

An Intelligent Decision Support System for Production Planning in Garments Industry

Rui Ribeiro^{1,2}[0000-0001-8078-4148], André Pilastrri¹[0000-0002-4380-3220], Hugo Carvalho¹[0000-0003-3631-1487], Arthur Matta¹[0000-0002-4902-9483], Pedro José Pereira²[0000-0002-6169-8778] Pedro Rocha³, Marcelo Alves³, and Paulo Cortez²[0000-0002-7991-2090]

¹ EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute, Guimarães, Portugal

{`rui.ribeiro`, `andre.pilastrri`, `hugo.carvalho`, `arthur.matta`}@`ccg.pt`

² ALGORITMI Centre, Dep. Information Systems, University of Minho, Guimarães, Portugal, `id6927@alunos.uminho.pt`, `pcortez@dsi.uminho.pt`

³ INFOS - Informática e Serviços, Leça da Palmeira, Portugal
{`pedro.rocha`, `marcelo.alves`}@`infos.pt`

Abstract. In this paper, we propose an Intelligent Decision Support System (IDSS) that combines prediction and optimization for production planning. We worked with a company that provides software for the garments Industry and that had access to real-world data related with a client that works with subcontractors. Using an Automated Machine Learning (AutoML) approach, we firstly target four predictive tasks that are crucial to estimate production planning indicators. Then, we use historical data and one of the predicted indicators to search for the best subcontractor allocation plan, which minimize both the cost and production time via an Evolutionary Multiobjective Optimization (EMO) algorithm (NSGA-II), achieving interesting results.

Keywords: Textile Production Planning · AutoML · NSGA-II

1 Introduction

Currently, there is a pressure in industries to increase efficiency (e.g., reduce operating costs and time) in order to compete in their markets. One way is to adopt an Intelligent Decision Support Systems (IDSS), which incorporate Artificial Intelligence techniques to provide actionable knowledge from raw data [2]. In this paper, we assume an IDSS for the garments Industry and that is based in the concept of Adaptive Business Intelligence (ABI) [11], which combines Machine Learning (ML), to predict relevant decision context variables, with Modern Optimization (MO) [7], to search for the best decision choices (according to one or more objectives).

There are some related works that employ MO methods to support production plans in the textile industry. For instance, in [1] Genetic Algorithms (GAs) were used to create production orders involving the spinning and weaving areas of the fabrication process. GAs were also adopted in [12] to optimize

job orders of textile production lines. A combination of GA with Simulated Annealing was used by [13] to create energy efficient production orders. In another study, NSGA-II was used by [10] to solve a multi-objective multi-site order scheduling problem in the production planning stage with the consideration of multiple plants, multiple production departments and multiple production processes. More recently, [14] used a mathematical programming model to optimize textile production considering diverse “green” goals (e.g., waste reuse, energy recycling) and [3] optimized the master production scheduling using GA. Within our knowledge, none of these works adopted a data-driven ABI approach that combines predictive and prescriptive analytics, as provided by ML and MO algorithms. In this paper, we follow such innovative ABI combination by using an Automated Machine Learning (AutoML) [9] to first predict four important garment subcontractor decision variables. Then, we adopt historical data and one of the predicted variables (production time) to feed an Evolutionary Multiobjective Optimization (EMO) that searches for the best subcontractor allocation plan, simultaneously minimizing the total allocation cost and time.

2 Materials and Methods

2.1 Garment Data

The data was provided by INFOS, which is Portuguese software company that works with several textile industry clients. The company developed an Enterprise Resource Plan (ERP) that supports the production of garments. The goal of this research is to develop an IDSS based on the ABI concept and that will be integrated into the INFOS ERP system, allowing it to automatically design garment subcontractor plans regardless of size of the company and the complexity of the production order. The subcontractor selection is a non trivial task, since is a large range of textile operations, each involving costs and delivery dates. We collected all company garment related records, including purchase and manufacturing orders, from 2016 to 2020. The data was then divided in three major groups: purchase of raw material, manufacturing and subcontractor. Next, we implemented an Extraction, Transformation, Load (ETL) process to select and clean the data (e.g., removal of missing features and records with wrong dates). All data processing procedures (including the ABI system) were implemented in the Python language by the authors.

Table 1 describes the input features (**Attribute**), their description (**Description**), data **Type**, number of **Levels** and **Domain** values separated by objective (four predictive targets and one optimization task). The final set of input features was obtained after several iterations of predictive task executions. The datasets for the predictive (regression) tasks include: Lead Time – 3,315 records; Production Time – 25,449 examples; Production Waste – 24,425 instances; and Delivery Delays – 6,016 records. Finally, the optimization objective (Production Plan) contains 5,500 records related with subcontractors.

Regarding the target output target variables for the predictive tasks, we detected that the company does not have records of them, being necessary to

Table 1. Description of the input features by objective.

Objective	Attribute	Description	Type	Levels	Domain
Lead Time	Supp_cod	Supplier identification	Integer	102	[14, 2265]
	Date_purch	Date of purchase order	Date	873	-
	Rmat_cod	Raw material code	String	188	-
	Qty	Quantity to buy	Integer	876	[1, 11650]
Production Time	Subc_cod	Subcontractor identification	Integer	275	[0, 9999]
	Mat_cod	Final product code	String	846	-
	Oper_desc	Textile operation	String	93	-
	Qty	Quantity to produce	Float	9356	[1, 97512]
Production Waste	Mat_cod	Final product code	String	864	-
	Qty	Quantity to produce	Float	4646	[1, 13448.3]
	Subc_cod	Subcontractor identification	Integer	41	[8, 9996]
	Rmat_cod	Raw material code	String	1386	-
Delivery Delays	Plan_endate	Planned date to end production	Date	940	-
	Mat_cod	Code of the final product	String	205	-
	Qty	Quantity to produce	Integer	3156	[3, 75838]
Production Plan	Oper_desc	Textile operation	String	94	-
	Avgp_cost	Average cost of textile operation	Float	47	[0.01, 1.75]
	Price	Cost of textile operation for given product per subcontractor	Float	247	[0.01, 11.60]
	Subc_cod	Subcontractor identification	Integer	293	[24, 2254]
	Mat_cod	Code of the final product	String	169	-
	Capacity	Subcontractor production capacity by textile operation	Integer	14	[100, 3000]

calculate them: *Ldtime* was obtained by subtracting the receiving date of a order from the placement order date and if resulting value was negative that row was discarded; for *Prod.days* we create a function that subtracts the production finish date from the planned production start date and outputs the number of working days between the two dates and the if the number of days was negative that row was discarded; in the case if *Waste_ratio* we first subtracted the produced quantity from the quantity to produce and if the resulting value was positive it was changed to zero, afterwards we divided the absolute result by the quantity to produce, multiplying the final result by 100; finally for *Delay.days* we create a function that subtracts the scheduled delivery date finish date from the delivery date and outputs the number of working days between the two dates and the if the number of days was negative it was changed to zero. Table 2 describes the four output target variables with their description (**Description**), data type (**Type**), number of levels (**Levels**) and domain values (**Domain**).

In terms of preprocessing, since the String variables had a high cardinality, we employed a Label Encoder, in order to transform each level into a distinct numeric value. This option provided better results when compared with the known

Table 2. Description of the output target variables.

Target	Description	Type	Levels	Domain
Ldtime	Days between delivery date and purchase date	Integer	94	[0, 59]
Prod_days	Working days to produce a certain quantity	Integer	47	[1, 63]
Waste_ratio	Percentage of wasted material	Integer	247	[0, 100]
Delay_days	Working days between scheduled delivery date and delivery date	Integer	293	[0, 64]

One-Hot encoding, which created a very high number of input features. As for the Date features, we adopted the proleptic Gregorian ordinal of a date, allowing to provide a simpler numeric value. Then, all numeric inputs were normalized by using a z -score standardization.

2.2 Intelligent Decision Support System

The proposed IDSS contains three main modules (Figure 1): data extraction and processing, prediction and optimization. The first module is responsible for receiving the garment data, selecting the features for each objective and then creating the necessary input for prediction. The prediction module receives the data separated by predictive task splitting it into training and test sets (data separation, according to the adopted cross-validation method). Then, it trains the predictive models (model training), evaluating the models performance (model evaluation), selecting and storing the best prediction model (model selection). Then, the user inserts the data related to the lead time, using the respective model to predict the number of days that will take to receive the raw materials and can define a starting and end date for production. Finally the optimization module receives the subcontractors data (Table 1), filtered by the product to manufacture and the textile operations to execute, the quantity to produce and maximum allowed dates (all provided by the user). Then, the MO algorithm uses this data and also one of the predicted indicators (production time) to search for the best subcontractor quantity allocation, aiming to reduce the total costs and time.

To reduce the modeling effort during the development of the prediction module, we adopted the H2O AutoML tool that provided good results in recent AutoML benchmark study [9]. The AutoML was configured to automatically select the best regression model and its hyperparameters based on the best Mean Absolute Error (MAE), using a internal 10-fold cross-validation applied over the training data. Five different ML algorithms were searched by the tool: Random Forest, Extremely Randomized Trees, Generalized Linear Models, Gradient Boosting Machine and two Stacked Ensembles, one with best model of each family and other with all trained models. An external 10-fold cross-validation was executed to evaluate the ML models and the quality of the regression was accessed by using the MAE and Normalized MAE (NMAE) metrics. The lower the values, the better are the predictions. The NMAE measure normalizes the MAE

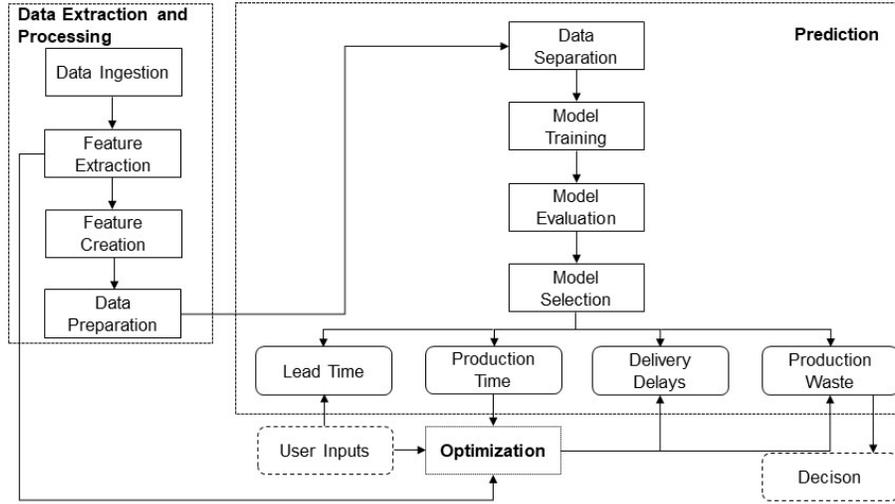


Fig. 1. Flow diagram describing the behaviour of the IDSS.

by the range of the output target on the test set, thus it provides a percentage that is easy to interpret and that is scale independent.

A production order can be defined as a composition of tasks that are executed sequentially. Each task can be represented by a set of candidate subcontractors offering similar services, where each service can have a different value in price and quantity per subcontractor. The subcontractor allocation is defined as a multi-objective task (i.e., reduce both cost and time), thus we employ a Pareto approach via an EMO algorithm, namely NSGA-II [7], as implemented in the `pymoo` Python module [4]. NSGA-II is a multi-objective optimization algorithm with three distinctive features: fast non-dominated sorting approach, fast crowded distance estimation procedure and usage of a simple crowded comparison operator [8]. When compared with other hypervolume based algorithms (e.g, SMS-EMOA), the NSGA-II algorithm tends to obtain competitive results when only two or three objectives are optimized [6]. The algorithm returns a population of non dominated solutions, each representing a different subcontractor allocation and that is associated with a distinct cost-time trade-off. The full subcontractor optimization can be defined in terms of x textile sequential operations that need to be executed. For each operation, there are y candidates (subcontractors) with different price and capacity parameters. Each solution is naturally represented as a sequence of q_i integer values ($0 \leq q_i \leq q_{max}$), denoting the quantity assigned for each subcontractor i , where q_{max} denotes the total required quantity for operation x , and $i \in \{1, \dots, M\}$ and M represent the number of available subcontractors for operation x . We repair solutions by ignoring any excess of subcontractor allocation (first allocated subcontractor is served first) or by randomly distributing the deficit allocation to any of the available subcontractors. Each solution is evaluated in terms of total production plan cost

and allocation time. To compute these two goals, the EMO algorithm uses the production time prediction (as shown in Figure 1). Once the Pareto curve is optimized and for user selected trade-offs, we then compute the prediction indicators of the remaining targets (e.g., production waste), such that the user can further inspect the quality of the obtained solutions. In order to obtain a single measure per Pareto curve, we selected the Hypervolume (HV) measure, which represents the volume of the objective space when assuming a “worst” reference point [5]. The higher the HV value, the better is the Pareto curve optimization.

3 Experiments and Results

The average of the external 10-fold iteration predictive results (in terms of **MAE** and **NMAE**) are presented in Table 3. The table also presents the best ML **Model**. In general, low regression errors were achieved, with the NMAE values ranging from 3.6% to 9.2%. We particularly note that the best NMAE values were obtained for the target that is directly used by the NSGA-II MO (Prod_days produces an average NMAE error of just 3.6%). The selected ML algorithm was a stacked ensemble for three of the targets, while the Gradient Boosting Machine obtained the best results for the production waste prediction.

Table 3. AutoML predictive results for each predicted target.

Target	Model	MAE	NMAE
Ldtime	Stacked Ensemble (All Models)	3.31	9.20%
Prod_days	Stacked Ensemble (All Models)	1.60	3.63%
Waste_ratio	Gradient Boosting Machine	4.24	4.24%
Delay_days	Stacked Ensemble (Best of each family)	3.57	5.71%

For the optimization experiments, we analyzed a production order of 10,000 units of a product that requires three textile operations (cutting, tailoring and packaging) using one raw material. Using historical data, we then selected all the subcontractors that could execute these operations along with the respective cost and production capacity to create a subcontract allocation case study to utilize in the experiments. In total, the case study includes 26 subcontractors (which corresponds to the number of searched integers by the NSGA-II algorithm): cutting - 4 candidates, tailoring - 8 candidates and packaging - 14 candidates (4+8+14=26). To compute the cost and time associated with each solution, we use four attributes from Table 1 (*Subc_cod*, *Capacity*, *Price* and *Oper_desc*) and also the predicted **Prod_days** variable (see Table 2). We assumed some reasonable assumptions (defined by the INFOS company): one subcontractor cannot execute two or more tasks simultaneously, the subcontractor is always available and there is no shortage of raw materials.

The two objective functions that need to be minimized are the Total Cost (TC) and Total Production Time (TPT). The TC function is the sum of the

multiplication of the assigned quantity to a individual by price of operation for that individual operation. As for TPT, the function is the sum of the maximum days required by each sequential operation (cutting, tailoring and packaging). Since subcontractors can work simultaneously in the same operation (e.g., cutting), we consider the slowest operator time (measured in terms of number of days). Solutions that split the q_i quantities by different operators for an operation will thus contribute for a lower TPT value. The lower bound is always zero and the upper bound was set to the quantity to be produced. When needed, a repair procedure is used to convert an unfeasible solution to a feasible one, see Section 2.2.

The NSGA-II algorithm was configured with a check procedure that eliminates duplicates, making sure that the mating produces offspring that are different from themselves and the existing population regarding their design space values. A grid search was used to set the NSGA-II hyperparameters (e.g., the population size was ranged within $\{50,100,150,\dots,500\}$), assuming the HV measure as the selection criterion and a reference point of (30 days, 20,000 EUR). The best obtained values correspond to a normalized HV (when each objective is divided by the respective reference point value) of 0.71, which requires 157 seconds of execution time on an Intel Xeon processor. The selected NSGA-II setup includes: population size of 100, two-point crossover with 90%, polynomial mutation probability of 20% and total of 200 generations.

The left of Figure 2 shows the Pareto front obtained after 200 generations when considering our case study. The Pareto front contains 100 solutions, with the TPT ranging from 12 to 30 working days and TC ranging from 18,000 to 20,000 EUR. The right of Figure 2 shows the evolution of the NSGA-II algorithm, in terms of the full HV measure (y -axis) through the executed 200 generations.

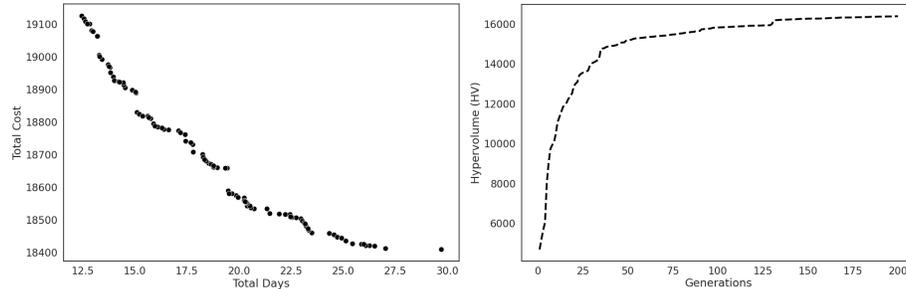


Fig. 2. Optimized Pareto front (left) and NSGA-II HV generation evolution (right).

The graph shows a substantial improvement that is obtained by NSGA-II. In effect, in the first generation the HV measure is 4,700 (normalized value of 0.2). After 200 generations, the value increased to 16,391 (normalized value of 0.71), which corresponds to an improvement of 51 percentage points when

considering the normalized HV scale. The results were shown to the INFOS company, which provided a very positive feedback. In particular, the obtained TPT and TC ranges were considered realistic. Moreover, the company signaled that the obtained Pareto front provides a more richer set of trade-off solutions, while also being faster to compute when compared with the currently adopted manual subcontractor allocation.

4 Conclusions

We propose an IDSS that creates a textile production plan to allocate subcontractors. The IDSS is based on the ABI concept that combines predictive (via ML) with prescriptive (via MO) analytics in order to provide actionable knowledge from raw data. The IDSS was designed to work with real-world data from a Portuguese software company (INFOS) that works with diverse textile clients. Firstly, an AutoML tool was adopted to automatically select the best ML model among five algorithms when targeting four relevant allocation decision context variables. Interesting results were achieved by the prediction models (error that ranges from 3.6% to 9.2%). Then, we designed a MO model that uses one of the predicted variables (production time) and historical data to automatically allocate subcontractors to execute sequential operations associated with a textile order. The MO model, based on the NSGA-II algorithm, assumes a Pareto approach and it was designed to simultaneously minimize the cost and time to execute the order. To demonstrate the MO, we selected a case study that includes four operations and 26 potential textile subcontractors.

The obtained results were shown to the INFOS company, which considered them very positive. In future work, we intend to augment the IDSS by incorporating more problem-domain constraints, such as incorporating updated data about the current availability of subcontractors. Furthermore, we wish to deploy the designed IDSS into the INFOS ERP system, in order to get more valuable feedback from a real environment usage.

Acknowledgments

This work was carried out within the project “Connect@Fashion” reference POCI-01-0247-FEDER-045296, co-funded by *Fundo Europeu de Desenvolvimento Regional* (FEDER), through Portugal 2020 (P2020).

References

1. Ángeles Solari, M.d.l., Ocampo, E.: Application of genetic algorithms to a manufacturing industry scheduling multi-agent system. In: Sobh, T., Elleithy, K., Mahmood, A., Karim, M. (eds.) *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*. pp. 263–268. Springer Netherlands, Dordrecht (2007)

2. Arnott, D., Pervan, G.: A critical analysis of decision support systems research revisited: the rise of design science. *J. Inf. Technol.* **29**(4), 269–293 (2014). <https://doi.org/10.1057/jit.2014.16>
3. Ben-Ammar, O., Bettayeb, B., Dolgui, A.: Optimization of multi-period supply planning under stochastic lead times and a dynamic demand. *International Journal of Production Economics* **218**, 106–117 (2019). <https://doi.org/https://doi.org/10.1016/j.ijpe.2019.05.003>
4. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. *IEEE Access* **8**, 89497–89509 (2020). <https://doi.org/10.1109/ACCESS.2020.2990567>
5. Campos-Ciro, G., Dugardin, F., Yalaoui, F., Kelly, R.: A nsga-ii and nsga-iii comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine* **49**(12), 1272–1277 (2016). <https://doi.org/https://doi.org/10.1016/j.ifacol.2016.07.690>, 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016
6. Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F.: Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications* **63**(5), 912–942 (2012). <https://doi.org/https://doi.org/10.1016/j.camwa.2011.11.057>
7. Cortez, P.: *Modern Optimization with R*. Springer (2021). <https://doi.org/10.1007/978-3-030-72819-9>
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
9. Ferreira, L., Pilastri, A., Martins, C.M., Pires, P.M., Cortez, P.: A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost. In: *Int. Joint Conference on Neural Networks, IJCNN 2021, July*. IEEE (2021)
10. Guo, Z., Wong, W., Li, Z., Ren, P.: Modeling and pareto optimization of multi-objective order scheduling problems in production planning. *Computers & Industrial Engineering* **64**(4), 972–986 (2013). <https://doi.org/https://doi.org/10.1016/j.cie.2013.01.006>
11. Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriach, C.: *Adaptive business intelligence*. Springer (2006)
12. Mok, P.Y., Cheung, T.Y., Wong, W.K., Leung, S.Y., Fan, J.T.: Intelligent production planning for complex garment manufacturing. *J. Intell. Manuf.* **24**(1), 133–145 (2013). <https://doi.org/10.1007/s10845-011-0548-y>
13. Mokhtari, H., Hasani, A.: An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Comput. Chem. Eng.* **104**, 339–352 (2017). <https://doi.org/10.1016/j.compchemeng.2017.05.004>
14. Tsai, W.H.: Green production planning and control for the textile industry by using mathematical programming and industry 4.0 techniques. *Energies* **11**(8), 2072 (Aug 2018). <https://doi.org/10.3390/en11082072>