Foundations of Management, Vol. 6, No. 3 (2014), ISSN 2080-7279
DOI: 10.1515/fman-2015-0014

DECLARATIVE MODELING FOR PRODUCTION ORDER PORTFOLIO SCHEDULING

Zbigniew BANASZAK*, Grzegorz BOCEWICZ**

*Warsaw University of Technology, Faculty of Management, Warsaw, Poland
e-mail: z.banaszak@wz.pw.edu.pl

** Koszalin University of Technology, Dept. of Computer Science and Management, Poland
e-mail: bocewicz@ie.tu.koszalin.pl

Abstract: A declarative framework enabling to determine conditions as well as to develop decision-making software supporting small- and medium-sized enterprises aimed at unique, multi-project-like and mass customized oriented production is discussed. A set of unique production orders grouped into portfolio orders is considered. Operations executed along different production orders share available resources following a mutual exclusion protocol. A unique product or production batch is completed while following a given activity's network order. The problem concerns scheduling a newly inserted project portfolio subject to constraints imposed by a multi-project environment The answers sought are: Can a given project portfolio specified by its cost and completion time be completed within the assumed time period in a manufacturing system in hand? Which manufacturing system capability guarantees the completion of a given project portfolio ordered under assumed cost and time constraints? The considered problems regard finding a computationally effective approach aimed at simultaneous routing and allocation as well as batching and scheduling of a newly ordered project portfolio subject to constraints imposed by a multi-project environment. The main objective is to provide a declarative model enabling to state a constraint satisfaction problem aimed at multi-project-like and mass customized oriented production scheduling. Multiple illustrative examples are discussed.

Keywords: project portfolio, scheduling, routing, declarative modeling, customized production.

1 Introduction

The current manufacturing environment can be characterized in terms of many factors but the key one for companies confronting the challenge of remaining competitive in an era of globalization is undoubtedly the capability of fast and accurate decision making, especially so in the domain of mass customized production/services management. Most companies, particularly small- and medium-sized enterprises (SMEs), have to manage various projects sharing a pool of constrained resources and taking into account various objectives at the same time. That means SMEs have to deal with multi-project-like and mass customized oriented production. In that context, executives want to know how much a particular production order will cost, what resources are needed, which resource allocation can guarantee due time production order completion, and so on. Consequently, the decision support tools employing methods and techniques aimed at such marked demands caused by consumer need are of great importance. Such methods enhancing an online project management, and supporting a manager in the course of decision making, for example, in the course of evaluation whether a new order can be accepted to be processed in a multi-project environment of a manufacturing system at hand or not, could be implemented into dedicated decision support system (DSS) [7] tools or into on-demand decision support software packages, for example, cloud computing available such as Software as a Service (SaaS).

The main objective of a DSS aimed at multi-product production flow planning is the coordination of processes and activities related to work order processing, that is, the transportation, inventory management, warehousing, and manufacturing. In other words, the goal is to achieve a well-synchronized behavior of dynamically interacting components, where the right quantity of the right material is provided in the right place at the right time.

Declarative approaches [2, 7, 13, 20, 23 and 25] to systems and/or process modeling promise a high degree of flexibility. Constraint programming (CP) is an emergent software technology for declarative description and effective for solving large combinatorial problems especially so in areas of integrated production planning. In that context, CP can be considered as a well-suited framework for the development of decision-making software supporting SMEs in the course of a multi-project-like and mass customized [17, 20, 23] oriented production where unique production orders grouped into portfolios have to be completed in assumed time periods.

Therefore, the considered problem regards the development of a CP-driven modeling framework providing a methodology for DSS design aimed at prompt and interactive service to a set of routine queries formulated either in a direct or reverse way: Whether a given production order portfolio (POP) [4, 5, 24] specified by its cost and completion time can be completed within the assumed time period in a manufacturing system in hand? Which manufacturing system capability guarantees the completion of a given POP ordered under assumed cost and time constraints? The sought reference model, encompassing consumer order requirements and available production capabilities, has to provide a formal framework allowing one to develop a class of DSSs dedicated to interactive production flow planning subject to multi-project environment constraints.

The rest of the paper is organized as follows. Section 2 introduces a concept of project portfolio management through a declarative modeling framework and project portfolio scheduling. Section 3 provides a reference model for the project portfolio scheduling problem. Two approaches to its solution, called straight and reverse, are discussed in deep modeling problem formulation focused at POP prototyping. The methodology behind DSS dedicated to project portfolio scheduling and conclusions are presented in Sections 4 and 5, respectively.

2 Project Portfolio Management

An optimal assignment of available resources to production steps in a multi-product job shop is often economically indispensable. The goal is to generate a plan/schedule of production orders for a given period of time while minimizing the cost that is equivalent to the maximization of profit [7, 9, 19, 21]. In that context, executives want to know how much a particular production order will cost, what resources are needed, which resource allocation can guarantee due time production order completion, and so on. In other words, they are searching for responses to the standard, routine questions, such as [7, 8, 13, 20, 24]: Can the production order be completed before an arbitrary given deadline? What is the production completion time following assumed robot operation time? Is it possible to undertake a new production order under given (constrained in time) resource availability while guaranteeing disturbance-free execution of the already executed orders? What values and of what variables guarantee the production order will be completed following an assumed set of performance indexes?

2.1 Modeling Framework

The problems behind the quoted questions belong to the class of so-called project scheduling. In turn, project scheduling can be defined as the process of allocating scarce resources to activities over a period of time to perform a set of activities in a way taking into account a given set of performance measures. The scheduling of multi-stage batch processes has received significant attention from researchers in the process systems engineering community. Existing methods assume that routing and allocation as well as batching and scheduling decisions are made independently, that is, each production order is treated as an activity network and is assigned to processing units, and then divided into a number of batches (batching), and sequenced (scheduling). Several techniques have been proposed in the past 50 years, including MILP [18, 20, 22], branch-and-bound [11], or more recently artificial intelligence. The last class of techniques concentrates mostly on fuzzy set theory and CP frameworks [1, 5, 14].

Very limited works, however, focus on the joint technological processes, transportation routing, and financial aspects [6]. Constraint programming/constraint logic programming (CP/CLP) languages [6, 11, 15, 20] seem to be well suited for modeling such real-life and day-to-day decision-making processes [10].

Furthermore, there is another aspect of the addressed problem, namely multi-criteria decision making under uncertain conditions. Fuzzy multi-criteria decision making is primarily adopted for selecting, evaluating, and ranking alternative solutions to problems [7]. To do this in a way compatible with real-life settings necessitates the use of stochastic and fuzzy logic frameworks [18]. The fuzzy model of project portfolio online control can be specified as a declarative one and then implemented using CP techniques and finally implemented as a decision support system [7]. Some applications of fuzzy set theory in production management [26] show that most of the research on project scheduling has been focused on fuzzy PERT and fuzzy CPM.

Of course, in the general case a hybrid model specified by discrete distinct and/or imprecise (e.g., fuzzy) variables and renewable and/or non-renewable resources can be considered [1, 5, 12 and 14].

The assumed reference model enabling a descriptive way of a direct or reverse problem formulation encompasses a constraint satisfaction problem (CSP) structure while taking into account different types of variables and constraints of project planning problems.

Therefore, an approach proposed assumes a kind of reference model encompassing an open structure enabling one to take into account different types of variables and constraints as well as to formulate straight and reverse kinds of project planning problems. In this context, the contribution provides the framework allowing one to take into account both crisp and fuzzy data describing modeled objects and then to treat them in terms of the CSP [11].

In order to illustrate the approach proposed, let us focus on a reference model of decision problem encompassing equilibrium between possible expectations regarding potential order completion (e.g. following a set of routine queries) and available production capabilities. The considered decision problem concerns of resources conflict resolution, that is, conflicts arising when different activities simultaneously request their access to renewable and non-renewable resources of limited quantity.

2.2 Constraint Satisfaction Problem

Constraint satisfaction problem (CSP) is determined by the set of decision variables $X = \{x_1, x_2,..., x_n\}$, the family of variable domains $D = \{D_i \mid D_i = (d_{i,1}, d_{i,2},..., d_{i,j},..., d_{i,m}), i = 1,..., n\}$, and the set of constraints $C = \{c_i \mid i = 1, 2,.., l_c\}$ encompassing relations linking variables. Each constraint c_i can be seen as the relation defined on the relevant subset of variables $X_i \subset X = \{x_1, x_2,..., x_n\}$. Consequently the CSP is denoted as follows: $CS = ((X, D), C)$.

Consider the vector $V = (v_1, v_2,\ldots, v_n) \in D_1 \times D_2 \times\ldots \times D_n$. The vector V such that all the constraints hold is treated as the admissible solution of CS. Let us suppose, the constraint c_i defined on the subset $X_i = \{x_l, x_k,\ldots, x_m\}$ follows the logic value equal to "true" (noted as $w(c_i) = 1$); in this case there exists $V_i \in D_l \times D_k \times \ldots \times D_m$ such that c_i holds. In that context, the set of admissible solutions is defined as follows:

$$\mathrm{V} = \{V = (v_1, v_2, \ldots, v_n) \mid v_i \in D_i, i = 1, \ldots, n, \\ w(c_1) = w(c_2) = \ldots = w(c_{lc}) = 1\}.$$

Therefore, CSP can be seen also as a triple set (data, constraints, query), that is, the set of variables and family of variable domains, the set of constraints, and the question: Does there exist non-empty set V? Of course, in the general case, instead of an admissible solution an optimal one can be searched for, as well.

Solution strategies are based on two subsequently used mechanisms, that is, constraint propagation and variable distribution. Variable distribution can be executed either through systematic (e.g., breath-first-search) or stochastic search of the whole or constrained state space of potential solutions obtained from constraint propagation. The searching strategies are implemented in constraint logic programming or CP languages such as CHIP, OzMozart, ILOG, and so on [11, 13, 18 and 25].

2.3 Project Portfolio Scheduling

The declarative model considered assumes that each new portfolio of production orders can be accepted for execution in a given workshop only if its performance will do not disturb other job executions while its completion will follow presumed demands imposed by customers, for example, deadlines, production costs, and so on. The above problem belongs to multiple projects or POP scheduling and can be modeled and then resolved by different methods mentioned above. However, the advantages of using CP technique are: (1) it reduces the search space and therefore, it can find a feasible solution in a short time, which is required for online control; (2) it can be implemented in standard software such as ILOG and PROLOG [18, 25].

In case an unforeseen event occurs, for example, caused by the occurrence of production flow disturbance and/or new production orders, the current schedule of production flow becomes infeasible. Thus, it is necessary to reschedule project portfolios and to reallocate resources in online mode. The idea of proactive scheduling is presented in Fig. 1.

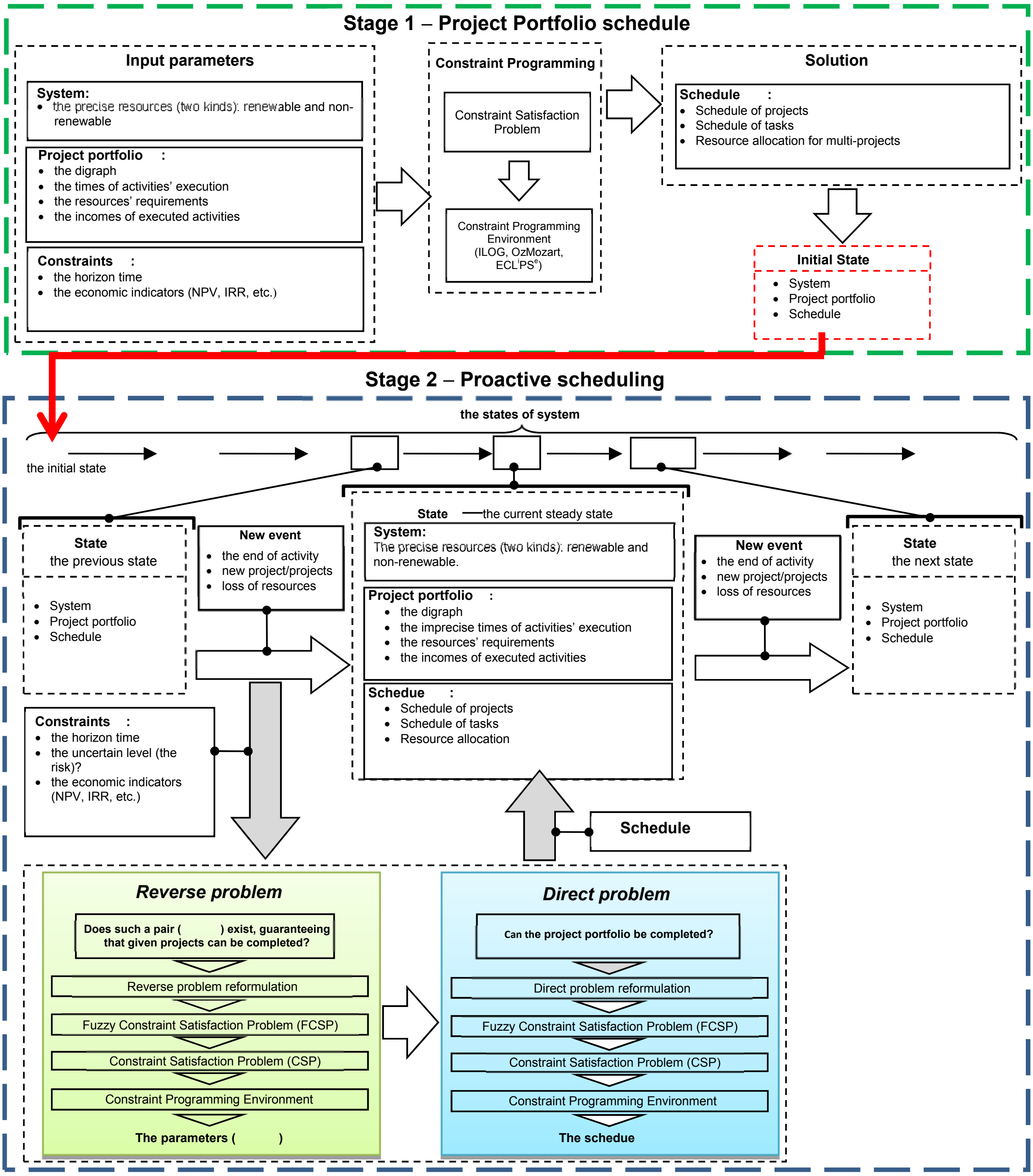

Figure 1. The idea of multi-product production flow scheduling and rescheduling

To be able to achieve this requires solving the problem in two steps: first, as a reverse problem and then as a direct problem. The reverse problem is formulated to establish the range of values of parameters guaranteeing a feasible plan exists. Therefore, the result from the reverse problem will guarantee finding a feasible solution in the direct problem and significantly reduce computational time. The solution of the reverse problem will be used as an input parameter for the direct problem, which aims to find a new plan for projects with minimum cost. In other words, besides direct formulation of the scheduling problem its alternative statement formulated as a reverse scheduling problem can be as follows: Which values of the system parameters guarantee that the set of orders will be completed while giving a certain set of values for performance indexes? In the case of any new event, caused for instance by including a new project, a new system state has to be considered to determine a new project portfolio schedule. In that context, the proposed approach involves solving both direct and reverse problems for systems where project portfolios (specified by fuzzy data [1, 12]) change over time as a result of randomly occurring events.

Due to the high complexity of planning problems, it is assumed that reverse/direct problems are represented in constraint satisfaction formalism to give the current state of the system. Therefore, the proposed approach assumes the portfolio rescheduling takes place at states; the declarative model representing a reverse/direct problem of production flow scheduling is transformed to CSP and then solved using CP-based techniques. Its distinct advantage is that it separates the problem statement and its resolution methods. In addition, integrating the cash flows and the resource allocations with data describing the stochastic nature of possible disturbances is considered in the online control.

The proposed methodology should provide the following contributions:

- a method for follow-up planning and online control subject to financial, time and resource capacity constraints, in an uncertain multi-project environment;
- an approach solving the reverse problem for project portfolio planning by integrating stochastic, fuzzy logic and the CP methods.

3 Declarative Modeling

Let us consider the reference model of a decision problem concerning multi-resource task allocation in a multi-product job shop assuming the precise character of decision variables. The model specifies both the job-shop capability and production order requirement in a unified way, that is, through the description of determining the sets of variables and sets of constraints restricting domains of discrete variables. Some conditions concerning the routine questions are included in the set of constraints. That means if such conditions hold, the response to associated questions is positive.

3.1 Reference Model

The reference model considered specifies both SMEs and project portfolio in terms of describing their variables and constraints.

Set of decision variables – X:

- **SME – a job-shop perspective**

Given an amount lz of renewable discrete resources ro_i specified by (e.g., workforce, machine tools, automated guided vehicles, etc.): $Ro = (ro_1, ro_2, \ldots, ro_z)$.

Given amounts $zo_{i,k}$ of available renewable resources $zo_i = (zo_{i,1}, zo_{i,2}, \ldots, zo_{i,h})$, where $zo_{i,k}$ limited amount of the i-th renewable resource available at the k-th moment of H: $H = \{0,1,\ldots, h_{max}\}$, specified by $Zo = (zo_1, zo_2, \ldots, zo_{lz})$.

Given amount ln of non-renewable resources (i.e., money) rn_i specified by: $Rn = (rn_1, rn_2, \ldots, rn_{ln})$.

Given amounts zn_i of available non-renewable resources rn_i specified by: $Zn = (zn_1, zn_2, \ldots, zn_{ln})$, where zn_i denotes the amount of the resource rn_i being available at the beginning of time horizon H.

- **Project portfolio**

Given a set of projects $P = \{P_1, P_2, \ldots, P_{lp}\}$, where P_i is specified by the set composed of lo_i activities, that is, $P_i = \{O_{i,1}, \ldots, O_{i,loi}\}$, where:

$$O_{i,j} = (s_{i,j}, t_{i,j}, Tp_{i,j}, Tz_{i,j}, Dp_{i,j}, Tr_{i,j}, Ts_{i,j}, Cr_{i,j}, Cw_{i,j}) \quad (1)$$

$s_{i,j}$ – the starting time of the activity $O_{i,j}$, that is, the time counted from the beginning of the time horizon H,

$t_{i,j}$ – the duration of the activity $O_{i,j}$,

$Tp_{i,j} = (tp_{i,j,1}, tp_{i,j,2}, \ldots, tp_{i,j,lz})$ – the sequence of moments the activity $O_{i,j}$ requires new amounts of renewable resources: $tp_{i,j,k}$ – the time counted since the moment $s_{i,j}$ of the $dp_{i,j,k}$ amount of the k-th resource allocation to the activity $O_{i,j}$. That means a resource is allotted to an activity during its execution period: $0 \le tp_{i,j,k} < t_{i,j}$; $k = 1,2,\ldots,lz$.

$Tz_{i,j} = (tz_{i,j,1}, tz_{i,j,2}, \ldots, tz_{i,j,lz})$ – the sequence of moments the activity $O_{i,j}$ releases the subsequent resources: $tz_{i,j,k}$ – the time counted since the moment $s_{i,j}$ the $dp_{i,j,k}$ amount of the k-th renewable resource was released by the activity $O_{i,j}$. That is, assumed a resource is released by activity during its execution: $0 < tz_{i,j,k} \le t_{i,j}$ and $tp_{i,j,k} < tz_{i,j,k}$; $k = 1, 2, \ldots, lz$.

$Dp_{i,j} = (dp_{i,j,1}, dp_{i,j,2}, \ldots, dp_{i,j,lz})$ – the sequence of the k-th resource amounts $dp_{i,j,k}$ allocated to the activity $O_{i,j}$, that is, $dp_{i,j,k}$ – the amount of the k-th resource allocated to the activity $O_{i,j}$. That assumes: $0 \le dp_{i,j,k} \le zo_k$; $k = 1, 2, \ldots, lz$.

$Cr_{i,j} = (cr_{i,j,1}, cr_{i,j,2},..., cr_{i,j,ln})$ – the sequence of non-renewable resource amount required by activity $O_{i,j}$: $cr_{i,j,k}$ – the amount of the k-th resource required by the activity $O_{i,j}$, $cr_{i,j,1} \leq 0$; k = 1, 2,…, ln, $cr_{i,j,k} = 0$ means the activity does not consume the k-th resource.

$Cw_{i,j} = (cw_{i,j,1}, cw_{i,j,2},..., cw_{i,j,ln})$ – the sequence of amounts of non-renewable resources released by activity $O_{i,j}$: $cw_{i,j,k}$ – the amount of the k-th resource involved by activity $O_{i,j}$, $cw_{i,j,1} \geq 0$; k = 1,2,…, ln; $cr_{i,j,k} = 0$ means the activity does not engage the k-th resource.

$Tr_{i,j} = (tr_{i,j,1}, tr_{i,j,2},..., tr_{i,j,ln})$ – the sequence of moments the determined amounts of subsequent non-renewable resources are required by activity $O_{i,j}$: $tr_{i,j,k}$ – the time counted since the moment $s_{i,j}$ the $dp_{i,j,k}$ amount of the k-th non-renewable resource was released by the activity $O_{i,j}$.

That is, assumed a resource is collected by activity during its execution: $0 \leq tr_{i,j,k} < t_{i,j}$; k = 1, 2, …, ln.

$Ts_{i,j} = (ts_{i,j,1}, ts_{i,j,2},..., ts_{i,j,ln})$ – the sequence of moments the determined amounts of subsequent non-renewable resources are generated (released) by activity $O_{i,j}$: $ts_{i,j,k}$ – the time counted since the moment $s_{i,j}$ the $cw_{i,j,k}$ amount of the k-th non-renewable resource was generated by the activity $O_{i,j}$.

That is, assumed the resource is generated during activity execution, however, not earlier than the beginning of its collection, that is: $0 \leq ts_{i,j,k} < t_{i,j}$; k = 1, 2, …, ln, as well as $tr_{i,j,k} \leq ts_{i,j,k}$; k = 1, 2,…,ln.

NPV – the net present value used to measure a project's efficiency and calculated using the following formula:

$$NPV = \sum_{t=0}^{n} \frac{CF_t}{(1+k)^t} \quad (2)$$

where:

CF_t – the money netto flow expected in the year t,

k – the discount rate (alternative capital investment cost),

n – the period of a project exploitation [years].

Consequently, each activity $O_{i,j}$ is specified by the following sequences:

- starting times of activities in the activity network P_i:
  $S_i = (s_{i,1}, s_{i,2}, …, s_{i,loi})$, $0 \leq s_{i,j} < h$, i = 1, 2,…, lp; j = 1, 2,…, lo_i,
- duration of activities in the activity network P_i:
  $T_i = (t_{i,1}, t_{i,2},…, t_{i,loi})$.

Elements of sequences: $Tp_{i,j}$, $Tz_{i,j}$, $Dp_{i,j}$, $Cr_{i,j}$, $Cw_{i,j}$, $Tr_{i,j}$, and $Ts_{i,j}$ specify the activity network P_i:

- starting times the j-th resource is allocated to the k-th activity in the activity network P_i:
  $TP_{i,j} = (tp_{i,1,j}, ..., tp_{i,k,j}, ..., tp_{i,loi,k})$,
- starting times the j-th resource is released by the k-th activity in P_i:
  $TZ_{i,j} = (tz_{i,1,j}, ..., tz_{i,k,j}, ..., tz_{i,loi,j})$,
- the sequence of moments the *j*-th non-renewable resource is collected by activities of the projects P_i:
  $TR_{i,j} = (tr_{i,1,j}, ..., tr_{i,k,j}, ..., tr_{i,loi,j})$,
- the sequence of moments the j-th non-renewable resource is released by activities of the project P_i:
  $TS_{i,j} = (ts_{i,1,j}, ..., ts_{i,k,j}, ..., ts_{i,loi,j})$,
- amounts of the j-th resources allotted to the k-th activity in the project P_i:
  $DP_{i,j} = (dp_{i,1,j}, …, dp_{i,k,j}, …, dp_{i,loi,j})$,
- sequences of amounts of the j-th non-renewable resource consumed by activities of the project P_i:
  $CR_{i,j} = (cr_{i,1,j}, …, cr_{i,k,j}, …, cr_{i,loi,j})$,
- sequences of amounts of the j-th non-renewable resource involved by activities of the project P_i:
  $CW_{i,j} = (cw_{i,1,j}, …, cw_{i,k,j}, …, cw_{i,loi,j})$.

- **Set of constraints – C**

  Given the project portfolio and available amounts of renewable and non-renewable resources as well as the above-mentioned sequences: T_i, $TP_{i,j}$, $TZ_{i,j}$, and $DP_{i,j}$.

  Given the time horizon H = {0,1,…, h_{max}}, the project portfolio should be completed. That is, assumed the activities cannot be suspended during their execution, and moreover:

  - each activity can request any kind and quantity (not exceeding the resource's limited amount) of any resource,

- each resource can be uniquely used by an activity,
- the quantity of renewable resource used by an activity cannot be changed or allotted to other activity,
- an activity can start its execution only if required amounts of renewable and non-renewable resources are available at the moments given by $Tp_{i,j}$ and $Ts_{i,j}$.

The project P_i is represented by activity-on-node networks, where nodes represent activities and arcs determine an order of activities' execution. Consequently, the following activity order constraints are considered [1, 2, 3]:

- the $O_{i,k}$ activity follows the $O_{i,j}$ -th one:

  $co^1_{i,k}$: $s_{i,j} + t_{i,j} \le s_{i,k}$, (3)

- the $O_{i,k}$ activity follows other activities:

  $co^2_{i,k}$: $(s_{i,j}+t_{i,j} \le s_{i,k}) \wedge (s_{i,j+1}+t_{i,j+1} \le s_{i,k}) \wedge (s_{i,j+2}+t_{i,j+2} \le s_{i,k}) \wedge \ldots \wedge (s_{i,j+n}+t_{i,j+n} \le s_{i,k})$, (4)

- the $O_{i,k}$ activity is followed by other activities:

  $co^3_{i,k}$: $(s_{i,k} + t_{i,k} \le s_{i,j}) \wedge (s_{i,k} + t_{i,k} \le s_{i,j+1}) \wedge (s_{i,k}+t_{i,k} \le s_{i,j+2}) \wedge \ldots \wedge (s_{i,k} + t_{i,k} \le s_{i,j+n})$. (5)

  Each activity $O_{i,k}$ should be finished before the h-th unit of time (h is the completion time of project portfolio P), that is, it should follow the constraint:

  $co^4_{i,k}$: $s_{i,k} + t_{i,k} \le h$ (6)

  Constraints $co^1_{i,k}$, $co^2_{i,k}$, $co^3_{i,k}$, and $co^4_{i,k}$, encompassing relations existing among activities $O_{i,k}$ produce the set of precedence constraints

  $Co = \{co^1_{i,k}, co^2_{i,k}, co^3_{i,k}, co^4_{i,k}\}$

Note that limited resource amounts may cause resource conflicts, that is, requiring the selection of relevant dispatching rules deciding the order of resources' allocation. In order to avoid such a conflict, the relevant constraints have to be taken into account. The set of conflict avoidance constraints $Cr = \{cr^1_{m,n,k}, cr^2_{m,n,k}\}$ introduced in [4, 5, 12] consists of:

$$cr^1_{m,n,k}: \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \bar{1}(s_{m,n} + tp_{m,n,k}, s_{i,j} + tp_{i,j,k}, s_{i,j} + tz_{i,j,k})\right] \le zo_{k,x_{m,n}+tp_{m,n}-1} \quad (7)$$

$\forall$ (m,n) $\in$ {(a,b) | a = 1,2,…,lp, b = 1,2,…,lo_a},

where:

lp – the number of projects, lo_a – the number of activities in the project P_a,

$\bar{1}(u,a,b)$ – an unary function determining the time of the resource occupation $\bar{1}(u,a,b) = 1(u-a) - 1(u-b)$

$1(u)$ – the unit step function

and

$$cr^2_{m,n,k}: \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \bar{1}(vg_{k,d}, s_{i,j} + tp_{i,j,k}, s_{i,j} + tz_{i,j,k})\right] \le zo_{k,vp_{k,d}-1} \quad (8)$$

$\forall d \in \{1,2,\ldots,q\}$

where:

$vg_{k,i}$ – the moments $vg_{k,i} \in H$ at which the available number of the k-th resource units is changed,

q – number of characteristic points.

Note that $Cr = \{cr^1_{m,n,k}, cr^2_{m,n,k}\}$ regards renewable resources. The similar ones should be considered in the case of non-renewable resources, for instance concerning the money. The set of conflict avoidance constraints $Cn = \{cn^1_{m,n,k}\}$ introduced in [4, 5, 12] consists of:

$$cn^1_{m,n,k}: zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot 1(s_{m,n} - s_{i,j} - tr_{i,j,k})\right] + \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cw_{i,j,k}\cdot 1(s_{m,n} - s_{i,j} - ts_{i,j,k})\right] \ge 0 \quad (9)$$

$\forall$(m,n) $\in$ {(a,b) | a = 1,2,…,lp; b = 1,2,…,lo_a}

where:

lp – the number of projects, lo_a – the number of activities in the project P_a.

Finally, the considered set of constraints has the following form: C = Co∪Cr∪Cn.

Therefore, the reference model considered can be seen as the constraint problem CS = ((X, D), C). Consequently, depending on the questions stated the relevant context dedicated CSP can be considered. The standard questions can be formulated either in the straight or reverse way. So, the new problems can be aimed both at the determination of [8, 12, 13 and 16]:

- the criteria values implied by the assumed variables and constraints, for instance:

  Do the given activities' times guarantee completion of the project portfolio within assumed time horizon H?
- the variables guaranteeing expected values of the assumed goal functions, for example:
  What are the beginning times T_i of activities guaranteeing the project portfolio completion time does not exceed a given time horizon H?

The above questions belong to the class of so-called problems formulated in a reverse way, that is, problems our considerations are focused on. Some examples illustrating the above-mentioned two types of context dedicated *CS* are discussed in the section below.

3.2 Direct versus Reverse Approach

The following two classes of standard routine queries are usually considered and formulated in:

- a direct way (i.e. corresponding to the question: What results from premises?
  - What the portfolio makespan follows from the given project constraints specified by activity duration times, resource amount, and their allocation to projects' activities?
  - Does a given resource allocation guarantee the production order makespan does not exceed the given deadline?
  - Can the project portfolio be completed before an arbitrary given deadline?
  - and so on.
- a reverse way (i.e., corresponding to the question: What implies conclusion?)
  - What activity duration times and resource amount guarantee the given POP makespan does not exceed the deadline?
  - Does there exist resource allocation such that production order makespan does not exceed the deadline?
  - Does there exist a set of activities' operation times guaranteeing a given project portfolio completion time will not exceed the assumed deadline?

The above-mentioned categories encompass the different reasoning perspectives, that is, deductive and abductive ones. The corresponding queries can be stated in the same model that can be treated as the composition of variables and constraints, that is, assumed sets of variables and constraints limiting their values. In that context, both an enterprise and the portfolio of production orders can be specified in terms of distinct and/or imprecise variables, discrete and/or continuous variables, renewable and/or non-renewable resources, limited and/or unlimited resources, and so on.

What are the moments the activities starts their execution?

Given the following project portfolio, that is, the set of projects P = {P_1, P_2, P_3, P_4}. Activities $O_{i,j}$ of projects are specified by corresponding sets:

$P_1 = \{O_{1,1},\dots, O_{1,10}\}$, $P_2 = \{O_{2,1},\dots, O_{2,12}\}$,

$P_3 = \{O_{3,1},\dots, O_{3,11}\}$, $P_4 = \{O_{4,1},\dots, O_{4,13}\}$.

The relevant activity networks [4, 12] are shown on the following figures: Figs 2–5.

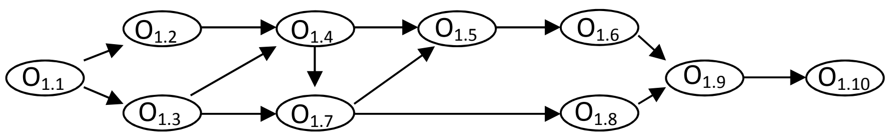

Figure 2. Activity network of the project P_1

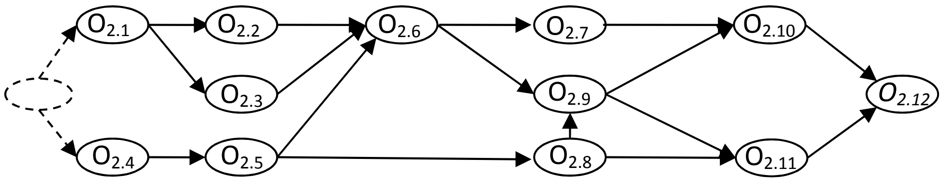

Figure 3. Activity network of the project P_2

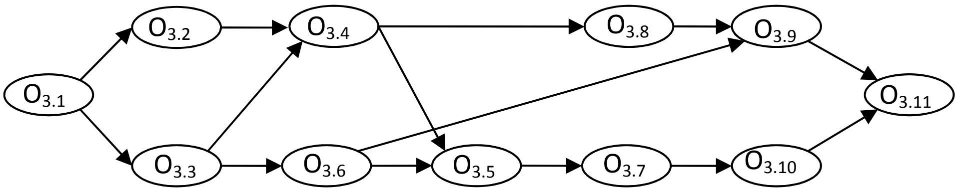

Figure 4. Activity network of the project P_3

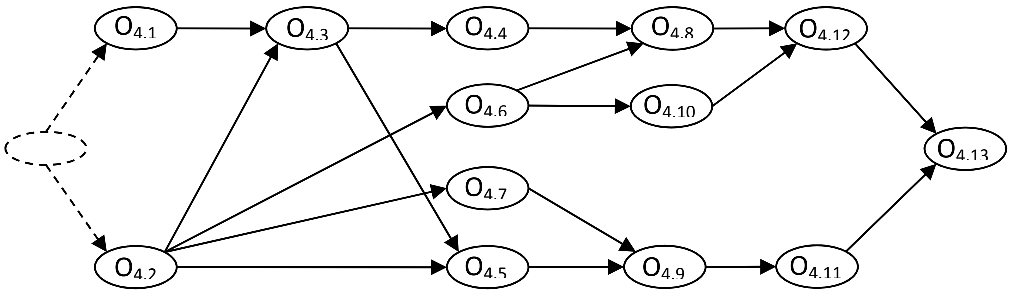

Figure 5. Activity network of the project P_4

Given the time horizon H = {0, 1,…, 40} (h_{max} = 40). Operation times for particular projects P_1, P_2, P_3, and P_4 are determined by the following sequences:

$T_1 = (1, 2, 3, 4, 4, 8, 3, 2, 1, 6)$,

$T_2 = (3, 1, 6, 3, 2, 5, 1, 5, 2, 4, 2, 1)$,

$T_3 = (3, 7, 2,7, 2, 1,8, 3, 3, 4, 8)$,

$T_4 = (3, 3, 2, 8, 3, 1, 4, 1, 8, 4, 3, 3, 8)$.

Given are three kinds of renewable resources ro_1, ro_2, and ro_3. Resource amounts are limited by the following number of units: 11, 14, and 12, respectively. Resource amounts are constant in whole time horizon H. That is, assumed an amount of resources allocated to the activity at the moment of its beginning can be released only by this activity and only at the moment of its completion. The amounts of particular resources required by projects' (P_1, P_2, P_3, and P_4) activities are given in the following tables: Tables 1–4.

Table 1. Amounts of resources required by the project P_1 activities (the sequences $DP_{1,1}$, $DP_{1,2}$, $DP_{1,3}$)

	$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{1,4}$	$O_{1,5}$	$O_{1,6}$	$O_{1,7}$	$O_{1,8}$	$O_{1,9}$	$O_{1,10}$
$DP_{1,1}$	3	1	1	1	1	1	2	1	2	1
$DP_{1,2}$	2	1	2	1	1	2	3	3	1	1
$DP_{1,3}$	2	2	3	1	1	1	1	1	2	1

Table 2. Amounts of resources required by the project P_2 activities (the sequences $DP_{2,1}$, $DP_{2,2}$, $DP_{2,3}$)

	$O_{3,1}$	$O_{2,2}$	$O_{2,3}$	$O_{2,4}$	$O_{2,5}$	$O_{2,6}$	$O_{2,7}$	$O_{2,8}$	$O_{2,9}$	$O_{2,10}$	$O_{2,11}$	$O_{2,12}$
$DP_{2,1}$	4	3	2	2	1	1	1	3	1	2	2	2
$DP_{2,2}$	1	2	3	1	2	1	2	1	1	2	1	1
$DP_{2,3}$	2	1	1	1	3	1	2	2	2	1	1	1

Table 3. Amounts of resources required by the project P_3 activities (the sequences $DP_{3,1}$, $DP_{3,2}$, $DP_{3,3}$)

	$O_{3,1}$	$O_{3,2}$	$O_{3,3}$	$O_{3,4}$	$O_{3,5}$	$O_{3,6}$	$O_{3,7}$	$O_{3,8}$	$O_{3,9}$	$O_{3,10}$	$O_{3,11}$
$DP_{3,1}$	2	4	1	2	2	2	1	2	2	1	3
$DP_{3,2}$	2	1	3	2	2	2	1	1	1	2	2
$DP_{3,3}$	2	4	1	2	2	2	1	2	2	1	3

Table 4. Amounts of resources required by the project P_4 activities (the sequences $DP_{4,1}$, $DP_{4,2}$, $DP_{4,3}$)

	$O_{4,1}$	$O_{4,2}$	$O_{4,3}$	$O_{4,4}$	$O_{4,5}$	$O_{4,6}$	$O_{4,7}$	$O_{4,8}$	$O_{4,9}$	$O_{4,10}$	$O_{4,11}$	$O_{4,12}$	$O_{4,13}$
$DP_{4,1}$	1	2	3	4	3	2	2	1	1	1	3	1	4
$DP_{4,2}$	1	1	1	2	1	2	1	3	2	2	2	1	2
$DP_{4,3}$	1	2	2	1	1	2	4	1	2	2	2	1	2

It is assumed that some activities besides renewable resources require non-renewable ones. Given are two kinds of non-renewable resources rn_1 and rn_2. The initial amount of the resource rn_1 is equal to 10 units, and of the resource rn_2 is equal to 7 units. Activities may be used up and generate some number of resources rn_1, rn_2 units. It is assumed that each activity uses up some resource units at the beginning and generates some resource units at the activity's end. The amounts of used up and generated resource rn_1 units determine sequences: $CR_{i,j}$ and $CW_{i,j}$, respectively, in the following tables: Tables 5–8

Table 5. Amount of used up (CR) and generated (CW) non-renewable resources required by activities of the project P_1 (the sequences $CR_{1,1}$, $CR_{1,2}$, $CW_{1,1}$, $CW_{1,2}$)

	$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{1,4}$	$O_{1,5}$	$O_{1,6}$	$O_{1,7}$	$O_{1,8}$	$O_{1,9}$	$O_{1,10}$
$CR_{1,1}$	1	1	2	1	2	1	3	1	1	1
$CR_{1,2}$	1	2	1	1	1	0	1	0	1	1
$CW_{1,1}$	3	2	0	2	4	4	2	0	2	4
$CW_{1,2}$	1	2	3	2	2	2	0	2	1	2

Table 6. Amount of used up (CR) and generated (CW) non-renewable resources required by activities of the project P_2 (the sequences $CR_{2,1}$, $CR_{2,2}$, $CW_{2,1}$, $CW_{2,2}$)

	$O_{2,1}$	$O_{2,2}$	$O_{2,3}$	$O_{2,4}$	$O_{2,5}$	$O_{2,6}$	$O_{2,7}$	$O_{2,8}$	$O_{2,9}$	$O_{2,10}$	$O_{2,11}$	$O_{2,12}$
$CR_{2,1}$	1	0	1	2	1	1	1	3	1	0	1	1
$CR_{2,2}$	3	2	1	2	0	2	3	2	2	2	1	2
$CW_{2,1}$	3	2	0	2	1	2	0	2	0	2	0	1
$CW_{2,2}$	3	2	1	2	0	2	3	2	2	2	1	2

Table 7. Amount of used up (CR) and generated (CW) non-renewable resources required by activities of the project P_3 (the sequences $CR_{3,1}$, $CR_{3,2}$, $CW_{3,1}$, $CW_{3,2}$)

	$O_{3,1}$	$O_{3,2}$	$O_{3,3}$	$O_{3,4}$	$O_{3,5}$	$O_{3,6}$	$O_{3,7}$	$O_{3,8}$	$O_{3,9}$	$O_{3,10}$	$O_{3,11}$
$CR_{3,1}$	1	1	2	1	1	1	0	1	3	1	1
$CR_{3,2}$	0	1	1	0	2	1	1	1	3	1	0
$CW_{3,1}$	2	3	2	0	2	1	2	2	2	3	2
$CW_{3,2}$	3	2	1	2	0	2	3	2	2	2	1

Table 8. Amount of used up (CR) and generated (CW) non-renewable resources required by activities of the project P_4 (the sequences $CR_{4,1}$, $CR_{4,2}$, $CW_{4,1}$, $CW_{4,2}$)

	$O_{4,1}$	$O_{4,2}$	$O_{4,3}$	$O_{4,4}$	$O_{4,5}$	$O_{4,6}$	$O_{4,7}$	$O_{4,8}$	$O_{4,9}$	$O_{4,10}$	$O_{4,11}$	$O_{4,12}$	$O_{4,13}$
$CR_{4,1}$	1	1	2	1	1	1	0	1	3	1	1	1	1
$CR_{4,2}$	0	1	1	0	2	1	1	1	3	1	0	1	1
$CW_{4,1}$	2	3	2	0	2	1	2	2	2	3	2	3	2
$CW_{4,2}$	3	2	1	2	0	2	3	2	2	2	1	2	2

In the context of the above assumed data, that is, based on the requirements of the standard *CS* the following question is considered.

Q_1: Does there exist a schedule following constraints assumed on availability of renewable and non-renewable resources and NPV > 0 such that the production order's completion time does not exceed the deadline H?

Note that the schedule we are looking for is determined by moments $s_{i,j}$ the activities start their execution [4, 12].

The solution to the problem results in determination of moments the activities start their execution $s_{i,j}$. So, the solution we are searching for has the form of the following sequences:

$S_1 = (s_{1,1},\ldots, s_{1,10})$,

$S_2 = (s_{2,1},\ldots, s_{2,12})$,

$S_3 = (s_{3,1},\ldots, s_{3,11})$, and

$S_4 = (s_{4,1},\ldots, s_{4,13})$.

Of course, the elements of sequences S_1, S_2, S_3, and S_4 have to follow activities' order constraints from Figs 2–5 as well as constraints assumed for renewable (see Tables 1–4) and non-renewable (see Tables 5–8) resource allocation (guaranteeing deadlock avoidance). Constraints have a form similar to the formulas (3)–(9).

The question considered implies the relevant context dedicated CS.

Given

CS = ((X, D), C),

where:

- the set of decision variables X containing:
  - the input variables: $U = \{s_{1,1}, s_{1,2}, \ldots, s_{1,lo1}, s_{2,1}, \ldots s_{lp,lolp}\}$;
  - the output variables: Y = {h, NPV};
- the family of domains D, where the domains of variables T_i, $Tp_{i,j}$, $Tz_{i,j}$, $Dp_{i,j}$, $Cr_{i,j}$, $Cw_{i,j}$, $Tr_{i,j}$, and $Ts_{i,j}$ are determined by Tables 1–8, and variables corresponding to the beginning times of activities $s_{i,j} \in [0,50]$;
- the set of constraints C.

The considered problem CS and the question Q_1 can be specified in terms of newly stated problem CCS:

CCS = ((X, D), C∪Cy),

where:

Cy – the set of output constraints corresponding to Q_1: Cy = {cy_1, cy_2},

cy_1: $h \le 40$; cy_2: NPV > 0.

The result of CCS examination by the consistency-checking procedure is positive, so there exists the solution Vu following all the constraints C∪Cy.

The results obtained from the OzMozart implemented procedure consists of the non-empty set of solutions Vu. Admissible values of considered variables $s_{i,j}$ have the following values:

$S_1 = (0, 1, 1, 4, 11, 15, 8, 11, 23, 24)$,

$S_2 = (0, 3, 7, 10, 13, 15, 20, 17, 23, 25, 25, 29)$,

$S_3 = (0, 3, 3, 10, 17, 5, 19,17, 20, 27, 31)$, and

$S_4 = (0, 0, 3, 5, 5, 3, 3, 13, 8, 6, 14, 16, 19)$.

The NPV index value calculated for projects: P_1, P_2, P_3, P_4 follows the requirement NPV > 0, that is,

$NPV_{P1} = 0.3649$,

$NPV_{P2} = 2.4775$,

$NPV_{P3} = 1.3248$, and

$NPV_{P4} = 0.8134$.

Therefore, the example presented illustrates the capability of an interactive multi-criteria project planning (e.g., taking into account a particular project deadline, project portfolio deadline, resource limits, and so on) approach to project prototyping issues. The problem of size just considered took less than 5 minutes (i.e., finding the first solution Vu; the AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1,00 GB platform has been used).

What are the times of activities' duration?

Given the following projects' portfolio, that is, the set of projects P = {P_1, P_2, P_3, P_4} specified by the same activity networks (see Figs 2–5) and resource allocations (see Tables 2–9) as previously. However, the new time horizon H = {0,1,…, 36} is considered.

Given the projects' portfolio containing the following projects P_1, P_2, P_3, and P_4. The makespan admissible cannot exceed 36 units of time. Activities' operation times are not known; however constraints determining their execution constraints are given.

Table 9. Constraints linking activities $O_{i,j}$ execution times

	Constraint		Constraint
ct_1	$t_{3,7} + t_{3,9} = 11$	ct_6	$t_{3,3} + t_{3,4} = 9$
ct_2	$t_{4,12} + t_{4,13} = 11$	ct_7	$t_{2,3} + t_{2,4} = 9$
ct_3	$t_{4,3} + t_{4,4} = 11$	ct_8	$t_{2,3} + t_{2,4} = 9$
ct_4	$t_{1,5} + t_{1,6} = 12$	ct_9	$2t_{2,5} + t_{3,3} = 8$
ct_5	$t_{1,9} + t_{1,10} = 7$	ct_{10}	$2t_{4,1} + t_{2,8} = 12$

For instance, the following constraint: $t_{3,7} + t_{3,9} = 11$ means the activities' operation times are tightly linked, that is, increase of activity time associated with one operation (for example, $O_{3,7}$) results in a decrease of the other one (in this case $O_{3,9}$). The set of constraints considered is shown in Table 9.

Therefore, taking into account the above-mentioned assumptions the problem considered now reduces to the question:

Q_2: What values and of what variables T_1, T_2, T_3, and T_4 guarantee the makespan of the projects' portfolio does not exceed a given deadline subject to limits imposed on available amounts of renewable and non-renewable resources as well as NPV > 0?

In order to respond to this question, the values of the following sequences are sought:

$T_1 = (t_{1,1},\ldots, t_{1,10})$,

$T_2 = (t_{2,1},\ldots, t_{2,12})$,

$T_3 = (t_{3,1},\ldots, t_{3,11})$,

$T_4 = (t_{4,1},\ldots, t_{4,13})$

and

$S_1 = (s_{1,1},\ldots, s_{1,10})$,

$S_2 = (s_{2,1},\ldots, s_{2,12})$,

$S_3 = (s_{3,1},\ldots, s_{3,11})$,

$S_4 = (s_{4,1},\ldots, s_{4,13})$.

Taking into account the data assumed, consider the following formulation of the relevant CCS. Given CS = ((X, D), C), where:

- the set of decision variables X, containing:

  the input variables: $U = \{ t_{1,1}, t_{1,2},\ldots, t_{1,lo1}, t_{2,1}, \ldots t_{lp,lolp}, s_{1,1}, s_{1,2},\ldots, s_{1,lo1}, s_{2,1},\ldots, s_{lp,lolp}\}$,

  the output variables: Y = {h, NPV};

- the family of domains D, where the domains of variables T_i, $Tp_{i,j}$, $Tz_{i,j}$, $Dp_{i,j}$, $Cr_{i,j}$, $Cw_{i,j}$, $Tr_{i,j}$, and $Ts_{i,j}$ are determined by Tables 1–8, and variables corresponding to the beginning times of activities $s_{i,j} \in [0,50]$, and variables corresponding to activities' duration times $t_{i,j} \in [1,15]$;
- the set of constraints $C \cup \{ct_1, ct_2,\ldots,ct_{10}\}$ (following Table 9).

The considered problem CS and the question Q_2 can be specified in terms of CCS:

CCS = ((X, D), $C \cup Cy$),

where:

Cy – the set of output constraints corresponding to Q_1: Cy = $\{cy_1, cy_2\}$, cy_1: $h \le 36$; cy_2: NPV > 0.

The result of CCS examination by the consistency-checking procedure is positive, so there exists the solution Vu following all the constraints $C \cup Cy$. The results obtained from the OzMozart implemented procedure consists of the non-empty set of solutions Vu. The first admissible solution has been obtained in 250 s. So, the sequences of obtained activities' operation times are as follows:

$T_1 = (1, 2, 3, 4, 6, 6, 3, 2, 3, 4)$,

$T_2 = (3, 1, 6, 3, 2, 5, 1, 6, 2, 4, 2, 1)$,

$T_3 = (3, 7, 4, 5, 2, 1, 6, 3, 5, 4, 8)$, and

$T_4 = (3, 3, 2, 5, 6, 1, 4, 1, 8, 4, 3, 5, 6)$.

In turn, the sequences of the moments of activities beginning are as follows:

$S_1 = (0, 1, 1, 4, 11, 17, 8, 11, 23, 26)$,

$S_2 = (0, 3, 8, 10, 13, 15, 20, 15, 21, 23, 23, 27)$,

$S_3 = (0, 3, 3, 10, 15, 7, 17, 15, 18, 23, 27)$, and

$S_4 = (0, 0, 3, 5, 5, 3, 3, 10, 11, 4, 11, 19, 24)$.

The NPV index value calculated for projects: P_1, P_2, P_3, and P_4 follow the requirement NPV > 0, that is,

$NPV_{P1} = 0.262$,

$NPV_{P2} = 2.386$,

$NPV_{P3} = 0.86$, and

$NPV_{P4} = 1.339$.

The introduced CP-based reference model provides a formal framework allowing one to formulate project portfolio planning problems in direct and reverse ways. In other words, it provides a base for designing interactive task oriented decision support tools. This offers the possibility of responding to questions such as: What values and of what variables guarantee the production orders will be completed due to assumed values of performance indexes? (besides such standard questions as: What is the project portfolio completion time?).

The main idea behind this approach lies in searching for the conditions guaranteeing the existence of responses to the standard queries as well as for conditions guaranteeing the employed search strategies can be used in online mode for the given size of project planning problems. Therefore, the reference model of decision problems can be seen as a knowledge base kernel of a methodology aimed at designing dedicated and interactive decision support systems.

4 DSS for Production Order Portfolio Scheduling

In multi-project planning, the main focus is on deciding on a schedule for all activities of projects and allocating resources in order to finish projects before their deadlines. One of our objectives is to propose a method that allows generating a schedule for the execution of a set of orders with resource allocation for a given period of time, guaranteeing a solution meeting a set of enterprise specific goals. Furthermore, due to the fact that unpredicted circumstances frequently happen during the execution of orders, it is required to control them in an online mode in order to quickly react to these circumstances. Among the activities of online control, rescheduling the activities of multiple projects and reallocating resources are critical. Therefore, another of our objectives is to develop a method for rescheduling the project portfolio and reallocating resources with the consideration of budget, cash flow, resource capacity, new projects, and so on.

4.1 Structure and Functioning

It seems obvious, that not all behaviors (functionalities) are reachable under constraints imposed by a given system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural constraints. So, since system constraints determine its behavior, both the system structure and the desired behavior have to be considered simultaneously. In that context, our contribution provides a discussion of some solubility issues concerning structural properties providing conditions guaranteeing assumed system behavior (straight problem formulation) as well as behavioral requirements imposing conditions that have to be satisfied by system structure (reverse problem formulation).

Regardless of the character and scope of business activities, a modern enterprise has to build a project-driven development strategy in order to respond to challenges imposed by growing complexity and globalization. Managers need to be able to utilize a modern DSS so as to undertake optimal business decisions in a further strategic perspective of enterprise operations.

The idea behind an interactive interface module employs a navigation multi-board concept shown in Fig. 6. Its solution assumes hybridization of *Drag and Drop, Touch Screen Panel*, and *Virtual Table* technologies. The menu composed of a set of tabs and folders allows one to specify parameters and decision variables describing both the enterprise's capability (e.g., following from its structure and possible ways of work flow organization) and requirements imposed by production orders at hand (determining, for instance, the batch size, production cycles, work in progress, and so on).

Dependent on the kind of decision problem considered, the relevant tabs are selected and structured on the board so as to encompass one of the following problem formulations:

- a straight planning problem (e.g., Is it possible to undertake the given project portfolio under a given resource availability while guaranteeing disturbance-free execution of activities?);
- a reverse planning problem (e.g., Which values of system parameters guarantee that the project portfolio at hand will be completed while following constraint assumed performance index values?);

Note that in the course of interactive solution search, any change in parameters describing:

- an enterprise – results in different values of criteria matching-up production order requirements;
- the criteria specifying production order requirements – results in suggestion of an enterprise structure change.

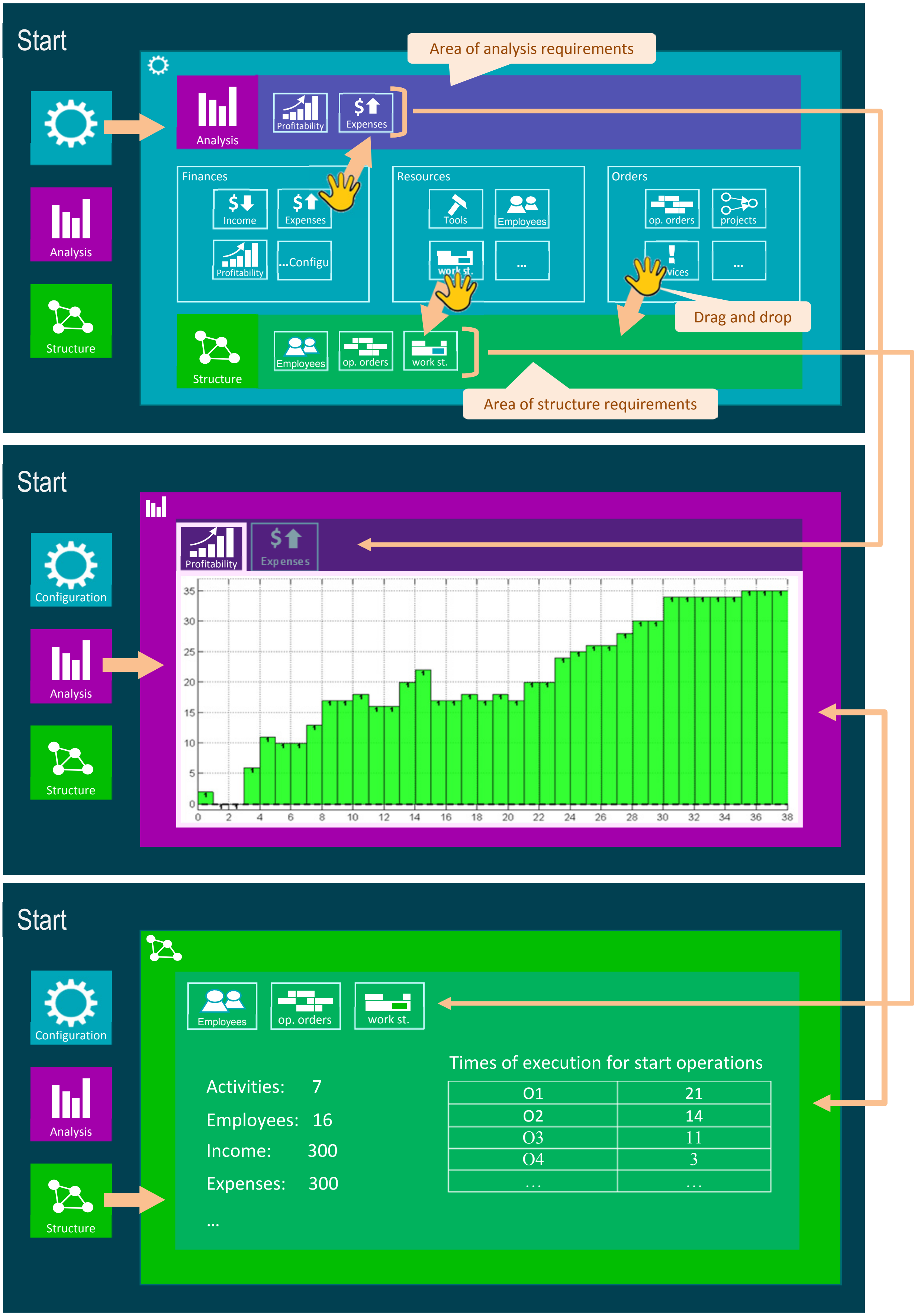

Figure 6. Exemplary multi-board interface configuration composed of folders and/or directories used in the course of decision problem formulation/solution

4.2 Customized oriented production flow scheduling

A production system is given in which a set of projects (the POP) has to be executed. The state of the system encompasses its resource allocation to projects planned for execution. Traditionally stated multi-criteria planning problems formulated as direct ones address standard questions such as: Is it possible to undertake the given project portfolio under a given resource availability while guaranteeing disturbance-free execution of activities? Such a formulation, however, may lead to rejecting projects, which could actually be approved by the system if a satisfactory solution could be found by changing the levels of constraints.

In order to illustrate how it is possible to cope with this kind of problems, let us consider a multi-product job shop where POP Z^1 aimed at device A manufacturing is processed. The device should be completed within the time period $\tau^1 = [360, 420]$ minutes. Consider two newly submitted POPs Z^2 and Z^3 containing the unique production orders Z_1^2 and Z_1^3, respectively. The activity networks encompassing activities orders in the considered POPs are shown in Fig. 7. The possible assignment of activities and their duration times are collected in Table 10. Distinguished activities can be executed on shared resources m_1-m_7. The resources m_7 and m_1 can replace each other, that is, lead to alternative scenarios. The following question is considered: Is it possible to complete devices B and C following the portfolios Z^2 and Z^3 within arbitrary, assumed time periods $\tau^2 = [420, 480]$ and $\tau^3 = [250, 300]$ minutes, respectively?

For the aforementioned data, implemented in CP language OzMozart (Dual Core 2.67 GHz, 2.0 GB RAM), the only admissible solution (admissible schedule $X = \{X^1, X^2, X^3\}$) is shown in Fig. 8; it means the considered devices A, B, and C can be completed within the assumed time periods, that is, τ^1, τ^2, and τ^3, respectively. Due to the implemented scenario, four activities among 28 others have to be realized on resource m_7 instead of m_1. The calculation time required by each scenario is equal to 2 s.

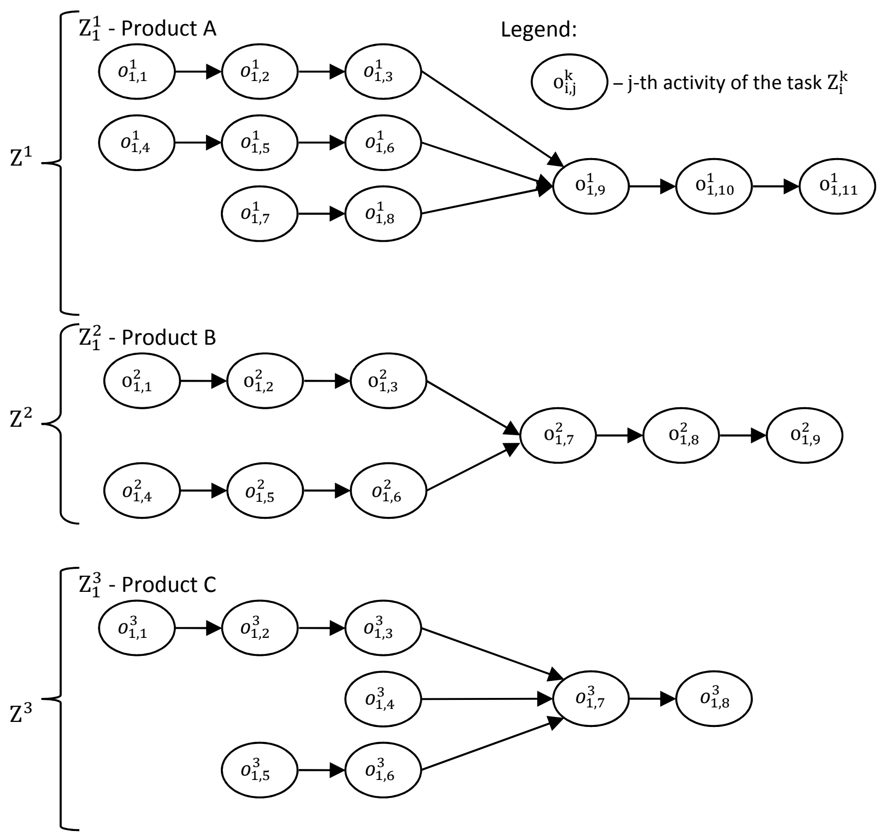

Figure **7.** Activity networks associated with devices A, B, and C

Table 10. Alternative allocations for activities and their duration times

Tasks	Activities	Variants	Resources	Time [min]
Z_1^1	$o_{1,1}^1$	${}^1s_{1,1}^1$	m_5	30
	$o_{1,2}^1$	${}^1s_{1,2}^1$	m_1	50
		${}^2s_{1,2}^1$	m_7	30
	$o_{1,3}^1$	${}^1s_{1,3}^1$	m_2	50
	$o_{1,4}^1$	${}^1s_{1,4}^1$	m_5	30
	$o_{1,5}^1$	${}^1s_{1,5}^1$	m_1	50
		${}^2s_{1,5}^1$	m_7	50
	$o_{1,6}^1$	${}^1s_{1,6}^1$	m_2	50
	$o_{1,7}^1$	${}^1s_{1,7}^1$	m_5	30
	$o_{1,8}^1$	${}^1s_{1,8}^1$	m_3	30
	$o_{1,9}^1$	${}^1s_{1,9}^1$	m_1	30
		${}^2s_{1,9}^1$	m_7	50
	$o_{1,10}^1$	${}^1s_{1,10}^1$	m_2	50
	$o_{1,11}^1$	${}^1s_{1,11}^1$	m_1	50
		${}^2s_{1,11}^1$	m_7	30
Z_1^2	$o_{1,1}^2$	${}^1s_{1,1}^2$	m_5	30
	$o_{1,2}^2$	${}^1s_{1,2}^2$	m_4	30
	$o_{1,3}^2$	${}^1s_{1,3}^2$	m_3	30
	$o_{1,4}^2$	${}^1s_{1,4}^2$	m_5	50
	$o_{1,5}^2$	${}^1s_{1,5}^2$	m_1	90
		${}^2s_{1,5}^2$	m_7	50
	$o_{1,6}^2$	${}^1s_{1,6}^2$	m_2	130
	$o_{1,7}^2$	${}^1s_{1,7}^2$	m_1	30
		${}^2s_{1,7}^2$	m_7	30
	$o_{1,8}^2$	${}^1s_{1,8}^2$	m_2	50
	$o_{1,9}^2$	${}^1s_{1,9}^2$	m_1	50
		${}^2s_{1,9}^2$	m_7	130
Z_1^3	$o_{1,1}^3$	${}^1s_{1,1}^3$	m_1	80
		${}^2s_{1,1}^3$	m_7	50
	$o_{1,2}^3$	${}^1s_{1,2}^3$	m_5	30
	$o_{1,3}^3$	${}^1s_{1,3}^3$	m_3	30
	$o_{1,4}^3$	${}^1s_{1,4}^3$	m_1	50
		${}^2s_{1,4}^3$	m_7	30
	$o_{1,5}^3$	${}^1s_{1,5}^3$	m_2	50
	$o_{1,6}^3$	${}^1s_{1,6}^3$	m_1	50
		${}^2s_{1,6}^3$	m_7	70
	$o_{1,7}^3$	${}^1s_{1,7}^3$	m_4	50
	$o_{1,8}^3$	${}^1s_{1,8}^3$	m_1	30
		${}^2s_{1,8}^3$	m_7	50

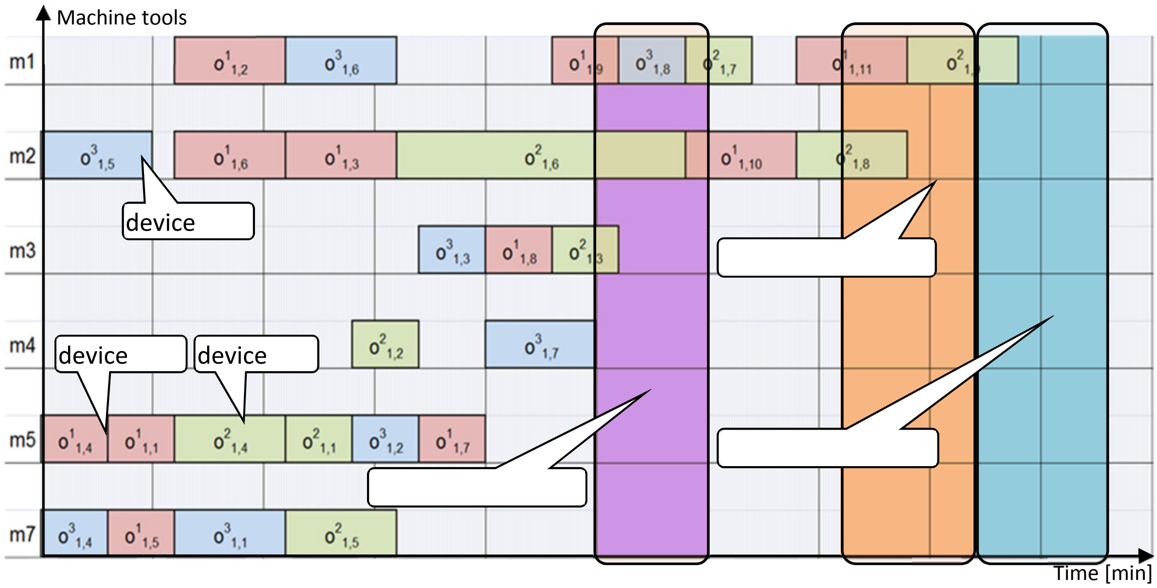

Figure 8. Gantt's chart (schedule $X = \{X^1, X^2, X^3\}$) of production order portfolio execution

5 CONCLUSIONS

Our approach to interactive task oriented decision support tools provides the framework allowing one to take into account both straight and reverse problem formulations. This advantage can be seen as a possibility for responding (besides such standard questions as: Is it possible to complete a given set of production orders at a scheduled project deadline?) to questions such as: What variables' values guarantee the production order makespan follows the assumed deadline? The CP paradigm behind the methodology aimed at designing such tools allows to take into account both the distinct and imprecise characters of the decision variables as well as to consider multi-criteria decision problems.

Better planning, in the manner supported by the proposed approach, can improve companies' competitiveness through satisfying budgetary constraints and improving utilization of resources from a cash-flow perspective. A computer implementation of the proposed methodology should provide a new generation DSS supporting one in cases of online resource allocation and task scheduling as well as production order batching and routing. Such a tool should be especially helpful when actually the processed products' portfolios do not spend all the company's capability reserves, that is, there is a room for additional work order considerations.

6 REFERENCES

[1] Bach I., Bocewicz G., Banaszak Z., Muszyński W. - *Knowledge based and CP-driven approach applied to multi product small-size production flow* [in] *Control and Cybernetics*. Vol. 39, No. 1, 2010, pp. 69-95.

[2] Bach I., Bocewicz G., Banaszak Z. - *Constraint programming approach to time-window and multi-resource-constrained projects portfolio prototyping. Agent and Multi-Agent Systems: Technologies and Applications*. Lecture Notes in Artificial Intelligence 4496, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 578-587.

[3] Bach I., Bocewicz G., Banaszak Z. - *Constraint programming approach to time-window and multi-resource-constrained projects portfolio prototyping. Industrial, Engineering and Other Applications of Applied Intelligent Systems.* IEA/AIE 2008, N.T. Nguyen et al. (ed.), Lecture Notes in Artificial Intelligence 5027, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 767-776.

[4] Bach I., Bocewicz G., Muszyński W., Banaszak Z. - *Knowledge based and CLP-driven approach to multi product small-size production flow* [at]. Carlos P. (ed.) - Proceedings of the 9th International Workshop on Intelligent Manufacturing Systems, 2008, pp. 32-39.

[5] Bach I., Wójcik R. Bocewicz G. - *Projects portfolio prototyping subject to imprecise activities specification* [at] Conference proceedings of 14th International Congress of Cybernetics and Systems

of WOSC – ICCS'08, Wroclaw, Poland, 2008, pp. 261-272.

[6] Badell M., Romero J., Huertas R., Puigjaner L. - *Planning, scheduling and budgeting value-added chains* [in] *Computers and Chemical Engineering*, Vol. 28, 2004, pp. 45-61.

[7] Banaszak Z. - *CP-based decision support for project driven manufacturing. Perspectives in Modern Project Scheduling* [in] *International Series in Operations Research and Management Science*, Springer, Vol. 92, U.S., 2006, pp. 409-437.

[8] Banaszak Z., Bocewicz G. - *Abductive reasoning driven approach to project-like production flow prototyping* [in] *Foundations of Management*, Vol. 1, No. 1, pp. 43-64.

[9] Banaszak Z. A., Zaremba M.B. - *Project-driven planning and scheduling support for virtual manufacturing* [in] *Journal of Intelligent Manufacturing*, Vol. 17, 2006, pp. 641-651.

[10] Banaszak Z., Zaremba M., Muszyński W. - *CP-based decision making for SME*. Preprints of the 16th IFAC World Congress (ed.: P. Horacek, M. Simandl), P. Zitek, DVD, 2005, Prague, Czech Republic.

[11] Barták R. - *Incomplete Depth-First Search Techniques: A Short Survey*. Proceedings of the 6th Workshop on Constraint Programming for Decision and Control, Ed. Figwer J., pp. 7-14.

[12] Bocewicz G., Bach-Dąbrowska I., Banaszak Z. - *Deklaratywne projektowanie systemów komputerowego wspomagania planowania przedsięwzięć*. Akademicka Oficyna Wydawnicza Exit, p. 302.

[13] Bocewicz G., Banaszak Z. - *Declarative approach to cyclic steady state space refinement: periodic process scheduling* [in] *International Journal of Advanced Manufacturing Technology*, Vol. 67, 2013, p. 137-155.

[14] Bocewicz G., Banaszak Z., Nielsen I. - *Multimodal processes prototyping subject to fuzzy operation time constraints*. Proceedings of IFAC Symposium on Information Control in Manufacturing (INCOM 2015), Canada, Ottawa 2015 (in print).

[15] Bocewicz G., Banaszak Z., Wójcik R. - *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach.* Agent and Multi-Agent Systems: Technologies and Applications, Nguyen N.T., Grzech A., Howlett R.J., Jain L.C. (Eds.), Lecture Notes in Artificial Intelligence 4496, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 578-587.

[16] Bubnicki Z. - *Learning processes and logic-algebraic method for the systems with knowledge representation. Systems analysis and management*. PAS, Warsaw, 1999.

[17] Bzdyra K., Banaszak Z., Bocewicz G. - *Multiple project portfolio scheduling subject to mass customized service [in] Progress in Automation, Robotics and Measuring Techniques.* Series: Advances in Intelligent Systems and Computing, Vol. 350, Szewczyk R., Zieliński C., Kaliczyńska M. (ed.), Springer International Publishing 2015, pp. 11-21.

[18] Dang Q.-V., Nielsen I., Steger-Jensen K., Madsen O. - *Scheduling a single mobile robot for part-feeding tasks of production lines* [in] *Journal of Intelligent Manufacturing*, Vol. 25, pp. 1-17.

[19] Groover M. P. - *Automation, Production Systems and Computer-Integrated Manufacturing*. 3rd edition, Prentice Hall, NJ, 2007, p. 840.

[20] Khayat G. E., Langevin A., Riope D. - *Integrated Production and Material Handling Scheduling Using Mathematical Programming and Constraint Programming* [in] *European Journal of Operational Research*, Vol. 175(3), 2006, pp. 1818-1832.

[21] Krenczyk D., Kalinowski K., Grabowik C. - *Integration Production Planning and Scheduling Systems for Determination of Transitional Phases in Repetitive Production. Hybrid Artificial Intelligent Systems*. Corchado E. (ed.), et al., Vol. 7209, 2012, pp. 274-283.

[22] Linderoth T., Savelsbergh. M.W.P. - *A computational study of search strategies in mixed integer programming* [in] *INFORMS Journal on Computing*, 1999, 11, pp. 173-187.

[23] Nielsen I., Bocewicz G., Dung D.A. - *Production and Resource Scheduling in Mass Customization with Dependent Setup Consideration.* Proceedings of the 7th World Conference on Mass Customization, Personalization, and Co-Creation (MCPC 2014), Lecture Notes in Production Engineering, 2014, pp. 461-472.

[24] Parcher, N., Chasemzadech F. - *An integrated framework for project portfolio selection* [in] *International Journal of Project Management*, Vol. 17, No. 4, 1999, pp. 207-216.

[25] Sitek P., Wikarek J. - *Hybrid Solution Framework for Supply Chain Problems. Distributed Computing and Artificial Intelligence* (DCAI 2014), Book Series: Advances in Intelligent Systems and Computing, Vol. 290, 2014, pp. 11-18.

[26] Zimmermann H.J. - *Fuzzy sets theory and its applications*. London: Kluwer Academic Publishers. 1994.