Ensuring the Quality of Optimization Solutions in Data Generated Optimization Models

Segev Wasserkrug[1] , **Orit Davidovich**[1] , **Evegeny Shindin**[1] , **Dharmashankar Subramanian**[2] , **Parikshit Ram**[2] , **Pavankumar Murali**[2] , **Dzung Phan**[2] , **Nianjun Zhou**[2] and **Lam M. Nguyen**[2]
[1]IBM Research, IBM Haifa Research Lab
[2]IBM Research, Yorktown Heights
segevw@il.ibm.com, Orit.Davidovich@ibm.com, EVGENSH@il.ibm.com, dharmash@us.ibm.com, Parikshit.Ram@ibm.com, pavanm@us.ibm.com, phandu@us.ibm.com, jzhou@us.ibm.com, LamNguyen.MLTD@ibm.com

Abstract

Mathematical optimization models can improve decision making in a wide variety of domains, both industrial and societal. However, creating an optimization model requires rare optimization expertise and a significant amount of time, thereby limiting the widespread use of this technology. One way to overcome this is to enable data driven generation of complete optimization models from historical data.

A major challenge posed by such data-driven generation is that the uncertainties and inaccuracies resulting from the model generation must be explicitly accounted for to ensure the quality of the solutions ultimately produced by the generated optimization model. Moreover, the historical data used for such model generation must also contain decision related data, which has very different characteristics from the input typically used for machine learning models, further compounding the difficulty of accounting for the uncertainties.

In this work, it is our goal to bring to light this important topic of end-to-end data driven optimization model generation and encourage additional research in this area. Our contributions here are therefore to formally define the problem and outline several promising approaches for addressing it. We also describe areas for future work, which gives some indication of the breadth and depth of topics to be addressed. We hope this will also motivate others to carry out research in the broader field of enabling widespread creation of optimization models by people who are not optimization experts.

1 Introduction and Motivation

Mathematical optimization can provide improved decision support to a variety of real world problems, thereby providing significant value to almost any domain (supply chain management, cloud computing operations, healthcare, environmental impact reduction,...). Creating an optimization model requires both modeling the *constraints* which govern the system, as well as the *objective function* to be optimized. Currently, creating a mathematical optimization model requires optimization modeling expertise, which is quite rare, as well as significant time (typically on the order of months). This severely limits the actual application of mathematical optimization and the benefits it currently provides. Therefore, in order to enable mathematical optimization to realize its true potential in terms of business benefits, there is a need to enable none optimization experts to create such models in a much shorter amount of time. A possible approach to address this issue is to enable professionals such as data scientists without extensive optimization background, to automatically derive the optimization model, i.e., the objectives and constraints, from a combination of historical data and easily specifiable problem knowledge. To enable such model generation requires techniques and algorithms that are able to transform the data and knowledge into an optimization specification as described in Figure 1 (examples of some relevant techniques appear in Section 3 which describes related work).

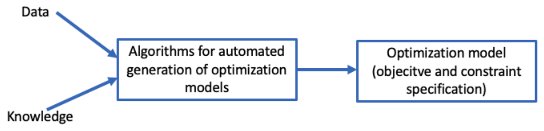
Data
Knowledge
Algorithms for automated generation of optimization models
Optimization model (objecitve and constraint specification)

Figure 1: Automated Optimization Model Derivation

As optimization models need to incorporate the effect of the decision variables, the input data for such optimization model generators needs to include information about the historical decisions, e.g., as made by human experts. In many real world use cases such historical data exists, potentially making this approach feasible. Learning any model from data introduces uncertainty and inaccuracy in the learnt model. In this setting, such uncertainty and inaccuracy is compounded due to historical decisions' data. This is as decisions are explicit choices made by decision makers and therefore all possible decisions cannot be naturally modelled by a probability distribution. In addition, typically, historical data will not cover all of the possible decisions, but rather, will be from a more restricted set, for example determined by use case specific best practices. Moreover, the actual decisions made could also influence the distribution of the other variables in the dataset. Therefore, we cannot reasonably expect the automatically generated optimization model to capture the true behavior for all possible decisions, both in terms of the objective and the constraints.

As the generated optimization model is subsequently used to make decisions, any inaccuracies in the model could result in a decision which is suboptimal. Indeed, it may even be worse than the decision making practices currently in place, which were used to generate the data. Moreover, as the ultimate goal of this automatic model derivation is to enable none optimization experts to create optimization models, one cannot rely on optimization experts to manually tweak the model to improve the results. Therefore, such data-based model derivation must be carried out in an almost fully automated fashion. It is therefore the goal of this article to formally define the problem and outline possible approaches and promising directions for addressing these uncertainties and inaccuracies while allowing for such automated generation.

2 Formal Problem Definition

There is a system which can be influenced by a set of d decision variables x, i.e., $x \in \mathcal{X} \subset \mathbb{R}^d$ and a set of input covariates $p \in P \subset \mathbb{R}^k$. The system can be described by an objective function $f(x,p)$ and a feasibility set $\mathcal{X} \cap \Omega(p)$ where $\Omega(p) \subset \mathbb{R}^d$ can be defined by a set of T equations, $g_t(x,p) \leq 0$, $t \in \{1,\dots,T\}$, and § is used to incorporate constraints such as discreteness. Given this specification, the goal is to solve the following problem:

$$x^* = \arg\min_{x \in \mathcal{X} \cap \Omega(p)} f(x,p) \tag{1}$$

However, we do not have the model of the system, nor the real set of parameters p. Rather, we have some problem related knowledge K, and a set of data D so that each $d_i \in D$ is composed of a tuple of vectors $d_i =< x_i, y_i, p_i >$ such that $x_i \in \mathcal{X} \cap \Omega(p_i)$, and $y_i \in Y$ is a feature vector positively correlated with p_i. Moreover (y_i, p_i) are i.i.d from some joint distribution $\mathbb{P}$. Also, we have a mechanism which can generate a model composed of: A *predictive model* $m : Y \to P$ and an *optimization generation model* $o : K \times Y \times \mathbb{R}^d \times P \to \{f', g'_1, \dots, g'_T\}$. In this case, m is a typical predictive model, that (for example, a forecasting model), and o is the mechanism which takes the historical data, together with problem specific knowledge K, and outputs an objective function and set of optimization constraints.

Given a vector pair (y,p) drawn from $\mathbb{P}$, We can then use this model to find

$$x'^* = \arg\min_{x \in \mathcal{X} \cap \Omega'(y)} f'(x, m(y)) \tag{2}$$

Where $\Omega'(y)$ is defined by $x \in \mathbb{R}^d$ s.t such that $g'_t(x, m(y)) \leq 0, t \in \{1, \dots, T'\}$.

The first condition for $x^{'*}$ to be a good solution is that it is feasible with a high enough probability, i.e.:

$$\Pr(x^{'*} \in \Omega(p)) \geq (1 - \delta_1) \tag{3}$$

The second condition is that $x^{'*}$ provides a sufficiently good objective function value. One way to define this is that the solution found by the approximate model $\{f', g'_1, \dots, g'_T\}$ ϵ-approximates the optimal solution of the true system with a high probability. This is defined by the following equation:

$$\Pr(|f(x'^*, p) - f(x^*, p)| \leq \epsilon) \geq (1 - \delta_2) \tag{4}$$

An alternative way to define a good solution is by how much it improves over the existing *decision policy*, which would capture for example, the way that currently such decisions are made. We model this decision policy with a function $h : Y \times \mathcal{X} \to [0,1]$, where $h(y,x) = \Pr_y(x)$, $y \in Y$, $x \in \mathcal{X}$, where $\Pr_y$ is a function of x determined by y which for each x, gives the probability that x is selected (We assume that existing decision policies enable selecting between a finite number of actions). Let us denote by $X_h(y)$ the random variable that is the actual value x selected by $h(y,x)$. Then we can define the following condition:

$$\Pr(f(X_h(y), p) - f(x'^*, m(y)) \geq \epsilon) \geq (1 - \delta_2) \tag{5}$$

Which specifies that x'^* improves h by ϵ with a high enough probability. An alternative formulation can require that the improvement is by at least an ϵ percentage: :

$$\Pr(f(x'^*, m(y)) \leq (1 - \epsilon) f(X_h(y))) \geq (1 - \delta_2) \tag{6}$$

Needless to say, there is a significant difference between the quality requirement as specified in Equation 4 and the quality requirement as specified by Equation 5 or Equation 6, as Equation 4 requires finding an approximation to the global optimum, while the other only requires a local improvement. Therefore, it should be much easier to obtain algorithms that enable Equation 5 or Equation 6. Of course, the value resulting from enabling Equation 4 is expected to be much higher unless h already is close to optimal. However, any solution that is able to generate a model that provides Equation 5 or 6 can be used repeatedly to provide incremental improvements as described in Figure 2. Therefore, there is also significant value in algorithms that are able to generate optimization that meets Equation 5 or 6.

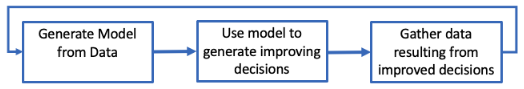
Generate Model from Data
Use model to generate improving decisions
Gather data resulting from improved decisions

Figure 2: Incremental improvement process

3 Related Work

There are existing works on creating optimization models or parts of optimization models from data. For example, [Arcangioli *et al.*, 2016] and [De Raedt *et al.*, 2018] focus on learning constraints from data. In [Lombardi *et al.*, 2017], a machine learning model of a complex system (an example system in the paper is a thermal aware dispatching system of a CPU) is generated from data, and then subsequently automatically transformed into an optimization model. The work in [Subramanian *et al.*, 2019] not only learns machine learning models for multiple components of a complex industrial process, but also incorporates knowledge specification in the form of intermediate storage nodes, from which a process-wide optimization model is generated. However, while these works address automatic derivation of optimization models, they do not address the uncertainty resulting from the generation process. Rather, there seems to be an implicit reliance on an optimization expert to validate these models, refining them if necessary.

Works such as [Wilder *et al.*, 2019] and [Elmachtoub and Grigas, 2020] address the issue of integrating machine learning and optimization models, as well as addressing how to account for the uncertainty in the optimization models. Indeed, these works define an objective function for the machine learning model that takes into account the optimization objective. However, the problem addressed by these works are significantly different than the problem described in Section 2: it assumes that the structure of the functions describing the objectives and constraints are given, and that there is uncertainty only regarding the covariates of the objective function. Moreover, these works do not account for machine learning models that can be influenced by decision variables, which is historical data with very different behavior as described above. This makes it a significantly simpler (although still challenging) problem than the one defined in Section 2. In addition, it tries to optimize the expected objective given the predicted parameters, and not the probability of optimality or probability of improvement as defined by Equations 4 - 6.

Section 2 assumes that the optimization problem is specified explicitly through a set of mathematical functions $\{f, g_1, \ldots, g_t\}$. However, there are also alternative approaches to specify optimization problems. The use of metahueristics, for example, (see [Yang, 2011]) enable the constraints and objectives to be specified in a functional form using any programming language, enabling none optimization experts to easily specify such problems. However, in many cases, the metahueristics algorithms have to be manually tuned by experts to provide a valuable solution, which means that experts still need to be involved in the end-to-end solution process. The explicit mathematical representation our work is targeting can typically be used as input to state of the art mathematical programming solvers, significantly increasing the likelihood that good solutions will be found.

Another well known optimization approach is *Reinforcement Learning*, or RL ([Sutton and Barto, 2018]). RL by definition seems to be more suitable for none optimization experts for several reasons. First, the problem is specified at a high level, in terms of states, actions, rewards and transition probabilities. In addition, learning is by design embedded into the solution concepts. However, creating good RL models still requires experts, for example to define a good state space representation and parameterization, decide on the appropriate learning algorithm, reward function, etc. Therefore, while in this work we focus on explicit optimization model representation, automatic derivation of RL models is another important topic to address, and indeed, is the subject of some of our future work. In addition, RL algorithms are currently less suitable for handling optimization problems with complex constraints. Finally, some of the RL work has direct relevant to this work as well: The work in [Thomas *et al.*, 2015] defines a problem similar in spirit to Equation 6 in the sense that a policy is sought that improves, with a high enough probability, the objective as compared to the objective obtained by the policy used to generate the data used for RL. In [Yu *et al.*, 2020], an environment model, which is then used as input to the RL algorithm, is learnt from data. and the uncertainty of the environment model is explicitly taken into account in the reward function.

4 Solution Requirements and Possible Approaches

To satisfactorily address the problem described in Section 2 requires incorporating data-driven regression equations in both the estimated objective function f' and the estimated constraints $\{g'_1, \ldots, g'_{T'}\}$, as well as involve the optimization decision variables as features in the learnt models.

It is important to note that the accuracy requirement for the generated objective function is very different from the requirements for the constraints. For example, if f' has the monotonicity property as appearing in Equation 7 (with p, y being in the support of $\mathbb{P}$), then any x that maximizes f' will also maximize f.

$$\forall x_1, x_2, \quad f(x_2, p) > f(x_1, p) \Rightarrow f'(x_2, m(y,)) > f'(x_1, (m(y)) \tag{7}$$

This does not hold, however, for the constraints, which may require much better estimates, in the traditional metrics used to measure prediction models, than the objective. Therefore, to achieve the required accuracy, it may be required to enforce two desired characteristics: *model fidelity* and *data sufficiency*. By model fidelity, we require that each of the regression equations that participate in the objective function or the constraints enjoys acceptable prediction accuracy in the neighborhood of the optimal solution. By data sufficiency we mean that the optimal solution also should ideally belong to a region in the space of decision variables where we have sufficient historical data. This is important because decision variables are input features that excite the various regression functions in the above problem formulation. These two desiderata are likely related to each other, but distinctly important for producing acceptable solutions in practice.

In order to achieve these desiderata, it will be necessary to restrict the solution space. I.e., Equation 2 will have to be further restricted so that the feasible region for the solution x'^* will have sufficient data sufficiency and model fidelity. Given this, it will be difficult to obtain a solution which is an approximate global optimal as desired by Equation 4. Therefore, in this section, we focus on ways to achieve an *improved solution* as defined by Equation 5 or Equation 6.

Focusing on finding an improved solution, and keeping the requirements of model fidelity and data sufficiency in mind, we suggest below a general approach. Assume we fix the parameter δ_2 which defines the probability by which a solution should be better than existing practices. What we would now like is to find the spaces of model fidelity and data sufficiency that maximize the ϵ so that Equation 5 holds. (a similar formulation can be derived for Equation 6). Specifically, our methodology follows the following steps: we begin with an initial set of model fidelity and data sufficiency regions. We then derive an optimization model $(\{f', g'_1, \ldots, g'_{T'}, m\}$ using these regions. We then test how much of an improvement the generated model provides. We repeat this process using multiple model fidelity and data sufficiency regions.

In the sequel we outline two possible methods to implement this approach: One using robust optimization, and one using Gaussian processes.

Robust optimization ([Ben-Tal *et al.*, 2009; Bertsimas *et al.*, 2011]) is a method for solving optimization problems with uncertainty. The method consider that uncertain parameters of the optimization problem belong to some bounded region known as an *uncertainty set* and constructs an alternative optimization problem, known as the *robust counterpart*, which, in many cases, can be solved efficiently. The method guarantee that solution of the robust counterpart problem is feasible for any realization of the uncertainty within the bounded region and provide a best objective value for the worst possible set of parameters. Moreover, under some natural assumptions on the distributions of the uncertain parameters this method allows to construct bounded uncertainty set even if the corresponding probability distributions have infinite support, ensuring that any constraint of the optimization problem holds with given probability. Similarly, the method can guarantee the worst bound of the objective value with any given probability.

Given this ability to formulate and solve robust optimization problems, a possible approach would be to find the optimal regions in terms of model fidelity and data sufficiency so that the generated uncertain optimization model will be computationally tractable and so that the resulting uncertainty set will have: (*i*) the minimal possible volume, allowing to build a robust counterpart whose solution guarantees the best possible improvement, and (*ii*) , the structure that allows to build the computationally tractable robust counterpart. This requires that the optimization generation mechanism o is able to provide an estimate for error between the actual optimization problem and learnt optimization problem, both in terms of the constraints and the objectives. In addition, obtaining a computationally tractable optimization model places additional restrictions on optimization generation mechanism. For example, restricting o to generate piecewise linear functions allows construction of an uncertain mixed integer linear program (MILP), that can be further transformed into a MILP robust counterpart, assuming a polyhedral uncertainty set is selected. While the choice of the structure of the uncertainty set depends on the tractability requirements of the robust counterpart, the volume of the uncertainty set depends on the precision of the predictive model. This means that the quality of the objective function could be improved, once we obtain a better prediction of the uncertain parameters.

Gaussian Processes The solution method discussed in this section can also be tackled using Gaussian processes (GPs). Examining the problem from a Bayesian perspective, we can view both f and Ω stochastically. In fact, restricting Ω to subsets defined via a finite set of (linear, quadratic, etc.) inequalities $\{f_i(x) \leq 0\}$ provides us with a well-defined notion of stochasticity for Ω (though this requires a global upper bound on the number of constraints). Given $\Omega \sim \mathcal{O}$, which is established either as a prior derived from business problem knowledge or as a posterior that additionally absorbs historical data, we can formulate a joint posterior $GP_{\mathcal{O}}$ that ties f to Ω, so that $GP_{\mathcal{O}}(\Omega = \Omega') \equiv GP_{f \mid D \cap \Omega'}$, the latter being the Gaussian process posterior established on f given D restricted to Ω'. Given some x_0, for instance, the current best practice or policy, $GP_{\mathcal{O}}$ then provides the marginal probability of improvement $\Pr[f(x', \cdot) < f(x_0, \cdot) + \epsilon]$ for the minimization problem defined.

This marginal probability of improvement is a strong metric with which to assess any prediction-optimization scheme, namely, any scheme that, given data D associated with an objective target f and a (prior or posterior) $\Omega \sim \mathcal{O}$, produces an optimal solution x'. It combines the potential for gain on f with the risk of choosing x' far from what the business expert believes or the data considers to be sensible constraints, while also making sure that we learn f from the most relevant data points. We can subsequently use the marginal probability of improvement to continuously improve our estimate x' of x^* by iteratively adapting the feasibility and sufficiency regions so that we optimize the marginal probability of improvement. This can be done, for example, by using Bayesian optimization.

5 Summary and Future work

In this work, we introduced the challenge of automating the creation of optimization models from data, which also needs to include a unique type of data, historical decisions. As we discussed, such generation must take into account the inherent uncertainty of both the generated objectives and generated constraints, while accounting for the fact that the optimizations' decision variables are features in the learnt models. We formally defined the problem to be addressed, and provided possible approaches to address this problem. Of course, the outlined methods have to be fully fleshed out, placed on a firm theoretical basis and tested empirically. We already have some more detailed results and algorithms which are beyond the scope of this work, and are continuing to expend them as a major thrust of our ongoing research.

The approaches we outlined in Section 4 follow the traditional "predict-then-optimize" approach, where we first learn an optimization model from data, and then use this model to optimize. Therefore, a possible avenue of future work is to enhance the work to include techniques such as the ones appearing in [Elmachtoub and Grigas, 2020] and [Wilder *et al.*, 2019] which incorporate directly the optimization model objective during the models' learning. Another avenue for future work is to find methods which attempt to address the global optimality of the generated models (Equation 4), rather then just seeking gradual improvement (Equations 5 and 6). An additional future direction is to enable the end-to-end generation of RL models as discussed in Section 3.

Incorporating the uncertainty of model generation so as to enable end-to-end optimization model creation is a huge area, and we feel that we have only begun to scratch the surface. Moreover, it has immense potential benefits in terms of the value the widespread use of optimization can provide to industry and society. We therefore hope that this work will encourage and motivate others to carry out research in this field. Furthermore, this is just one, albeit major, aspect in enabling none optimization experts to create optimization models. Addressing this broader topic of optimization model creation by none experts is also a ripe area for research and innovation.

References

[Arcangioli *et al.*, 2016] Robin Arcangioli, Christian Bessiere, and Nadjib Lazaar. Multiple constraint acquisition. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 698–704. AAAI Press, 2016.

[Ben-Tal *et al.*, 2009] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[Bertsimas *et al.*, 2011] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[De Raedt *et al.*, 2018] Luc De Raedt, Andrea Passerini, and Stefano Teso. Learning constraints from examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Elmachtoub and Grigas, 2020] Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize", 2020.

[Lombardi *et al.*, 2017] Michele Lombardi, Michela Milano, and Andrea Bartolini. Empirical decision model learning. *Artificial Intelligence*, 244:343–367, 2017. Combining Constraint Solving with Mining and Learning.

[Subramanian *et al.*, 2019] Dharmashankar Subramanian, Pavankumar Murali, Nianjun Zhou, Xiang Ma, Giovane Cesar Da Silva, Raju Pavuluri, and Jayant Kalagnanam. A prediction-optimization framework for site-wide process optimization. In *2019 IEEE International Congress on Internet of Things (ICIOT)*, pages 125–132. IEEE, 2019.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[Thomas *et al.*, 2015] Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2380–2388, Lille, France, 07–09 Jul 2015. PMLR.

[Wilder *et al.*, 2019] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1658–1665, 2019.

[Yang, 2011] Xin-She Yang. Metaheuristic optimization. *Scholarpedia*, 6:11472, 01 2011.

[Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020.