APEM journal **Advances in Production Engineering & Management** 3 (2008) 2, 59-70
ISSN 1854-6250 Original scientific paper

AGVS SCHEDULING SUBJECT TO DEADLOCK AVOIDANCE CONSTRAINTS

Bocewicz G.; Bzdyra K. & Banaszak Z.
Department of Computer Science and Management
Technical University of Koszalin,75-453 Koszalin, Poland,
E-mail: zbigniew.banaszak@tu.koszalin.pl

Abstract:
The subject matter of the study are the automated guided vehicle (AGV) operation synchronisation mechanisms in flexible manufacturing systems, enabling determination of the travel schedules via some preset travelling route intervals. In many practical cases, transport operations are repetitive. For this type of systems, the transport processes can be modelled as a system of cyclic concurrent processes, sharing common resources with mutual exclusion. The processes examination has to guarantee the collision-free and deadlock free AGV-s flow. In this paper, the problem of determination of the rules coordinating access of the vehicles to the shared travel route intervals, ensuring the collision-free and deadlock-free execution of the repetitive processes, was reduced to determination of the sufficient conditions. In particular, the problem of searching for a pair (initial state, priority rule set), is defined in the form of the constraint satisfaction problem (CSP), and is solved with the use of the constraint programming (CP) techniques.

Key Words: Knowledge Enginekering, Constraints Logic Programming, Deadlock Avoidance, Scheduling

1. INTRODUCTION

A considered class of objects covers the transport subsystems of the flexible manufacturing systems (FMS). In subsystems of that type, along given travelling routes, automated guided vehicles (AGVs) move. In their routes, the vehicles stop at given workstations at strictly defined instants in order to load and/or unload appropriate product batches. In many cases, the vehicles execute repetitive operations, and/or move along cyclic routes, creating cyclic transport processes The due time service of a workstation decides on the admissibility of the given transport subsystem solution alternative. Among the existing admissible, i.e. collision-free and deadlock-free schedules, next there may be chosen those that meet the defined cost criteria. The existing constraints, connected with the available travelling route width (not allowing for vehicle passing by), the topology of travelling routes and itineraries of individual vehicles, lack of simultaneous access to the stations, etc. imply the necessity to investigate conditions leading to possible vehicle collisions and deadlocks [5]. This means that the problem of the given transport alternative solution admissibility check problem belongs to the NP-hard problems [8].

The existing approach solving the problem base usually upon the simulation models, e.g. the Petri nets [6] or the algebraic models, e.g. upon the (max,+) algebra [7]. In this context, this work constitutes some continuation of the investigations conducted in [1, 7].

The problem, considered in this paper, reduces itself for the repetitive transport systems to determination of the rules coordinating the access of the vehicles to the shared system resources (travelling route intervals), ensuring the collision-free and deadlock-free execution of the processes. Assuming existence of local priority decision rules (controlling the access to the shared resources); the problem reduces to determination of the sufficient conditions, in the form of a pair (initial state, priority rule set). The accepted rule-based transport subsystem specification way reduces the synchronization task to solution of an appropriate decision problem of the logic-algebraic method [3]. The problem solution is achieved through application of the constraint programming techniques [1].

The rest of this paper is organized as follows. Section 2 provides basic assumptions and notations, and then states the problem. Section 3 introduces to the logic-algebraic methods, and then discusses the issue of the sufficient conditions guaranteeing deadlock-free AGV-s movement. Section 4 recalls concepts standing behind constraint satisfaction problem. Section 5 provides an illustrative example for application of the approach. The final section summarises this contribution, and sheds light on the future direction of research.

2. PROBLEM FORMULATION

The AGV service systems co-sharing the resources and executing repetitive tasks, may be presented in a form of appropriately formulated Cyclic Concurrent Process Systems (CCPS), wherein the cyclic processes (encompassing AGVs flow) are interconnected one with another by use of the common resources (tracks, machine tools, etc.). In Figure 1, a graphical representation of an exemplary CCPS is presented. In the system, three processes are used, P1, P2, … P8, that reflect the operation of individual vehicles. The vehicles are served (s) by the resources R1 – R14.

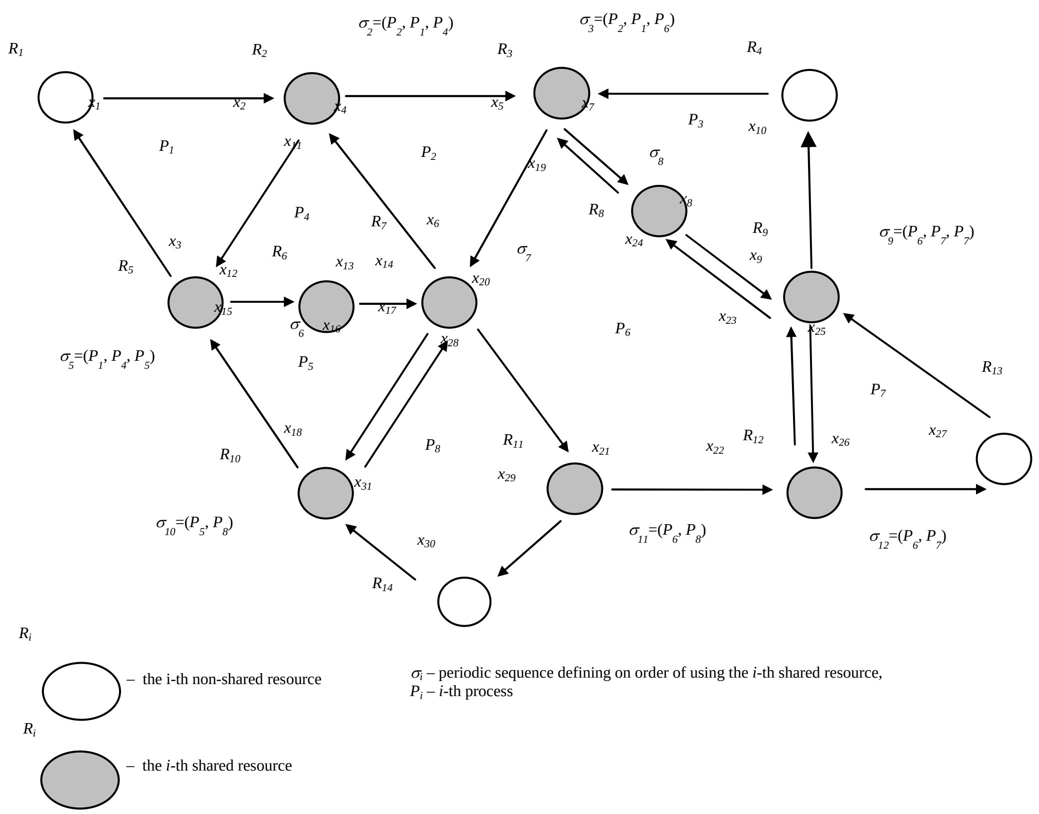

Figure 1: Graphical representation of CCPS.

2.1 Assumptions

For the systems of that type, it is assumed that the cooperation of the processes is determined by the following constraints (Polak, et al., 2004):

- The processes share the common resources in the mutual exclusion mode,
- Commencement of a successive process operation happens immediately after completing of the current operation provided, that there is a possibility of making use of the successive resource requested by the given process,
- During waiting for a busy resource, the process does not release the resource allocated for execution of the previous operation,
- The process is not pre-emptive, i.e. the resource may not be taken of the process while it is using it,
- The processes are executed cyclically,
- In one cycle, a process may pass via any resource along its transportation route once only.

2.2 Notations

In the model of CCPS, the following definitions are used: cyclic process, time representation of a process, shared resource use sequence, initial system state, operation sequence, operation time sequence, priority dispatching rule encompassing access of the processes to the shared resources, and state sequence [7].

A cyclic process $P_i = (p_{i1}, p_{i2}, ..., p_{in})$ is a sequence, the components of which define the numbers of the resources used for execution of individual process operations, where: p_{ij} – denotes the resource used by the i-th process in the j-th operation; after completion of the operation with a share of the resource p_{in}, the operation with the share of the resource $p_{i1,}$ is executed again. The sequence defines the transportation route, i.e., the ordering, in which the given process is executed. Obviously, the process P_i – may begin at an arbitrary resource of the sequence presented.

The time representation of the i-th cyclic process is constituted by the sequence $T_i = (t_{i1}, t_{i2}, ... , t_{in})$, the components of which define the times of execution of individual operations of the relevant process, where t_{ij} denotes the time of execution of the j-th operation by the i-the process.

The sequence $\Theta = (\sigma_j, \sigma_k, ... ,\sigma_z)$ of the priority dispatching rules, i.e., accesses of the processes to the shared resources of the transport subsystem, where $\sigma_i = (s_j, s_k, ... ,s_l)$ – is the sequence, the components of which determine the process service order by the i-th shared resource, s_k – is the process number. Each sequence σ_i is periodical, and gives access to the i-th resource to every process using it. This ensures starvation-free system execution.

The initial state $S_0 = (R_i,R_j, ...,R_k)$ of the transport subsystem is the sequence, the components of which define the initial process resource numbers, where $crd_iS_0 = R_j$ – denotes that the i-th process is begun from realization on the resource R_j; crd_iS_0 – denotes the i-th coordinate of the vector S_0.

The sequence of operations of the size equal to the number of all operations, executed in the system, is defined as $p = (P_1,P_2,...,P_r) = (p_{11}, ..., p_{1n1}, p_{21}, ..., p_{2\,n2}, ..., p_{r1}, ..., p_{r\,nr})$, where p_{ij} – denotes the j-th operation of the i-th process (the number of the resource used for realization of the j-th operation of the i-th process).

The operation execution time sequence $T = (T_1,T_2,...,T_r) = (t_{11}, ..., t_{1\,n1}, t_{21}, ..., t_{2\,n2},..., t_{r1} ,...,t_{r\,nr})$, where t_{ij} denotes the execution time of the j-th operation by the i-th process.

The state sequence (state vector) $x = (x_1,x_2,...,x_r)$, where x_i corresponds the operation represented in the sequence p by the i-th coordinate (crd_ip), the value x_i denotes the instant that the operation is begun in the first cycle.

2.3 Problem statement

For the system described in such way, the following problem is defined: There is given a system of class CCPS, mapping the operation of automated guided vehicles. The system structure and the process parameters are given in the form of vectors defining the vehicles P_i routes and service times T_i in subsequent stations. The following question should be answered: Does it exist a pair (initial system state S_0, priority rule set Θ) ensuring that the assumed transport processes are executed, with the cycle time not exceeding the arbitrarily given value H?

The answer to the problem formulated in such way covers, therefore, the response to the question, if there exist the sufficient conditions that, when met, ensure the cyclic (i.e. deadlock-free) execution of the concurrent processes.

3. LOGIC-ALGEBRAIC METHOD

The components of the system class under consideration may be described in the form of the representation of the knowledge base: $RW = <C,W,Y; R>$, where: $R = \{(c,w,y): F(c,w,y) = 1\}$ – a relation being the set of all triples (c,w,y), for which the facts F describing the system are true; $F(c,w,y) = (F_1(c,w,y), F_2(c,w,y),...,F_K(c,w,y))$ is the composition of the logic fact values being the functions of the variables c, w, y; $c = (c_1, c_2,...,c_k)$ – the set of the input variables; $y = (y_1,y_2,...,y_r)$ – the set of the output variables; $w = (w_1,w_2 ...,w_r)$ – the set of the auxiliary variables; c□C, y□Y, w□W, C,Y,W – the sets defining the domains of the variables c,y,w.

3.1 Knowledge base

The knowledge base representation RW, describing an arbitrary system, is presented in the form of the sets C, W, Y, that define the domains of some c, y, w, describing (on the qualitative level) some system properties. The variables c are called the input variables, describing the input properties of the system, the variables y - the output variables, describing the output properties of the system, the variables w are the auxiliary variables. The knowledge defining the properties of the system under consideration, is presented in the form of the fact set $F(c,w,y)$. The facts $F(c,w,y)$ are propositions reflecting, on the logic level, the connections occurring between individual variables c,w,y. The triples c,w,y, for which all facts $F(c,w,y)$ are true, are presented in the form of the relation R. In this context, the representation of the knowledge for the systems CCPS is defined as follows [9]:

$$RW = <S0, \Sigma, X; R> \quad (1)$$

where: $S0$ – the set of all possible initial states S_0 (input variables), Σ - the set of all possible access rules Θ for the shared resources (input variables), X – the set of all possible forms of the state vector x (output variables), $R = \{(S_0,\Theta, x): F(S_0,\Theta, x) = 1\}$ – the relation defining the values S_0,Θ, x, for which the facts $F(S_0,\Theta, x)$ are true. The set R covers the facts $F(S_0,\Theta, x)$, being logic propositions that describe the system properties in dependence of the initial state S_0, the rules of access to shared resources Θ and of the starting times of individual operations x.

3.2 Knowledge generation

The knowledge base *RW* considered, can be treated as specification of general assumptions the transportation system has to follow (see Section 2.1), while taking into consideration the time and resources constraints. For example the assumption: *"during waiting for a busy resource, the process does not release the resource allocated for execution of the previous operation"*, has to be specified by a set of propositions (constraints) guaranteeing its satisfaction for any process, and any resource at any moment of time. In other words, the knowledge about situations that might happened is specified by a set of facts $F(S_0, \Theta, x)$. The assumptions regarding the transportation system considered are as follows [9]:

- Constraints regarding an order in which processes have to be executed.
- Constraints limiting processes servicing by local resources: the moment x_j that the process P_i may start its execution at the resource R_k is equal to the time required by this process to complete its previous operation:

$$x_j = x_{j-1} + t_{j-1} \tag{2}$$

where: t_{j-1} – the operation time on the resource R_{k-1}, x_{i-1} – the instant that the operation is begun at the resource R_{k-1}.

- Constraints regarding processes servicing by shared resources: the instant x_j that the operation of the process P_i is begun at the shared resource R_k is determined by the
- Maximum within the completion time of the process P_i on the subsequent resource R_{k-1}, and the instance the operation corresponding to x_p begins its execution on the resource R_{k+1} has been served P_o previously executed on R_k just before P_i.

$$x_j = \max\{x_p, x_{j-1}+t_{j-1}\} \tag{3}$$

The constraints provided (2), (3), can be seen as a part of a transport system specification in the context of its parameters x, S_0, Θ, [9] i.e. a part of the relevant knowledge base.

Additional to the mentioned assumptions guaranteeing of collision-free and starvation-free processes execution, other guaranteeing processes deadlock-freeness can be introduced. Illustration of processes deadlock is shown in Figure 2.

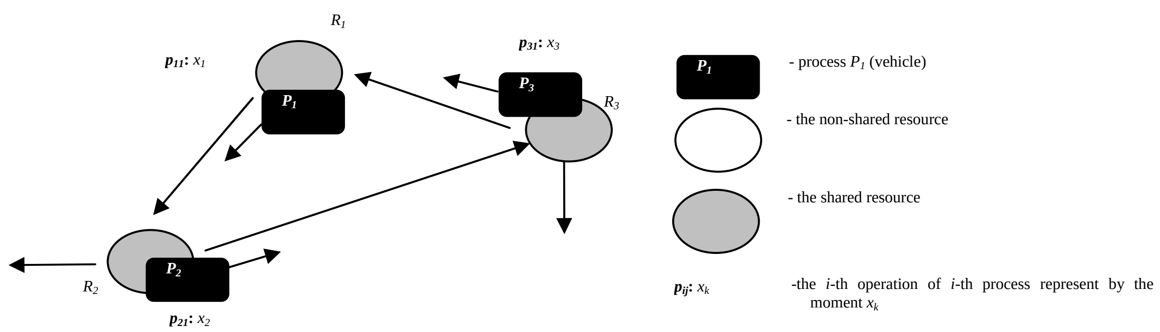

Figure 2: Illustration of the processes deadlock.

Each of three AGVs associated to processes P_1, P_2 and P_3 waits for resource occupied by one of them – processes call for resources and their calls form a closed circle of resources request, i.e. the processes deadlock. It is easy to note, that the formulas (2) and (3), can be unified in the following recurrent formulae, named a state equation (4):

$$^{p}x_v^{s(i+1)}(x) = \max\{x_j + t_j, \max\{x_{j+1} + t_{j+1}, \dots, \max\{x_{nz-1} + t_{nz-1}, \max\{x_{nz} + t_{nz}, \dots, \max\{x_{j-1} + t_{j-1}, x_j\}\dots\}\}\}\} \quad (4)$$

where: x_q – denotes the moment the process P_j starts its execution at the state S_i, $^{p}x_v^{s(i+1)}(x)$ – denotes the moment the p-th process starts its execution at the next state S_{i+1}.

For the given state S_i the moment the process P_i starts its execution at the next state S_{i+1} can be calculated due to the formulae (4). For example the state equation for the process P_1 from Figure 2 has the following form: $^{1}x_v^{s(i+1)}(x) = \max\{x_1+t_1, \max\{x_3 + t_3, \max\{x_5+t_5, x_2\}\}\}$.

It can be shown that if the state equation belongs to a class of the identity equations, then the system considered is deadlock-free one. So, the state equation (4) can be employed for calculation of an operation starting moment at any further state. This can be realized through the so-called general state equation (5):

$$^{p}X_v^{s(i+k)}(x) = {}^{ps}x_v^{s(i+k)}({}^{ps}x_v^{s(i+k-1)}({}^{ps}x_v^{s(i+k-2)}(\dots {}^{ps}x_v^{si}(x) \dots))) \quad (5)$$

where $^{p}X_v^{s(i+k)}(x)$ – determines the moment the *p*-th process starts its operation at the state S_{i+k}. In other words, for any state S_i the beginning moment of particular operation at any next state S_{i+k} can be determined.

Consider the case the state Si states for the initial state S0, and the Si+k is the last one Sw in the system cycle. Such assumption leads to the equation (6),

$$^{p}X_v^{sw}(x) = {}^{p}x_v^{sw}({}^{p}x_v^{s(w-1)}({}^{p}x_v^{s(w-2)}(\dots {}^{p}x_v^{s0}(x) \dots))) \quad (6)$$

where: $^{p}X_v^{sw}(x)$ – determines the moment the *p*-th process starts its operation at the state S_w, the last one in the cycle. Consequently, the following Theorem 1 can be proven.

Theorem1.

Given a transportation system, specified by a set of transportation routes $\{P_1, P_2, \dots, P_n\}$. Given an initial state S_0, and a set of dispatching rules Θ. If the general state equation (6) is not identity one, then the transportation system is deadlock-free.

In other words, the Theorem 1 provides the condition examination of which allows one to avoid deadlocks. Therefore, a set of facts $F(S_0, \Theta, x)$ following constraints (2), (3), and (4) allows one to detect possible deadlocks occurrence in the course of concurrent processes execution. It means the knowledge base representation *RW* (1) may be determined automatically on the basis of the system structure, logistic constraints and parameters P_i, T_i.

The accepted way of the system specification, based upon the rules, enables to determine the sufficient conditions in the form of the initial states S_0 and the process service rules Θ, ensuring the deadlock-free operation of the system in cycles not exceeding the preset value of *H*. In this connection, so-called decision problem is to be solved [3].

3.3 Decision problem

For a given system CCPS, described by the representation *RW* (1), one searches for such form of the input relation R_x, that will guarantee meeting of the known output relation R_y. The relations R_x and R_y are defined as follows: $R_x = \{(S_0,\Theta): F_c(S_0,\Theta) = 1\}$ – the set of the values S_0, Θ, for which the input system property $F_c(S_0,\Theta)$ is met, while $R_y = \{x: F_y(x) = 1\}$ – the set of values of x for which the output property $F_y(x)$ is met, where: $F_c(S_0,\Theta)$ is a set of the logic propositions that describe the input system properties in dependence on the initial state S_0 and the rules of access to the shared resources Θ, while $F_y(x)$ is a set of the logic sentences that describe the output properties of the system in dependence of the values of the sequence x. For the system under consideration, the output property is of the form: $F_y(x)$: $(x_1+t_1 \le H) \wedge (x_2+t_2 \le H) \wedge ... \wedge (x_{rn}+t_{rn} \le H)$.

The decision problem consists in determination of such relation form Rx for which the input property Fc meets the following implication.

$$F_c(S_0,\Theta) \Rightarrow F_y(x) \tag{7}$$

The relation R_x determined constitutes, therefore, the answer to the question asked, R_x is a set of such values S_0, Θ, for which the system will operate without collisions and deadlocks in cycles not lasting longer than the value of H. The derivation of the relation R_x on the basis of the logic-algebraic method [4] occurs with use of the sets S_{x1}, S_{x2}:

$$R_x = S_{x1} \setminus S_{x2} \tag{8}$$

where: $S_{x1} = \{(S_0,\Theta): F(S_0,\Theta, x) = 1, F_y(x) = 1\}$;
$S_{x2} = \{(S_0,\Theta): F(S_0,\Theta, x) = 1, F_y(x) = 0\}$.

The set S_{x1} can be seen as a set consisting of S_0, Θ, whose follow the facts: $F(S_0,\Theta, x)$, $F_y(x)$. In turn the set S_{x2} can be seen as a set consisting of S_0, Θ, whose follow the fact $F(S_0,\Theta,x)$ and do not follow $F_y(x)$. The intersection of sets S_{x1}, S_{x2} can consist values of S_0, Θ, for which the fact $F_y(x)$ is not determined. It happens when some facts of the knowledge base *RW*-results in state equations (8) being identity ones (an equation that is valid for all values of its variables) – such property means the system is not deadlock-free. Therefore, $S_{x1} \setminus S_{x2}$ guarantees the transportation system considered is deadlock-free.

Determination of the set $R_x = \emptyset$ denotes the lack of answer to the question asked. In a general case searching for the relation R_x is an NP-hard problem. For example the following knowledge base: $RW = \langle A;R \rangle$, where: $R = \{ a: F(a) = 1\}$ relation linking variables a treated as facts $F(a)$, $a = (p, r, s, t, q, w)$, $p,r,\ldots,w \in A$, $A = \{0,1\}$, $F(a) = (F_1(a), F_2(a), F_3(a), F_4(a))$. Considered sequence of facts consists of: $F_1(a)$: $w \Rightarrow p$; $F_2(a)$: $p \wedge q \Rightarrow r$; $F_4(a)$: $s \Rightarrow q$; $F_5(a)$: $t \Rightarrow q$.

Set of variables consists a subset of so called input variables: $a_x = (w, s, t)$; a subset of auxiliary: $a_w = (p, q)$; and a subset of output variables: $a_y = r$.

The considered question is: *What input facts $F_x(a_x)$, guarantee the fact $F_y(a_y)$ holds?* , where: $F_y(a_y)$: r.

Searching for $F_x(a_x)$ (i.e., the set R_x) requires both sets S_{x1} and S_{x2}. The sets are determined through the true tables searching (see Figure 3).

S_{x1}

w	s	t	...	r
0	0	0	...	1
0	0	1	...	1
0	1	0	...	1
0	1	1	...	1
1	0	0	...	1
1	0	1	...	1
1	1	0	....	1
1	1	1	...	1

S_{x2}

w	s	t	...	r
0	0	0	...	0
0	0	1	...	0
0	1	0	...	0
0	1	1	...	0
1	0	0	...	0

$S_x = S_{x1} \backslash S_{x2}$

w	s	t	...	r
1	0	1	....	1
1	1	0	...	1
1	1	1	...	1

Figure 3: True tables corresponding to sets: S_{x1}, S_{x2}, S_x.

Gray rows in Figure 3 correspond to solution, i.e. S_x= =S_{u1}/S_{u2}. In terms of triples (*w*, *s*, *t*) the set S_x has the form: $S_x = \{(1,0,1),(1,1,0),(1,1,1)\}$. Finally S_u corresponds to the fact: $F_x(a_x)$: $(w \wedge (\neg s) \wedge t) \vee (w \wedge s \wedge (\neg t)) \vee (w \wedge s \wedge t)$, i.e. $F_u(a_u)$: $(s \wedge t) \vee (t \wedge w)$.

4. CONSTRAINT SATISFACTION PROBLEM

Each knowledge base representation *RW* (1) of the concurrent cyclic process system may be presented in the form of the constraint satisfaction problem (*CSP*); [2]. The problem *CSP* = ((*Q*, *D*), *Co*) is defined as follows:

There is given a finite set of discrete decision variables $Q = \{q_1, q_2, ... ,q_n\}$, a family of finite variable domains $D = \{D_i \mid D_i = \{d_{i1}, d_{i2}, ..., d_{ij}, ..., d_{im}\}, i = 1..n\}$, and the finite set of constraints $Co = \{Co_i \mid i = 1..L\}$ limiting the decision variable values. The admissible solutions where the values of all variables meet all constraints of the set *Co* are sought for.

4.1 Knowledge base representation

In the case of the problem *CSP,* mapping the knowledge representation *RW*, the role of the constraints *Co* is fulfilled by the facts included in $F(S_0,\Theta, x)$ while the role of the variables *Q* – the values of the variables S_0, Θ, *x*. The variable domains are in the form of the sets D_{S0}, D_Θ, D_x. The problem *CSP* results in the form:

$$CSP = (((S_0,\Theta, x), D), \{F(S_0,\Theta, x) = 1\}) \quad (9)$$

where: $D = \{D_{S0}, D_\Theta, D_x\}$, D_{S0} is the set of the resources included in the system; D_Θ is the set including the processes realized in the system; $D_x = \{0,1, ... ,H\}$ is the set of the time values; $F(S_0,\Theta, x) = 1$ denotes the series of facts: $(F_1(S_0,\Theta, x) = 1, ... , F_K(S_0,\Theta, x) = 1)$.

4.2 Logic-algebraic method based solution

The solution of the *CSP* problem formulated in such way is a set of sequence values of the initial state of the system S_0, the rules of the process access to the shared resources Θ and of the starting times of individual operations on the resources x for which all constraints presented in the form of the logic sentences $F(S_0, \Theta, x)$ are true.

Solving the decision problem (determination of the relation R_x) in the context of *CSP* is connected with solving the following two problems:

$$CSP_{Sx1} = ((S_0, \Theta, x), D), \{F(S_0, \Theta, x) = 1, F_y(x) = 1\}) \qquad (10)$$
$$CSP_{Sx2} = ((S_0, \Theta, x), D), \{F(S_0, \Theta, x) = 1, F_y(x) = 0\})$$

The results of solving the problems formulated in such way are the sets S_{x1} and S_{x2}. The sets enable to determine the set R_x (8).

In turn, the obtained set R_x can be treated than as solution of the considered *CSP*, i.e. determines the response to the assumed query; moreover R_x fulfils the role of the sufficient conditions. The set R_x includes a group of alternative solutions in the form of the value of the sequence of the initial state S_0 and the sequence of the access rules of the processes to the shared resources Θ ensuring, for the given transport process system CCPS, the deadlock-free vehicles service during the assumed time period *H*.

5. EXAMPLE OF SCHEDULE DESIGN

There is given an automated guided vehicle system of the structure depicted in the Figure 1. The processes $P_1, P_2, \ldots, P_8$ reflect service of three *AGVs* according to the service times $T_1, T_2, \ldots, T_8$. The vehicles are served by the service points $R_1, R_2, \ldots, R_{13}$, the shared system resources. The following parameters are known:

$P_1 = (R_1, R_2, R_5)$,
$P_2 = (R_2, R_3, R_7)$,
$P_3 = (R_3, R_8, R_9, R_4)$,
$P_4 = (R_2, R_5, R_6, R_7)$,
$P_5 = (R_5, R_6, R_7, R_{10})$,
$P_6 = (R_3, R_7, R_{11}, R_{12}, R_9, R_8)$,
$P_7 = (R_9, R_{12}, R_{13})$,
$P_4 = (R_7, R_{11}, R_{14}, R_{10})$,

$T_1 = (1,2,3)$,
$T_2 = (1,2,3)$,
$T_3 = (2,1,3,4)$,
$T_4 = (1,2,3,4)$,
$T_5 = (1,2,3,4)$,
$T_6 = (1,2,3,1,2,3)$,
$T_7 = (1,2,3)$,
$T_4 = (1,2,3,8)$,

$t=(1,2,3,1,2,3,2,1,3,4,1,2,3,4,1,2,3,4,1,2,3,1,2,3,1,2,3,1,2,3,4)$,

$x = (x_1, x_2, \ldots, x_{30})$, $p = (R_1, R_2, \ldots, R_{13})$.

The initial state vector $S_0 = (R_i, R_j, R_k, R_l, R_m, R_n, R_o, R_p)$, where: $R_i \in \{R_1, R_2, R_5\}$, $R_j \in \{R_2, R_3, R_7\}$, $R_k \in \{R_3, R_8, R_9, R_4\}$, $R_l \in \{R_2, R_5, R_6, R_7\}$, $R_m \in \{R_5, R_6, R_7, R_{10}\}$, $R_n \in \{R_3, R_7, R_{11}, R_{12}, R_9, R_8\}$, $R_o \in \{R_9, R_{12}, R_{13}\}$, $R_p \in \{R_7, R_{11}, R_{14}, R_{10}\}$, is distinguished, defining the initial process resources, the access rule vector $\Theta = (\sigma_2, \sigma_3, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}, \sigma_{11}, \sigma_{12})$, where $\sigma_2 = (s_{21}, s_{22})$, $\sigma_2 = (s_{21}, s_{22}, s_{23})$, ... ,$\sigma_{12} = (s_{121}, s_{122})$. In accordance with the rules described in [9], the knowledge representation corresponding to the specification given is of the form: $RW = <S0, \Sigma, X; R>$, where: $R = \{(S_0, \Theta, x): F(S_0, \Theta, x) = 1\}$.

5.1 A routine query

The answer to the question: Does it exist such combination of the initial states S_0 of the system and the set of the vehicle access rules σ_2, ... , σ_{12}, for the shared resources that guarantee that the process realization cycle will not exceed 13 time units?

The answer to such formulated question is the solution of the decision problem. In accordance with the procedure illustrated in Figure 3, the following *CSP* problems are being solved (10): $CSP_{Sx1} = ((S_0, \Theta, x), D), \{F(S_0, \Theta, x) = 1, F_y(x) = 1\})$, $CSP_{Sx2} = ((S_0, \Theta, x), D), \{F(S_0, \Theta, x) = 1, F_y(x) = 0\})$, where: $D = \{D_{S0}, D_\Sigma, D_x\}$, $D_\Sigma = \{D_{\sigma 2}, ... , D_{\sigma 12}\}$, $D_{S0} = \{R_1 ..., R_{14}\}$, $D_{\sigma 2} = \{P_1, P_2, P_4\}$,… ,$D_{\sigma 12} = \{P_6, P_7\}$, $D_x = \{1,...,40\}$.

5.2 Sufficient conditions

The fact $F_1(S_0,\sigma_{12},x)$ defining the starting time of the operation x_2, assuming the mutual exclusion of the processes P_6, P_7 on the resource R_{12}, is of the form: $F_1(S_0,\sigma_2,x)$: $\neg(crd_6 S_0 = R_{12}) \wedge \neg(crd_6 S_0 = R_9) \wedge \wedge (crd_1 \sigma_{12} = P_7) \Rightarrow (x_{22} = \max(x_{25}, x_{21}+t_{21}))$ i.e., if in the state S_0 the process P_6 does not use the resource R_{12}, and in the state S_0, the process P_7 does not use the resource R_9, and – as the first one, the P_7 process uses of the resource R_{12}, then $x_{22} = \max(x_{25}, x_{21}+t_{21})$. The fact defining the output property $F_y(x)$: $(x_1+ t_1 \leq 13) \wedge (x_2+t_2 \leq 13) \wedge \ldots \wedge (x_{30}+t_{30} \leq 13)$ corresponds to the preset condition: "*the cycle will not exceed 13 time units*".
For solving the problems CSP_{Sx1}, CSP_{Sx2}, the constraints programming language, OzMozart (Banaszak, *et al*. 2005), was applied. The obtained solution R_x contains 318 sufficient conditions. One of them is as follows:
$\{S_0 = (R_1, R_3, R_4, R_7, R_5, R_{12}, R_{13}, R_{10})$, $\sigma_2 = (P_4, P_1, P_2)$, $\sigma_3 = (P_2, P_6, P_3)$, $\sigma_5 = (P_5, P_4, P_1)$, $\sigma_6 = (P_5, P_4)$, $\sigma_7 = (P_4, P_8, P_5, P_2, P_6)$, $\sigma_8 = (P_6, P_6)$, $\sigma_9 = (P_6, P_7, P_3)$, $\sigma_{10} = (P_8, P_5)$, $\sigma_{11} = (P_8, P_6)$, $\sigma_{12} = (P_6, P_7)\}$.
The set R_x constitutes the set of the alternative sufficient conditions that must be fulfilled by the system under consideration in order that the vehicles may realize the planned operations within cycles not exceeding 13 time units.

5.3 Admissible solutions

An exemplary Gantt's chart illustrating the use of the system resources has been presented in the Figure 4. The diagram corresponds to the following sufficient condition $\{S_0 = (R_2, R_3, R_8, R_6, R_7, R_{11}, R_{13}, R_{14})$, $\sigma_2 = (P_1, P_4, P_2)$, $\sigma_3 = (P_2, P_6, P_3)$, $\sigma_5 = (P_1, P_5, P_4)$, $\sigma_6 = (P_4, P_5)$, $\sigma_7 = (P_5, P_4, P_2, P_8, P_6)$, $\sigma_8 = (P_3, P_6)$, $\sigma_9 = (P_3, P_6, P_7)$, $\sigma_{10} = (P_5, P_8)$, $\sigma_{11} = (P_6, P_8)$, $\sigma_{12} = (P_6, P_7)\}$.
The processes are executed cyclically within the cycle not exceeding 13 time units.

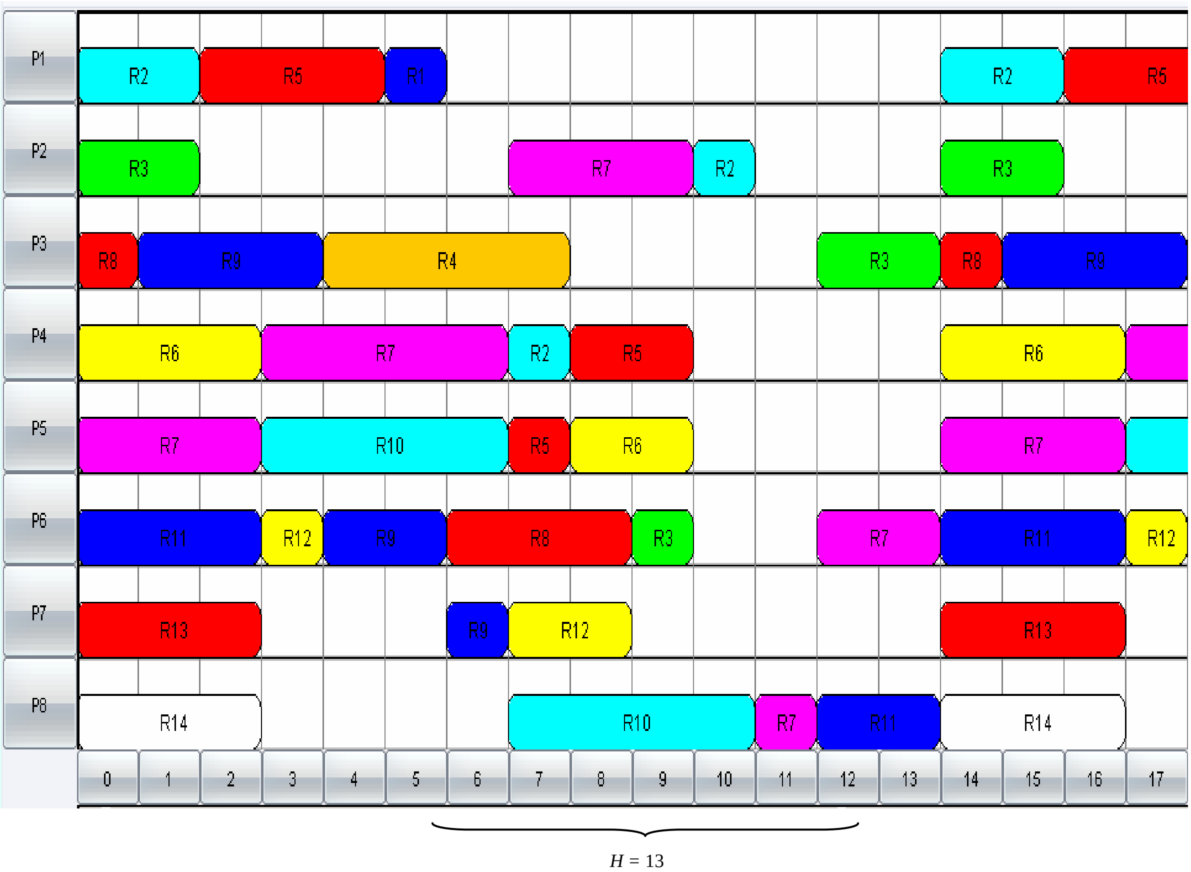

Figure 4: The Gantt's chart.

6. CONCLUSIONS

On the basis of the rule-based specification of a transport system, formulated in the form of the knowledge base representation *RW*, it is possible to seek for schedules guaranteeing the collision-free and deadlock-free operation of the AGV-s system. The presented concept of automated generation of the knowledge representation *RW,* and the approach to determination of the sufficient conditions through specification of the CSP problem enables construction of the interactive (operating in the on-line mode) computer aided prototyping systems for distributed control procedures.

REFERENCES

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

[2] Barták R. (2004). Incomplete Depth-First Search Techniques. In: *A Short Survey, Proceedings of the 6th Workshop on Constraint Programming for Decision and Control*, (Ed. Figwer J.), pp. 7-14

[3] Bubnicki Z. (1990). *Introduction to Expert Systems*. PWN, Warsaw

[4] Bubnicki Z. (1999). Learning processes and logic-algebraic method for the systems with knowledge representation. *Systems analysis and management. Systems Research Inst*. PAS

[5] Lawley M.A., Reveliotis S.A., Ferreira P.M. (1998). A correct and scalable deadlock avoidance policy for flexible manufacturing systems. *IEEE Trans. on Robotics and Automation*; Vol.14, No.5: pp. 796-809

[6] Lee T., Song J. (1996). Petri net modeling and scheduling of periodic job shops with blocking. In: *Proc. of the Workshop on Manufacturing and Petri nets*, Osaka, Japan, 25 June, pp. 197-214

[7] Polak M., Majdzik P., Banaszak Z.A., Wójcik R. (2004). The performance evaluation tool for automated prototyping of concurrent cyclic processes. *Fundamenta Inf.*, Vol.60, No.1-4: pp. 269-289

[8] Ramamritham K. (1995). Allocation and scheduling of precedence-related periodic tasks. *IEEE Trans. on Parallel and Distributed Systems*, Vol.4, No. 6: pp. 412-420

[9] Bocewicz G., Wójcik R., Banaszak Z. (2006). Scheduling of AGV systems under access restrictions to shared resources of the ESP (logic-algebraic model). *Proc. of National Conf. on Robotics* (K. Tchoń), WKŁ, Wroclaw, pp. 149-163