Argumentation based negotiation in multi-agents application

Ivana Budinská
Institute of Informatics SAS
Dúbravská cesta 9
845 07 Bratislava
Slovakia
utrrbudi@savba.sk

Baltazár Frankovič
Institute of Informatics SAS
Dúbravská cesta 9
845 07 Bratislava
Slovakia
utrrfran@savba.sk

Abstract – **The paper introduces a scheduling problem solution in MAS (Multi agents system), where each agent advocates its alternative of solution. An agent for each task in a system is supposed. This is a part of a product-oriented control of a manufacturing system. Agents propose schedule for its products independently on the other agents. Agents have to negotiate their proposed schedules and to find the final schedule, which is the best for the whole system. Support of alternative solutions is based on argumentation. Formalization of arguments using weights is described in the paper. The global criterion, which is used to choose the best alternative solution, is based on a respond of the whole system to the chosen alternative. An agents with a " winning" alternative is a ruling agent and all other agents has to adapt their schedules to according the alternative of the ruling agent.**

I. INTRODUCTION

Recently manufacturing system face to new control principles. The most challenging is to satisfy customers' demands, that means, the production process moves from manufacturing oriented – mass production (that means within a shop many products with the same features and specifications were produced) towards product-oriented, mass customisation production. Products are produced in very small series, even only one or some pieces according to customers' requirements. To satisfy this new approach in production system, production line, or job shops have to be flexible, and control of production should support product-oriented manufacturing. It certainly doesn't mean to redesign production lines for each product. Products differ from each other in some specifications, like colour, number of additional equipment, etc. For each product an extra production plan and schedule is elaborated. The problem is to coordinate the production of more products within one job-shop to ensure all time requirements for all products and also to ensure the high utilization of production lines, the least delays and the least idle time of workstations. The solution is offered by using of MAS and holonic systems with combination of multi agent systems (MAS), which are flexibile, fault tolerant, and should have real-time information system combined with dynamic scheduling and rescheduling. Using MAS enables to use an agent for each product to be produced within a job-shop. In the beginning, agents build a schedule for products independently on the other agents. That means, it is a one product multi-machine scheduling problem, which is very well described in many publications [9]. The problem is, that there are more products and more agents, and they have to coordinate their schedules to share resources. This leads to a multi task–multi machine scheduling problem, which can be solved in many ways [1], [2], [3], [4] etc. The proposed solution is based on negotiation among agents. Agents use arguments to support their alternatives solutions. The final solution is the best from the point of the whole system.

The section 2 formalizes a scheduling problem for which the proposed method can be used. The section 3 is addressed to MAS and one-product-one–agent method for supporting product oriented manufacturing. Section 4 describes possible methods for MAS coordination and cooperation by using arguments in a negotiation process. The Section 5 describes an example of production system to illustrate how the proposed theory can be used for real world systems. In the conclusion some ideas for the future work are discussed.

II. FORMULATION OF A PROBLEM

A flexible manufacturing system is supposed with a number of products to be produced is supposed. A job shop consists of a number of machines that represents resources. Production of each product is considered as a job. Each job consists from a number of operations. Operations are associated with machines. For each operation some facts are known. The following formalization is used in this paper:

A set of machines:

$$M = \{m_1, m_2, \dots m_m\} \quad (1)$$

A set of jobs, each job is associated with a product:

$$J = \{j_1, j_2, \dots, j_n\} \quad (2)$$

Each job consists of a set of operations:

$$O_j = \{o_{1j}, o_{2j} \dots, o_{n_j j}\} \quad (3)$$

A scheduling problem may be formalized as follows:

$$\{r_j, d_j, (m_1; p_{1j1})^{o_{1j}}, \dots (m_m; p_{1jm})^{o_{1j}}, \dots (m_1; p_{n_j j1})^{o_{n_j j}} \dots (m_m; p_{n_j jm})^{o_{n_j j}}\} \quad (4)$$

Where

r_j - a time when a job is ready to start

d_j - a demand time, where a job should be finished

p_{ijm} - a processing time of an operation o_{ij} on a machine m_m

A schedule is a mapping from a set of operations to the set of machines and set of starting times T_0.

$$sch: \bigcup_{j \in J} O_j \rightarrow M \times T_0 \quad (5)$$

A feasible schedule has to keep also constraints. There are typically hard and soft constraints. Hard constraints represent a technological sequence of operations and some time constraints (any operation on the same machine must not start before the preceding operation on the machine is finished, any operation must not start before its ready time, etc.) (e.g.[1], [2], [3]) Soft constraints represents some optimization criteria.

Constraints are formulated as follows:

An operation must not start before its ready time:

$$start\_t(o_j^1) \geq r_j \quad (6)$$

An operation must not start before a preceding operation from the same job is finshed:

$$start\_t(o_j^i) + p_j^i \leq start\_t(o_j^{i+1}) \quad (7)$$

An operation must not start before a preceding operation on the same machine is finished:

$$start\_t(o_j^h) + p_j^h \leq start\_t(o_k^i)$$

$$(8)$$

$$start\_t(o_k^i) + p_k^i \leq start\_t(o_j^h)$$

An operation must not finish after a demand time for the job:

$$start\_t(o_j^{nj}) + p_j^{nj} \leq d_j \quad (9)$$

A feasible schedule may be formulated as follows:

$$sch(j_i) = \left[(p_{ijm}, t_{01_i})^{m_i}, (p_{ijm}, t_{02})^{m_i}, (p_{ijm}, t_{03})^{m_i} \right] \quad (10)$$

III. AN ARCHITECTURE OF A MAS FOR PRODUCT-ORIENTED MANUFAFACTURING

Advantages of using MAS is in implementation may be formulated as follows [5]:

Modularity: Each agent is an autonomous module and can work without interventions of the external world. Each agent can have different capabilities or functionalities and through cooperation the agents are able to achieve a variety of goals. From the practical point of view, producing a number of agents (e.g. software agents - programs) with different capabilities is more effective than creating one agent (e.g. a program), which is able to do everything. In addition, the MAS approach allows separating the original problem solving to a number of sub-problems of a manageable size.

Parallelism: The MAS approach supports parallel processing. A complicated problem could be solved in an acceptable time by using a number of agents, e.g., schedule for all jobs can be calculated independently and then a consensus about a final schedule is done via negotiation process.

Flexibility: MAS are able to react flexibly to each change occurred in the environment. Through cooperation the agents can assist each other to compensate the lack of capability or knowledge. They can share information or own capacity to resolve a newly appeared situation, if one agent is not able to resolve. Beside that, each intelligent agent can do reasoning about with whom and when to cooperate, in order to achieve effective performance.

There are some types of architectures of MAS know form literatures.

Hierarchical architecture is very similar to a centralized approach. Agents are arranged according their function in a hierarchical manner. One agent is on the top of these architecture. Centralization is a big disadvantage of such architecture.

Autonomous agents represents a system built form a number of autonomous agents. Their activities have to be coordinated. A solution for this approach may be coalition creation. Agents are grouped in coalitions according their goals and after negotiation. An agent has to decide whether to join a coalition or not by assessment its benefits and also benefits of the whole system. A new created coalition figures now as a new agent with new properties, which are not supposed to be a superposition of all agents in a coalition. This approach is addressed in [6],[7], [8].

Federative architecture enables grouping agents according various rules. Opposite to a coalition, agents in a group in federative architecture do not create a new entity. They cooperate with other groups of agents via a special agent. According to the level of agent's mediator autonomy, three types of agents mediator are known:

- Agent facilitator - transforms messages from an agent to another
- Agent broker – in addition to the facilitator also monitors changes in environment
- Agent mediator – guarantees cooperation among intelligent agents, provides access to the right information on right time, include learning algorithms to adapt on environmental changes.

MAS architecture for manufacturing system supposed three types of agents. There are

- Agent of resource -AVZ
- Agent of a job - AVU
- Agent mediator - AMVZ

Agents of resources can be grouped according properties of resources. One group of resources represents all resources where similar operations can be provided (difference is in different settings).

A simple communication schema is depicted in Fig.: 1.

Agents communicate among themselves using ACL (Agent Communication Language). Using standards is MAS is a very good habit, because it enables to substitute one agent by another without changing its body. The message sent in ACL is comprehensible by any other agent that keeps FIPA standards.

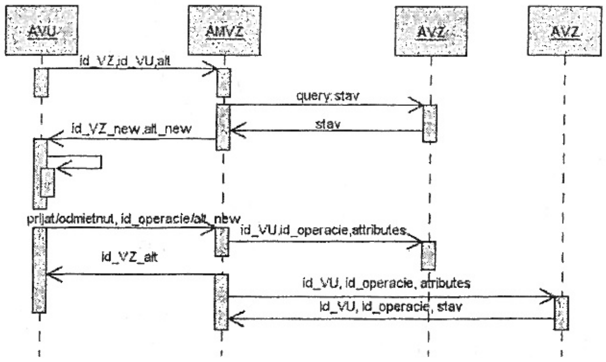

Fig.: 1 A simple communication among agents in a manufacturing system

IV. ARGUMENTATION BASED NEGOTIATION

A systems with a number of intelligent agents of jobs is considered. Negotiation process ensures the best solution for the system as a whole. If more agents are involved in a negotiation process, the negotiation becomes very difficult. All agents prefer own alternatives. Following the common goal an agent to rule other agent should be established. A good criterion for establishing a ruling agent is to assess local criteria function and then the agent with the highest contribution to the global criterion function is established to rule other agent. Thus, in a case of conflict situation the "ruling agent's preferred alternative" is chosen. For the beginning, agents calculate schedules independently. It is obvious, that each agent calculate a schedule, which satisfies its constraints and optimisation requirements the best. Those schedules represent alternatives that are forced by each agent. Alternatives are created by agents as follows:

An agent AVUi calculates a schedule as if there aren't any other jobs in the system. Then new constraints for other agents are defined, according this "optimal" schedule of AVUi. Then schedules for other agents are calculated. The schedule created in such manner, represents an alternative of an agent AVUi. Alternatives are evaluated according defined criteria. The minimal makespan for the whole system is a criterion to choose the best alternative, in this paper. A global criterion functions, for proposed agents' schedule, are evaluated.

According a global criterion function an alternative for all agents is chosen and a profit for all agents, when they accept this chosen alternative, is calculated.

If an alternative is chosen, all agents have to calculate new schedules with the constraints given by the chosen schedule. Then a criterion function is evaluated again.

Negotiation process:

If the chosen schedule does not satisfy (in many cases) all agents requirements, agents propose another schedule to evaluate and negotiate. Then step 1 is repeated.

Evaluation of alternatives:

Let k is an agent from [1,n]. A reward function of agent k is $a_k(x)$. Each agent is associated with a weight w_k.

If $\sum_{k=1}^{n} w_k = 1$, then a combined reward from all agents can be expressed as

$$a(x) = \sum_{k=1}^{n} w_k a_k(x) \qquad (11)$$

Each agent is associated with a set of possible alternatives. Let α_{ik} is an alternative i associated with an agent k.

Rewards of an agent are defined as a difference between positive and negative rewards. A positive reward is a function of profit and gain of an agent, and a negative reward is a function of costs, e.g. production costs, distribution costs, spending and lost.

A reward function for an agent depends also on an alternative plan for this agent.

Let

$$a_k(x) \mid \alpha_i = a_k(x)\alpha_{ik} \qquad (12)$$

is a value of rewards for agent k using an alternative α_{ik}.

To evaluate alternatives an average value of a_k is computed.

$$avga_k(x) = \frac{\sum_i a_k(x) \mid \alpha_{ik}}{I_k} \qquad (9)$$

where I_k is a number of alternatives associated with an agent k.

Then a difference between avg a_k and a value of $a_k(x) \mid \alpha_{ik}$ is calculated for all alternatives.

The weight of 1 is given to an alternative plan, where the $a_k(x) \mid \alpha_{ik}$ is the maximum.

Other alternatives are inhibited. They are given weights of 0 (zero).

For negotiation process it is necessary to establish weights that are corresponding to rewards, but in such a way that no alternative is given a weight of 0 (zero). In that case the negotiation process cannot be executed.

A relative weight of an individual agent alternative can be interpreted as a probability unit with which the alternative is chosen.

Let define a weight as follows:

$$w_k \mid \alpha_{ik} = \frac{a_k(x) \mid \alpha_{ik}}{\sum_i a_k(x) \mid \alpha_{ik}} \qquad (13)$$

It is clear that

$\sum_{k=1}^{n} w_k = 1$; for all agents k.

After weights are assigned to alternatives a negotiation process starts.

Each agent prefers an alternative with the highest weight. A global criterion for choosing an alternative is a global reward for the whole system, which is a function of rewards of all agents. A value of criteria function is calculated and if it is higher then an average value of criteria function for the whole system, negotiation process

stops. If it is less, agents are sequentially called to make another offer, that means, to admit an alternative with a worse results for its, but maybe the whole system results would be better.
Negotiation runs in loops. For the beginning the first agent is called to compromise and all other agents keep their highest bids, then the second one, etc. Such a way the negotiation doesn't require backing up all values to find a sub optimal solution.

Reinforcement learning in a process of negotiation

Reinforcement learning (RL) is learning from interactions in an uncertain environment. Usually the goal is not known. The learner has to pick an action and try it and then to choose, which of action yields the best rewards. The four basic sub elements of RL are [11]

- A policy – which is a function that associates a state to an action. It can be also a lookup table
- A reward function – as it is defined in Section V.
- A value function – opposite to a reward function that answers a question what is good now, a value function yields results in long run term. It is a prediction.
- An environmental model – it is not required.

A policy for a scheduling problem is a function, which associates an alternative of an agent to a proposed schedule using this agents as a ruling agent. The proposed schedule represents an action taken by the agent.

A general RL algorithm:
At each time t following steps are performed:

- For all actions a a reward is evaluated. Rewards are calculated for all proposed alternatives.
- For each state s a value function is computed
- The final output at time t is $Q(t) = \sum q(t)$
- A new state, with a highest reward is establish
- New alternatives for agents are calculated
- Go back to the first step until schedule for all agents are calculated.

The system is forced to choose the best solution in each loop.

A backup operator takes into account only action with the best reward. The states with many good actions are neglected, that is one of disadvantages of the algorithm [12].

V. AN ILLUSTRATIVE EXAMPLE

A manufacturing system with two jobs, each consists of three operations, is supposed. Three resources are available. Each job can be done on any resource.
A scheduling task is defined (according Eq. (4)):

$$j_1 : \{0;7;[(1;2)^1,(2;2)^1,(2;1)^2,(1;2)^2,(1;3)^3,(2;2)^3]\}$$
$$j_2 : \{0;10;[(2;3)^1,(1,2)^2,(2;4)^2,(1;3)^3,(2;4)^3]\}$$

A MAS consists of an agent of job 1 – AVU1, and. an agent of a job 2 – AVU2. In the beginning, agents calculate a schedule independently on each other. The goal is to find a schedule with a minimal makespan.

$$sch(j_1) = [(1;0)^1,(2;3)^2,(2;4)^3]$$
$$sch(j_2) = [(2,0)^1,(1,3)^2,(1,5)^3]$$

The calculated schedules represent preferred alternatives for each agent.
Then the goal states for each alternative are calculated.
A set of goal states is defined as follows:

$$CS_i(j) = \{cs_i(j); cs_i(j) \in S_i; \forall x_i \in X; j \in J\} \quad (14)$$

The goal for each agent is to reach the minimal makespan - C. The goal states represent schedules, when agents reach the minimal makespan:

$$AVU_2 \mid \alpha_2 : cs(j_2) = [(2;0)^1,(1;3)^2,(1;5)^3]$$
$$AVU_1 \mid \alpha_1 : cs(j_1) = [(1;0)^1,(2;2)^2,(2;3)^3]$$

	$C(sch(j_i))$	$C(cs(j_i))$
AVU1	6	5
AVU2	8	8

Table 1: Goals and makespan for a feasible schedule for AVU1 and AVU2

It is obvious from the Table 1, that the goal makespan and makespan for AVU1 is not the same. New alternatives are proposed. Agents propose new alternatives sequentially. New values are computed far all alternatives and all alternatives are evaluated according to a satisfactory function (15). The scheme is very simple for small amount of agents. In the case the system contains more agents, not every alternatives can be evaluated. In that case the agents propose new alternatives on the basis of their importance, which is given by weight. The most important agent proposes the new alternative first, then alternatives of other agents are adapted and new values are computed. New alternatives are evaluated and the next step is the less important agent propose the new alternative and the process repeat. If the alternative proposed by the second less important agent is worst as the previous alternative, the process stops. In case the alternative proposed by less important agent is better, than the next agent in order proposes new alternative and the process repeat, until the better solution is found. Such way not all alternatives are examined, but the solution can be considered as a suitable alternative for all agents.
First is the alternative where alternative of AVU2 remains and new alternative for agent AVU1 is proposed.

$$AVU_2 \mid \alpha_{1nova} : sch(j_1) = [(1;0)^1,(2;3)^2,(2,4)^3]$$

Then alternative of an agent AVU2 remains, and alternative for an agent AVU1 changes:

$$AVU_2 \mid \alpha_{2nova} : sch(j_2) = [(2;5)^1,(1;8)^2,(2;10)^3]$$

Goals for new proposed alternatives are calculated:

$$C(sch(j_2)) \mid \alpha_{1nova} = 7$$

$$C(sch(j_1)) \mid \alpha_{2nova} = 13$$

A simple satisfactory function using fuzzy theory is defined:

$$H_{\alpha_1\alpha_{2nova}} = \left\{ \begin{array}{l} 1 - \frac{|C(sch(j_1)) - C(cs(j_1))|}{C(cs(j_1))} \Big/ AVU_1 ; \\ 1 - \frac{|C(sch(j_2)) - C(cs(j_2))|}{C(cs(j_2))} \Big/ AVU_2 ; \\ 1 - \frac{|C(sch) - C(cs)|}{C(cs)} \Big/ AVU \end{array} \right\} \quad (15)$$

Using the above values, satisfactory functions are calculated:

$$H_{\alpha_1\alpha_{2nova}} = \left\{ 1\big/ AVU_1 ; 0{,}38\big/ AVU_2 ; 0{,}38\big/ AVU \right\}$$

$$H_{\alpha_{1nova}\alpha_2} = \left\{ 0{,}6\big/ AVU_1 ; 1\big/ AVU_2 ; 1\big/ AVU \right\}$$

>From the satisfactory function a wining alternative is determined as a new schedule for AVU1 and remaining schedule for AVU2.

VI. CONCLUSION

The paper discusses some new approach in scheduling problem solution – utilization of MAS. This approach is very suitable for product-oriented manufacturing, which is very important for satisfying user's requirements. MAS suppose flexible manufacturing systems. The presented approach enables dynamic scheduling and handling with alternatives of agents. The future improvements is in using holonic systems, which represent a new approach to answer new trends in production systems – low-volume-high-variety production. Theoretically holonic-manufacturing systems aroused from distributed artificial intelligence and from multi-agent systems. Comparing to traditional manufacturing systems, HMS are controlled by interaction among holons. Holon presents an autonomous entity, which consists of a resident machine, and a software agent.

Advantages of HMS in comparison with traditional manufacturing systems are:

- robustness – holons can restart and reschedule their manufacturing tasks according to current situation
- adaptability and flexibility – users can control holons' behavior whenever it is needed
- effectivity – holons can balance load among themselves and that way to increase utilization of resources and minimize the idle time of them

The problem discussed in this paper – negotiation among agents is very actual also for holonic manufacturing systems.

VII. ACKNOWLEDGMENT

This work was partially supported by a Science and Technology Assistance Agency under the contract No. APVT-51-011602 and a VEGA agency, under project No. 2/4148/04.

VIII. REFERENCES

For a paper citation:

[1] B. Frankovič, I. Budinská: A framework for manufacturing scheduling within production systemy. In Journal of Modelling and Control, STU Bratislava, Vol. 4, 2001

[2] Rudas J. (1999): Evolutionary operators; new parametric type operator families. *Int. Journal of Fuzzy Systems*, **Vol. 23, No. 2**, 147-166.

For a book citation:

[3] B.Frankovič, T.T.Dang: Scheduling of production using the multi-agent approach by hierarchical structures. Monostori L., Váncza J., Ali Moonis (Eds.): Engineering of Intelligent Systems, Springer, 2001, ISBN 3-540-42219-6 str. 622-632

For a conference citation:

[4] Dang T.Tung, Frankovič B., Budinská I.: Case-based reasoning applied for CAS-Decision system, In Proc. of the 2nd IFAC Conference on Control System Design. Bratislava, Slovakia, September, 7-10, 2003. Š. Kozák, M. Huba (Eds.). Bratislava, Slovakia, 2003. 6 pages; Compact Disc, Section C: DEDS Control System Design.

[5] Dang T.T., B. Frankovič: Agent based scheduling in production systems. In: International Journal of Production Research, vol. 40, No. 15, 2002 ISSN 0020-7543 print/ISSN 1366-588X online, Taylor&Francis Ltd., p. 3660-3679 CCL.

[6] Dang T.T., Frankovič B., Budinská I.: Agents Coalition in Coordination process. In: Proceedings of the 15-th IFAC Congress B'02, Barcelona, July 19-26, 2002, (CD).

[7] Dang T.T., Frankovič B., Budinská I.: Create agents' Coalition Based on a Dynamic Programming Approach. In: Proceedings of the ECAI 02, Workshop "Agent Technologies and Logistics", July, 2002, Lyon, France str. 16-24

[8] Dang T.T.: A Polynomial algorithm for single machine batch processing. Applied for publishing in Int. Journal of Production research.

[9] Dang, T.T., Hluchy L., Nguyen T. G., Budinska I. Balogh Z., Laclavik M.: Knowledge Management and Data Classification in Pellucid, In Proc. of Intelligent Information Processing and Web Mining 2003 (IIPWM'03), Zakopane, Poland, June 2-5, 2003, ISSN 1434-9922, pp. 563-568, Springer-Verlag

[10] Frankovič B.: "Aplikácia prístupov teórie multiagentových systémov v informatizácii spoločnosti". In Proceedings of the conference The Mission of Universities and Science in the 21st Century , TrenčianskeTeplice, 27.- 28. júna 2002, Slovakia, (CD) in Slovak

[11] Frankovič B.: Hierarchické metódy rozvrhovania úloh v diskrétnych výrobných systémoch, Časopis SSKI: Journal Cybernetics and Cybernetics and Informatics, Vol. 2, 2003, 6 pages (Internet journal http://www.kasr.elf.stuba.sk/sski/casopis/)

[12] Maturana, F., Shen, W., Hong, M. and Norrie, D. H.: Multi-agent Architectures for Concurrent Design and Manufacturing. In Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, 355-359, 1997

[13] Perugini, D., Lambert, D., Sterling, L., Pearce, A.: Agents for military logistic planning In. Proc. 15th European Conference on Artificial Intelligence, Workshop 20. Lyon p. 35-44

[14] Cantamessa, M. and Villa, A.: Negotiation Models and Production Planning for Virtual Enterprises. In Proceeding of The IFAC Workshop on Manufacturing, Modeling, Management and Control – MIM 2001, Prague, Czech Republic, 1-5, 2001.

[15] Sutton, S.R., 1991. Planning by incremental dynamic programming. In Proceedings Ninth Conference on Machine learning, 353 –357, Morgan-Kaufman

[16] Satinder, P.Singh., 1994. Learning to solve Markovian decision processes. A dissertation thesis, University of Massachusetts

For other citations:

[17] Dang, T.T.: Methods of creating agent coalition for solving planning problems in flexible manufacturing systems, Doctoral Thesis, 2003