Appeared at BNAIC'22, Mechelen, Belgium, November 2022

Machine Learning for the Cyclic Hoist Scheduling Problem

Nikolaos Efthymiou and Neil Yorke-Smith[0000−0002−1814−3515]

STAR Lab, Delft University of Technology, The Netherlands
n.efthymiou@student.tudelft.nl, n.yorke-smith@tudelft.nl

Abstract. The Cyclic Hoist Scheduling Problem (CHSP) is a well-studied combinatorial optimization problem. One of the existing approaches to solving CHSP is Constraint Programming (CP). In this study, we examine the possibility of predicting the optimal (minimum) cycle period p of a CHSP instance – without solving it – using supervised Machine Learning (ML) approaches. We also suggest using this prediction to calculate upper and lower bounds of p, and we investigate the impact of these bounds on the performance of a CP solver. The results of our experiments show that: 1) ML models, in particular deep neural networks, can be good predictors of the optimal p, and 2) providing tight bounds for p around the predicted value to a CP solver can significantly reduce the solving time without compromising the optimality of the solutions.

Keywords: Cyclic Hoist Scheduling · Machine Learning · Constraint Programming

1 Introduction

The *Cyclic Hoist Scheduling Problem* (CHSP) is an optimization problem of practical and theoretical importance [2]. The aim is to find a schedule for one or multiple industrial hoists that move objects between tanks, while minimizing the *cycle period* p, which is defined as the difference between the start time of processing two consecutive objects [3, 4, 6].

One of the existing techniques for solving CHSP is *Constraint Programming* (CP) [1, 7]. An efficient exact CP model for the CHSP problem suggested by Wallace and Yorke-Smith [7] uses calculated lower and upper bounds of p (p^{calc}) to specify the space of feasible solutions. Given that such computation reflects the theoretical maximum range of the period, p^{calc} tend to be quite loose.

We explore the idea of predicting the optimal value of p – without solving the CSHP instance – and then restricting the range in which the solver is trying to find a solution. The hypothesis is that this could result in lower solving times (t) without affecting the period of the best solution found. Further, when the bounds (p^{pred}) derived from the prediction become tighter, the solving time could decrease even further.

2 Methodology

In order to study our hypotheses, we train various ML models using Keras and we test their accuracy. Specifically, we fine-tune Deep Neural Network (DNN), Random Forest (RF) and Gradient Boosting Tree (GBT) models. For training these models, we obtain a large number of CSHP instances ($N = 166,320$) by implementing a random generator (by following patterns and settings found in industry examples [3, 4, 6]). As a test set, we use a subset of the randomly generated instances, together with several industry instances. We solve the random and industry instances using the CP model proposed by Wallace and Yorke-Smith [7] with the Google OR-Tools CP solver [5]. In this way, we find the actual optimal value of p for each instance, which is used as the target value in training the ML models. A challenge is that, for a CHSP instance with n tanks, considering all features leads to a dimensionality of $(n+1)^2 + 3n + 4$. We suggest using a fixed number of independent variables for the ML models, by replacing instances' attributes per tank with their descriptive statistics.

After predicting the p of each instance, we modify the CP model by providing tighter bounds for p, around p^{pred}. For this we explored $\pm 5\%$ and $\pm 20\%$ margins. We then assess the effectiveness of the CP solver when these tighter bounds are used, as explained next.

3 Results

Computational experiments showed that the DNN ML model performed best, with a MAPE of 3.38 on the random test set. When the predicted bounds p^{pred} are used instead of the calculated bounds p^{calc}, the CP solver found the original (optimal) p in most cases: 94.6% in the case of $\pm 5\%$ margin and 98.8% in the case of $\pm 20\%$ margin. As hypothesised, the solving time is significantly lower when these predicted bounds of p (p^{pred}) are used ($t^{pred}_{5\%}$: $\overline{X} = 0.58$, $s = 4.93$; $t^{pred}_{20\%}$: $\overline{X} = 1.27$, $s = 11.01$; t^{calc}: $\overline{X} = 1.91$, $s = 14.09$). Moreover, such a decrease is much larger when the predicted bounds become tighter: the relative decrease in solving time, when an optimal solution was found, is -70.7% in the case of $\pm 5\%$ margin and -33.1% when $\pm 20\%$ margin is used. This improvement is more modest in the case of satisfied solutions, but remains statistically significant.

In conclusion, predicting the optimal p value of a CHSP instance is possible and integrating such a prediction into a CP solver can considerably accelerate the solving phase. Given that the ML models implemented in this study do not consider CSHP instance attributes like the number of tracks and the loading/unloading times, this could be investigated in future work.

Acknowledgements Thanks to the BNAIC reviewers. This research was partially supported by TAILOR, a project funded by the EU Horizon 2020 programme under grant 952215.

References

1. Feng, J., Chu, C., Che, A.: Cyclic jobshop hoist scheduling with multi-capacity reentrant tanks and time-window constraints. Computers & Industrial Engineering **120**, 382–391 (2018), https://doi.org/10.1016/j.cie.2018.04.046
2. Lee, C.Y., Lei, L., Pinedo, M.: Current trends in deterministic scheduling. Annals of operations Research **70**, 1–41 (1997), https://doi.org/10.1023/A:1018909801944
3. Leung, J.M., Zhang, G., Yang, X., Mak, R., Lam, K.: Optimal cyclic multi-hoist scheduling: A mixed integer programming approach. Operations Research **52**(6), 965–976 (2004), https://doi.org/10.1287/opre.1040.0144
4. Manier, M.A., Lamrous, S.: An evolutionary approach for the design and scheduling of electroplating facilities. Journal of Mathematical Modelling and Algorithms **7**(2), 197–215 (2008), https://doi.org/10.1007/s10852-008-9083-z
5. Perron, L., Furnon, V.: OR-Tools release 9.3. https://developers.google.com/optimization/ (2022)
6. Phillips, L.W., Unger, P.S.: Mathematical programming solution of a hoist scheduling program. AIIE transactions **8**(2), 219–225 (1976), https://doi.org/10.1080/05695557608975070
7. Wallace, M., Yorke-Smith, N.: A new constraint programming model and solving for the cyclic hoist scheduling problem. Constraints **25**(3), 319–337 (2020), https://doi.org/10.1007/s10601-020-09316-z