Appendix N

Multi-Objective Evolutionary Algorithm (MOEA) Tool Utilization and Development

MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM (MOEA) TOOL UTILIZATION AND DEVELOPMENT

TECHNICAL MEMORANDUM

Truckee Basin Water Management Options Pilot Study

2021 Hydrologic Engineering Analysis Tasks, Task J.1: "Review Available Information and Select Preferred MOEA Algorithm and Wrapper"

June 14, 2022

Truckee Meadows Water Authority
Reno, NV

Precision Water Resources Engineering
Loveland, CO

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

The purpose of this technical memorandum is to describe the tools that were developed by Precision Water Resources Engineering (PWRE) to support the Multi-Objective Evolutionary Analysis (MOEA) component of the Truckee Basin Water Management Options Pilot Study (WMOP).

This includes:

- MOEA Background (**Section 2**):
    - A brief summary of what MOEA is and references for those interested in further details.
    - Why MOEA will benefit the WMOP study.
    - How and where MOEA has been used in water resource engineering and, specifically, RiverWare© modeling studies.
- MOEA Review and Selection (**Section 3**):
    - The specific needs and requirements of an MOEA tool for the WMOP study.
    - A summary of available MOEAs and related components that were considered.
    - The reasoning and discussion explaining the selection of the MOEA algorithm utilized (NSGA-II) and the MOEA "wrapper" programming framework (Python).
- MOEA Tool Development (**Section 4**):
    - What an MOEA "wrapper" is and why it is necessary.
    - A description of the previously existing and available components that were utilized.
    - How those components were integrated together to form the MOEA tool used for the WMOP study.

2 Multi-Objective Evolutionary Algorithm (MOEA) Background

2.1 What is MOEA?

Multi-Objective Evolutionary Algorithms (MOEAs) are non-linear, stochastic optimization methods that can be used to identify the best compromise solutions along a path of potential policy alternatives given a set of defined objectives and decision variables. MOEA provides an intelligent, systematic process for developing a solution that balances the achievement of multiple (often competing) objectives. It provides users a quantitative way to evaluate tradeoffs. For a thorough introduction to MOEAs, please refer to Reed et al., 2013.

Given an overall scenario "space", as defined by ranges of potential values for certain ***decision variables***, MOEAs can be used to "search" and evaluate a much wider range of policy alternatives than traditional types of optimization analysis allow. As they run, MOEAs create and test new scenarios based on where previous scenarios found success in optimizing the values of defined ***objectives***. The result of the algorithms are policy alternatives that represent the optimal results for each of the defined objectives.

Due to the multi-objective nature of the optimization analysis, there is often a competing nature between different objectives (i.e., what's good for one objective is not always good for other objectives). Thus, optimal results are generally not single solutions, but are instead represented by sets of "nondominated solutions" (also called "Pareto optimal points"). A ***nondominated solution*** is a solution that provides an optimal trade-off between objectives, in that no objective can be further improved without harming another objective. In contrast, a ***dominated solution*** is a solution where one of the objectives can be improved without harming any of the other objectives (i.e., there is no trade-off to improve that objective), and thus, is not an optimal trade-off point. The collection of nondominated solutions is often referred to as the ***Pareto front***. These concepts are illustrated below in Figure 2-1 for a conceptual two objective (i.e., two-dimensional) problem where the objectives are to minimize both the x and y values.

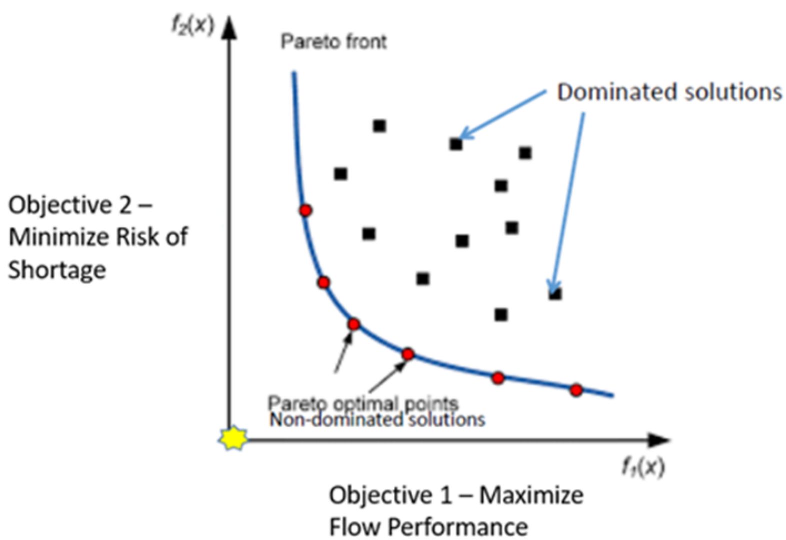


Figure 2-1: Illustration of 2D Pareto-Front and Non-dominated vs. Dominated Solutions (adapted from CADSWES, 2019).

Additionally, there are a few more important MOEA terms to define as they will be used through the rest of this technical memorandum. The ***function*** is the equation (simpler) or model (more complex) that is used to evaluate the objective values for a given set of decision variable values. In this case, the function is the TROA Planning RiverWare© model. Thus, a function ***evaluation*** is a single run of the model configured to simulate the alternative scenario that is defined by a set of decision variable values.

Furthermore, ***constraints*** can also be applied in MOEA analyses to define bounds/limits of applicable performance. Since the overall scenario space — in terms of the input ranges and combinations of decision variables as well as the output ranges of objectives — can be very large, constraints can be useful to "focus" the analysis on areas of interest or benefit. For example, the WMOP MOEA analysis will use constraints to limit the solutions to only alternatives that improve the performance of relevant objectives relative to the baseline scenario values. Constraints can also be used to exclude certain combinations of decision variables that do not represent feasible scenarios.

Finally, the term ***MOEA Wrapper*** is used to refer to the computer program that connects the various components of the MOEA process and facilitates and manages the data transfer between them. The description of the MOEA Wrapper used for the WMOP MOEA analysis, and its development, is the main subject of this technical memorandum. Thus, the MOEA Wrapper in the context of this study is thoroughly discussed throughout the rest of the report, especially in Section 4.

The general process of an MOEA analysis is as follows, and is illustrated below in Figure 2-2:

1. The MOEA Wrapper generates a scenario consisting of a unique combination of Decision Variables (DVs) that define a policy/operation alternative to be simulated. This is done randomly at first (within user-defined ranges) after which the selection of DVs begins to be informed by successful model results.
2. Next, the MOEA Wrapper sends that scenario to the function for evaluation (in this application, the function evaluation is a TROA Planning model run).
3. When the function evaluation has completed (i.e., the TROA Planning model run has finished), its results are returned to the MOEA Wrapper.
4. The MOEA Wrapper processes and sends the results back to the MOEA algorithm, which uses the successful results to inform the next alternative scenarios to be evaluated.
5. (The process repeats a user-specified number of times.)

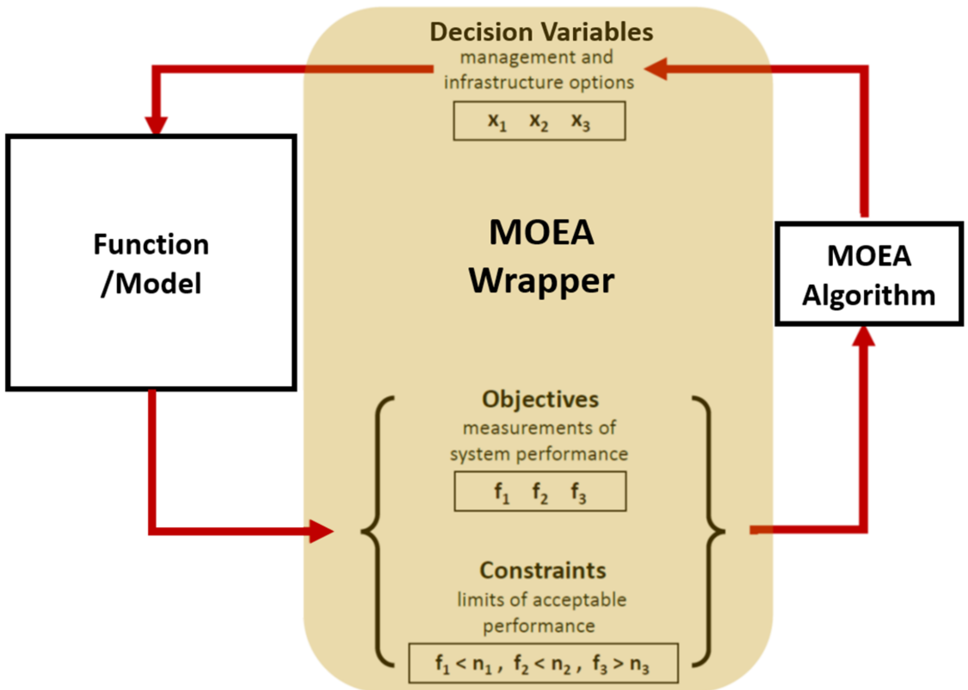


Figure 2-2: MOEA Process Schematic (adapted from CADSWES, 2019).

2.2 Why MOEA Will Benefit the WMOP Study

The current flood control policy of the Truckee Basin, as set forth in the 1985 Truckee River Water Control Manual (USACE, 1985), inherently represents some balance point of the trade-offs between the various objectives of many stakeholders throughout the basin.

For explanatory purposes, let's assume that the suite of objectives can be boiled down to two main objectives of the policy, (1) to minimize the risk of flooding and the associated damages, and (2) to maximize water supply. If the developers of the policy were only concerned with flood control, they would not have allowed any storage in the reservoirs outside of flood operations, which would have minimized the risk of flood damages as much as possible. However, this would clearly not be favorable for the water supply objective as there would have been no reservoir storage space that could be used to benefit water supply. Conversely, it they were only concerned with maximizing water supply, they would have not required that any empty space be reserved in the reservoirs to absorb and mitigate flooding. That policy would have maximized the water supply but would not have provided any flood control benefits.

It is reasonable to assume that when the current policy was developed, it was selected because it was estimated to represent a "non-dominated" point within the trade-off relationship between objectives to the extent that was feasible at the time. If the water managers at the time had believed that they could reduce flood risk further without negatively impacting water supply, or vice-versa, it would have been prudent for them to do so.

However, there have been numerous and significant changes that the Truckee Basin has seen in the 37 years since that policy was developed. These include improvements to runoff forecasting skill through SNOTELs and RFC Ensemble forecasts, improvements to precipitation forecasting skill, the Stampede spillway project increasing its ability to surcharge in extreme events, Martis Creek Reservoir not currently being operable, and several flood mitigation projects that have occurred through the Truckee Meadows. Additionally, it is important to point out that the current policy as defined in the 1985 WCM is statically based on the forecasting process and skill in 1985, which at that point was then driven by a manual snow survey in late March/early April. Since forecasting skill (through RFC) and the associated technology is continuously evolving and improving as new methods are developed, a policy that automatically adjusts for improving forecasting ability has become even more prudent.

Thus, there is little assurance that the current policy even still represents a non-dominated solution. It is entirely possible, and perhaps even likely, that alternative flood control policies exist that would benefit some objectives without harming others, or even improve all objectives together.

Of course, there are more than these two competing objectives present in the Truckee system, and even those two simplified objectives are not as simple as they may seem. For example, who's water supply should be maximized and at what times of the year? For flood control, are peak flows to be minimized? Or should the occurrence rate and/or duration of flows above some specific threshold be minimized? In examining how the existing policy may be improved and updated, it is important to consider how any proposed changes may impact the full suite of objectives. Any changes made to the current policy may have variable impacts on the various objectives, and therefore it is prudent to conduct a thorough analysis of any potential alternatives.

As part of the WMOP study, various water managers who represent a diverse set of stakeholders and interests have come together to identify the set of objectives they feel best represent the system. Likewise, a set of decision variables has been developed to represent the basin's flood control policy, and each different combination of values of these decision variables represents a unique flood control policy alternative. Given the number and ranges of these decision variables, there may be 10s, if not 100s of thousands of unique policy alternatives that could be considered. The number of objectives and sheer size of this "decision variable space" of policy alternatives is a recipe for an immensely challenging optimization analysis.

MOEA analysis will allow the evaluation of a wide range of alternative scenarios. It will systematically evaluate the objective performance of each scenario and use what it learns is good for the various objectives to generate subsequent scenarios to evaluate. After a long run time consisting of thousands of TROA Planning Model runs, the MOEA's output will consist of a set of non-dominated scenarios. Within this set, each alternative scenario will represent some optimal scenario in terms of the trade-off relationships between the objectives. As the MOEA does not prioritize any of the objectives over any others, this set of non-dominated solutions will facilitate discussion and negotiation among the basin stakeholders to select a policy scenario that represents an optimal balance point between objectives.

2.3 Recent Applicable Water Resource MOEA Studies

In the recent years, MOEAs have been used to assist decision making in several high-profile river basins, including the Colorado River basin. As a relevant example, Alexander et al. (2018) explored using the

Borg-MOEA in the Colorado River basin to evaluate different policy alternatives with regard to operating Lakes Mead and Powell with specific Lower and Upper Basin objectives relating to Colorado River Compact. The study evaluated provisions of the 2007 Interim Guidelines and tested the MOEA process as a tool to be used in future analysis and Colorado Basin negotiations. The project has led to continued and expanded use of the Borg-MOEA with the Colorado River Simulation System (CRSS) RiverWare© model.

In 2021, Precision Water Resources conducted an MOEA analysis for the San Juan Recovery Implementation Program (SJRIP) to explore how well some of Navajo Reservoir's operational policy parameters (specifically relating to the SJRIP spring peak release schedules) were optimized in regard to the competing objectives of meeting fish flow targets as well as maximizing water supply to the basin's users (SJRIP, 2022). For the analysis, the Borg-MOEA was connected to the SJRIP RiverWare© model with the Borg-RiverWare© wrapper (see Section 3.3). The full SJRIP RiverWare© model run (85-years at daily timestep) takes over an hour to run, and it was found early on in the project that the long model run times were going to be problematic for MOEA analysis that depends on thousands of model runs. To overcome the issue for the SJRIP study, the 85-year full model period was divided into 12 overlapping 30-year hydrology sequences and a RiverWare© multiple run manager (MRM) was configured so that all 12 runs could be run simultaneously which took about 10 minutes per MOEA iteration. This allowed a 2000-iteration MOEA to be completed in approximately 2 weeks of computer run time. Although at the low-end of recommended iterations, it was found that the 2,000-iteration analysis did a satisfactory job of defining the pareto front for the problem with 3 decision variables and 7 objectives.

In 2018, RTI International partnered with the U.S. Army Corps of Engineers Fort Worth District to explore the potential for multi-objective reservoir optimization, and their effort was presented as a proof-of-concept at the RiverWare© User Group Meeting (Watson et al., 2018). They used a hypothetical reservoir with three operational objectives (flood risk management, hydropower generation, and water supply) and several operational challenges including coordinated operations with a downstream reservoir and an out-of-date (i.e., no longer optimized) operation manual. A framework was developed between a RiverWare© model of the hypothetical reservoir system and the NSGA-II algorithm. Significantly, the framework and the NSGA-II algorithm were successfully implemented to allow for parallel RiverWare© runs during each MOEA-iteration. Another important take-away was that they recommended using appropriate constraints to ensure that MOEA-generated alternative scenarios represent relatively realistic options.

3 MOEA Options – Review and Selection

3.1 Overview

Many MOEA algorithms exist, and the list is continues to expand due to the fact that MOEA, and optimization analysis in general, is a popular and rapidly evolving field of study. For example, the pymoo.org framework and resources (discussed in Section 3.5), currently provides 17 algorithms. MOEA algorithms differ by purpose, type of optimization problem (single vs. multi vs. many objectives), scenario generation criteria and processes, complexity of implementation, and many more factors.

In-depth review of the multitude of MOEA algorithm options was not a focus of this study. Rather, the best-known MOEA options in the field of civil and water resources engineering were evaluated for their ability to be integrated with the TROA Planning RiverWare© Model and to support the needs of the WMOP study.

Ultimately, two MOEA algorithms were considered for the WMOP study, the NSGA-II and Borg-MOEA algorithms. These two algorithms are further discussed below.

3.2 Requirements of an MOEA Tool for the WMOP Study

There are several important needs and requirements for an MOEA tool for the WMOP study. These include, but are not limited to:

- Must provide a sufficient MOEA search with a documented history of use in civil and water resource engineering project.
- Must be able to be integrated into a “wrapper” that automates the connection between the MOEA algorithm and the “function”, which for this study is the TROA Planning Model.
- Must be able to be run in parallel.
    - MOEAs are typically designed to work with thousands and even millions of function evaluations (model runs), generally, the more the better.
    - Each 31-year run of TROA Planning Model used for the MOEA study takes about 2 hours to run, and therefore overall run time and computational resource availability is a significant concern.
    - For 2000 model runs conducted in series (one after the other), this would take 167 days to run. This is obviously infeasible for the WMOP project.
    - If 15 runs can be conducted at a time in parallel, however, this reduces the estimated run time to 11 days.
- Must provide relatively efficient convergence, for the same reasons as the previous point.
- Must allow for both continuous and discrete decision variables.
- Must allow for constraints on objective performance, which are used to eliminate non-dominated solutions outside of a desired range (in this case, worse than baseline model performance), in order to facilitate efficient search and convergence.
- Must be able to be stopped and restarted without needing to run the entire analysis up to that point over again (efficiency and stability concern).

- Flexible enough to allow for customization of the analysis process as needed. For example, the functional evaluation is a 31-year daily timestep Truckee RiverWare© planning model run, however, within that model run there may be multiple hourly model runs that must be kicked off at certain points within the daily run and are dependencies to the daily model run itself. The MOEA tool must be able to support this process.

3.3 Borg-MOEA and Borg-RiverWare©

The Borg Multiobjective Evolutionary Algorithm (MOEA) is an optimization algorithm developed by David Hadka and Patrick Reed at the Pennsylvania State University (http://borgmoea.org/). Borg-MOEA was developed from precursor MOEAs (including NSGA-II) and related algorithms to improve efficiency and functionality and to reduce issues that were observed in other MOEA options (Hadka & Reed, 2013).

MOEAs being used in coordination with RiverWare© models as an optimization tool is a relatively new front, although there is some precedence with other Reclamation modeling applications as mentioned above. To support those and other efforts, CADSWES has developed a Borg-RiverWare© (aka Borg-RW) wrapper application that links the BORG-MOEA algorithm with RiverWare© models. This development began around 2017 and continues through the present. The Borg-RW wrapper facilitates communication and data transfer between a RiverWare© model and the Borg-MOEA algorithm (CADSWES, 2019 & 2021).

Borg-RiverWare© was not selected for the WMOP study because it does not support parallel function evaluations. Supporting parallel evaluations is a functional requirement for the project because it is necessary to generate a high enough number of function evaluations (i.e., model runs) within a reasonable timeframe. This limitation was the chief reason that Borg-MOEA and Borg-RiverWare© was ultimately determined to not be an option for the WMOP study.

3.4 NSGA-II

The NSGA-II — Non-dominated Sorting Genetic Algorithm II — algorithm was first published in 2000 (Deb et al., 2000) and quickly became popular. The algorithm was developed to improve upon several of the main criticisms of algorithms that depend on non-dominated sorting. The NSGA-II sorting approach has been shown to be fast and computationally efficient. NSGA-II also contained significant improvements to the selection and sharing operators, which have to do with how the algorithm generates new sets of decision variables based on the objective performance values of past scenarios. The NSGA-II scenario generation process is also built in a manner that pays attention to not only the objective performance (fitness), but also the spread of the scenarios, which helps it find a better spread of solutions and a better representation of the true Pareto-optimal front relative to other algorithms.

Today, NSGA-II remains a popular and efficient algorithm and is one of the oldest MOEAs that is still in active use today (Hadka and Reed, 2012, and Blank and Deb, 2020). It has been used in many civil engineering applications, both in applied engineering and in research-centric fields, including in water resources.

Ultimately, NSGA-II was selected as the MOEA algorithm for the WMOP study for a few primary reasons. First, it is well-established and has been found to be successful and dependable in civil engineering applications. Second, it is available and supported within the pymoo.org framework (discussed next) that was critical for the development the MOEA Wrapper. Further, it does not require reference points or directions or further complicating inputs to the analysis, and therefore is more straightforward to implement, especially with varied objectives that may be subject to change through several MOEA analyses.

3.5 PYMOO.ORG

Developing an MOEA Wrapper program that can meet the many requirements outlined above necessitates a flexible and accessible programming framework. Python is a popular and powerful programming language with many applications across a broad range of scientific and engineering industries. The need for a flexible MOEA framework that can be adapted to meet the specific needs of a given project has led to the development of pymoo.org and the many resources that it makes available (Blank and Deb, 2020). It should be noted that one of the developers of the NSGA-II algorithm and one of the most renowned MOEA researchers, Kalyanmoy Deb of Michigan State University, is also one of the primary developers of the pymoo.org resources.

Abstract from "Pymoo: Multi-Objective Optimization in Python". Blank and Deb, 2020.

> *Python has become the programming language of choice for research and industry projects related to data science, machine learning, and deep learning. Since optimization is an inherent part of these research fields, more optimization related frameworks have arisen in the past few years. Only a few of them support optimization of multiple conflicting objectives at a time, but do not provide comprehensive tools for a complete multi-objective optimization task. To address this issue, we have developed pymoo, a multi-objective optimization framework in Python. We provide a guide to getting started with our framework by demonstrating the implementation of an exemplary constrained multi-objective optimization scenario. Moreover, we give a high-level overview of the architecture of pymoo to show its capabilities followed by an explanation of each module and its corresponding sub-modules. The implementations in our framework are customizable and algorithms can be modified/extended by supplying custom operators. Moreover, a variety of single, multi- and many-objective test problems are provided and gradients can be retrieved by automatic differentiation out of the box. Also, pymoo addresses practical needs, such as the parallelization of function evaluations, methods to visualize low and high-dimensional spaces, and tools for multi-criteria decision making. For more information about pymoo, readers are encouraged to visit: https://pymoo.org.*

The pymoo resources are varied, flexible, and customizable. They include python-based implementations of MOEA algorithms, function execution and processing managers, and example and test problems that facilitate development of custom MOEA applications. Especially important for the WMOP study, the pymoo framework can support parallelization of function evaluations in the MOEA context. Furthermore, the pymoo framework supports decision variables of different types, including continuous, discrete, binary, and mixed variable applications.

The pymoo resources were heavily utilized for the development of the python-NSGA-II MOEA Wrapper for the WMOP project. These resources greatly sped up the development by providing many generalized components, such as the NSGA-II MOEA algorithm itself. These components have been widely used and reviewed and are ready out-of-the-box for integration into a custom MOEA Wrapper application. Thus, the focus and effort of the MOEA Wrapper development can be where it should be, on implementing a specific problem (optimizing the Truckee River flood operation policy) and evaluation function (the TROA planning model), rather than on coding the complex MOEA algorithm process or other advanced componentry.

4 MOEA TOOL DEVELOPMENT – PYTHON NSGA-II WRAPPER

4.1 WHAT IS AN MOEA WRAPPER?

The term MOEA Wrapper is used to refer to the computer program that connects the various components of the MOEA process and facilitates and manages the data transfer between them. The description of the MOEA Wrapper used for the WMOP MOEA analysis, and its development, is the main subject of this technical memorandum.

As illustrated in the generalized Figure 2-2 above, the MOEA algorithm and the evaluation function (i.e., the RiverWare© model) are independent of the MOEA Wrapper and may be considered different modules that the wrapper connects to. Development of the TROA Planning Model for the WMOP study is covered within a different report. It is also important to note that Precision Water Resources did not write or modify the NSGA-II algorithm code from the version obtained from pymoo, which is documented at https://pymoo.org/algorithms/moo/nsga2.html. Rather, the MOEA Wrapper was developed to configure the MOEA problem (the combination of the model, objectives, decision variables, and constraints), and then execute the MOEA analysis by calling the MOEA algorithm and managing the transfer of information between the algorithm and the model runs.

Additionally, aside from facilitating the analysis by configuring and executing the problem, the other major function of the MOEA Wrapper is file and processing management. This is no small task given that an MOEA analysis consists of thousands of model runs conducted in both series and parallel, and thus it must be ensured that the inputs, outputs, and other supporting files for each model run are kept independent and are not allowed to interfere with each other.

4.2 COMPONENTS OF THE MOEA WRAPPER

The following is a list of the primary components that were utilized to construct the WMOP study MOEA wrapper:

Python programming language, version 3.9.6:

- Python was selected as the primary programming language for the WMOP MOEA Wrapper due to its flexibility, broad user base, and extensive support resources.
- Additionally, the pymoo framework and resources are also implemented in Python.

Visual Studio Code, version 1.67.2:

- The Visual Studio Code source-code editing software was used to write, debug, test, and execute the MOEA Wrapper.

Python and Pymoo MOEA classes, functions, and algorithms:

- NSGA-II algorithm (pymoo)

  - The NSGA-II algorithm is a python class module available through pymoo that is imported and utilized by the MOEA Wrapper. The algorithm's code itself was not modified in any way from the publicly available pymoo implementation.
  - Source and documentation: https://pymoo.org/algorithms/moo/nsga2.html
- Variable Types and Sampling/Crossover/Mutation Processes (pymoo)
  - The WMOP MOEA wrapper was developed to support mixed variable types, including continuous, discrete, and binary decision variables.
  - Source and documentation: https://pymoo.org/customization/mixed.html
- Parallelization of Processes (pymoo)
  - The starmap, multiprocessing parallelization object was utilized to allow for parallelization of the function evaluations during each iteration of the MOEA process. This means that some number (i.e., the number of parallel runs that can be made at once, which is dependent on computer resources) of new alternative scenarios are generated by the MOEA process based on the non-dominated solution set as it exists prior to any of those alternative scenarios being run. Then, following those runs being made, the results are digested together back into the MOEA algorithm and evaluated for inclusion in the updated non-dominated solution set. This is different than an MOEA analysis conducted purely in series, where one alternative scenario is evaluated at a time, and the results of all previous scenarios can then impact the generation of the next scenario.
  - Source and documentation: https://pymoo.org/problems/parallelization.html
- NumPy and Panda Python libraries
  - Various functions from these libraries were utilized throughout the MOEA wrapper.
  - The Pandas DataFrame data structure was used to organized and output the final MOEA results to csv files for further analysis.

TROA Planning Model (with TR Hourly River Model integration):

- The daily timestep TROA Planning Model is the evaluation function for the WMOP MOEA analysis. It is thoroughly documented in other reports. The version of the TROA Planning Model used for MOEA analysis has the TR Hourly River Model runs integrated within it for flood routing operations and constraint purposes. The TR Hourly River Model is also thoroughly documented in other reports.
- While the final MOEA analysis uses 31-year model runs, the length of the model run is not important to the MOEA Wrapper, and thus the MOEA Wrapper can be used to conduct analysis with runs of various lengths.

"GetStorageTargets" (aka "By-A-Model Part 2") Forecast Processing Python Script:

- This is the python script developed by PWRE that processes the forecast (hindcast) data according to the "A, B, and C" decision variable values in order to generate the daily Storage Target timeseries that is input to the RiverWare© model run. The "A, B, and C" parameters define the Exceedance-Outlook curve that is applied to the multiple outlook forecasts to get that day's flood space requirement according to the formula:

Exceedance (%) = max[min([C + A * (Outlook Days)^B, 100%], 0%]

4.3 WRAPPER DEVELOPMENT AND STRUCTURE

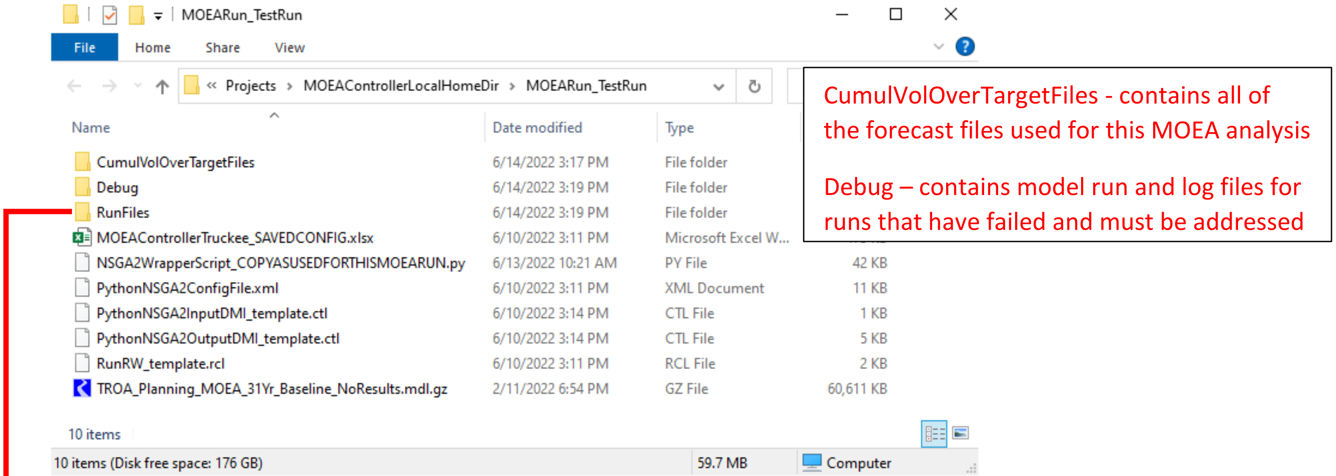
In broad terms, development of the Python-NSGA-II MOEA Wrapper for the WMOP project simply consisted of linking the various existing components together in a programming context to allow an MOEA analysis to be defined. Global and local environment variables are utilized by the MOEA Wrapper and are coordinated between the Excel MOEA Controller Workbook and the RiverWare© model and its supporting files. These environment variables to manage the various run directory paths and the unique paths of each separate RiverWare© model run (which are identified by a unique "RunID" string consisting of the decision variable values separated by underscores).

There are essentially 5 parts of the python MOEA Wrapper module, which do the following:

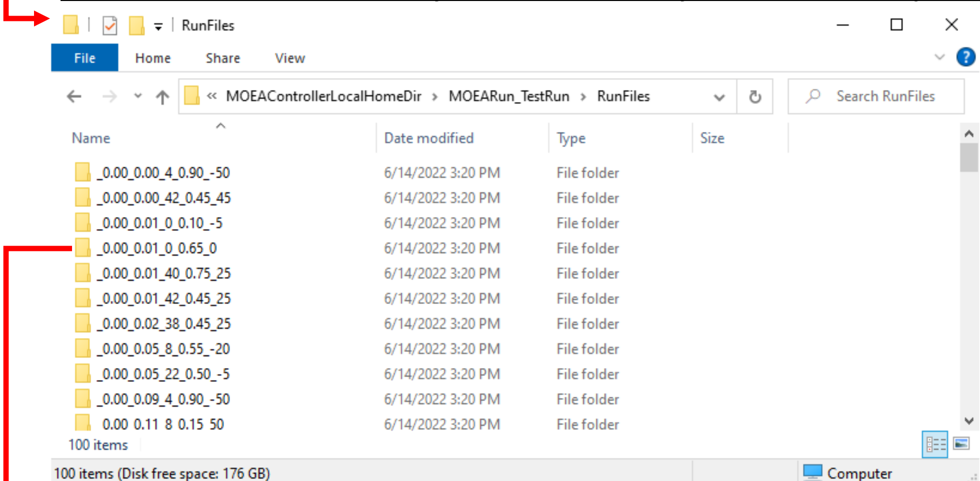
1. Import the MOEA Configuration and Preprocessing
    - Read the MOEA Analysis Configuration XML File that is generated by the Excel MOEA Controller workbook.
    - Organize and preprocess the necessary information to be passed to the MOEA algorithm and to each RiverWare© model run.
    - Perform various input checks and stop the run if anything is inconsistent or missing.
    - For decision variables, set up the sampling, crossover, and mutation properties depending on the desired step size and variable type (continuous, discrete, or binary).
    - Set up the base directory structure for the MOEA analysis to be run and copy in base files as necessary. It is important to note that all files necessary for the specific MOEA analysis run (e.g., the base RiverWare© model file) are copied into the base MOEA analysis directory, and are then subsequently used from that location alone during the run. This is important to maintain file and analysis fidelity and prevent unintentional accessing and modifications of these files during an MOEA analysis run.
2. Define the Problem to be Optimized, Configure and Execute each RiverWare© run. This is the problem class module that is independently run for each function evaluation during an MOEA analysis run. This is the part of the MOEA wrapper that can be run in parallel using multiprocessing, which allows for multiple simultaneous function evaluations.
    - For each function evaluation, the MOEA-generated set of decision variables is passed into this module from the MOEA algorithm (parts 3 and 4 below) as it is executed.
    - Set up the unique RunID and the directory structure for that specific RiverWare run based on the decision variable values.
    - Determine if this exact same RiverWare© run already exists, and if so, reuse the previous output rather than rerunning a duplicated run, for efficiency purposes. This is how duplicate runs are handled within the MOEA Wrapper (the creation of them by the MOEA algorithm is not able to be avoided).
    - Create the input and output DMI control files, input value flat files (within the input folder), and RCL execution file for the specific RiverWare© run.
    - Execute the "GetStorageTargets" python script to generate the flood storage requirements input for the specific RiverWare© run.
    - Execute RiverWare© run via the RCL script, and wait for run to complete.
    - Read output files and organize for transfer back to MOEA algorithm.
3. Define MOEA Algorithm and Sampling, Mutation, and Crossover Parameters

    - This part of the MOEA Wrapper module imports (from the pymoo library) and defines the NSGA-II algorithm as the optimization algorithm to be used for the analysis. The algorithm is initialized with the population size and the number of function evaluations per generation (i.e., the number of parallel runs that are to be made simultaneously during each generation).
    - This part also is where the sampling, mutation, and crossover parameters are imported from the pymoo library and defined based on variable type (continuous, discrete, or binary).
4. Execute the MOEA Analysis
    - This part of the MOEA Wrapper module is where the MOEA analysis is actually started. This is accomplished with the call of the "minimize" function, to which the problem (defined in part 2 above) and the algorithm (part 3 above) are arguments.
    - Also defined in this part is the termination criteria of the MOEA analysis, which in this case is based on a user-defined maximum number of evaluations.
5. Postprocessing of Results
    - This part of the MOEA Wrapper is only reached after the MOEA analysis has completed. Here, the results are unpacked from the output format provided by the "minimize" function and are reorganized to support post-processing and output to csv format.
    - Other results are read in from the individual model run files and combined with the non-dominated solution set.
    - The generation of various output graphics can also be added to this section as desired. However, for the WMOP study MOEA analysis, separate output viewer tools are being developed using Excel and PowerBI.

MOEA Analysis Base Directory (for an example MOEA Analysis named "TestRun")



Within the "RunFiles" Directory – Each subdirectory here is a RunID representing a unique model run



Within a "RunID" Directory – Containing the files associated with the single model run

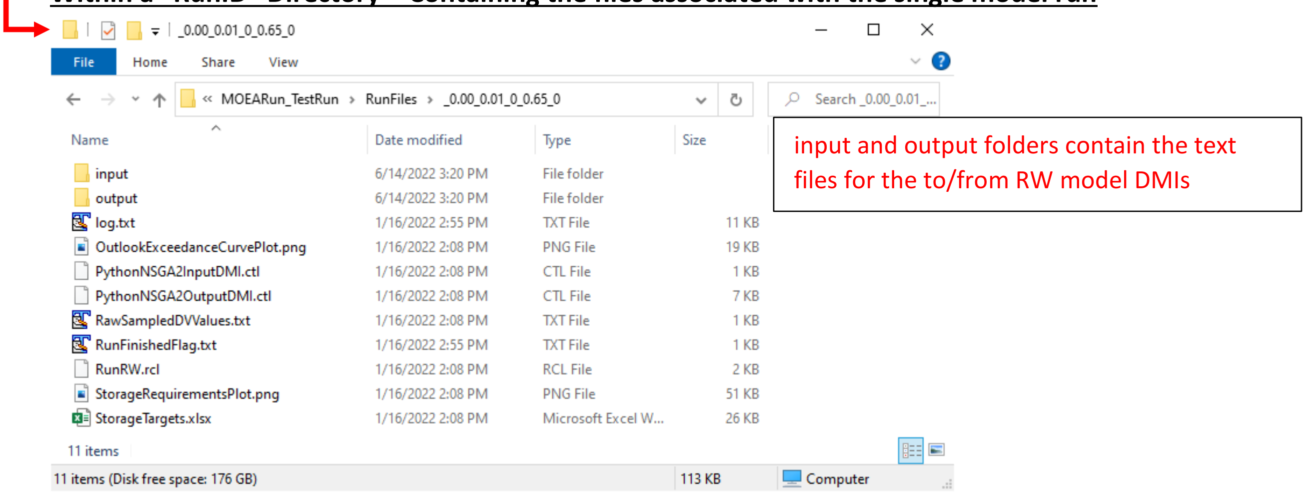


Figure 4-1: Notated screenshots of an example base MOEA Analysis directory and select subdirectories.

4.4 EXCEL MOEA CONTROLLER/CONFIGURATION WORKBOOK

To facilitate the configuration and running of MOEA analyses with the MOEA Wrapper and TROA Planning Model, an Excel MOEA controller/configuration workbook was developed. This controller workbook automates the configuration and generation of various input files required to run an MOEA analysis, and thus greatly improves efficiency. It is within this workbook that the RiverWare© model file, RiverWare© version, and the various working directories and environment variables are defined, as well as MOEA analysis parameters such as the number of function evaluations and number of parallel runs. This workbook is also where the specific decision variables, objectives, and constraints that are to be used for the analysis are defined (it should be noted that these must already be developed and integrated within the TROA Planning Model and associated supporting files).

The workbook uses Excel VBA macros to generate various necessary files, including the template input and output DMI control files, the template RCL script, and an XML configuration file. Subsequently, when the python MOEA Wrapper is run, it imports these files and uses them and the information they contain as necessary throughout the MOEA analysis run.

A screenshot of the main configuration sheet of the Excel MOEA controller/configuration workbook is shown below in Figure 4-2.

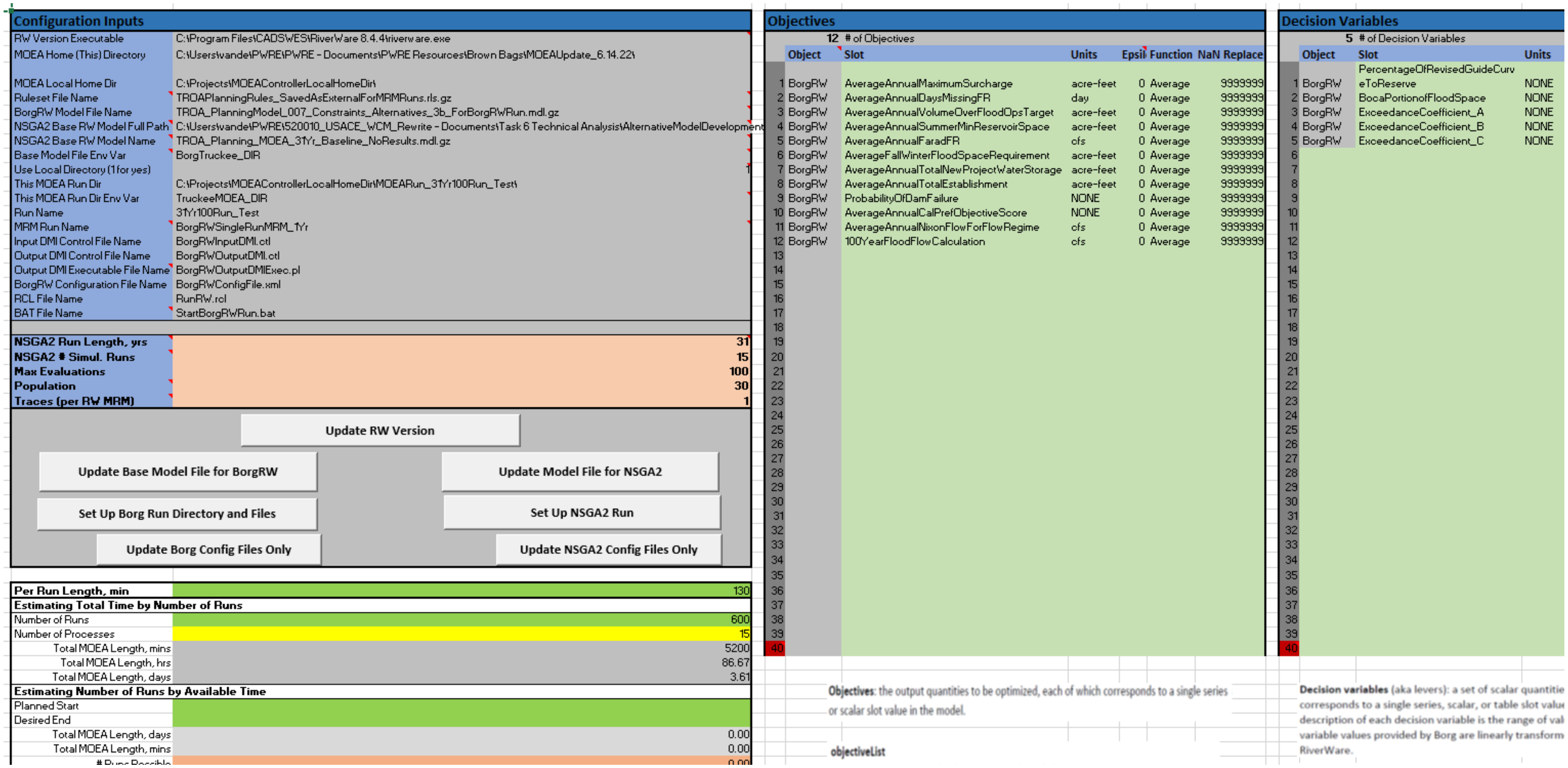


Figure 4-2: Screenshot of Excel MOEA Controller/Configuration Workbook.

4.5 HARDWARE RESOURCES UTILIZED

Precision Water Resource's powerful desktop computer ("Deluge") is the intended machine for running the MOEA analyses for the WMOP study. MOEA wrapper and associated script development and various test runs were made on Deluge as well as various PWRE staff laptops. A single 31-year TROA Planning Model run uses ~15GB of RAM, and therefore RAM is the limiting factor on all relevant machines.

Table 4-1, below, summarizes the main computers used for MOEA development and analysis for the WMOP study. For comparison purposes, the TROA Watermaster "Superconductor" desktop computer is also included in the below table, although no MOEA analysis or development was conducted on that machine. Also shown in the table are the run times for the full 31-year TROA Planning Model run and the number of simultaneous runs possible on each machine, and the estimated total run time to complete a 2,000 iteration MOEA analysis. Note that 2,000 runs are used here for estimation purposes, however the actual MOEA analysis may vary based on preliminary results and performance. It is clear from these estimates that Precision's Deluge machine is a critical component of being able to complete a satisfactory MOEA analysis in a feasible timeframe.

Table 4-1. Hardware Utilized and Estimated MOEA Analysis Time Requirements.

Machine Name	Make and Model	Processor and Speed	RAM	Time to Complete Single TROA Planning Model Run	Simultaneous TROA Planning Model Runs Possible	Time to Complete 2000 Run MOEA Analysis
PWRE "Deluge"	Custom Build	AMD Ryzen Threadripper 3960X 24-Core Processor Base 3.79 GHz Max 4.50 GHz	256 GB	130 mins	15	12 days
TROA Watermaster "Superconductor"	Alienware	Intel Core i9-7900X 10-Core Base 3.30 GHz Max 4.30 GHz	64 GB	150 min	4	52 days
PWRE Staff Laptop	Dell Precision 5540	Intel Core i9-9880H 8-Core Base 2.30 GHz Max 4.80 GHz	32 GB	140 min	2	97 days

5 REFERENCES

Alexander, E., Kasprzyk, J., Zagona, E., Prairie, J., Jerla, C., & Butler, A. 2018. Using Multi-Objective Optimization to Explore Reservoir Operating Policies in the Colorado River Basin. 2018 RiverWare User Group Meeting presentation. Boulder, CO.

Blank, J., & Deb, K. 2020. pymoo: Multi-Objective Optimization in Python, in IEEE Access, vol. 8, pp. 89497-89509, 2020, doi: 10.1109/ACCESS.2020.2990567. www.pymoo.org. Version 0.5.0.

Center for Advanced Decision Support for Water and Environmental Systems (CADSWES). 2019. Borg-RiverWare Wrapper Presentation. 2019 RiverWare User Group Meetings. Boulder, CO.

Center for Advanced Decision Support for Water and Environmental Systems (CADSWES). 2021. Borg-RiverWare Version 8.2, User Guide.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2000). A fast elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197.

Hadka, D., & Reed, P. (2012). Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. Evolutionary Computation, doi: 10.1162/EVCO a 00053.

Hadka, D., & Reed, P. (2013). Borg: An auto-adaptive many-objective evolutionary computing framework. Evolutionary Computation, 21(2), 231–259. https://doi.org/10.1162/EVCO_a_00075

Reed, P. M., Hadka, D., Herman, J. D., Kasprzyk, J. R., & Kollat, J. B. 2013. Evolutionary multiobjective optimization in water resources: The past, present, and future. Advances in Water Resources, 51, 438–456. https://doi.org/10.1016/j.advwatres.2012.01.005

San Juan Recovery Implementation Program (SJRIP). 2022. FY2021 Optimization and Sensitivity Analysis Modeling Exercises, Technical Report.

Smith, R., Kasprzyk, J., & Zagona, E. 2016. Many-objective analysis to optimize pumping and releases in multireservoir water supply network. Journal of Water Resources Planning and Management, 142(2), 04015049. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000576

U.S. Army Corps of Engineers (USACE). 1985. Truckee River Basin Reservoirs, Truckee River, Nevada and California. U.S. Army Corps of Engineers, Sacramento District.

Watson, A., et al. 2018. First Steps Towards a Multi-Objective Reservoir Optimization. RiverWare User Group Meeting Presentation, February 1, 2018.

Zagona, E.A., et al. 2001. RiverWare: A generalized tool for complex reservoir system modeling. JAWRA Journal of the American Water Resources Association 37.4: 939 913-929. https://www.riverware.org.