# TimeStruct.jl – flexible multi-horizon time modelling in optimization models

**Truls Flatberg** [1*¶], **Julian Straus** [2*], **and Lars Hellemo** [1*]

**1** SINTEF Industry, Postboks 4760 Torgarden, 7465 Trondheim **2** SINTEF Energy Research, Postboks 4761 Torgarden, 7465 Trondheim ¶ Corresponding author * These authors contributed equally.

**DOI:** 10.21105/joss.07578

**Software**

- Review
- Repository
- Archive

**Editor:** Vissarion Fisikopoulos

**Reviewers:**

- @datejada
- @lucaferranti

**Submitted:** 01 November 2024
**Published:** 03 March 2025



## Summary

`TimeStruct.jl` is a Julia (Bezanson et al., 2017) package that provides an interface for abstracting time structures, primarily intended for use with the mathematical programming DSL `JuMP.jl` (Lubin et al., 2023).

`TimeStruct.jl` simplifies the writing of key equations in optimization problems through separation of the indexing sets and the equations. Consequently, equations unaffected by the the chosen time structure, e.g., simple deterministic operational or stochastic programming models, need not be adjusted when changing the time structures. Hence, it simplifies both model development and subsequent switching between different time structures.

The package is already used in several optimization packages developed at SINTEF, e.g., EnergyModelsX, ZeroKyst, and MaritimeNH3.

## Statement of need

For complex optimization models, a significant amount of code is typically used to track the relationships between time periods, further complicated if stochastic versions of the model are developed. Time constraints can be tricky to implement correctly. They can be a source of subtle bugs, in particular when more complicated structures are involved in models with linking constraints between time periods or scenarios. One example of these types of constraints is in keeping track of a storage inventory over time or incorporating dispatch constraints.

Modellers typically use extra indices to keep track of time and scenarios, making the code harder to read, maintain and change to support other or multiple time structures. This complexity can be an obstacle during development, testing and debugging, as it is easier to work with simpler time structures.

By abstracting out the time structures and providing a common interface, `TimeStruct.jl` allows the modeller to concentrate on other properties of the model, keeping the code simple while supporting a large variety of time structures (pure operational, strategic/investment periods and operational periods, including operational uncertainty and/or strategic uncertainty).

Through providing a common interface with time structure semantics, `TimeStruct.jl` simplifies running a single model for different time structures. It may be hence used to develop decomposition techniques to exploit specific structures.

To the best of the authors' knowledge, no software packages with similar functionality currently exist. There are several examples of optimization models that incorporate complex time structures, particularly within energy modeling (e.g., (Loulou et al., 2016; Tejada-Arango et al., 2024)), as well as in facility location (e.g., (Correia & Melo, 2016)) and scheduling problems (e.g., (Iyer et al., 1998)) where strategic decisions are separated from operational decisions.

The Journal of Open Source Software

However, the time structures in these models are tailored to their specific applications and are tightly integrated with the models themselves, making it difficult to reuse them directly in other contexts.

## Example of use

For a full overview of the functionality of `TimeStruct.jl`, please see the online documentation.

During development and for operational analyses, simple time structures where, e.g., time is divided into discrete time periods with (operational) decision variables in time period, can be useful. With `TimeStruct.jl`, such structures can be easily used in any optimization model. The example in Figure 1 shows the basic time structure `SimpleTimes` which represents a continuous period of time divided into individual time periods of varying duration. The length of each time period is obtained by the `duration(t)` function.


**Figure 1:** Simple time structure with only operational periods.

One of the main motivations for the development of `TimeStruct.jl` is to support multi-horizon time structures (Kaut et al., 2014). As a simple example of a multi-horizon time structure, the time structure `TwoLevel` allows for a two level approach, combining an ordered sequence of strategic periods (typically used for binary capacity expansion) with given duration and an associated operational time structure (for operational decisions using the available capacity in the associated strategic period) as illustrated in Figure 2.

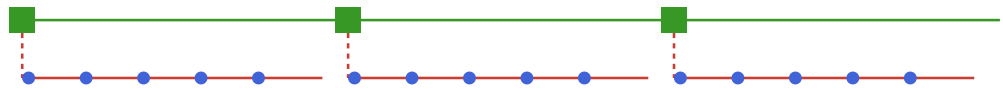

**Figure 2:** A typical two-level time structure.

Using the interfaces defined in `TimeStruct.jl`, it is easy to write models that are valid across different time structures. The following example shows a simple model with a production variable, $x$, defined for all operational time periods and a constraint on the maximum total production cost for each strategic period:

```
using JuMP, TimeStruct
function create_model(periods::TimeStructure, cost::TimeProfile, max_cost)
    model = Model()
    @variable(model, x[periods])
    for sp in strat_periods(periods)
      @constraint(model, sum(cost[t] * x[t] for t in sp) <= max_cost)
    end
    return model
end
```

This model will be valid for both examples above, producing one constraint for the `SimpleTimes` and three constraints for the strategic periods of the `TwoLevel` example.

```
latex_formulation(
  create_model(SimpleTimes([1, 1, 1, 5, 5]), FixedProfile(3), 10)
)
```

$$3x_{t1} + 3x_{t2} + 3x_{t3} + 3x_{t4} + 3x_{t5} \leq 10$$

```
latex_formulation(
  create_model(TwoLevel(3, SimpleTimes(5,1)), StrategicProfile([3, 4, 5]), 10)
)
```

$$\begin{aligned}
3x_{sp1-t1} + 3x_{sp1-t2} + 3x_{sp1-t3} + 3x_{sp1-t4} + 3x_{sp1-t5} \leq 10 \\
4x_{sp2-t1} + 4x_{sp2-t2} + 4x_{sp2-t3} + 4x_{sp2-t4} + 4x_{sp2-t5} \leq 10 \\
5x_{sp3-t1} + 5x_{sp3-t2} + 5x_{sp3-t3} + 5x_{sp3-t4} + 5x_{sp3-t5} \leq 10
\end{aligned}$$

Different time structures may be combined to construct more complex structures. One example is the combination of a `TwoLevel` time structure with more complex operational structures like `RepresentativePeriods` and `OperationalScenarios`. These may be used alone or in combination, as shown in Figure 3.

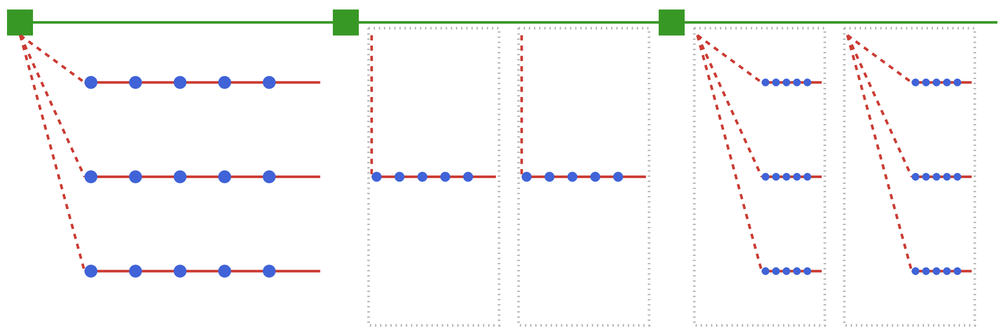

**Figure 3:** A more complex two-level time structure.

`TimeStruct.jl` also provides data structures for representing parameter data, providing efficient representation and indexing by time period for data with varying level of redundancy. Functionality for computation of discount factors for each time period to facilitate calculation of present values is also included.

## Example applications

`TimeStruct.jl` is used in multiple optimization models developed at SINTEF. One early application is in EnergyModelsX (Hellemo et al., 2024), simplifying the code in `EnergyModelsBase.jl` considerably, and allowing to add capabilities for stochastic programming versions of the model with little extra effort, see, e.g., (Bødal et al., 2024; Svendsmark et al., 2024) for example applications of EnergyModelsX.

It has also been used in the logistics models developed in the project 'Sirkulær masseforvaltning' for planning in the rock and gravel industry, as well as for hydrogen facility location optimization in the 'ZeroKyst' project. Ongoing activities in the EU funded projects 'H2GLASS' and 'FLEX4FACT' involve the use of `TimeStruct.jl` (Kitch et al., 2024).

## Acknowledgements

The development of `TimeStruct` was funded by the Research Council of Norway through the projects ZeroKyst (328721), MaritimeNH3 (328679) and Clean Export (308811)

## References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bødal, E. F., Holm, S. E., Subramanian, A., Durakovic, G., Pinel, D., Hellemo, L., Ortiz, M. M., Knudsen, B. R., & Straus, J. (2024). Hydrogen for harvesting the potential of offshore wind: A North Sea case study. *Applied Energy*, *357*, 122484. https://doi.org/10.1016/j.apenergy.2023.122484

Correia, I., & Melo, T. (2016). Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research*, *255*(3), 729–746. https://doi.org/10.1016/j.ejor.2016.06.039

Hellemo, L., Bødal, E. F., Holm, S. E., Pinel, D., & Straus, J. (2024). EnergyModelsX: Flexible energy systems modelling with multiple dispatch. *Journal of Open Source Software*, *9*(97), 6619. https://doi.org/10.21105/joss.06619

Iyer, R., Grossmann, I., Vasantharajan, S., & Cullick, A. (1998). Optimal planning and scheduling of offshore oil field infrastructure investment and operations. *Industrial & Engineering Chemistry Research*, *37*(4), 1380–1397. https://doi.org/10.1021/ie970532x

Kaut, M., Midthun, K. T., Werner, A. S., Tomasgard, A., Hellemo, L., & Fodstad, M. (2014). Multi-horizon stochastic programming. *Computational Management Science*, *11*(1), 179–193. https://doi.org/10.1007/s10287-013-0182-6

Kitch, M., Støen, M., Muñoz Ortiz, M., Lima Silva, T., Juanpera, M., & Fisco-Compte, P. (2024). Optimal industrial clusters: Flex4Fact summer research report. *SINTEF AS (978-82-14-07014-9)*.

Loulou, R., Goldstein, G., Kanudia, A., Lettila, A., & Remme, U. (2016). *Documentation for the TIMES model*. IEA Energy Technology Systems Analysis Programme.

Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., & Vielma, J. P. (2023). JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*. https://doi.org/10.1007/s12532-023-00239-3

Svendsmark, E., Straus, J., & Granado, P. C. del. (2024). Developing hydrogen energy hubs: The role of $\mathrm{H}_2$ prices, wind power and infrastructure investments in Northern Norway. *Applied Energy*, *376*, 124130. https://doi.org/10.1016/j.apenergy.2024.124130

Tejada-Arango, D., Morales-España, G., Clisby, L., Wang, N., Soares Siqueira, A., Ali, S., Soucasse, L., & Neustroev, G. (2024). *Tulipa energy model*. Zenodo. https://doi.org/10.5281/zenodo.8363262