Control and Cybernetics

vol. **39** (2010) No. 1

Knowledge based and CP-driven approach applied to multi product small-size production flow*

by

Irena Bach[1], Grzegorz Bocewicz[1], Zbigniew A. Banaszak[2], Wojciech Muszyński[3]

[1]Department of Computer Science and Management
Technical University of Koszalin, 75-453 Koszalin, Poland

[2]Department of Business Informatics, Warsaw University of Technology, 02-524 Warsaw, Poland

[3]Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology, 50-372 Wrocław, Poland

e-mail: bocewicz@ie.tu.koszalin.pl, Z.Banaszak@wz.pw.edu.pl, wojciech.muszynski@pwr.wroc.pl

Abstract: Constraint Programming (CP) is an emergent software technology for declarative description and effective solving of large combinatorial problems, especially in the area of integrated production planning. In this context, CP can be considered an appropriate framework for development of decision making software, supporting scheduling of multi-robots in a multi-product job shop. The paper deals with the multi-resource problem, in which more than one shared renewable and non-renewable resource type may be required by a manufacturing operation and the availability of each type is time-windows limited. The problem is NP-complete. The aim of the paper is to present a knowledge based and CP-driven approach to multi-robot task allocation providing prompt service to a set of routine queries, stated both in direct and reverse way. Provided examples illustrate the cases with consideration of accurate and uncertain specification of robot and worker operation time.

Keywords: knowledge engineering, modeling, constraint logic programming, scheduling.

1. Introduction

Some industrial processes simultaneously produce different products using the same production resources. For example, in recycling industries, different items are recovered simultaneously from the recycled products. The distribution of

*Submitted: January 2009; Accepted: November 2009

the cumulative demand for each item is known over a finite planning horizon and all unsatisfied demand is fully backlogged (Van Hentenryck, 1991). 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, so as to maximize the profit.

In this context the executives want to know how much a particular production order will cost, which resources are needed, what resource allocation can guarantee due time production order completion, and so on (Banaszak, 2006). So, a dispatcher's needs might be formulated in the form of standard, routine questions, such as: Can the production order be completed before an arbitrary given deadline? What is the production completion time following assumed robots 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 planned orders? Which values of given variables guarantee that production order will be completed following the assumed set of performance indexes?

The problems standing behind such questions belong to the class of so called project scheduling problems. 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 so as to take into account a given performance measure. Such problems belong to NP-complete ones. Therefore, new methods and techniques addressing the impact of real-life constraints on decision making are of great importance, especially for interactive and task oriented Decision Support System (DSS) design (Banaszak, 2006).

Several techniques have been proposed in the past fifty years, including Mixed-integer Linear Programming problem (MILP) (Linderoth and Savelsbergh, 1999), or, more recently, Artificial Intelligence. The latter techniques concentrate mostly on fuzzy set theory and constraint programming frameworks. Constraint Programming/Constraint Logic Programming (*CP*/*CLP*) languages (Barták, 2004; Van Hentenryck, 1991) seem to be well suited to modelling of real-life and day-to-day decision-making processes in an enterprise (Banaszak, 2006). In turn, applications of fuzzy set theory in production management show that most of research on project scheduling has been focused on fuzzy *PERT* and fuzzy *CPM* (Chanas and Komburowski, 1981; Dubois, Fargier and Fortemps, 2003).

The paper touches upon various issues of decision making, while employing the knowledge and *CP* based framework. The proposed approach provides the framework allowing for considering both precise (crisp), and imprecise (fuzzy) data in a unified way and treat them in a unified form of a discrete, constraint satisfaction problem (*CSP*). The framework considered enables treating multi product small-size production flow prototyping as an iterative process of direct and reverse problem resolution.

Moreover, the proposed approach concerns the logic-algebraic method (*LAM*, see Bubnicki, 1999) based and *CP*-driven methodology, aimed at interactive decision making for precise and imprecise data. In this context the paper can be seen as a continuation of previous work concerning project portfolio management, whose objective is to determine the optimal mix and sequencing of proposed projects, while satisfying constraints imposed by the precise and imprecise specification of available resources (Bocewicz, Bach and Banaszak, 2008; Bach, Bocewicz and Banaszak, 2008, 2009).

We first provide an illustrative example of the problem considered, see Section 2, and then we present some details of the modelling framework assumed, in particular, we describe the reference model employed in Section 3. In Section 4, the problem statement is provided, and its *CSP* implementation, as well as the *LAM* based approach to *CSP* resolution is discussed. An illustrative example of the possible application of the approach developed is discussed in Section 5. We conclude with some results and lesson learned in Section 6.

2. Illustrative example of the decision problem

Consider the job shop composed of eight work stations where out of the two semi-products K_1, K_2, two products, W_1 and W_2, are manufactured following the production route P_1, see Fig. 1. At the work stations three kinds of manufacturing operations are considered: decomposition, e.g. disassembly, $\{O_{1,2}, O_{1,4}\}$, composition, e.g. assembly $\{O_{1,5}, O_{1,8}\}$, and processing, e.g. milling $\{O_{1,1}, O_{1,3}, O_{1,6}, O_{1,7}\}$. The workstations are serviced by three robots (ro_1, ro_2, ro_3) and two workers (ro_4, ro_5). At least one robot and/or worker is allocated to each $O_{i,j}$, see Table 1. Since production routes specify an order of operation execution, in further considerations, whenever it does not lead to confusion, the operations will stand for activities and the production routes will be treated as activity networks.

Activities (operations) are non-preemptable. Each activity requires for its processing $dp_{i,j,k}$ units of the k-th renewable resource, and consumes at its beginning a units as well as results at its end in b units of non-renewable resource flow (see Table 2).

Given are activity times as well as associated moments of the relevant resource allocation. This type of decision variables, e.g. activity execution times for robots or workers, can be specified as precise or imprecise ones. Note that, since the amount of common shared resources is limited, their allocation to simultaneously executed activities has to avoid occurrence of closed loop resource requests, i.e. deadlocks.

In this context, the problem of multi-robot task allocation in a multi-product job shop reduces to a class of dispatcher's routine questions, such as: Does a given resource allocation guarantee that production order completion time does not exceed the deadline h and the amount of renewable resources is positive at any moment of time horizon $H = \{0, 1, \ldots, h\}$? Does there exist a resource allo-

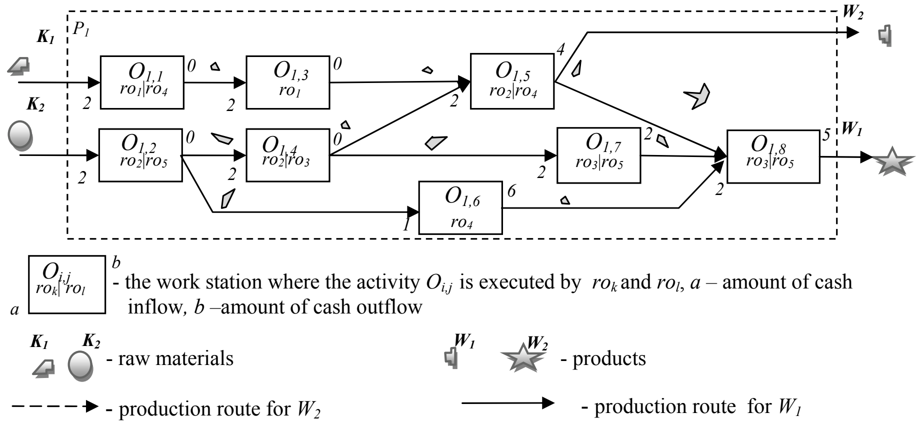

Figure 1. Job shop following the production route P_1

Table 1. Robot and worker allocation to activities of the production route P_1

		$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{1,4}$	$O_{1,5}$	$O_{1,6}$	$O_{1,7}$	$O_{1,8}$
robots	ro_1	1	0	1	0	0	0	0	0
	ro_2	0	1	0	1	1	0	0	0
	ro_3	0	0	0	1	0	0	1	1
workers	ro_4	1	0	0	0	1	1	0	0
	ro_5	0	1	0	0	0	0	1	1

Table 2. Cash inflows and outflows associated with activities

cash \ activity	$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{1,4}$	$O_{1,5}$	$O_{1,6}$	$O_{1,7}$	$O_{1,8}$
outflow	2	2	2	2	2	1	2	2
inflow	0	0	0	0	4	6	2	5

cation such that production order completion time not exceeding the deadline h is guaranteed and amount of renewable resources is positive at any moment of time horizon H? What values and of what variables guarantee that production orders will be completed under the assumed values of performance indexes?

3. Reference model

Consider a reference model, oriented at the following routine question: Do given job shop capabilities and assumed resources allocation guarantee that production order completion time does not exceed the deadline h and the amount of renewable resources is positive at any moment of time horizon H?

Given is the amount lz of renewable discrete resources ro_i specified by: $Ro = (ro_1, ro_2, \ldots, ro_{lz})$. Given are the 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 at the k-th moment of H, specified by $Zo = (zo_1, zo_2, \ldots, zo_{lz})$.

Given is the amount ln of non-renewable resources rn_i specified by: $Rn = (rn_1, rn_2, \ldots, rn_{ln})$. Given are also the amounts zn_i of available non-renewable resources rn_i specified by: $Zn = (zn_1, zn_2, \ldots, zn_{ln})$, where zn_i denotes amount of the resource rn_i available at the beginning of time horizon H.

Decision variables. Given is a set of projects specified by the set of production routes $P = \{P_1, P_2, \ldots, P_{lp}\}$. Each P_i is specified by the set composed of lo_i activities, i.e. $P_i = \{O_{i,1}, \ldots, O_{i,lo_i}\}$, with (Bach, Bocewicz and Banaszak, 2008):

$$O_{i,j} = (x_{i,j}, t_{i,j}, Tp_{i,j}, Tz_{i,j}, Dp_{i,j}, Tr_{i,j}, Ts_{i,j}, Cr_{i,j}, Cs_{i,j}), \tag{1}$$

where:

$x_{i,j}$ — starting time of activity $O_{i,j}$, i.e., time counted from the beginning of time horizon H,

$t_{i,j}$ — duration of activity $O_{i,j}$,

further, in what follows, a general notation is used, in two forms: $XY_{i,j}$ or $Xy_{i,j}$, where X, Y and y take on a number of values; generally, these two forms correspond to the same tables, determined by a triple of indices, in the first case ($XY_{i,j}$) the second index being variable, while in the second case ($Xy_{i,j}$) the third index is variable; thus, in formula (1) we have:

$Tp_{i,j} = (tp_{i,j,1}, tp_{i,j,2}, \ldots, tp_{i,j,lz})$ — the sequence of time moments when activity $O_{i,j}$ requires new amounts of renewable resources, with $tp_{i,j,k}$ — time counted since the moment $x_{i,j}$ the amount $dp_{i,j,k}$ of the k-th resource has been allocated to activity $O_{i,j}$; hence, a resource is allotted to an activity during its execution period: $0 \leq tp_{i,j,k} \leq t_{i,j}$, $k = 1, 2, \ldots, lz$;

$Tz_{i,j} = (tz_{i,j,1}, tz_{i,j,2}, \ldots, tz_{i,j,lz})$ — the sequence of moments, when activity $O_{i,j}$ releases the subsequent resources, $tz_{i,j,k}$ — time counted since the moment $x_{i,j}$ the amount $dp_{i,j,k}$ of the k-th renewable resource was released by activity

$O_{i,j}$; so, it is assumed that a resource is released by an activity during its execution: $0 < tz_{i,j,k} \leq t_{i,j}$ 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 $dp_{i,j,k}$ that are allocated to activity $O_{i,j}$, i.e. $dp_{i,j,k}$ is the amount of the k-th resource, allocated to activity $O_{i,j}$; this is linked with assumption that $0 \leq dp_{i,j,k} \leq zo_k$, $k = 1, 2, \ldots, lz$;

$Tr_{i,j} = (tr_{i,j,1}, tr_{i,j,2}, ..., tr_{i,j,ln})$ — the sequence of time moments, when the determined amounts of subsequent non renewable resources are collected by activity $O_{i,j}$; $tr_{i,j,k}$ — the time counted since the moment $x_{i,j}$ when the amount $dp_{i,j,k}$ of the k-th non renewable resource was released by activity $O_{i,j}$; it is assumed that a resource is collected by activity during its execution: $0 \leq tr_{i,j,k} < t_{i,j}$; $k = 1, 2, \ldots, ln$;

$Ts_{i,j} = (ts_{i,j,1}, ts_{i,j,2}, ..., ts_{i,j,ln})$ — the sequence of moments when the determined amounts of subsequent non renewable resources are generated (released) by activity $O_{i,j}$; $ts_{i,j,k}$ — time counted since the moment $x_{i,j}$ the amount $cs_{i,j,k}$ of the the k-th non renewable resource was generated by activity $O_{i,j}$; it is assumed that the resource is generated during activity execution, but not earlier than its collection started, i.e.: $0 \leq ts_{i,j,k} < t_{i,j}$; $k = 1, 2, \ldots, ln$, as well as $tr_{i,j,k} \leq ts_{j,k}$; $k = 1, 2, \ldots, ln$;

$Cr_{i,j} = (cr_{i,j,1}, cr_{i,j,2}, \ldots, cr_{i,j,ln})$ — the sequence of non-renewable resource amounts consumed by activity $O_{i,j}$; $cr_{i,j,k}$ — the amount of the k-th resource required by the activity $O_{i,j}$, $cr_{i,j,k} \leq 0$; $k = 1, 2, \ldots, ln$; $cr_{i,j,k} = 0$ means that this activity does not consume the k-th resource;

$Cs_{i,j} = (cs_{i,j,1}, cs_{i,j,2}, \ldots, cs_{i,j,ln})$ — the sequence of amounts of non-renewable resources released by activity $O_{i,j}$; $cs_{i,j,k}$ — the amount of the k-th resource released by activity $O_{i,j}$, $cs_{i,j,k} \geq 0$, $k = 1, 2, \ldots, ln$; $cs_{i,j,k} = 0$ means that the activity does not release the k-th resource.

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

- starting times of activities on the project P_i:
  $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,lo_i})$, $0 \leq x_{i,j} < h$, $i = 1, 2, ..., lp$; $j = 1, 2, ..., lo_i$;
- duration of activities on the project P_i:
  $T_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,lo_i})$;
- starting times the j-th resource is allocated to the k-th activity on the project P_i:
  $TP_{i,j} = (tp_{i,1,j}, tp_{i,2,j}, \ldots, tp_{i,k,j}, \ldots, tp_{i,lo_i,j})$;
- starting times the j-th resource is released by the k-th activity on P_i:
  $TZ_{i,j} = (tz_{i,1,j}, tz_{i,2,j}, \ldots, tz_{i,k,j}, \ldots, tz_{i,lo_i,j})$;
- amounts of the j-th resource allotted to the k-th activity on P_i:
  $DP_{i,j} = (dp_{i,1,j}, dp_{i,2,j}, \ldots, dp_{i,k,j}, \ldots, dp_{i,lo_i,j})$;
- the sequence of moments the j-th non renewable resource is collected by the activities of the projects on P_i:
  $TR_{i,j} = (tr_{i,1,j}, tr_{i,2,j}, \ldots, tr_{i,lo_i,j})$;

- the sequence of moments the j-th non renewable resource is released by the activities of the projects on P_i:
  $TS_{i,j} = (ts_{i,1,j}, ts_{i,2,j}, \ldots, ts_{i,lo_i,j})$;
- sequences of amounts of the j-th non-renewable resource consumed by the activities of the project P_i:
  $CR_{i,j} = (cr_{i,1,j}, cr_{i,2,j}, \ldots, cr_{i,lo_i,j})$;
- sequences of amounts of the j-th non-renewable resource involved by the activities of the project P_i:
  $CS_{i,j} = (cs_{i,1,j}, cs_{i,2,j}, \ldots, cs_{i,lo_i,j})$.

Assume that some of chosen execution times are defined roughly, i.e. are treated as fuzzy variables, specified by fuzzy sets. Therefore, the activity $O_{i,j} = (\hat{x}_{i,j}, \hat{t}_{i,j}, Tp_{i,j}, Tz_{i,j}, Dp_{i,j}, Tr_{i,j}, Ts_{i,j}, Cr_{i,j}, Cs_{i,j})$, is specified by the sequences of:

- starting times of activities on the project P_i:
  $\hat{X}_i = (\hat{x}_{i,1}, \hat{x}_{i,2}, ..., \hat{x}_{i,lo_i})$,
- duration of activities on the project P_i:
  $\hat{T}_i = (\hat{t}_{i,1}, \hat{t}_{i,2}, ..., \hat{t}_{i,lo_i})$,

where:

$\hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp})$ — is the sequence determining (consisting of) the sequences $\hat{X}_i$ (where $i = 1, 2, ..., lp$), whose elements are fuzzy starting times of activities $O_{i,j}$,

$\hat{T} = (\hat{T}_1, \hat{T}_2, ..., \hat{T}_{lp})$ — is the sequence determining (consisting of) the sequences $\hat{T}_i$ (where $i = 1, 2, ..., lp$), whose elements are fuzzy duration times of activities $O_{i,j}$,

$Tp_{i,j}, Tz_{i,j}, Dp_{i,j}, Tr_{i,j}, Ts_{i,j}, Cr_{i,j}, Cs_{i,j}$ are the sequences, defined by (1).

Activity order constraints. Let us consider a set of projects P, composed of lo_i precedence and resource constrained, non-preemptable activities that require renewable resources. Assume lz is a number of available renewable discrete resources and the sequences $r_{i,j} = (ro_k, ro_g, \ldots, ro_f)$, $j = 1, \ldots, lo_i$ consist of $ro_k, ro_g, ..., ro_f$, which are elements of Ro and determine resource requirements of the activities $O_{i,j}$. The total number of units of the discrete resource ro_i, $i = 1, \ldots, lz$, is limited by zo_i. The resource can be allotted (and constant within the activity operation time) to activities in arbitrary amount from the set $\{1, \ldots, zo_i\}$. This means that two different resources can be allotted to activity $O_{i,j}$ at different, also overlapping, periods of time.

Note that, since the volume of common shared resources is limited, their allocation to simultaneously executed activities must avoid occurrence of closed loop resource request, i.e. deadlocks. For this purpose, relevant constraints should be imposed (Bach, Bocewicz and Banaszak, 2008).

Project P_i is represented by an activity-on-node network, where an activity corresponds to a node and arcs determine the order of activity execution. Consequently, the following constraints on the order of activities are considered:

- k-th activity follows the i-th one:

$$\hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k}, \tag{2}$$

- k-th activity follows other activities:

$$\hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k}, \hat{x}_{i,j+1} \hat{+} \hat{t}_{i,j+1} \hat{\leq} \hat{x}_{i,k}, ..., \tag{3}$$
$$\hat{x}_{i,j+n} \hat{+} \hat{t}_{i,j+n} \hat{\leq} \hat{x}_{i,k},$$

- k-th activity is followed by other activities:

$$\hat{x}_{i,k} \hat{+} \hat{t}_{i,k} \hat{\leq} \hat{x}_{i,j}, \hat{x}_{i,k} \hat{+} \hat{t}_{i,k} \hat{\leq} \hat{x}_{i,j+1}, ..., \tag{4}$$
$$\hat{x}_{i,k} \hat{+} \hat{t}_{i,k} \hat{\leq} \hat{x}_{i,j+n}$$

The relevant fuzzy arithmetic operations $\hat{+}$, $\hat{\leq}$ are defined in the Appendix. Due to the formulas (a8) and (a12) from the Appendix, any fuzzy constraint C_i (e.g. $\hat{v}_i \hat{<} \hat{v}_l$) can be characterized by a value $E(C_i)$, $E(C_i) \in [0, 1]$. In turn, values $E(C_i)$ allow for determining the level of uncertainty, DE, of reference model constraint satisfaction, i.e. a kind of uncertainty threshold. For instance, $DE = 1$ means that all constraints hold, and $DE = 0.8$ means that they are, altogether, close to satisfaction. The value of DE is determined through formula:

$$DE = \min_{i=1,2,...,lo_c} \{E(C_i)\} \tag{5}$$

where: lo_c is the number of constraints of the reference model.

In the course of decision making, based on constraints, assuming fuzzy variables, an uncertainty threshold should be assumed (e.g. following operator's experience). This means that the decision maker should be able to decide about the membership functions of the decision variables used, as well as the uncertainty thresholds of fuzzy constraints employed.

Renewable resource constraints. Because of limited amount of available discrete renewable resources the constraints protecting against their allocation exceeding available output should also be considered. This means that constraints taking into account imprecise character of such variables as activity operation times $\hat{t}_{i,j}$ and the time instants of activity start, $\hat{x}_{i,j}$, have to be considered. The approach proposed follows the method applied in the case of precise (crisp) variables (Bocewicz, Bach and Banaszak, 2008; Bach, Bocewicz and Banaszak, 2009). Note that the cases, when resource availability limits are exceeded, follow from conflicts resulting from wrong resource allocation, leading to the closed loops of resource requests. So, in order to be able to develop the constraints allowing for avoiding violation of the assumed limits of the available amounts of renewable resources, consider the following functions f_k^* and g_k^*, defining available amounts of the k-th resource, required at the moment $\hat{v}$:

- $f_k^*(\hat{v}, \hat{X}, DEf_k)$ — defines the required number of the k-th resource units at the fuzzy moment $\hat{v}$, which depends on assumed fuzzy moments of beginning of activities, $\hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp})$; it is assumed that the set H is the domain of the membership function $\mu(v)$ of the variable $\hat{v}$, determining $DEf_k \in [0,1]$. So, for the sake of simplicity in further considerations, the following phrase will be used: "the variable $\hat{v}$ defined on the set H''. Note that the function f_k^* is calculated for a given level of uncertainty. This means that, for instance, if $DEf_k = 0.8$, then at the moment $\hat{v}$ the number of required units of the k-th resource does not exceed the number f_k^* with uncertainty level equal to 0.8.

- $g_k^*(\hat{v}, DEg_k)$ — defines, with uncertainty level $DEg_k \in [0,1]$, the available amount of the k-th resource at the moment $\hat{v}$. We assume that the available amount of the k-th resource is constant over entire time horizon H, i.e. $g_k^*(\hat{v}, DEg_k) = gv_k$, where $gv_k = const$, $\forall v \in H$.

Therefore, the above functions f_k^* and g_k^* can be treated as some generalization of functions f_g and g_k, defining, respectively, the quantity of required and available units of the k-th resource in case variables considered are precise (Bocewicz, Bach and Banaszak, 2008; Bach, Bocewicz and Banaszak, 2009). The occurrence of the closed loops of resource requests implies the inequality $f_k(v_b, X) > g_k(v_b)$. So, assuming the variables are treated as imprecise, the inequality $f_k(v_b, X) > g_k(v_b)$ can be seen as a consequence of occurrence of a closed loop of resource request with uncertainty level $f_k^*(\hat{v}, \hat{X}, DEf_k) > gv_k$. Such generalization leads to the following Property 1:

PROPERTY 1 *The inequality $f_k^*(\hat{v}, \hat{X}, DEf_k) > gv_k$ is a necessary condition for the occurence of closed loops of resource request with the uncerntainty level DEf_k.*

Moreover, assuming that activities cannot be stopped (suspended) during their execution, the following Lemma 1 holds:

LEMMA 1 *If resource allocation to activities in the set of projects P at the moment $\hat{v}$ satisfies the condition $f_k^*(\hat{v}, \hat{X}, DEf_k) \leq gv_k$, $\forall k \in \{1, 2, \dots, lz\}$, for the assumed $\hat{X}, \hat{T}, TP_{i,j}, TZ_{i,j}, DP_{i,j}, H$, then execution of respective activities does not lead to the deadlocks with uncertainty level $DEf = \min_{k\in\{1,2,\dots,lz\}}\{DEf_k\}$.*

Proof. The proof follows directly from Property 1. Due to Property 1 a deadlock cannot occur with uncertainty level $DEf = \min_{k\in\{1,2,\dots,lz\}}\{DEf_k\}$ in the fuzzy moment $\hat{v}$ only in the case when condition $f_k^*(\hat{v}, \hat{X}, DEf_k) > gv_k$ holds for at least one resource. So, in case for each resource ($\forall k \in \{1, 2, ..., lz\}$) Property 1 does not hold (i.e., condition $f_k^*(\hat{v}, \hat{X}, DEf_k) > gv_k$ does not hold) the closed cycle occurs with uncertainty level $DEf = \min_{k\in\{1,2,\dots,lz\}}\{DEf_k\}$. When Property 1 does not hold, condition $f_k^*(\hat{v}, \hat{X}, DEf_k) \leq gv_k$ holds. So, the closed cycle (i.e. a deadlock) does not occur with uncertainty level DEf, in case condition $f_k^*(\hat{v}, \hat{X}, DEf_k) \leq gv_k$ holds for each resource ($\forall k \in \{1, 2, ..., lz\}$). ■

PROPERTY 2 *If at any fuzzy moment $\hat{v}$ in the time horizon H the condition $f_k^*(\hat{v}, \hat{X}, DEf_k) \leq gv_k$ holds, $\forall k \in \{1, 2, ..., lz\}$, then execution of activities is deadlock-free with uncertainty level $DEf = \min_{k\in\{1,2,\dots,lz\}}\{DEf_k\}$.*

Due to the above introduced assumptions, functions f_k^* and g_k^* have the following form:

- function $f_k^*(\hat{v}, \hat{X}, DEf_k)$:

$$f_k^*(\hat{v}, \hat{X}, DEf_k) = \sum_{i=1}^{lp} \sum_{j=1}^{lo_i} \Big[dp_{i,j,k} \cdot \hat{\hat{1}}(\hat{v}, \hat{x}_{i,j} \hat{+} tp_{i,j,k}, \hat{x}_{i,j} \hat{+} tz_{i,j,k}, DEf_k) \Big] \quad (6)$$

where:

$tp_{i,j,k} < tz_{i,j,k}$, lp — is the number of projects,

lo_i — is the number of activities in the i-th project,

$dp_{i,j,k}$ — is the volume of the k-th resource engaged by the activity $O_{i,j}$,

$\hat{\hat{1}}(\hat{v}, \hat{a}, \hat{b}, DEf_k) = \hat{1}(\hat{v}, \hat{a}, DEf_k) - \hat{1}(\hat{v}, \hat{b}, DEf_k)$ — is a fuzzy function of time resource occupation, where $\hat{1}(\hat{v}, \hat{a}, DEf_k)$ is a fuzzy unit step function; the fuzzy unit step function corresponds to the precise unit step function $1(v)$.

The following fuzzy unit step function is considered:

$$\hat{1}(\hat{v}, \hat{a}, DEf_k) = f, \quad f \in \{0, 1\}, \quad DEf_k \in [0, 1], \quad (7)$$

where f is a precise number, for which the value of the following expression equals DEf_k:

$$\big[(\hat{v} \hat{\geq} \hat{a}) \vee (f = 0)\big] \wedge [(\hat{v} \hat{<} \hat{a}) \vee (f = 1)] \quad (8)$$

i.e.

$$E\left[\big[(\hat{v} \hat{\geq} \hat{a}) \vee (f = 0)\big] \wedge [(\hat{v} \hat{<} \hat{a}) \vee (f = 1)]\right] = DEf_k \quad (9)$$

For further considerations, let us note that the expressions $\beta_1 \wedge \beta_2$; $\beta_1 \vee \beta_2$; $\neg\beta_1$ correspond to the following formulas:

$$E(\beta_1 \wedge \beta_2) = E(\beta_1) \cdot E(\beta_2), \quad (10)$$
$$E(\beta_1 \vee \beta_2) = E(\beta_1) + E(\beta_2) - E(\beta_1) \cdot E(\beta_2), \quad (11)$$
$$E(\neg\beta_1) = 1 - E(\beta_1). \quad (12)$$

Formula (9) leads, therefore, to:

$$E(\hat{v} \hat{\geq} \hat{a}) + E(f = 0) \cdot [1 - 2E(\hat{v} \hat{\geq} \hat{a})] = DEf_k \quad (13)$$

and finally:

$$E(f = 0) = \frac{DEf_k - E(\hat{v} \hat{\geq} \hat{a})}{1 - 2E(\hat{v} \hat{\geq} \hat{a})}. \quad (14)$$

Taking into account $E(f=0) = 1 - f$, $f \in \{0,1\}$, we obtain:

$$f = 1 - \frac{DEf_k - E(\hat{v}\hat{\geq}\hat{a})}{1 - 2E(\hat{v}\hat{\geq}\hat{a})}. \tag{15}$$

Finally, the fuzzy unit step function has the following form:

$$\hat{1}(\hat{v}, \hat{a}, DEf_k) = 1 - \frac{DEf_k - E(\hat{v}\hat{\geq}\hat{a})}{1 - 2E(\hat{v}\hat{\geq}\hat{a})}, \tag{16}$$

where $\hat{1}(\hat{v}, \hat{a}, DEf_k) \in \{0,1\}$, $DEf_k \in [0,1]$.

Let us treat $\hat{a}$ of the function $\hat{\hat{1}}(\hat{v}, \hat{a}, \hat{b}, DEf_k)$ as a characteristic point. Since in formula (6) the considered function is determined by

$$\hat{\hat{1}}(\hat{v}, \hat{x}_{i,j}\hat{+}tp_{i,j,k}, \hat{x}_{i,j}\hat{+}tz_{i,j,k}, DEf_k),$$

hence the corresponding characteristic point is determined by the fuzzy moments $\hat{x}_{i,j}\hat{+}tp_{i,j,k}$ when $dp_{i,j,k}$ units of the k-th resource are alloted to the activity $O_{i,j}$. In further considerations, such points are called characteristic points of the function $f_k^*(\hat{v}, \hat{X}, DEf_k)$. Note that increase of the value of function $f_k^*(\hat{v}, \hat{X}, DEf_k)$ can occur only at characteristict points of this function.

- function $g_k^*(\hat{v}, DEg_k)$:

$$g_k^*(\hat{v}, DEg_k) = gv_k = zo_{k,1} = zo_{k,2} = \cdots = zo_{k,h} \tag{17}$$

where:

gv_k — the available amount of the k-th renewable resource,

$zo_{k,i}$ — the element of sequence zo_k determining the limited amount of the k-th renewable resource at the i-th moment of H.

Therefore, since (6) and (17) hold, the following Theorem 1 is also true.

Theorem 1 *Given the set of projects P, consider assumptions imposed by the reference model regarding the specification of activities $\hat{X}$, $\hat{T}$, $TP_{i,j}$, $TZ_{i,j}$, $DP_{i,j}$, H, and functions $f_k^*(\hat{v}, \hat{X}, DEf_k)$ and $g_k^*(\hat{v}, DEg_k)$, following the formulae* (6), (17). *If for any moment $\hat{v}$ in the assumed time horizon H and for each k-th resource, $k \in \{1, 2, \ldots, lz\}$, conditions* (18) *hold, then the execution of the set of projects will be deadlock free with uncertainty level $DEf = \min_{k\in\{1,2,\ldots,lz\}}\{DEf_k\}$.*

$$\begin{cases}
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{1,1}\hat{+}tp_{1,1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,1,1})]\leq zo_{k,1}\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{1,2}\hat{+}tp_{1,2,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,1,2})]\leq zo_{k,1}\\
\qquad\qquad\qquad\qquad\ldots\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{1,lo_1}\hat{+}tp_{1,lo_1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,1,lo_1})]\leq zo_{k,1}\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{2,1}\hat{+}tp_{2,1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,2,1})]\leq zo_{k,1}\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{2,2}\hat{+}tp_{2,2,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,2,2})]\leq zo_{k,1}\\
\qquad\qquad\qquad\qquad\ldots\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{2,lo_2}\hat{+}tp_{2,lo_2,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,2,lo_2})]\leq zo_{k,1}\\
\qquad\qquad\qquad\qquad\ldots\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{lp,1}\hat{+}tp_{lp,1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,lp,1})]\leq zo_{k,1}\\
\qquad\qquad\qquad\qquad\ldots\\
\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}[dp_{i,j,k}\cdot\hat{\hat{1}}(\hat{x}_{lp,lo_{lp}}\hat{+}tp_{lp,lo_{lp},k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_{k,lp,lo_{lp}})]\leq zo_{k,1}
\end{cases} \tag{18}$$

where: $DEf_k = \min_{k\in\{1,2,\ldots,lz\}}\{DEf_{k,1,1}, DEf_{k,1,12}, ..., DEf_{k,lp,lo_{lp}}\}$.

Proof. Due to the forms of functions $f_k^*(\hat{v},\hat{X},DEf_k)$ and $g_k^*(\hat{v},DEg_k)$ (see formulae (6), (17)), the condition $f_k^*(\hat{v},\hat{X},DEf_k)\leq gv_k$ can be stated as follows:

$$\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\Big[dp_{i,j,k}\cdot\hat{\hat{1}}(\widehat{v},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},DEf_k)\Big]\leq zo_{k,1}. \tag{19}$$

According to Property 2, in order to avoid deadlocks, condition (19) must hold for each moment $\hat{v}$ in time horizon H. Note that conditons (18) are generalizations of inequality (19) for the cases, when variable $\hat{v}$ follows the values of characteristic points of function $f_k^*(\hat{v},\hat{X},DEf_k)$. Due to (6), a change of states of projects considered can occur in P only for characteristic points. So, if for each k-th resource, $k\in\{1,2,\ldots,lz\}$, formula (19) holds in characteristic points, then it holds for each $v\in H$. This observation, due to Property 2 and Lemma 1, guarantees that the execution of activities within the time horizon (i.e. the execution of the entire set of projects) is deadlock-free with uncertainty level $DEf=\min_{k\in\{1,2,\ldots,lz\}}\{DEf_k\}$. ∎

Non-renewable resource constraints. Because of limited amount of the available discrete non-renewable resources, constraints protecting against their allocation exceeding available output should also be considered, also because non-renewable resources can be allotted at the same time to different activities in a way causing occurrence of closed loop resource requests, i.e. deadlocks.

By analogy to renewable resources we shall formulate the constraints determining resource allocation guaranteeing deadlock-free execution of activities. Assume that the number of required and obtained units of the k-th non-renewable resource is determined by functions

$$b_k^*(\hat{v}, \hat{X}, DEn_k \text{ and } m_k^*(\hat{v}, \hat{X}, DEn_k),$$

respectively:

- $b_k^*(\hat{v}, \hat{X}, DEn_k)$ — function defining total amount of required units of the k-th resource at the moment $\hat{v}$, taking into account the moments the activities begin, $\hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp})$. Of course, the values of function b_k^* are determined for a given level of uncertainty, $DEn_k \in [0, 1]$.
- $m_k^*(\hat{v}, \hat{X}, DEn_k)$ — function defining total amount of the used up units of the k-th resource at the moment $\hat{v}$, taking into account the moments the activities begin, $\hat{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp})$. Of course, the values of function m_k^* are determined for a given level of uncertainty, $DEn_k \in [0, 1]$.

It can be proven that for deadlock to occur, the inequality $b_k(v, X) > m_k(v, X)$ must hold for each $v \in H$. This observation can be generalized by replacing accurate variables by the imprecise (fuzzy) ones, i.e. the inequality $b_k^*(\hat{v}, \hat{X}, DEn_k) > m_k^*(\hat{v}, \hat{X}, DEn_k)$ results as a consequence of the closed cycle of resource requests, with uncertainty level DEn_k. Therefore, we are led to Property 3.

Property 3 *The inequality $b_k^*(\hat{v}, \hat{X}, DEn_k) > m_k^*(\hat{v}, \hat{X}, DEn_k)$ is a necessary condition for the occurrence of closed loops of the non-renewable resource requests (i.e. deadlocks) with the uncertainty level DEn_k.*

Property 3 allows for proving the following Lemma 2:

Lemma 2 *Assume that the reference model is specified by $\hat{X}$, $\hat{T}$, $TP_{i,j}$, $TZ_{i,j}$, $DP_{i,j}$, H. If for an allocation of non-renewable resources to project activities P at the moment $\hat{v}$ the condition $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$ holds, $\forall k \in \{1, 2, ..., ln\}$, then realization of these activities is deadlock-free with uncertainty level $DEn = \min_{k \in \{1,2,...,\text{ln}\}}\{DEn_k\}$.*

Proof. Proof follows directly from Property 3. Due to Property 3 a deadlock can occur with uncertainty level DEn_k, only in the case when at the fuzzy moment $\hat{v}$ for at least one k-th resource the condition $b_k^*(\hat{v}, \hat{X}, DEn_k) > m_k^*(\hat{v}, \hat{X}, DEn_k)$ holds. This means that if for no resource k ($\forall k \in \{1, 2, ..., lz\}$) Property 3 holds (i.e. the condition $b_k^*(\hat{v}, \hat{X}, DEn_k) > m_k^*(\hat{v}, \hat{X}, DEn_k)$ does not hold), then the execution of activities is deadlock-free with uncertainty level $DEn = \min_{k \in \{1,2,...,\text{ln}\}}\{DEn_k\}$. In case Property 3 does not hold, then condition $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$ holds. This implies that execution of activities is deadlock-fee if for each resource k ($\forall k \in \{1, 2, ..., lz\}$) condition $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$ holds. ∎

Property 4 *If at any fuzzy moment* $\hat{v}$ *in the time horizon* H *the condition* $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$ $\forall k \in \{1, 2, ..., ln\}$ *holds, then execution of activities is deadlock-free with uncertainty level* $DEn = \min_{k \in \{1,2,\ldots,\mathrm{ln}\}}\{DEn_k\}$.

Due to the assumptions imposed on the reference model, the functions included in the inequality $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$ are of the following form:

$$b_k^*(\hat{v}, \hat{X}, DEn_k) = \sum_{i=1}^{lp} \sum_{j=1}^{lo_i} \left[cr_{i,j,k} \cdot \hat{1}(\hat{v}, \hat{x}_{i,j} \hat{+} tr_{i,j,k}, DEn_k)\right] \tag{20}$$

where: $tr_{i,j,k}$ — the moment of the k-th non-renewable resource allocation to an activity, lp — the number of projects, lo_i — the number of i-th project activities, $cr_{i,j,k}$ — the number of the k-th non-renewable resource units used by activity $O_{i,j}$, $\hat{1}(\hat{v}, \hat{a}, DEn_k)$ — the fuzzy unit step function (16).

The value of the function $\hat{1}(\hat{v}, \hat{a}, DEn_k)$ is called characteristic point of the fuzzy unit step function. In formula (20), where the fuzzy unit step functions are added, the characteristic points determine the moments $(\hat{x}_{i,j} \hat{+} tr_{i,j,k})$, when the activities $O_{i,j}$ require assumed quantities $(cr_{i,j,k})$ of the k-th resource units. Such points are called characteristic points of the function $b_k^*(\hat{v}, \hat{X}, DEn_k)$.

By analogy to the case of renewable resources, changes of the function $b_k^*(\hat{v}, \hat{X}, DEn_k)$ may occur only in such characteristic points.

$$m_k^*(\hat{v}, \hat{X}, DEn_k) = \sum_{i=1}^{lp} \sum_{j=1}^{lo_i} \left[cs_{i,j,k} \cdot \hat{1}(\hat{v}, \hat{x}_{i,j} \hat{+} ts_{i,j,k}, DEn_k)\right] + zn_k \tag{21}$$

where: $ts_{i,j,k}$ — moment, when the k-th non-renewable resource is allocated to activity $O_{i,j}$, lp — the number of projects, lo_i — the number of the i-th project activities, $cs_{i,j,k}$ — the number of units of the k-th non-renewable resource, used by activity $O_{i,j}$, $\hat{1}(\hat{v}, \hat{a}, DEn_k)$ — the fuzzy unit step function (16).

By analogy to $b_k^*(\hat{v}, \hat{X}, DEn_k)$, the values of $(\hat{x}_{i,j} \hat{+} ts_{i,j,k})$ are called the characteristic points of the function $m_k^*(\hat{v}, \hat{X}, DEn_k)$.

With functions $b_k^*(\hat{v}, \hat{X}, DEn_k)$, $m_k^*(\hat{v}, \hat{X}, DEn_k)$ (see (20) and (21)) and Property 4, the condition $b_k^*(\hat{v}, \hat{X}, DEn_k) \leq m_k^*(\hat{v}, \hat{X}, DEn_k)$, can be transformed to the following inequality:

$$\begin{aligned} zn_k &- \sum_{i=1}^{lp} \sum_{j=1}^{lo_i} \left[cr_{i,j,k} \cdot \hat{1}(\hat{v}, \hat{x}_{i,j} \hat{+} tr_{i,j,k}, DEn_k)\right] \\ &+ \sum_{i=1}^{lp} \sum_{j=1}^{lo_i} \left[cs_{i,j,k} \cdot \hat{1}(\hat{v}, \hat{x}_{i,j} \hat{+} ts_{i,j,k}, DEn_k)\right] \geq 0 \end{aligned} \tag{22}$$

where: lp and lo_i are the same as before, $\hat{1}(\hat{v}, \hat{a}, DEn_k)$ is the fuzzy unit step function (16).

It follows from Lemma 2 that execution of activities is deadlock-free when condition (22) holds for each moment $\hat{v}$ in time horizon H. Values of the functions $b_k^*(\hat{v}, \hat{X}, DEn_k)$ and $m_k^*(\hat{v}, \hat{X}, DEn_k)$ can change only for variable values corresponding to characteristic points. The variable $\hat{v}$ in formula (22) can then be replaced by a set of relevant characteristic points. Consider the characteristic points of the function $b_k^*(\hat{v}, \hat{X}, DEn_k)$, for which the left hand side of inequality (22) decreases. Then, the following Theorem 2 holds.

THEOREM 2 *Given the set of projects P, take assumptions imposed by the reference model regarding specification of activities $\hat{X}$, $\hat{T}$, $TP_{i,j}$, $TZ_{i,j}$, $DP_{i,j}$, H, and functions $b_k^*(\hat{v}, \hat{X}, DEn_k)$, and $m_k^*(\hat{v}, \hat{X}, DEn_k)$, according to* (20), (21). *If for each moment $\hat{v}$ in the assumed time horizon H and for each k-th resource, $k \in \{1, 2, \ldots, ln\}$, conditions* (23) *hold, then execution of the set of projects will be deadlock free with uncertainty level $DEn = \min_{k\in\{1,2,\ldots,\mathrm{ln}\}}\{DEn_k\}$.*

$$\begin{cases} zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{1,1}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,1,1})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{1,1}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,1,1})\right] \geq 0 \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{1,2}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,1,2})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{1,2}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,1,2})\right] \geq 0 \\ \qquad\qquad\qquad ... \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{1,lo_1}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,1,lo_1})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{1,lo_1}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,1,lo_1})\right] \geq 0 \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{2,1}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,2,1})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{2,1}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,2,1})\right] \geq 0 \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{2,2}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,2,2})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{2,2}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,2,2})\right] \geq 0 \\ \qquad\qquad\qquad ... \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{2,lo_2}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,2,lo_2})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{2,lo_2}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,2,lo_2})\right] \geq 0 \\ \qquad\qquad\qquad ... \\ zn_k - \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cr_{i,j,k}\cdot \hat{1}(\hat{x}_{lp,lo_{lp}}, \hat{x}_{i,j}\hat{+}tr_{i,j,k}, DEn_{k,lp,lo_{lp}})\right] \\ +\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[cs_{i,j,k}\cdot \hat{1}(\hat{x}_{lp,lo_{lp}}, \hat{x}_{i,j}\hat{+}ts_{i,j,k}, DEn_{k,lp,lo_{lp}})\right] \geq 0 \end{cases} \tag{23}$$

for $k = 1, 2, \ldots, ln$, where ln is the number of non-renewable resources, and $DEn_k = \min_{k\in\{1,2,\ldots,\mathrm{ln}\}}\{DEn_{k,1,1}, DEn_{k,1,12}, ..., DEn_{k,lp,lo_{lp}}\}$.

Proof. Conditions (23) can be seen as a generalization of inequality (22) at moments $\hat{v}$, corresponding to characteristic points of functions $b_k^*(\hat{v}, \hat{X}, DEn_k)$, $m_k^*(\hat{v}, \hat{X}, DEn_k)$. Due to formulae (20), (21), the change of the state of projects P can take place only at such characteristic points. Therefore, when inequality

(22) holds for each characteristic point (see function $b_k^*(\hat{v}, \hat{X}, DEn_k)$), such that the left hand side of this inequality decreases its value, then it holds for any moment $\hat{v}$ in the time horizon H. This observation, due to Property 4 (and Lemma 2) provides the guarantee that activity execution in whole time horizon (i.e. project execution taking into account non-renewable resources) is deadlock-free with a given uncertainty level. ∎

4. Problem formulation

The introduced model provides the formal framework making it possible to state the problem considered.

4.1. Problem statement

Given are: time horizon $H = \{0, 1, \ldots, h\}$, set of projects (specified by the set of production routes) P, set of resources and their availabilities, Zo within H, initial amount of the non-renewable resource, zo_1, precise and imprecise decision variable values, treated as fuzzy numbers, i.e. the sequences $\hat{T}_i$, $\hat{T}P_{i,j}$, $\hat{T}Z_{i,j}$. For these data, the following questions should be answered:

(1) Does a given resources allocation guarantee that the production order makespan does not exceed the deadline h and that the amount of each non-renewable resource is positive at any moment of time horizon H? Response to this question results in the determination of the sequences $\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp}$.

(2) Do there exist resource allocations such that production order makespan does not exceed the deadline h and the amount of each non-renewable resource is positive at any moment of time horizon H? Response to this question results in determination of the sequences: $\hat{T}_1, \hat{T}_2, ..., \hat{T}_{lp}$.

The questions stated above correspond to the direct and reverse problems of the multi-product scheduling, respectively.

4.2. Constraint satisfaction problem

Since a constraint can be treated as a relation among several variables, each one taking a value in a given (usually discrete) domain, the idea of constraint programming (CP) approach to problem solving can be employed. More formally, CP is a framework for solving combinatorial problems specified by pairs: (a set of variables and associated domains, a set of constraints restricting the possible combinations of the variable values). In this context, the constraint satisfaction problem (CSP) (Barták, 2004) is defined as follows:

$$CS = ((A, D), C), \tag{24}$$

where:

$A = \{a_1, a_2, \ldots, a_g\}$ — a finite set of discrete decision variables,

$D = \{D_i \mid D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,j}, \ldots, d_{i,ld}\},\ i = 1, \ldots, g\}$ — a family of finite domains,

$C = \{C_i \mid i = 1, \ldots, L\}$ — a finite set of constraints limiting the domains of variables.

The solution to the CS is a vector $(d_{1,i}, d_{2,k}, \ldots, d_{n,j})$ with coordinates satisfying each constraint of the set C. Of course, solutions considered are admissible.

The inference mechanism consists of the following two components: constraint propagation and variable distribution. Constraint propagation uses constraints actively to prune the search space. The aim of the propagation techniques, i.e., local consistency checking, is to reach certain level of consistency in order to accelerate search procedures by drastically reducing the size of the search tree (Banaszak, 2006). In general case, however, the consistency techniques are incomplete. For instance, in the problem $CS = (\{a_1, a_2, a_3\}, \{\{0,1\}, \{0,1\}, \{0,1\}\}, \{a_1 \neq a_3, a_1 \neq a_2, a_2 \neq a_3\})$ constraints propagation is not effective and domains of variables are not reduced. Moreover, an admissible solution does not exist.

4.3. Knowledge base representation

The CSP can be seen as a well suited representation of the knowledge base. Let us assume that the knowledge base (KB), describing a system, is represented in the form of the sets $\mathbb{U}$, $\mathbb{W}$, $\mathbb{Y}$, defining some system properties $U \in \mathbb{U}$, $W \in \mathbb{W}$, $Y \in \mathbb{Y}$. U consists of input variables, Y consists of output variables, and W consists of auxiliary variables. The knowledge specifying the properties of the system under consideration is described in the form of a set of facts $F(U, W, Y)$. Facts $F(U, W, Y)$ are propositions encompassing the relationships (i.e., constraints) between individual variables of U, W, Y.

The decision maker is usually faced with the problem whether a query he is interested in is properly, or not, addressed in the context of the DSS at hand. So, in the case of query: what sufficient conditions guarantee the existence of an admissible solution? the relevant decision problem can be formulated as follows. Given are sets of input $U = \{u_1, ..., u_n\}$, output $Y = \{y_1, ..., y_m\}$, and auxiliary $W = \{w_1, ..., w_k\}$ variables, where u_i, y_i belong to domains Du_i, Dy_i, Dw_i (where: $\mathbb{U} = Du_1 \times Du_2 \times ... \times Du_n$, $\mathbb{Y} = Dy_1 \times Dy_2 \times ... \times Dy_m$, $\mathbb{W} = Dw_1 \times Dw_2 \times ... \times Dw_k$) and $F(U)$, $F(Y)$ are the sets of constraints (properties), linking variables from different domains. The decision problem consists in finding $R \subset \mathbb{U} \times \mathbb{W} \times \mathbb{Y}$ such that $F(U)$ implies that $F(U) \Rightarrow F(Y)$ holds.

Due to the logic-algebraic method (LAM), Bubnicki (1999), the following solution is considered:

$$R_u = S_{u1} \setminus S_{u2}, \tag{25}$$

$$S_{u1} = \{U : w(F(U, W, Y)) = 1,\ w(F(Y)) = 1,\ U \in \mathbb{U}\}, \tag{26}$$

$$S_{u2} = \{U : w(F(U, W, Y)) = 1,\ w(F(Y)) = 0,\ U \in \mathbb{U}\}, \tag{27}$$

where:

$w(F(.))$ — the logical value of the fact $F(.)$.

$w(F(.)) = \begin{cases} 1 \text{ if } F(.) \text{ holds} \\ 0 \text{ if } F(.) \text{ does not holds} \end{cases}$

The set S_{u1} consists of those elements of U, for which all facts of sets $F(U, W, Y)$ and $F(Y)$ hold. The set S_{u2}, in turn, consists of those elements of U, for which all facts from the set $F(U, W, Y)$ are true, and at least one fact from the set $F(Y)$ does not hold. Therefore, $R_u \neq \emptyset$ and $R_u = \emptyset$ denote existence, and lack of an answer to the query, respectively. In other words, the response DO NOT KNOW is not allowed.

Consequently, the CSP considered can be determined as follows:

$$CS = ((U \cup Y \cup W, D),\ \{w(F(U, W, Y)) = 1\}), \tag{28}$$

where $D = \{D_U \cup D_Y \cup D_W\}$ is the set of values of input (U) and output (Y) variables, and $w(F(U, W, Y)) = 1$ denotes a set of facts, $\{w(F_1(U, W, Y)) = 1, \ldots, w(F_K(U, W, Y)) = 1\}$.

Finally, relation R, i.e. solution to the problem CS, is obtained from the following equations:

$$CS_{Su1} = ((U \cup Y \cup W, D), \{w(F(U, W, Y)) = 1\}, w(F(Y)) = 1\}), \tag{29}$$

$$CS_{Su2} = ((U \cup Y \cup W, D), \{w(F(U, W, Y)) = 1\}, w(F(Y)) = 0\}). \tag{30}$$

The set $R_u = S_{u1} \backslash S_{u2}$, where sets S_{u1} and S_{u2} are solutions to problems (29) and (30), includes alternative solutions, for which the implication $F(U) \Rightarrow F(Y)$ holds. For a detailed discussion of the advantages of method proposed see Banaszak, Bocewicz and Bach (2008). We shall provide here just a short overview:

- The CSP framework provides a declarative way of problem specification, without a guarantee, though, that any feasible solution exists (constraint propagation techniques cannot guarantee reduction of variable domains nor that any admissible solution exists),
- The CSP framework allows for taking into account constraints composed of variables not contained by other constraints (i.e. the queries as to whether a given subset of variables implies another one cannot be considered),
- The CSP framework is useless for queries checking whether a given subset of variables implies another one (i.e. the case when the same preconditions imply mutually contradicting post-conditions can be considered),
- The LAM provides a framework for knowledge base representation (forward and backward reasoning, i.e. the direct and the reverse problems can be considered),
- The LAM can be directly implemented in the CSP framework (meaning that queries such as whether a given subset of variables implies another one can be considered as well),

- The LAM implemented in CSP framework allows using commercially available CP/CLP platforms (that is, decision problems can be easily formulated in a declarative knowledge base in a uniform way and solved with guarantee that the response DO NOT KNOW will not occur), such as Oz/Mozart (Schulte, Smolka and Wurtz, 1998), Ilog (Puget, 1994).

5. Illustrative example

Consider the project P_1 composed of eight activities (Fig. 1), where W_1 and W_2 are manufactured. The problem of production flow prototyping can be seen as an iterative process of adjustment and evaluation of decision variables.

Case of the direct problem 1. The activity times are treated as fuzzy variables and determined by z-cuts: $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,8})$:

$$\hat{t}_{1,1} = (\{[1,3],[2,3],[3,3]\},\{0;0.5;1\}),\quad \hat{t}_{1,2} = (\{[2,6],[3,5],[4,4]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,3} = (\{[1,3],[1,2],[1,1]\},\{0;0.5;1\}),\quad \hat{t}_{1,4} = (\{[3,5],[3,4],[3,3]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,5} = (\{[2,4],[3,4],[4,4]\},\{0;0.5;1\}),\quad \hat{t}_{1,6} = (\{[1,5],[2,4],[3,3]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,7} = (\{[2,4],[2,3],[2,2]\},\{0;0.5;1\}),\quad \hat{t}_{1,8} = (\{[5,5],[5,5],[5,5]\},\{0;0.5;1\}),$$

Five different renewable resources $ro_1, \ldots, ro_5$ are used. Resource allocation follows Table 1. Hence, $DP_{i,j} = (dp_{i,1,j}, dp_{i,2,j}, \ldots, dp_{i,8,j})$:

$$DP_{1,1} = (1,0,1,0,0,0,0,0),\quad DP_{1,2} = (0,1,0,1,1,0,0,0),$$
$$DP_{1,3} = (0,0,0,1,0,0,1,1),\quad DP_{1,4} = (1,0,0,0,1,1,0,0,),$$
$$DP_{1,5} = (0,1,0,0,0,0,1,1).$$

It is assumed that the instants of resource allocation and release follow the instants of activity beginning and completion. Therefore, $tp_{1,j,k} = 0$, $j = 1,2,\ldots,8$, $k = 1,2,\ldots,5$. Assumed are the following sequences: $\hat{tz}_{i,j,k} = \hat{t}_{1,j}$, $j = 1,2,\ldots,8$, $k = 1,2,\ldots,5$. ($\hat{tz}_{i,j,k}$ means fuzzy variable corresponding to $tz_{i,j,k}$, as well as $Zo = (zo_1, \ldots, zo_5)$, such that $zo_1 = zo_2 = \cdots = zo_5 = 1$. Given are also: the discrete time horizon $H = \{0,1,\ldots,20\}$ and the uncertainty threshold $DE \geq 0.8$.

There is one non-renewable resource rn_1, e.g. money, with the initial amount $zn_1 = 8$. The cash outflows associated with activities are given by $CR_{1,1} = (2,2,2,2,2,1,2,2)$. In turn, the cash inflows associated to activities are given by $CS_{1,1} = (0,0,0,0,4,6,2,5)$.

The question considered: Does there exist a production schedule with makespan not exceeding the given deadline of 20 units of time and for which the amount of cash is positive at any moment of time horizon H? concerns the values of $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,8})$, assuming the moments $\hat{x}_{i,j}$ are fuzzy numbers with triangle membership functions.

Therefore, the problem considered can be seen as a CSP (see the formulae (24)), where the moments $\hat{x}_{i,j}$ of beginning of operations play the role of fuzzy decision variables. The set of constraints C consists: activity order constraints, (2), (3), (4), renewable resource constraints (18), and non-renewable resource constraints (23). Due to the approach proposed in Section 4.3 the Logic-Algebraic Method (LAM) was implemented for solution search. The sought values $\hat{x}_{i,j}$ belong to the set R_u, (25), determined by CS_{Su1} and CS_{Su2}, (29), (30) (note that decision variables $\hat{x}_{i,j}$ correspond to variables $u_i \in U$ in (26), (27)).

The CSP considered has been implemented in OzMozart system (Schulte, Smolka and Wurtz, 1998). In 30 seconds of AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1.00 GB the response found to the question: Does there exist a production schedule with makespan not exceeding the given deadline of 20 units of time and for which the amount of cash is positive at any moment of time horizon H? was negative.

However, in the case of a similar question assuming extension of the considered deadline to 21 units of time the admissible solution $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,8})$ has been found in 300 seconds, see Fig. 2, where:

$$\begin{aligned}
\hat{x}_{1,1} &= (\{[0,0],[0,0],[0,0]\},\{0;0.5;1\}),\\
\hat{x}_{1,2} &= (\{[0,0],[0,0],[0,0]\},\{0;0.5;1\}),\\
\hat{x}_{1,3} &= (\{[2,4],[3,4],[4,4]\},\{0;0.5;1\}),\\
\hat{x}_{1,4} &= (\{[6,6],[6,6],[6,6]\},\{0;0.5;1\}),\\
\hat{x}_{1,5} &= (\{[10,12],[10,11],[10,10]\},\{0;0.5;1\}),\\
\hat{x}_{1,6} &= (\{[2,4],[3,4],[4,4]\},\{0;0.5;1\}),\\
\hat{x}_{1,7} &= (\{[10,12],[10,11],[10,10]\},\{0;0.5;1\}),\\
\hat{x}_{1,8} &= (\{[16,16],[16,16],[16,16]\},\{0;0.5;1\}).
\end{aligned}$$

Case of the reverse problem. Besides the assumptions considered in the former case, let us assume that the activity times are not known, but are linked through following constraints:

$$\begin{aligned}
&C_1 : \hat{t}_{1,8} \hat{+} \hat{t}_{1,3} \hat{=} \hat{6}, \quad C_2 : \hat{t}_{1,6} \hat{+} \hat{t}_{1,7} \hat{=} \hat{5},\\
&C_3 : \hat{t}_{1,1} \hat{+} \hat{t}_{1,4} \hat{=} \hat{6}^*, \quad C_4 : \hat{t}_{1,2} \hat{+} \hat{t}_{1,5} \hat{=} \hat{8},
\end{aligned}$$

where:

$$\begin{aligned}
&\hat{6} = (\{[6,8],[6,7],[6,6\},\{0;0.5;1\}), \quad \hat{5} = (\{[3,9],[4,7],[5,5\},\{0;0.5;1\}),\\
&\hat{6}^* = (\{[4,8],[5,7],[6,6\},\{0;0.5;1\}), \quad \hat{8} = (\{[4,10],[6,9],[8,8]\},\{0;0.5;1\}).
\end{aligned}$$

The constraints are relations $a\hat{t}_{i,j} \hat{+} b\hat{t}_{k,l} \hat{=} \hat{c}$, defined using the operators given in the Appendix.

Five different renewable resources $ro_1, \ldots, ro_5$ are used. Resource allocation follows Table 1. Assume $Dp_{i,j}$, $tp_{i,j,k}$, $\hat{t}z_{i,j,k}$, Z_o are defined as in the direct problem 1 before. Given the discrete time horizon $H = \{0, 1, \ldots, 20\}$, and the

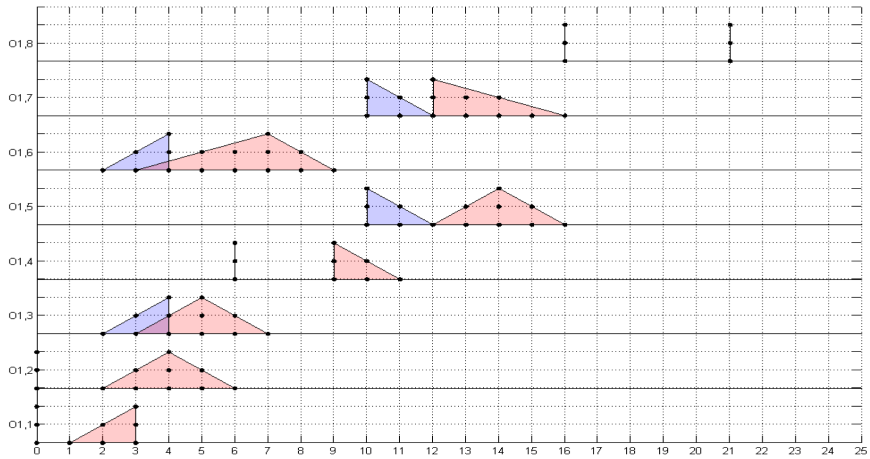

Figure 2. Admissible solution for time horizon H ={0,1,. . . ,21}

uncertainty threshold $DE \geq 0.8$ the question considered is: What activity times $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,8})$ (if any) guarantee that projects will be completed within the time horizon and the amount of cash will be positive at any moment of time horizon H?

Assume that activity times $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,8})$ and moments, when activities begin, $\hat{x}_{i,j}$, are characterized by triangle fuzzy numbers. The constraints on the order of activities, (2), (3), (4), resource use conflicts (18), (23), and timing relations (C_1, C_2, C_3, C_4 have been implemented in OzMozart (Schulte, Smolka and Wurtz, 1998). The first admissible solution $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,8})$, was obtained within 30 minutes with AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1.00 GB. The activity times are treated as fuzzy variables and determined by z-cuts ($\alpha = \{0; 0.5; 1\}$):

$$\hat{t}_{1,1} = (\{[1,3],[2,3],[3,3]\},\{0;0.5;1\}), \quad \hat{t}_{1,2} = (\{[2,6],[3,5],[4,4]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,3} = (\{[5,5],[5,5],[5,5]\},\{0;0.5;1\}), \quad \hat{t}_{1,4} = (\{[3,5],[3,4],[3,3]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,5} = (\{[2,4],[3,4],[4,4]\},\{0;0.5;1\}), \quad \hat{t}_{1,6} = (\{[2,4],[2,3],[2,2]\},\{0;0.5;1\}),$$
$$\hat{t}_{1,7} = (\{[1,5],[2,4],[3,3]\},\{0;0.5;1\}), \quad \hat{t}_{1,8} = (\{[1,1],[1,2],[1,3]\},\{0;0.5;1\}).$$

Case of the direct problem 2. The activity times are treated as fuzzy variables and determined by z-cuts: $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,8})$ from the previous problem. Given the discrete time horizon $H = \{0, 1, \ldots, 20\}$, the uncertainty threshold $DE \geq 0.8$, and the amount of available cash, $rn_1 = 8$, we consider the question: Does there exist a production schedule makespan which does not exceed the given deadline and the amount of cash is positive at any moment of time horizon

H? This question concerns the values $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,8})$, assuming the moments $\hat{x}_{i,j}$ are fuzzy numbers with triangle membership function.

First admissible solution $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,8})$ (see Fig. 3) was obtained within 300 seconds, with $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,8})$:

$$\hat{x}_{1,1} = (\{[0,0],[0,0],[0,0]\},\{0;0.5;1\}), \quad \hat{x}_{1,2} = (\{[0,0],[0,0],[0,0]\},\{0;0.5;1\})$$
$$\hat{x}_{1,3} = (\{[4,4],[4,4],[4,4]\},\{0;0.5;1\}), \quad \hat{x}_{1,4} = (\{[6,6],[6,6],[6,6]\},\{0;0.5;1\}),$$
$$\hat{x}_{1,5} = (\{[11,11],[11,11],[11,11]\},\{0;0.5;1\}),$$
$$\hat{x}_{1,6} = (\{[3,5],[4,5],[5,5\},\{0;0.5;1\}),$$
$$\hat{x}_{1,7} = (\{[10,12],[10,11],[10,10]\},\{0;0.5;1\}),$$
$$\hat{x}_{1,8} = (\{[14,16],[15,16],[16,16]\},\{0;0.5;1\}).$$

Requirements concerning intuitively comprehensible decision making imply the transformation of the fuzzy schedule obtained (see Fig. 3a) into the crisp one, e.g. by providing results with the membership grade ≥ 0.5 (see Fig. 3b)).

6. Concluding remarks

The proposed approach to task allocation in a multi-product job shop provides a framework allowing for taking into account both the direct and the reverse problem statements. The advantage can be seen in the possibility of answering, besides the standard questions, like: *Is it possible to complete a given set of production orders within a given deadline?* also the questions like: *What values of decision variables guarantee that execution of production orders follows the assumed values of performance indexes?* The examples provided illustrate the reference model implementation in the constraint programming environment, as well as capabilities of its use in the course of production flow control problem solution. Some experimental results (Bach, Muszyński and Bocewicz, 2008), confirm feasibility and validity of the approach proposed for the scale of real-life problems that the Small and Medium Enterprises usually face.

Moreover, besides the time-window constraints, imposed on renewable and non-renewable resources, the approach proposed provides the framework allowing for accounting for both an accurate and an uncertain specification of activity times of robots and workers. The study can also be considered as a contribution to project-driven production flow management applied in make-to-order manufacturing as well as for prototyping of the virtual organization structures.

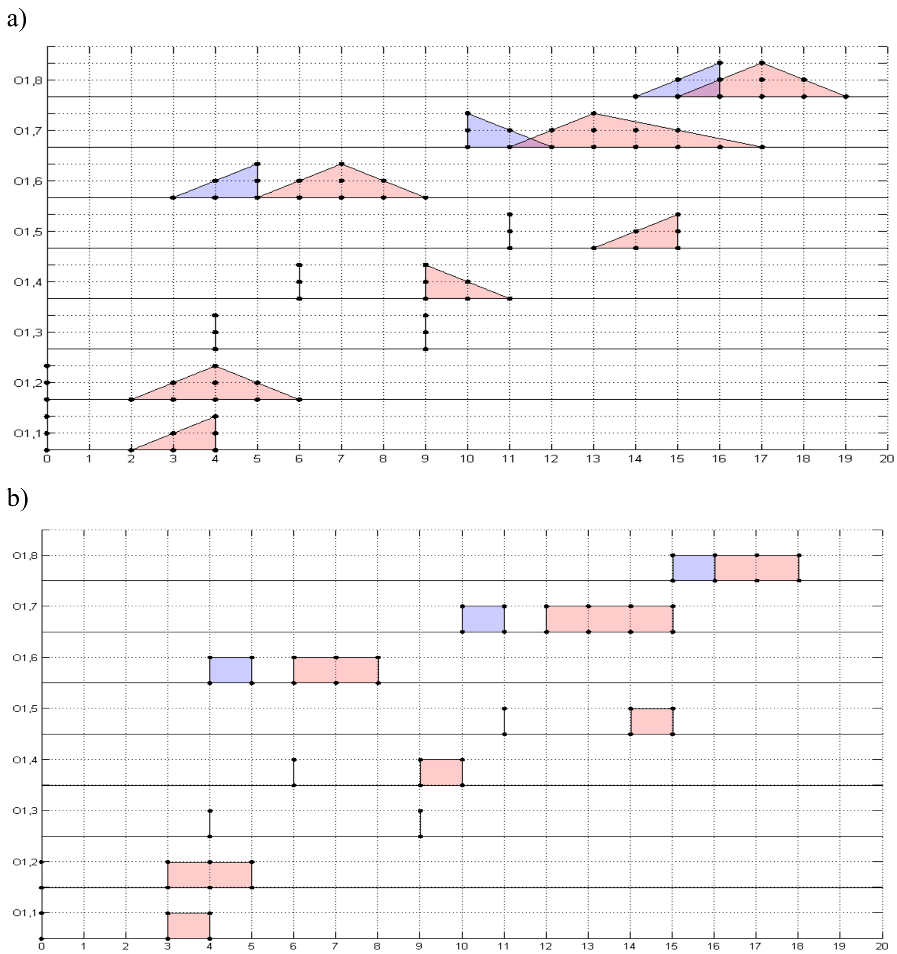

Figure 3. Admissible solution for time horizon $H = \{0,1,\ldots,20\}$

References

Bach, I., Bocewicz, G. and Banaszak, Z. (2008) Constraint programming approach to time-window and multiresource-constrained projects portfolio prototyping. In: N. T. Nguyen et al., eds., *Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2008.* **LNAI 5027**, Springer-Verlag, Berlin-Heidelberg, 767–776.

Bach, I., Bocewicz, G. and Banaszak, Z. (2009) Knowledge based and CLP-driven approach to multi product small-size production flow. *Decision Making in Manufacturing and Services*, **2** (1–2), 5-32.

Bach, I., Muszyński, W. and Bocewicz, G. (2008) CP approach to project portfolio prototyping. In: *New trends in data base management.* WNT, Gliwice, 387–399.

Banaszak, Z. (2006) CP-based decision support for project-driven manufacturing. In: J. Józefowska and J. Węglarz, eds., *Perspectives in Modern Project Scheduling.* International Series in Operations Research and Management Science, **92**, Springer Verlag, New York, 409–437.

Banaszak, Z., Bocewicz, G. and Bach, I. (2008) CP-driven Production Process Planning in Multiproject Environment. In: *Decision Making in Manufacturing and Services* **2** (1–2), 5-32.

Barták, R. (2004) Incomplete Depth-First Search Techniques: A Short Survey. In: J. Figwer, ed., *Proc. 6th Workshop on Constraint Programming for Decision and Control*, 7–14.

Bocewicz, G., Bach, I. and Banaszak, Z. (2008) Decision support tool for project portfolio prototyping. *Applied Computer Science (Performance evaluation models of manufacturing systems)* **4** (2), 43–62.

Bubnicki, Z. (1999) Learning processes and logic-algebraic method for the systems with knowledge representation. *Systems analysis and management.* Polish Academy of Sciences, Warsaw (in Polish).

Chanas, S. and Komburowski, J. (1981) The use of fuzzy variables in PERT. *Fuzzy Sets and Systems*, **5** (1), 11–19.

Dubois, D., Fargier, H. and Fortemps, P. (2003) Fuzzy scheduling: Modeling flexible constraints vs. coping with incomplete knowledge. *EJOR* **147**, 231–252.

Linderoth, T. and Savelsbergh, M.W.P. (1999) A computational study of search strategies in mixed integer programming. *INFORMS Journal on Computing* **11**, 173–187.

Piegat, A. (1999) *Fuzzy modelling and control.* Exit, Warsaw (in Polish).

Puget, J.F. (1994) A C++Implementations of CLP. *Proceeding of SPICS 94.*

Schulte, Ch., Smolka, G. and Wurtz, J. (1998) *Finite Domain Constraint Programming in Oz.* DFKI OZ documentation series, German Research Center for Artificial Intelligence, Saarbruecken, Germany.

Van Hentenryck, P. (1991) Constraint Logic Programming. *Knowledge Engineering Review* **6**, 151–194.

Appendix A

Imprecise variables, specified by fuzzy sets and defined by a convex membership function $\mu_i(v)$ can be characterized by α-cuts (Piegat, 1999), and then by pairs:

$$(A_i, \alpha_i), \tag{A1}$$

where:

$A_i = \{A_{z_i,1}, A_{z_i,2}, ..., A_{z_i,l_{cut}}\}$ — finite set of discretized α-cuts, furthercalled z-cuts,

$\alpha_i = \{\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,l_{cut}}\}$ — finite set of α-cut levels, $\alpha_{i,1} = 0$, $\alpha_{i,l_{cut}} = 1$,

l_{cut} — the number of z-cuts,

Z-cut $A_{z_{i,k}}$ — a set of values corresponding to levels $\alpha_{i,k} : A_{z_i,k} = \{v : v \in N, \mu_i(v) \geq \alpha_{i,k}\}$, represented by the following formulae in short:

$$A_{z_i,k} = [a_{i,k}, b_{i,k}]_N \, . \tag{A2}$$

where: $a_{i,k}$, $b_{i,k}$ are the smallest and the highest values of the k-th z-cut, $a_{ik}, b_{i,k} \in N$.

The z-cut can be seen as a discretized form of the α-cut, i.e. $A_{z_i,k} = A_{\alpha_i,k} \bigcap N$, see Fig. A1.

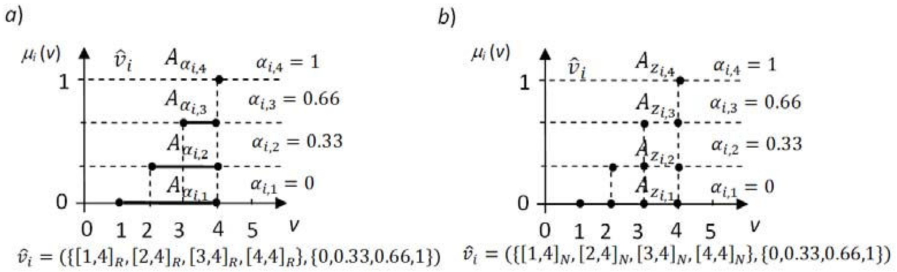

Figure A1. Fuzzy set $\hat{v}_i$ specified by: a) α-cuts, b) discretized α-cuts, i.e., z-cuts

Note that in the assumed specification the distinct values are represented as singletons.

Imprecise character of decision variables, e.g. $\hat{x}_{i,j}$,$\hat{t}_{i,j}$, implies imprecise character of constraints, in which they appear, which, in turn, can be considered as a consequence of implementation of assumed operations. Therefore, consider the set of fuzzy operations: "$\hat{=}$", "$\hat{<}$", "$\hat{>}$", mapping standard algebraic operations, such as: $=$, $\neq$, $<$, $>$, $\geq$, $\leq$. Of course, the considered fuzzy operations, linking two fuzzy variables $\hat{v}_i$, $\hat{v}_l$ have to follow the condition

$$E(\hat{v}_i \hat{<} \hat{v}_l) + E(\hat{v}_i \hat{=} \hat{v}_l) + E(\hat{v}_i \hat{>} \hat{v}_l) = 1 \tag{A3}$$

where: $E(a)$ is the fuzzy logic value of the proposition a, $E(a) \in [0, 1]$

In order to define fuzzy operations used for description of the deadlock avoidance conditions, (18), (23), the auxiliary sets $\hat{v}_i^L$, $\hat{v}_i^*$, $\hat{v}_i^P$ and $\hat{v}_l^L$, $\hat{v}_l^*$, $\hat{v}_l^P$ are defined, as well as the concept of size of fuzzy variable, S_i, and the sizes of subsets S_i^L, S_i^P, S_l^L, S_l^P, S^* of S_i.

For each pair of fuzzy variables $\hat{v}_i$, $\hat{v}_l$, defined by $\{(\mu_i(v), v)\}\ \forall v \in K_i$, where K_i is the domain of variable $\hat{v}_i$, the following sets can be distinguished: $\hat{v}_i^L$, $\hat{v}_i^*$, $\hat{v}_i^P$, $\hat{v}_l^L$, $\hat{v}_l^*$, $\hat{v}_l^P$. For instance, for the set $\hat{v}_l$ the following subsets can be determined:

$\hat{v}_i^L$ — the set of elements v smaller than all elements from $\hat{v}_l$,

$\hat{v}_i^*$ — the set of elements shared with $\hat{v}_l$,

$\hat{v}_i^P$ — the set of elements v bigger than all elements from $\hat{v}_l$.

The sets $\hat{v}_i^L$, $\hat{v}_i^*$, $\hat{v}_i^P$ are defined as follows:

$$v_i^L = \{(\mu_i^L(v), v)\}, \quad \forall v \in K_i, \tag{A4}$$

where:

$$\mu_i^L(v) = \begin{cases} \mu_i(v) - \mu_l(v) \quad \text{if } \mu_i(v) \geq \mu_l(v), \quad v < w_{\min} \\ 0 \quad \text{if } \mu_i(v) < \mu_l(v) \ \text{ and } v < w_{\min}) \ \text{ or } v \geq w_{\min} \end{cases}$$

$$w_{\min} = \min\{K_w\}, K_w = \{v : v \in K_i, \mu_l(v) = 1\}$$

$$v_i^* = \{(\mu_i^*(v), v)\}, \quad \forall v \in K_i, \tag{A5}$$

where: $\mu_i^*(v) = \min\{\mu_i(v), \mu_l(v)\}$.

$$v_i^P = \{(\mu_i^P(v), v)\}, \quad \forall v \in K_i, \tag{A6}$$

where:

$$\mu_i^P(v) = \begin{cases} \mu_i(v) - \mu_l(v) \ \text{ if } \mu_i(v) \geq \mu_l(v), \quad v > w_{\min} \\ 0 \ \text{ if } (\mu_i(v) < \mu_l(v) \ \text{ and } v > w_{\min}) \ \text{ or } v \leq w_{\max} \end{cases}$$

$$w_{\max} = \max\{K_w\}, \quad K_w = \{v : v \in K_i, \mu_l(v) = 1\}.$$

Subsets $\hat{v}_l^*$, $\hat{v}_l^P$, corresponding to fuzzy variable $\hat{v}_l$ are defined in the same way.

For each fuzzy variable $\hat{v}_i$, $\hat{v}_l$ and the corresponding subset $\hat{v}_i^L$, $\hat{v}_i^*$, $\hat{v}_i^P$, $\hat{v}_l^L$, $\hat{v}_l^*$, $\hat{v}_l^P$ an associated size value can be determined. For instance, the size value S_i corresponding to the fuzzy variable $\hat{v}_i$, and specified in terms of z-cuts can be defined as:

$$S_i = \sum_{k=1}^{l_{cut}} \|A_{z_i,k}\|, \tag{A7}$$

where: $\|A_{z_i,k}\|$ — the number of elements of the set $A_{z_i,k}$.

In a similar way, the values S_i^L, S_i^*, S_i^P, S_l^L, S_l^*, S_l^P, corresponding to the sets $\hat{v}_i^L$, $\hat{v}_i^*$, $\hat{v}_i^P$, $\hat{v}_l^L$, $\hat{v}_l^*$, $\hat{v}_l^P$ are defined.

In the case considered, equality $S_i^* = S_l^*$ holds for the given $\hat{v}_i^*$, $\hat{v}_l^*$, because the decision variables $\hat{v}_i$, $\hat{v}_l$ belong to the time domain. Therefore, for the sake of simplicity, in further considerations, sizes S_i^*, S_l^* are denoted by the same symbol S^*.

Given fuzzy variables $\hat{v}_i$, $\hat{v}_l$, consider algebraic-like fuzzy operations following the condition (A3). Fuzzy logic value of the proposition $\hat{v}_i \hat{=} \hat{v}_l$ is defined by:

$$E(\hat{v}_i \hat{=} \hat{v}_l) = \frac{2S^*}{S_i + S_l}, \tag{A8}$$

where S_i — the size of $\hat{v}_i$, S_l — the size of $\hat{v}_l$, S^* — the size of the common part of sets $\hat{v}_i$, $\hat{v}_l$. Fuzzy logic value of the proposition $\hat{v}_i \hat{<} \hat{v}_l$ is defined by:

$$E(\hat{v}_i \hat{<} \hat{v}_l) = \frac{S_i^L + S_l^P}{S_i + S_l}, \tag{A9}$$

where: S_i — the size of $\hat{v}_i$, S_l — the size of $\hat{v}_l$, S_i^L — the size of $\hat{v}_i^L$, S_l^P — the size of $\hat{v}_i^P$.

Fuzzy logic value of the proposition $\hat{v}_i \hat{>} \hat{v}_l$ is defined by

$$E(\hat{v}_i \hat{>} \hat{v}_l) = \frac{S_i^P + S_l^L}{S_i + S_l}, \tag{A10}$$

while the fuzzy logic value of the proposition $\hat{v}_i \hat{\geq} \hat{v}_l$ is defined by

$$E(\hat{v}_i \hat{\geq} \hat{v}_l) = \frac{2S^* + S_i^P + S_l^L}{S_i + S_l}, \tag{A11}$$

and the fuzzy logic value of the proposition $\hat{v}_i \hat{\leq} \hat{v}_l$ is defined by

$$E(\hat{v}_i \hat{\leq} \hat{v}_l) = \frac{2S^* + S_i^L + S_l^P}{S_i + S_l}, \tag{A12}$$

Formulae (A8)-(A12) allow for designing the constraints describing basic relations among two fuzzy variables, i.e. equal, less than, greater than, less or equal, and greater or equal.

In order to take into account other constraints, e.g. including crisp variables, fuzzy operations, such as fuzzy addition and fuzzy subtraction have to be employed as well. The definitions of relevant operations, "$\hat{+}$", "$\hat{-}$", can be found in Piegat (1999).