Multi-objective Approach for Non-dominated Solutions in Digital Twin

Gregor Papa, Peter Korošec
Jozef Stefan Institute
Jamova cesta 39
Ljubljana Slovenia
gregor.papa@ijs.si
peter.korosec@ijs.si

ABSTRACT: *The value of Digital twins in the smart world has been realized in the recent years. In the information technology there is a strong move towards the digital twin currently. In this work we introduce an initial step to upgrade simulations to digital twins to enhance the productivity even further. The multi-objective optimisation approach is important in achieving high efficiency of production scheduling. The goal of the optimization is to end a production schedule that satisfies different, contradictory production constraints. We take a simulation tool that was used by a memetic version of the Indicator-Based Evolutionary Algorithm with customized reproduction operators and local search procedures to end a set of feasible, non-dominated solutions and analyse the required steps to achieve a digital twin. We show that with a multi-objective approach that is able to find high-quality solutions and flexibility of many "equal" solutions, the digital twin becomes a powerful tool for a decision maker.*

Keywords: Multi Objective, Scheduling, Optimisation, Real World, Digital Twin

Received: 4 May 2018, Revised 11 June 2018, Accepted 15 June 2018

1. Introduction

Since production scheduling is important for smart specialization goals in factories of the future, we decided to take relevant results from [4], and apply them to see the impact of digital twins. A digital twin is a digital copy of physical world (physical twin) in form of processes and systems. It provides both, the elements and dynamics of the real-word, so one can simulate and predict the future events with an up-to-date model, which is relevant for a decision maker.

In [4] we applied the multi-objective approach that uses specific local search procedures to the problem of production scheduling. As the basic algorithm we used the Indicator-Based Evolutionary Algorithm (IBEA) [8]. We decided to use the IBEA because it was shown that it can substantially outperform results generated by other multi-objective algorithms, such as the improved Strength Pareto Evolutionary Algorithm [9] and NSGA-II [2], in terms of different performance measures [8]. Due to the addition of local search procedures, we called our approach the Memetic Indicator- Based Evolutionary Algorithm (M-IBEA). As such it represents a synergy of the multi-objective evolutionary approach with separate, individual, learning or local improvement procedures (local searches).

If the approach would be left as is, it would be considered only as multi-objective approach using a simulation tool to find an approximation set of non-dominated solutions. But since it can be introduced into the actual production, meaning that the current information of the state of production, with regard to standing orders and orders which have already been processed so far, we can consider such an enhanced simulation model to be a digital twin of the production. With it, we could not only simulate theoretical future capacities, but also include actual production and its daily specifics to predict future events with higher accuracy.

The rest of the paper is organized as follows: in Section 2,we briefly describe the production scheduling problem; in Section 3, we introduce required changes to create a digital twin; in Section 4, we present the main idea of Memetic IBEA; in Section 5, we present the experimental environment and the results of the evaluation with the real-world data; in Section 6, we present the usability study; and in Section 7, we draw conclusions and propose possible future work.

2. Production Scheduling Problem

The scheduling problem was introduced for a company that produces components for domestic appliances, including hot plates, thermostats and heating elements. The fabrication process for components used in different types of plates is similar, but due to clients' demands the models differ in size (height, diameter), connector type, and power characteristics(wattage). For logistic reasons, the clients group different models of plates within the same order, implying the same due-dates for different products. As a consequence, their production must be scheduled very carefully to full all the demands (quantities and due-dates), to maintain the specified amounts of different models in stock, to optimally occupy their workers, and to make efficient use of all the production lines. The assignment of due-dates is usually performed separately and before the production scheduling, but since there are strong interactions between the two tasks, using the proposed digital twin can allow for more accurate arrangement of due-dates. For each order, the completion time should be as close as possible to the due-date in order to reduce the waiting time and costs [7]. Furthermore, not all the production lines are equal, since each of them can produce only a few different models. A detailed formulation of the production scheduling problem is presented in [5].

The required inputs to such a problem are:

- Production norms that specify which products are being produced on each line and what is the changeover time from one product to another for each specific line.
- Amount of stock for each product.
- Orders that need to be processed and their deadlines.
- Number of planned shifts.
- Number of lines.

Looking from the perspective of a simulation tool that is able to take into account all this inputs and evaluate the expected time of production for every order, it is a simple simulation tool. But such a tool alone lacks the dynamics of the real world, so it is not able to react "instantly" to the changes in the production environment.

3. Digital Twin

For a simulation tool to become a digital twin, some capabilities need to be added. Mainly, the interaction between what is happening in the real world and the description of the problem instance. First of all, the relevant information, which defines the problem instance, can be gathered from the company's information system. This allows receiving up-to-date information about

new orders, the current stock, and amount of products that were produced so far in the day. With the way production companies are working, usually this needs to be done only once a day, since production plans do not change for the current day (actually they are fixed for up to several days in advance), due to the requirements of having the required materials for producing orders at hand. The main reason for this is that an additional requirement is also to have the stock of materials at the factory as small as possible. We must be aware that any unnecessary stock is actually an expense that every company would like to reduce or even remove.

The simulation tool only takes into account the technical data provided by the company with regard to the above mentioned required inputs. Though any changes in production can be \detected" by the simulator through changes in inputs (e.g., how many products were actually produced), this does not provide a good baseline for predicting future production with inclusion of predicting maintenance. For prediction maintenance to be included in the digital twin a machine learning techniques should be used to estimate/model any informalities that happen, but are not included in production norms (e.g., failures on lines). All this is based on previous experiences and requires to gather lots of data, so the machine learning algorithm is able to be trained to detect abnormal, correlated patterns in production, which will lead to better predicting future production and provide insight into preventing maintenance, which will lead to further reducing of delays on production lines due to failures by applying maintenance before a defect happens.

4. Memetic IBEA

The IBEA is a multi-objective version of a genetic algorithm, where the selection process is based on quality indicators. An indicator function assigns each Pareto-set approximation a real value that reflects its quality. The optimization goal becomes the identification of a Pareto-set approximation that minimizes an indicator function. The main advantage of the indicator concept is that no additional diversity preservation mechanisms are required [1].

The detailed description of the memetic IBEA can be found in [4], but the main idea is presented as a pseudo code in Algorithm 1. In our implementation of the basic version, the IBEA is used to guide the local search procedures. Since we are dealing with a combinatorial problem, we implemented problem-specific versions of the crossover and mutation operators. Additionally, we added different local search procedures to enhance the efficiency of the algorithm.

Algorithm 1 Memetic IBEA

```
1: SetInitialPopulation(P)
2: Evaluate(P)
3: while not EndingCondition() do
4:     P' = MatingSelection(P)
5:     Crossover(P'; p_c)
6:     Mutation(P'; p_m)
7:     Evaluate(P')
8:     LocalSearch(P')
9:     P = CalculateFitness(P ∪ P')
10:    P = RemoveWorse(P)
11: end while
```

Compared to the basic version of the algorithm, the main difference is in the procedure LocalSearch(P'). Here, not only one but many problem specific local search procedures are applied [4].

Such a version of the algorithm is suitable for running a simulation based approach, but it lacks the required dynamicity to actively adapt to changes in the production environment. Two things need to change, first, the changes in the production environment should be transferred to the algorithm solution space, and second, the algorithm should be able to detect and adapt to such

changes. Since the production is not a living system that changes every second and requires immediate changes (as mentioned above, the production is fixed for several days in advance) this is not a crucial aspect, since this changes could be applied to the algorithm on a daily basis. But from the point of view of acquiring new orders and providing potential deadlines to the customers, this is another matter. By providing a more dynamic system, a product sales person could easily insert a new potential order and determine what would be the most efficient and safe deadline to be offered to the customer. And if a customer requires an earlier deadline, which could force other orders to be put in jeopardy of missing the deadline, it allows a product sales person to better estimate the required higher price for covering the costs occurred from delays of other orders. The use of machine learning would also cover the irregularities that happen in production.

5. Case Study

5.1 Test Cases

The algorithm was tested on two real order lists from the production company. Task 1 consisted of $n = 470$ orders for 189 different

	Evaluations		Time		Pareto
n	BF	M-IBEA	BF	M-IBEA	matching
7	3:94 10^8	3:5 10^4	22 s	17 s	4/4
8	1:58 10^{10}	5 10^5	15 min	33 s	5/5
9	7:09 10^{11}	5 10^6	11 h	5 min	15/15

Table 1. Comparison of the BF (12 threads) and M-IBEA approach (1 thread)

products and Task 2 consisted of $n = 393$ orders for 175 different products. The number of orders n represents the problem dimension, with $m = 5$ representing the number of available production lines.

To mimic the digital twin which is being updated with information once a day (after the end of the daily production) we ran a task overnight and looked at the results. In this time, the algorithm made about 300 million evaluations, so this was set as our stopping criterion for future tests. A lexicographic evaluation [6] was used for presenting multiobjective solutions. In the simulation evaluation, the number of delayed orders (n_{orders}) was set as the most important objective, followed by the required number of workers ($n_{workers}$), the sum of delayed days for all the delayed orders (n_{days}), and the sum of the change-over downtime in minutes (t_{change}). This order was set according to the most common objective hierarchy.

5.2 Evaluating the Approach

To make sure that our proposed M-IBEA was working well, we ran a brute-force (BF) approach where all the possible solutions were evaluated for $n < 10$ orders and the optimal Pareto front was constructed for each of them. Table 1 shows a comparison of the number of problem evaluations, the execution time, and the matching of the Pareto front obtained for $n = 7, 8, 9$. We did not include smaller n values, since in all cases a sub-one-second time was needed with perfect Pareto matching. From the obtained results it is clear that with more than nine orders, the complexity increases well beyond an acceptable time (approximately two months) to calculate all the solutions. Also, in all cases we were able to acquire the same Pareto front using the BF and M-IBEA approaches. When considering times, one must take into consideration that the BF was ran multithreaded with 12 threads fully utilized, while the M-IBEA approach was single threaded. The perfect Pareto-front matching is unsurprising, since the IBEA already proved to be one of the best algorithms for solving multi-objective problems with more than three objectives [3], which was also the main reason that we selected the IBEA as our base algorithm.

5.3 Results

In [5], we optimized only according to the number of orders. To show that the multi-objective approach presented in [4] is a better alternative, we compared the results with regard to the best result from the single-objective to the multiobjective approach. The results showed that the single objective solution primarily concentrated on the number of orders, while it neglected other objectives. But this is not a surprise, as multi-objective solutions were able to nd equally good solutions with regard to the number of orders and significantly better for other objectives, compared to single-objective solution. Though we used the same

Statistics	n_{orders}	$n_{workers}$	t_{change}	n_{days}
Pareto min	18	631	353	127
Pareto max	88	823	867	681
Single-objective	18	767	714	156

Table 2. Results of optimisation for Task 1

Statistics	n_{orders}	$n_{workers}$	t_{change}	n_{days}
Pareto min	16	538	355	59
Pareto max	50	778	433	330
Single-objective	15	702	443	155

Table 3. Results of optimisation for Task 2

number of evaluations, this single-objective solution does not stand out with respect to any objective - quite the opposite is the case. This can also be observed from Table 2, where the single-objective solution returns an average quality solution on all the objectives except norders. The results are summarised in Tables 2 and 3, where the width of the Pareto approximation front is denoted with "Pareto min" and "Pareto max".

From the results we can conclude that using the Pareto-front approach gives us an expected greater versatility in choosing a good solution, while at the same time we are not sacrificing one, likely the most important, objective. The only important drawback is that multi-objective approaches need many more evaluations than single-objective approaches. So, if we do not have time to carry out enough evaluations, then the single-objective approach is the only way.

6. Usability of Multi-objective Solutions

The multi-objective approach provides a set of feasible solutions, offering the possibility to select the final schedule based on the specific decision maker needs. Since none of the given solutions dominates the other solutions, all of them are acceptable. Based on the current conditions, and according to the proposed set of solutions, a decision maker can give more weight to some of the decision criteria. For this an intuitive representation of the resulting solutions inside the GUI of the Planer application was provided, which is presented in Figure 1.

After the M-IBEA algorithm found the set of non-dominated solutions, they are presented in the Planer application. In the upper-right section there is a list of all the non-dominated solutions. In general, there might be up to several hundred possible solutions.

However, some of the criteria can be set tighter according to the resulting range of each criterion, and according to the current business conditions. In the specific example shown in Figure 1, the initial set consisted of 518 solutions. The decision maker put the first objective into the range from 16 to 17 out of 50, which in consequence moved the sliders of the second objective from 697 to 738, the third objective from 405 to 415, and the fourth objective from 60 to 111. So irregardless of which slider is moved, the ranges move accordingly to the possible solutions of other objectives. Simultaneously, the list of possible solutions is updated to reflect the current setting of the objectives' ranges. In the above example, the list narrowed down to 14 solutions. From them, the decision maker can select one solution which best fulfil the current demands. The visual representation consists of all the relevant data, i.e., the production lines' load, the order types' distribution, and change-over downtime lengths, which are necessary to make the final decision. If the visual representation of the solution is accepted, it becomes the production schedule. By determining (using sliders), which objective is the most important in the current situation and to what extent, we automatically determine which part of Pareto front is important and at the same time disregard all the solutions from the Pareto

which do not full the selected conditions. This way we are able to freely move the useful part of the Pareto front by moving sliders.

7. Conclusion and Future Work

We presented what steps would be needed to make a memetic, multi-objective approach that used a simulation tool to asses some real-world test cases of a production scheduling problem a more dynamic system by upgrading a simulation tool to a digital twin. From the perspective of the algorithm not many changes would be required, since with a restart procedure being already implemented any changes in the problem description could be "inserted" into the problem solving part.

On the other hand, more substantial changes are required within the simulation tool. Primarily, how required inputs are being automatised (gathering data directly from the company's system). Additionally, an inclusion of some machine learning algorithm, that would be able to detect and predict failures on production lines, is foreseen for better longterm estimation of production.

For future work, we are planning to implement the proposed changes, which will enable for more real-life scenarios (including uncertainties-based worst-case scenarios), while currently only "ideal" solutions are provided, which are often not realistic.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2- 0098). This work has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 692286 (project SYNERGY) and No 722734 (under the Marie Sk lodowska-Curie action project UTOPIAE).

References

[1] Basseur, M., Burke, E. (2017). Indicator-based multi-objective local search. In Evolutionary Computation, 2007. CEC 2007. *IEEE Congress on*, p. 3100 - 3107, (September). 2007.

[2] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, editors, Parallel Problem Solving from Nature PPSN VI, 1917 of Lecture Notes in Computer Science, p. 849-858. Springer Berlin / Heidelberg, 2000.

[3] Ishibuchi, H., Tsukamoto, N., Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. *In*: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). *IEEE Congress on*, p. 2424-2431, (June).

[4] Korosec, P., Bole, U., Papa, G. (2013). A multi-objective approach to the application of real-world production scheduling. *Expert Systems with Applications*, 40 (15) 5839-5853.

[5] Papa, G., Vukasinovic, V., Korosec, P. (2012). Guided restarting local search for production planning. *Engineering Applications of Articial Intelligence*, 25 (2) 242-253.

[6] Rentmeesters, M., Tsai, W., Lin, K.-J. (1996). A theory of lexicographic multi-criteria optimization. *In*: Engineering of Complex Computer Systems, 1996. Proceedings., *Second IEEE International Conference on*, 76 -79. (October).

[7] Zhang, R., Wu, C. (2012). A hybrid local search algorithm for scheduling real-world job shops with batch-wise pending due dates. *Engineering Applications of Artificial Intelligence*, 25 (2) 209 - 221, 2012. Special Section: Local Search Algorithms for Real-World Scheduling and Planning.

[8] Zitzler, E., Knunzli, S. (2004). Indicator-based selection in multiobjective search. *In*: Proceedings 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), p. 832-842. Springer.

[9] Zitzler, E., Laumanns, M., Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *In*: K. C. Giannakoglou, D. T. Tsahalis, J. P–eriaux, K. D. Papailiou, and T. Fogarty, editors, Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, p. 95-100, Athens, Greece, 2001. *International Center for Numerical Methods in Engineering*.