



HAL
open science

A robust fix-and-optimize matheuristic for timetabling problems with uncertain renewable energy production

Akylas Stratigakos

► **To cite this version:**

Akylas Stratigakos. A robust fix-and-optimize matheuristic for timetabling problems with uncertain renewable energy production. IEEE Symposium Series on Computational Intelligence 2021, IEEE, Dec 2021, Orlando, United States. hal-03449920

HAL Id: hal-03449920

<https://hal.science/hal-03449920v1>

Submitted on 25 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A robust fix-and-optimize matheuristic for timetabling problems with uncertain renewable energy production

Akylas Stratigakos

Center PERSEE

MINES ParisTech, PSL University

Sophia-Antipolis, France

akylas.stratigakos@mines-paristech.fr

Abstract—This work presents a fix-and-optimize matheuristic to solve timetabling problems under uncertainty. Specifically, a combined university timetabling and electricity storage scheduling problem is considered, subject to uncertainty stemming from renewable energy production and electricity demand. The problem is formulated as a large Mixed Integer Program (MIP) and the proposed solution combines Large Neighborhood Search coupled with scenario-based robust optimization for handling uncertainty in the objective function. First, an adequate feasible schedule is derived considering only hard problem constraints, in this case scheduling of recurring lecture activities. Next, the solution is improved with a fix-and-optimize heuristic search. In each iteration, the MIP solver explores a large neighborhood by fixing a subset of variables and optimizing over the remaining free variables. The process is repeated several times until a stopping criterion is met. To address uncertainty in the objective, probabilistic scenarios are derived from interval forecasts and the worst-case energy cost is minimized. The results derived from the participation in a technical challenge show that the proposed approach provides competitive solutions relatively fast, even for large problem instances, while also hedging against large forecast errors.

Index Terms—fix-and-optimize, local neighborhood search, renewable energy forecasting, robust optimization, university timetabling.

I. INTRODUCTION

The integration of renewables in the generation mix is key to the ongoing decarbonization process. Their stochastic nature, however, poses an important challenge to their integration within larger operational processes. Dealing with this stochasticity typically involves a process where several prediction and optimization components are deployed in sequence. This work presents a methodology to tackle large size scheduling and timetabling problems with uncertain renewable production and electricity demand, based on the participation in the technical challenge organized by the IEEE Computational Intelligence Society (IEEE-CIS) [1].

In this challenge, several instances of a university timetabling problem are considered, which can be modelled as a Mixed Integer Program (MIP). It is known that such problems are NP-hard, therefore, an optimal solution within reasonable time is not attainable. The proposed solution is

guided by several challenges that revealed themselves during the early stages of the competition. First, the limited computational resources did not allow to solve the (multiple) problem instances to optimality. Second, the computational cost also hindered the ability to explore different strategies during the validation phase, e.g. how to tackle the parameter uncertainty. Lastly, as the time to be allocated in this challenge was also limited, the decision was made to focus on the optimization component at the expense of the prediction component. Considering the above, the proposed solution adheres to the following: i) can be implemented in a standard machine, ii) provides competitive results relatively fast, iii) and provides hedging against large forecast errors.

To this end, the described solution is based on a fix-and-optimize heuristic search to iteratively improve the solution of the MIP solver (*matheuristic*). Uncertainty in problem parameters (i.e., renewable production and electricity demand) is handled with scenarios based on marginal predictive intervals. A robust formulation is provided to minimize the worst-case expected cost within the provided set of scenarios, thus offering protection against miscalibrated forecasting models.

II. BACKGROUND

The individual components are briefly described in this section. In short, the problem concerns university timetabling and storage scheduling with the goal to minimize the total energy cost of a university complex, subject to uncertain electricity demand and solar production. For a detailed description of the problem, see [1].

1) *Problem formulation*: The MIP formulation of the university timetabling problem is omitted due to space limitations (see [2], [3] for relevant formulations). Recurring activities (lectures) are considered "hard" constraints, as they are required for a feasible schedule solution. Once-off activities (corresponding to lab experiments) are considered "soft" constraints, as they are not required for a feasible solution. Recurring activities are repeated in a weekly basis, thus a single set of variables per week suffices. For once-off activities, the defined variables span the whole scheduling period.

In short, a binary variable is assigned for each activity and each required time period. Further, an integer variable is defined for each activity, time period, building, and room type, to determine the number of rooms assigned. Note that this formulation is somewhat excessive. However, modern MIP solvers remove redundant constraints and variables during a presolve phase, thus leading to significantly smaller problems during the branch-and-bound search. For reference, for a large instance the initial problem comprises approximately $9 \cdot 10^3$ continuous and $11 \cdot 10^6$ integer ($4.5 \cdot 10^5$ binary) variables. After the presolve phase, these are decreased to 640 continuous and $2 \cdot 10^5$ integer ($7.8 \cdot 10^3$ binary), respectively. To the author’s opinion, the requirement of activities to be scheduled in consecutive time slots poses the greater modeling challenge, which is handled with the introduction of auxiliary variables and applying a big-M reformulation of constraints.

2) *Forecasting*: The uncertain electricity demand and solar production must be predicted for the whole scheduling period; alternatively one could consider the aggregated net load. A separate forecasting model is trained for each series provided. Regarding solar panels, several base models, namely Random Forest, Gradient Boosting Machines, Ridge Regression, and Local Learning Regression, are developed and aggregated via averaging for the final prediction.

Regarding buildings’ demand, the prediction task is more challenging due to missing data and restrictions related to the COVID-19 pandemic. Overall, buildings 1 and 4-6 show relatively low demand, which in turn reduces their impact on final energy cost. The decision was made to discard all missing data, as most of them relate to buildings 4-6. All demand forecasts were derived with a Random Forest model. Feature data included the weather data provided by the organisers, alongside categorical variables to model the calendar effect, impact of COVID-19 restrictions, exams period, etc. For buildings 0 and 3 a pre-processing step was included to remove possible outliers. Lastly, a Quantile Regression Forests model is employed to generate probabilistic forecasts for building 3, which are subsequently used to derive a high and low impact scenario.

3) *Objective*: The solutions are evaluated based on aggregate energy cost, which includes a peak load tariff. A scenario-based robust objective is derived to hedge against large forecast errors. Specifically, two scenarios for aggregated net load are derived based on the marginal predictive density of demand in building 3. The objective is reformulated in a standard epigraph form, and the worst-case cost over all scenarios is minimized. Ideally the scenarios should incorporate correlation between the different sources of uncertainty. Due to limited time and after a trial-and-error period, it was decided to derive scenarios solely based on the predictive density of building 3, which represented the largest load. Fig. 1 presents an indicative set of forecasts for a single day. Effectively, this approach creates a small box uncertainty set that aims to protect against larger forecast errors.

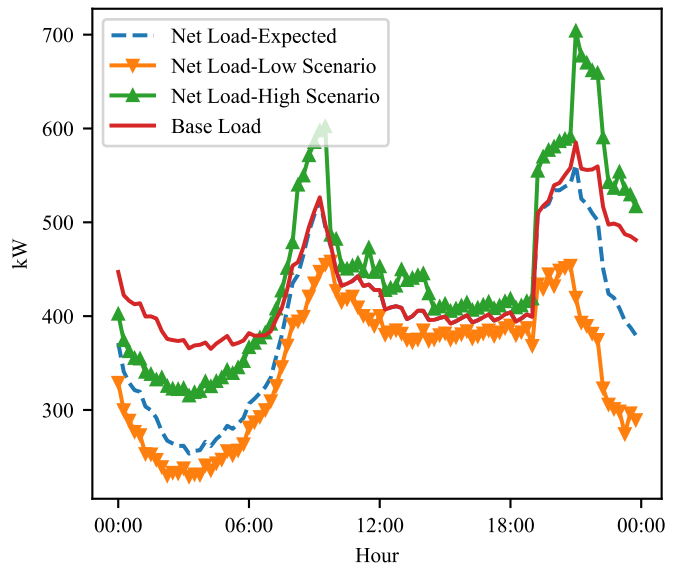


Fig. 1. One day of aggregated net load forecast and high/low scenarios for Phase 2. Base Load refers to the expected buildings demand without considering solar production.

III. METHODOLOGY

This section presents the complete methodology that builds on the individual components discussed above. A fix-and-optimize matheuristic is proposed to solve the MIP problem [4], which belongs to the class of Large Neighborhood Search (LNS) heuristics [5]. The approach is *method based*, meaning that the underlying model is the same, but the local neighborhood to search an improved solution changes at each iteration. Note that modern optimization solvers (e.g. GUROBI) require the model to be built once and each iteration is warm-started from the previous solution by adding or removing constraints, which significantly speeds-up computations.

The solution process is described in the flowchart shown in Fig. 2. The derived predictions described in the previous Section are considered as input in this process. First, the problem is solved once considering only “hard” constraints, providing a minimum feasible solution, which translates into optimizing only over recurring activities. The sample solutions provided by the organisers are used to warm-start the solver. Next, the iterative search begins by sampling a subset of activities, considering both recurring ($r^\#$) and once-off ($a^\#$). Variables related to non-selected activities are fixed to their previous solutions and the subsequent problem is solved to optimize over the remaining free variables. The process is repeated several times until the maximum number of iterations is reached (max-iter) or until the solution stops improving after a number of iterations (patience) subject to a pre-defined tolerance (tol). Variables related to storage remain free during all iterations, as the computational burden increased only marginally. The search algorithm is detailed in Algorithm 1. Note, this search is greedy and random. Due to precedence constraints, subsequent problems with fixed

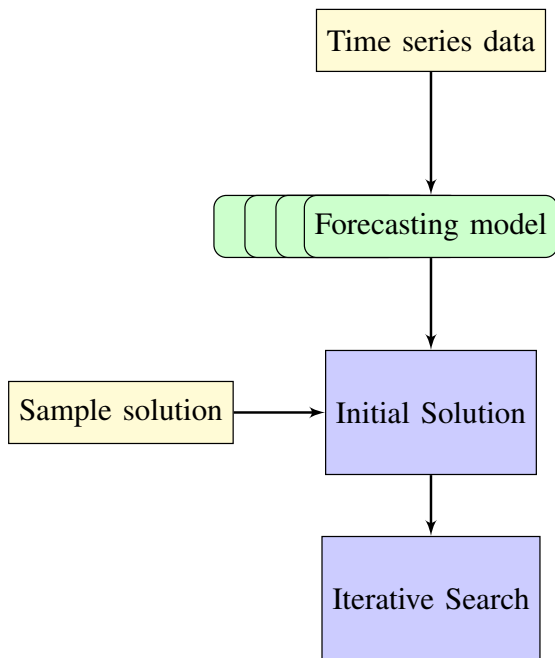


Fig. 2. Flowchart of the solution. Yellow boxes denote data input, green boxes denote forecasting models, and blue boxes denote optimization components.

constraints are generally easier to solve close to optimality, due to reduced feasible space. In all cases, soft time constraints based on the current optimality gap are imposed by tracking the progress of the branch-and-bound search within the MIP solver through callback functions.

Algorithm 1 Fix-and-Optimize

Input: Built GUROBI model, initial objective v^{init} , search hyperparameters

Output: Improved solution

- 1: Initialize counters $count, iter = 0$
 - 2: Set best objective $v^* \leftarrow v^{init}$
 - 3: **while** $iter \leq \text{max-iter}$ **and** $count \leq \text{patience}$ **do**
 - 4: Sample $r^\#$ recurring and $a^\#$ once-off activities
 - 5: Variables pertaining to the selected activities are free, the rest are fixed to previous solution
 - 6: Solve MIP, determine objective v^{new}
 - 7: **if** $\frac{v^* - v^{new}}{v^*} \geq \text{tol}$ **then**
Update solution, $v^* \leftarrow v^{new}$, $count = 0$
 - 8: **else**
 $count + = 1$
 - 9: **end if**
 $iter + = 1$
 - 10: **end while**
 - 11: **return:** Improved solution with objective v^*
-

IV. EXPERIMENTS

This section provides some experimental results and details on tuning the hyperparameters. During Phase 1 (validation),

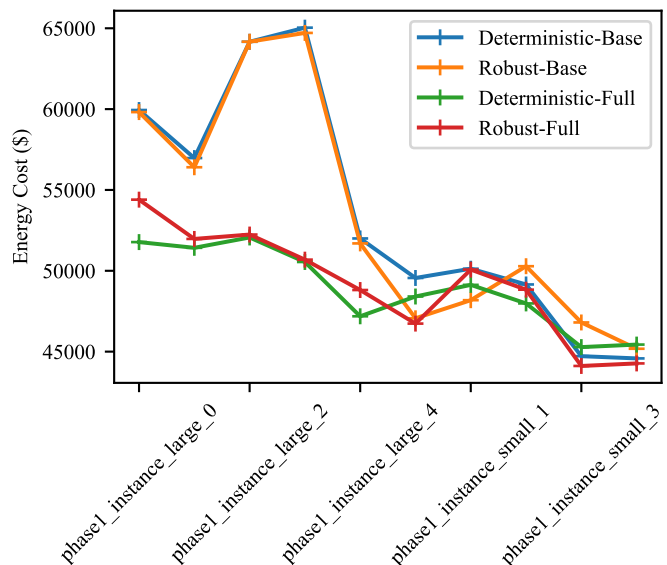


Fig. 3. Results on Phase 1. "Base" denotes the initial solution, "Full" denotes the final solution after the fix-and-optimize search is implemented.

different strategies were explored, mainly concerning how to deal with uncertainty in the objective function. Fig. 3 provides a comparison between a deterministic formulation (i.e., using point forecasts) and the scenario-based robust approach for the Phase 1 data. Note that the validation data includes a large demand outlier, which actually obscures the impact of predictive accuracy in the final cost. Regardless the objective formulation, Fig. 3 highlights that the fix-and-optimize search significantly reduces costs, with results being more pronounced for larger problem instances.

Regarding the Phase 2 (testing phase), all experiments are conducted using a standard machine with an Intel i7@2.30GHz and 32GB of RAM. The problems are modeled and solved with GUROBI using the Python-API. For reference, the required time to build a model is approximately 200-250 sec for small instances and 900-950 sec for large ones. To create the aggregated net load scenarios, the 10% and 90% quantile of the predictive marginal density of building 3 are selected. For the initial feasible solution a hard time limit of 2 hours is set, although a smaller time limit would also suffice. Subsequent iterative solutions are solved with a hard time limit of 20 min. A soft time limit of 5 min, if the optimality gap is below 0.005%, is also set. The total number of iterations is set at $\text{max-iter} = 150$, the patience is set at $\text{patience} = 15$, and the improvement tolerance is set at $\text{tol} = 0.001\%$. Lastly, at each iteration the number of sampled activities are $r^\# = 10$ for recurring and $a^\# = 5$ for once-off, same for both small and large problem instances. The code is provided in [6]. For reference, Fig. 4 shows the evolution of the objective value for a large instance in the testing phase.

V. DISCUSSION

This section provides a brief discussion on how to improve the performance of the proposed solution, insights gained from

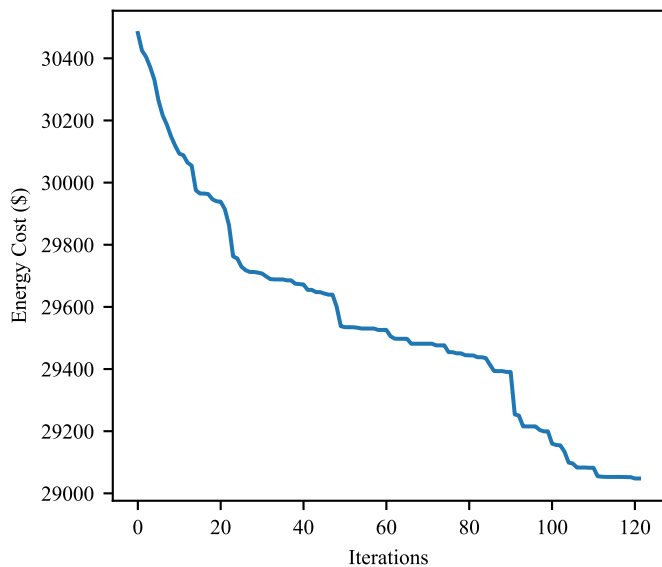


Fig. 4. Progression of the local search.

the participation in the technical challenge, and promising research directions. Regarding the performance of the final submission, several improvements could be made. First, considering better informed scenarios could improve the solution. The decision to base scenarios solely on the predictive density of building 3 is somewhat arbitrary; considering scenarios that account for the correlation between different sources of uncertainty could improve final performance. Second, improving the formulation of the timetabling problem could reduce computational costs. Third, note that the performance of the iterative search procedure is bounded by the solution to the full MIP problem. Smaller problem instances could probably be solved to optimality within reasonable time, given adequate time and computational resources. Moreover, the search is greedy and random; considering precedence constraints and sampling blocks of activities instead of individual activities could also improve the performance. Overall, it is the author's opinion that the iterative search proposed offers higher utility for larger problem instances. Lastly, the optimization window changed during Phase 2 to match the forecasting window. As a result, additional effort was required to modify the optimization code to the new setting, which somewhat hindered the ability to properly evaluate results in the validation data. Avoiding modeling errors also proved important, as it hindered the ability to exploit and evaluate different solutions during within the timeframe of this challenge.

The impact of forecast uncertainty on optimization efficacy in this challenge, to the author's opinion, remains somewhat obscure. The validation data (Phase 1) included an extremely large demand outlier, which affected the peak demand and the respective tariff. In turn, this mitigated the impact of the objective formulation (deterministic versus robust). Further, examining the results on validation data (Phase 1) showed that, at least for the large instances, the peak demand tariff

comprised the biggest part of total energy cost. However, the magnitude of the load to be scheduled during Phase 2 (relating to the respective activities), was significantly smaller. If the problem instances are viewed as data points from a problem distribution to be learned [7], this could be considered as a change in the underlying distribution. Overall, the peak tariff became less important during Phase 2, which to the author's opinion, somewhat obscures the impact of forecast accuracy in total costs.

Finally, increased predictive accuracy does not always translate into improved optimization performance. An interesting avenue to explore would be to train the model to directly minimize downstream optimization costs. For example, [8] proposes training deep learning models in conjunction with combinatorial optimization problems to improve the utility of the predictions (i.e., *forecast value*).

VI. CONCLUSIONS

This report detailed the participation in a technical challenge organized by IEEE-CIS. Specifically, a robust fix-and-optimize metaheuristic to tackle large scale timetabling and scheduling problems under uncertainty was described. Uncertainty in renewable production and electricity demand is modelled with scenarios and the worst-case energy cost is minimized. The results show that the proposed solution provides fast solutions that achieve competitive performance, while also hedging against large forecast errors.

ACKNOWLEDGMENT

The author would like to thank IEEE-CIS for organizing this very interesting technical challenge and, in particular, Dr. Bergmeir for his prompted responses to all the queries.

REFERENCES

- [1] C. Bergmeir, "Ieee-cis technical challenge on predict+optimize for renewable energy scheduling," 2021. [Online]. Available: <https://dx.doi.org/10.21227/1x9c-0161>
- [2] S. Daskalaki, T. Birbas, and E. Housos, "An integer programming formulation for a case study in university timetabling," *European journal of operational research*, vol. 153, no. 1, pp. 117–135, 2004.
- [3] G. Lach and M. E. Lübbecke, "Curriculum based course timetabling: new solutions to udine benchmark instances," *Annals of Operations Research*, vol. 194, no. 1, pp. 255–272, 2012.
- [4] M. Lindahl, M. Sørensen, and T. R. Stidsen, "A fix-and-optimize metaheuristic for university timetabling," *Journal of Heuristics*, vol. 24, no. 4, pp. 645–665, 2018.
- [5] D. Pisinger and S. Ropke, "Large neighborhood search," in *Handbook of metaheuristics*. Springer, 2010, pp. 399–419.
- [6] <https://git.persee.mines-paristech.fr/akylas.stratigakos/ieee-cis-ppo>.
- [7] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021.
- [8] B. Wilder, B. Dilkina, and M. Tambe, "Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1658–1665.