Improving the performance of non-linear multi-objective optimization problems

case study – Rijnland operational system

Author:

Ailbhe Mitchell

Project team:

Ailbhe Mitchell

Bernhard Becker

Sjoerd Geevers

Summary

Within optimization, the added accuracy in using non-linear relationships often comes with the trade-off of longer run times. A software solution was developed which allows modellers to easily import a seed to an RTC-Tools optimization run. Combined with time-dependent linearization, performance gains of 60% regarding run-times were observed in complex, non-linear multi-objective operational systems.

With improved run-times, non-linear optimization problems become more accessible. Clients can solve more accurate models of their systems, allowing them to make more informed decisions about how they use energy. Additionally, the advice of the model is far more valuable when run times are fast and advice can be implemented more quickly before conditions change.

Introduction

During this project, as part of the mathematics for Industry course at Utrecht Universiteit, a group of 7 students worked for 10 weeks developing ideas and designing a prototype implementation to speed up solving non-linear optimization problems. Following this student project, Deltares generalized this approach and combined it with other initiatives to evaluate the performance and integrate these methods within the RTC-Tools software.

The aim was to find methods which could be applied to the operational Rijnland system which consists of a complex non-linear optimization of the water system. This model has many variables and thus was deemed too complex for students to focus on for a short period of time. Thus, students considered the example model “cascading channels” available as part of the RTC-Tools repository to test approaches to speed up non-linear models with few variables.

In this report we consider

1. Motivation
2. Results from the student project
3. Generalizations to apply method within RTC-tools
4. Evaluation of results applied to Rijnland system
5. Conclusions
6. Next steps

1. Motivation

Non-linear optimizations problems are in general more difficult to solve than linear ones, in part due to their nature with the possibility of many local minima. Methods are applied in RTC-Tools to support solving such problems. In this project we aim to improve these and provide scientific foundation to the methodology. Therefore, the goals of the project were two-fold

a. Find ways to solve non-linear problems more efficiently.
b. Explore if mathematical statements be made about these approaches.

Homotopy is a method implemented in RTC-Tools with the aim to facilitate the solving of non-linear optimization problems. At is core, this is a continuation method which first solves and linearized problem, and then perturbs a parameter from 0 to 1 until the non-linear problem is solved. We refer to this parameter as theta.

The idea here is to achieve a path stable solution by slowly perturbing the equations defining the optimization problem. The seed (initial guess for the solution) is given by the solution to the previous problem. The assumption in the current methodology is that the global minima for the linearized problem will be close to a local minima of the non-linear problem. In practise, a global minima to the non-linear problem is not necessary, but it is important that advices to not change much from one hour to the next.

Claims can be made regarding path stability when using this method with a single optimization (single objective), but it is more difficult to make such claims for multi-objective optimization problems solved in a lexicographic way. In Lexicographic goal programming, goals are solved in order of priority, where constraints are added sequentially such that a high priority objective cannot be met any worse when solving for a low priority objective.

For example, consider we have two goals, listed in order of priority,

1. keep water levels within bounds
2. minimizing energy consumption

Then we first solve the optimization problem over the entire time horizon with the single objective to keep water levels within bounds. If it is possible to keep water levels within bounds at all times, then this will also be reflected in the solution to the optimization problem for the second priority when we minimize energy consumption. If it was not possible to keep water levels within bounds at all times, then we make sure the exceedance of these bounds is no worse when also solving for the second priority.

With this lexicographic approach, no research had been conducted on the impacts of continuation methods (homotopy) on path stability.

The homotopy method as applied in RTC-Tools required solving optimization problems for all priorities for each value of theta. If a step size of 0.5 is considered sufficient (default setting), this means three times as many optimization problems are solved as there are priorities. This can become very computationally expensive as

- These runs cannot be parallelized as they rely on the solution to the previous problem in the sequence for a seed.
- Non-linear optimization problems in general take longer to solve than linear problems, only optimization problems where theta=0 (1/3 of problems) are linear and easy to solve.

Therefore, methods were explored which would improve the existing workflow and lead to a reduction in run-time.

2. Results from the student project

Conclusions:

a. **Computational efficiency: A method was outlined to speed up solutions to non-linear problems by providing a seed from a previous model run, and directly solving the non-linear problem.**
   This approach was tested on the minimal model case "casacading channels" and the results showed that we could expect a speed up in model performance when the provided seed was close enough to the optimal advice.

b. **Mathematical theory: Although we have tried to study path-connectedness rigorously, and it does look promising for multi-objective optimization, the topic still requires more research. Our endeavours do not guarantee global path-connectedness for the goal-programming approach. However, we can guarantee local optimality.**
   The main insight is that we think one needs to focus on one objective at a time to ensure a traceable Homotopy path. This supports the approach outlined in section 4, experiment "b" where only the first priority is solved for theta != 0.

For details we refer to the report written by the students.

3. Generalizations to apply method within RTC-tools

The students result showed promise but the implementation was specific to the "cascading channels" example. The next steps were to generalize this feature within RTC-Tools and ensure it is robust enough for use in an operational context. This consists of the following steps

- Method to provide seed to the model (in a transparent way)
  - MultiSeedMixin: An/multiple additional xml timeseries can be provided to the model containing seeds for variables. RTC-Tools will detect for which timesteps relevant to the current optimization problem a seed has been provided and use this for the first optimization problem.
    - This approach allows for simple provision of seeds. For example via FEWS – this can be the full solution to a previous model run.
    - Values do not need to be provided for all variables, nor for all timesteps. If no seed is provided the default will be used.
    - By allowing separate inputs for the seed and the timeseries_import this ensures maintenance for the models is simple – it is clear which values are used for seeds only, and which are read to the model as bounds/targets/additional information.
      - This also allows for seeds to be provided for "controlled inputs". This addition results in significant performance gains.
    - Multiple seeds can be provided to the model and chosen dynamically or used as a fallback (see below).
- Fallback options such that the model will not fail due to a bad seed (possible in extreme conditions where forecasts/conditions change in a short amount of time).
  - RTC-tools code refactored to make seeding of the model transparent in all cases

        - If a seed is used then an info message is added to the logger
    - General goal programming:
        - imported seed is used for solving the very first priority. If this priority fails, the model will be resolved using the default seed
    - Homotopy:
        - If an imported seed is provided then the non-linear problem can be solved directly. If this fails then the fallback option is to solve the problem using the usual homotopy routine.
- Unit tests
    - Added to ensure new code is included in code coverage.

Throughout this process we collaborated with the community of contributors to RTC-tools as it is an open source project. This highlighted the need for an agreed-upon governance structure for the software, as no existing framework was in place to resolve disagreements among stakeholders.

4. *Evaluation of results applied to Rijnland system*

Homotopy + time-dependent linearized time horizon

In parallel to the student project, developments were implemented in the Rijnland model to decrease run time by other methods. This included choosing a portion of the time horizon to linearize thus simplifying the optimization problem. Such an approach is appropriate in this case as the model runs for a 48 hour period and is rerun every hour with updated inputs/forecasts. Thus, the end of the time horizon is only included in a model run such that optimal advice is given upcoming hours. However, for a given run, the advice for the end of the time horizon is not implemented directly – the model will have re-run 47 times before that time with updated data. This approach also included solving only the first priority of the optimization problem until the full non-linear system is considered (theta = 1). The thesis behind this is that this is the most efficient way to solve the problem without an external seed provided, and also aligns with the conclusion above regarding path stability for multi-objective optimization.

We studied the results of the optimization where the linearization described above was used. It was concluded that there was not a significant loss in accuracy. That is because the model is re-run every hours with a 48hr time horizon. It is important to include the full 48 hour forecast in the model run such that the advice for the first hour is influenced by future forecasted events, but the forecast has higher uncertainty in the future.

An example is seen in the figure below. Here we consider various experiments, a and b contain no linearization, c and d contain linearization of the final 24 or 36 hours of the time horizons. We see that the advice for the first few hours remains consistent between all experiments and then begins to deviate. Since it is only the advice of the first hour which is implemented, this is not a cause for concern.

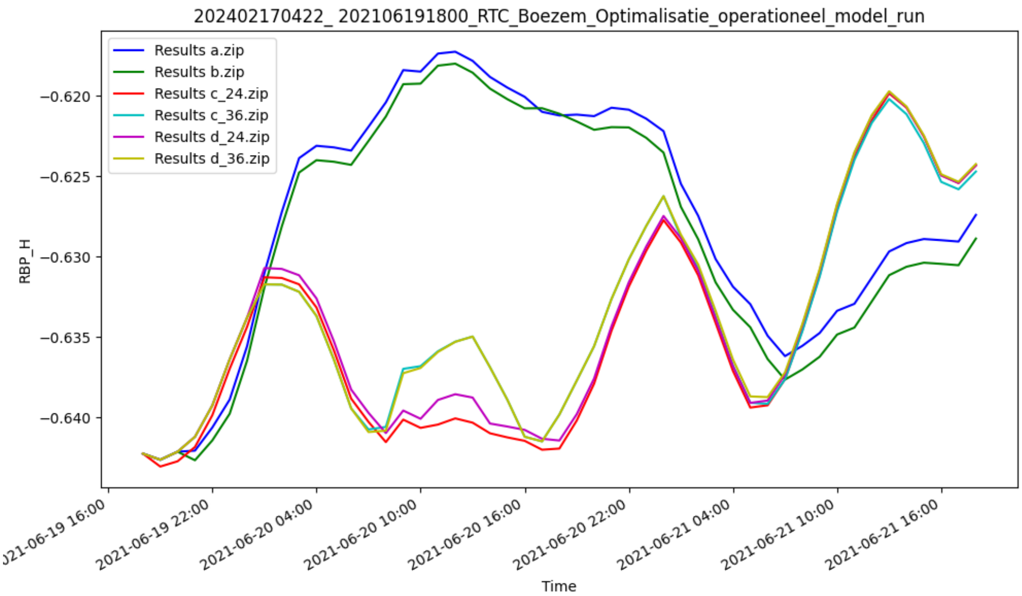

Figure1Results showing the impact of linearizing portions of the time horizon on model results. It can be seen that the advice for the first few hours remains largely unchanged

Combined results

We tested the same batch of 80 closed loop runs with various model set ups, finally incorporating seeding with the approach outlined above. Concretely these were

- “original”: Original set up: no seeding, default homotopy options
  - Solve every priority for theta = 0 (linear)
  - Solve every priority for theta = 1/2
  - Solve every priority for theta = 1 (non-linear
- “b”: solve only the fist priority for theta !=1
- “c(24)”: linearize final 24 hours of time horizon
- “d(24)”: solve only the first priority for theta !=1 and linearize final 24 hours of time horizon
- “seed”: solve non-linear problem directly and linearize final 24 hours of time horizon and provide a seed (solution from previous run)

It can be seen in the boxplots below that average run times are significantly decreased when using the final approach “seed” compared to all other methods. A performance increase of more than 60% in terms of run time is observed with using the new available approach with seeding plus linearization compared with the original set up.

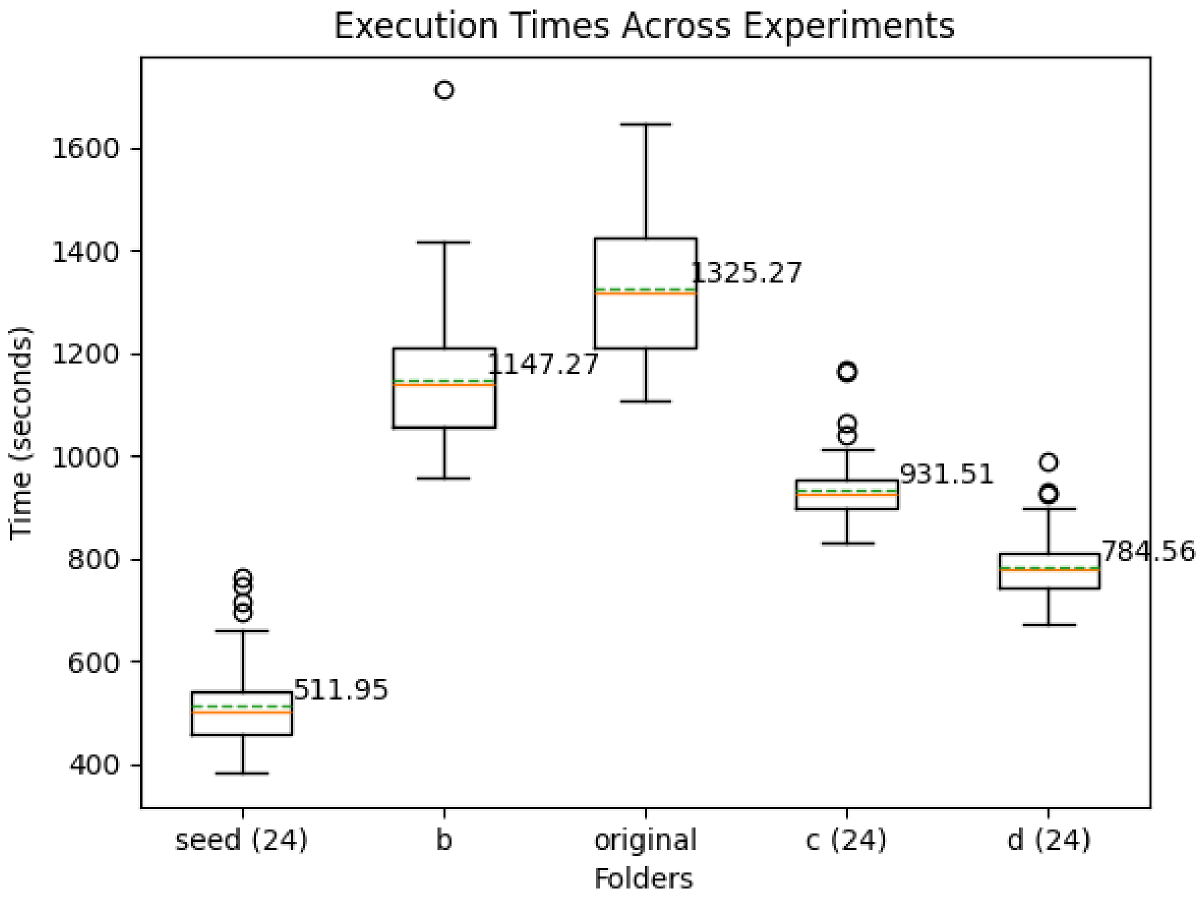

Figure 2: boxplots showing the runtimes of identical closed loop models under different set-ups. Mean run times are also displayed per experiment. A significant improvement can be observed when comparing the original setup (center) with the results from this project (far left).

5. Conclusions

In this project we were able to develop a software solution which allows modellers to easily import a seed to an RTC-Tools optimization run which can be used to significantly speed up the run time of a model. This is particularly useful for non-linear optimization where the added accuracy in using non-linear relationships often comes with the trade-off of longer run times.

With improved run-times for non-linear optimization problems, clients can solve more accurate models of their systems, allowing them to make more informed decisions about how they use energy. For instance in the Rijnland system, one objective of the model is to minimize energy consumption. The advice of the model is far more valuable when run times are fast and advice can be implemented more quickly.

The insights and time spent by the students from Utrecht was invaluable. They were able to explore the problem through an original lens, and utilizes their backgrounds in an array of mathematical disciplines to propose fresh approaches.

6. Next steps

We have seen significant improvements in the run time for complex non-linear models which are directly applicable to other operational systems we maintain and develop.

Applications: This method is applicable in all applications where and RTC-Tools advice module is solved in a closed loop environment. Both for non-linear models using homotopy, or general goal programming applications. This approach could also be applied in RTC-Tools simulation where non-linear equations are often solved – when very complex the seed can be important.

Integration with AI/ML: In applications where models are not run in closed loop, it would be possible to use AI/ML algorithms to find a suitable seed from a database of previous model runs. This process could assess the inputs to the models, find a similar import and provide the result of this run as a seed to the model. The approach we have implemented also supports multiple seeds to be provided to the model.

Analysis of objectives: Often if a good seed is provided then high priority objectives are already satisfied by the seed. If we can determine this in advance then we would save the time spent in solving these objectives. This is likely applicable in many closed loop operational systems as advice should not vary much from one model run to the next, and often high priority objectives are expected to be satisfied under non-extreme conditions. Note: for such an approach the provided seed should be extrapolated for the final timestep(s) which do not overlap with the current run.

Model explainability: Within the Rijnland system there is potential to further increase model performance by performing the "pump analysis" in a more efficient manner. This is a module designed to evaluate which goals types have the largest impact on the final advice of the model. Whether these be energy related, salt driven or water levels. The current implementation means solving a sub (non-linear) optimization problem after each priority which is a very expensive approach. In fact with the new implementation from this project, the pumping analysis can account for as much as 40% of the run time. Therefore, alternative approaches should be considered. In fact this module answers a question which is valuable information for every operational system with an optimization advice model where goal programming is used. Therefore it would be invaluable to develop a generic approach which could be easily applied to RTC-Tools models in general. This would lead to great advancements in the explainability of optimization advice for water management.

Other hot starting options: Different optimization problems support different methods of warm or hot starting the solver. This can often lead to large improvements in computation time. For instance MIP solvers allow users to provide a basis which can often be more valuable than a seed. Once such functionality is supported by casadi this could further improve run-times for operational models which are run with high frequency.

Mathematical theory: Exploring path stability for multi-objective non-linear optimization problems remains an interesting topic on the cutting edge of operations research. More time invested in investigating the underlying problem definition and path-connectedness would likely lead to strong supporting claims for our applied methodology, or better methods of solving such problems.

Validation of linearized equations: We observed significant differences in solutions when using linearized/non-linearized equations. Validation should be performed on the linearization of these equations as they have a large influence on the solution.

Model explainability: In the Rijnland system, the model explainability module now accounts for ~50% of the run-time, more computationally efficient and accurate methods should be considered which delivers core information to decision makers.