Stochastic optimization Module : Application to concrete column simulation

Atul Agrawal

October 28, 2022

Notation:

- $\boldsymbol{x}$: concrete mix parameters (also optimization variables ultimately)
- $\boldsymbol{b}$: hydration model input parameters
- $\boldsymbol{y}_c$ or $\boldsymbol{y}_c(\boldsymbol{b})$: hydration model output(s) relevant for calibration
- $\boldsymbol{y}_o$ or $\boldsymbol{y}_o(\boldsymbol{b})$: hydration model output(s) relevant for optimization (e.g. KPIs)

Suppose N data-pairs $\mathcal{D}_N$ are available which consist of $\mathcal{D}_N = \{\hat{\boldsymbol{x}}^{(i)}, \hat{\boldsymbol{y}}_c^{(i)}\}_{i=1}^N$. We would like to use those to infer the corresponding $\boldsymbol{b}^{(i)}$ but more importantly the relation between $\boldsymbol{x}$ and $\boldsymbol{b}$ (performed in the calibration module) which would be of relevance for downstream, optimization tasks.

1 Optimization

The task of optimization can be summarized as finding the optimum concrete mix parameters $\boldsymbol{x} \in \mathbb{R}$ (e.g, cement type ratio) in order to minimize certain (stochastic) objective such that a stochastic inequality constraints are satisfied. Consider the following for the current formulation:

$$\min_{\boldsymbol{x}} \mathcal{O}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}|\boldsymbol{x},\boldsymbol{\varphi}^*)}[y_o(\boldsymbol{b})] \tag{1}$$

$$\text{s.t} \quad \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}|\boldsymbol{x},\boldsymbol{\varphi}^*)}[y_{c_i}(\boldsymbol{b})] \leq \alpha_i, \quad i = 1, \cdots, m \tag{2}$$

Here, $\boldsymbol{\varphi}^*$ denotes the learnt parameters between $\boldsymbol{b}$ and $\boldsymbol{x}$, y_o denotes the solver output corresponding to the objective and y_{c_i} the solver output corresponding to the i^{th} constraint. The problem can be approached with penalty-based methods [1] [2] by defining the following objective:

[1] https://web.stanford.edu/class/ee364a/lectures/stoch_prog.pdf

$$\min_{\boldsymbol{x}} \mathcal{O}(\boldsymbol{x}) + \sum_{i=1}^{m} c_i \mathcal{C}_i(\boldsymbol{x}) \tag{3}$$

where $c_i > 0$ are penalty parameters for violating constraints and $\mathcal{C}_i(\boldsymbol{x}) = \max(\mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}|\boldsymbol{x},\boldsymbol{\varphi}^*)}[y_{c_i}(\boldsymbol{b})] - \alpha_i, 0)$. Alternatively, the problem can also be approached with the Augmented Lagrangian Method [2]

To test the scheme, a concrete column simulation with the hydration model is formulated [] (@Erik can add the description if need be) for which the objective (Key performance Indicator (KPI)) is the critical time (t_c) at which the max yield is reached and there is a single constraint pertaining to the expected max. temperature (T). Following the notations above, we have:

$$\min_{\boldsymbol{x}} \mathcal{O}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}|\boldsymbol{x},\boldsymbol{\varphi}^*)}[t_c(\boldsymbol{b})] \tag{4}$$

$$\text{s.t} \quad \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}|\boldsymbol{x},\boldsymbol{\varphi}^*)}[T(\boldsymbol{b})] \leq \alpha \tag{5}$$

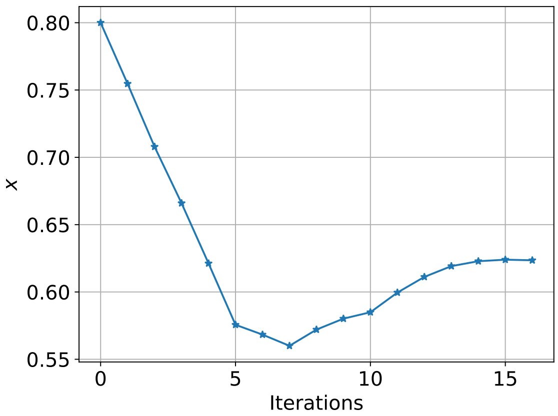
The problem is solved using the penalty-based objective described above and score function gradient estimators [1] are used to compute the expectation gradients. The implementation is performed in PyTorch library. Some initial results are depicted in Fig. 1 for $\alpha = 68^o$. As can be seen from the Fig. 1a, the constraints started to be violated close to $\boldsymbol{x} \sim 0.6$ which led to negative gradients, which reversed the descend of the optimization parameter (by observation $\mathbb{E}[T] \propto 1/\boldsymbol{x}$ and $\mathbb{E}[t_c] \propto \boldsymbol{x}$) and stabilized it to a value which also satisfies the constraints.

It is crucial to mention that computational cost of the forward solver can be a limiting factor in the optimization process. Efficient *offline* surrogates can be developed for that. It can serve two purposes: i) Alleviate the computational bottleneck, ii) Potentially reduce the estimated variance of the gradients since the reparametrization trick can be used when the surrogate is differentiable.

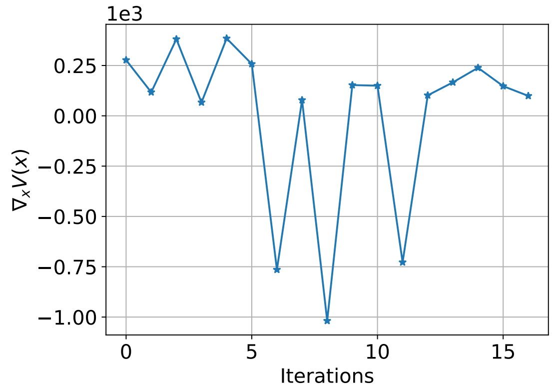
The workflow of the stochastic optimization process is summarized in the Fig. 2. The stochastic optimization of the concrete column simulation just serves as a proof of concept and in theory the workflow should not be altered for change in the physics based model, objective function, constraints function etc. After the relationship between the mix parameters $\boldsymbol{x}$ and the latents (solver inputs $\boldsymbol{b}$) is inferred in the calibration module (if an analytic relation is not existing), the following *links* are necessary to perform the optimization:

- A physics based model $\boldsymbol{y}(\cdot)$ relating the inputs $\boldsymbol{b}$ and the outputs y_o, y_{c_i}.
- If possible, a surrogate of the model linking the inputs and the outputs.
- Clearly defined objective and the constraints.

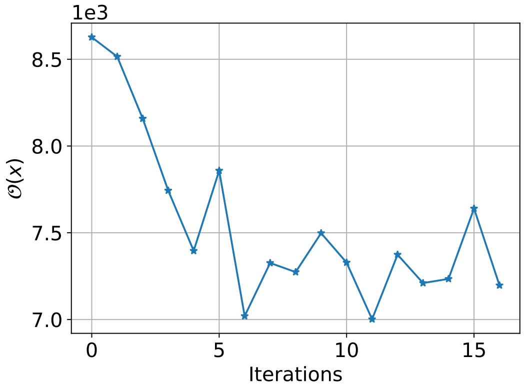
[2] https://www.him.uni-bonn.de/fileadmin/him/Section6_HIM_v1.pdf



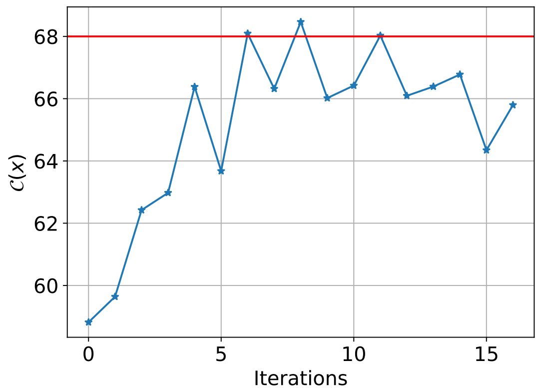
(a) Evolution of the Optimization variable x



(b) Monte Carlo estimated gradients



(c) Evolution of the objective $\mathcal{O}(\boldsymbol{x})$



(d) Evolution of the temp. constraint $\mathcal{C}(\boldsymbol{x})$

Figure 1: Optimization under uncertainty for concrete column with hydration model

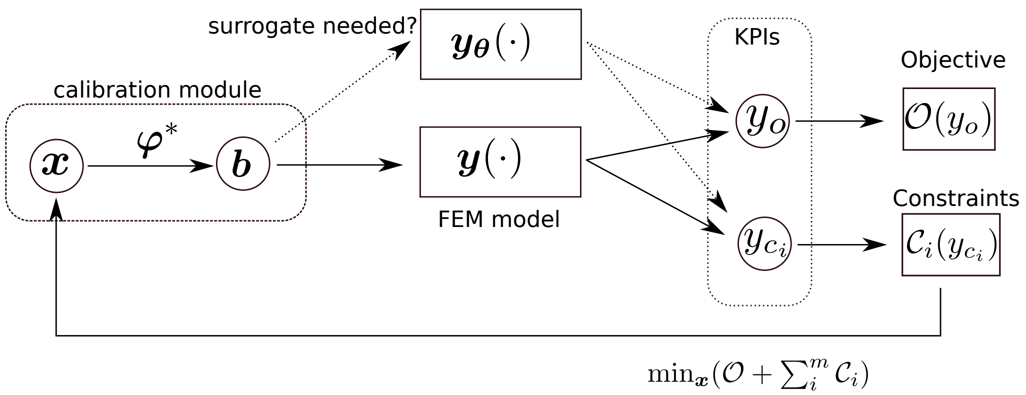


Figure 2: *Stochastic optimisation workflow* : Circular nodes represents the stochastic nodes and the square represents the deterministic nodes.

References

[1] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient Estimation Using Stochastic Computation Graphs, January 2016. arXiv:1506.05254 [cs].

[2] I.-J. Wang and J.C. Spall. Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, pages 3808–3813, Maui, HI, USA, 2003. IEEE.