Learning From Scenarios for Repairable Stochastic Scheduling

Kim van den Houten[1 [0009−0001−7540−980X]], David M.J. Tax[1 [0000−0002−5153−9087]], Esteban Freydell[2 [0009−0001−6992−2869]], and Mathijs de Weerdt[1 [0000−0002−0470−6241]]

[1] Delft University of Technology, Delft, The Netherlands
[2] DSM-Firmenich, Delft, Netherlands

Abstract. This is an encore abstract of the CPAIOR paper [3]. This paper shows an approach for learning a deterministic representative for an optimization problem with uncertain parameter values in the constraints.

Keywords: Stochastic Scheduling · Repair · Decision-focused learning

Introduction Decision-making is challenging due to the stochastic nature of real-world processes. Constrained Optimization (CO) models are commonly used, but unknown parameters during decision-making can lead to infeasibility. In practice, such infeasibilities are corrected as reality unfolds. Historical data, represented as scenarios, are often available, but simple averaging ignores uncertainty. Stochastic programming [7] offers an alternative but faces challenges like scalability.

Decision-focused learning (DFL) [6] introduces a novel approach to stochastic optimization by embedding an optimization model, like Constraint Programming, into a training process to minimize regret loss [1]. This method faces challenges in backpropagating through combinatorial problems due to discontinuous solution changes. Building on DFL, our idea is to learn a deterministic surrogate for the stochastic problem by minimizing the post-hoc regret loss [4]. Research shows that the backpropagation can be handled using score function gradient estimation [8]. Our contribution (Algorithm 1) demonstrates how this technique can be applied to a stochastic scheduling problem (Resource Constrained Project Scheduling (RCPSP) [5]) with historical processing time data and thus uncertainty in the constraints.

Algorithm 1: DFL

Require: $\mathcal{D}_{train} = \{y_i\}_{i=1}^{n_{train}}$, $\mathcal{D}_{test} = \{y_i\}_{i=1}^{n_{test}}$
    Initialize $\widehat{y} \sim p_\theta(\widehat{y})$ such that $\widehat{y} \sim \mathcal{N}(\mu = \theta_\mu \cdot \bar{y}, \sigma = \theta_\sigma \cdot \bar{\sigma})$
    for each epoch **do**
        for each batch in $\mathcal{D}_{train}$ **do**
            for each instance $(y_i, z^*(y_i))$ in batch **do**
                Sample $\widehat{y}$ from $p_\theta(\widehat{y})$
                Pass $\widehat{y}$ to solver to get schedule
                Compute post-hoc regret$(\widehat{y}, y_i)$
            end for
            Update θ with score-function: $\theta = \theta - \mathrm{lr} \cdot \nabla_\theta PRegret(\widehat{y}, y_i) \nabla_\theta \log(p_\theta(\widehat{y}))$
        end for
    end for
    Pass $\widehat{y} = \mu$ to solver to get schedule
    Evaluate post-hoc regret on $\mathcal{D}_{test}$

Background. DFL procedures minimize the post-hoc regret loss by gradient-based optimization: $\frac{\delta PRegret(\widehat{y},y)}{\delta\theta} = \frac{\delta PRegret(z_{corr}(\widehat{y},y),y)}{\delta z_{corr}(\widehat{y},y)} \frac{\delta z_{corr}(\widehat{y},y)}{\delta \widehat{y}} \frac{\delta \widehat{y}}{\delta \theta}$. The term $\frac{\delta z_{corr}(\widehat{y},y)}{\delta \widehat{y}}$ gives a zero-gradient problem.

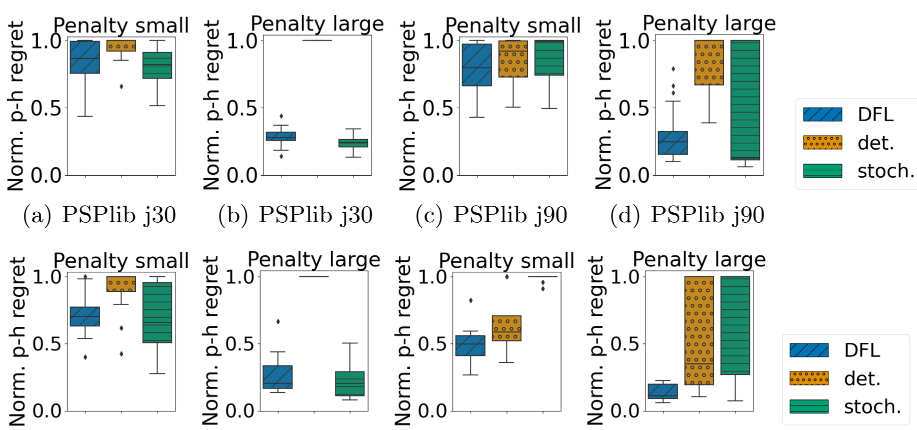
Silvestri et al. [8] show that this zero-gradient problem can be solved by using the score-function gradient estimator (also known as likelihood ratio gradient estimator [2]) that uses:

$$\nabla_\theta \mathbb{E}_{\widehat{y}\sim p_\theta(y)}[PRegret(\widehat{y},y)] = \mathbb{E}_{\widehat{y}\sim p_\theta(y)}[PRegret(\widehat{y},y)\nabla_\theta \log(p_\theta(\widehat{y}))] \tag{1}$$

for which the derivation can be found in [8].

Contribution We adapt DFL to align with our scheduling problem in Algorithm 1. The data $\mathcal{D} = \{y_i\}_{i=1}^n$ comprises historical examples of processing times y. We aim to learn which predictor $\widehat{y}$ minimizes the post-hoc regret. During training, we sample predictions from an Normal $\widehat{y} \sim \mathcal{N}(\mu = \theta_\mu \cdot \bar{y}, \sigma = \theta_\sigma \cdot \bar{\sigma})$. In each training step, we sample a point y_i from the trainig data and a prediction $\widehat{y}$, compute schedule $z^*(\widehat{y})$, and update θ using the score-function gradient estimator.

Experiments To understand the potential of DFL, we compare performance to deterministic and stochastic programming formulations. We hypothesize that both stochastic programming and DFL outperform the more naive deterministic approach. Furthermore, we explore when DFL or stochastic programming performs better; we expect that DFL has better scalability to larger instances.



(a) PSPlib j30 (b) PSPlib j30 (c) PSPlib j90 (d) PSPlib j90

(f) Small indus. (g) Small indus. (h) Large indus. (i) Large indus.

Fig. 1: Normalized post-hoc regret per instance set - penalty setting (smaller regret is better). The box spans from the 25th to the 75th percentile, visualizing the median and interquartile range.

Conclusion Results indicate that stochastic programming is dominant when it can find the optimal solution. However, we have shown that DFL is a promising alternative to stochastic programming, especially for larger instances and with a large penalty setting. Further interesting directions are investigating alternative gradient estimators or reinforcement learning-inspired algorithms

References

1. Elmachtoub, A., Grigas, P.: Smart "predict, then optimize". Management Science **68**(1), 9–26 (2022)
2. Glynn, P.W.: Likelihood ratio gradient estimation for stochastic systems. Commun. ACM **33**(10), 75–84 (oct 1990). https://doi.org/10.1145/84537.84552
3. van den Houten, K., Tax, D.M., Freydell, E., de Weerdt, M.: Learning from scenarios for repairable stochastic scheduling. In: International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 234–242. Springer (2024)
4. Hu, X., Lee, J.C.H., Lee, J.H.M.: Predict+optimize for packing and covering LPs with unknown parameters in constraints. arXiv **2209.03668** (2022)
5. Kolisch, R., Sprecher, A.: PSPLIB - a project scheduling problem library. European Journal of Operational Research p. 205–216 (1996)
6. Mandi, J., Kotary, J., Berden, S., Mulamba, M., Bucarey, V., Guns, T., Fioretto, F.: Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. arXiv **2307.13565** (2023)
7. Ruszczyński, A., Shapiro, A.: Stochastic Programming, Handbook in Operations Research and Management Science. publisher (2003)
8. Silvestri, M., Berden, S., Mandi, J., İrfan Mahmutoğulları, A., Mulamba, M., Filippo, A.D., Guns, T., Lombardi, M.: Score function gradient estimation to widen the applicability of decision-focused learning. arXiv **2307.05213** (2023)