Collaborative Multidisciplinary Design Optimization with Neural Networks

Jean de Becdelièvre, Ilan Kroo

Stanford ENGINEERING
Aeronautics & Astronautics

Abstract: Complex engineering system design involves solving large optimization problems that include several disciplines. We study Collaborative Optimization, a strategy that allows disciplines to optimize in parallel by providing sub-objectives and splitting the problem into smaller parts. In this work we obtain better sub-objectives by solving an interesting instance of binary classification, where the data for one of the classes contains the distance to the decision boundary. We propose to train a neural network with an asymmetric loss function and a regularization that encourages basic distance function properties.

Multidisciplinary Design Optimization (MDO)

$$\underset{z,x_1,\ldots,x_N}{\text{minimize}} \quad f(z)$$
$$\text{subject to} \quad c_i(z, x_i) \leq 0 \ \forall i \in \{1,\ldots N\}$$

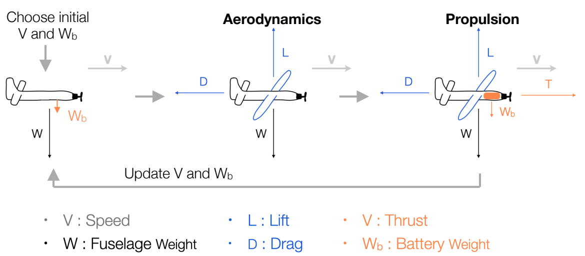
MDO minimizes a system objective, f, subject one set of constraints c_i per discipline i. **Local variables** x_i only appear in discipline i, but **shared variables** z couple disciplines together.

Simple MDO Illustration: Aircraft Marathon

Consider designing an aircraft to fly a marathon as quickly as possible. The fuselage, motor and propeller are provided. The **aero** team designs a wing, the **propulsion** team picks the size of the battery and the operating conditions of the motor.

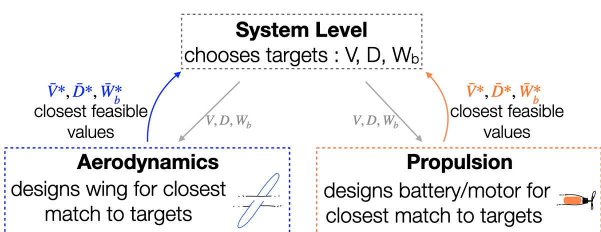
Sequential Optimization

One approach is to go through tasks sequentially. However this can lead to **convergence issues**, and prevents **parallel** operations.



Collaborative Optimization (CO)

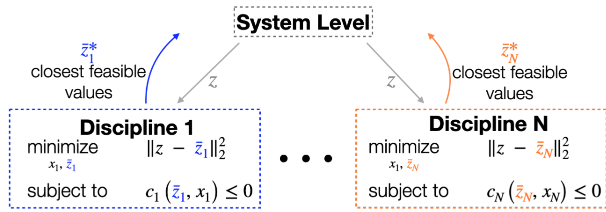
A more efficient approach is to propose target values for the shared variables and ask each team to match them. The targets are updated iteratively by an optimization algorithm.



General Collaborative Optimization Framework

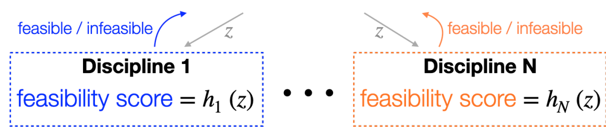
CO decomposes the problem in two levels, called successively:

1. The **system-level** chooses targets for all shared variables, z.
2. At the **subspace-level**, disciplines attempt to set their local copy of the shared variables, $\bar{z}$, as close to the targets as possible.



Choosing Better Target Variables Using Surrogate Models

The performance of CO relies on the quality of the targets proposed by the system-level. In this work we train neural networks, h_i, as the optimization progresses. They classify targets as feasible or infeasible, allowing the system-level to make better-informed target choices.



Training Data: Distance to the Feasible Set

For a target z, a subspace evaluation computes:

$$J^*(z) = \min_{\bar{z} \in \{\bar{z} \mid \exists x, c(\bar{z}, x) \leq 0\}} \|z - \bar{z}\|_2^2$$

If z is infeasible, it returns the squared distance to the feasible set J^*, and the closest feasible point z^*. Otherwise, it returns $J^* = 0$.

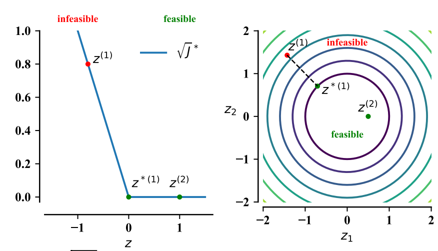


Figure: Plots of $\sqrt{J^*}$ for examples in 1-d (*left*) and 2-d (right) where only the positive halfspace or only the unit disk (respectively) is feasible.

Mixed Regression-Classification to Learn Signed Distances

Instead of exactly learning the outcome of the subspace evaluation $\sqrt{J^*}$, we learn a **signed distance function** h using the loss below:

$$l(z) = \begin{cases} |h(z) - \sqrt{J^*}| & \text{if } z \text{ is infeasible} \\ \max(h(z), 0) & \text{otherwise} \end{cases}$$

We encourage $\|\nabla h\| = 1$ using regularization at random points.

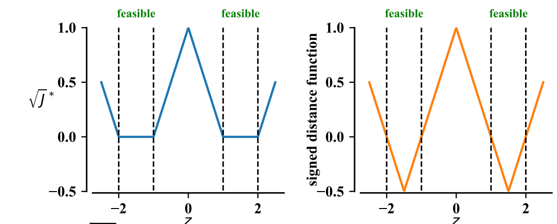


Figure: $\sqrt{J^*}$ (*left*) and signed distance function (*right*) of [-2,-1] U [1,2].

Application to the Aircraft Marathon Problem

Surrogate models of the subspaces are used to solve design problems using the following procedure:

1. Targets are sampled in the domain and each subspace is evaluated at these points.
2. A signed distance function neural network is trained to describe the set of feasible targets for each discipline.
3. The best feasible candidate point, according the networks, is found and added to the dataset. (We return to step 2 until convergence.)

	Conventional CO with SQP	Gaussian Process surrogate model of J^i	Signed Distance approach (ours)
No. of system-level iterations	74 (+/- 28)	18 (+/- 6)	6 (+/- 2)
No. of aerodynamic function evaluations	2161 (+/- 796)	267 (+/- 63)	110 (+/- 57)
No. of propulsion function evaluations	496 (+/- 209)	94 (+/- 23)	31 (+/- 8)

Baselines use Sequential Quadratic Programming (SQP) at the system-level, or rely on a Gaussian process fit of each J.

Areas for Improvement

1. Training a neural network at each iteration requires a time-consuming, automated hyper-parameter search.
2. Epistemic uncertainty is not taken into account: the neural network output is trusted even if the query is far away from training data. A better exploration-exploitation trade-off might be obtained with tools from Bayesian Optimization.