Numerical Methods Project Report

Aniket Bagade, Mit Phanase, Nachiket Soni, Naitik Dalwadi

March 31, 2025

Abstract

This paper focusses on surrogate-model based optimization to find the optimal parameters of the Llamas model that computes speed lines for given compressor data. The Llamas model used five parameters to compute the speed lines, which can be optimized using a surrogate approach. The main goal of this study is to analyze whether a surrogate model based approach can get the same results as a direct optimization method. It was found that providing initial starting points from a direct optimization approach to the surrogate models improved accuracy significantly. Finally, the aim of this study is to try and find an answer to the question of how the surrogate-model based optimization algorithms perform on the compressor data sets and whether a suroagate-model based approach is comparable to direct optimization methods.

1 Introduction

Turbochargers play a crucial role in both automotive and marine diesel engines (Song et al., 2019). The physical data point collection for a compressor is an expensive endeavour and thus, conducting extensive experiments is not feasible. This necessitates the development of a mathematical model with optimal predictive accuracy for their working cycle dynamics. The major mathematical modeling techniques of marine compressors lie under curve-fitting techniques, as they effectively capture the fundamental physical characteristics within the mathematical framework.

This study aims to analyze whether a surrogate model based approach can get the same results as a direct optimization method. Moreover, this study addresses the following questions: How can be the setup for the surrogate-model based optimization algorithms improved? And, is a surrogate-model based approach comparable to direct optimization methods?

The rest of this paper is organized as follows: Section 2 presents an overview of surrogate-based models, initial estimates derived from the prior task, the objective function, and the evaluation criteria. Then Section 3 presents results and Section 4 discusses the pros and cons of the surrogate model based approach and compares the surrogate model based approach with the direct optimization methods.

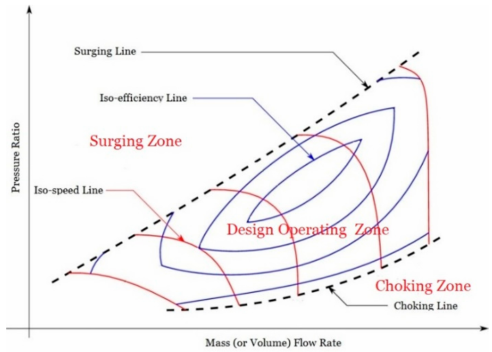


Figure 1: Compressor Performance Map

2 Methods

2.1 Characteristic Maps: Speed-Line Modeling

The performance map of a compressor usually represents its working characteristic, as shown in figure 1. The horizontal and vertical axes represent the mass (or volume) flow rate and pressure ratio, respectively. The measured data points at the same rotational speed are connected in performance map, which forms iso-speed lines. Similarly, those data points with the same isentropic efficiency are connected, which forms iso-efficiency lines (Fang et al., 2014).

The compressor performance map is divided into following three zones:

- Design Operating Zone: It is the zone, where compressor works stably and achieve high efficiency.
- Surging Zone: In this zone, the compressor does not have sufficient amount of air at high pressure ratio, which leads to reverse flow of air and premature failure of the compressor. On each iso-speed line, the curve passes through the surge point is known as **surging line.**
- Choking Zone: In this zone, though the decrease in pressure ratio, the mass flow rate will not increase further and it flows at sonic velocity. On each iso-speed line, the curve passes through the choking point is known as **choking line.**

Based on the selection of the input and output parameters, the models are classified into two following categories:

$$\dot{m} = f(\Pi, N) \tag{1}$$

$$\Pi = f(\dot{m}, N) \tag{2}$$

Where, N is the rotational speed of the compressor, $\dot{m}$ is the mass flow rate, and Π is the pressure ratio.

Dataset	Summary Statistics	Pi_tot_V	m_V
experiment_0.1	mean	3.811611	5.198377
	std	1.278679	1.892726
experiment_0.01	mean	3.805166	5.194258
	std	1.266266	1.895922
experiment_0.001	mean	3.804175	5.194126
	std	1.266433	1.894269

Table 1: Summary Statistics of the Datasets

Moreover, Llamas Ellipse Model is capable of working in both direction, which means it can predict either mass flow rate or pressure ratio based on the situation. In the design operating zone, the iso-speed line is represented by the super ellipse represents the relationship between $\dot{m}$ and Π

$$(\frac{\dot{m} - \dot{m_{zs}}}{\dot{m_{ch}} - \dot{m_{zs}}})^{CUR} + (\frac{\Pi - \Pi_{ch}}{\Pi_{zs} - \Pi_{ch}})^{CUR} = 1 \tag{3}$$

Where,

- $\dot{m_{zs}}$ is the mass flow rate at the surge point.
- $\dot{m_{ch}}$ is the mass flow rate at the choking point.
- Π_{zs} is the pressure ratio at the surge point.
- Π_{ch} is the pressure ratio at the choking point.
- CUR is the curvature of the super ellipse.

These five parameters are encapsulated in ß parameter vector, and this parameter vector is optimized to minimize the error between the predicted and actual values of the mass flow rate and pressure ratio.

2.2 Data Sets

The three different datasets were used in this study. Each dataset consists of three columns: ***Speedclass[m/s]***, ***Pi_tot_V[-]***, and ***m_V[kg/s]***, and 63 rows contain empirical data of pressure ratio and mass flow rate at 10 different compressor speeds ranging from 3750 to 8550 m/s. Moreover, each dataset does not contain any missing values. The summary statics of the datasets are shown in Table 1.

As it is observed in the Table 1, the experiment_0.1 dataset has more dispersion, followed by the experiment_0.01 and experiment_0.001 datasets. For the current study, initial results are drawn from the experiment_0.001 dataset. This choice is made so as to understand and dissect the behaviour of different optimizers with SPOT first and see if they can match the performance of a classical optimizer on a noisy dataset. This creates a controlled environment to test the configurations on and gives more room to SPOT to show its capabilities.

2.3 Surrogate Models

Surrogate model based optimization techniques are usual approaches in the simulation and optimization field. To address the requirement of effective statistical analysis of simulation and optimization algorithms, the Sequential Parameter Optimization Toolbox (SPOT) was built. This toolbox is available on GitHub as a python package. SPOT contains sophisticated methods for hyperparameter tuning based on classical regression and variance analysis methodologies. It contains tree-based models such as classification and regression tree and random forests as well as Bayesian optimization(Gaussian process models, known as Kriging). Additionally, SPOT has optimization methods work based on surrogate model.

The `spot` loop consists of the steps shown in Algorithm 1.

Algorithm 1 The Spot Loop

1: **Initialize:** Build initial design X
2: Evaluate the initial objective f: $y = f(X)$
3: Build surrogate: $S = S(X, y)$
4: Optimize on surrogate: $X_0 = \text{optimize}(S)$
5: Evaluate on real objective: $y_0 = f(X_0)$
6: Impute (infill) new points: $X = X \cup X_0$, $y = y \cup y_0$
7: Go to 3.

2.3.1 Optimization Algorithms

To analyze the effect of different optimization algorithms on the available compressor datasets, following four algorithms have been used from Scipy Library (Virtanen et al., 2020):

- Differential Evolution (Storn and Price, 1997)
- Dual Annealing (Mullen, 2014)
- DIRECT (Jones et al., 1993)

2.4 Initial Guess

By default, the spot loop implementation uses randomly generated values for the optimization of the function before it creates a surrogate model for evaluations. One way to influence the evaluations is to provide it with good starting points such that it takes less time to converge. In this work, 10 starting points are provided. These points are the final values of per speedline minimization of the datasets using L-BFGS-B (Zhu et al., 1997) optimization algorithm with Ortho error metric. Therefore, the experiments are conducted in two ways - one with only `init_size` and the other with `init_size` plus 10 starting points.

2.5 Objective Function

A residual function was used as the objective function for minimization. The idea is to get the minimum values and use them for understanding the speedline fits. The residuals are calculated as follows:

$$\text{residual} = \sum_{i=1}^{n} \left[(\Delta x_i)^2 + (\Delta y_i)^2 \right] \tag{4}$$

where n is the number of points and Δx_i and Δy_i are the differences between the predicted and actual values for the pressure ratio and mass flow, respectively.

2.6 Boundary Conditions

This study has conducted in the default bounds setting selected as the minimum and maximum values of Pi tot V and m V,which are then scaled between 0.75 and 1.25. For CUR, the bounds are set between 2 and 15.

2.7 Evaluation

To compare the results of the different configurations (optimization algorithms, metrics, and boundary conditions), the following Overall normalised Root Mean Squared Error of the predicted values of mass flow rate and pressure ratio for all speedlines.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \hat{y}_i}{\hat{y}_{\max}} \right)^2}, \tag{5}$$

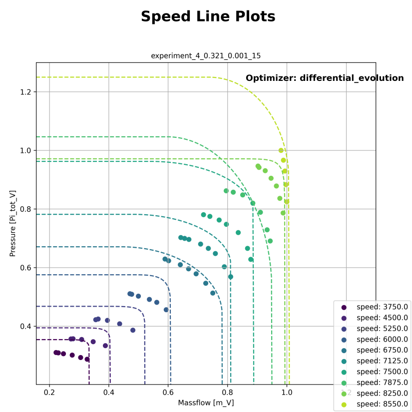
where N is the total number of points and $\hat{y}_{max}$ is the maximum predicted value.

3 Results

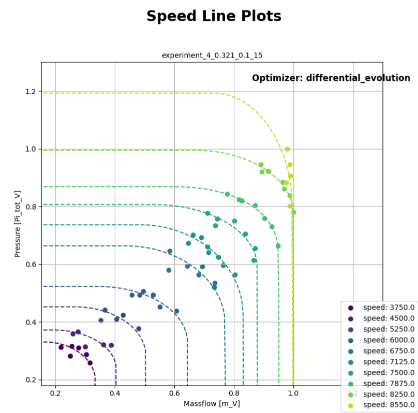
To compare the performance of the different optimizers, the `fun_evals` and `init_size` parameters of the function were initially set to 40 and 5, respectively and no startings points were provided. All three optimizers were applied to the least noisy compressor dataset and evaluated based on the **overall RMSE** metric. The overall RMSE values for different optimizers are observed in Table 2.

Optimizer	Overall RMSE (Pressure)	Overall RMSE (Mass)
Dual Annealing	0.09669000	0.08540000
Direct	0.12230931	0.11299756
Differential Evolution	0.07678185	0.07548894

Table 2: Comparison of normalised RMSE values for different optimization methods (With Least Noisy dataset and without Starting Points).



(a) `fun_eval`=40, `init_size`=5 without starting points



(b) `fun_eval`=25, `init_size`=3 with starting points

Figure 2: Speedline Plots for different configurations

Form the Table 2, it is observed that the differential evolution performs well compared to other two optimizers by achieving the lowest overall RMSE values for pressure and mass. On providing the starting points and keeping `fun_evals` and `init_size` as 25 and 3, both, dual annealing and differential evolution, give similar error but the former takes more time to converge. This can be due to the way Dual Annealing works where it has a cooling schedule that terminates based on either time or number of iterations. Hence, for further experiments, differential evolution was chosen. Moreover, the results were drastically better when starting points were provided.

Further, the experiments were shifted to the noisiest dataset. `fun_evals` and `init_size` were kept 25 and 3 respectively and starting points were provided. Table 3 shows the overall normalised RMSE values for pressure ratio and mass flow for the experiment.

Optimizer	Overall RMSE (Pressure)	Overall RMSE (Mass)
Differential Evolution	0.02599097	0.02696389

Table 3: Overall normalised RMSE values for Differential Evolution with the noisiest dataset and with Starting Points.

3.1 Visual Inspection

Figure 2 shows the effect of starting points on the speedline fit. On the left is the speedline fit for the same optimizer on the least noisy dataset without starting points. It can be observed that the speedcurves do not fit with the actual data points but do follow the curvature. It is better able to fit the higher speedlines compared to the lower ones. Figure on the right shows the fit for the noisest dataset with starting points. The speed curves fit the datapoints properly. Additionally, it took less number of function evaluations and the initial size to converge.

4 Discussion

4.1 Pros and cons of the Surrogate Model Based Approach

The surrogate model-based approach offers several advantages, particularly when real-world experimentation is constrained. Its benefits include:

- Fewer actual experiments are needed, Especially when collecting large amounts of real data is impractical or unethical.(Bartz-Beielstein, 2023)
- Surrogate models are computationally less demanding than evaluating complex functions at every iteration, speeding the optimization process.
- Contour plots derived from surrogate models facilitate the assessment of parameter importance in multidimensional optimization problems.
- In many scenarios, surrogate models can provide a faster and more cost-effective alternative to real experiments.

However, the surrogate model-based approach has several drawbacks:

- Highly dependent on the initial design; the quality and accuracy of results rely heavily on the real data used for training.
- Can be more computationally expensive and time-consuming than direct optimization methods, particularly when constructing highly accurate surrogate models.
- As approximations of real models, surrogate models may introduce errors, leading to misleading or uninterpretable results.
- Surrogate models approximate the original function which may fail to capture the complexity of the real function, resulting in suboptimal solutions.

4.2 Comparison of the Surrogate Model Based Approach with the Direct Optimization Methods

In this work, surrogate models achieved comparable performance to direct optimization methods in terms of overall root mean square error (RMSE). However, surrogate-based models required significantly more time to identify the minimum compared to direct optimizers. It should also be noted that when the starting points were provided, the results improved significantly. This difference arises because surrogate modeling focuses on exploring the entire search space to locate the global minimum, which demands greater computational effort and time for convergence. Given that both approaches produced similar outcomes, it can be inferred that the local minimum identified by direct optimization methods likely corresponds to the global minimum for the specified objective function. Moreover, considering that the problem is closely defined and the underlying physics is known, direct optimization approaches work sufficiently well.

References

Thomas Bartz-Beielstein. 2023. Hyperparameter Tuning Cookbook: A guide for scikit-learn, PyTorch, river, and spotpython. *arXiv e-prints*, Article arXiv:2307.10262 (July 2023), arXiv:2307.10262 pages. doi:10.48550/arXiv.2307.10262 arXiv:2307.10262 [cs.LG]

Xiande Fang, Weiwei Chen, Zhanru Zhou, and Yu Xu. 2014. Empirical models for efficiency and mass flow rate of centrifugal compressors. *International Journal of Refrigeration* 41 (2014), 190–199. doi:10.1016/j.ijrefrig.2014.03.005

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. 1993. Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications* 79 (1993), 157–181.

Katharine M Mullen. 2014. Continuous global optimization in R. *Journal of Statistical Software* 60 (2014), 1–45.

Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11 (1997), 341–359.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. doi:10.1038/s41592-019-0686-2

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)* 23, 4 (1997), 550–560.

A Online Resources

The software used in this study is available online at:

- `https://github.com/sequential-parameter-optimization/spotoptim`
- `https://github.com/NaitikDalwadi/Numerical-Methods-and-Optimization`.