be non-critical at the household level, they may have to be included for larger-scale or industrial instances (cf. Remark 2). We anticipate further applications of our model and corresponding algorithmic results.

References

1. Alaa Alhamoud, Felix Ruettiger, Andreas Reinhardt, Frank Englert, Daniel Burgstahler, Doreen Böhnstedt, Christian Gottron, and Ralf Steinmetz. SMARTENERGY.KOM: An intelligent system for energy saving in smart home. In *Proceedings of the 3rd IEEE LCN International Workshop on GlObal Trends in SMART Cities (IEEE goSMART 2014)*, pages 685–692. IEEE Xplore, 2014. `doi:10.1109/LCNW.2014.6927721`.
2. Alaa Alhamoud, Pei Xu, Andreas Reinhardt, Frank Englert, Philipp Scholl, Doreen Böhnstedt, and Ralf Steinmetz. Extracting human behavior patterns from appliance-level power consumption data. In *Proceedings of the 12th European Conference on Wireless Sensor Networks (EWSN 2015)*, volume 8965 of *Lecture Notes in Computer Science*, pages 52–67. Springer, 2015. `doi:10.1007/978-3-319-15582-1_4`.
3. Evripidis Bampis, Vincent Chau, Dimitrios Letsios, Giorgio Lucarelli, Ioannis Milis, and Georgios Zois. Energy efficient scheduling of mapreduce jobs. In *Proceedings of the 20th International Conference Parallel Processing (Euro-Par 2014)*, volume 8632 of *Lecture Notes in Computer Science*, pages 198–209. Springer, 2014. `doi:10.1007/978-3-319-09873-9_17`.

4 Georgios A. Barzegkar-Ntovom, Nikolas G. Chatzigeorgiou, Angelos I. Nousdilis, Styliani A. Vomva, Georgios C. Kryonidis, Eleftherios O. Kontis, George E. Georghiou, Georgios C. Christoforidis, and Grigoris K. Papagiannis. Assessing the viability of battery energy storage systems coupled with photovoltaics under a pure self-consumption scheme. *Renewable Energy*, 152:1302–1309, 2020. doi:10.1016/j.renene.2020.01.061.

5 Matthias Bentert, Robert Bredereck, Péter Györgyi, Andrzej Kaczmarczyk, and Rolf Niedermeier. A multivariate complexity analysis of the material consumption scheduling problem. *J. Sched.*, 26(4):369–382, 2023. doi:10.1007/S10951-022-00771-5.

6 Robert Bredereck, Till Fluschnik, and Klaus Heeger. buhtig-tf/Scheduling-Tasks-towards-Energy-Autarky. Software, swhId: swh:1:dir:ec658833659409634e2c49460594460b33fb79bc (visited on 2026-08-12). URL: https://github.com/buhtig-tf/Scheduling-Tasks-towards-Energy-Autarky. doi:10.4230/artifacts.27626.

7 Robert Bredereck, Till Fluschnik, and Klaus Heeger. Scheduling tasks towards energy autarky: Benefits and computational costs of flexibility, 2026. arXiv:2607.03260.

8 Dirk Briskorn, Morteza Davari, and Jannik Matuschke. Single-machine scheduling with an external resource. *Eur. J. Oper. Res.*, 293(2):457–468, 2021. doi:10.1016/J.EJOR.2020.12.045.

9 Peter Brucker, Andreas Drexl, Rolf H. Möhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur. J. Oper. Res.*, 112(1):3–41, 1999. doi:10.1016/S0377-2217(98)00204-5.

10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

11 Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994. doi:10.1201/9780429246593.

12 Robert Ganian, Thekla Hamm, and Guillaume Mescoff. The complexity landscape of resource-constrained scheduling. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1741–1747. ijcai.org, 2020. doi:10.24963/IJCAI.2020/241.

13 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

14 Sathish Gopalakrishnan, Nima Nasiri, and Jared Paul. Job scheduling with battery recharging constraints: Applications to uav flight planning. In *2022 IEEE Real-Time Systems Symposium (RTSS)*, pages 422–434, 2022. doi:10.1109/RTSS55097.2022.00043.

15 Martin A. Green, Ewan D. Dunlop, Jochen Hohl-Ebinger, Masahiro Yoshita, Nikos Kopidakis, and Xiaojing Hao. Solar cell efficiency tables (version 56). *Progress in Photovoltaics: Research and Applications*, 28(7):629–638, 2020. doi:10.1002/pip.3303.

16 Anya Heider, Ricardo Reibsch, Philipp Blechinger, Avia Linke, and Gabriela Hug. Flexibility options and their representation in open energy modelling tools. *Energy Strategy Reviews*, 38:100737, 2021. doi:10.1016/j.esr.2021.100737.

17 Verena Heinisch, Mikael Odenberger, Lisa Göransson, and Filip Johnsson. Prosumers in the electricity system—household vs. system optimization of the operation of residential photovoltaic battery systems. *Frontiers in Energy Research*, Volume 6 - 2018, 2019. doi:10.3389/fenrg.2018.00145.

18 Danny Hermelin, Dvir Shabtay, Chen Zelig, and Michael L. Pinedo. A general scheme for solving a large set of scheduling problems with rejection in FPT time. *J. Sched.*, 25(2):229–255, 2022. doi:10.1007/S10951-022-00731-Z.

19 Lyu-Guang Hua, S. Haseeb Ali Shah, Baheej Alghamdi, Ghulam Hafeez, Safeer Ullah, Sadia Murawwat, Sajjad Ali, and Muhammad Iftikhar Khan. Smart home load scheduling system with solar photovoltaic generation and demand response in the smart grid. *Frontiers in Energy Research*, Volume 12 - 2024, 2024. doi:10.3389/fenrg.2024.1322047.

20 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013. doi:10.1016/j.jcss.2012.04.004.

21 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.

22 Dusan Knop and Martin Koutecký. Scheduling meets n-fold integer programming. *J. Sched.*, 21(5):493–503, 2018. doi:10.1007/S10951-017-0550-0.

23 Tobias Kraschewski, Maximilian Heumann, and Michael H. Breitner. Adoption and spatial distribution disparities of residential plug-in and rooftop photovoltaic systems in germany. *Renewable and Sustainable Energy Reviews*, 209:115092, 2025. doi:10.1016/j.rser.2024.115092.

24 Arvind Kumar and Bashir Alam. Task scheduling in real time systems with energy harvesting and energy minimization. *Journal of Computer Science*, 14(8):1126–1133, August 2018. doi:10.3844/jcssp.2018.1126.1133.

25 Mithilesh Kumar and Daniel Lokshtanov. A $2lk$ Kernel for l-Component Order Connectivity. In *Proceedings of the 11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.IPEC.2016.20.

26 Lissy Langer and Thomas Volling. An optimal home energy management system for modulating heat pumps and photovoltaic systems. *Applied Energy*, 278:115661, 2020. doi:10.1016/j.apenergy.2020.115661.

27 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

28 Fernando Lezama, Ricardo Faia, Pedro Faria, and Zita Vale. Demand response of residential houses equipped with PV-battery systems: An application study using evolutionary algorithms. *Energies*, 13(10), 2020. doi:10.3390/en13102466.

29 Jinpeng Liu, Hao Yang, Delin Wei, and Xiaohua Song. Time distribution simulation of household power load based on travel chains and monte carlo–a study of beijing in summer. *Sustainability*, 13(12), 2021. doi:10.3390/su13126651.

30 Maximilian Lödl, Georg Kerber, Ralf Witzmann, Christian Hoffmann, and Michael Metzger. Abschätzung des Photovoltaik-Potentials auf Dachflächen in Deutschland. Technical report, Technische Universität München, 2010. URL: https://mediatum.ub.tum.de/doc/969497/969497.pdf.

31 Peter D. Lund, Juuso Lindgren, Jani Mikkola, and Jyri Salpakari. Review of energy system flexibility measures to enable high levels of variable renewable electricity. *Renewable and Sustainable Energy Reviews*, 45:785–807, 2015. doi:10.1016/j.rser.2015.01.057.

32 Lennart Merkert, Iiro Harjunkoski, Alf Isaksson, Simo Säynevirta, Antti Saarela, and Guido Sand. Scheduling and energy – industrial challenges and opportunities. *Computers & Chemical Engineering*, 72:183–198, 2015. A Tribute to Ignacio E. Grossmann. doi:10.1016/j.compchemeng.2014.05.024.

33 Matthias Mnich and René van Bevern. Parameterized complexity of machine scheduling: 15 open problems. *Comput. Oper. Res.*, 100:254–261, 2018. doi:10.1016/J.COR.2018.07.020.

34 Matthias Mnich and Andreas Wiese. Scheduling and fixed-parameter tractability. *Mathematical Programming*, 154(1-2):533–562, 2015. doi:10.1007/S10107-014-0830-9.

35 Riyad Mubarak, Eduardo Weide Luiz, and Gunther Seckmeyer. Why PV modules should preferably no longer be oriented to the south in the near future. *Energies*, 12(23), 2019. doi:10.3390/en12234528.

36 Nina Munzke, Bernhard Schwarz, Felix Büchle, and Marc Hiller. Evaluation of the efficiency and resulting electrical and economic losses of photovoltaic home storage systems. *Journal of Energy Storage*, 33:101724, 2021. doi:10.1016/j.est.2020.101724.

37 Jesper Nederlof and Céline M. F. Swennenhuis. On the fine-grained parameterized complexity of partial scheduling to minimize the makespan. *Algorithmica*, 84(8):2309–2334, 2022. doi:10.1007/S00453-022-00970-8.

38 Nico Orth, Nina Munzke, Johannes Weniger, Christian Messner, Robert Schreier, Michael Mast, Lucas Meissner, and Volker Quaschning. Efficiency characterization of 26 residential photovoltaic battery storage systems. *Journal of Energy Storage*, 65:107299, 2023. `doi:10.1016/j.est.2023.107299`.

39 PS Pravin, Zhiyao Luo, Lanyu Li, and Xiaonan Wang. Learning-based scheduling of industrial hybrid renewable energy systems. *Computers & Chemical Engineering*, 159:107665, 2022. `doi:10.1016/j.compchemeng.2022.107665`.

40 Jonas Quernheim and Eberhard Waffenschmidt. A comparative evaluation of community-used district and individual battery storage systems for photovoltaic energy systems. *Energies*, 17(17), 2024. `doi:10.3390/en17174306`.

41 Marc Ringel, Jessica Kluge, Lisa Hillemacher, Johanna Gutscher, and Dominik Möst. Quantifying benefits of renewable investments for german residential prosumers in a changing electricity market. *Nature Communications*, 15:3212, 2024. `doi:10.1038/s41467-024-43330-1`.

42 Mariam Sangaré, Eric Bourreau, Bernard Fortz, Amaury Pachurka, and Michael Poss. Loads scheduling for demand response in energy communities. *Computers & Operations Research*, 160:106358, 2023. `doi:10.1016/j.cor.2023.106358`.

43 Baruch Schieber, Bhargav Samineni, and Soroush Vahidi. Interweaving real-time jobs with energy harvesting to maximize throughput. In *Proceedings of the 17th International Conference and Workshops on Algorithms and Computation (WALCOM 2023)*, volume 13973 of *Lecture Notes in Computer Science*, pages 305–316. Springer, 2023. `doi:10.1007/978-3-031-27051-2_26`.

44 Muhammad Salman Shahid, Seun Osonuga, Nana Kofi Twum-Duah, Sacha Hodencq, Benoit Delinchant, and Frédéric Wurtz. An assessment of energy flexibility solutions from the perspective of low-tech. *Energies*, 16(7), 2023. `doi:10.3390/en16073298`.

45 Feroz Shaik, Syam Sundar Lingala, and Punnaiah Veeraboina. Effect of various parameters on the performance of solar pv power plant: a review and the experimental study. *Sustainable Energy Research*, 10(1):6, 2023. `doi:10.1186/s40807-023-00076-x`.

46 Peter Stenzel, Timo Kannengießer, Leander Kotzur, Peter Markewitz, Martin Robinius, and Detlef Stolten. Emergency power supply from photovoltaic battery systems in private households in case of a blackout – a scenario analysis. *Energy Procedia*, 155:165–178, 2018. 12th International Renewable Energy Storage Conference, IRES 2018, 13-15 March 2018, Düsseldorf, Germany. `doi:10.1016/j.egypro.2018.11.058`.

47 Fhazhil Wamalwa and Ariane Ishimwe. Optimal energy management in a grid-tied solar PV-battery microgrid for a public building under demand response. *Energy Reports*, 12:3718–3731, 2024. `doi:10.1016/j.egyr.2024.09.054`.

48 Renjie Yu and Daniel Oron. Single-machine scheduling with fixed energy recharging times to minimize the number of late jobs and the number of just-in-time jobs: A parameterized complexity analysis. *European Journal of Operational Research*, 324(1):40–48, 2025. `doi:10.1016/j.ejor.2025.01.007`.

A Appendix

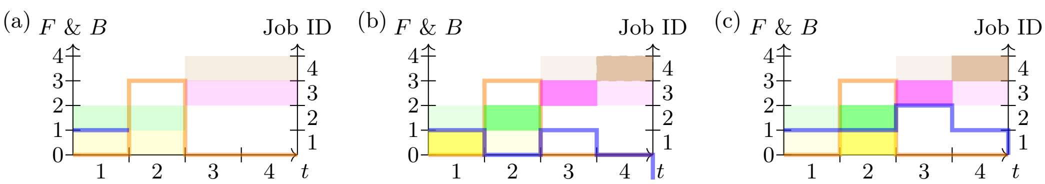


Figure 11 Counter example with $b_\ell = 1 < b_c$, where the orange line describes the forecast, the blue line the battery, and filled rectangles either a job's horizon (more opaque) or a scheduled job (less opaque). Each job has horizon two and length one, and requires one energy per time. (a) The input instance. (b) An infeasible schedule output by the greedy algorithm. (c) A feasible schedule.

Figure 12 Illustration of ILP solution for RA with $\lambda = 0.94$ for operational day March 13 in year 2020 (upper half) and in year 2022 (lower half). In each half, the upper and lower plot corresponds to flexibility 0 and 3, respectively. The total required external energies for flexibility 0 and 3 are 84210 respectively 75497 for 2020, and 9876 respectively 5275 for 2022. Jobs are depicted as colored stripes, each at its unique y-value (corresponding to the job's index). We present both battery states: those computed by the ILP, and the "real" ones in the sense of when the jobs are scheduled according to the ILP solution.

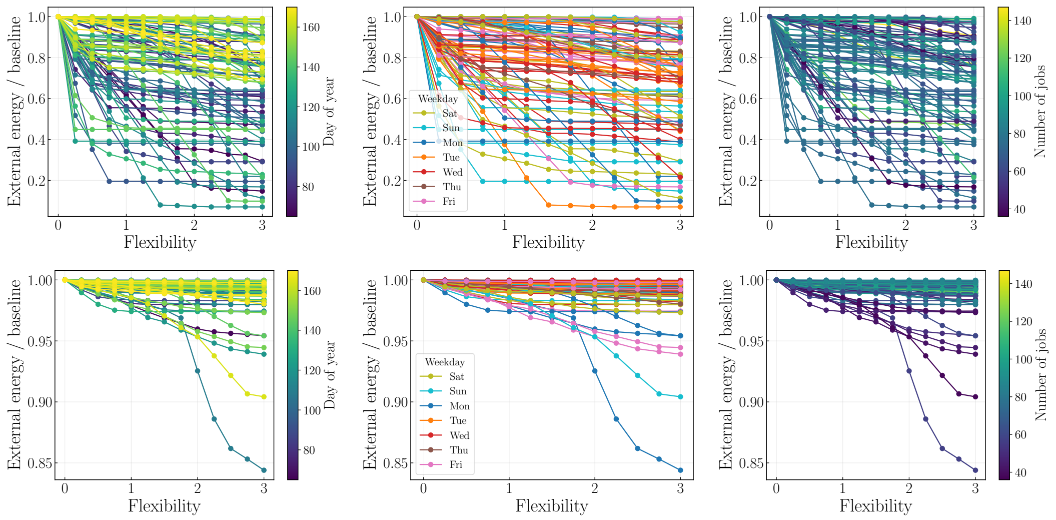


Figure 13 External energy reduction per flexibility for all instances in the year 2022 with $\lambda = 0.94$ for (top) `RA` and (bottom) `BS`. Instances are colored by day of the year (left), weekday (middle), and number of jobs (right).