Production Flow Planning Based On CLP Approach

Izabela Tomczuk
Technical University of Opole, Department of Management and Production Engineering,
45-370 Opole, Poland
itomczuk@po.opole.pl

Wojciech Muszyński
Technical University of Wrocław, Institute of Technical Cybernetics,
50-371 Wrocław, Poland
wojciech.muszynski@pwr.wroc.pl

Zbigniew Banaszak
Technical University of Koszalin, Department of Electronics and Informatics,
75-453 Koszalin, Poland
banaszak@tu.koszalin.pl

Abstract

The paper provides a presentation of a method of balancing the work order requirements with given production capabilities in a planning horizon.

Constraint logic programming (CLP) is an emergent software technology for declarative description and effective solving of combinatorial problems especially in the area of integrated production planning. In that context, the CLP can be considered as a well-suited framework for development of decision-making software supporting small and medium size enterprises in the course of production process planning.

The aim of this paper is to present the CLP modeling framework as well as to illustrate its application to decision making in the case of a new production order prototyping.

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, 7, 9]. 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 that specify certain relations between decision variables. Because of their declarative character they can be implemented in decision support systems [5, 7].

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 [5, 9], e.g. at the planning of goods transportation in distribution networks [6, 8].

The most important issues that contribute to the efficiency of CLP techniques are the procedures of a constraint sets, sets of discrete decision variables), CLP techniques facilitate the application of constraint propagation procedures (e.g., at the planning of goods transportation in distribution networks [6, 8].

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: \( x_1 \in \{1,...,7\} \), \( x_2 \in \{2,...,4\} \), \( x_3 \in \{3,...,11\} \) and a constraint set: \( x_1 \leq x_2 - 1 \), \( x_3 = 2x_1 + x_2 \), \( x_1 + x_3 > 6 \). We are looking for the variable values, which meet the constraints.

As a result of the first constraint propagation the following solution was obtained: \( x_1 = 1 \), \( x_2 = 2 \), \( x_3 = 5 \). 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.

![Figure. 1. Searching process](image)

Searching process of the potential solutions space (implementing the CLP techniques) can be executed with the application of methods based on a searching strategy „deep-first”, methods of „iterative broadening IB” or the methods of „Limited Assignment Number Search” (LAN) [2]. 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 [2, 11, 12]. 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) [4,7,9], 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 \( x_1, x_2, x_3 \), 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.

From the available works [2, 3] 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 $D = \{D_i | D_i = [d_{i1}, d_{i2}, \ldots, d_{im}], i = 1..n\}$, and a set of constraints $C = \{C_i | i = 1..L\}$.

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 $P[x_1, x_2, \ldots, x_n]$ 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 $CSP = ((X,D), C)$, 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: $c_1 = P_{1}[x_1, x_2, x_3]$, $c_2 = P_{2}[x_2, x_4, x_5]$, $c_3 = P_{3}[x_4, x_6]$, $c_4 = P_{4}[x_7, x_8]$, $c_5 = P_{5}[x_4, x_7]$, $c_6 = P_{6}[x_6, x_{10}]$, $c_7 = P_{7}[x_7, x_9]$, and $c_8 = P_{8}[x_{11}, x_{12}]$. Two arbitrary chosen feasible decompositions of the CSP considered are shown in fig. 2. The subproblems that cannot be decomposed are side to be so called the elementary problems.

```
CSP=((\{x_1, x_2, \ldots, x_{12}\}, \{D_1, D_2, \ldots, D_{12}\}), \{c_1, c_2, \ldots, c_8\})
```

![Figure 2. The CSP feasible decompositions](image-url)
language provides (if possible) solution to a given constraint satisfaction problem? - What searching strategy minimizes the number of potential backtracks?

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:

\[ \text{CSP} = ((X, D), C) \]

- the specification of the CSP decomposition,
- the specification of constraints satisfaction problem.
- The reference model of decomposition of the CSP = ((X,D),C) – the graph-like representation of the CSP decomposition, where: CSP\(^{(i)}\) = (\(X^{(i)},D^{(i)},C^{(i)}\)) – the direct decomposition of the CSP\(^{(i+1)}\) problem, which decomposition, or corresponding searching strategy is the best one (e.g. from the time consumption point of view) a number of potential backtracks is determined as follows:

\[ \forall c \in \text{CSP}_{ji} \quad |c| \cap X_{ji} = \emptyset \quad \& \quad |c| \cap X_{ji+1} = \emptyset \]

\[ \sum_{r=1}^{w} r \cdot C_{jr} = |C_{jr}|, \quad \forall r \in \{1,...,w\} \quad |C_{jr}| \neq \emptyset \]

\[ \sum_{r=1}^{w} r \cdot C_{jr} = |C_{jr}|, \quad \forall r \in \{1,...,w\} \quad |C_{jr}| = |C_{jr}| \]

For illustration let us consider the reference model corresponding to the example shown on fig. 2. Given CSP\(=\{(x_1,x_2)\}, D\{D_1,D_2\}, \{c_1,c_2\}\). The following notation is applied:

\[ \text{CSP} \sim \{(\text{CSP}_1, \text{CSP}_2), R_1\}, \text{CSP}_1 = \{(x_1,x_2,x_3), \{D_1, D_2, D_3\}, \{c_1,c_2,c_3\}\} \]

\[ \text{CSP}_2 = \{(x_4,x_5,x_6), \{D_4,D_5,D_6\}, \{c_4,c_5,c_6\}\} \]

\[ R = \{1,2\} \]

\[ \text{CSP}_2 \sim \{(\text{CSP}_2, \text{CSP}_2, \text{CSP}_2), R_2\}, \text{CSP}_2 = \{(x_7,x_8), \{D_7, D_8\}, \{c_7\}\} \]

\[ \text{CSP}_2 = \{(x_9,x_10), \{D_9,D_{10}\}, \{c_9\}\} \]

\[ R = \{1,2\} \]

The graphical, object-like representation of considered (see fig. 2 a) decomposition of the CSP problem is shown in fig. 3.

**Figure 3. The graphic illustration of the CSP objective instance decomposition**

Links between objects mean that subproblems should be solved jointly.

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 backtracks is proposed as an evaluating criterion.

The number of potential backtracks \(Nw\) is determined as follows:
\[ N_w = \sum_{i=1}^{LP} (\prod_{k=1}^{i} ZD_{k,j} - 1) \]

where: \( LP \) – a number of subproblems, 
\( ZD_{k,j} \) – 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 \( A=((\{x_1\}, \{f_1, f_2, f_3\}, c_1) \) is \( Z_1=5 \), for subproblem \( B=((\{x_2\}, \{p_1, p_2, p_3\}, c_2) \) it is \( Z_2=3 \).

Fig. 4 presents solution trees for two possible search strategies. Fig. 4 a) presents a strategy where subproblem A is solved first and then B is solved. Fig. 4 b) presents a reverse order.

**Figure. 4. Solution trees**

In case of fig. 4 a) the number of backtrackings is the following:

\[ N_w = (ZD_{1,1} \cdot -1) + (ZD_{1,2} \cdot ZD_{2,2} \cdot -1) = (5-1) + (5 \cdot 3 \cdot 1) = 18 \]

In case of fig. 4.12 b) the number of backtrackings is the following:

\[ N_w = (ZD_{1,1} \cdot -1) + (ZD_{1,2} \cdot ZD_{2,2} \cdot -1) = (3-1) + (3 \cdot 5 \cdot 1) = 16 \]

The search strategy for a possible solution presented in fig. 4 b) is characterized by a lower number of potential backtrackings and in light of this criterion it is accounted as optimum strategies.

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 application of CLP 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:
- 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).

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 (B₁, B₂, B₃) planned for the execution (tab. 1).

<table>
<thead>
<tr>
<th>Work orders</th>
<th>Name</th>
<th>Number of operations in a process</th>
<th>Production size (pcs.)</th>
<th>Suggested price (cost units)</th>
<th>Execution time (time units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B₁</td>
<td>Filter set</td>
<td>10</td>
<td>100</td>
<td>1000</td>
<td>3500</td>
</tr>
<tr>
<td>B₂</td>
<td>Main body</td>
<td>27</td>
<td>120</td>
<td>1500</td>
<td>5500</td>
</tr>
<tr>
<td>B₃</td>
<td>Valve</td>
<td>7</td>
<td>50</td>
<td>1200</td>
<td>4200</td>
</tr>
</tbody>
</table>

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 B₁. The filter set consists of two parts: filter and connector.

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. The following sequence of execution of the set production orders B₂, B₁, B₃ is considered: 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 B₁ cannot be processed, however, the remaining production orders can be taken for execution.
Tab. 2. Specification of the operations and their duration time in work order B

<table>
<thead>
<tr>
<th>Operation name</th>
<th>Production operation number</th>
<th>Execution time (time unit/pcs.)</th>
<th>Resources</th>
<th>Operation name</th>
<th>Production operation number</th>
<th>Execution time (time unit/pcs.)</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations in the execution of the filter</td>
<td></td>
<td></td>
<td></td>
<td>Operations in the execution of the connector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cutting</td>
<td>A₁</td>
<td>1</td>
<td>50</td>
<td>R₈</td>
<td>Cutting</td>
<td>A₄</td>
<td>1</td>
</tr>
<tr>
<td>Washing</td>
<td>A₂</td>
<td>1</td>
<td>120</td>
<td>R₂</td>
<td>Turning</td>
<td>A₃</td>
<td>1</td>
</tr>
<tr>
<td>Control</td>
<td>A₃</td>
<td>3</td>
<td>20</td>
<td>R₁₀</td>
<td>Washing</td>
<td>A₀</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Turning</td>
<td>A₇</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hand treatment</td>
<td>A₈</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Washing</td>
<td>A₀</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Blacking</td>
<td>A₁₀</td>
<td>2</td>
</tr>
</tbody>
</table>

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

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.

Figure 5. Verified work orders

Figure 6. Obtained solution
In order to check a different opportunity to execute production orders, their priorities are changed as follows: B₁, B₂, B₃. The corresponding verification facilitates acceptance of all orders for their execution (fig. 5). It means, the production orders B₁, B₂, and B₃ may be finally taken for execution. The plan obtained provides the time of starting the production order B₁ at 1 time unit, the production order B₂ at 151 time unit, and B₃ at 1 time unit (fig. 6).

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. 3).

Tab. 3. Comparison of the computer experiments result

<table>
<thead>
<tr>
<th>EXPERIMENT executed in the POVS programme</th>
<th>Assumptions</th>
<th>Work order verification sequence in the system</th>
<th>Solution</th>
<th>Programmed execution commencement deadline [time unit]</th>
<th>Programmed execution finishing time [time unit]</th>
<th>Work order execution cost [cost unit]</th>
<th>Solution searching time in the programme POVS [sec.]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Experiment a)</td>
<td>Free order execution sequence</td>
<td>B₂</td>
<td>Obtained</td>
<td>1</td>
<td>5101</td>
<td>4489,05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B₁</td>
<td>No solution</td>
<td>-</td>
<td>exceeded</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B₃</td>
<td>Obtained</td>
<td>1</td>
<td>1661</td>
<td>1444,9</td>
</tr>
<tr>
<td></td>
<td>Experiment b)</td>
<td>Setting order priority</td>
<td>B₁</td>
<td>Obtained</td>
<td>1</td>
<td>2279</td>
<td>1875,7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B₂</td>
<td>Obtained</td>
<td>151</td>
<td>5483</td>
<td>4489,05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B₃</td>
<td>Obtained</td>
<td>1</td>
<td>1762</td>
<td>1444,9</td>
</tr>
</tbody>
</table>

Legend:

Experiment a) – work orders: B₂ and B₃ may be executed in the system, work order B₁ cannot be executed (exceeded directive execution time).

Experiment b) – work orders: B₁, B₂ and B₃ 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
