decision support, constraint logic programming, production planning, modelling, scheduling

Izabela TOMCZUK*, Krzysztof BZDYRA**, Zbigniew BANASZAK***

TOWARDS CLP-BASED TASK-ORIENTED DSS FOR SME

Abstract

Decision making supported by task-oriented software tools plays a pivotal role in a modern enterprises. That is because commercially available ERP systems are not able to respond in an interactive on-line/real-time mode. It means a new generation of DSS that enable a fast prototyping of production flows in multiproject environment as well as an integrated approach to a layout planning, production routing, batch-sizing and scheduling problems is needed. In that context, the constraint logic programming techniques allowing declarative representation of a decision making problem provide a quite attractive alternative. So, some issues regarding modelling of decision making and searching strategies development are discussed in the contribution. The results obtained are implemented in a software package supporting production flow planning in the SMEs. Illustrative example of the ILOG-based software application is provided.

1. INTRODUCTION

One of the most important factors contributing to the maintaining of a position by small and medium size enterprises (SME) on the consumer market is their ability to evaluate the market demand and a fast reaction to the demand. Most decisions taken in practice in the industry refer to the balance between the customer's needs and the manufacturer's abilities [1, 8, 10]. Parallel execution of work orders imposes a necessity to evaluate the time and cost abilities to execute a given work order (set of work orders) in a possibly short time. In order to achieve this it is necessary to plan the production flow from the stage of obtaining new work orders. Production flow planning requires solving simultaneously many different subproblems (e.g. batching, routing and scheduling).

The increased requirements concerning the time necessary to establish a production plan implies a need to apply methods and tools which facilitate a fast and cheap variants of alternative ways for taking and execution of production orders. The new ways of decision support in the project management process are supplied by tools, which are based on the Constraint Logic Programming (CLP). CLP techniques facilitate description of numerous real problems with regard to constraints

* M. Sc., Technical University of Opole, Department of Management and Production Engineering, 45-370 Opole, Poland, e-mail: itomczuk@po.opole.pl

** Ph. D., Technical University of Koszalin, Department of Electronics and Informatics, 75-453 Koszalin, Poland, e-mail: kbzdyra@tu.koszalin.pl

*** Prof., Technical University of Koszalin, Department of Electronics and Informatics, 75-453 Koszalin, Poland, e-mail: banaszak@tu.koszalin.pl

that specify certain relations between decision variables. Because of their declarative character they can be implemented in decision support systems [6, 8].

2. PROBLEM STATEMENT

Consider a manufacturing system providing a given production capability while processing some other work orders. So, only a part of the production capability (specified by the time-restricted resource availability) is available for use in the system.

A given production order is represented by an activity-on-node network, and specified by project duration deadline, which is equivalent to a presumed completion time (the production order cycle) as well as a total project cost constraint. Each activity may be executed in one out of the set of system resources. Also, each activity may not be pre-empted and the resource once selected may not be changed.

The problem consists in finding a makespan-feasible schedule that fulfils the constraints imposed by the precedence relations and by the time-constrained resources availability as well as assumed duration deadline.

Searching for feasible solutions, regarding for example resources allocation, time lags, makespan, costs, etc, has to be preceded by formulation of a feasibility problem or equivalently by a constraint satisfaction problem (CSP). Moreover, solution to a makespan-feasible problem permits a user to investigate the effect of a new production order impact on the performance of a manufacturing system. In other words, it enables finding an answer to the most important question whether a given production order can be accepted to be processed in the manufacturing system, i.e., whether its completion time, batch size, and its delivery period satisfy the customer requirements while satisfying constraints imposed by the enterprise capability [1].

3. CONSTRAINT LOGIC PROGRAMMING

CLP techniques can be applied in decision process support, both in production and in service enterprises [6, 10], e.g. at the planning of goods transportation in distribution networks [7, 9].

The most important issues that contribute to the efficiency of CLP techniques are the procedures of a feasible solution selection (constraints propagation, assignment) and searching strategies.

Constraint propagation procedures deal with the eliminating of decision variables, which are not in accordance with the constraints. This is supplemented with a mechanism, which assigns certain values to the variables (assignment). Linking of constraint propagation with assignment of variables facilitates setting a feasible solution or indicating lack of such solution.

To illustrate this, a simple problem can be analysed – a set of decision variables $\{x_1, x_2, x_3\}$, their domains: $x_1 \in \{1,...,7\}$, $x_2 \in \{1,...,4\}$, $x_3 \in \{3,...,11\}$ and a constraint set: $x_1 \le x_2 - 1$; $x_3 = 2x_1 + x_2; x_1 + x_3 > 6$. We are looking for the variable values, which meet the constraints.

As a result of the first constraint propagation the following solution was obtained: $x_1 \in \{1..3\}$, $x_2 \in \{2..4\}$, $x_3 \in \{4..10\}$. The searching space was reduced from 252 to 63 possible solutions. The next stage includes an assignment of the first decision variable value $x_1=1$ (fig. 1), and then a constraint propagation (third stage). As a result, the domains of decision variables become narrowed down to one-item set (i.e., to the first feasible solution). In general case, subsequent stages of assignment and propagation are repeated till the set of all possible assignments is worn out.

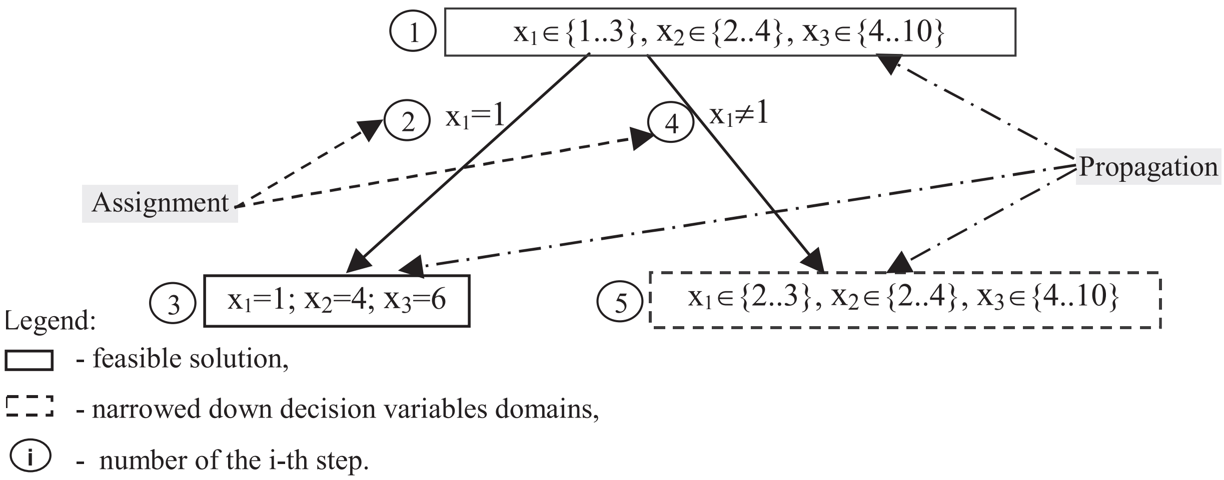


Fig.1. Searching process

Searching process of the potential solutions space (implementing the CLP techniques) can be executed with the application of methods based on a searching strategy *„depth-first"*, methods of *„iterative broadening IB"* or the methods of *„Limited Assignment Number Search"* (*LAN*)) [3]. The application of limited tree searching methods (*LAN*, *IB*) allows to control the solution searching process, which can shorten the calculation time in a significant way. It facilitates establishment of such a solution set which includes solutions differing in a significant way from the previously obtained solutions.

CLP techniques facilitate the application of numerous potential solution searching strategies [3, 15, 16]. Some of them may be implemented in the tools, which apply CLP methods (e.g. *IDFS*, *DFS*, *SBS*, and *DDS* in the OPL language of the Ilog system) [5, 8, 10] or may be implemented by the user (e.g. in the OZ language of the Mozart system).

The efficiency of a given searching strategy is significantly influenced by the sequence of assignments of decision variables (e.g. x_1-x_2-x_3 or x_3-x_1-x_2, etc.), as well as the way of assignment of the variable domain values (e.g., bottom-up or upper-down limit of a variable domain). For instance, in the example considered a *first fail* strategy is applied (see fig. 1). That is characterised by an assignment, which starts from the bottom limit of domain variables. Application of the *first fail* strategy provides (already on the third stage) the first feasible solution.

Experiments which were carried out with the use of the system Ilog OPL Studio 3.7 have proved the possibility to apply certain techniques for the establishment of efficient procedures for decision problems solving, including the production flow planning problems.

3.1. Constraint satisfaction problem

In order to balance the producer's abilities with the customer's requirements a producer – consumer model is proposed. The model consists of a production system model, which reflects the parameters of a potential production system, and a production order model, which takes into account the order's requirements.

The model of a production system and the model of production order include parameters (such as: constraint sets, sets of discrete decision variables), which assure the correctness of obtained solutions and their application. The variables reflect various values – from the resource availability periods, through the production and transportation batch sizes, to the deadlines and taking over prices of the particular batches (fig. 2, fig. 3).

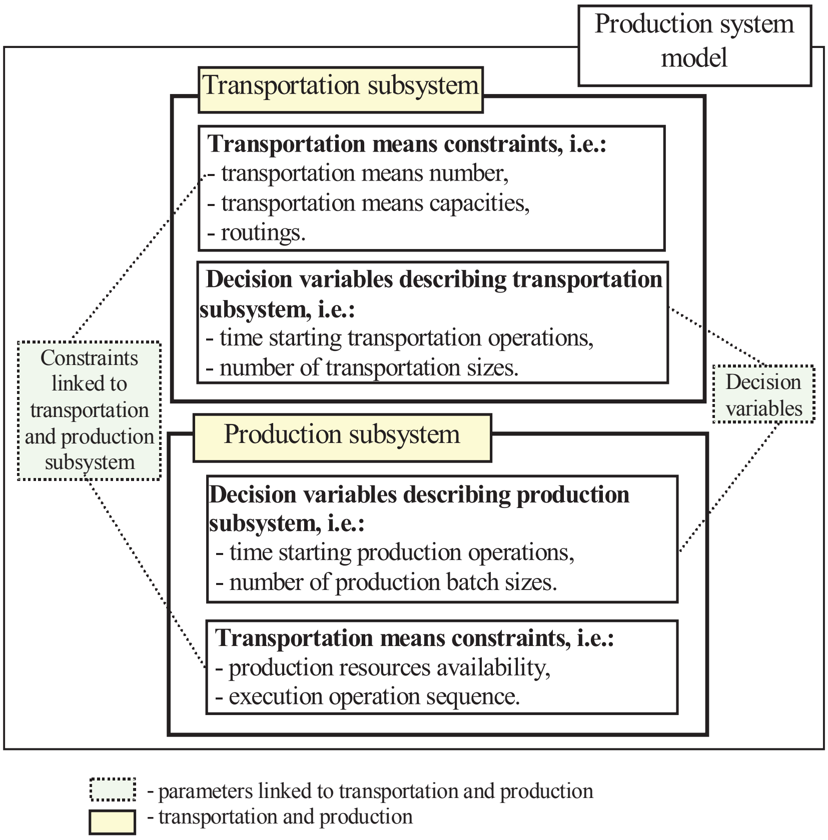


Fig.2. Production system model

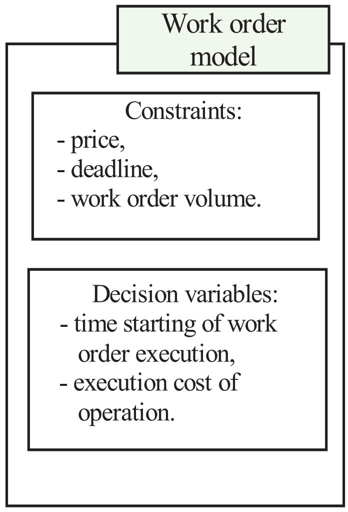


Fig. 3. Work order model

The solution of the problem is a feasible solution (a set of solutions), i.e. a solution which meets a set of constraints which link both decision variables describing a producer's capabilities as well as variables which characterize the conditions for the execution of a production order. Constraints which link some of the decision variables characterizing the producer-consumer relation. Fig. 4 presents an objective scheme of the model producer-consumer relation in which the production system constraints are determined by the order parameters. Constraints which link model production system and a production order are determined by the production order parameters. The constraints which link the model of a production system and a production order are: production order execution cost is at most equal with the adopted price level. The total number of production batches must be equal with the size of a given production order, the execution deadline of the last production batch does not exceed a directive production order execution deadline.

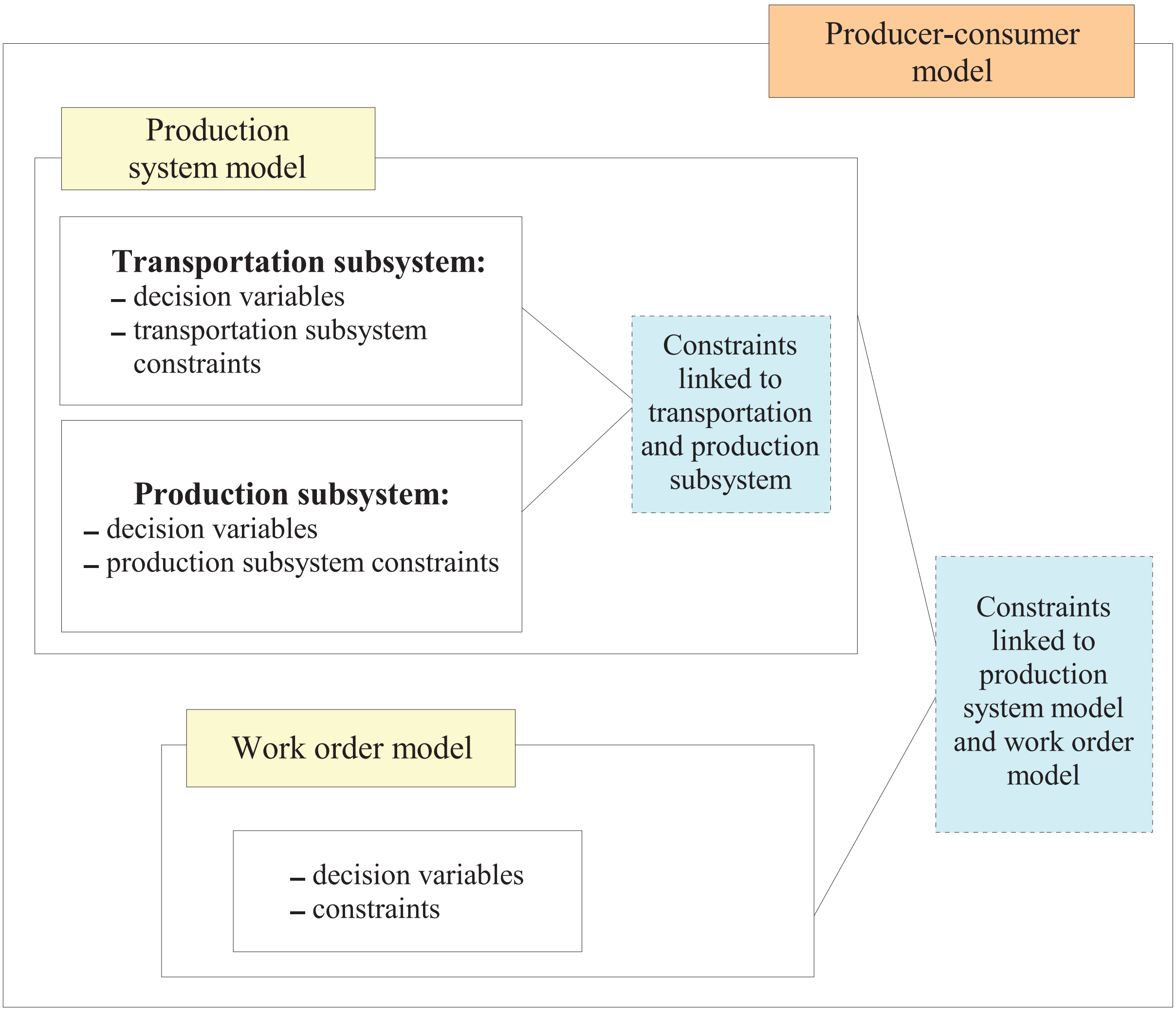


Fig.4. Producer-consumer model

Assumptions of the producer-consumer model:

1) Every resource, in a given time unit can execute at most one operation
Allocation of operation A_j to a resource Z_l takes the value zero or one

$$P_{A_j,t,z_l} = \begin{cases} 1 - \text{if an } A_j \text{ operation is allocated to a resource } Z_l \text{ in a time unit } t \in N \\ 0 - \text{otherwise} \end{cases}$$

$$\underset{t,z_l \in N}{\forall} \sum_{j=1}^{m} P_{A_j,t,z_l} \leq 1 \qquad (1)$$

2) The operations can not be preempted, the time of their execution is:

$$T_{A_j,Z_l} = tk_{A_j} - tp_{A_j} \tag{2}$$

where: tk_{A_j}, tp_{A_j} - signify, subsequently the time of finishing and commencement of operation A_j, on the resource Z_l.

3) A resource once chosen cannot be changed

$$\underset{\substack{j\in(1,m)\\ l\in(1,n)}}{\forall} \sum_{tp_{A_j}}^{tk_{A_j}} P_{A_j,t,z_l} = T_{A_j,z_l} \tag{3}$$

4) Available production capacities cannot be exceeded

$$\underset{\substack{l\in(1,n)\\ j\in(1,m)}}{\forall} \sum_{t=tp_{A_j}}^{TD_{B_i}} D_{Z_l,t} \geq T_{A_j,z_l} \tag{4}$$

where:

B_i - production order, i=1,...,b,

TD_{B_i} - directive deadline for the execution of an work order B_i,

$D_{z_l,t}$ - availability of a resource Z_l in a time unit t.

5) Every operation can be executed by at least one of the system resources

$$\underset{t,j}{\forall} \sum_{l=1}^{n} P_{A_j,t,z_l} \geq 1 \tag{5}$$

From the available works [3, 4] it results that over 95% of all manufacturing and services decision problems are included in the Constraint Satisfaction Problems (CSPs), for which many Constraint Programming (CP) languages were worked out (especially Constraint Logic Programming (CLP)). The declarative character of CP languages and a high efficiency in solving combinatorial problems creates an attractive alternative for the currently available (based on conventional operation research techniques) systems of computer-integrated management.

Consider the CSP that consists of a set of variables $X = \{x_1, x_2, \ldots, x_n\}$, their domains $D = \{D_i \mid D_i = [d_{i1}, d_{i2}, \ldots, d_{ij}, \ldots, d_{im}], i = 1..n\}$, and a set of constraints $C = \{C_i \mid i = 1..L\}$

A solution is such an assignment of the variables that all the constraints are satisfied.

The following CSP notation is applied: CSP = ((X,D),C), where $c \in C$ is a constraint specified by a predicate $P[x_k,x_l,\ldots,x_h]$ defined on a subset of the set X. In general case the CSP problem may be decomposed into a set of subproblems.

For the purpose of illustration lets us consider the following problem example:

Given a CSP = ((X,D),C), where $X = \{x_1,x_2,\ldots,x_{12}\}$, $D = \{D_1,D_2,\ldots,D_{12}\}$, $C = \{c_1,c_2,\ldots,c_8\}$, where: $c_1 = P_1[x_1,x_2,x_3]$, $c_2 = P_2[x_2,x_4,x_5]$, $c_3 = P_3[x_4,x_6]$, $c_4 = P_4[x_7,x_8]$, $c_5 = P_5[x_4,x_7]$, $c_6 = P_6[x_9,x_{10}]$, $c_7 = P_7[x_8,x_9]$, and $c_8 = P_8[x_{11},x_{12}]$. Two arbitrary chosen feasible decompositions of the CSP considered are shown in fig. 5. The subproblems that cannot be decomposed are side to be so called the elementary problems.

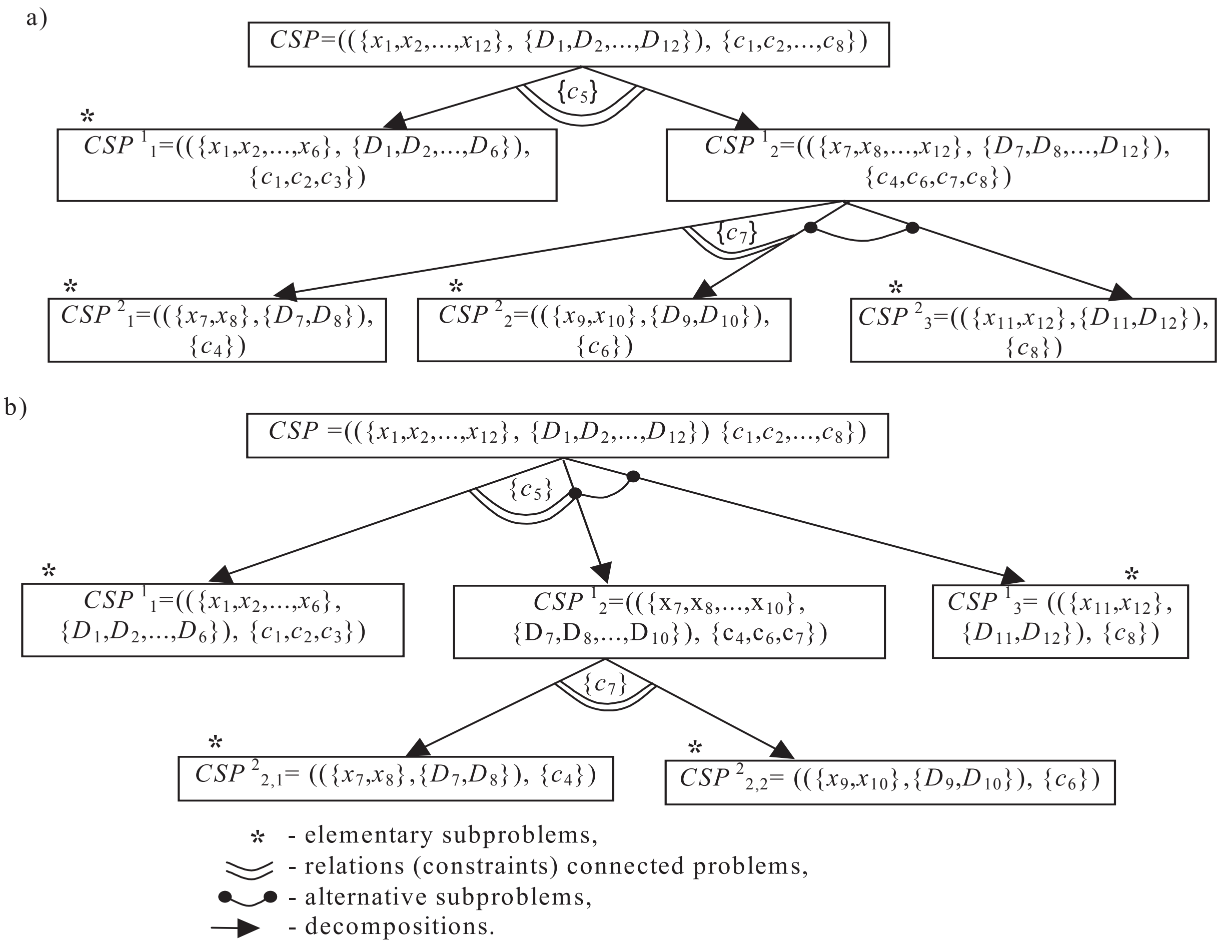


Fig.5. The CSP feasible decompositions

The presented example illustrates the possibility of choosing of the searching strategy that minimizes the number of potential backtrackings.

It is assumed that the available variants of possible searching strategy are subject to the principles of the CSP decomposition. They take into account available programming system operators, as well as the possible techniques of constraint propagation.

For the given specification of the problem it is necessary to assort such a method, which can solve it without introducing (assuming) any additional simplification. This observation implies the need to work out the reference model of constraint satisfaction problem decomposition. The model considered should be able to facilitate response to the following questions: what implementation of the CP/CLP language provides (if possible) solution to a given constraint satisfaction problem? – What searching strategy minimizes the number of potential backtrackings?

3.2. Reference model

The problem representation and the potential of the available CP/CLP language assume a possibility of CSP decomposing into a set of subproblems. The possible problem decompositions may be interpreted as appropriate searching strategies, determined by a specified number of subproblems and the sequence of solving them.

The following notation is applied:

$CSP = ((X,D),C)$ – the specification of constraints satisfaction problem.

$CSP^{1}_{j} = ((X^{1}_{j},D^{1}_{j}),C^{1}_{j})$ – the specification of the j-th subproblem of CSP.

$CSP^{i}_{j1,j2,\dots ji} = (X^{i}_{j1,j2,\dots ji}, D^{i}_{j1,j2,\dots ji}), C^{i}_{j1,j2,\dots ji})$ – the specification of the ji-th subproblem of the CSP^{1}_{j}, where $CSP^{i-1}_{j1,j2,\dots ji-1}$ - its direct decomposition

$(\{CSP^{i+1}_{j1,j2,\dots ji+1} | j_{i+1} \in \{1 \dots w\}\}, R^{i}_{j})$ – the graph-like representation of the $CSP^{i}_{j1,j2,\dots,ji}$, problem decomposition,

where: $CSP^{i+1}_{j1,j2,\dots ji+1}$– is the direct decomposition of the $CSP^{i}_{j1,j2,\dots,ji}$, and R^{i}_{j} – the relation imposed on the $\{CSP^{i+1}_{j1,j2,\dots ji+1} | j_{i+1} \in \{1,\dots,w\}\}$.

The reference model of decomposition of the CSP = ((X,D),C) problem refers to an object architecture $(\{CSP^{i+1}_{j1,j2,\dots,ji+1} | j_{i+1} \in \{1 \dots w\}\}, R^{i}_{j})\})$ meeting the following conditions:

$$\sum_{r=1}^{w} \left| {}^{r}C^{i}_{ji} \right| = \left| C^{i}_{ji} \right|$$

In order to simplify the notation the first i-th indexes of the $_{j1,j2,\dots,ji}$ are omitted, so the notation CSP^{i+1}_{ji+1} will stand for $CSP^{i+1}_{j1,j2,\dots,ji+1}$.

i) $CSP^{i+1}_{ji} = ((X^{i+1}_{ji},D^{i+1}_{ji}),C^{i+1}_{ji})$, where:

$X^{i+1}_{jr} = {}^{r}X^{i}_{ji}$; $X^{i}_{ji} = {}^{1}X^{i}_{ji} \cup {}^{r}X^{i}_{ji} \cup \dots \cup {}^{w}X^{i}_{j}$;

$\forall r \in \{1,\dots,w\} \mid {}^{r}X^{i}_{ji} \neq \varnothing$

$\forall r,u \in \{1,\dots,w\} \mid r \neq u \Rightarrow {}^{r}X^{i}_{ji} \cap {}^{u}X^{i}_{ji} = \varnothing$

$D^{i+1}_{jr} = {}^{r}D^{i}_{ji}$

$C^{i+1}_{jr} = {}^{r}C^{i}_{ji}$; $C^{i}_{ji} = {}^{1}C^{i}_{ji} \cup {}^{r}C^{i}_{ji} \cup \dots \cup {}^{w}C^{i}_{ji}$

$\forall r \in \{1,\dots,w\} \mid {}^{r}C^{i}_{ji} \neq \varnothing$,

$\forall r,u \in \{1,\dots,w\}\ \forall c \in {}^{r}C^{i}_{ji} \mid r \neq u \Rightarrow \phi(c) \cap {}^{u}X^{i}_{ji} = \varnothing$,

where $\phi(c) = \{x_a,x_b,x_v\}$ for $c = P[x_a,x_b,x_v]$

ii) $CSP^{i+1}_{ji} = ((X^{i+1}_{ji},D^{i+1}_{ji}),C^{i+1}_{ji})$, where:

$X^{i+1}_{jr} = {}^{r}X^{i}_{ji}$; $X^{i}_{ji} = {}^{1}X^{i}_{ji} \cup {}^{r}X^{i}_{ji} \cup \dots \cup {}^{w}X^{i}_{j}$;

$\forall r \in \{1,\dots,w\} \mid {}^{r}X^{i}_{ji} \neq \varnothing$

$\forall r,u \in \{1,\dots,w\} \mid r \neq u \Rightarrow {}^{r}X^{i}_{ji} \cap {}^{u}X^{i}_{ji} = \varnothing$

$D^{i+1}_{jr} = {}^{r}D^{i}_{ji}$

$C^{i+1}_{jr} = {}^{r}C^{i}_{ji}$; $C^{i}_{ji} = {}^{1}C^{i}_{ji} \cup {}^{r}C^{i}_{ji} \cup \dots \cup {}^{w}C^{i}_{ji} \cup {}^{k,l}C^{i}_{j}$;

$\forall r \in \{1,\dots,w\} \mid {}^{r}C^{i}_{ji} \neq \varnothing$, ${}^{k,l}C^{i}_{j} \neq \varnothing$

$$\sum_{r=1}^{w} |{}^{r}C^{i}_{ji}| + |{}^{k,l}C^{i}_{j}| = |C^{i}_{ji}|$$

$\forall r,u \in \{1,\dots,w\}\ \forall c \in {}^{r}C^{i}_{ji} \mid r \neq u \Rightarrow \phi(c) \cap {}^{u}X^{i}_{ji} = \varnothing$,

where $\phi(c) = \{x_a,x_b,x_v\}$ for $c = P[x_a,x_b,x_v]$

$\forall c \in R(C^{i}_{ji})\ \exists k,l \in \{1,\dots,w\} \mid \phi(c) \cap {}^{k}X^{i}_{ji} \neq \varnothing\ \&\ \phi(c) \cap$

${}^{l}X^{i}_{ji} \neq \varnothing$.

For illustration lets us consider the reference model corresponding to the example shown on fig. 5. Given CSP=(({$x_1 \div x_{12}$},{$D_1 \div D_{12}$}),{$c_1 \div c_8$}).

The following notation is applied:

$CSP \sim (\{CSP^1{}_1, CSP^1{}_2\}, R)$,

$CSP^1{}_1 = ((\{x_1,x_2,...,x_6\}, \{D_1, D_2,..., D_6\}), \{c_1,c_2,c_3\})$

$CSP^1{}_2=((\{x_7,x_8,....,x_{12}\},\{D_7,D_8,...,D_{12}\}), c_4,c_6,c_7,c_8\})$

$R=\{^{1,2}C\};\ ^{1,2}C = \{c_5\}$

$CSP^1{}_2 \sim (\{CSP^2{}_{2,1}, CSP^2{}_{2,2}, CSP^2{}_{2,3}\}, R^1{}_2)$,

$CSP^2{}_{2,1} = (\{x_7,x_8\}, \{D_7, D_8\}),\{c_4\})$

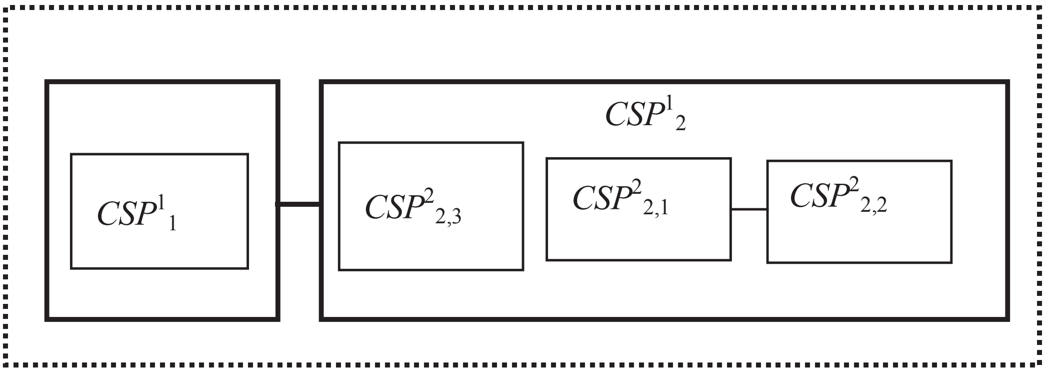
$CSP^2{}_{2,2} = (\{x_9,x_{10}\},\{D_9, D_{10}\}),\{c_6\})$

$CSP^2{}_{2,2} = (\{x_{11},x_{12}\},\{D_{11}, D_{12}\}),\{c_8\})$

$R^1{}_2=\{^{1,2}C^1{}_2\};\ ^{1,2}C^1{}_2=\{c_7\}$

The graphical, object-like representation of considered (see fig. 5) decomposition of the CSP problem is shown in fig. 6.

a)



b)

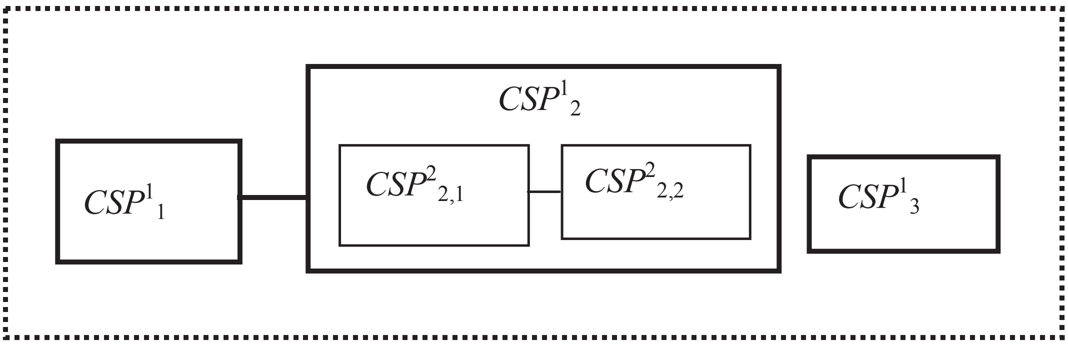


Fig.6. The graphic illustration of the CSP objective instance decomposition

Links between objects mean that subproblems should be solved jointly.

CSP decompositions instances presented in fig. 5 do not exhaust all potential decomposition possibilities.

Alternative decompositions of CSP problem presented in fig. 5 is presented as a graph (fig. 7).

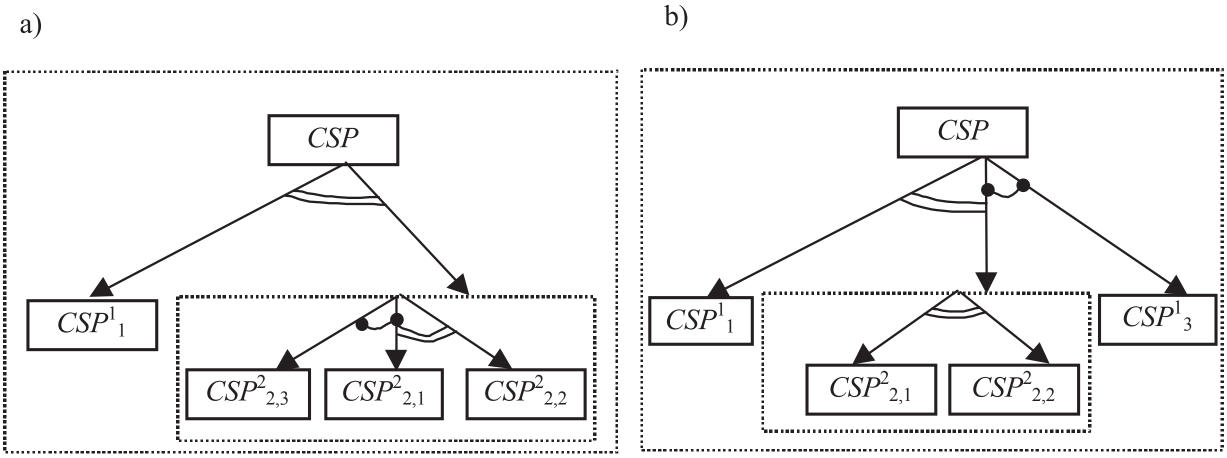
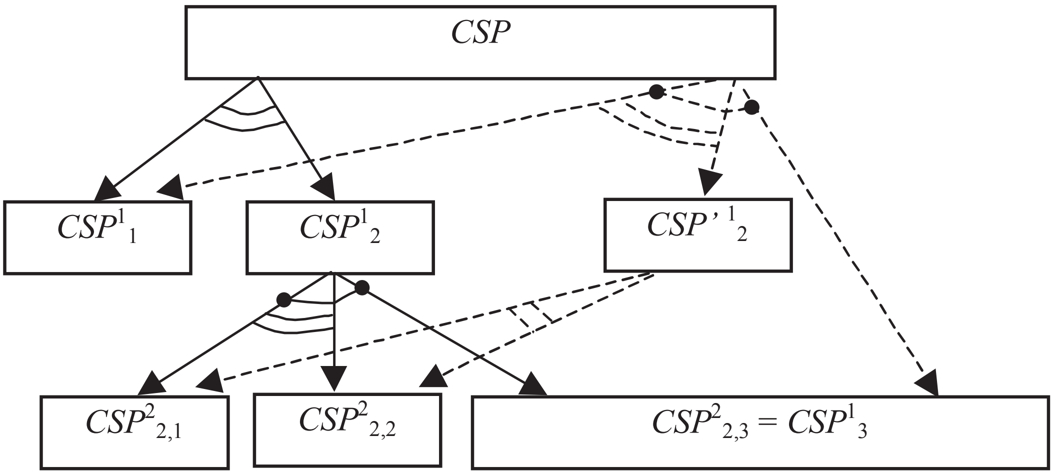


Fig.7. Alternative decompositions of CSP

Let us introduce decomposed subproblems notation: $CSP^{i}{}_{j',k,l}$ – represents l-th decomposition of the i-th problem (where i=|{j,k,l}|). This problem constitutes a k-th decomposition of an l-th problem which is a j-th decomposition of problem i-2-th which constitutes an i –2-th decomposition of the output CSP problem. According to his notation ‘j’- represents a decompositions problem which are respectively mutually independent (i.e. appropriate subsets of variables are not linked by any constraints).

Elementary individual decompositions subproblems are in practice identical. Graph AND/OR (fig. 8) is illustrated by various possibilities of CSP problem.



Legend:
- subproblems related by constraints,
- alternative subproblems,
- decompositions.

Fig.8. Graph AND/OR of CSP decomposition

AND/OR graphs present possible alternative decompositions of a CSP problem. The decompositions can be interpreted as searching strategies, determined by a specific number of subproblems, and the sequence of solving them.

In a situation when we have a given set of specifications Z = {z1, z2,..., zi}, their decompositions W = {w11, w12,..., wij} and solution strategies S = {s1,1,1, s1,1,2,..., sijl} where i –specification number, j – decomposition number, l – solution strategy number, we are searching for an answer to the question: Which solution searching strategy (and of which decomposition) is the best one (i.e. it allows for the fastest obtained decision)?

The presented instance of the CSP decomposition is the one of the decompositions. In order to estimate which decomposition, or corresponding searching strategy is the best one (e.g. from the time consumption point of view) a *number of potential backtrackings* is proposed as an evaluating criterion.

As an illustration let us consider two subproblems which can be solved in a free order. The strength of subproblem domain $A=((\{x_1\}, \{f_1,f_2,f_3,f_4, f_5\}), c_1)$ is $Z_A=5$, for subproblem $B=((\{x_2\}, \{p_1,p_2,p_3\}), c_2)$ it is $Z_B=3$. Object form of CSP decomposition is illustrated in fig. 9.

Possible searching strategies, within a given CSP decomposition is illustrated in fig. 10.

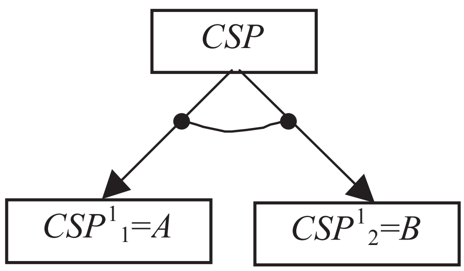


Fig.9. Object form of CSP decomposition

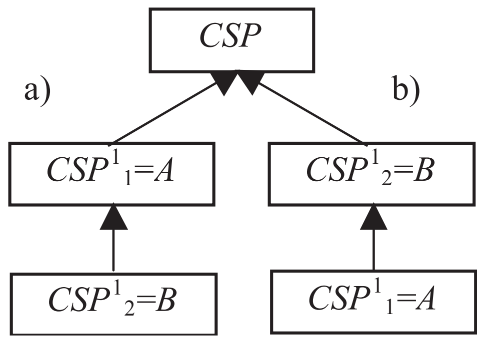


Fig.10. Possible strategies

Fig. 11 presents solution trees for two possible searching strategies. Fig. 11 a) presents a strategy where subproblem A is solved first and then B is solved. Fig. 11 b) presents a reverse order.

The number of potential backtrackings Nw is determined as follows:

$$\mathrm{Nw} = \sum_{i=1}^{LP} \left(\prod_{k=1}^{i} \mathrm{ZD}_{k,i} - 1 \right)$$

where: LP – a number of subproblems,

$ZD_{k,i}$ – a number of potential assignments of the i-th decision variable of the subproblem in the k-th sequence.

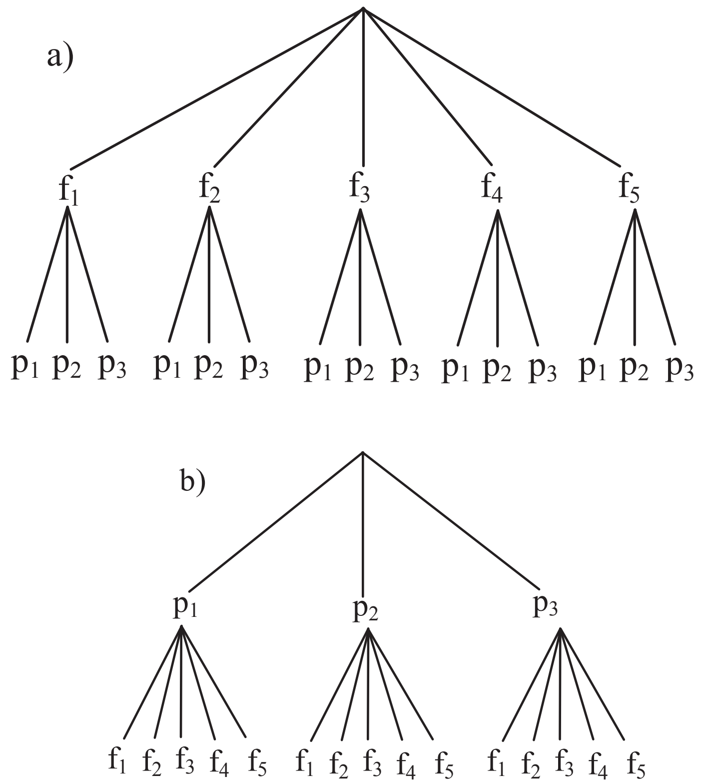


Fig.11. Solution trees

In case of fig. 11 a) the number of backtrackings is the following:
$Nw = (ZD_{1,1} - 1) + (ZD_{1,2} \cdot ZD_{2,2} - 1) = (5-1) + (5 \cdot 3-1) = 18$
In case of fig. 11 b) the number of backtrackings is the following:
$Nw = (ZD_{1,1} - 1) + (ZD_{1,2} \cdot ZD_{2,2} - 1) = (3-1) + (3 \cdot 5-1) = 16$

The searching strategy for a possible solution presented in fig. 11 b) is characterized by a lower number of potential backtrackings and in light of this criterion it is accounted as optimum strategies.

The AND/OR graph representation of possible CSP problem decompositions, facilitates an analysis of all potential (unconstrained with possibilities of applied CLP language programming systems) ways of solving a problem.

A reference model allows one to estimate a number of assignments of decision variables in particular searching strategy. So, it allows using a searching strategy requiring smallest number of backtrackings. The reference model facilitated a series of experiments which helped specifying (before implementation) what kind of searching leads (in a possibly short time) to obtaining a solution which would meet all constraints. The model helps evaluating specific feasible solutions (within different searching strategies) according to a chosen criterion.

Using the model and the possibility of initial evaluation of the searching strategy, an approach for finding possible solution was established. This approach has been implemented in the software package Production Order Verification System.

The application of CP techniques, for the small and medium size enterprises (SMEs) constitutes a possibility to build relatively fast and cheap decision support systems tailored to an enterprise needs, i.e., the task oriented decision supporting tools.

4. ILLUSTRATIVE EXAMPLE

The software package developed *Production Order Verification System* (POVS), aimed at the class of small and medium size enterprises enables to (fig. 12):

- specify a production and transportation system capabilities,
- specify the production order's requirements,
- store the production order specification into the data base,
- select searching strategy (to specify calculation parameters),
- monitor admissible solutions (i.e. results of production flow planning).

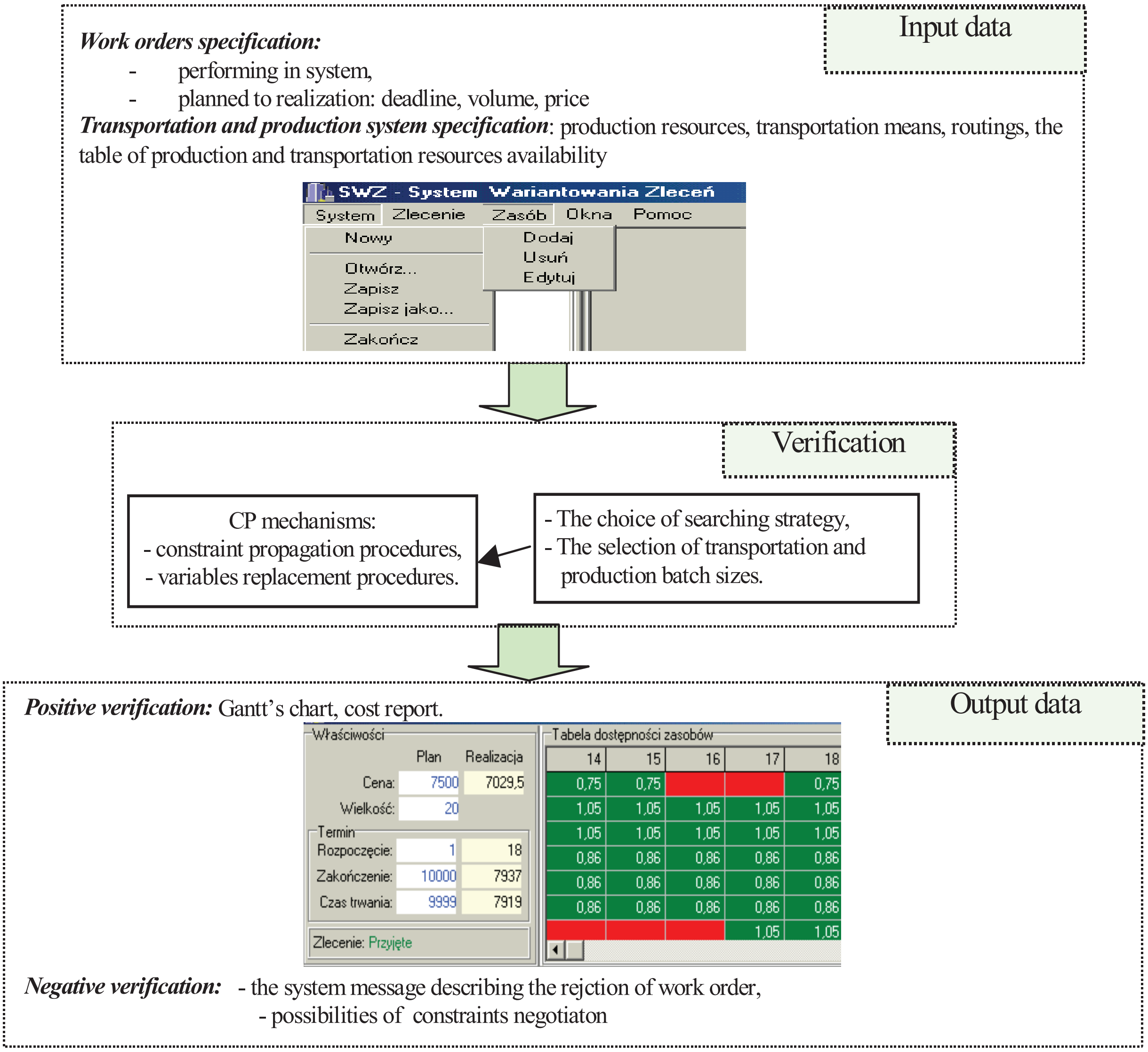


Fig.12. Data flow in POVS

The POVS has been applied at a SME producing the hydraulic and pneumatic equipment. In order to illustrate its application let us consider the following example regarding of three production orders (B_1, B_2, B_3) planned for the execution (tab. 1).

Tab.1. Specification of production work orders

Work orders	Name	Number of operations in a process	Production size (pcs.)	Suggested price (cost units)	Execution time (time units)
B_1	Filter set	10	100	1000	3500
B_2	Main body	27	120	1500	5500
B_3	Valve	7	50	1200	4200

After the introduction of data included in table 1 user/operator can introduce the production process of a given production order. The process can be chosen from the system database or can be defined by the user. For example, table 2 covers subsequent production operations and their execution times, which make the execution of the production order B_1. The filter set consists of two parts: filter and connector.

Tab.2. Specification of the operations and their duration time in work order B_1

Operation name	Production operation number	Execution time (time unit/pcs.)	tpz_{A_j}	Resources	Operation name	Number of the subsequent production operation	Execution time of an operation (time units/pcs.)	tpz_{A_j}	Resources
Operations in the execution of the filter					Operations in the execution of the connector				
Cutting	A_1	1	50	R_8	Cutting	A_4	1	20	R_{12}
Washing	A_2	1	120	R_2	Turning	A_5	1	90	R_{13}
Control	A_3	3	20	R_{19}	Washing	A_6	1	60	R_2
					Turning	A_7	2	110	R_{27}
					Hand treatment	A_8	1	50	R_7
					Washing	A_9	1	60	R_{31}
					Blacking	A_{10}	2	60	R_{17}

Legend: A_j – j-th operation, i = 1,...,10,
R_i – i-th production resource, i = 1,...,32,
tpz_{A_j} – preparation-finishing time.

Work order B_2 contains 27 production operations which are performed by using available resources. The work order B_3 specification illustrates table 3.

Tab.3. Specification of the operations and their duration time in work order B_3

Operation name	Production operation number	Execution time (time unit/pcs.)	tpz_{A_j}	Resources
Operations in the execution of the valve				
Cutting	A_1	2	50	R_{12}
Hand treatment	A_2	2	50	R_7
Turning	A_3	6	90	R_{13}
Grinding	A_4	2	110	R_{32}
Embossing	A_5	40	80	R_{11}
Washing	A_6	1	60	R_2
Heat treatment	A_7	3	40	R_4

After completion of each production operation a transportation operation to the next position of a given technological production route is executed.

The transportation means, their capacity, transportation routings and the duration times are defined.

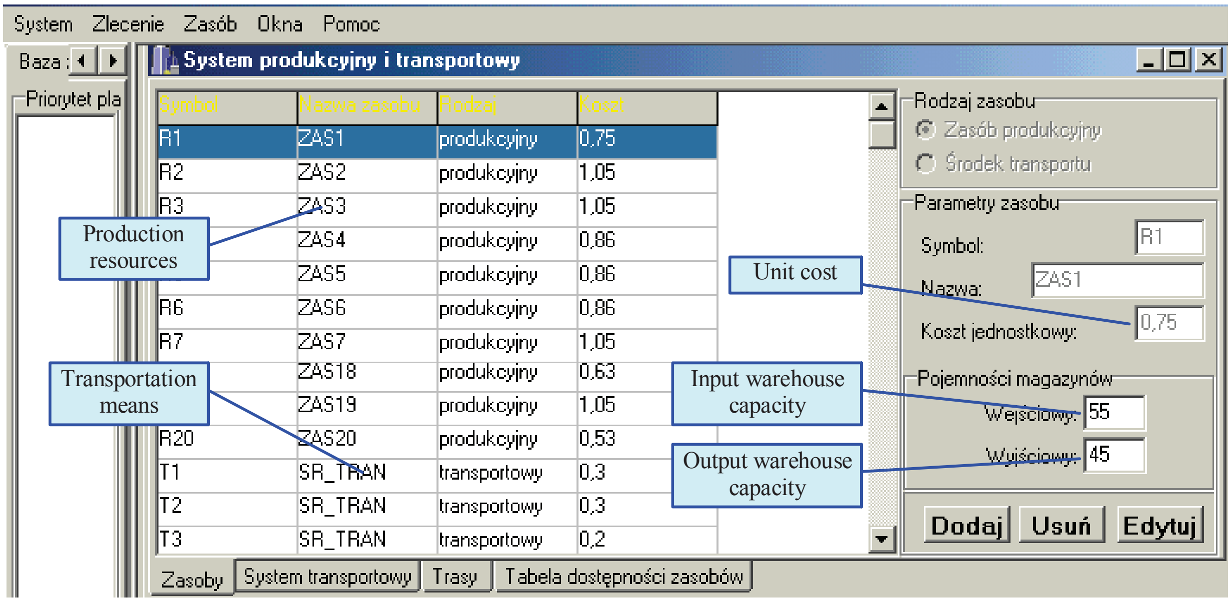


Fig.13. The transportations means

The following sequence of execution of the set production orders B_2 (ZlE_ARCH2_2), B_1 (ZLE_ARCH2_1), B_3 (ZLE_ARCH2_3) is considered (fig. 14).

Input data introduction facilitates commencement of the verification of single production order (group of production orders).

Due to the system's capability following from the currently realized production plan, all the introduced production orders cannot be taken for production. So, the production order B_1 (ZLE_ARCH2_1) cannot be processed, however, the remaining production orders can be taken for execution (fig. 14).

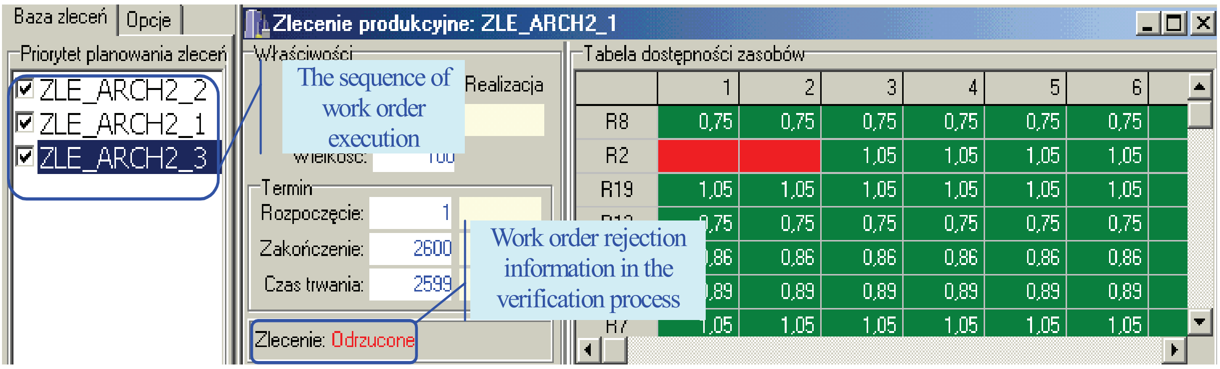


Fig.14. Work orders execution sequence

In case if a production order is rejected (i.e. a negative verification was obtained), POVS allows to change the priority of the planned production orders. It facilitates a next on-line verification, which leads to their "over planning". This gives another chance for a positive verification of the set of production orders introduced to the system. In other words, production orders that can use up the possibility of introduction of subsequent production orders (e.g. they engage too much resources or have a long operation time) are considered at the end.

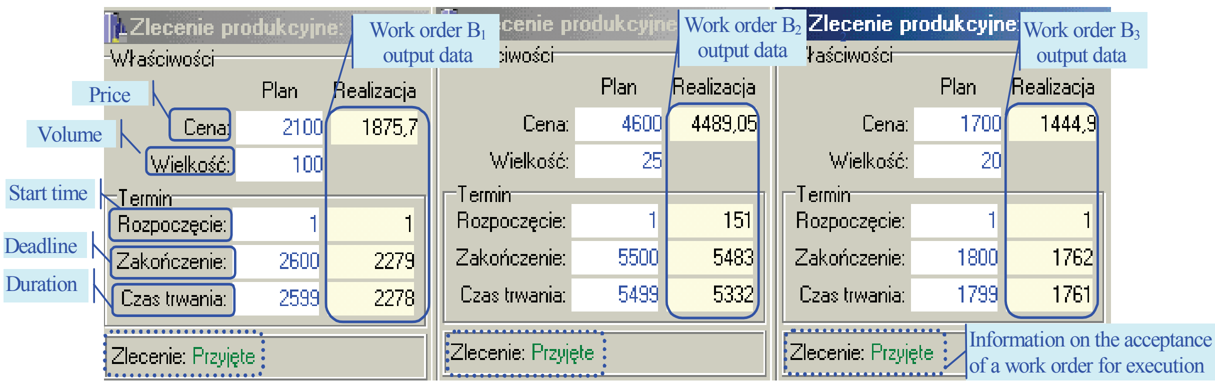


Fig.15. Verified work orders

In order to check a different opportunity to execute production orders, their priorities are changed as follows: B_1, B_2, B_3. The corresponding verification facilitates acceptance of all orders for their execution (fig. 14). It means, the production orders B_1, B_2, and B_3 may be finally taken for execution. The plan obtained provides the time of starting the production order B_1 at 1 time unit, the production order B_2 at 151 time unit, and B_3 at 1 time unit (fig. 16).

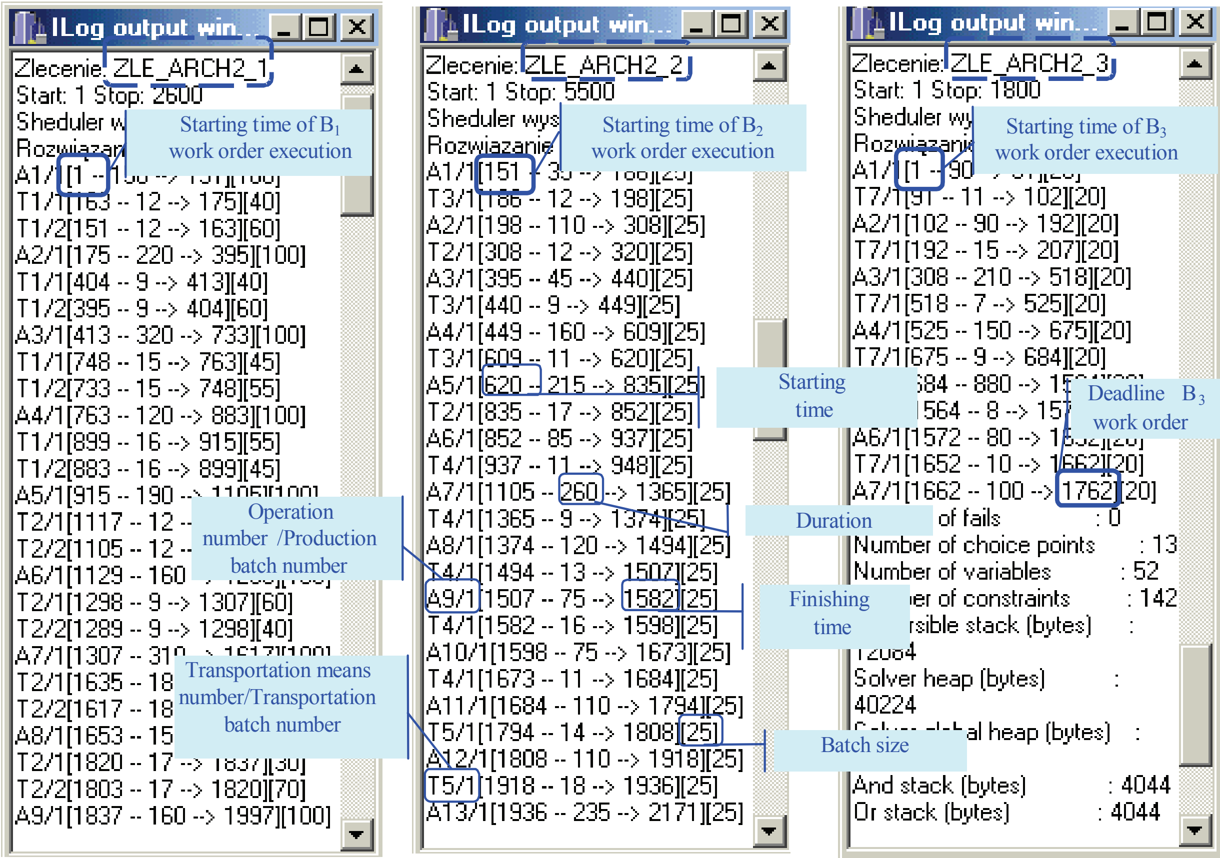


Fig.16. Obtained solution

The solution takes into account the possibility of execution of a production order due to the technological sequence of operations, transportation routings among resources and the production volume, capacity of buffers and their allocation. Moreover, knowing the resources availability the cost of the production order execution can be easily estimated as well.

Preparation and execution of the above experiment takes about 10 minutes, most of which is caused by the introduction of input data. In a case the production and transportation systems are already defined the introduction of a new input data consists in defining just the new production orders only. Then, the data introduction time takes about 5 minutes.

POVS software facilitates a fast evaluation of production orders assuming to be processed in a system already involved in the currently executed production plan. A newly introduced production orders are verified according to the company's possibilities and producer's constraints, i.e. the transportation-warehouse efficiency and the customer's requirements (directive deadline and the work order volume). It also facilitates establishing the sequence of production order execution by setting the priority of the verified production orders (tab. 4).

Tab.4. Comparison of the computer experiments result

EXPERIMENT executed in the POVS programme							
				Output data			Solution searching time in the programme POVS [sec.]
Assumptions		Work order verification sequence in the system	Solution	Programmed execution commencement deadline [time unit]	Programmed execution finishing time [time unit]	Work order execution cost [cost unit]	
Experiment a)	Free order execution sequence	B_2	Obtained	1	5101	4489,05	1,2
		B_1	No solution	-	exceeded	-	
		B_3	Obtained	1	1661	1444,9	
Experiment b)	Setting order priority	B_1	Obtained	1	2279	1875,7	1,4
		B_2	Obtained	151	5483	4489,05	
		B_3	Obtained	1	1762	1444,9	

Legend:

Experiment a) – work orders: B_2 and B_3 may be executed in the system, work order B_1 cannot be executed (exceeded directive execution time),

Experiment b) – work orders: B_1, B_2 and B_3 may be executed in the system,

Column „Work order verification sequence in the system" – means that the work orders with the highest priority are executed first.

The system assists the user in answering to the following questions: Are the company production capabilities are sufficient for the execution of a production order in accordance with the customer's requirements? What is the planned production order execution deadline? What is the cost of the order execution?

POVS facilitates setting a possible variant of current production organization. It includes routing, batching and scheduling and at the same time a timely and cost efficient execution of production orders.

5. CONCLUDING REMARKS

The need for planning of a broad assortment executed production, which is typical for SMEs, is not satisfied by systems which are currently available on the market. It gives rise to the increased demand for the decision support packages dedicated for these enterprises. Such tools should facilitate production flow planning (also in short planning horizons) in SME with a wide assortment production.

The system takes into consideration the financial abilities of the SME (purchase cost, software implementation cost) and the staff references. It is simple (e.g. easy data introduction into the system), therefore it does not require additional cost related with training or employment of a qualified staff. POVS (which is based on the CLP techniques) constitutes an attractive alternative for the commercial decision support online systems. It may be applied in SMEs which deal with the production of a broad variety of goods, increasing their competitiveness on the market.

References

[1] BANASZAK Z., JÓZEFOWSKA J. (red.): *Project-driven manufacturing*, WNT, Warszawa, 2003.

[2] BANASZAK Z., TOMCZUK I.: *Harmonogramowanie przedsięwzięć z wykorzystaniem technik programowania z ograniczeniami*, Red. Knosala R., Komputerowo Zintegrowane Zarządzanie, WNT, Warszawa, 2004, pp. 38-47.

[3] 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, pp. 7-14.

[4] BARTÁK R.: *Visopt shopfloor: a technology overview*, Proceedings of the Workshop on Constraint Programming for Decision and Control, June, Gliwice, Poland, 2002, pp. 7-15.

[5] Ilog Solver, *Object oriented constraint programming*, Ilog S.A., 12, Av. Raspail, BP 7, 94251 Gentilly cedex, France, 1995.

[6] ROSSI F.: *Constraint (Logic) programming: A Survey on Research and Applications*, K.R. Apt et al. (Eds.), New Trends in Constraints, LNAI 1865, Springer-Verlag, Berlin, 2000, pp. 40-74.

[7] TOMCZUK I., BANASZAK Z., JAKUBOWSKI J., BZDYRA K.: *Planning of goods transportation in distribution networks*, Logistics and urban infrastructure, Wrocław, 2004, pp. 150-157.

[8] TOMCZUK I., BANASZAK Z.: *Production flow planning based on CLP approach,* ed. Knosala R., Computer Integrated Management, WNT, Warszawa, 2005, pp. 589-600.

[9] TOMCZUK I., SANIUK S., BZDYRA K.: *Distribution model in recycling cars system*, Machinery Engineering, eds. Jędrzejewski J., Nowakowski T., FSNT NOT, Wrocław, 2003, pp. 88-98.

[10] TOMCZUK I., BANASZAK Z.: *Constraint programming approach for production flow planning*, Proceedings of the 6th Workshop on Constraint Programming for Decision and Control, 2004, pp. 47-54.

[11] TOMCZUK I., BANASZAK Z.: *Zastosowanie technik CP do planowania przedsięwzięć w warunkach ograniczeń zasobowych i logistycznych*, Zeszyty naukowe Wydziału Mechanicznego Politechniki Koszalińskiej nr 35, Polioptymalizajca i Komputerowe Wspomaganie Projektowania, pp. 246- 253.

[12] TOMCZUK I., BANASZAK Z.: *Zastosowanie technik programowania z ograniczeniami do planowania przepływu produkcji w przedsiębiorstwie produkcyjnym*, VII KK Project Management „Pokonać kryzysy w projekcie", Wrocław, 2003, pp.144-160.

[13] TOMCZUK I., BOROWIECKI T., BANASZAK Z.: *Project-driven decision support: a CLP approach.*, Automatyzacja-Nowości i Perspektywy, Automation 2004, pp. 333-343.

[14] TOMCZUK I., BZDYRA K.: *Reference model of CSP decomposition*. Materiały konferencyjne, Automatyzacja-Nowości i Perspektywy, Automation 2005, pp. 225-232.
[15] VAN HENTENRYCK P., PERRON L., PUGET J.: *Search and Strategies in OPL*, ACM Transactions on Computational Logic, Vol. 1, No. 2, 2000, pp. 1-36.
[16] VAN HENTENRYCK P.: *Constraint Logic Programming*, Knowledge Engineering Review, Vol. 6, 1991, pp. 151–194.
[17] WALLACE M.: *Constraint Logic Programming*, Ed. Kakas A.C., Sadri F., Computat. Logic, LNAI 2407, Springer-Verlag, Berlin, Heidelberg 2000, pp. 512-532.