scheduling, constraint programming,
decision support system

Grzegorz BOCEWICZ, Irena BACH, Zbigniew BANASZAK*

DECISION SUPPORT TOOL FOR PROJECTS PORTFOLIO PROTOTYPING

Abstract

Constraint Programming (CP) is an emergent software technology for declarative description and effective solving of large combinatorial problems especially in the area of projects portfolio prototyping. The paper deals with multi-resource and multi-criteria 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 CP-driven approach to resource allocation conflicts resolution framework. Proposed framework stands behind a methodology aimed at task oriented DSS tolls designing. The Portfolio Project Prototyping System designed due to this methodology provides a prompt and interactive service to a set of routine queries stated both in straight and reverse way. Multiple illustrative examples are discussed.

1. INTERACTIVE TASK ORIENTED DECISION SUPPORT TOOLS

Some industrial processes simultaneously produce different products using the same production resources. 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 completion, and so on. So, a manager'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?

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

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

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 [4], [8].

Several techniques have been proposed in the past fifty years, including MILP, Branch-and-Bound [6] or more recently Artificial Intelligence. The last sort of techniques concentrates mostly on fuzzy set theory and constraint programming frameworks. Constraint Programming/Constraint Logic Programming (*CP*/*CLP*) languages [6], [20] seems to be well suited for modeling of real-life and day-to-day decision-making processes in an enterprise [5]. In turn, applications of fuzzy set theory in production management [21] show that most of the research on project scheduling has been focused on **fuzzy *PERT*** and **fuzzy *CPM*** [12], [13].

In this context, the contribution 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 a discrete, constraint satisfaction problem (*CSP*) [4]. The approach proposed concerns of logic-algebraic method *(LAM)* 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 [5], [11].

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

a straight way (i.e. corresponding to the question: What results from premises?)

- What the portfolio makespan follows from the given project constraints specified by activity duration times, resources amount and their allocation to projects' activities?
- Does a given resources allocation guarantee the production orders makespan do not exceed the given deadline?
- Does the projects portfolio can be completed before an arbitrary given deadline?
-

a reverse way (i.e. corresponding to the question: What implies conclusion?)

- What activity duration times and resources amount guarantee the given production orders portfolio makespan do not exceed the deadline?
- Does there exist resources allocation such that production orders makespan do not exceed the deadline?
- Does there exist a set of activities' operation times guaranteeing a given projects portfolio completion time will not exceed the assumed deadline?
-

Above mentioned categories encompass the different reasoning perspectives, i.e. deductive and abductive ones. The corresponding queries can be stated in different models that in turn may be treated as compositions of variables and constraints, i.e. assumed sets of variables and constraints limiting their values. In that context both an enterprise and the relevant production orders can be specified in terms of distinct and/or imprecise variables, discrete and/or continuous variables, renewable and/or non-renewable resources, limited and/or unlimited resources, and so on.

Possible problems formulation taking into account commercially available software packages capabilities is shown in the Table 1. So, that is easy to observe that commercially available tools are not able to consider cases assuming imprecise data as well as are not able to state a problem in an reverse way (e.g., looking for values of some input variables guaranteeing the assumed output variables reach required values). Moreover, the commercially available DSSs are not able to respond in an interactive, i.e. on-line/real-time mode, as well as to support a project-like production flow prototyping (i.e. integrated production planning containing such partial problems as routing, batch-sizing and scheduling).

That disadvantage is our motivation to develop methodology supporting one in the course of designing of an interactive and task oriented decision support systems aimed at projects portfolio prototyping. By projects prototyping we mean a decision process resulting in selection (variables adjustment) both an enterprise and projects portfolio parameters fulfilling assumed requirements, e.g. an admissible solution being a kind of an equilibrium between enterprise capabilities and projects' cost and makespan.

Tab. 1. Possible problems formulation available in commercially available software packages perspective

DSS	variables		resources		queries	
	precise	imprecise	renewable	non-renewable	straight	reverse
Primavera	✓	×	✓	×	✓	×
Planiswere	✓	×	✓	×	✓	×
Tracker Suite	✓	×	✓	×	✓	×
Project Net	✓	×	✓	×	✓	×
Team Work	✓	×	✓	×	✓	×
…	…	…	…	…	…	…
MSProject	✓	×	✓	×	✓	×

An approach proposed assumes a kind of reference model encompassing open structure enabling one to take into account different sorts of variables and constraints as well as to formulate straight and reverse kind of project planning problems. So, the elementary as well as hybrid models can be considered, see the Fig. 1. Of course, the most general case concerns of the hybrid model specified by discrete distinct and/or imprecise (fuzzy) variables and renewable and/or non-renewable resources.

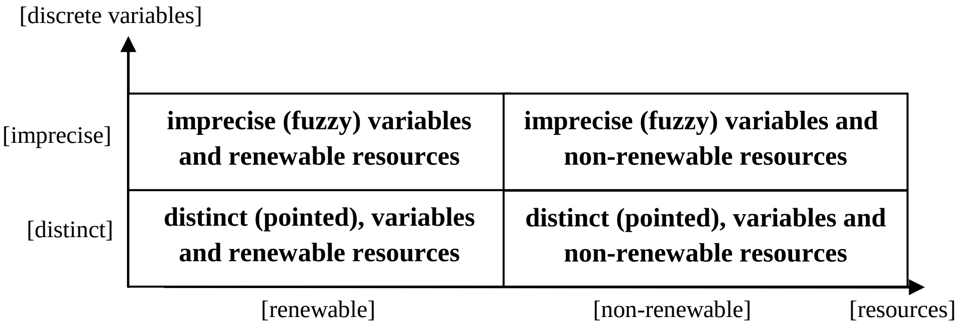


Fig. 1. Elementary decision problems

Note that assumed model enabling descriptive way of a problem statement encompasses constraint satisfaction problem structure and then allows implementing the problem considered in constraint programming environment. That is because the constraint programming treated as programming paradigm enables to specify both variables and relations between them in the form of constraints and then to implement them in the one of popular constraint logic languages such as: **CHIP V5**, **ECLiPSe**, and **SICStus**, or imperative constraint programming languages (assuming that a statement computation results in a program state change) such as: **Choco**, **ILOG**, and **python-constraint**, or public domain concurrent constraint programming language as **Oz Mozart**.

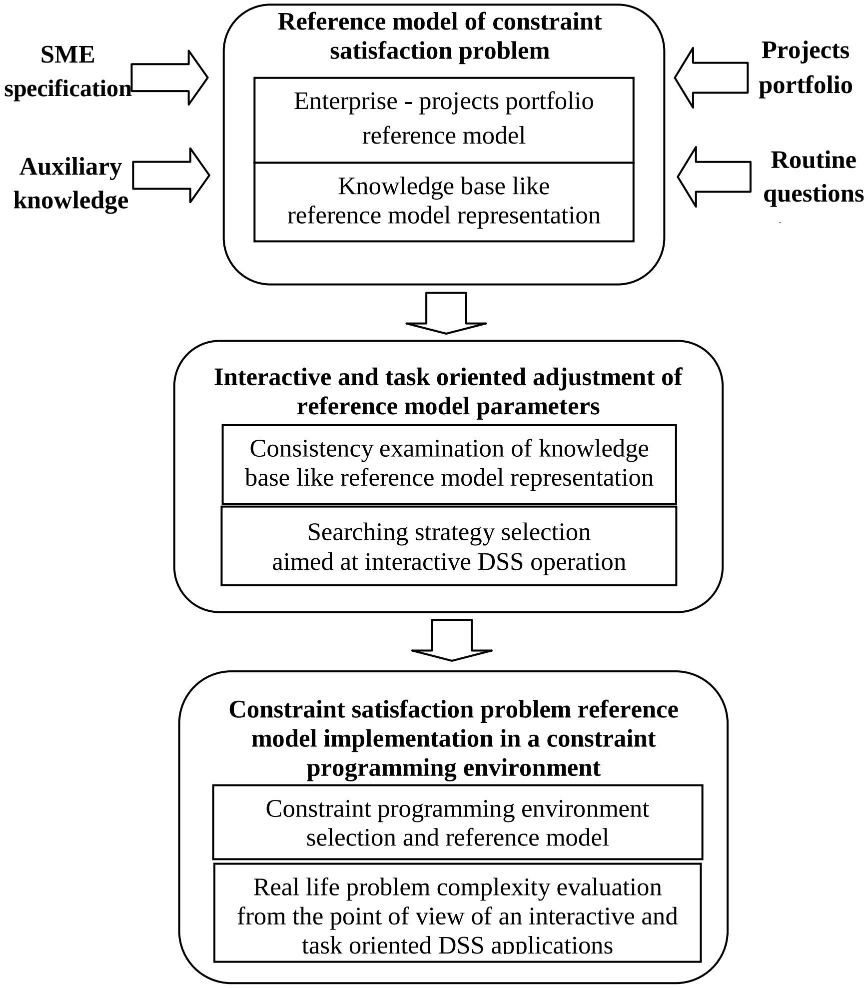


Fig. 2. Main stages of an interactive and task oriented DSS designing

In that context the methodology proposed consists of the following three stages, see Fig. 2. At the first stage the reference model of constraint satisfaction problem is considered. That means that on the base of available specifications of a small and medium sized enterprise and projects portfolio as well as the assumed routine queries and possible auxiliary (suggested by experts) knowledge a relevant **reference model of constraint satisfaction problem** is designed. The model encompasses technical parameters, experts' experience and user expectations in the form of knowledge base, i.e. as a set of variables, and their domains, and a set of relations (constraints, e.g. time-window resource availability) linking some subsets of constraints. Such model's interpretation allows using the logic-algebraic method as a reference engine.

At the second stage, the knowledge base obtained is examined from the point of view of its future implementation in assumed, implementing CP framework, real-life DSS. Since the CP framework is useless in case variables can be gathered in disjoint clusters and is useless also for queries checking whether a given subset of variables implies other one, thus the knowledge base (KB) consistency (guaranteeing response to the set of assumed queries) and its discrepancy (guaranteeing the unique response to each query) point of view must be examined. Besides of that, the KB has to be examined also from the point of view of the time efficiency of possible searching strategies (especially variables distribution). That means the searching strategy guaranteeing an interactive DSS operation has to be developed. The above mentioned examinations guarantee the KB specification can be directly implemented in CP framework (that means the straight and reverse problems' formulation and queries such as whether a given subset of variables implies other one can be considered).

Therefore, the third stage transforms the knowledge-based and CP-driven framework into the commercially available CP/CLP platforms (i.e., taking advantage of the fact the decision problems can be friendly formulated in a declarative way, and solved with guarantee the response DO NOT KNOW will not be allowed). So, besides of the right constraint programming environment selection and the reference model implementation the complexity of a class of real life problems guaranteeing an interactive DSS application should be estimated**.**

Detailed discussion concerning the reference model structure as well as developed methodology can be found in [2], [8], [9]. Brief introduction to design and functioning of decision support system developed due to this methodology provides the section below.

2. DECISION SUPPORT TOOL FOR PROJECT PORTFOLIO PROTOTYPING

The considered ***Decision Support Tool for Project Portfolio Prototyping*** **(DST4P^3)** aimed at project planning in small and medium sized enterprises (SME) has been developed in Oz Mozart [18] and Delphi languages environment. The main components of the **DST4P^3** structure are shown in Fig. 3. The system considered is composed of two modules serving for computations and interfacing, respectively. Of course, the main role plays the first module responsible for implementation of the reference model specified in terms of a fuzzy constraint satisfaction problem [9] and operation of an inference engine (implementing the logic-algebraic method) operation [8], [10]). Moreover, the module employs procedures enabling constraint compression and time effective searching strategies [6] as well as a newly introduced algebraic and logic operations allowing to calculate fuzzy constraints including fuzzy numbers [3], [9].

The second module of the **DST4P^3** enables problems specification, i.e. input data insertion, and queries selection, as well as an output data visualization and documentation. The following kinds of project planning problems are allowed:

- "straight" with distinct variables specifying the SME at hand,
- "straight" with imprecise variables specifying the SME at hand,

- “reverse” with distinct variables specifying the SME at hand,
- “reverse” with distinct variables specifying the SME at hand.

Illustrative examples of the **DST4P^3** application to the above mentioned problems provide the section below.

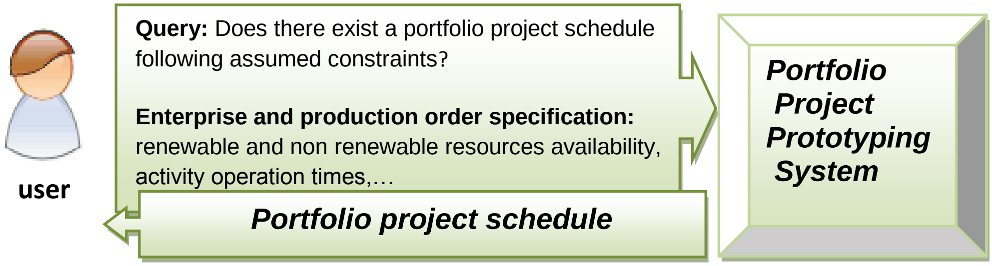


Fig. 3. Mains components of DST4P^3 structure

3. APPLICATIONS – ILLUSTRATIVE EXAMPLES

Example 1 – „straight”/distinct variables

Given the following projects portfolio, i.e. the set of projects $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$. Activities $O_{i,j}$ of projects are specified by corresponding sets: $P_1 = \{O_{1,1}, \ldots, O_{1,10}\}$, $P_2 = \{O_{2,1}, \ldots, O_{2,12}\}$, $P_3 = \{O_{3,1}, \ldots, O_{3,11}\}$, $P_4 = \{O_{4,1}, \ldots, O_{4,13}\}$. The relevant activity networks [2] are shown on the following figures: Fig. 4, Fig. 5, Fig. 6, and Fig. 7.

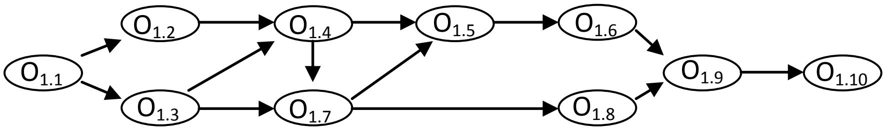


Fig. 4. Activity network for the project P_1

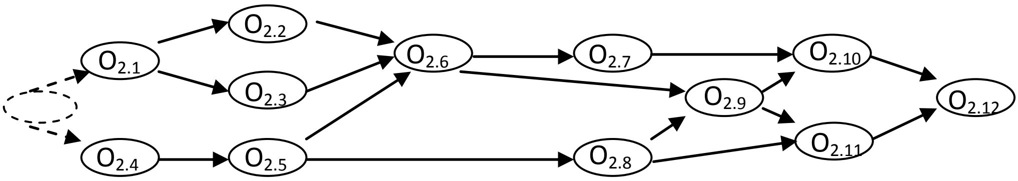


Fig. 5. Activity network for the project P_2

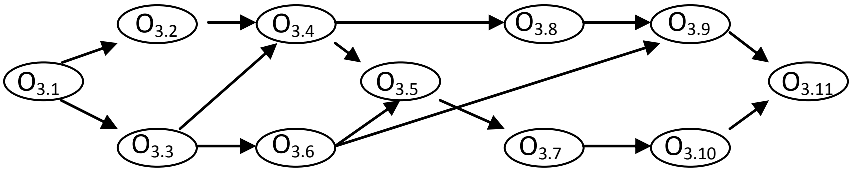


Fig. 6. Activity network for the project P_3

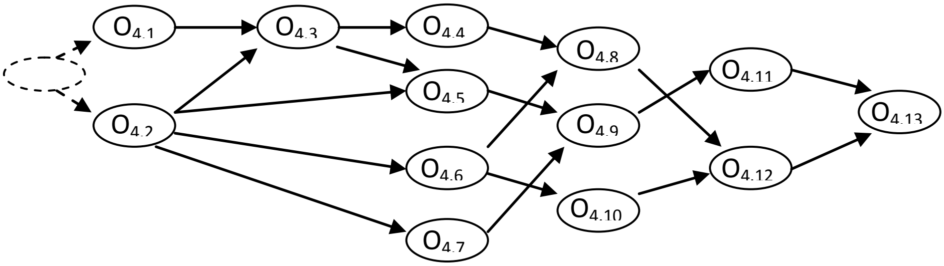


Fig. 7. Activity network for the project P_4

Given the time horizon $H = \{0,1,\ldots,40\}$. Operation times for particular projects P_1, P_2, P_3, P_4 are determined by the following sequences:

$T_1 = (1, 2, 3, 4, 4, 8, 3, 2, 1, 6)$, $T_2 = (3, 1, 6, 3, 2, 5, 1, 5, 2, 4, 2, 1)$,
$T_3 = (3, 7, 2,7, 2, 1,8, 3, 3, 4, 8)$, $T_4 = (3, 3, 2, 8, 3, 1, 4, 1, 8, 4, 3, 3, 8)$.

Given are three kinds of renewable resources ro_1, ro_2, ro_3. Resources' amounts are limited by following units number: 11, 14, 12 , respectively. Resource amounts are constant in whole time horizon H. That is assumed the relevant amount of resources required by particular activity can be released only by this activity and only at the moment of its completion. The amounts of particular resources required by projects' P_1, P_2, P_3, P_4 activities are given in the following tables: Table 2, Table 3, Table 4, and Table 5.

Tab. 2. Amounts of resources required by activities of the project P_1

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

Tab. 3. Amounts of resources required by activities of the project P_2

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

Tab. 4. Amounts of resources required by activities of the project P_3

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

Tab. 5. Amounts of resources required by activities of the project P_4

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

That is assumed some activates besides of renewable resources require also non-renewable resources. Given are two kinds of non-renewable resources rn_1, rn_2. Initial amount of the resource rn_1 is equal to 10 units, and of the resource rn_2 is equal to 7 units. Activities may use up and generate some number of resources rn_1, rn_2 units. That is assumed each activity uses up some resource units at the beginning and generates some resource units at the activity's end. The amounts of used up and generated resource rn_1 units determine sequences: $CR_{i,j}$, $CS_{i,j}$ respectively in the following tables: Table 6, Table 7, Table 8, and Table 9.

Tab. 6. Amount of used up (CR) and generated (CS) non-renewable resources required by activities of the project P_1

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

Tab. 7. Amount of used up (CR) and generated (CS) non-renewable resources required by activities of the project P_2

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

Tab. 8. Amount of used up (CR) and generated (CS) non-renewable resources required by activities of the project P_3

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

Tab. 9. Amount of used up (CR) and generated (CS) non-renewable resources required by activities of the project P_4

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

Let us assume each project's efficiency is measured by Net Present Value (*NPV*) performance index calculated due to the following formulae:

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

where: CF_t – the money netto flow expected in the year t,
k – the discount rate (alternative capital investment cost),
n – the period of a project exploitation [years].

The problem considered belongs to the class of „straight" ones and reduces to the following question: Does there exist a schedule following constraints assumed on availability of renewable and non-renewable resources and $NPV > 0$ such that production orders completion time not exceeds the deadline h?

Solution to the problem results in determination of moments the activities start their execution $x_{i,j}$ [8]. So, the solution we are searching for has the form of the following sequences: $X_1 = (x_{1,1}, \ldots, x_{1,10})$, $X_2 = (x_{2,1}, \ldots, x_{2,12})$, $X_3 = (x_{3,1}, \ldots, x_{3,11})$, $X_4 = (x_{4,1}, \ldots, x_{4,13})$.

The first admissible solution provided by **DST4P**[3] (obtained in 10 s) has the following form:

X_1 = (0, 1, 1, 4, 11, 15, 8, 11, 23, 24), X_2 = (0, 3, 7, 10, 13, 15, 20, 17, 23, 25, 25, 29),
X_3 = (0, 3, 3, 10, 17, 5, 19, 17, 20, 27, 31), X_4 = (0, 0, 3, 5, 5, 3, 3, 13, 8, 6, 14, 16, 19).

The NPV index value calculated for projects: P_1, P_2, P_3, P_4 follow the requirement $NPV > 0$, i.e. NPV_{P1} = 0,3649, NPV_{P2} = 2,4775, NPV_{P3} = 1,3248, NPV_{P4} = 0,8134 **.**

The graphical representation of the projects portfolio schedule is show in the Fig. 8. The schedule obtained follows all constrains imposed by an enterprise capacity and projects execution requirements. The system considered allows one to obtain the Gantt's-like chart illustrating the rates of resources usage both renewable and non-renewable ones. An example of graphical representation of the resource zo_1 usage rate containing assumed resource's limit in whole time horizon is shown on Fig. 9. It can be observed the assumed resource's limit was not exceeded, the same regards of resources zo_2, zo_3.

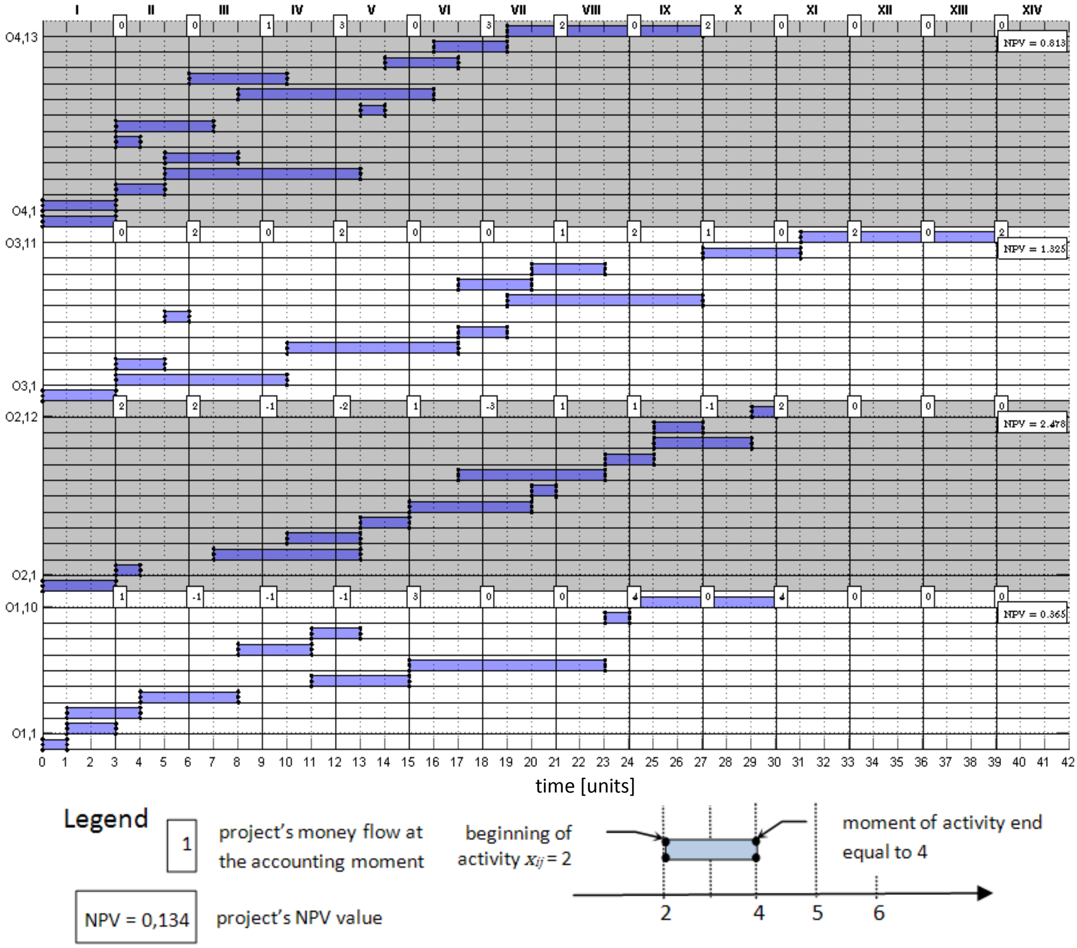


Fig. 8. Projects portfolio schedule

The Fig. 9 in turn illustrates changes regarding the rate of resource usage concerning of the non-renewable resource zn_2. That is easy to note that the assumed minimal level of resource usage equal to 0 was never exceeding in whole time horizon. The same remark concerns the resource zn_1.

Therefore, the example presented illustrates the main capabilities of the **DST4P[3]** package possessing capability of multi-criteria project planning (e.g. taking into account a particular project deadline, projects portfolio deadline, resources limits, and so on) and an interactive approach to projects prototyping problems formulated either in a straight or in a reverse way. The problem of the size just considered took less than 5 minutes (the AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1,00 GB platform has been used).

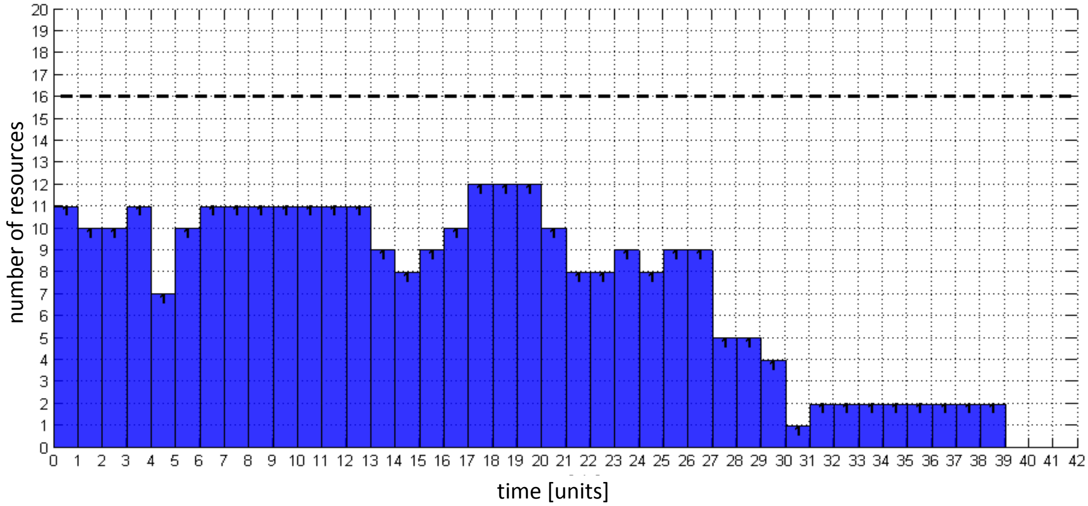


Fig. 9. Gantt's-like chart of the renewable resource zo_1 usage

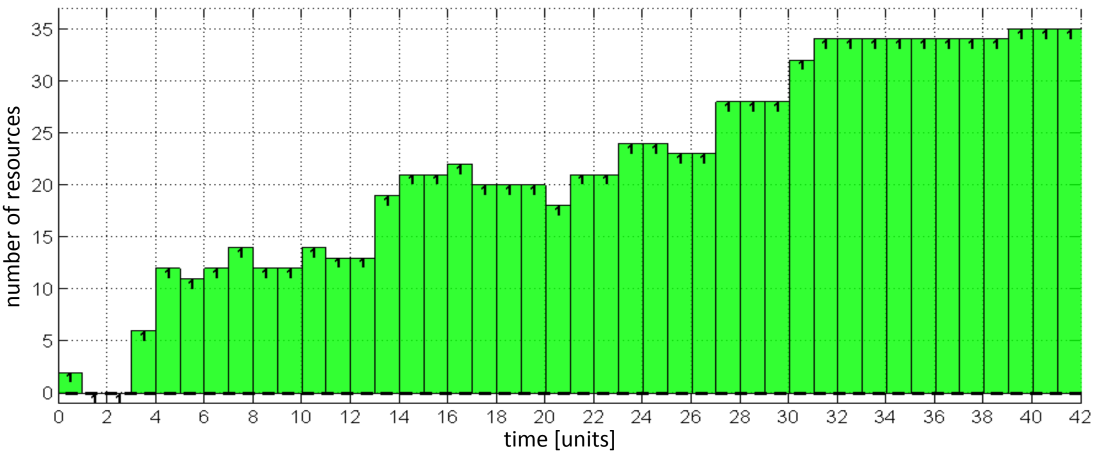


Fig. 10. Gantt's-like chart of the non-renewable resource zn_2 usage

Example 2 – „straight"/precise variables

Given the following projects portfolio, i.e. the set of projects $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$ specified by the same activity networks (see Fig. 4, Fig. 5, Fig. 6 and Fig.7) and resources allocations (see the Table 2 – Table 9) as in the Example 1. However, the new time horizon $H = \{0,1,\ldots,36\}$ is considered.

The problem considered belongs to the class of „straight" ones and reduces to the following question: Does there exist a schedule following constraints assumed on availability of renewable and non-renewable resources and $NPV > 0$ such that production orders completion time not exceeds the deadline h?

Similarly to the previous case the solution to the problem results in determination of the moments activities start their execution $x_{i,j}$. So, the solution we are searching for has the same form of the following sequences: $X_1 = (x_{1,1},...., x_{1,10})$, $X_2 = (x_{2,1},...., x_{2,12})$, $X_3 = (x_{3,1},..., x_{3,11})$, $X_4 = (x_{4,1},..., x_{4,13})$, however regards of the shorter deadline.

In the case considered, in 2 seconds, the **DST4P**[3] package's response was: ***Lack of any solutions.*** That means no schedule there exists. In such situation, however there is still a possibility to reformulate the problem considered by stating it in terms of imprecise variables, i.e. looking for a solution specified by an uncertainty measure. Such the case is just considered below.

Example 3 – „straight"/imprecise variables

Given the following projects portfolio, i.e. the set of projects $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$ specified by the same activity networks (see Fig. 4, Fig. 5, Fig. 6 and Fig.7) and resources allocations (see Table 2 – Table 9) as in the Example 1. However, the new time horizon $H = \{0,1,…,36\}$ is considered.

Given the uncertainty threshold value $DE \geq 0,7$ limiting uncertainty of constraints specifying projects portfolio. So, DE determines the minimal grade value guaranteeing the all constraints hold. For instance, DE = 0,9 means that the makespan of the projects portfolio considered will hold within the given time horizon H. Moreover, the operations times of activities: $O_{1,6}$, $O_{1,10}$, $O_{2,3}$, $O_{2,6}$, $O_{2,8}$, $O_{3,2}$, $O_{3,4}$, $O_{3,7}$, $O_{3,11}$, $O_{4,4}$, $O_{4,9}$, $O_{4,13}$ have an imprecise character. So, the relevant sequences of activities' operation times are as follow:

$\hat{T}_1$ = (1, 2, 3, 4, 4, "about 6", 3, 2, 1, "about 4"),
$\hat{T}_2$ = (3, 1, "about 4", 3, 2, "about 3", 1, "about 4", 2, 4, 2, 1),
$\hat{T}_3$ = (3, "about 5", 2, "about 5", 2, 1, "about 6", 3, 3, 4, "about 6"),
$\hat{T}_4$ = (3, 3, 2, „about 6", 3, 1, 4, 1, „about 6", 4, 3, 3,"about 6").

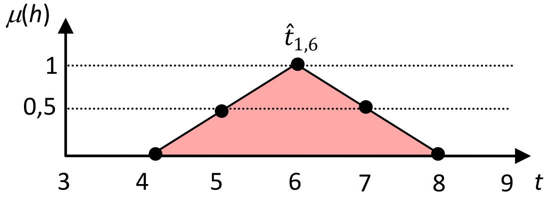


Fig. 11. $\hat{t}_{1,6}$ = "about 6 ".

For instance, in the case the activity's $O_{1,6}$ operation time is „about 6" (see Fig. 11) that means the activity can be executed within the time period of 4 till 8 units of time. In order to be able to distinguish crispy and imprecise variables the following symbol „^" is used.

The problem considered belongs to the class of „straight" ones and reduces to the following question: Does there exist a schedule following constraints assumed on availability of renewable and non-renewable resources and $NPV > 0$ such that production orders completion time not exceeds the deadline h with the uncertainty threshold $DE \geq 0,7$?

The proposed problem formulation assumes the solutions obtained are imprecise, so their implementation can be risky because of such uncertainty. Moreover we assume some variables e.g. operations times, and uncertainty threshold value DE are imprecise.

Similarly to the Example 1 the solution to the problem results in determination of the moments activities start their execution $x_{i,j}$. So, the solution we are searching for has the same form of the following sequences: $X_1 = (x_{1,1}, \ldots, x_{1,10})$, $X_2 = (x_{2,1}, \ldots, x_{2,12})$, $X_3 = (x_{3,1}, \ldots, x_{3,11})$, $X_4 = (x_{4,1}, \ldots, x_{4,13})$, however regards of the shorter deadline, as in the Example 2.

The first admissible solution provided by **DST4P**[3] (obtained in 10 s) has the following form:

$X_1 = (0, 1, 1, 4, 11, 15, 8, 11, 22, 23)$, $X_2 = (0, 3, 10, 10, 13, 15, 19, 18, 23, 25, 25, 29)$,
$X_3 = (0, 3, 3, 9, 15, 5, 17, 15, 18, 24, 28)$, $X_4 = (0, 0, 3, 5, 5, 3, 3, 13, 8, 6, 14, 16, 19)$.

The NPV index value calculated for projects: P_1, P_2, P_3, P_4 follow the requirement $NPV > 0$, i.e. $NPV_{P1} = 0{,}3649$, $NPV_{P2} = 2{,}6024$, $NPV_{P3} = 1{,}6177$, $NPV_{P4} = 0{,}8165$

The graphical representation of the projects portfolio schedule is show in the Fig. 12. The schedule obtained follows all constrains imposed by an enterprise capacity and projects execution requirements.

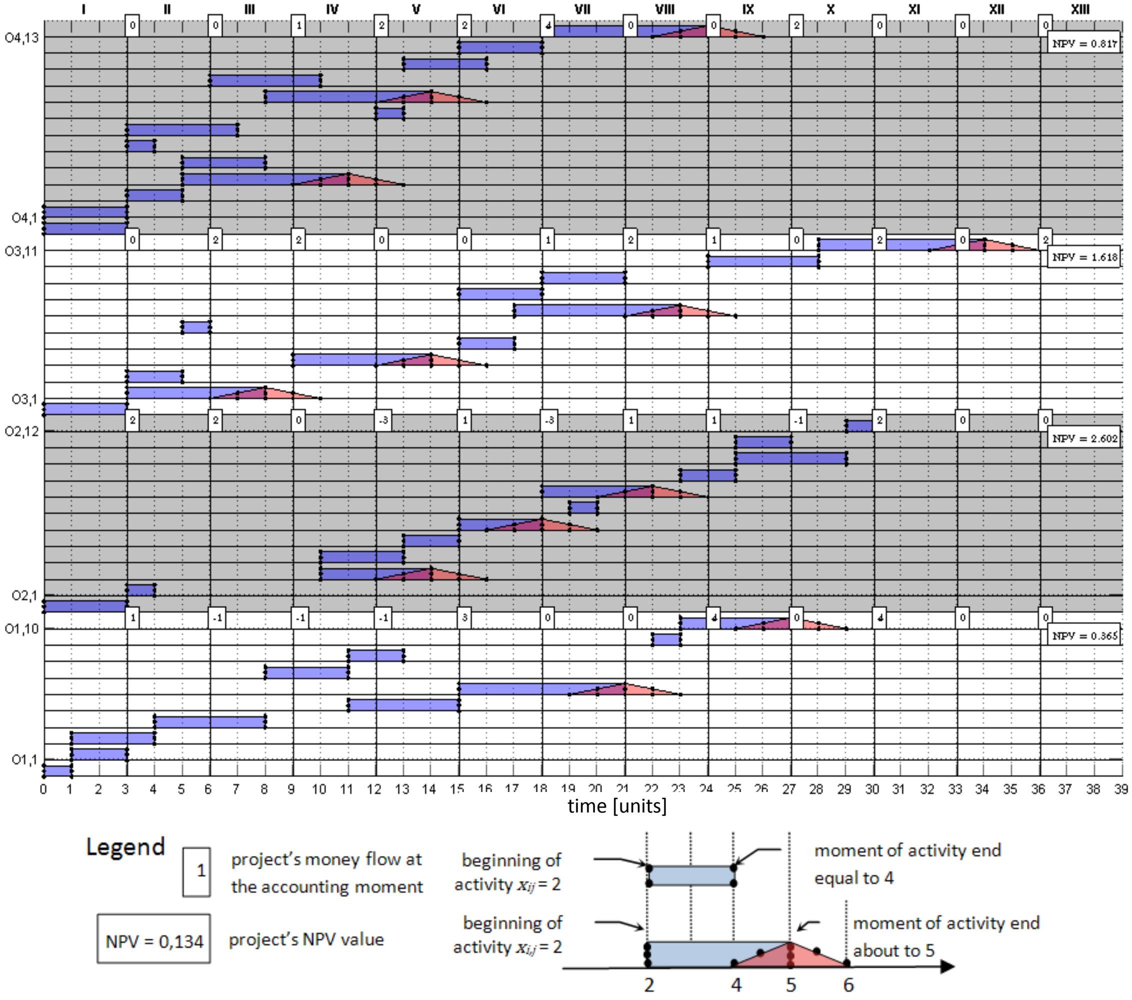


Fig. 12. Projects portfolio schedule with the uncertainty threshold ≥ 0,7

Obtained schedule provides the plan for projects portfolio execution, where uncertainty threshold level for all constraints is equal or less than 0.7 (that one may interpret as a risk of due time completion on the level equal to 0.3). The level the planed schedule fits its real live execution depends on a decision maker. The way such fitting can be adjusted is show in the Example 5.

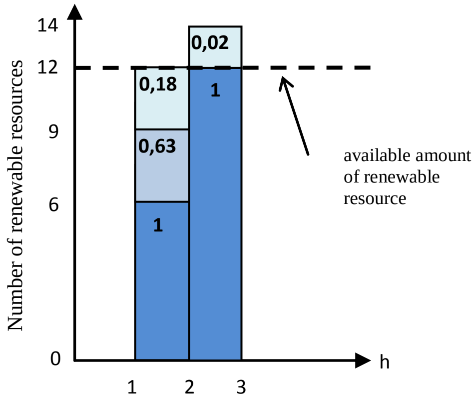


Fig. 13. Illustration of the renewable resource usage rate estimation

The system considered allows one to obtain the Gantt's-like chart illustrating the rates of resources usage both renewable and non-renewable ones. An example of graphical representation of the renewable resource zo_4 usage rate containing assumed resource's limit (equal to 12 units) in whole time horizon is shown on Fig. 14. Assumed resource's limit is distinguished by bold and dashed line. The chart considered provides information about the number of currently used resources units. For instance (see Fig. 13), between the first and the second unit of time there are used six resource units (i.e. with certainty equal to 1).

That number changes, however with uncertainty level, for instance with the uncertainty level equal to 0,18, the 12 resource units are required. In turn, in the period between the second and the third units of time a risk level equal to 0,02 resulting in exceeding the assumed resource limit (by two units) is observed. The similar cases concerning the resource limits exceeding can be observed for the resource zo_3 (see Fig. 14). That means, due to the schedule from Fig. 12, resource shortage may occur at the following time units 12, 15 i 19.

Quite similar observations regards of the non-renewable resource zn_2 see Fig. 15. In this case, however any risk of the resource shortage does not occur. In turn, the constraint determining the minimal, i.e. equal to 0, amount of renewable resources holds in both kinds of resources in whole time horizon.

The problem of the size just considered took less than 1 minute (the AMD Athlon(tm)XP 2500 + 1.85 GHz, RAM 1,00 GB platform has been used).

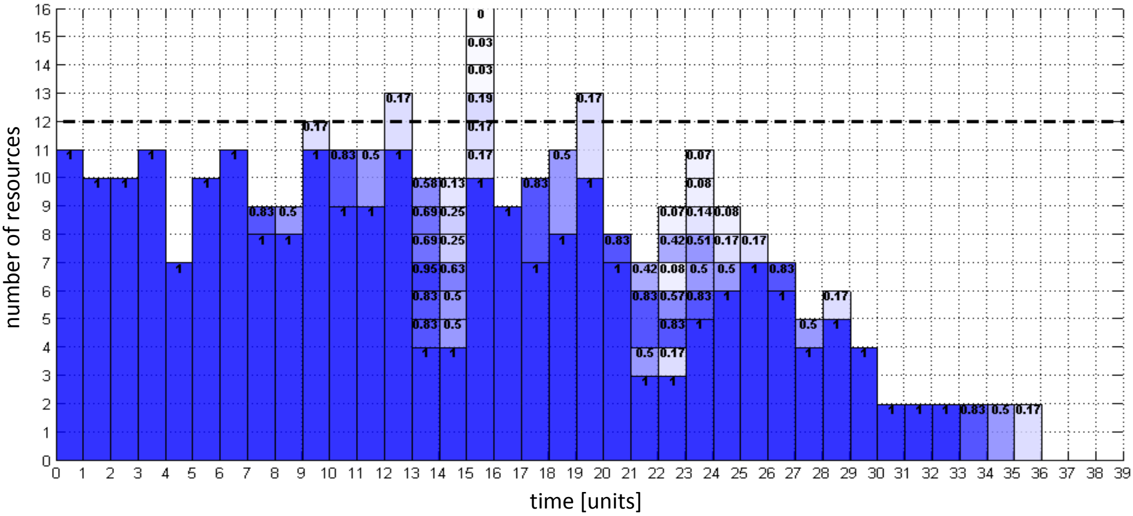


Fig. 14. Usage rate of renewable resource zo_3

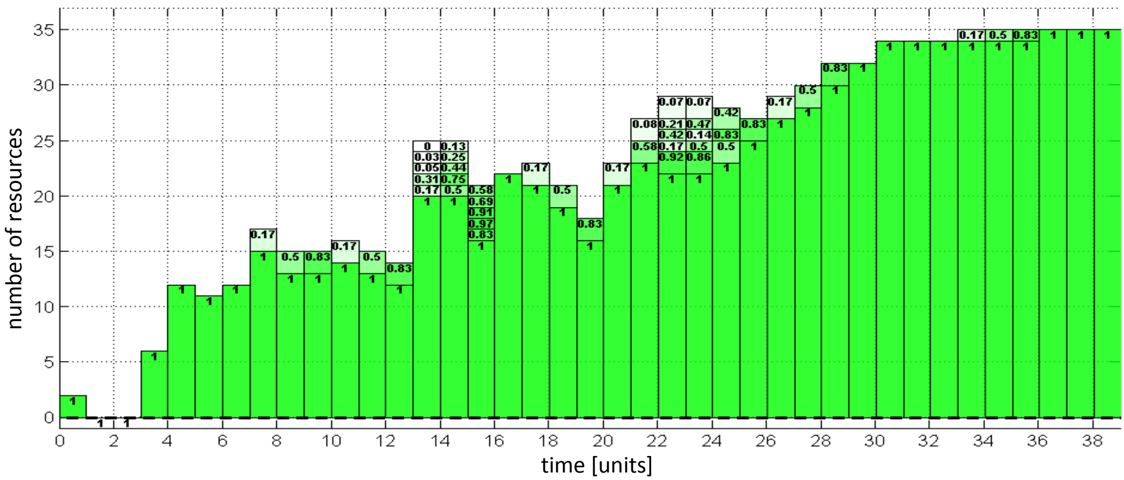


Fig. 15. Usage rate of non-renewable resource zn_2

Example 4 – „straight"/imprecise variables

Given the following projects portfolio, i.e. the set of projects $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$ as in the Example 3. However, the new, shorter time horizon $H = \{0,1,\ldots,33\}$ is considered.

The problem considered belongs to the class of „straight" ones and reduces to the following question: Does there exist a schedule following constraints assumed on availability of renewable and non-renewable resources and $NPV > 0$ such that production orders completion time not exceeds the deadline h with the uncertainty threshold $DE \geq 0{,}7$?

In the case considered, in 3 seconds, the **DST4P³** package's response was: ***Lack of any solutions.*** That means no schedule there exists under assumption of the uncertainty threshold $DE \geq 0{,}7$. In such situation, however there is still a possibility to reformulate the problem considered by assuming greater uncertainty level (less value of *DE*). In case the increasing of an uncertainty level is not acceptable the decision maker can reformulate the problem statement, e.g. stating it in a reverse way. In such a new formulation the decision maker is looking for the values of a subset of assumed decision variables guaranteeing the makespan of considered projects portfolio will not exceed the deadline h. Such the case is just considered below.

Example 5 – „reverse"/imprecise variables

Given the following projects portfolio, i.e. the set of projects $\mathbb{P} = \{P_1, P_2, P_3, P_4\}$ as in the Example 4. $H = \{0,1,\ldots,33\}$ is the time horizon considered.

Given the uncertainty threshold value $DE \geq 0{,}7$ limiting uncertainty of constraints specifying projects portfolio. Similarly to the Example 4, the operations times of activities: $O_{1,6}$, $O_{1,10}$, $O_{2,3}$, $O_{2,6}$, $O_{2,8}$, $O_{3,2}$, $O_{3,4}$, $O_{3,7}$, $O_{3,11}$, $O_{4,4}$, $O_{4,9}$, $O_{4,13}$ have an imprecise character. Moreover, let us assume that besides of operation times of the following activates $O_{3,7}$ and $O_{3,11}$ the values of all rest parameters are known. Given is the following relationship linking operation times of activates $O_{3,7}$ and $O_{3,11}$: $\hat{t}_{3,7} + \hat{t}_{3,11} = \text{"}about\ 8\text{"}$.

The problem considered belongs to the class of „reverse" ones and reduces to the following question: Do there exist such activities operation times guaranteeing production orders completion time not exceeds the deadline h with the uncertainty threshold $DE \geq 0{,}7$ while constraints assumed on availability of renewable and non-renewable resources and $NPV > 0$ hold?

In such problem formulation the searched solution regards of the values of some decision variables guaranteeing the makespan of considered projects portfolio do exceed the deadline h. So, in the particular case the considered variables are activities $O_{3,7}$ and $O_{3,11}$, and the searched values concerns of activities operation times $\hat{t}_{3,7}, \hat{t}_{3,11}$ and moments of activities $x_{i,j}$ beginning, i.e. the components of the following sequences: $X_1 = (x_{1,1}, \ldots, x_{1,10})$, $X_2 = (x_{2,1}, \ldots, x_{2,12})$, $X_3 = (x_{3,1}, \ldots, x_{3,11})$, $X_4 = (x_{4,1}, \ldots, x_{4,13})$.

The first admissible solution provided by **DST4P³** (obtained in 15 s) has the following form concerning the operation times: $\hat{t}_{3,7} = 6$, $\hat{t}_{3,11} = \text{"}about\ 3\text{"}$ (see Fig. 16)

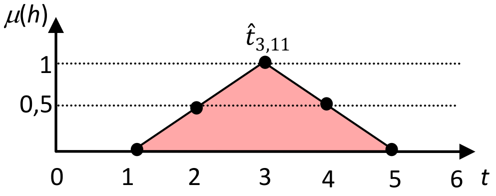


Fig. 16. $\hat{t}_{3,11} = \text{"about 3"}$

and the following form concerning the moments of activities start:

$X_1 = (0, 1, 1, 4, 11, 15, 8, 11, 22, 23)$, $X_2 = (0, 3, 10, 10, 13, 15, 19, 18, 23, 25, 25, 29)$,
$X_3 = (0, 3, 3, 9, 15, 5, 17, 15, 18, 24, 28)$, $X_4 = (0, 0, 3, 5, 5, 3, 3, 12, 8, 6, 13, 15, 18)$.

The NPV index value calculated for projects: P_1, P_2, P_3, P_4 follow the requirement $NPV > 0$, i.e. NPV_{P1} = 0,3649, NPV_{P2} = 2,6024, NPV_{P3} = 1,6014, NPV_{P4} = 0,8165.

The graphical representation of the projects portfolio schedule is show on the Fig. 17. The schedule obtained assumes operation times of activities $O_{3,7}$, $O_{3,11}$ equal to "about 6" and "about 3", respectively, and follows all constrains imposed by an enterprise capacity and projects execution requirements.

Obtained schedule provides the plan for projects portfolio execution, and provides a base for further adjustment aimed at fitting to real live execution [3]. The adjustment process consist in narrowing down the periods of operation times (by changing the beginning and ending moments of activities executions) as to avoid their overlapping, i.e. removing the confusion regarding the cases where an activity's ending exceeds its begging. It that context the schedule fitting leads to a minimal, confusion-free periods of operation times. The illustration of such fitting is shown on Fig. 18 presenting the projects portfolio schedule assuming the uncertainty threshold value $DE \geq 0,5$. The new schedule has been obtained from the former one shown in the Fig. 17 under assumption the uncertainty threshold value $DE \geq 0,7$.

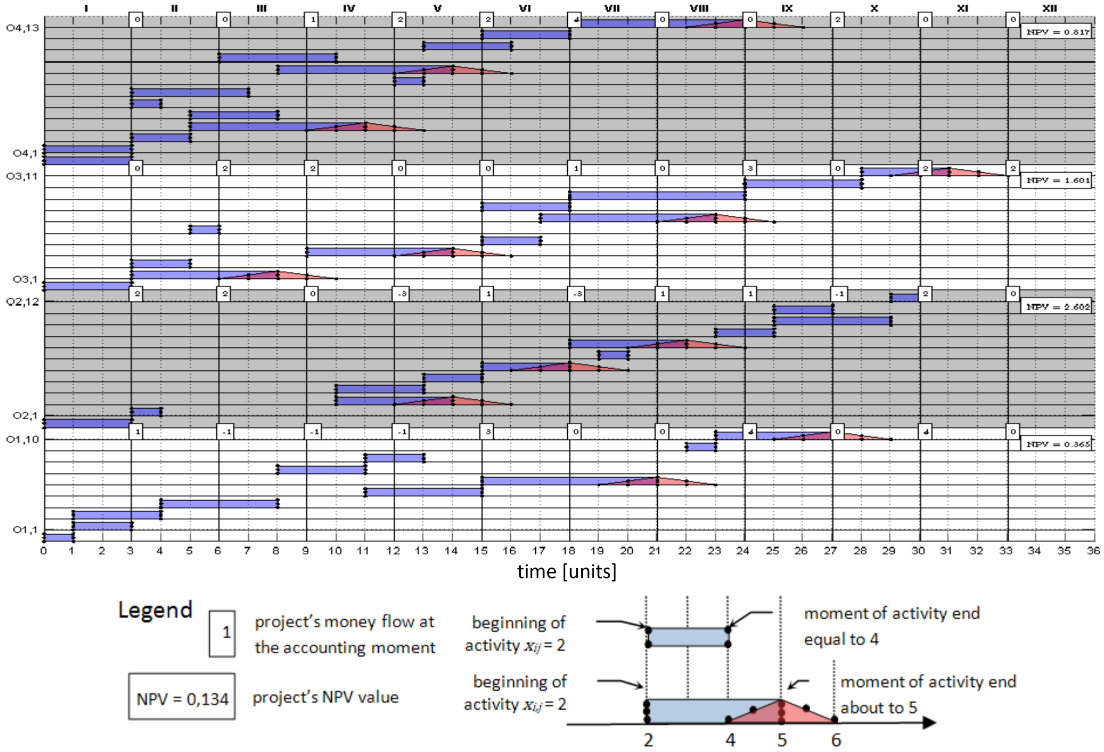
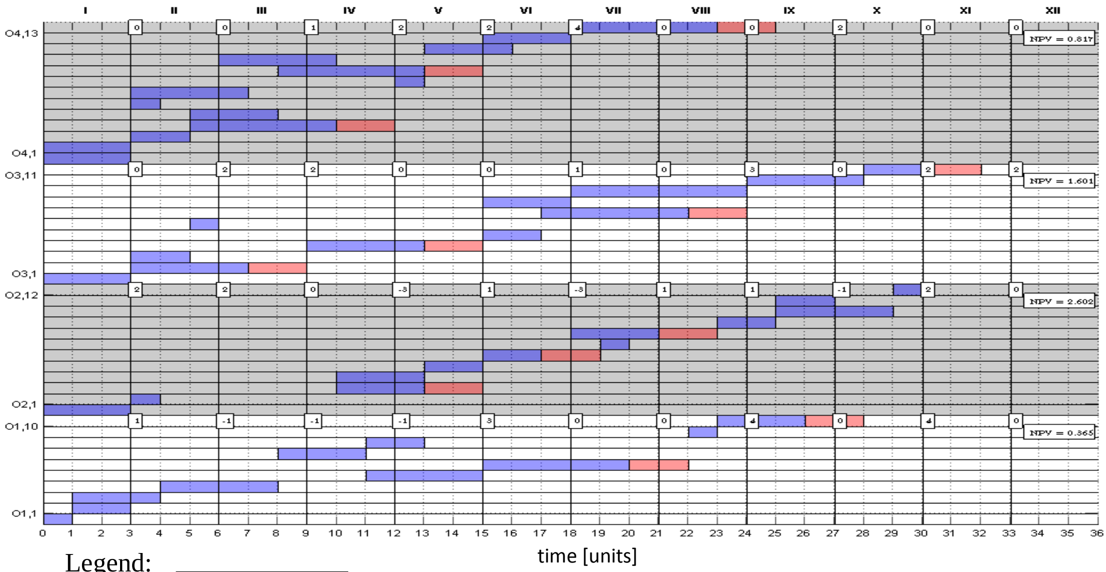


Fig. 17. Projects portfolio schedule with the uncertainty threshold ≥ 0,7



Legend: - the period the activity starts its execution

- the period the activity finish its execution

Fig. 18. Projects portfolio schedule with the uncertainty threshold $\geq 0{,}5$

4. CONCLUDING REMARKS

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

The methodology developed is based on the concept of the decision problem reference model [3]. The model considered can be seen as a knowledge base encompassing the structure of a constraint satisfaction problem, where the logic-algebraic method plays a role of inference engine. So, the main idea standing behind of the methodology lies in a way the knowledge base is “adjusted”, i.e. adding the conditions guaranteeing the responses to the standard queries there exist as well as conditions guaranteeing the employed them searching strategies can be used in an on-line mode for the real-life size of project planning problems.

Provided multiple examples illustrate a way some arbitrary selected cases can be managed by **DST4P**^{3} package. Its current version is aimed at an interactive projects portfolio prototyping aimed at SMEs where the number of simultaneously considered projects do not exceeds 5 and whole number of activities do not exceeds 80. In that context the approach presented can be considered as a new alternative contribution to project-driven production flow management, and its DSS implementation can be applied in make-to-order manufacturing as well as for prototyping of the virtual organization structures.

References

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

[2] BACH I., BOCEWICZ G., BANASZAK Z.: *Constraint programming approach to time-window and multiresource-constrained projects portfolio prototyping*. In: Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2008, N.T. Nguyen et al. (Eds.):, Lecture Notes in Artificial Intelligence 5027, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 767–776.

[3] BACH I., WÓJCIK R. BOCEWICZ G.: *Projects portfolio prototyping subject to imprecise activities specification,* In: Conference proceedings of 14th International Congress of Cybernetics and Systems of WOSC – ICCS'08, Wroclaw, Poland, 2008; 261-272.

[4] 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.

[5] BANASZAK Z., ZAREMBA M,. 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.

[6] 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; p. 7-14.

[7] 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.

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

[9] BOCEWICZ G., WÓJCIK R., BZDYRA K.: *Fuzzy Logic And Logic-Algebraic Method For Constraint Programming-Driven Project Prototyping*. W: Zarządzanie wiedzą i technologiami informatycznymi, Red. Orłowski C., Kowalczuk Z., Szczerbicki E., PWNT Gdańsk 2008, 317-326.

[10] BUBNICKI Z.: *Logic-algebraic method for a class of knowledge based systems*. F. Picher, R. Moreno Diaz (red.). Computer Aided Systems Theory. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1333, 1997.

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

[12] CHANAS S., KOMBUROWSKI J.: *The use of fuzzy variables in PERT, Fuzzy Sets and Systems*, 5(1), 1981,11-19,

[13] 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.

[14] 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.
[15] MARTINEZ, E. C., D., DUJE G.A. PEREZ: *On performance modeling of project-oriented production*. Computers and Industrial Engineering, Vol. 32, 1997; 509-527.
[16] PUGET J-F.: *A C++ Implementations of CLP*, Proceeding of SPICS 94, 1994.
[17] PIEGAT A.: *Fuzzy modeling and control*, Exit, Warsaw, 1999.
[18] SCHUTLE H., SMOLKA G., WURTZ J.: *Finite Domain Constraint Programming in Oz*. German Research Center for Artificial Inteligence, Germany, D-66123 Saarbrucken, 1998.
[19] 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.
[20] VAN HENTENRYCK P.: *Constraint Logic Programming*, Knowledge Engineering Review, 6, 1991, 151–194.
[21] ZIMMERMANN H.J.: *Fuzzy sets theory and its applications*. London: Kluwer Academic Publishers. 1994.