Constraint logic programming, decision support, flow planning, constraint satisfaction problem

Krzysztof BZDYRA*, Izabela TOMCZUK-PIRÓG **

CP-BASED DECISION SUPPORT PACKAGE FOR PRODUCTION FLOW PLANNING - AN SME PERSPECTIVE

Abstract:

Solving complex, decision-making problems requires application of decision support systems. The best solving strategy can be defined for each particular problem. The strategy refers to a sequence solving elementary subproblems. The aim of the paper is to present the evaluating criterion to estimate the efficiency of searching strategies. The criterion proposed offers a possibility to estimate the time needed to solve a constraint satisfaction problem. An illustrative example is provided.

1. INTRODUCTION

Small and medium size enterprises (SMEs) constitute a growing group within the companies in the market, both as far as their number and the production value are concerned. An intensive growth of this market in the recent years results from several advantages they have in comparison with large companies. The characteristic feature of the SMEs is that their organisation schedule and the goods they offer are very diversified. This is the effect of searching for a structure which would give a possibility of functioning in growing market competition conditions [1, 4].

Such situation leads to a growing demand for SMEs dedicated decision support systems. There are, however several factors which hinder development of such systems. These are first of all constrained SMEs financial resources which cause that a purchase of ready systems is impossible. Apart from that, different individual features of these companies impose a necessity to offer individually diversified decision support systems. Another difficulty is the fact that a highly qualified personnel needs to be employed to operate the IT (Information Technology) system. Worth noting is also the fact that the offer addressed to SMEs is usually a simplified version of software applied in large companies. Therefore, the algorithms applied in the systems do not take under consideration the specific structure, organization and functioning of SMEs.

Solutions applied in these packages do not meet the above mentioned requirements. Methods applied either disregard the SMEs specific features (e.g. mass service method), or are

* Krzysztof Bzdyra, Ph.D., Department of Electronics and Informatics, Technical University of Koszalin, 75-453 Koszalin, Poland, E-mail: krzysztof. bzdyra@tu.koszalin.pl

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

above their financial and labour resources e.g. related with the purchase and use of professional packages such as: LINGO (implementing the methods of operational research methods), or TAYLOR (implementing a computer simulation methods). Artificial intelligence methods (evolution algorithms, simulated burning etc.), offer a chance to meet the requirements, especially in case of constrained logic programming (CLP). CHIP, ILOG, Mozart packages are examples of their application.

The paper concentrates on a chosen decision problems class related with production flow planning in the SMEs production, with a special stress on verifying new production orders. Verifying production orders offers a possibility to evaluate, whether resource capacity of a manufacturer is balanced with the production orderer's requirements [6, 7]. As the searching for a feasible solution requires the same calculation resources as a searching for an optimum solution (both categories belong to an NP – difficult group) it was assumed, that the work of the system should deal with searching for the first possible solution, i.e. a solution meeting the constraints assumed.

SMEs are characterized with a diversified organization and functioning as well as with a variety of goods offered. This is the reason for individual, dedicated approach to the decision support system structure. Experience obtained in the construction and exploitation process of a production order variation system contributes to the general methodology in constraint logic programming. This methodology offers a possibility to establish applications which take into consideration both constraints resulting from company characteristic features and constraints resulting from the character of the problems solved or the CLP tools. It constitutes a framework of actions which, depending on requirements, is extended with items characteristic for a currently solved decision problem. The framework presented in the paper has been limited to production flow planning problems in SMEs [1].

2. PROBLEM STATEMENT

Consider a manufacturing system with a given production capacity while processing some other work orders. Therefore, only a part of the production capacity (determined by 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. Every activity cannot 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.

3. CONSTRAINT SATISFACTION PROBLEM

Production flow planning problems can be formulated as the Constraint Satisfaction Problems (CSPs), for which many Constraint Programming (CP) languages were worked out.

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 $X = \{x_1, x_2, \dots, x_n\}$, their domains $D = \{D_i \mid D_i = [d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{im}], i = 1..n\}$, 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 $P[x_k,x_l,\dots,x_h]$ 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 $X = \{x_1,x_2,\dots,x_{12}\}$, $D = \{D_1,D_2,\dots,D_{12}\}$, $C = \{c_1,c_2,\dots,c_8\}$, 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. 1.

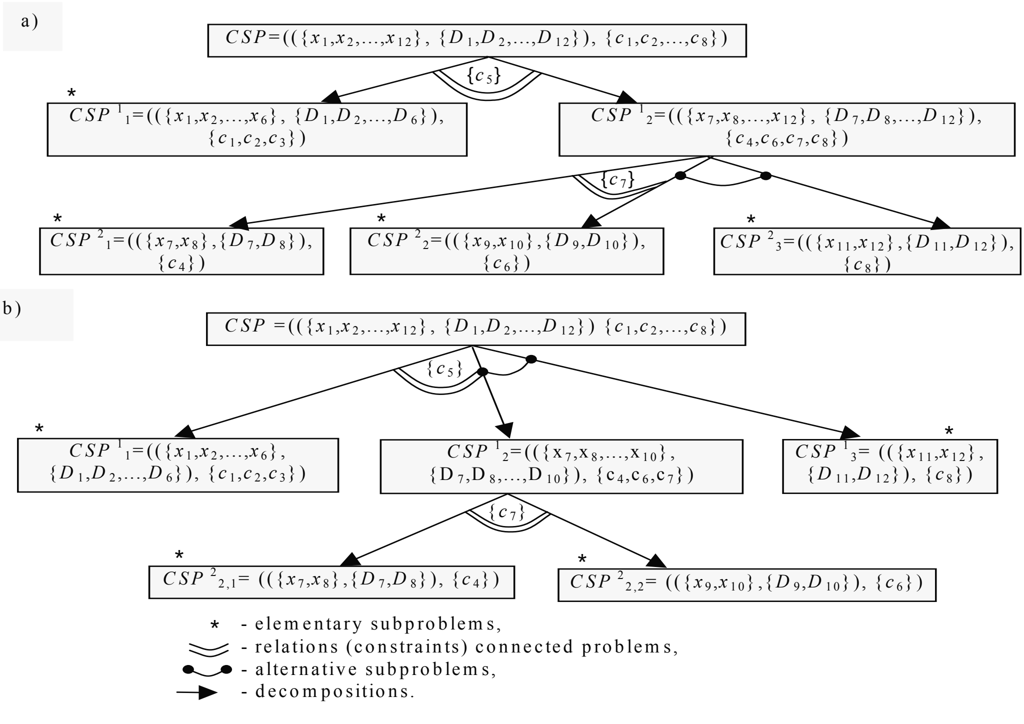


Fig.1. The CSP feasible decompositions

The subproblems that cannot be decomposed are side to be so called the elementary problems.

The presented example illustrates the possibility of choosing of the searching strategy that minimizes the number of potential backtrackings [7].

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 language provides (if possible) solution to a given constraint satisfaction problem? – What searching strategy minimizes the number of potential backtrackings?

4. REFERENCE MODEL

The problem representation and the potential of the available CP 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.

A problem is elementary if for every pair of its decision variables there is a constraint which links the variables directly or indirectly in a sequence of constraints.

Decomposition of $((X,D),C)$ into elementary subproblems $((X_i,D_i),C_i)$, $((X_j,D_j),C_j)$ assumes that:

$$\forall X_i, X_j \subset X \mid X_i \cap X_j = \varnothing$$

For every elementary $((X_i,D_i),C_i)$ problem it is assumed that all variables from X_i set are related directly or indirectly.

$\forall x_a, x_b \in X_i\ \exists \pi(\varsigma), \varsigma \subseteq C_i | \forall k=1..||\varsigma||-1,$

$G(crd_k\pi(\varsigma)) \cap G(crd_{k+1}\pi(\varsigma)) \wedge x_a \in G(crd_1\pi(\varsigma)) \wedge x_b \in G(crd_{||\varsigma||-1}\pi(\varsigma))$

where:

$\pi(\varsigma)$ – the permutation of subset $\varsigma \subseteq C_i$

$crd_k\pi(\varsigma)$ – the value of $\pi(\varsigma)$ the k-th entry

$G(crd_k\pi(\varsigma)) \subseteq X_i$ - the subset of constraint arguments on the k-th entry of constraints permutation.

Elementary problems $((X_i,D_i),C_i)$, $((X_j,D_j),C_j)$ are mutually dependent if the following condition is met:

$$\exists\ c_k \in C \setminus C_j \cup C_i \mid D(c_k) \cap (\cup\{x_h | x_h \in X_i\}) \cap (\cup\{x_h | x_h \in X_j\}) \neq \varnothing$$

Dependent problems have been marked with an arch in fig. 4. It was also given what constraint links the analysed elementary problems.

In order to simplify further analysis, the following notation of the decomposition problems was proposed:

CSP – constraint satisfaction problem

$CSP^{i}_{a(1),\ a(2),\ldots,\ a(i)}$ – the a(i)-th decomposition of $CSP^{i-1}_{a(1),\ a(2),\ldots,\ a(i-1)}$ problem where :

a(1)...a(i) – marking of the subsequent problem decompositions,

i – decomposition depth.

Links between objects mean that subproblems should be solved jointly.

The presented instance of the CSP decomposition is 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.

The number of potential backtrackings (Nw) is determined as follows [6]:

$$Nw = \sum_{i=1}^{LP} \left(\prod_{k=1}^{i} ZD_{k,i} - 1 \right) \tag{1}$$

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.

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 $Z_B=3$.

Fig. 6 presents solution trees for two possible searching strategies. Fig. 2 a) presents a strategy where subproblem A is solved first and then B is solved. Fig. 2 b) presents a reverse order.

In case of fig. 6 a) the number of backtrackings is the following:

$Nw = (ZD_{1,1} -1) + (ZD_{1,2} \cdot ZD_{2,2} -1) = (5-1) + (5\cdot3-1) = 18$

In case of fig. 6 b) the number of backtrackings is the following:

$Nw = (ZD_{1,1} -1) + (ZD_{1,2} \cdot ZD_{2,2} -1) = (3-1) + (3\cdot5-1) = 16$

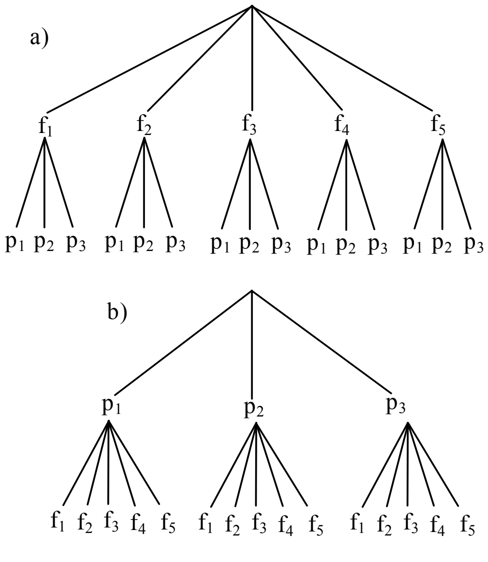


Fig.2. Solution trees

The searching strategy for a possible solution presented in fig. 2 b) is characterized by a smaller number of potential backtrackings and in light of this criterion the strategy is as an optimal one.

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.

5. PRODUCTION ORDER VERIFICATION SYSTEM

The established decision support system is dedicated to small and medium size enterprises. It offers a possibility to verify the incoming production orders quickly. The verification is based on finding a production flow programme including transportation, treatment, assembly and quality control operations meeting constraints imposed by the production system structure, technology and constraints related with the current production programme. The system idea has been presented in fig. 3. The system work is based on balancing the manufacturer's resource capacity with the requirements resulting from new production orders. If the verification is possible the user disposes of a ready production programme including choice of routes in cases when considering of technological routes, division into production and transportation batches, routing and scheduling of tasks is possible.

Due to financial and labour constraints resulting from the SMEs specifics, the operating of the system does not a require highly trained staff. A system structure consisting of two parts was assumed. First of them constitutes a calculation module making use of the CLP techniques, the second one is a user interface which gives a possibility of easy and quick system data input. The system has been implemented in Oz and Borland Delphi languages.

It is possible to distinguish three layers in the decision support system structure. The layers differ in tasks which they have been implemented for and in a possibility to interfere in their work by the user. In the work on the programme a lot of attention was given to the data exchange between these layers. Due to the necessity to assure an easy use of the programme the user does not have to prepare calculation data in a form which is legible for the CLP language applied. In practice, every production order verification would involve a necessity of the programmer's interference in the programme structure.

The verification procedure goes as follows:

- defining the production and transportation systems (if necessary),
- defining processes and updating of products data base (if necessary),
- defining of production orders and starting calculations,
- analysis of results

It is easy to notice, that some of the mentioned activities appear only in some cases e.g. changes in the production system impose changes in both – the part dealing with the system definition and the part dealing with the process definition. Information flow has been illustrated in fig. 3.

Legend:

A – system and production order process definition,

B – data preparation in a form acceptable for the CLP language applied,

D – input data in the calculation module,

E – generating the results file,

F – results presentation.

Arrows show the information flow direction.

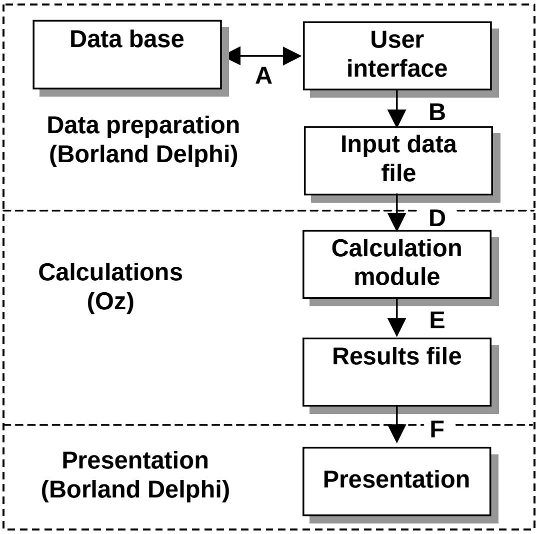


Fig.3. Information flow diagram in the production order verification system structure

The problem of balancing the company resources with the customer's requirements is decomposed into elementary problems in Production Order Verification System (POVS). According to the CSP problem reference model, every elementary problem is also a CSP problem. A solution strategy was assumed which is characterized with the smallest number of possible substitutions. Fig. 4. illustrates the sequence of solving elementary problems.

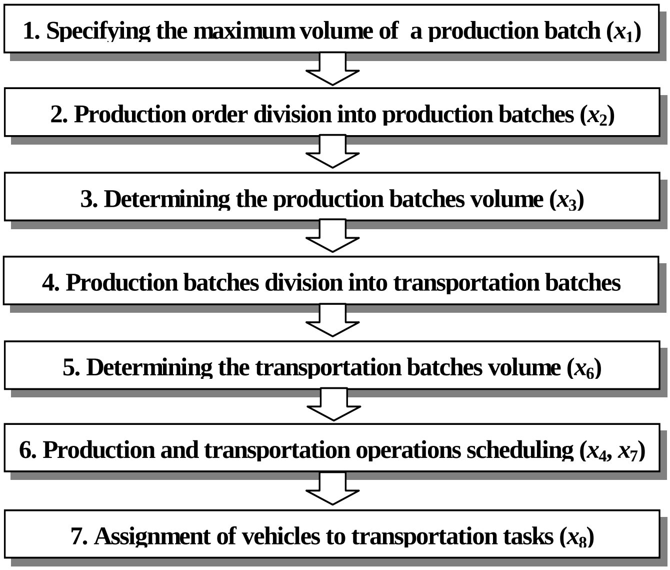


Fig.4. Solution searching algorithm

System verification was carried out on the basis of data obtained from Archimedes S.A. An order for the production of a pneumatic engine was verified. The diagram of the production process is included in fig. 5.

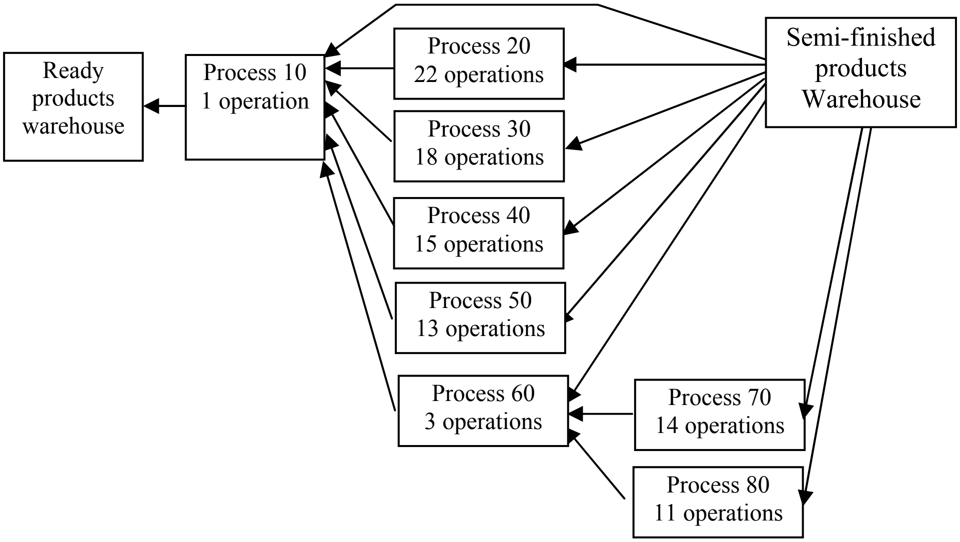


Fig.5. Production process structure

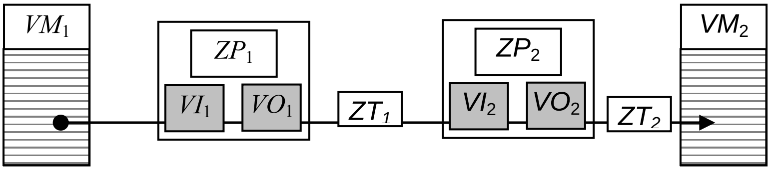
In the experiments the response time for an enquiry in an uncharged system was examined – the problem was to plan the production flow in a deadlock-free manner. En experiment which was to verify the production orders incoming to a charged system was carried out. Sample results have been listed in table 1.

Tab.1. POVS verification results in Archimedes S. A.

Batches number	System charge	Variables number	Substitutions number	Calculations time
1	No	1136	295	7.81
1	Yes	1136	186	29.21
2	No	2110	8643	282.12
2	Yes	2110	8217	187.34

What needs to be noticed is the fact that the calculation time in experiments in which the production order was divided into three production batches has been extended in a significant way. In case of system charge the number of active constraints which exclude the operations execution in forbidden times increases. As a result the calculation time increases, as the additional constraints influence directly the length of OZ language notation. On the other hand, as a result of propagation the constraints narrow down the decision variables domains (in the process of scheduling), which results in a smaller amount of substitutions.

Apart from actual data verification, comparative tests of the system with the LINGO language application were carried out. LINGO has been chosen due to the CSP problem specification possibility. Non linear constraints specification is possible in this programme as well. The comparison referred to a production system presented in fig.6.



Legend:

VM_1, VM_2 – warehouses; ZP_1, ZP_2 – production resources;
VI_1, VI_2 – input buffers; VO_1, VO_2 – output buffers;
ZT_1, ZT_2 – transportation vehicle; ●⟶ details flow direction

Fig. 6. Production system in comparative experiments

Table 2 includes the comparative experiments results. Calculation time, production batches number and the volume of the production order have been collected.

Tab. 2. Comparative experiments results listing

Experiment	Production order volume	Calculations time [s]		Production batches number
		LINGO	POVS	
1	1	13	0.141	1
2	2	13	0.140	1
3	5	12	0.140	1
4	10	12*	0.281	2

* LINGO responded: "no possible solution"

In the cases presented LINGO obtained worse searching times. Worth emphasizing is the fact that in the fourth case (marked with *) LINGO did not find a possible solution although such solution existed and has been given by the POVS.

In a case when a possible solution exists, POVS can always find it. In some comparative experiments LINGO has generated a message about a lack of possible solution although such solutions existed. It results from the fact that in methods based on mathematic programming an optimum solution is usually searched, which requires giving initial possible solution. In situations, when a task is formulated as possible solution searching task (or in other words searching for initial solution in optimization tasks), algorithms based on mathematic programming methods do not work.

What is important is the size of problems which can be solved by means of POVS. The computer calculation capacity is the basic constraint. Average class equipment (Processor ca. 2 GHz, RAM – 256 MB), can solve tasks with up to 200 production operations with planning horizons of about 1 000 000 agreed time units. The number of operations is in this case refers to all production batches. In a trial of increasing the number of operations it is advised to recalculate (rescale) the preparatory, finishing and unit times so that the number of potential substitutions for all decision variables can be reduced.

6. CONCLUSIONS

The approach presented allows estimating probability of finding quickly the first acceptable solution. The presented criterion (specifying the number of potential backtrackings) determines relations between decision variables. These relations give a possibility to estimate the number of potential substitutions more accurately.

Taking into account the criteria of connections between variables allows differentiating strategies more efficiently. The results of theoretical considerations were confirmed by computer experiments. It is therefore possible to choose the most effective problem solving strategy before the commencement of calculations.

The methodology presented can be applied to design task oriented decision support systems. Such systems "choose" the best strategy depending on the parameters of the problem to be solved.

References

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

[2] 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.

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

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

[5] BZDYRA K., BACH I., MĄDRY M.: *Strategia przeszukiwania dedykowana dla problemu spełniania ograniczeń*. Automation 2006, X Konferencja N-T, Warszawa 2006, pp 289-297.

[6] 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.

[7] TOMCZUK I, BZDYRAK.: *Reference model of CSP decomposition*, *Automation* 2005, pp. 225-232.

[8] P. VAN HENTENRYCK: *Constraint Logic Programming*, *Knowledge Engineering Review*, Vol. 6, 1991, pp. 151–194.

[9] WALLACE M.: *Constraint Logic Programming*, Ed. Kakas A.C., Sadri F., *Computat. Logic*, LNAI 2407, Springer-Verlag, Berlin, Heidelberg 2000, pp. 512-532.