

# SOLVING TWO-STAGE STOCHASTIC MILP CHEMICAL BATCH SCHEDULING PROBLEMS BY EVOLUTIONARY ALGORITHMS AND ORDINAL OPTIMIZATION

Thomas Siwczyk\*<sup>1</sup> and Sebastian Engell<sup>1</sup>

<sup>1</sup>Process Dynamics and Operations Group, Department of Biochemical and Chemical Engineering, Technische Universität Dortmund, Dortmund, Germany

## Abstract

Chemical batch scheduling is mostly solved for problems where all data is assumed to be known. While this assumption makes scheduling problems much easier to handle, it cannot be upheld in reality. A possible way to introduce uncertainties into scheduling problems is to use two-stage stochastic mixed-integer linear programming where the uncertainties are represented by a discrete set of scenarios. With an increasing number of uncertainties, the complexity of these models increases rapidly and makes it impossible to solve them in a monolithic fashion in a reasonable amount of time. In this contribution we present a new approach to solve chemical batch scheduling problems by combining a hybrid evolutionary algorithm with a scenario decomposition technique from our previous work with the ideas of Ordinal Optimization. The proposed heuristic replaces the exact MILP solution of the scenario problems by fast non-exact solutions to perform a ranking (with small errors) of different promising first stage solutions.

## Keywords

Two-stage stochastic programming, Scheduling, MILP, Heuristics, Ordinal Optimization

## Introduction

Real-world scheduling problems often have to deal with significant uncertainties about rush-orders, availability of resources and varying demands (Harjunkski et al., 2014). These uncertainties are often neglected during mathematical optimization due to the high computational complexity. One possible option to introduce uncertainty in the data into mathematical models is by using *stochastic programming* (Birge and Louveaux, 2011), where over multiple stages the uncertainty is being resolved and decisions have to be made in between. This approach is often approximated by *two-stage stochastic programming* which only distinguishes between two stages. The first-stage describes decisions that have to be made immediately and the second-stage models future decisions for which additional information will be available in the future.

A typical way to model scheduling problems is by

using *mixed-integer linear problems (MILP) monolithic formulations* (Harjunkski et al., 2014), which can be extended to two-stage stochastic mixed-integer problems (2S-MILP). Using the two-stage formulation standard solvers can be used to optimize these problems under uncertainty. However with an increasing number of uncertain parameters these problems become very hard to solve in a monolithic fashion. Therefore several algorithms were introduced which make use of special properties of this formulation. Carøe and Schultz (1999) use Lagrangian relaxation of the non-anticipativity constraints to decompose the problems into its scenarios. A hybrid evolutionary algorithm was introduced by Till et al. (2007), Garcia-Herreros et al. (2014) presented a method using an improved L-shape method for solving large 2S-MILPs from the field of supply chains.

In this contribution a new approach called EA+OO is presented for solving computationally hard 2S-MILPs for practical problems like chemical batch scheduling.

---

\* [thomas.siwczyk@bci.tu-dortmund.de](mailto:thomas.siwczyk@bci.tu-dortmund.de)

This approach extends the method of Till et al. (2007) by incorporating ideas from Ordinal Optimization (Ho et al., 2007). The new approach replaces exact solutions for the scenario problems by fast non-exact solutions while trying to maintain an overall good solution quality.

## Two-Stage Stochastic Programming

A promising approach to introduce uncertain data into mathematical optimization is 2S-MILP (Birge and Louveaux, 2011). Different evolutions of the parameters in the future are modeled in the form of a discrete set of scenarios  $\Omega = \{1, \dots, |\Omega|\}$ . The probability of a scenario  $\omega$  to materialize in the future is denoted by  $\pi_\omega$ . In 2S-MILP, distinctions are made between two types of decisions: first-stage decisions and second-stage decisions. The first-stage decisions have to be implemented *now* and have an influence on the future evolution and the future decisions. These future decisions are represented by the second-stage, which consists of multiple scenarios. The second stage decisions can be adapted to the scenarios that materialize, they can be different for each scenario. Hence, these decisions provide the option of recourse in the future. A general formulation of a 2S-MILP can be written as follows:

$$\min c^T x + \sum_{\omega=1}^{\Omega} \pi_\omega q_\omega^T y_\omega \quad (1)$$

$$\text{s.t. } Ax \leq b \quad (2)$$

$$T_\omega x + W_\omega y_\omega \leq h_\omega \quad x \in X, y_\omega \in Y, \omega = 1, \dots, \Omega. \quad (3)$$

In a MILP the domains of the decision variables  $X$  and  $Y$  may contain (bounded) continuous and discrete values. The constraints related to the first-stage are depicted by (2) and the constraints for all scenarios (second-stage) are given by (3). While (2) only contains first-stage decisions  $x \in X$ , the second-stage contains additional second-stage decisions  $y_\omega \in Y$  for all scenarios  $\omega \in \Omega$ . The parameters of the first-stage are represented by  $A$  and  $b$ , and the parameters for each separate scenario  $\omega \in \Omega$  are given by  $W_\omega$ ,  $T_\omega$  and  $h_\omega$ . The objective of these problems is to minimize the objective function (1), which consists of the costs for the first-stage decisions and the expected costs for all scenarios, with the cost vectors  $c$  for the first-stage and  $q_\omega$  for the second-stage decisions in each scenario.

When designing algorithms for 2S-MILP, it is important to take into account that not for all problems feasibility of the first-stage constraints (2) also implies that a feasible completion in the second-stage exists. A

problem has *relatively completely recourse* (Birge and Louveaux, 2011) if for each solution  $x$  which satisfies all first-stage constraints  $Ax \leq b$ , a solution  $y$  exists which fulfills the constraints (3).

For each 2S-MILP a so-called expected value problem (EVP) can be formulated. In this variant of the problem, the uncertain parameters of the second-stage  $q_\omega$ ,  $h_\omega$ ,  $T_\omega$  and  $W_\omega$  are replaced by their expected values  $\bar{h}$ ,  $\bar{T}$  and  $\bar{W}$ . The EVP is easier to solve than the corresponding 2S-MILP due to the reduction of the number of variables and the reduction from  $|\Omega|$  scenarios to *one artificial scenario*. The optimal solution of this problem does not provide the optimal solution of the original two-stage problem. Furthermore while the first-stage decisions for EVP fulfill the first-stage constraints, there might be no solutions for one or many scenarios which satisfy all constraints of the second-stage.

## Hybrid Evolutionary Algorithm for 2S-MILP

With an increasing number of uncertainties, 2S-MILP become very hard to solve in a monolithic fashion. In practice when good (not optimal) solutions are needed in a reasonable amount of computation time, heuristic approaches are useful. One such approach which combines an evolutionary algorithm (EA) and stage decomposition, was presented by Till et al. (2007) and later improved by Tometzki and Engell (2011). The problem is decomposed into a *master problem* (MASTER), which represents the first-stage problem, and into  $|\Omega|$  subproblems (SUB $_\omega$ ) which represent each scenario of the second-stage:

$$\text{(MASTER): } \min_x c^T x + \sum_{\omega=1}^{\Omega} \pi_\omega Q_\omega(x) \quad (4)$$

$$\text{s.t. } Ax \leq b, x \in X \quad (5)$$

$$\text{(SUB)}_\omega: Q_\omega(x) = \min_{y_\omega} q_\omega^T y_\omega \quad (6)$$

$$\text{s.t. } W_\omega y_\omega \leq h_\omega - T_\omega x \quad \forall \omega = 1, \dots, \Omega \quad (7)$$

$$y_\omega \in Y. \quad (8)$$

For each candidate first-stage solution  $x^*$  the value  $Q_\omega(x)$  has to be calculated by solving  $|\Omega|$  subproblems for each scenario problem separately by fixing the values of all variables  $x$  in (SUB) $_\omega$  for all  $\omega \in \Omega$  to the respective values of  $x^*$ . After solving all subproblems the objective value for one candidate solution can be calculated. However for  $x^*$  there may not be a solution to each scenario problem that fulfills the constraints of the second-stage.

The method proposed by Till et al. (2007) uses an hybrid evolutionary algorithm (HEA) to search for good solutions for (MASTER), while an exact solver for MILP (e. g. CPLEX) is used to solve the subproblems (SUB) $_{\omega}$  for each candidate solution. An EA is a gradient-free search method which utilizes ideas from the theory of evolution (Bäck et al., 1997). The first-stage decisions are in this case regarded as the individuals and the objective function of (MASTER) is utilized as the fitness function. Hence, the fitness evaluation for each individual consists of fixing the values of the first-stage solutions to the values represented by an individual and then solving all subproblems (SUB) $_{\omega}$  separately for the individuals. Because an EA is a metaheuristic for unconstrained optimization problems, a penalty function which measures the violation of constraints is introduced to tackle the constrained problem (MASTER).

### New Approach: EA+OO

The new EA+OO approach combines the HEA for 2S-MILP from Till et al. (2007) with the main aspects of *Ordinal Optimization* (OO). OO (Ho et al., 2007) is an optimization method from the field of stochastic simulation which can be also applied to deterministic problems where the challenge is not the time that is needed for simulations but the time spent to perform complex calculations. OO is based upon two principles: “*Order* is easier than *Value*” and “Nothing but the best is very costly”. The first aspect of OO describes the fact that it is easier to decide whether a solution  $A$  has a better performance than a solution  $B$ , than it is to determine the exact performance of both solutions. The second principle proposes that it might be sufficient to search for *good enough* solutions instead of searching for a single optimal one. By combining these two principles, the computational time to find adequate solutions for practical problems can be decreased significantly. Hence, the focus of an OO-inspired optimization method shifts from searching for the single optimal solution to obtaining a set of solutions which are obtained by using a simplified model or or approximate solution. The ranking of the solutions in this set will in some cases be erroneous, but the set as a whole can be robust against perturbations and contains one of the best solutions with a high probability (Ho et al., 2007).

The main idea of EA+OO is to replace the costly *exact evaluations*  $f_{ex}$  using an exact method to find the

optimal solutions of all subproblems (SUB) $_{\omega}$  in the HEA by *heuristic evaluations*  $f_{heu}$  which use a heuristic optimization method or a simplification of the problem to solve the subproblems. A heuristic optimization method does not necessarily find the optimal solution but is in general faster than the exact evaluation and provides a performance indicator for the individuals (first stage variables). From a practical point of view, only a solution for the first-stage of the problem will be implemented, while the solutions for the subproblems of the second-stage are only indicators for the quality of a first-stage solution, exact solutions have to be found in the future.

In this contribution two different heuristic evaluation methods are proposed. The first method  $f_{heu}^{LP}$  uses a LP-relaxation for the second stage, allowing continuous values for all variables  $y_{\omega}$  of the second-stage. The second method  $f_{heu}^{EV}$  uses the EVP to measure the fitness of an individual by fixing all first-stage decisions and using a MILP-solver to obtain the objective value for the EVP problem. However, the first tests using the EVP solution as a performance measure showed a very bad overall performance of the HEA due to providing too many individuals without a feasible completion in the second-stage when evaluated exactly, while the rankings of feasible first-stage solutions showed very small errors. Therefore this evaluation method is extended by a test for feasibility. The *feasibility pump* (FP) heuristic (Bertacco et al., 2007) is used to quickly test whether an individual induced a feasible solution for both stages of the original problem. The feasibility pump is a fast method which tests whether a MILP is feasible or not, but does not provide good solutions with respect to the objective value for the problem. If an individual  $x'$  induces a feasible solution for the original problem and of the EVP the objective of the EVP  $f^{EV}(x')$  is used as the fitness value for this individual. If  $x'$  represents a feasible solution only for the EVP but not for the original problem, the fitness value of  $x'$  is set to  $f_{heu}^{EV}(x') = f_{max} + f^{EV}(x')$  where  $f_{max}$  is an upper bound of the cost function and  $f_{heu}^{EV}(x') = 2 \cdot f_{max} + \sum_i \max 0, A^i$  with  $A^i x - b^i$  being the  $i$ th constraint of (MASTER) if  $x'$  neither induces a feasible first-stage solution for the original problem nor for the EVP.

In the context of the HEA the purpose of the evaluation of the first stage solutions is to obtain a fitness value for a given individual, which is then used to compare two different solutions and creates a ranking of solutions. While using a heuristic evaluation method to obtain the order (with a small error) of different solutions fulfills

the first principle of OO, the second principle is realized by searching for a set of top- $s$  solutions, where  $s$  is a problem-specific value. It is assumed that at least one of these  $s$  solutions is of high quality, when evaluated exactly.

Heuristic evaluations might evaluate feasible solutions as infeasible and vice versa. To deal with this problem the following procedure is suggested: At first the HEA proposed by Till et al. (2007) is executed using  $f_{heu}$  and all feasible individuals are saved and ranked according to their performance as measured by the heuristic method. Afterwards the individuals are re-evaluated according to the calculated ranking in an ascending order until  $s$  feasible solutions have been found. This is based on the assumption that only few re-evaluations are necessary to find  $s$  feasible individuals and that at least one of these individuals has a reasonably good solution quality. By performing re-evaluations the real performance of an individual is also calculated and the error of the ranking is reduced.

### Case-Study: Scheduling of an EPS-Plant

We tested the proposed method EA+OO using a real-world application from the polymer industry. This scheduling problem concerns a plant for expandable polystyrene (EPS) and was used before as a test case for the HEA (Till et al., 2007; Tometzki and Engell, 2011).

The plant can produce two different types of EPS ( $p \in \{A, B\}$ ) in five different grain size fractions  $f_p \in \{1, \dots, 5\}$  in a batch-wise fashion. The production of the batches consists of three stages: preparation, polymerization and finishing stage. The preparation stage is not restricting the production schedule, hence it is not considered in the problem formulation. Each produced batch contains one type of EPS, which in the finishing stage is divided into different grain size fractions. The distribution of the grain size fractions is controlled by a recipe  $r_p$  from a discrete set of recipes  $\{1, \dots, 5\}$ . The production of one batch of polymer is assumed to take the same amount of time, regardless of the choice of the recipe. The schedule is created for several discrete periods  $i \in I = \{1, \dots, i_{\max}\}$ . The variables  $N_{i,r_p}$  denote the number of batches that are produced in period  $i$  using recipe  $r_p$ . For  $i \in I_1 = \{1, \dots, i_{\max}^1\}$  these variables represent the first-stage decisions that have to be made immediately. The capacity of the plant is limited due to the polymerization stage, hence it can only produce a maximum number of batches  $N_i^{\max}$  in each

period  $i \in I$ :

$$\sum_p \sum_{r_p} N_{i,r_p} \leq N_i^{\max} \quad \forall i. \quad (9)$$

The plant has two finishing lines for each type of EPS. Each of them can only be operated when a minimal amount of batches  $F_p^{\min}$  is being transferred to them in each period. If the number of transferred batches drops below a threshold, the finishing line has to be shut down for at least two periods. After a restart it has to stay operational for at least two more periods. In addition, the capacity of the finishing line is limited to  $F_p^{\max}$  batches per period. In each period the availability of a finishing line is indicated by the binary variable  $z_{i,p}$ . These restrictions can be formulated as follows:

$$F_p^{\min} z_{i,p} \leq \sum_{r_p} N_{i,r_p} \leq F_p^{\max} z_{i,p} \quad \forall i, p. \quad (10)$$

The switching of the finishing line is modeled by the following constraints, where  $z_p^0$  represents the initial state of the finishing lines:

$$\left. \begin{array}{ll} z_p^0 & \text{if } i = 1 \\ z_{i-1,p} & \text{else} \end{array} \right\} - z_{i,p} - z_{i+1,p} \leq 0 \quad \forall i, p \quad (11)$$

$$\left. \begin{array}{ll} z_p^0 & \text{if } i = 1 \\ -z_{i-1,p} & \text{else} \end{array} \right\} + z_{i,p} - z_{i+1,p} \leq 0 \quad \forall i, p.. \quad (12)$$

These constraints force the value of  $z_{i+1,p}$  to be 0 if a finishing line was shut down in period  $i$  and force the value to be 1 in case of a start up in period  $i$ . The variables  $w_{i,p}$  are used to record changes of the state of the finishing line. They are used in the objective function to model the costs of startups and shutdowns of finishing lines according to  $z_{i,p}$ :

$$\left. \begin{array}{ll} z_p^0 & \text{if } i = 1 \\ z_{i-1,p} & \text{else} \end{array} \right\} - z_{i,p} \leq w_{i,p} \quad \forall i, p \quad (13)$$

$$\left. \begin{array}{ll} -z_p^0 & \text{if } i = 1 \\ -z_{i-1,p} & \text{else} \end{array} \right\} + z_{i,p} \leq w_{i,p} \quad \forall i, p. \quad (14)$$

Besides the costs for switching on and off finishing lines, the objective function also contains other costs, revenues and penalties. To keep track of these values additional variables and constraints are introduced.  $M_{l,i,f_p}$  denotes the amount of a product  $f_p$  that is sold in period  $i$  with a lateness of  $l$ . The parameter  $B_{i,f_p}$  denotes the total demand of product  $f_p$  in period  $i$  and the amount of demands that could not be satisfied after  $L^{\max}$  periods and will be penalized. These unsatisfied demands are modeled by the variable  $B_{i,f_p}^-$ . It is also possible to

store batches that could not be sold in the same period they were produced. The amount of stored products  $f_p$  is denoted by the variable  $M_{i,f_p}^+$  and the initial storage is given by the parameter  $M_{f_p}^0$ . The development of the production and the sales is represented by the constraint (15) and the balance between the sales and the demands is represented by (16):

$$\sum_{l|i+l-1 \leq i_{\max}} M_{l,i+l-1,f_p} = B_{i,f_p} - B_{i,f_p}^- \quad \forall i, f_p, p \quad (15)$$

$$\sum_{j=1}^i \sum_l M_{l,j,f_p} + M_{i,f_p}^+ \quad (16)$$

$$= \sum_{j=1}^i \sum_{r_p} \rho_{f_p,r_p} N_{j,r_p} + M_{i,f_p}^+ \quad \forall i, f_p, p \quad (17)$$

where  $\rho_{f_p,r_p}$  denotes the yield of the grain size fraction  $f_p$  of recipe  $r_p$ .

The optimization task is to maximize the profit of the plant, it is modeled as a minimization problem. The objective function (18) contains the profit from revenues  $\alpha_{l,i,f_p}$  per batch for satisfying customer demands, penalties for supply shortage  $\alpha_{i,f_p}^-$  per unit of product, costs for storing products  $\alpha_{i,f_p}^+$  per batch and the operating costs for stage changes of the finishing lines  $\gamma_{i,p}$  and the fixed production costs  $\beta_{i,r_p}$  per batch:

$$\min - \sum_{i,p} \left( \sum_{l,f_p} \alpha_{l,i,f_p}^- M_{l,i,f_p} - \sum_{f_p} \alpha_{i,f_p}^+ M_{i,f_p}^+ - \quad (18)$$

$$\sum_{f_p} \alpha_{i,f_p}^- B_{i,f_p}^- - \sum_{r_p} \beta_{i,r_p} N_{i,r_p} - \gamma_{i,p} \right). \quad (19)$$

The demands and the maximum capacity of the plant are uncertain. The different evolutions of the demands are denoted by the parameter  $B_{i,f_p}^{\eta_1}$  and the different evolutions of the maximum capacities are denoted by  $N^{\max,\eta_2}$  where  $\eta_1 = 1, \dots, \#\eta_1$  and  $\eta_2 = 1, \dots, \#\eta_2$  denote single realizations. Hence  $\#\eta_1 \cdot \#\eta_2$  scenarios are considered in the two-stage model.

## Experimental Results

We used the EPS case-study to test the performance of the EA+OO method in comparison to the original HEA which uses the exact evaluation method  $f_{ex}$ . In Tometzki and Engell (2011) it was shown that the HEA provides very good results for the EPS problem. The goal of the experimental evaluation is to investigate whether comparable results can be achieved with less computational effort using only heuristic evaluations. Another question, which is of interest, is how many exact re-evaluations are necessary to find the best solution of one run.

The experiments were performed using the same HEA configuration that was used by Tometzki and Engell (2011), only the calculation of the fitness function was changed to the proposed methods  $f_{heu}^{LP}$  and  $f_{heu}^{EV}$ . The methods were implemented using the Java 8 programming language. The solver used for the MILP- and LP-problems was IBM ILOG CPLEX 12.6.0. For  $f_{heu}^{EV}$  the feasibility pump function implemented in CPLEX was used to test for feasibility. The calculations were performed using a computer with an Intel(R) Xeon processor at 2.50 GHz with eight cores and 64 GB RAM. We tested four parameter sets with 256 scenarios and four instances with 512 scenarios, using 100 generations as the termination criterion and also a time-limit of 7.5 minutes for the first four and 15 minutes for the second four instances. Evolutionary algorithms are stochastic optimization methods, hence each optimization was repeated 20 times. The results can be seen in Figure 1. All individuals that were generated were re-evaluated by  $f_{ex}$ . It can be observed that the heuristic methods  $f_{heu}^{LP}$  and  $f_{heu}^{EV}$  can find solutions of even better quality (after the re-evaluation with  $f_{ex}$ ) than the original HEA using  $f_{ex}$ . Although  $f_{heu}^{LP}$  is 20 times faster than  $f_{ex}$  when applied to an individual which induces a feasible solution, the total time to complete 100 generations was comparable due to the higher number of individuals which are only feasible when  $f_{heu}^{LP}$  is used as an evaluation method. The evaluation method  $f_{heu}^{EV}$  needed only 1/3 of the time to complete 100 generations and was much faster than  $f_{heu}^{LP}$  and  $f_{ex}$ . Using the *Wilcoxon rank sum test* (Derrac et al., 2011) it could be shown that the EA+OO approach using  $f_{heu}^{EV}$  performed significantly better (with a level of significance of  $\alpha = 0.01$ ) than the original approach. While the performance using  $f_{heu}^{EV}$  is significantly better than using only  $f_{ex}$ , it cannot be determined which of the heuristic evaluation methods performs better over all cases. A reason for the better performance of the OO-based approach might be the difference in the calculation of the fitness value in the case of an individual which induces a solution which fulfills all first-stage constraints but violates constraints of the second-stage. For this group of individuals the original approach uses a fixed value as a fitness value, while  $f_{heu}^{EV}$  and  $f_{heu}^{LP}$  calculate different values for these individuals. Due to the LP-relaxation,  $f_{heu}^{LP}$  provides different fitness values for individuals that are not feasible when being evaluated with  $f_{ex}$ . In the case of  $f_{heu}^{EV}$  the value of the EVP, which is added to an upper bound of the cost function, is used to indicate

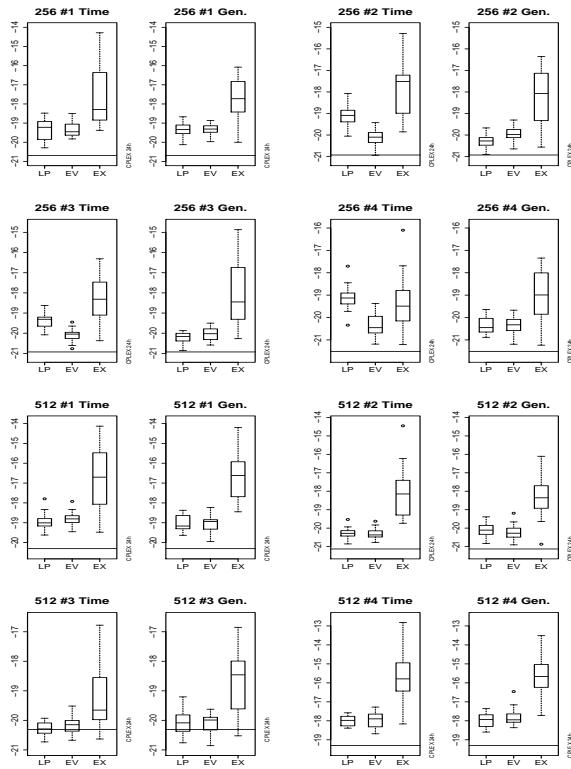


Figure 1. Results for all instances

the performance of an individual. These aspects seem to improve the overall optimization process.

All candidate individuals were re-evaluated to research how many re-evaluations were necessary to find the best solution of the optimization run. It could be shown that for  $f_{heu}^{LP}$  in over 75% of the cases it was enough to find the first feasible solution to find the best solution of the entire run and in over 90% of all runs 200 re-evaluations or less were necessary to find the best individual of the respective run, which is equal to the amount of evaluations during four generations of the HEA. In the case of  $f_{heu}^{EV}$  in over 95% of the runs it was sufficient to re-evaluate up to ten individuals to find the best individual of a run.

## Conclusions

This contribution proposed a modification to the HEA (Till et al., 2007; Tometzki and Engell, 2011) for 2S-MILP based on the ideas of OO. The new approach EA+OO was tested using a real-world case-study from the field of chemical batch scheduling. The proposed method replaces computationally expensive exact evaluations of the second-stage solutions by approximate evaluations and provided significantly better results for the EPS problem than the original approach in all in-

stances.

## Acknowledgments

This research was funded by DFG in the context of the RTG 1855: Discrete Optimization of Technical Systems under Uncertainty at TU Dortmund.

## References

- Bäck, T., Fogel, D. B., and Michalewicz, Z., editors (1997). *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK, 1st edition.
- Bertacco, L., Fischetti, M., and Lodi, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76.
- Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Series in Operations Research and Financial Engineering. Springer Science, New York.
- Carøe, C. C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45.
- Derrac, J., García, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.
- Garcia-Herreros, P., Wassick, J. M., and Grossmann, I. E. (2014). Design of resilient supply chains with risk of facility disruptions. *Industrial & Engineering Chemistry Research*, 53(44):17240–17251.
- Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., and Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62:161–193.
- Ho, Y.-C., Zhao, Q.-c., and Jia, Q.-S. (2007). *Ordinal optimization: Soft optimization for hard problems*. Springer US, New York, NY.
- Till, J., Sand, G., Urselmann, M., and Engell, S. (2007). A hybrid evolutionary algorithm for solving two-stage stochastic integer programs in chemical batch scheduling. *Computers & Chemical Engineering*, 31(56):630 – 647.
- Tometzki, T. and Engell, S. (2011). Systematic initialization techniques for hybrid evolutionary algorithms for solving two-stage stochastic mixed-integer programs. *IEEE Transactions on Evolutionary Computation*, 15(2):196–214.