Graphical Deterministic Equivalent Algorithm for Robust Process Optimization

Albertus Fuad,a,\* Edwin Zondervan,a Meik Franke,a

aSustainable Process Technology, Department of Chemical Engineering, Faculty of Science and Technology, University of Twente, Drienerlolaan 5, Enschede 7522NB, The Netherlands

albertus.fuad@utwente.nl

Abstract

The uncertainty of process parameters is always a challenge during the design of chemical processes. There are several ways to consider uncertainty during process optimization, for example: stochastic programming, chance-constrained programming, and robust optimization (Li and Grossman, 2021). Robust optimization, in particular, yields the highest risk avoidance, i.e. to plan for the worst-case scenario (Li and Grossman, 2021).

In this contribution, a graphical method is proposed to estimate the robust optimum of an optimization problem, given a bounded uncertainty set. This method is based on the Monte Carlo method to find a combination of uncertain parameters that would give the robust optimum. With the robust uncertain parameter as the input, the optimization problem can be solved deterministically to obtain the robust optimum.

To demonstrate the method, a reactor-separator system with recycle, where the uncertain parameters follow Gaussian distribution is used. The proposed method was benchmarked and compared against PyROS solver that is based on the Generalized Robust Cutting Set algorithm (Isenberg et al., 2021).

The results show that compared to the more robust PyROS solver, the graphical method with 5000 samples gives a 3% difference in the objective value function with 33% computation time requirement. The speed can be increased further by utilizing parallelization during the feasibility checks of each sample.

**Keywords**: Graphical algorithm, Robust Optimization, Uncertainty.

* 1. Introduction

Optimal design of chemical processes is important in the chemical industry for safe and economic operation. One of the main challenges in process design is the uncertainty of the process parameters. Uncertainty in process parameters can lead to overdesign requirements that can be costly or even lead to infeasible designs. One of the possible solutions is robust optimization. Robust optimization is an optimization method that considers the worst-case, i.e., risk-averse, scenario given a bounded set of possible scenarios (Li and Grossman, 2021). Due to the risk-averse nature of robust optimization, the solution can have conservativism issue (Ning and You, 2018), but this conservative result gives a baseline for the possible outcome, which can also be beneficial to plan for the worst outcome. Wiebe and Misener (2021) stated that the large number of methods for robust optimization and the relatively steep background knowledge requirements inhibits the application of robust optimization. This paper proposes a graphical robust optimization method based on Monte Carlo simulations. The method can be used to graphically obtain a combination of uncertain parameters that would give the robust optimum.

* 1. Methodology

The proposed method can be used find the robust optimum of an optimization program under uncertainty, with two uncertain parameters. The optimization problem is constructed as a minimization program:

|  |  |
| --- | --- |
|  | (1) |

Where f is the objective function, g are inequality constraints including variable boundaries, and h are equality constraints. The uncertain parameters are denoted as q, decision variables as x, and state variables as y. The state variables y are outside the degree of freedom and solved such that the equality constraints holds.

This method solves the uncertainty by finding a realization of the uncertain parameters, denoted , that will give the robust optimum when the problem is solved deterministically with as the input. In the first step, the optimization problem is solved deterministically at the nominal value of the uncertain parameters. The feasibility optimal argument of the deterministic solution is checked against random scenarios of uncertain parameters.

A feasibility boundary, which separates feasible and infeasible samples, is then drawn for each inequality constraint based on a linear classifier:

|  |  |
| --- | --- |
|  | (2) |

Where µ1 and µ0 are the mean of feasible and infeasible samples, respectively, and N1 is the number of feasible samples. This equation approximates the feasibility boundary, and it approaches the true feasibility boundary as the number of samples increases. This feasibility boundary is then translated to be a tangent to the boundary of the uncertainty set, and the intersection between the infeasibility boundaries is the robust parameter. The optimization problem is finally solved with this robust parameter to obtain the robust optimum.

* 1. Case Study

To test the method an ideal reactor-separator system (Figure 1) from Isenberg et al. (2021) is used as a case study. The system consists of an ideal isothermal mixed-flow reactor with elementary reactions in liquid phase as follows:

|  |  |
| --- | --- |
|  | (3) |

There is also a separator where pure C is completely recovered as top product. The remaining A, B, D, and E go to the bottom of the splitter. One recycle stream with a recycle ratio of δ contains A and B only, and another recycle stream contains only D and E with a recycle ratio of β. The objective is to minimize the annualized operating cost of the system where the decision variables are reactor volume and recycle ratios β and δ. The uncertainty set is defined to cover 99.7% (3σ) of the normal distribution.



Figure 1. Process Flow Diagram of the Reactor-Separator System

The full optimization problem is as follows:

|  |  |
| --- | --- |
|  | (4) |

subject to:

* equality constraints (mass balances):

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |
|  | (7) |
|  | (8) |
|  | (9) |
|  | (10) |

* inequality constraints (performance requirements):

|  |  |
| --- | --- |
|  | (11) |
|  | (12) |

* and boundaries:

|  |  |
| --- | --- |
|  | (13) |
|  | (14) |

The parameter data (Isenberg et al., 2021) is presented in Table 1.

* 1. Result and Discussion
		1. Robust Optimization

The robust solution can be calculated from the optimization problem with the robust parameter as the input. Figure 2 shows how the graphical method progresses. The result of the robust optimization by this algorithm is then compared with the result from Isenberg et al. which uses the Generalized Robust Cutting Set (GRCS) method implemented in PyROS package. The result is summarized in Table 2.

Table 1. Parameters Value [1]

|  |  |
| --- | --- |
| Certain Parameter | Value |
| CA0 | 10 mol/m³ |
| FA0 | 100 mol/hour |
| c1 | 0.1 $/((m³)² year) |
| c2 | 0.125 $/mol |
| Χ | 40 mol C/hour |
| Ω | 0.4 mol D/hour |
| k1 | 0.9945 |
| k2 | 0.5047 |
| Uncertain Parameter | Mean (Hour-1) |
| k3 | 0.3866 |
| k4 | 0.3120 |

The covariance matrix between k3 and k4 is:

Table 2. Robust Optimization Result

|  |  |  |
| --- | --- | --- |
| Algorithm | GRCS (PyROs) | Graphical |
| No. of runs | 1 | 1 | 1 | 200 (averaged) |
| Solvers | IPOPT | fmincon | fmincon | fmincon |
| Parallelization | No | No | Yes8 workers | Yes4 workers |
| No. scenarios | n/a | 5000 | 5000 | 5100 |
| V (m³) | 109.5 | 107.1 | 107.1 | Not recorded |
|  | 0.77 | 0.78 | 0.78 |
|  | 0.016 | 0.016 | 0.016 |
| Total cost | 25823 | 25071 | 25071 | 24645 |
| Time | 12.1 s | 4.5 s | 1.5 s | 1.99 s |

The proposed algorithm is used both with parallel computing and serial computing during the handling of the 5000 scenarios. For both parallelized and serial computation, there is no difference on result of the optimization problem. However, the implementation of parallel computing with 8 workers reduces the computation time to one third as compared to the one without parallelization. This shows that the required time is not linearly proportional on the number of workers, because the handling of the parallelization also requires additional computational power.

* + 1. Effect of Sampling

The use of randomly generated samples can affect the outcome of the optimization over different runs. The method was tested with number of random scenarios ranging from 100 to 5100 scenarios, with a step size of 250 scenarios. In each number of scenarios, 200 different set of random scenarios are used. However, given enough samples, the optimization results converge reliably with a rather small variation between runs. The number of samples also increases the required computation time, therefore a balance between them needs to be determined. At larger sample size, the variation in computation time increases, this is possibly due to longer convergence for certain samples.



1. Sample Generation



1. Feasibility Check of Scenarios against Inequality Constraints



1. Drawing Feasibility Boundary and Tangents



1. Intersection of the Feasibility Tangents, i.e., the robust parameter

Figure 2. The Progression of the Graphical Method



Figure 3. Effect of Number of Samples to the Optimization Results

* + 1. Advantages and Further Improvements

This graphical method can estimate robust optimum in a relatively short time. The graphical nature of the method also gives an illustration on how robust optimization works, which helps learning process for optimization under uncertainty. However, it still has some limitations for further improvements. First, it can only handle two uncertain parameters and two inequality constraints that are affected by the uncertain parameters. Further research is still required for upscaling to higher dimension, but the graphical illustration would be difficult for dimension higher than 3. The second limitation is that the uncertain parameters must be linear to the objective function and constraints within the domain of the uncertainty set.

* 1. Conclusions

The proposed graphical method is a promising way to estimate a robust optimum in a short time. It also gives consistent results as long as enough samples are provided. Compared to the more robust GRCS method, the graphical method with 5000 samples gives a 3% difference in the objective value function with 33% computation time requirement, which can still be accelerated by parallelization. The algorithm also works consistently as long as enough random scenario are provided. Further research is required for upscaling to higher dimensions.

References

C. Li, I.E. Grossman, 2020, A Review of Stochastic Programming Methods for Optimization of Process Systems Under Uncertainty, Front. Chem. Eng, 2:622241.

J. Wiebe, R. Misener, 2021, ROmodel: A Python Robust Optimization Modeling Toolbox, Computer Aided Chemical Engineering, 50, 683-688.

C. Ning, F. You, 2018, Leveraging the Power of Big Data Analytics for Process Scheduling under Uncertainty using a Stochastic Robust Optimization Approach, Computer Aided Chemical Engineering, 43, 319-324.

N.M. Isenberg, P. Akula, J.C. Eslick, D. Bhattacharyya, D.C. Miller, C.E. Gounaris, 2021, A generalized cutting‐set approach for nonlinear robust optimization in process systems engineering, AIChE J., 67.