Baltic J. Modern Computing, Vol. 2 (2014), No. 1, 1-15

Deterministic methods for a production effectiveness navigator

Jolita BERNATAVIČIENĖ[1], Olga KURASOVA[1],
Virginijus MARCINKEVIČIUS[1], Viktor MEDVEDEV[1],
Piotr PODGÓRNI[2] and Povilas TREIGYS[1]

[1]Vilnius University, Institute of Mathematics and Informatics,
Akademijos St. 4, LT 08663, Vilnius, Lithuania
[2]IT Vision, Chełmońskiego St.166, 31-348 Kraków, Poland

jolita.bernataviciene@mii.vu.lt, olga.kurasova@mii.vu.lt,
virginijus.marcinkevicius@mii.vu.lt, viktor.medvedev@mii.vu.lt,
piotr.podgorni@itvision.pl, povilas.treigys@mii.vu.lt

Abstract: In the paper, we investigate deterministic approaches for solving scheduling and rescheduling problems. Most of them are based on mixed-integer programming. Descriptions of solvers are also presented here. They allow solving complicated optimization problems with many variables and constraints in acceptable time. The deterministic approaches could be implemented in a production effectiveness navigator under development, one of the aims of which is to react to unexpected events occurring in shop-floors. Thus, rescheduling is necessary and optimal decisions must be made invoking the state-of-the-art optimization methods. Moreover, the management focus should be taken into account, when not only the initial schedule, but also a new schedule is created.

Keywords: deterministic algorithms, scheduling, rescheduling, mixed-integer programming, management focus.

Introduction

Nowadays, the majority of business companies use Enterprise Resource Planning (ERP) software to store and manage data from every stage of business. In many ERP systems, modules are implemented for financial and management accounting, human resource management, manufacturing, supply chain, project and customer relationship managements. In each step, optimal solutions should be found. We deal with complex optimization problems with many variables and constraints. Scheduling is one of the important issues of the manufacturing process that has a major impact on the productivity of the process (Baker and Trietsch, 2009). However, usually schedules are of limited duration and scheduling is a continuous process of responding to unexpected events, for example, machine break down, illness of employees, lacking material, etc. Thus, rescheduling is often necessary (Vieira et al., 2003). Optimization problems should be solved for both scheduling and rescheduling. Various optimization techniques are implemented in the ERP software. Moreover, the Advanced Planning and Scheduling (APS) tools have been integrated into the ERP systems: Preactor[1], Talika PMS, CyberPlan[2], AIMMS[3], Asprova APS[4], etc. Recently, a new framework, called Production Effectiveness Navigator (PEN), has been designed for advanced planning and scheduling (Čaplinskas et al., 2012). Heuristic and meta-heuristic approaches are often applied to solve scheduling and rescheduling problems (Ziaee, 2014), (Pérez-González et al., 2009), (Avci et al., 2003). The approaches are flexible, but the solving way is not grounded mathematically, and solutions can be only near to optimal. However, complicated optimization problems can be solved in an acceptable time. Another type of optimization methods is deterministic. Deterministic methods take into account analytical properties of the problem solved. The majority of methods allow us to find global optimal solutions. Recently, the increase of computational resources has enabled deterministic approaches to solve complex optimization problems, such as scheduling and rescheduling (Roslöf et al., 2001), (Floudas and Lin, 2005). The goal of the paper is to overview the deterministic methods suitable for implementation in the production effectiveness navigator for rescheduling and resource planning.

1. Framework of production effectiveness navigator

Manufacturing planning, scheduling, and control are a large area that involves different approaches on different planning levels. Various advanced planning and scheduling systems emphasize different planning aspects, problems, methods, and solutions. APS is a computer program that uses advanced mathematical algorithms for optimization, and/or simulation of finite capacity scheduling, sourcing, resource planning and so on. Commercial APS systems involve a number of software modules that support the planning tasks at different planning levels and in different supply chain processes (Ivert, 2009).

The production effectiveness navigator is an advanced production planning and scheduling system that is implemented as an add-on in enterprise recourse planning systems. This novel predictive-reactive planning and scheduling system will be used for the following aims (Čaplinskas et al., 2012):

- to evaluate the severity of potential impacts of predictable exceptions on medium term level production plans and on short-time schedules generated by a ERP system;
- to rework these plans and schedules so that the risk of predictable exceptions could be mitigated, i.e., either their likelihood can be minimized or the potential severity of their consequences (their impact) can be reduced.

The PEN system is different from the other similar products by the following aspects: a) it should provide specific rule-based mechanisms allowing us to take into account the knowledge about the specificity of the production system of a particular

[1] http://www.preactor.com
[2] http://www.cybertec.it/en/cyberplan.html
[3] http://business.aimms.com
[4] http://www.asprova.com/

target enterprise in the production rescheduling process; b) it should offer a special sandbox functionality for experimenting scenarios of special events, risks, or failures; c) it should provide integrated handling of business goals and management focuses during the optimization of new schedules.

The PEN system combines various simulation techniques, expanded by optimization procedures. The PEN system focuses on the production scheduling and rescheduling module, and, especially, on processing of undesirable business events which disrupt the schedule. The module provides integrated handling of business goals and management focuses during the optimization of the new schedules.

Deterministic algorithms presented below will be used in PEN for rescheduling shop-floor level schedules. These schedules are created taking into consideration a fixed set of undesirable events. An event is undesirable facts or circumstances which disturbs the normal implementation of the actual running schedule. An event is described by an event type, time stamp, impact estimation (in hours), related processes, resources and materials. All undesirable events are registered in the queue of events, the events are transformed into problems, caused by these events, and the actual schedule is reworked so that all the problems to be solved if possible.

In the scope of PEN, the events should be handled only those have a 48 hour-long impact on processes, at least. All the events are classified, based on the place of occurrence in the process or on their complexity:

- undesirable events of resources;
- missing technology resources or the infrastructure problem;
- problems with production time;
- reworking;
- undesirable events of orders (cancelling process);
- problems in the supply chain management process.

To reschedule a shop-floor level schedule, data on the following *business objects* are necessary: schedule, event, job, resource, machine, fixture/mould, operation, working place, staff worker, vehicle, process, material, supplier, infrastructure resource, logical factory map, and subcontractor.

The main object is a schedule in Fig. 1. There the represented objects are interrelated with each other. The objects (management focus, budget, customer’s order and logical factory map), represented by orange blocks, influence the objective function of deterministic algorithms. The dotted arrays mark dependences of the objects, eg., a schedule depends on the management focus, budget, customer order; an operation depends on resources and subcontractor. The solid arrays mark the components of the objects, eg., resources consist of a machine, material, human resources, vehicle, supplier, subcontractor, and infrastructure.

The decision on the management focus is made by assigning weights to the variables of the objective function when solving rescheduling problems. The weights can be used to achieve the following management goals:

- to keep a deadline;
- to keep the number of subcontractors as low as possible;
- to keep the budget (cost);
- to keep the number of workers, involved in the production, as low as possible;
- a specific goal for a target enterprise.

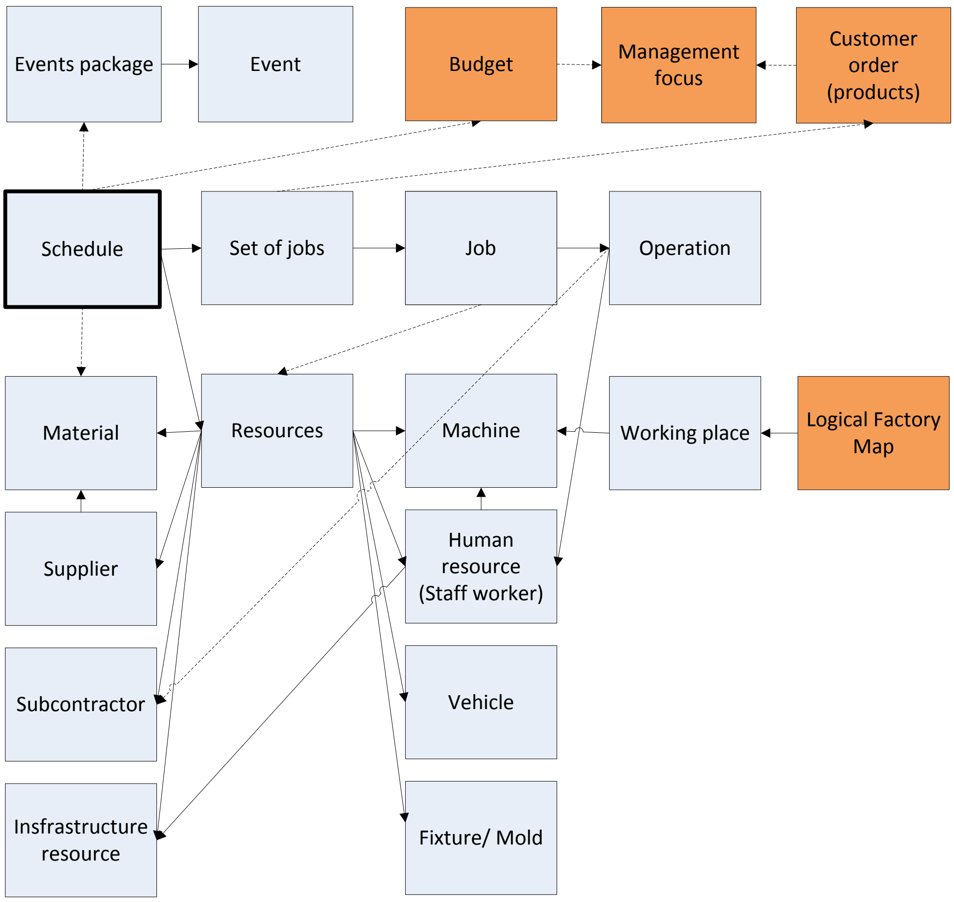

Fig. 1. Dependencies between PEN objects.

In the PEN system a composition of various deterministic or heuristic optimization methods is used for rescheduling. The output of the PEN system is a set of plans for enterprise management that are in accordance with that that prepared by ERP.

1.1. Optimization in the production effectiveness navigator

The PEN system is focused on the production program, production scheduling and rescheduling module, and, especially, on processing of unexpected business events which disrupt this program. In this paper, we focus only on optimization problems that arise when we reschedule a final assembly schedule or master production schedule, as well as on the implementation of management focus constraints in these schedules.

A particularity of PEN, in the sense of optimization, is that the management focus must be directed to customers, contracts and vendors/suppliers, when the objective functions are formulated. The management focus means that we set the preferences to suppliers, contracts (which one is more preferable) in the PEN system interface. The preferences should be stored as priorities, for example, from 20 (highest) to 1 (lowest). All the data of the enterprise management focus on these objects are defined in the form of table priorities. Deterministic algorithms will use the information on management to create new plans, scheduling and rescheduling in the future.

When we deal with the rescheduling problem in shop-floor level, management focus can be applied to the objective function as weights. The literature on scheduling theory (Pinedo, 2012), (Akturk and Ilhan, 2011), (Bülbül and Kaminsky, 2013) is related to regular objective functions such as the total flow time, number of tardy jobs, total tardiness. Suppose, the objective function is the total tardiness

$$T = \sum_j \alpha_j \max\left(0,\ C_j - d_j\right).$$

Here C_j is the completion time, i.e., the time at which the processing of job j is finished, d_j is the due date, i.e., the time at which the processing of job j is due to be completed. In this equation, management preferences are used to redefine the weight coefficients α_j (the unit time earliness costs for job j). From the PEN system, we derive management focus preferences for contracts, then we can apply some priorities to jobs (activities) of this contract (pr_j). Therefore, in the objective function, for example total tardiness, we replace α_j by $\alpha_j = pr_j \alpha_j$.

The Just-In-Time (JIT) scheduling environment, a job completed early must be held in inventory until its due date. Therefore, an ideal schedule S is that in which all the jobs terminate exactly on their assigned due dates

$$f(S) = \sum_j \left(\alpha_j \max\left(0, C_j - d_j\right) + \beta_j \max\left(0, d_j - C_j\right)\right).$$

Here α_j and β_j are weight coefficients. The same logic is applied to the objective function of other deterministic algorithms (eg., job flow time, average number of jobs, makespan) that are used in the PEN system.

2. Deterministic approaches for scheduling and rescheduling

Various deterministic approaches are used in solving scheduling and rescheduling problems. The well-known approaches, based on mathematical programming, are as follows: linear programming (LP), nonlinear programming (NLP), integer programming (IP), mixed-integer linear programming (MILP), mixed-integer nonlinear programming (MINLP), mixed-integer quadratic programming (MIQP), and mixed-integer signomial programming (MISP). The programming cases for scheduling and rescheduling, depending on the types of variables, are presented in Fig. 2.

The classification of optimization problems, investigated by practitioners and researchers in the last few decades, is presented and some deterministic optimization approaches are investigated in the paper (Lin and Tsai, 2012). The paper (Floudas and Gounaris, 2009) also presents an overview of the research progress in the deterministic global optimization during the last decade. When solving the optimization problems by deterministic methods, the problems are characterized by the convexity of the objective function or the feasible domain. In that case, continuous and/or discrete (integer) variables are involved. The convexity of the objective function or the feasible domain is very important. Some efficient numerical methods are developed to solve the optimization problem, if the objective function or the feasible domain is convex. However, optimization problems often include nonconvex functions. In this case, the standard local optimization methods do not guarantee the global optimality efficiently. When solving nonconvex or large-scale optimization problems, deterministic methods are confronted with difficulties in searching for an optimal solution during a reasonable time due to the high complexity of the problem (Lin and Tsai, 2012).

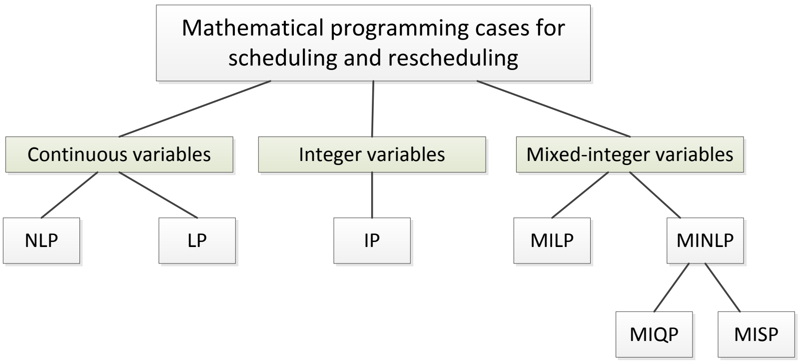

Fig. 2. Mathematical programming cases for scheduling and rescheduling depending on the types of variables.

This section presents short descriptions of mathematical programming, mixed-integer linear and nonlinear programming, mixed-integer signomial programming, constraint programing as well as some algorithms, used for scheduling and rescheduling.

Mathematical programming is used in many fields, where optimal solutions have to be found. The term "programming" has been in use since 1940 with a view to describe the planning or scheduling of activities in a large organization. If we express the amount or level of each activity as a variable, the value of which has to be determined, then we can mathematically describe the restrictions inherent in the planning or scheduling problem, as a set of equations or inequalities that involve the variables. The solution of all the constraints would be considered as an acceptable plan or schedule.

The sequence of events for finding the solution is as follows (Fourer et al., 2003):

- Formulate a model, an abstract system of variables, objectives, and constraints that represent the general form of the problem to be solved.
- Collect data that define a specific problem instance.
- Generate a specific objective function and constraint equations from the model and data.
- Solve the problem instance by running a program, or a solver, and applying an algorithm that finds optimal values of variables.
- Analyse the results.
- Refine the model and data, and repeat the process.

We can write a compact description of the general form of the problem, which we call a model, using the algebraic notation for the objective and constraints. Components of the model are (Fourer et al., 2003):

- Sets (eg., products).
- Parameters (eg., production and profit rates).
- Variables, the values of which should be determined by a solver.
- An objective, to be maximized or minimized.

- Constraints for the solution to be satisfied.

The model describes an infinite number of related optimization problems. If we provide specific values for data, the model becomes a specific problem, or the instance of the model that can be solved.

The general form of mathematical programming (in the minimization case) is expressed by:

$$\min_{x \in X} f(x),$$
$$\text{s.t. } g(x) \le 0,$$
$$h(x) = 0.$$

Here a feasible set $X \subseteq \mathbb{R}^n$; $x \in X$ is a variable vector; the function $f: X \to \mathbb{R}$ is an objective function; the functions $g: X \to \mathbb{R}^m$ and $h: X \to \mathbb{R}^k$ are constraints (Greenberg, 2010). The minimization problem can be replaced by maximization. Depending on the types of variables, properties of the objective function, and constraints, the general form of mathematical programming can be rearranged.

In the case of linear programming (LP), the objective function and constraints are linear functions. Otherwise, if some of the constraints or objective functions are nonlinear, we deal with nonlinear programming (NLP). If all the variables x_i $(i = 1, \ldots, n)$ are integer, the programming is called an integer linear programming (ILP, IP). The study of integer programming formulations of scheduling problems is presented in (Pan, 1997). A linear programming-based method for job shop scheduling is investigated in (Bülbül and Kaminsky, 2013).

In solving scheduling and rescheduling problems, at first, an objective function and constraints are constructed. Later, proper solvers are selected and the problems are solved. The solution obtained is a new schedule. Usually, a mathematical model includes the following goals: makespan, total weighted tardiness, mean flow time, number of tardy jobs, maximum tardiness, total workload of the machines, etc. (Özgüven et. al., 2012).

2.1. Mixed-integer linear programming

Frequently, there are both continuous and integer variables in scheduling and rescheduling problems, so the problems are solved by mixed-integer linear programming (MILP). Sometimes, it is called mixed-integer programming (MIP). The problem (in the minimization case) is formulated as follows:

$$\min_{x \in X} \mathrm{f}^T x,$$
$$\text{s.t. } Ax \le b \text{ (linear inequality constraints)},$$
$$A_{eq}x = b_{eq} \text{ (linear equality constraints)},$$
$$l_b \le x \le u_b \text{ (bound constraints)},$$
$$x_i \in \mathbb{Z} \text{ (some or all } x_i \text{ must take integer values, i.e. integer constraints)}.$$

Here f is an $\mathrm{n} \times 1$ vector containing the linear objective function; x is an $\mathrm{n} \times 1$ vector of decision variables; A is an $\mathrm{m} \times \mathrm{n}$ matrix, b is an $\mathrm{m} \times 1$ vector; A_{eq} is a $\mathrm{k} \times \mathrm{n}$ matrix, b is a $\mathrm{k} \times 1$ vector; l_b and u_b are $\mathrm{n} \times 1$ vectors.

MILP is used in the contexts, where it only makes sense to take integral quantities of certain goods or resources (eg., the number of men or power stations) or binary decisions need to be made (eg., producing a product, assigning a task to a worker).

The inclusion of integer variables enormously increases not only the modelling power, but also the cost of computations. IP is an NP-complete problem, so even small problems may be hard to solve. A very similar method to IP is linear programming (LP). In linear programming is no need to satisfy the $x_i \in \mathbb{Z}$ condition. LP can be solved in a polynomial time by interior-point methods (ellipsoid method, Karmarkar's algorithm (Karmarkar, 1984)). The simplex algorithm is another popular algorithm for linear programming. Usually specialized network algorithms are more efficient for these problems than simplex techniques. But the simplex techniques are general and can be used even if no implementation of network algorithms is available.

In the paper (Sawik, 2011), the scheduling in supply chains is investigated using mixed-integer programming. Simulation-based optimization versus mathematical programming is analysed in (Klemmt et al., 2009). In (Floudas and Lin, 2005), a review of the progress of MILP approaches for short-term scheduling systems has been made. In (Moreno and Montagna, 2009), the MILP model has been proposed to simultaneously optimize production planning and design decisions.

2.2. Mixed-integer nonlinear programming

Mixed-integer nonlinear programming (MINLP) problems, involving not only discrete, but also continuous variables, arise in many applications of engineering design, chemical engineering, operations research, and management. A review of the recent advances in the MINLP optimization of planning and design problems in the process industry is presented in (Kallrath, 2005).

Mixed-integer nonlinear programming (MINLP) is often used for solving scheduling and rescheduling problems:

$$\begin{gathered}\min_{x \in X} f(x),\\ \text{s.t. } Ax \leq b,\\ A_{eq}x = b_{eq},\\ l_b \leq x \leq u_b,\\ c\ \ x\ \ \leq d,\\ c_{eq}\ \ x\ \ = d_{eq},\\ x_i \in \mathbb{Z}.\end{gathered}$$

Here f is a nonlinear objective function; x is an $\mathrm{n} \times 1$ vector of decision variables; A is an $\mathrm{m} \times \mathrm{n}$ matrix, b is an $\mathrm{m} \times 1$ vector; A_{eq} is a $\mathrm{k} \times \mathrm{n}$ matrix, b_{eq} is a $\mathrm{k} \times 1$ vector; l_b and u_b are $\mathrm{n} \times 1$ vectors; c is a $u \times 1$ vector of functions containing nonlinear inequality constraints, d is a $u \times 1$ vector; c_{eq} is a $v \times 1$ vector of functions containing nonlinear inequality constraints, and d_{eq} is a $v \times 1$ vector.

In the paper (Akturk and Ilhan, 2011), scheduling with controllable processing times to minimize the total weighted tardiness is investigated, where the optimization problem is formulated as a mixed-integer nonlinear program. Newmann et al. (2002) introduced a mixed-integer nonlinear programming model for an advanced planning system. In the paper, the scheduling problem was formulated as a nonlinear mixed-integer program and transformed into a linear mixed–binary program. Orçun et al. (2001) introduced a continuous time model for production planning and scheduling. The proposed model is MINLP, which is reformulated as MILP using linearization techniques.

One of the cases of mixed-integer nonlinear programming is mixed-integer quadratic programming (MIQP) (Lee and Leyffer, 2012). It has the following form:

$$
\begin{aligned}
\min_{x \in X} \ & \frac{1}{2} x^T \mathrm{H} x + \mathrm{f}^T x \quad , \\
\text{s.t. } & Ax \le b, \\
& A_{eq} x = b_{eq}, \\
& l_b \le x \le u_b, \\
& x_i \in \mathbb{Z},
\end{aligned}
$$

where H is an $n \times n$ matrix and f is an $n \times 1$ vector containing the quadratic objective function. The other notation is as before.

2.3. Algorithms for solving MINLP

The basic concept of algorithms for solving MINLP is to generate and refine bounds on its optimal solution value. Lower bounds are generated by solving a relaxation of MINLP, and upper bounds are provided by the value of a feasible solution to MINLP. Specificity of algorithms depends on the way how the bounds are generated and the sequence of subproblems that are solved to generate these bounds (Lee and Leyffer, 2012).

A retrospective on optimization techniques applied in the process systems engineering is presented in the paper (Biegler and Grossmann, 2004). Mixed integer and nonlinear programming, applied in scheduling problems in the flexible job shop cases, is overviewed in the paper (Özgüven et. al., 2012). Many deterministic methods for solving convex MINLP problems have been reviewed in (Biegler and Grossmann, 2004). The algorithm can be as follows:

- branch and bound method can find a global solution only if the global solution of each subproblem can be found,
- generalized benders decomposition, outer-approximation, an extended cutting plane method cannot solve problems with nonconvex constraints or nonconvex objective functions, because the subproblems may not have a unique optimum.
- generalized disjunctive programming addresses discrete/continuous optimization problems that involve disjunctions with nonlinear inequalities and logic propositions.

Branch and bound (B&B) is a well-known method for solving combinatorial optimization problems. The main idea of the method is to enumerate all feasible solutions S of the problem. S is divided into n' subsets S_i, $i = 1, \dots, n'$, ${}_{i=1}^{n'} S_i = S$. The branch and bound method consists of three main steps – branching, bounding, and pruning:

1. The process of branching divides all feasible solutions S into subsets S_i. The branching is a recursive process: each S_i is divided into further subsets. The branching is represented as a branching tree, where S is the root and S_i are branches.
2. The process of bounding calculates lower and upper bounds for all feasible solutions.
3. If the lower bound of a subset is larger or equal to the best upper bound, the subset cannot produce a better solution than that already found and is discarded

from the search. If subsets cannot be pruned, the branching must continue from the current subsets.

The most effective branch and bound methods for the job shop problem are based on the so-called disjunctive graph model. All the operations of the same job are connected, using conjunctive (directed) arcs, and operations of different jobs are connected, using disjunctive (undirected) arcs. When building a complete schedule, precedence relations are fixed between operations by turning disjunctive arcs into conjunctive. A set of fixed disjunctions defines a feasible schedule, if and only if every disjunctive arc has been fixed and the resulting graph is acyclic (Brucker, 2007), (Pinedo, 2012).

In the paper (Özlen and Azizoğlu, 2009), a rescheduling problem, where a set of jobs has already been assigned to unrelated parallel machines, is solved by the branch and bound algorithm which generates all efficient solutions with respect to the efficiency and stability measures.

Outer-approximation is one of the main approaches for solving MINLP. The algorithm is based on the fact that MINLP is equivalent to MILP of finite size. The outer-approximation algorithm is the interplay between two solvers: one solves MILP models, another solves nonlinear ones (Lee and Leyffer, 2012).

The Generalized benders decomposition method is very similar to the outer-approximation method. Instead of using linearizations for each nonlinear constraint, this method uses the duality theory to derive one single constraint that combines the linearizations derived from all the original problem constraints (Lee and Leyffer, 2012).

Generalized disjunctive programming involves both Boolean and continuous variables that are specified in constraints, disjunctions, and logic propositions. It is an alternative representation of the traditional mixed-integer programming.

When a rescheduling problem is solved by the mentioned methods, the following commonly used priority rules are taken into consideration:

- First come, first served (FCFS);
- Last come, first served (LCFS);
- Earliest due date (EDD);
- Shortest processing time (SPT);
- Longest processing time (LPT);
- Critical ratio (CR): (time until the due date)/(processing time);
- Slack per remaining operations (S/RO);
- Slack /(number of the remaining operations).

2.4. Mixed-integer signomial programming

Signomial programming is an optimization technique to solve a class of nonconvex nonlinear programming problems. Although signomial programming problems frequently occur in the engineering management science, the problems with nonconvex functions are still difficult to be solved in order to obtain global optimal solutions (Lin et al., 2012). Signomial programming can be used for solving a mixed-integer problem.

The mixed-integer signomial programming problem formulation is as follows (Lundell et al., 2009):

$$\min_{x \in X} f(x),$$
$$\text{s.t. } Ax \le b,$$

$$A_{eq}x = b_{eq},$$
$$l_b \leq x \leq u_b,$$
$$g_l(x) \leq 0, \quad l = 1, \dots, s$$
$$q_t(x) + \sigma_t(x) \leq 0, \quad t = 1, \dots, r$$
$$x_i \in \mathbb{Z}.$$

The vector x may contain both continuous and integer variables. The differentiable real functions f and g are (pseudo)convex, the function q is convex, the function σ is signomial. A signomial function is a sum of z signomial terms, where each term consists of products of the power functions (x_i are positive):

$$\sigma(x) = \sum_{j=1}^{z} c_j \prod_{i=1}^{n} x_i^{p_{ji}}, \quad c_j, p_{ji} \in \mathbb{R}.$$

The GGPECP algorithm solves non-convex MINLP problems containing signomial functions (Lundell et al., 2009). Global optimization algorithms are developed in (Floudas, 2000) to solve signomial programming problems, based on the exponential variable transformation, the convex relaxation, and the branch and bound type algorithm. The algorithms transform the nonconvex problem into a convex problem and then solve it to obtain the global optimal solution.

2.5. Constraint programming

Constraint programming (CP) is also used for solving scheduling problems. This technique was started in 1980 by the artificial intelligence community. In recent years, it has often been combined with the operations research techniques in order to improve its effectiveness (Pinedo, 2012). Constraint programming tries to find a good solution that is feasible and that satisfies all the given constraints. In scheduling problems, the constraints may include different termination dates and due dates of jobs. It is not necessary that the objective function be minimized.

Constraint programming is applied to the job shop scheduling problem as follows: suppose that a schedule has to be found with a makespan less than or equal to a given deadline; the constraint satisfaction algorithm produces, for each machine, a sequence of operations such that the overall schedule has a makespan less than or equal to the deadline (Pinedo, 2012). In the paper (Heinz and Beck, 2002), the application of constraint linear programming to scheduling problems, that require resource and start-time assignments to satisfy resource capacities, has been investigated. Some solutions of planning and scheduling problems by the combined integer and constraint programming have been introduced in the paper (Timpe, 2002). The authors presented an industrial application of a combined MIP/CP algorithm that is able to find a good feasible solution to a problem.

3. Review of mixed-integer programming and constraint programming solvers

Various solvers can be used for solving mathematical programming problems. Some of them are integrated into advanced planning and scheduling systems. A review of MINLP solvers is presented in (Bussieck and Vigerske, 2010). The most popular solvers are described here.

IBM ILOG CPLEX Optimization Studio[5] includes mathematical programming and constraint programming optimization models and solvers. The CPLEX Optimizer provides flexible, high-performance mathematical programming solvers for linear programming, mixed-integer programming, quadratic programming, and quadratically constrained programming problems. The CPLEX CP Optimizer is a tool for constraint programming and it computes optimized schedules and solves other difficult optimization problems.

FICO Xpress Optimization Suite[6] software is a platform to develop optimization solutions that drive business process improvements. It provides many sophisticated, robust optimization algorithms for solving large-scale linear, mixed-integer linear and nonlinear, quadratic, mixed-integer quadratic and quadratically-constrained quadratic problems.

LINDO API[7] is a library of optimization solvers and mathematical programming tools. Various optimization methods are implemented: stochastic, linear, convex and nonconvex nonlinear, mixed-integer, quadratic, quadratically constrained, second order cone and integer optimization.

BARON[8] (Branch And Reduce Optimization Navigator) is a computational system for solving nonconvex optimization problems in order to achieve the global optimality. Continuous, integer and mixed-integer nonlinear problems can be solved using the software.

KNITRO[9] is a software package for solving large scale mathematical optimization problems. It is specialized for nonlinear optimization, but also solves linear programming problems, quadratic programming problems, mixed-integer linear, quadratic or nonlinear programming problems.

MOSEK[10] is a solver for linear programming, mixed-integer programming, quadratic programming, and convex nonlinear programming problems.

BONMIN[11] (Basic Open-source Nonlinear Mixed-INteger programming) is the experimental open source software for solving mixed-integer nonlinear problems.

The algebraic modelling languages are useful in developing optimization models. The most comprehensive and powerful, as well as most popular modelling languages are as follows: AIMMS[12], AMPL[13], GAMS[14]. They allow us to use the common notation in formulating optimization models. There are interfaces through which they are linked to various solvers.

---

[5] http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/
[6] http://www.fico.com/en/products/fico-xpress-optimization-suite/
[7] http://www.lindo.com/index.php?option=com_content&view=article&id=1&Itemid=9
[8] http://www.theoptimizationfirm.com
[9] http://www.ziena.com
[10] http://www.mosek.com
[11] https://projects.coin-or.org/Bonmin
[12] http://business.aimms.com
[13] http://www.ampl.com
[14] http://www.gams.com/

The network-enabled optimization system (NEOS)[15] is a free internet-based service for solving optimization problems. NEOS allows solving various optimization problems and provides several interfaces for accessing the solvers. It involves the solvers for bound constrained optimization, combinatorial optimization, integer programming, global optimization, linear programming, mixed-integer linear programming, mixed-integer nonlinearly constrained optimization, nonlinearly constrained optimization, semidefinite programming, etc.

The TOMLAB[16] optimization environment is a package for solving complex optimization problems in Matlab. TOMLAB is compatible with the MathWorks optimization toolbox, so Matlab solver algorithms are supported. TOMLAB solves more types of optimization problems, and it is faster and more robust than the MathWorks optimization toolbox. Moreover, the well-known state-of-the-art optimization solvers are integrated into TOMLAB. Various optimization problems can be solved: mixed-integer linear, quadratic and nonlinear programming, semidefinite programming, geometric programming, global optimization problems, etc.

In Table 1, the comparison of solvers is presented, where mixed-integer linear programming (MILP), mixed-integer nonlinear programming (MINLP), mixed-integer quadratic programming (MIQP) and constraint programming (CP) are implemented in the solvers.

Table 1. The comparison of solvers for MILP, MINLP, MIQP, CP.

Solvers	**MILP**	**MINLP**	**MIQP**	**CP**
IBM ILOG CPLEX Optimization Studio	×		×	×
FICO Xpress Optimization Suite	×	×	×	×
LINDO API	×	×		
BARON		×		
KNITRO	×	×	×	
MOSEK	×		×	
BONMIN		×		
NEOS	×	×		
TOMLAB	×	×	×	

Conclusions

In the paper, an overview of deterministic methods is presented, based on mixed-integer programming suitable for scheduling and rescheduling problems in the framework of the production effectiveness navigator. The navigator has to react to unexpected events occurring in the production process in the shop-floor level. The rescheduling must be done, as the process cannot be continued according to the initial schedule. The state-of-the-art optimization methods should be involved. Moreover, the management focus should be taken into account, when not only the initial schedule, but also a new schedule

[15] http://neos.mcs.anl.gov/neos/solvers/
[16] http://tomopt.com/tomlab/

is created. Deterministic approaches can be successfully used for solving rescheduling problems. Most of them are able to find the global optimal solutions. Despite that the computation is time-consuming in this case, modern computing technologies and their power allow us to get the desirable solutions in the acceptable time. On the other hand, if scheduling problems are highly complicated, deterministic methods can be integrated with heuristics.

Acknowledgements. The research was supported by the EUROSTARS Project E!6232-PEN "Production effectiveness navigator". The authors cordially thank all the project partners for contribution and the Lithuanian Agency for Science, Innovation and Technology (MITA) for the financial support of the research.

References

Akturk, M. S., Ilhan, T. (2011). Single CNC machine scheduling with controllable processing times to minimize total weighted tardiness. Computers & Operations Research, 38(4), 771–781.
Avci, S., Akturk, M. S., Storer, R. H (2003). A Problem Space Algorithm for Single Machine Weighted Tardiness Problems, IIE Transactions, 35 (5), 479–486.
Baker, K. R., Trietsch, D. (2009). Principles of Sequencing and Scheduling. New Jersey: John Wiley & Sons, Inc.
Biegler, L., Grossmann, I. (2004). Retrospective on optimization. Computers & Chemical Engineering, 28(8), 1169-1192.
Brucker, P. (2007). Scheduling algorithms., 5th ed., Springer, 371p.
Bussieck, M. R., Vigerske, S. (2010). MINLP Solver Software. In: Wiley Encyclopedia of Operations Research and Management Science.
Bülbül, K., Kaminsky, Ph. (2013). A linear programming-based method for job shop scheduling. Journal of Scheduling, 16, 161–183.
Čaplinskas, A., Dzemyda, G., Kiss, F., Lupeikienė, A. (2012). Processing of undesirable business events in advanced production planning systems. Informatica, 23(4), 563-579.
Floudas, C. A. (2000). Deterministic Global Optimization: Theory, Methods and Application, Boston, Mass: Kluwer Academic Publishers.
Floudas, C. A., Gounaris, C. E. (2009). A review of recent advances in global optimization. Journal of Global Optimization, 45(1), 3-38.
Floudas, C. A., Lin, X. (2005). Mixed-Integer Linear Programming in Process Scheduling: Modeling. Algorithms, and Applications. Annals of Operations Research 139,131–162.
Fourer, R., Gay, D. M., Kernighan, B. W. (2003). AMPL A MOdeling Language for Mathematical Programming, Second Edition mont., Books/Cole.
Greenberg, H. J. (2010). Myths and Counterexamples in Mathematical Programming, Mathematical proggramming Glossary.
Heinz, S., Beck, J.C. (2011). Solving resource allocation/scheduling problems with constraint integer programming. In: Salido, M.A., Barták, R., Policella, N. (eds.) Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS 2011), 23–30.
Ivert, L. K. (2009). Advanced Planning and Scheduling Systems in Manufacturing Planning Processes. Thesis for the degree of licentiate of engineering (Report L2009:036), Department of Technology Management and Economics, Division of Logistics and Transportation, Chalmers University of Technology, Göteborg, Sweden.
Kallrath, J. (2005). Solving Planning and Design Problems in the Process Industry Using Mixed Integer and Global Optimization. Annals of Operations Research, 140(1), 339–373.

Karmarkar, N. (1984). A New Polynomial Time Algorithm for Linear Programming. Combinatorica, 4(4), 373–395.

Klemmt, A., Horn, S., Weigert, G., Wolter, K. J. (2009). Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. Robotics and Computer-Integrated Manufacturing 25(6), 917–925.

Lee, J., Leyffer, S. (2012). Mixed Integer Nonlinear Programming. The IMA Volumes in Mathematics and its Applications, Vol. 154. New York: Springer.

Lin, M. H, Tsai, J. F., Yu, C. S. (2012). A Review of Deterministic Optimization Methods in Engineering and Management, Mathematical Problems in Engineering, 15 pages.

Lundell, A., Westerlund, J., Westerlund, T. (2009). Some transformation techniques with applications in global optimization. Journal of Global Optimization, 43(2–3), 391–405.

Moreno, M. Montagna, J. (2009). A multiperiod model for production planning and design in a multiproduct batch enviroment. Mathematical and Computer Modelling, 49, 1372–1385.

Newmann, K., Schwindt, C., Trautmann, N. (2002). Advanced Production Scheduling for Batch Plants in Process Industries. OR Spektrum, 24, 251–279.

Orçun, S., Altinel, I., Hortaçsu, Ö. (2001). General Continuous Time Models for Production Planning and Scheduling of Batch Processing Plants: Mixed Integer Linear Program Formulations and Computational Issues. Computers & Chemical Engineering, 25, 371–389.

Özgüven, C., Yavuz, Y., Özbakir, L. (2012). Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times. Applied Mathematical Modelling 36(2), 846–858.

Özlen, M., Azizoğlu, M. (2009). Generating all efficient solutions of a rescheduling problem on unrelated parallel machines. International Journal of Production Research, 47(19), 5245–5270.

Pan, Ch. H. (2007). A study of integer programming formulations for scheduling problems. International Journal of Systems Science 28(1), 33–41.

Pérez-González, P., Framiñán Torres, J. M., Molina Parente, J. M., Andrade Pineda, J. L. (2009). Proceedings of 3nd International Conference on Industrial Engineering and Industrial Management, 12–19.

Pinedo, M. (2012). Scheduling: Theory, Algorithms, and Systems, Springer, 2012.

Roslöf, J., Harjunkoski, I., Björkqvist, J., Karlsson, S., Westerlund, T. (2001). An MILP-based reordering algorithm for complex industrial scheduling and rescheduling. Computers and Chemical Engineering 25, 821–828.

Sawik, T. (2011). Scheduling in Supply Chains Using Mixed Integer Programming. Wiley & Sons, Inc.

Timpe, C. (2002). Solving Planning and Scheduling Problems with Combined Integer and Constraint Programming. OR Spectrum, 431–448.

Vieira, G. E., Herrmann, J. W. Lin, E. (2003). Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. Journal of Scheduling, 6(1), 39–62.

Ziaee, M. (2014). A heuristic algorithm for solving flexible job shop scheduling problem, The International Journal of Advanced Manufacturing Technology 71(1–4), Springer London, 519–528.

Received December 23, 2013, accepted January 27, 2014