scheduling, constraint programming, knowledge engineering, decision making

Grzegorz BOCEWICZ[1], Mariusz MĄDRY [2], Zbigniew BANASZAK[1]

KNOWLEDGE BASED AND CLP-DRIVEN APPROACH TO MULTI-ROBOT TASK ALLOCATION FOR MULTI-PRODUCT JOB SHOP

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 that context, CP can be considered as an appropriate framework for development of decision making software supporting scheduling of multi-robot in a multi-product flow shop. The paper deals with multi-resource problem in which more than one shared renewable resource type may be required by manufacturing operation and the availability of each type is time-windows limited. The problem belongs to a class of NP-complete ones. The aim of the paper is to present a knowledge based and CLP-driven approach to multi-robot task allocation framework providing a prompt service to a set of routine queries stated both in straight and reverse way. Provided example concentrates on the first case taking into account both an accurate and an uncertain specification of robots operation time..

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 common characteristic in these industries is that items are produced simultaneously with specified, or variable, productions. The distribution of the cumulative demand for each item is known order over a finite planning horizon and all unsatisfied demand is fully backlogged [15].

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 maximization of profit.

In that context executives want to know how much a particular production order will cost, what resources are needed, what resources allocation can guarantee due time production order

[1] Koszalin University of Technology, Dept. of Computer Science and Management ul. Śniadeckich 2, 75-354 Koszalin, Poland, zbigniew.banaszak@tu.koszalin.pl

[2] Vocational Higher Educational School in Sulechów, Polytechnic Institute, Computer Technology Division, Armii Krajowej 51, 66-100 Sulechów, Poland, e-mail: m.madry@pwsz.sulechow.pl

completion, and so on [2]. So, a dispatcher's needs might be formulated in a form of standard, routine questions, such as: Does the production order can 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) resources availability while guaranteeing disturbance-free execution of the already executed orders? What values and of what variables guarantee the production order will completed following assumed set of performance indexes?

Because the most companies have to manage various production orders which share a pool of constrained resources and taking into account various objectives at the same time the above stated questions can be reformulated in the multi-product job shop context, i.e., the job shop producing simultaneously different kind of items. From that point of view the problems standing behind of the quoted questions belong to the class of so called project scheduling ones. 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 performance measure. Such problems belong to NP-complete ones. Therefore, the new methods and techniques addressing the impact of real-life constraints on the decision making is of great importance, especially for interactive and task oriented DSSs designing [3].

Several techniques have been proposed in the past fifty years, including Mixed Integer Linear Programming [9], Branch-and-Bound [5] or more recently Artificial Intelligence. The last sort of techniques concentrate mostly on fuzzy set theory and constraint programming frameworks. Constraint Programming/Constraint Logic Programming (CP/CLP) languages [4], [15] seems to be well suited for modeling of real-life and day-to-day decision-making processes in an enterprise [2].

In turn, applications of fuzzy set theory in production management [16] shows that most of the research on project scheduling has been focused on fuzzy PERT and fuzzy CPM. The most popular solutions came from the formalism of fuzzy sets numbers [7], and are then implemented in fuzzy CPM, and fuzzy PERT [8].

In this context, the contribution covers various issues of decision making while employing the knowledge and CP based framework. The proposed approach provides the framework allowing one to take into account both: distinct (pointed), and imprecise (fuzzy) data, in a unified way and treated in a unified form of discrete constraint satisfaction problem (CSP) [3]. The approach proposed concerns of the logic-algebraic method based and CP-driven methodology aimed at interactive decision making based on distinct and imprecise data. The paper can be seen as continuation of our former works concerning projects portfolio prototyping [2], [6] and CP-based approach to the project-driven manufacturing.

We first provide an illustrative example of the problem considered, see the Section 2, and then we present some details of the modeling framework assumed, in particular we describe the reference model employed, see the Section 3. In the Section 4, the problem statement is provided, and then its CSP implementation is provided, see the Section 5. The logic-algebraic based approach to CSP resolution is discussed in the Section 6, and then an illustrative example of the possible application of the approach developed is discussed, see the Section 7. We conclude with some results and lesson learned in the Section 8.

2. ILLUSTRATIVE EXAMPLE OF DECISION PROBLEM

Consider the Job shop composed of 10 work stations where from the two semi-products K_1, K_2, two products W_1 and W_2 are manufactured following the production route P_1, see the Fig. 1. On the work stations three kinds of manufacturing operations are considered: decomposition, e.g. disassembly, $\{O_{1,2}, O_{1,4}\}$, composition, e.g. assembly $\{O_{1,5}, O_{1,9}, O_{1,10}\}$ and processing, e.g. milling $\{O_{1,1}, O_{1,3}, O_{1,6}, O_{1,7}, O_{1,8}\}$. The work stations are serviced by three robots (ro_1, ro_2, ro_3) and two workers (ro_4, ro_5). Two robots and/or workers can be allocated to each $O_{i,j}$, see the Table1.

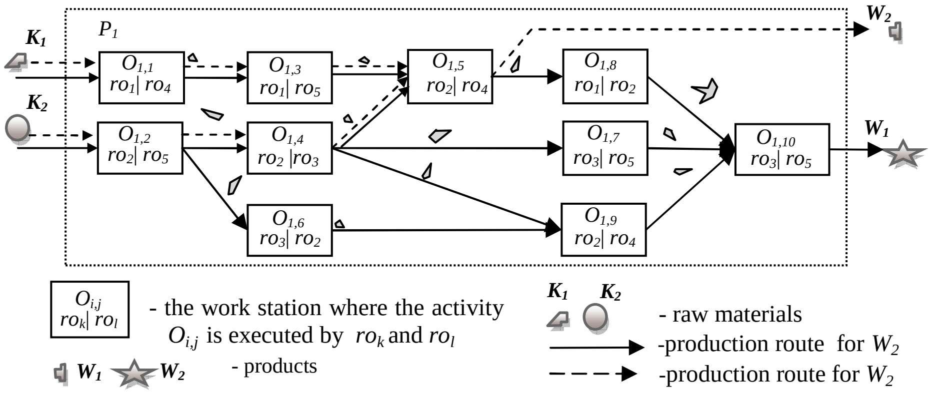

Fig. 1. Job shop following the production route P_1

Given are operations times as well as associated moments of the relevant resources allocation. Such kind of decision variables, e.g. operation times executed by robots or workers, can be specified either as distinct or imprecise ones.

Table 1. Robots and workers allocation to production route P_1 activities.

		$O_{1,1}$	$O_{1,2}$	$O_{1,3}$	$O_{1,4}$	$O_{1,5}$	$O_{1,6}$	$O_{1,7}$	$O_{1,8}$	$O_{1,9}$	$O_{1,10}$
robots	ro_1	1	0	1	0	0	0	0	1	0	0
	ro_2	0	1	0	1	1	0	0	1	1	0
	ro_3	0	0	0	1	0	1	1	0	0	1
workers	ro_4	1	0	0	0	1	1	0	0	1	0
	ro_5	0	1	1	0	0	0	1	0	0	1

Note that, since an amount of common shared resources is limited, hence their allocation to simultaneously executed activities has to avoid an occurrence of closed loop resources request, i.e. the deadlocks. Also, an imprecise nature of decision variables implies an imprecise (fuzzy) character of the performance evaluating criteria employed, e.g., an imprecise value of completion time concerning products W_1 and W_2. Moreover, because the constraints linking imprecise variables are also imprecise the relevant membership function grades, assumed to be involved in decision making, should be taken into account.

In that context, the problem of multi-robot task allocation in a multi-product job shop reduces to a class of the dispatcher's routine questions, such as: Does a given way of resources allocation guarantee the production orders completion time do not exceed the deadline H? Does there exist a way of resources allocation such that production orders completion time not exceeding the deadline H is guaranteed?

3. REFERENCE MODEL

Let us consider the reference model of a decision problem concerning of multi-robot task allocation in a multi-product job shop assuming imprecise character of decision variables. The model specifies both the job shop capability and production orders requirement in an unified way, i.e., through the description of determining them 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 in case such conditions hold the response to associated questions is positive. Of course, in order to avoid confusion the constraints guaranteeing the responses DO NOT KNOW are not allowed are also taken into account. In that context, the reference model is aimed at routine questions such as: Does a given job shop capabilities and a given way of resources allocation guarantee the assumed makespan of production orders do not exceed the deadline H?

Decision variables: Given amount lz of renewable discrete resources ro_i (for instance robots and workers) specified by the sequence $Ro = (ro_1, ro_2, \ldots, ro_{lz})$, and the sequence of resources availability $Zo = (zo_1, zo_2, \ldots, zo_{lz})$; zo_i –the availability of the i-th resource, assumed to be constant within the discrete time horizon H, where: $\{0,1,\ldots,h,\ldots,H\}$, $h \subset N$. Given a set of production routes $P = \{P_1, P_2, ..., P_{lp}\}$. Each the i-th route P_i is specified by the set composed of lo_i activities, i.e., $P_i = \{O_{i,1}, O_{i,2}, O_{i,3}, \ldots, O_{i,loi}\}$, where:

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

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

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

$Tp_{i,j} = (tp_{i,j,1}, tp_{i,j,2}, \ldots, tp_{i,j,lz})$ – the sequence of time moments the activity $O_{i,j}$ requires new amounts of renewable resources: $tp_{i,j,k}$ – the time counted since the moment $x_{i,j}$ 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: $0 \le tp_{i,j,k} < t_{ij}$; $k = 1,\ldots,lz$.

$Tz_{i,j} = (tz_{i,j,1}, tz_{i,j,2}, \ldots, tz_{i,j,lz})$ – the sequence of moments the activity $O_{i,j}$ releases the subsequent resources: $tz_{i,j,k}$ – the time counted since the moment $x_{i,j}$ 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: $0 < tz_{i,j,k} \le t_{ij}$; $k = 1,2,\ldots,lz$, 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 $dp_{i,j,k}$ are allocated to the activity $O_{i,j}$, i.e., $dp_{i,j,k}$ – the amount of the k-th resource allocated to the activity $O_{i,j}$. That assumes: $0 \le dp_{i,j,k} \le zo_k$; $k = 1, 2, \ldots, lz$.

Consequently, each activity $O_{i,j} = (x_{i,j}, t_{ij}, Tp_{i,j}, Tz_{i,j}, Dp_{i,j})$ is specified by the following sequences of:

- starting times of activities in the route P_i:

$$X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,lo_i}),\ 0 \le x_{i,j} < h;\ i = 1, 2, \ldots, lp;\ j = 1, 2, \ldots, lo_i,$$

- duration of activities in the route 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 in the route P_i:

$TP_{i,j} = (tp_{i,1,j}, ..., tp_{i,k,j}, ..., tp_{i,lo_i,j})$,

- starting times the j-th resource is released by the k-th activity in the P_i:

$TZ_{i,j} = (tz_{i,1,j}, tz_{i,2,j}, ..., tz_{i,lo_i,j})$,

- amounts of the j-th resources allotted to the k-th activity in the route P_i:

$DP_{i,j} = (dp_{i,1,j}, dp_{i,2,j}, \ldots, dp_{i,lo_i,j})$.

Assume some of chosen execution times are defined precisely, however a few of them are known roughly i.e., are treated as fuzzy variables specified by fuzzy sets. In case of imprecise decision variables such as operation times $\hat{T}_i = (\hat{t}_{i,1}, \hat{t}_{i,2}, ..., \hat{t}_{i,lo_i})$ where $\hat{t}_{i,j}$ denotes execution time of the operation $O_{i,j}$, and starting times of activities $\hat{X}_i = (\hat{x}_{i,1}, \hat{x}_{i,2}, ..., \hat{x}_{i,lo_i})$, where $\hat{x}_{i,j}$ denotes starting time of activity $O_{i,j}$. Therefore, the activity $O_{i,j}$ = (, , $Tp_{i,j}$, $Tz_{i,j}$, $Dp_{i,j}$) is specified by the following sequences of:

- starting times of activities in the route P_i:

$$\hat{X}_i = (\hat{x}_{i,1}, \hat{x}_{i,2}, ..., \hat{x}_{i,lo_i}), \qquad (2)$$

- duration of activities in the route P_i:

$$\hat{T}_i = (\hat{t}_{i,1}, \hat{t}_{i,2}, ..., \hat{t}_{i,lo_i}), \qquad (3)$$

where: $\hat{X}_i$ – is a fuzzy set determining the operation $O_{i,j}$ starting time,
$\hat{T}_i$ – is a fuzzy set specifying the operation time,
$Tp_{i,j}$, $Tz_{i,j}$, $Dp_{i,j}$ – the sequences defined as in formulae (1).

Considered fuzzy variables are specified by fuzzy sets described by convex membership function [12]. Since, that distinct decision variables can be seen as a special case of imprecise ones, hence all the further considerations are focused on the imprecise (fuzzy) kind of variables.

Activities order constraints: Let us consider a set of production routes P_i composed of lo_i precedence and resource constrained, non-preemptable activities that require renewable resources. Assume lz renewable discrete resources are available and sequences $r_i = (ro_1, ro_2, \ldots, ro_f)$, $i = 1, \ldots, lo_i$, determines fixed discrete resource requirements of the i-th activity. The total number of units of the discrete resource j, $j=1, \ldots, lz$, is limited by zo_j. The resource can be allotted (and constant within activity operation time) to activities in arbitrary amount from the set $\{1, \ldots, zo_j\}$. The resources allotted to the i-th activity have to be available at the moments $Tp_{i,j}$, $Ts_{i,j}$.

The production routes P_i are represented by activity-on-node networks, where activities state for nodes and arcs determine an order of activities execution. Consequently, the following activities order constraints are considered:

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

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

- the k-th activity follows other activities:

$$\hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k}, \hat{x}_{i,j+1} \hat{+} \hat{t}_{i,j+1} \hat{\leq} \hat{x}_{i,k}, ..., \hat{x}_{i,j+n} \hat{+} \hat{t}_{i,j+n} \hat{\leq} \hat{x}_{i,k}\,, \tag{5}$$

- the k-th activity is followed by other activities:

$$\hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k+1}, \hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k+2}, ..., \hat{x}_{i,j} \hat{+} \hat{t}_{i,j} \hat{\leq} \hat{x}_{i,k+n}\,, \tag{6}$$

The relevant fuzzy arithmetic operations $\hat{+}, \hat{\leq}$ are defined in the Appendix. Due to the formulas (8), (12), see the Appendix, any fuzzy constraint C_i (e.g. $\hat{v}_i \hat{<} \hat{v}_l$) can be characterized by the logic value $E(C_i)$, $E(C_i) \in [0,1]$. In turn, values $E(C_i)$ allow to determine the level of uncertainty DE of reference model's constraints satisfaction, i.e. a kind of uncertainty threshold. For instance, DE = 1 means the all constraints hold, and DE = 0,8 means that they are almost satisfied. The level DE is defined due to the formulae (7):

$$DE = \min_{i=1,2,...,lo_c}\{E(C_i)\}\,, \tag{7}$$

where: lo_c – a number of reference model constraints.

Resource conflict constraints: In order to avoid deadlocks the constraints providing conflicts resolution, i.e., avoiding the occurrence of closed loop resources request, are considered. The constraints guarantee the sum of allocated amounts of a given resource do not exceed its current availability $zo_{i,j}$, at any moment within the assumed time horizon $\{0,1,\ldots,H\}$. So, for each the k-th resource the following inequalities hold (8) at any time $g \in \{0,1,\ldots,H\}$,

$$\sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k} \cdot \bar{1}(g, x_{i,j} + tp_{i,j,k}, x_{i,j} + tz_{i,j,k})\right] \leq zo_k\,, \tag{8}$$

where: lp – a number of projects, lo_i – a number of activities contained by the i-th project,
$dp_{i,j,k}$ – an amount of the k-th resource allocated by $O_{i,j}$,
$\bar{1}(g,a,b) = 1(g-a) - 1(g-b)$ – the unit step function of the resource allocation where 1(a) - is a unit step function.

In case of fuzzy constraints design one has to take into account fuzzy variables $X_{i,j}$, $Tp_{i,j}$ $Tz_{i,j}$ as well as a fuzzy unit step function of the resource allocation (9):

$$\hat{\bar{1}}(\hat{g},\hat{a},\hat{b},E_{\hat{1}}) = \hat{1}(\hat{g},\hat{a},E_{\hat{1}}) - \hat{1}(\hat{g},\hat{b},E_{\hat{1}})\,, \tag{9}$$

where: $\hat{a},\hat{b}$ – fuzzy numbers (in the case variables are precise $\hat{a}$ and $\hat{b}$ are singletons.

Consider the following fuzzy unit step function of the resource allocation (10):

$$\hat{\bar{1}}(\hat{g},\hat{a},E_{\hat{1}}) = 1 - \frac{E_{\hat{1}} - E(\hat{g} \geq \hat{a})}{1 - 2E(\hat{g} \geq \hat{a})} \tag{10}$$

where: $\hat{\hat{1}}(\hat{g},\hat{a},E_{\hat{1}}) \in \{0,1\}, E_{\hat{1}} \in [0,1]$ – means the fuzzy logic value.

Due to constraints following (8), the sum of requested resources is calculated only at moments corresponding to the ones $x_{i,j}$ + $tp_{i,j}$, when resources are allocated to subsequent activities. Therefore, an amount of available resources may change within the time horizon H. So, in order to avoid the number of allocated resources exceeds the amount of available resources (similarly to (8)) the constraints (11) are introduced.

$$\begin{cases} \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \hat{\hat{1}}(\hat{x}_{1,1}\hat{+}tp_{1,1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},E_{\hat{1},i,j,1})\right]\le zo_k \\ \ldots \\ \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \hat{\hat{1}}(\hat{x}_{1,lo_1}\hat{+}tp_{1,lo_1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},E_{\hat{1},i,j,lo_1})\right]\le zo_k \\ \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \hat{\hat{1}}(\hat{x}_{2,1}\hat{+}tp_{2,1,k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},E_{\hat{1},i,j,lo_1+1})\right]\le zo_k \\ \ldots \\ \sum_{i=1}^{lp}\sum_{j=1}^{lo_i}\left[dp_{i,j,k}\cdot \hat{\hat{1}}(\hat{x}_{lp,lo_{lp}}\hat{+}tp_{lp,lo_{lp},k},\hat{x}_{i,j}\hat{+}tp_{i,j,k},\hat{x}_{i,j}\hat{+}tz_{i,j,k},E_{\hat{1},i,j,lo_1+lo_2+\ldots+lo_{lp}})\right]\le zo_k \end{cases} \quad (11)$$

for k = 1,2, …, lz,

where: lz – a number of renewable resources

$E_{\hat{1},i,j,q}$ – uncertainty threshold of the i,j –th fuzzy unit step function of the resource allocation.

Due to (7) the logic value $E(Co_q)$ of the particular constraint Co_q from the set (11) is calculated as follows (12):

$$DE = \min_{i=1,2,\ldots,lp}\left\{\min_{j=1,2,\ldots,lo_i}\{E_{\hat{1},i,j,q}\}\right\} \quad (12)$$

where: lp – a number of production routes, lo_i – activities number in the i-th production route.

Note that in the course of decision making supported by constraints defined on fuzzy variables the relevant uncertainty thresholds (e.g. following an operation's experience) should be assumed. That means, in order to guarantee an intuitive interpretation of decision making the manager should be able to decide about the membership functions of the decision variables used as well as uncertainty thresholds of fuzzy constraints employed.

4. PROBLEM STATEMENT

The introduced model provides the formal framework enabling one to state the problem considered. Given the time horizon {0,…,H}, the set of production orders (specified by the set of production routes) P, the set of resources and their availabilities Zo within {0,…,H}. Given are distinct and imprecise decision variables treated as fuzzy numbers, i.e. the sequences $\hat{T}_i$, $\hat{TP}_{i,j}, \hat{TZ}_{i,j}$. The following questions should be answered:

Does a given resources allocation guarantee the production orders makespan do not exceed the deadline *H*? Response to this question results in determination of the sequences: $\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp}$.

Does there exists such resources allocation guaranteeing the production orders makespan do not exceed the deadline *H*? Response to this question results in determination of the sequences: $\hat{X}_1, \hat{X}_2, ..., \hat{X}_{lp}$.

The following remarks should be stated:

- the problems considered are formulated in terms of the reference model proposed,
- the questions stated above correspond to the straight and reverse problems of multi-product scheduling.

5. CONSTRAINT SATISFACTION PROBLEM

Constraint programming (*CP*) is an emergent software technology for declarative description and effective solving of large combinatorial problems, especially in the areas of integrated production planning. Since a constraint can be treated as a logical relation among several variables, each one taking a value in a given (usually discrete) domain, the idea of *CP* is to solve problems by stating the requirements (constraints) that specify a problem at hand, and then finding a solution satisfying all the constraints [4]. Because of its declarative nature, it is particularly useful for applications where it is enough to state *what* has to be solved instead *how* to solve it [4]. 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 values of the variables>.** So, the constraint satisfaction problem (*CSP*) [4] is defined as follows:

$$CS = ((A, D), C) \tag{13}$$

where: $A = \{a_1, a_2, ..., a_g\}$ – a finite set of discrete decision variables,

$D = \{D_i \mid D_i = \{d_{i,1}, d_{i,2}, ..., d_{i,j}, ..., d_{i,ld}\}, i = 1,...,g\}$ – a family of finite variable domains and the finite set of constraints

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

The solution to the *CS* is a vector $(d_{1,i}, d_{2,k}, \ldots, d_{n,j})$ such that the entry assignments satisfy all the constraints *C*. So, the task is to find the values of variables satisfying all the constraints, i.e., a feasible valuation.

The inference engine consists of the following two components: constraint propagation and variable distribution. Constraints propagation uses constraints actively to prune the search space. The aim of propagation techniques, i.e., local consistency checking, is to reach a certain level of consistency in order to accelerate search procedures by drastically reducing the size of the search tree [3]. The constraints propagation executes almost immediately. What limits the size of the problem in practical terms is the variable distribution phase, which employs the backtracking-based search and is very time consuming as a result.

The declarative character of *CP* languages and their high efficiency in solving combinatorial problems offer an attractive alternative to the currently available DSSs that employ operation research techniques.

6. KNOWLEDGE BASE

Logic-algebraic method (*LAM*) [6] based inference engine in the *CP* environment permits to obtain a solution more efficiently, either in terms of the solution time or the scale of the problem. The idea behind the introduction of the *LAM* formalism consists in the assumption that response of the type DO NOT KNOW is not allowed.

It is assumed that the knowledge base *KB* describing a system (e.g. an enterprise) is presented in the form of the sets *U*, *W*, *Y*, that define the domains of some system properties *u*, *y*, *w* (at the qualitative level). The variables *u* describing the input properties of the system are called the input variables, the variables *y* describing the output properties of the system are called the output variables, and the variables *w* are called the auxiliary variables. The knowledge specifying the properties of the system under consideration is described in the form of the set of facts $F(u,w,y)$. The facts $F(u,w,y)$ are propositions encompassing, the relationships (i.e., constraints) occurring between individual variables u,w,y.

The decision problem can be then formulated in the following way. Given are sets of input variables $U=\{u_1,u_2,\ldots,u_n\}$, output variables $Y=\{y_1,y_2,\ldots,y_m\}$, auxiliary variables $W=\{w_1,w_2,\ldots,w_k\}$, with the variables u_i, y_j, and w_i defined in domains D_{ui}, D_{yi}, D_{wi}, and sets of constraints (properties) $F(U)$, $F(Y)$ linking the variables from different sets. The decision problem consists in finding such a relation $R \subset U \times Y \times W$ for which the input property $F(U)$ implies the satisfaction of the condition $F(U) \Rightarrow F(Y)$. The solution can be easily found based on the *LAM* theory [6]:

$$R_u = S_{u1} \setminus S_{u2} \tag{14}$$
$$S_{u1} = \{(U): w(F(U,Y,W)) = 1,\ w(F(Y)) = 1\} \tag{15}$$
$$S_{u2} = \{(U): w(F(U,Y,W)) = 1,\ w(F(Y)) = 0\} \tag{16}$$

where:

$$w(F(.)) = \begin{cases} 1 \text{ if } F(.) \text{ holds} \\ 0 \text{ if } F(.) \text{ holds} \end{cases}$$

The set S_{u1} consists of those elements of *U* for which all facts of sets $F(U,Y,W)$, $F(Y)$ hold. The set S_{u2}, in turn, consists of those elements of *U* values for which all facts of the set $F(U,Y,W)$ hold true, and at least one fact from the set $F(Y)$ does not hold true. $R_u=\varnothing$ denotes the lack of answer to the question asked.

Consequently, the *CSP* considered results in the following form:

$$CS = ((U,Y,W), D), \{w(F(U,Y,W)) = 1\}) \tag{17}$$

where: $D = \{D_U, D_Y\}$ - D_U is a set of input variables values *U*, D_Y is a set of the output variable values *Y*,
$w(F(U,Y,W)) = 1$ notes a set of facts $\{w(F_1(U,Y,W))=1,\ldots,w(F_K(U,Y,W)) = 1\}$.

Solving the decision problem (i.e., determination of the relation *R*) - in the context of *CSP* formalism - requires solving the following two problems:

$$CS_{Su1}=((U,Y,W),D),\{w(F(U,Y,W))=1,w(F(Y))=1\}) \tag{18}$$
$$CS_{Su2}=((U,Y,W),D),\{w(F(U,Y,W))=1,w(F(Y))=0\}) \tag{19}$$

Set $R = S_{u1} \setminus S_{u2}$, where sets S_{u1}, S_{u2} are the solutions to the above problems, includes a group of alternative solutions for which the implication $F(U) \Rightarrow F(Y)$ holds. The inference engine applied in the *LAM* is easily implementable in commercially available constraint logic programming languages, such as Oz/Mozart [13], Ilog [11].

7. ILLUSTRATIVE EXAMPLE

For illustration of the reference model based approach proposed let us consider the production route P_1 composed of 10 activities (see Fig.1). Two products are manufactured simultaneously: W_1 and W_2 respectively. The operation times are treated as fuzzy variables and determined by z-cuts: $\hat{T}_1 = (\hat{t}_{1,1}, \hat{t}_{1,2}, ..., \hat{t}_{1,10})$

$\hat{t}_{1,1}$ ={{[1,3],[2,3],[3,3]},{0;0,5;1}} (see Fig. 1 in the Appendix)

$\hat{t}_{1,2}$ ={{[2,6],[2,5],[1,1] },{0;0,5;1}}, $\hat{t}_{1,3}$ ={{[5,5],[5,5],[5,5]},{0;0,5;1}},

$\hat{t}_{1,4}$ ={{[3,5],[3,4],[3,3]},{0;0,5;1}}, $\hat{t}_{1,5}$ ={{[2,4],[3,4],[4,4]},{0;0,5;1}},

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

$\hat{t}_{1,9}$ ={{[2,2],[2,2],[2,2]},{0;0,5;1}}, $\hat{t}_{1,10}$ ={{[2,4],[3,4],[4,4]},{0;0,5;1}}.

Five different renewable resources ro_1, ro_2, ro_3, ro_4, ro_5 are used. The resources' allocation follows the Table 1. Therefore:

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

That is assumed the moments of resources' allocation and release follow the moments of operation's beginning and completion. Therefore $TP_{1,1}$ = $TP_{1,2}$ = $TP_{1,3}$ = $TP_{1,4}$ = $TP_{1,5}$= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0). Assumed are the following sequences: $\hat{T}Z_{1,1} = \hat{T}Z_{1,2} = \hat{T}Z_{1,3} = \hat{T}Z_{1,4} = \hat{T}Z_{1,5} = \hat{T}_1$ as well as $Zo = (zo_1, zo_2, zo_3, zo_4, zo_5)$ such that $zo_1 = zo_2 = zo_3 = zo_4 = zo_5$=1. Given the discrete time horizon $H = [0, 20]$, $H \subset N$, and the uncertainty threshold $DE \geq 0,8$.

The question considered: Does there exists a production schedule makespan of which do not exceeds a given deadline *H*? concerns of $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,10})$ assuming the moments are fuzzy numbers with triangle membership function.

The activities order (4), (5), (6) and resource conflict (11) constraints have been implemented in OzMozart [13].

First sufficient solution $\hat{X}_1 = (\hat{x}_{1,1}, \hat{x}_{1,2}, ..., \hat{x}_{1,10})$ (see Fig. 2) was obtained within three minutes (AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1,00 GB):

$\hat{x}_{1,1}$ ={{[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}}, $\hat{x}_{1,4}$ ={{[2,4],[3,4],[4,4]},{0;0,5;1}}

$\hat{x}_{1,5}$ ={{[7,9],[8,9],[9,9]},{0;0,5;1}}, $\hat{x}_{1,6}$ ={{[5,7],[6,7],[7,7]},{0;0,5;1}}

$\hat{x}_{1,7}$ ={{[7,9],[8,9],[9,9]},{0;0,5;1}}, $\hat{x}_{1,9}$ ={{[12,14],[12,13],[12,12]},{0;0,5;1}},

$\hat{x}_{1,8}$ ={{[11,13],[11,12],[11,11] },{0;0,5;1}},

$\hat{x}_{1,10}$ ={{[13,15],[14,15],[15,15] },{0;0,5;1}}.

Requirements following intuitive decision making imply the transformation of the fuzzy schedule obtained (see Fig.2) into the crispy-like one, e.g. providing results with the grade ≥ 0.5 (see Fig.3).That means, assuming the uncertainty threshold value $DE \geq 0{,}8$, the completion time of product W_2 does not exceed 15 units of time, and 19 units of time in case of product W_1.

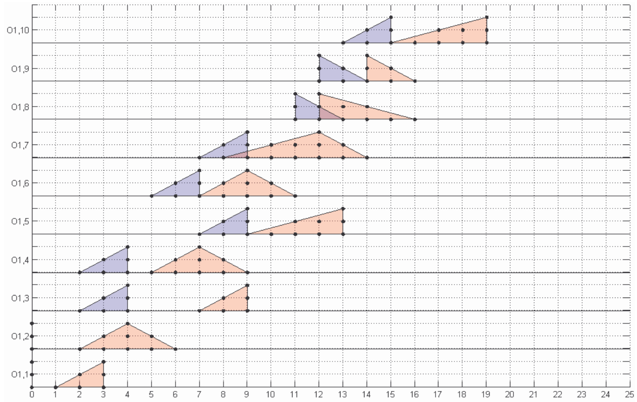

Fig. 2. Fuzzy schedule for production route P_1

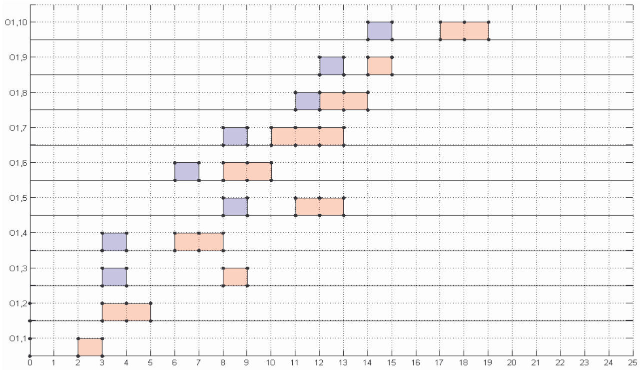

Fig. 3. Crispy-like at the grade 0.5 of membership function schedule for production route P_1.

8. CONCLUDING REMARKS

Proposed approach to multi-robot task allocation for multi-product job shop provides the framework allowing one to take into account both: straight and reverse problem statement. This advantage can be seen as a possibility to response (besides of standard questions: Is it possible to complete a given set of production orders at a scheduled project deadline?) to the questions like: What values and of what variables guarantee the production orders will completed due to assumed values of set of performance indexes? Provided example illustrates possibility of the straight implementation of the reference model in the constraint programming environment as well as capabilities of their usage in reverse problem solution.

Moreover the proposed approach provides the framework allowing one to take into account both: the sufficient conditions (guaranteeing the admissible solutions there exist) and choosing the best solution on the basis of chosen evaluation criteria. It 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.

References

[1] Archer, N., and F. Chasemzadech, An integrated framework for project portfolio selection. Int. Journal of Project Management, Vol. 17, No. 4, 1999; 207-216.

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

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

[4] 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., 2004; 7-14.

[5] Beale E. M. L.: Branch and bound methods for mathematical programming systems. In P. L.Hammer, E. L. Johnson, and B. H. Korte, editors, Discrete Optimization II, North Holland Publishing Co., 1979, 201–219.

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

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

[8] Dubois D., Fargier H., Fortemps P.: Fuzzy scheduling: Modeling flexible constraints vs. coping with incomplete knowledge, European Journal of Operational Research 147, 2003, 231 – 252.

[9] Linderoth T., Savelsbergh. M. W. P.: A computational study of search strategies in mixed integer programming, In.: INFORMS Journal on Computing, 11:173–187, 1999.

[10] Martinez, E. C., D., Duje G.A. Perez, On performance modeling of project-oriented production. Computers and Industrial Engineering, Vol. 32, 1997; 509-527.

[11] Puget J-F.: A C++ Implementations of CLP, Proceeding of SPICS 94, 1994.

[12] Piegat A. Fuzzy modeling and control, A Springer-Verlag Company, Heidelberg-New York, 2001.

[13] Schulte CH., Smolka G., Wurtz J.: Finite Domain Constraint Programming in Oz, DFKI OZ documentation series, German Research Center for Artificial Intelligence, Stuhlsaltzenhausweg 3, D-66123 Saarbrucken, Germany, 1998.

[14] Sung. C.S., A Production Planning Model for Multi-Product Facilities, In: Journal of the Operations Research Society of Japan, Vol. 28, No. 4, pp. 345-385,1985.

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

[16] Zimmermann H.J. (1994), Fuzzy sets theory and its applications. London: Kluwer Academic Publishers.

APPENDIX

Imprecise variables specified by fuzzy sets and determined by convex membership function can be characterized by α - cuts [12], and then defined by pairs (a1):

$$\{A_i, \alpha\} \tag{a1}$$

where: $A_i = \{A_{z_i,1}, A_{z_i,2},..., A_{z_i,lz}\}$ finite set of so called z - cuts,

$\alpha_{i,j} = \{\alpha_{i,1}, \alpha_{i,2},\ldots, \alpha_{i,lz}\}$ – is a set $A_{z_i,1}, A_{z_i,2},..., A_{z_i,l\alpha}$ of values corresponding to α - cuts at levels $\alpha_{i,j}$, lz – a number of z-cuts. And

$$A_{z_i,k} = [a_{i,k}, b_{i,k}]_{\mathrm{N}} \tag{a2}$$

where: $a_{i,k}$, $b_{i,k}$ – is the smallest and the highest value of the k-th α - cut, $a_{i,k}, b_{i,k} \in N$
The z-cut can be seen as a discretized form of the α - cut, i.e. $A_{z_i,k} = A_{\alpha_i,k} \cap (N \cup 0)$ see Fig.3.

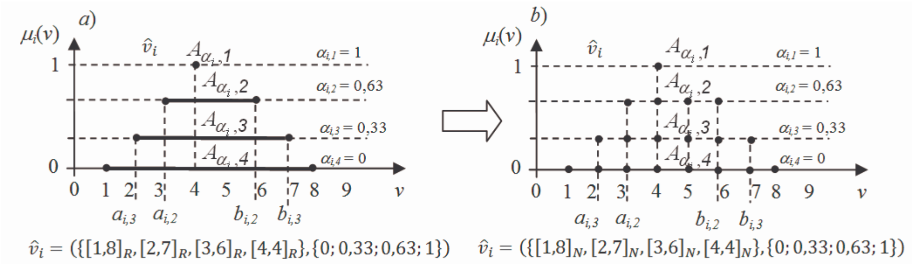

Fig. 1 Fuzzy set $\hat{v}_i$ specified by: *a*) α - cuts, *b*) discretized α - cuts, i.e., z-cuts.

Note, that under assumed specification the distinct values are represented by singletons.

Imprecise character of decision variables, e.g., $\hat{x}_{i,j}$, $\hat{t}_{i,j}$, implies imprecise character of employing them constraints, which in turn can be considered as a consequence of implementation of assumed operations. Therefore, consider the set of fuzzy operations: **„$\hat{=}$”, “$\hat{<}$”„$\hat{>}$”,** encompassing standard algebraic operations such as: =, ≠, <, >, ≥, ≤. Of course, the considered fuzzy operations linking two fuzzy variables $\hat{v}_i$, $\hat{v}_l$ have to follow the condition (a3):

$$E(\hat{v}_i \hat{<} \hat{v}_l) + E(\hat{v}_i \hat{=} \hat{v}_l) + E(\hat{v}_i \hat{>} \hat{v}_l) = 1 \tag{a3}$$

where: $E(a)$ – 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 (a10) the following auxiliary sets v_i^L, v_i^*, v_i^P and v_l^L, v_l^*, v_l^P are defined as well as the concept of a size of fuzzy variable S_i the size 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 the variable $\hat{v}_i$, the following sets can be distinguished: v_i^L, v_i^*, v_i^P and v_l^L, v_l^*, v_l^P. For instance, for the set the following subsets can be determined:

v_i^L – the set composed of elements v being less (smaller) than all elements from $\hat{v}_l$;

$v_{i,j}^*$ – the set of elements shared with $\hat{v}_l$;

v_l^P – the set composed of elements v being greater (bigger) than all elements from . The sets v_i^L, v_i^*, v_i^P are defined as follows:

$$v_i^L = \{(\mu_i^L(v), v)\}, \qquad \forall v \in K_i, \tag{a4}$$

where:

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

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

$$v_i^* = \{(\mu_i^*(v), v)\}, \qquad \forall v \in K_i, \tag{a5}$$

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

$$v_i^P = \{(\mu_i^P(v), v)\}, \qquad \forall v \in K_i, \tag{a6}$$

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

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

Corresponding to the fuzzy variable $\hat{v}_i$ subsets v_i^L, v_i^*, v_i^P are defined in the same way.

To each fuzzy variable $\hat{v}_i$, $\hat{v}_l$ and the corresponding subset v_i^L, v_i^*, v_i^P, v_l^L, v_l^*, 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 (a7):

$$S_i = \sum_{k=1}^{lz} \left\| A_{z_i,k} \right\| , \tag{a7}$$

where: $\left\| A_{z_i,k} \right\|$ – a number of elements of the set $A_{z_i,k}$.

In the similar way the size values S_l, S_i^L, S_l^*, S_i^P, S_l^L, S_i^*, S_i^P, corresponding to the sets v_i^L, v_i^*, v_i^P, v_l^L, v_l^*, v_l^P are defined.

In the case considered because the decision variables $\hat{v}_i$, $\hat{v}_l$ concern of the time domain the equation $S_l^* = S_i^*$ holds, for the given v_i^*, v_l^*. Therefore, for the sake of simplicity in further considerations the sizes S_l^*, S_i^*, will be 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 $\hat{v}_i \mathrel{\hat{=}} \hat{v}_l$ is defined by (a8):

$$E(\hat{v}_i \mathrel{\hat{=}} \hat{v}_l) = \frac{2S^*}{S_i + S_l} \tag{a8}$$

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 $\hat{v}_i \mathrel{\hat{<}} \hat{v}_l$ is defined by (a9):

$$E(\hat{v}_i \mathrel{\hat{<}} \hat{v}_l) = \frac{S_i^L + S_l^P}{S_i + S_l} \tag{a9}$$

where: S_i – the size of $\hat{v}_i$, S_l – the size of $\hat{v}_l$,

S_i^L – the size of v_i^L, S_i^P – the size of v_i^P,

Fuzzy logic value of the proposition $\hat{v}_i \mathrel{\hat{>}} \hat{v}_l$ is defined by (a10):

$$E(\hat{v}_i \mathrel{\hat{>}} \hat{v}_l) = \frac{S_i^P + S_l^L}{S_i + S_l} \tag{a10}$$

Fuzzy logic value of the proposition $\hat{v}_i \mathrel{\hat{\geq}} \hat{v}_l$ is defined by (a11):

$$E(\hat{v}_i \mathrel{\hat{\geq}} \hat{v}_l) = \frac{2S^* + S_i^P + S_l^L}{S_i + S_l} \tag{a11}$$

Fuzzy logic value of the proposition $\hat{v}_i \mathrel{\hat{\leq}} \hat{v}_l$ is defined by (a12):

$$E(\hat{v}_i \mathrel{\hat{\leq}} \hat{v}_l) = \frac{2S^* + S_i^L + S_l^P}{S_i + S_l} \tag{a12}$$

Formulaes (a8), (a9), (a10), (a11), (a12) allow one to design constraints describing basic relations among two fuzzy variables, such as equality, less than, greater than, less or equal, and greater or equal. In order to allow one to consider other constraints, e.g., taking into acount distinct variables, the fuzzy operations such as fuzzy addition and fuzzy subtraction have to be employed as well. The relevant operations **„** $\hat{+}$ **”, „** $\hat{-}$ **”** can be found in [6].