30 Years' Worth of Optimization in 30 Minutes

By CDR (Ret.), Sela Meyouhas, Former Project Manager, Israel MOD

06-10-15

Abstract

Modern defense systems are highly complex, such that the ever-improving advancements and capabilities in computing power has created a challenge in abstracting real-time analysis from the seemingly infinite number of system computations performed during simulation and real scenarios. An essential requirement for the development of complex systems are therefore testing algorithms which analyze data and present results in a relevant time frame.

Research on optimization algorithms has shown that incorporating a combination of different optimization algorithms can handle complex target functions with real time results, potentially transforming computation time from years to minutes. In this paper, optimization algorithms for handling complex target functions are presented which make possible the real-time assessment and optimization of multi-layer battlefield systems.

Background

Computing infrastructure has increased dramatically over the past 20 years. The exponential improvement in computing power, bandwidth and general hardware advancement has enabled the development of ever more complex defense systems, often called the system of systems. Transforming more capabilities from the classical analogue design to their digital form has not only improved total system capabilities, but also enables the engineer to more easily introduce algorithm improvements. However, this apparent advantage can easily become a liability when considering the time and effort required to prove the design, regression and actual field testing.

In a recent article published by Dr. Gil Zwirn (Omnisys Ltd.), it was shown how a system designer can significantly reduce effort during the design phase, and prove the design by using an engineering simulation system. As an example, simulation enables the engineering team to conduct quick operational research to evaluate the expected added value of the new design over the existing design, and make a decision based on quantitative analysis rather than theoretical approach. The simulation can further be used to analyze the sensitivity of the new design or any algorithm to its respective parameters. This allows the QA team to focus their testing around the predefined working points where the simulation reaches its limit (i.e., fidelity).

In other words, the digital era produces parameter based systems where each parameter controls the result and outcome of a particular section of the system, and is most often used to tune and calibrate the system to its optimum performance by trial and error. This is usually performed during validation and verification phases, and supported by a team of engineers who have deep understanding of the system design and behavior. Once the complex system is delivered to the customer, the system flexibility, in the form of variations to parameter sets, remains largely theoretical as the majority of operators lack the knowledge needed to predict the outcome of any parameter change of the system. The more complex the system, the more difficult the task becomes justifying the need of an additional system in the form of a decision support sub-system that enables a person to make decisions regarding system adaptation based on quantitative analysis performed in a relevant time frame.

Furthermore, as the motivation of the designers and system operators is to achieve continuous specification performance and adapt the system to support various battlefield conditions, the decision support sub-system must include two additional layers consisting of analysis and optimization engines. The role of the analysis engine is to identify gaps between predicted and actual performance, while the latter role is to find the optimal parameter sets which will enable the assessment of the necessary parameter alterations needed to achieve desired performance within the limit of the complex system's design.

A possible architecture of such a system can be seen in the Figure 1.

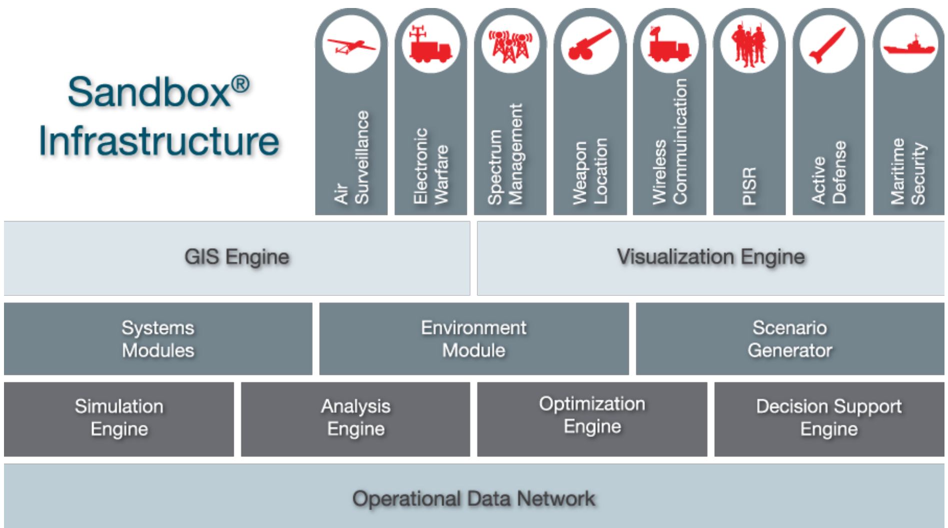

Figure 1: Possible architecture of an online decision support system

Additional difficulties are introduced when considering an online DSS (Decision Support System) that is required to provide the optimal parameter sets to achieve overall system performance. The main difficulty can be described as– "to provide recommendation results in a relevant time frame", focusing the effort on the optimization engine that must handle complex target functions and a large number of DOF's (Degrees of Freedom).

The following sections provide some insights into the challenges facing the designers and engineers in the development of the optimization engine or sub-system.

The Scope of Optimization

The purpose of most optimization algorithms is to identify the value of system parameters that yield the "best" performance for that system, based on a specified function that gives a quantitative measure of performance. From a mathematical perspective, the algorithm tries to find the maximum (or minimum) of a target function F(x1, x2, x3, ...), where F is the measure of performance, and x1, x2, x3, ... are the system parameters that require optimization. The ensemble of all possible optimization parameter sets can be described as the "Configuration Space" of interest.

An optimization algorithm does not concern itself with actually computing the target function for specific argument sets, as such computation may be carried out externally. It may be considered as an iterative procedure to determine which parameter sets serve as the best candidates for the next target function evaluation, based on the target function values computed so far.

There are two main requirements that an optimization algorithm should meet:

- Converge to the best (or at least, "good enough") optimization solution;
- Achieve this task with the minimal (or, at least, "low enough") computation time as determined largely by the number of iterations needed for convergence.

Why is Optimization Difficult?

Optimization algorithms are straight forward and relatively simple to implement as long as the target function is smooth, the criteria is easily translated to the configuration space and the number of local optima is relatively small. The computation time remains the important parameter which should be handled properly. However, optimization algorithms that are required for handling both complex systems and multiple systems can expect irregularities in place of smooth behavior of a theoretical target function.

The main obstacles for the optimization algorithm in the complex systems field are:

Common target function irregularities:

- Multiple local optima, sometimes, an extremely large number
- Non-smooth, and even non-continuous behavior
- Complicated constraints

Large number of Degrees-Of-Freedom (DOFs):

- The computation time for an exhaustive search in configuration space depends exponentially on the number of DOFs

High variety of optimization problems:

- Each problem may require a different approach for best performance

The key factors that control the computation time are detailed above in paragraph 1 and 2. In order to simply describe the optimization challenge, one may consider an optimization problem consisting of 3 different systems, where each system has 3 different parameters, and each parameter can be set to one of 10 different values. The number of combinations is 10^9.

Considering a single second required to evaluate the target function for each combination, the total time required for a single computing platform is more than 30 years! This time frame is totally non relevant for our optimization goal. So, the question that one can ask is how to turn 30 years to 30 minutes? Can it be done?

For simplicity, the top main requirements of the Multi-timescale optimization, and the overall processing time should be several seconds to several hours:

- Up to several hours for preliminary mission planning;
- Several seconds or minutes for online planning updates, (for example: Transmission frequency reallocation in response to jamming or mutual-interference; or radar search-fence parameters may be updated after several minutes).

Optimization challenges and basic solution schemes

Classic optimization algorithms are often insufficient, in the sense that they do not meet the top requirements listed above due to various reasons. Each one of the reasons introduces a level of complexity that affects the time needed for optimal solution. Furthermore, many classic algorithms do not easily support the implementation of important system engineering considerations, such as, a-priori information and constraints.

For example, forbidden or preferred regions in the configuration space (see Figure 2).

In addition, optimization algorithms are not robust enough to support convergence around stable solutions, and may result in unwanted solutions surrounding irregularities of the target function (see Figure 3). Another challenge can be thought of as a parameter set providing good system-wide performance, which requires minimal change in the present DOF setting (incremental optimization).

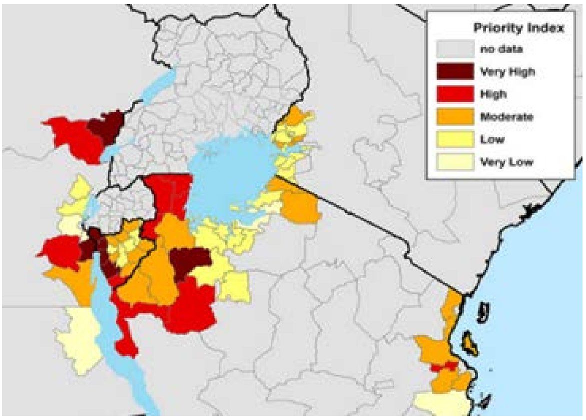

Figure 2: Example of a Region Priority Index

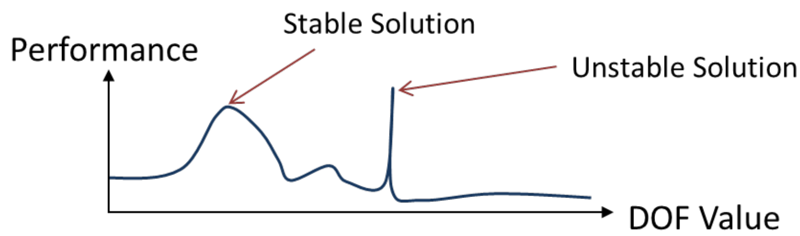

Figure 3: Stable vs. Unstable solution

A further optimization challenge includes the robustness of convergence for high dimensional problems. For instance, high dimensional steepest-descent analysis (e.g., Newton-Raphson) is sensitive to local system performance variations over the configuration space. In addition, many algorithms require that intermediate results should be stored internally to allow post-run sensitivity analysis. An example with two DOFs is shown in Figure 4. The color describes system performance and the x's mark the examined configurations. The resulting databases may be very large, taking the challenge to the realm of big-data.

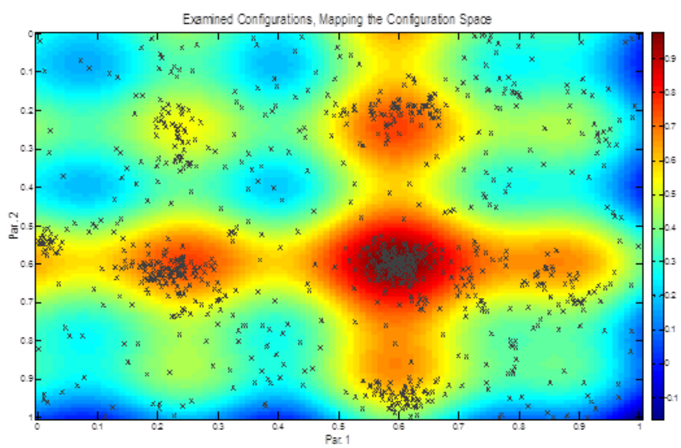

Figure 4: 2 DOF example of an optimization problem

Overview of basic optimization schemes

Grid search

The basic and most trivial scheme is a systematic scan of "all" the possible valid configurations. This approach is suitable only for very simple scenarios that have minimal degrees of freedom. The main advantage of this approach is that it may support prioritization, so that the more promising (as estimated a-priori) configurations would be examined first, and thus result in a reduction of computation time. On the other hand, in most cases the criterion is not met due to the need of evaluating all (or near all) combinations leading to non-relevant solutions.

Multinary Search

The multinary search is an iterative refinement of a search grid representing the parameter sets. In this approach, the target-function is evaluated for all the points on a coarse grid defined over the entire configuration space. The coarse grid points that produced the best target-function values (i.e., top n grids, where n<3) are selected. The next step is to evaluate the target function over a refined grid with the selected **n** coarse grids. The process is repeated for finer and finer grids until the result is achieved.

Figure 5 illustrate this scheme for 2 DOF's.

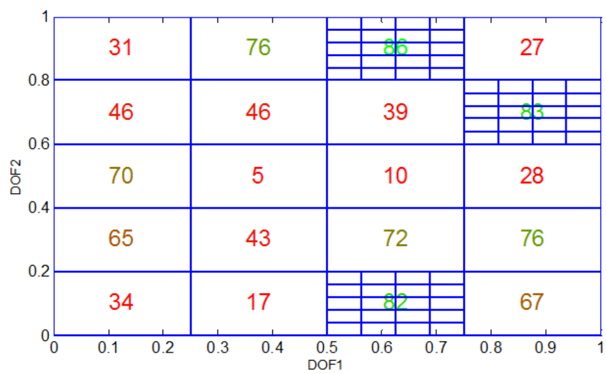

Figure 5: Multinary search scheme for 2 DOF's

Stochastic search ("Monte-Carlo")

The Stochastic search scheme represents an iterative refinement of a random set of parameter sets. This approach consists of the same idea as described in the "Multinary Search", where at each iteration the parameter sets are selected randomly in the relevant region of the configuration space. This method is a feasible approach even for a relatively high number of DOFs. However, while the number of multinary search points depends exponentially on the dimensionality of the problem, the number of required stochastic search points depends typically on the polynomial order on the dimensionality.

There are other optimization algorithms that provide an even better approach, and with some adaptation, can tailored to the optimization challenge. The description of those algorithms like Newton-Raphson, Simulated Annealing and Genetic algorithms, can be found in the open literature. However, none of them in their basic scheme is suitable for the challenge. The prime conclusion is to create a combined scheme that uses the advantages of each one. It has been shown that this approach significantly reduces the required computation time and may meet the optimization criterion. But even when meeting this criterion, another challenge should be considered – simulation evaluation time for each parameter set. Since this process is performed for each evaluation, a reduction of evaluation time of each process contributes significantly to the overall time required for the optimization process. The current and most advanced way to overcome this Independent challenge is by using multi-fidelity simulation engineering models where each one is used according to the required statistical error and computation time criteria.

Summary

The modern defense systems used by the world's premier military powers are highly complex with a parameter based performance. As technology improves, computational systems are continually being pressed to assess and optimize greater amounts of data variables. However, existing algorithm sets are insufficient to perform the level of data analysis in a time frame to produce results in relevant time frames. In particular, simulation models which assess each parameter set is a long and drawn out process which is a burden on the enhancement of the system.

To obtain optimal performance of combat systems, operators and commanders must have a DSS (Decision Support System), consisting of simulation, analysis and optimization engines to support commanders with quantitative data for better decision making.

The optimization of algorithms using a combined approach has been shown to reduce computation time while meeting the required criteria. The efficiency and precision of these algorithms are demonstrated on real and synthetic datasets, where results, which could theoretically require days or weeks to achieve, are obtained in a fraction of the time.

This design approach will enable the fine-tuning of system performance and quick adaption to new scenarios and changes in the environment, yielding continuous operational optimal performance.