

Autonomous Planning Framework for Distributed Multiagent Robotic Systems

Marko Švaco, Bojan Šekoranja, and Bojan Jerbić

University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture,
Department of Robotics and Production System Automation
Ivana Lučića 5, 10000 Zagreb, Croatia
{marko.svaco,bojan.sekoranja,bojan.jerbic}@fsb.hr

Abstract. In this paper a creative action planning algorithm (CAPA) is presented for solving multiagent planning problems and task allocation. The distributed multiagent system taken in consideration is a system of m autonomous agents. Agents workspace contains simplified blocks which form different space structures. By employing the planning algorithm and through interaction agents allocate tasks which they execute in order to assemble the required space structure. The planning algorithm is based on an inductive engine. From a given set of objects which can differ from the initial set agents need to reach a solution in the anticipated search space. A multiagent framework for autonomous planning is developed and implemented on an actual robotic system consisting of three 6 DOF industrial robots.

Keywords: Distributed robotic system, autonomous planning, multiagent system, assembly, industrial robotics.

1 Introduction

Substantial research and development is conducted to multiagent robotics; particularly in the fields such as service, humanoid or mobile robotics, but industrial robotics is still based on traditional postulates. Real flexibility and adaptivity to changes are shortcomings in today's industrial assembly and handling robotic applications and are issues that need to be addressed. Distributed multiagent robotics is a system based on human behavior patterns. When complex tasks arise humans are much more efficient when working in groups: they exhibit more axis of freedom, more data can be handled and they delegate particular tasks to individual agents.

Research concentrated around humanoid robotics ([1]-[2]) is developing rapidly. Dual arm configuration highly sophisticated perceptive mechanisms, human like motions enable robots to recreate human motion and work patterns. Major drawback of those kinematical structures is very low repeatability and precision primarily needed in industrial systems. For assembly and handling tasks which usually have high precision and repeatability demands industrial robots are necessary. Nowadays the most flexible industrial robots have 6 or 7 [3] degrees of freedom (DOF) without the end effector (gripper) which usually has 1 additional DOF. One human arm (with the hand) has 27 DOF [4]. The flexibility of a robotic arm is quite limited in

comparison to a human operator. Implementing two or more robots with own controllers that communicate each with other, a certain multiagent concept can be achieved. The whole system will be orchestrated and will be able to perform more demanding operations. Each controller running its own actuator unit should be an agent with defined level of autonomy. In such systems the multiagent control appears as the main issue.

In this paper a creative action planning algorithm (CAPA) for application in multi agent robotic systems is presented. One of the main goals is constructing a universal planning framework which can be implemented on various types of industrial robots and tasks.

Related works [5], [6] incorporating multiagent planning on similar tasks are virtual applications and cannot be easily implemented on real industrial systems. The approaches are primarily intended for autonomous planning done by multiple agents who cannot collide, are of infinite small dimension and share the same computational time domain. The developed CAPA and the distributed multiagent system (MAS) operate in a real world environment bounded by rules and limitations. The approach discussed in this paper is intended to show that some assembly and handling tasks can be done in close collaboration among agents to gain flexibility and increase overall system productivity.

2 Contribution to Sustainability

Robotics and in particular industrial robotics have always been a part of a central planning system. Agents (robots, machines) controlled by own computers are somehow subordinated to a central system controller [7]. Therefore they exhibit very low level of autonomy and in most cases do pre-programmed actions not being able to cope with uncertainties in the system and the environment. Uncertainties may vary from production quantities to failures of equipment or other agents, etc.

It is suggested that some handling and assembly industry tasks can be accomplished by interaction between agents (primarily industrial robots) in the system. Accordingly some level of autonomy must be introduced.

Production in recent years has switched from high quantity standardized products to lower quantities of customized products so demands from assembly systems have grown. Traditional approach with a centralized architecture and strict delegation of tasks needs to be replaced. Introducing a multiagent configuration and autonomous planning approach could be proven as a valuable addition. For an assembly system it implies that agents (robots and machines) before assembling need to generate a plan that best suits the current state and requirements of the system. After deriving consensus agents begin assembling the structure (product) constantly communicating and exchanging relevant information and data. In industrial assembly systems this is a novice approach and it has numerous benefits when implemented: it leads to increased flexibility and adaptivity to unexpected changes and uncertainties in the system, i.e. responsiveness [8]. The system becomes insensitive to number of agents (robots) and new assembly tasks can be resolved with less effort. Clearly, this approach is not suited for all products but it can be implemented on a variety of industrial examples. Development of such an industrial system scheme is beyond the scope of this paper and will be considered for further research. In this work an initial version

of the planning framework is presented. The framework is implemented on an actual system consisting of three 6 DOF robots.

3 The Multiagent System

3.1 System Formulation

The multiagent system consists of m autonomous agents a_l ($l = 1...m$). Their workspace contains simplified blocks with respect to a global Cartesian coordinate system K . Agents workspace $W(a_i, b_{j,k})$ contains blocks which form different space structures, where $b_{j,k}$ represents j^{th} block of k^{th} type. Each block has certain properties which agents perceive from their workspace: *size* (type) of a building block $T(b_{j,k}) = \{1, 2, 3...\}$ and *Cartesian position and orientation* in workspace: $P(b_{j,k}) = \{x, y, r\}$. All blocks have the same width and height (single unit) but their length can vary and can be one, two, three, etc. unit lengths. That results with flexibility so building blocks can be supplemented with each other i.e. block with two unit lengths can be replaced with two blocks of single unit length and vice versa. Each agent is defined as an autonomous, self-aware entity with limited knowledge of the global workspace [9] and with some cognition of other agents. It has a separate processing unit, actuators, vision system for acquiring information from its environment, force and torque sensors for haptic feedback and other interfaces. A space function $F(a_i)$ is defined to determine the consumed space by an agent a_i , $F = (x_1, y_1, x_2, y_2, r, t)$ in time t . The first pair of Cartesian coordinates depicts the first vertex of a rectangle which bounds the agent and the second pair depicts the second vertex respectively. Rotation angle r is defined with respect to the origin point of the coordinate system K .

The MAS is insensitive to dynamic changes in number of agents. Impact is lower system flexibility and longer times for achieving final goals when agents are excluded from the system.

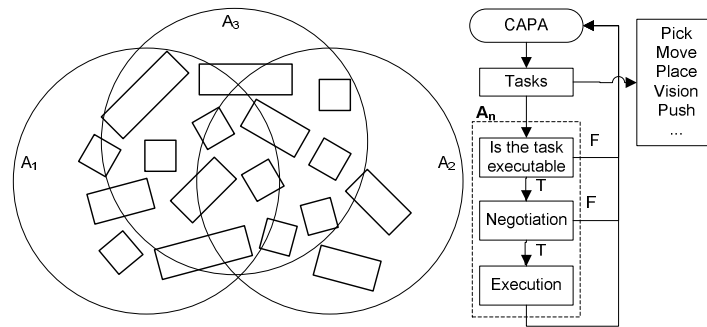


Fig. 1. Agent workspace and task allocation scheme

3.2 Structures and Decision Making

Agents' tasks are recreating structures which are defined as a final form put together from various objects with defined relationships. A structure is determined by

interrelations and arrangement of objects $b_{j,k}$ into a complex entity. Structure $S = \{R_i b_{j,k}\}$ is a set of relations R_i ($i = 1 \dots m-1$) between objects ($b_{j,k}$, $j = 1 \dots n$, $k = 1 \dots u$).

The MAS has properties of a market organization type [10], [11] where agents bid [12] for given resources (blocks) in their workspace (Fig. 1). Time schedules need to be negotiated when areas of interest in the global workspace are not occupied.

Global goal G is the required structure that must be assembled from available elements following the given set S . An example of a structure is illustrated in Fig. 2 a). After observing a structure and finding relations agents are given an arbitrary set of work pieces (blocks) as depicted in Fig. 2. (b). Using those elements a plan of actions is generated for assembling the initial structure. Possible solutions are presented in Fig. 2. (c₁) - (c₃). A set of rules and propositions for agent behavior is given in a cognition base (CB):

- Mathematical rules for structure sets
- Agents capabilities
- Grasping rules and limitations
- Object properties
- Agent workspace
- Vision system patterns database
- Force and torque sensor threshold values

If a simple structure with limited number of building blocks is presented to the agents (Fig. 2 a) there might be only one or few feasible solutions (sequence of steps). If more complex structures are presented (as shown in Fig. 4. a) a variety of feasible solutions might be possible.

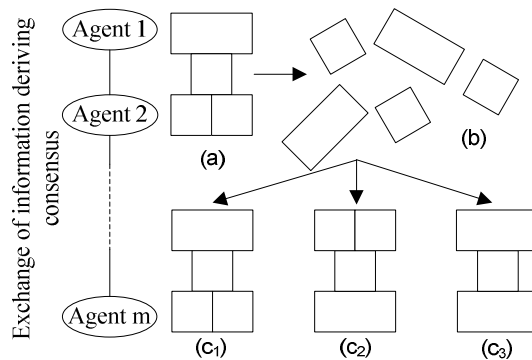


Fig. 2. Simple task for the multiagent system

Top down disassembling or bottom up assembling the structure can define a sequence of steps for the MAS. The CAPA utilizes a bottom up principle where from a provided set of objects $\{b_1 \dots b_{p,r}\}$, which can differ from the initial set $\{b_1 \dots b_{n,q}\}$ agents need to reach a solution in the given search space. Depending on the CB information agents can make decisions whether the desirable objectives can be performed in accordance to proposed restrictions and limitations. Implementing an

iterative algorithm a solution can be found as shown in Fig. 3. Branches represent solution sets and each branch leads to one solution. If finding a solution in one solution set isn't possible, the system takes one step back and explores other options until it finds a valid one.

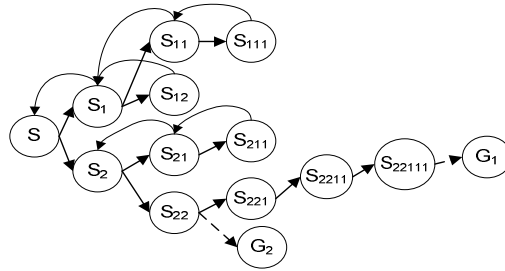


Fig. 3. Possible solution sets (sequence of steps) for a represented structure

Each agent attains a unified set of sub goals g_t which fulfill the global goal G . Execution of sub goals (tasks) can be done synchronous or asynchronous giving the space functions F of the agents. A resource function C is defined as a measure of resource and time consumption. $C(a_i, b_{j,k}, e)$ is a function of agents' a_i position, specifications of a building block $b_{j,k}$ (size and position in global workspace) and the position e where that block is planned to be moved.

3.3 Operators

By utilizing operators agents construct a sequence of actions for accomplishing each sub goal. By consecutively achieving all sub goals the global goal G is fulfilled and the agents await further tasks. The basic operators are:

- *Pick* (b_i, gr_k) – agent picks up a block b_i with a grasping method gr_k
- *Move* ($p_1, \dots, p_r, t_1, \dots, t_r$) – agent moves in the global workspace from point p_1 to point p_r through $r-2$ interpolation points with motion specification t_r defined for each point.
- *Place* ($b_{j,k}$) - agent places a block $b_{j,k}$
- *Vision* – vision operator is used for identifying objects and their coordinates in c
- *Push* (f, d, s) – agent uses force/torque sensor for auxiliary action of pushing an object with force/torque threshold t in vector direction d for s units
- *Force* – used for positioning correction

The vision operator utilizes the cognition base and solves problems of identifying work objects and associated data. Therefore vision processes have to be very stable and work in constantly changing light and scenery conditions [13]. A fix to this problem is to utilize algorithms that can change the exposition of the image acquisition process. This can be done through a way of search patterns. Few images are taken at different camera settings and the one where familiar objects are recognized is used as

reference. The downside is increase of image acquisition and processing time. If light and scenery conditions can vary this is a necessity due to the high level sensitivity of precision vision applications.

Furthermore if there is a need for even higher precision beyond capabilities of vision systems agents can use very sensitive force sensors. This method improves accuracy and corrects the pick&place positions. To determine which method to apply agents rely on vision identification of objects. Once the object is identified agents can decide which method of force correction to apply. In example they can successfully insert a shaft or a square object into adequate holes if their original position was slightly off the required one. Furthermore the force sensor allows an agent to correct larger errors. A controlled search pattern is used with a very low force not to damage the objects. Finding the adequate insertion position completes the process.

3.4 Global System Approach

The starting point of every assembling process is perception of the agent's environment. Each agent uses vision systems to acquire information from a portion of the global workspace and forwards it to the planning agent. From the global information set the planning agent extracts relations between objects forming the initial structure. The same principle is applied for the random set of work objects. Regarding the initial structure and available elements the planning agent decomposes the global goal into tasks which can be performed by an individual agent. Task priorities are also taken into consideration where some tasks are conditioned to be executed before others. After this initial process each agent bids for a task. Through comparing resource functions agents submit the task to the optimal candidate. Idle agents repeat this process and acquire free tasks. When processing a task an agent sends data regarding the consumed space through the F function for collision avoidance. After all task are allocated and executed agents inspect the reassembled structure.

4 Implementation

The CAPA has been tested to provide solutions for a structure such as the one shown in Fig. 4. When multiple solutions are possible the MAS executes the one where $\sum C$ in the entire solution set is minimal. Currently only two dimensional structures (R^3) are being solved but their solutions due to use of real world objects has to be three dimensional (R^4). The planning algorithm was tested on a virtual model of the multi-agent robotic system (Fig. 5. a). This was done for safety reasons (primarily collision) and the ability to test and debug the algorithm in parallel on multiple computers. After satisfactory computational results the algorithm and the entire framework have been implemented on an actual robotic system – Fig. 5. (b).

The first problem which emerged was sharing of agent workspace. In order to work on the same task, assembling the same structure, spatial relations need to be taken into consideration. Agents were calibrated using calibration tools and visual applications.

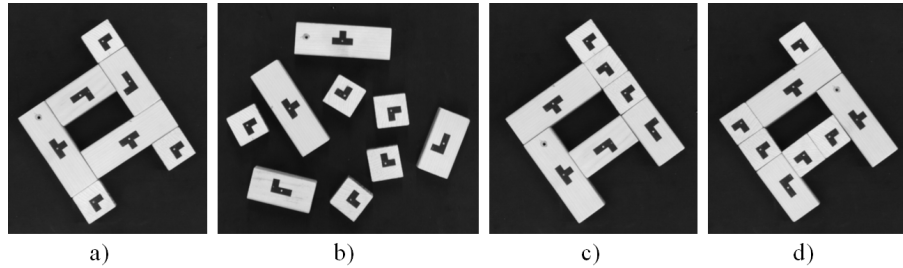


Fig. 4. a) Initial space structure b) Randomly scattered building blocks c), d) Space structures assembled by the agents

This creates relations with respect to agent positions (three translations) and rotations (three angular displacements); introducing a common global workspace (K). A problem that resulted from the decentralized multiagent architecture was sharing and synchronizing agent time domains. This didn't introduce an issue while tests were conducted on a computer where all agents used the same *CPU clock*. Adjustments have been done using handshaking with digital signals and through TCP/IP communication which allowed coordinated task execution.

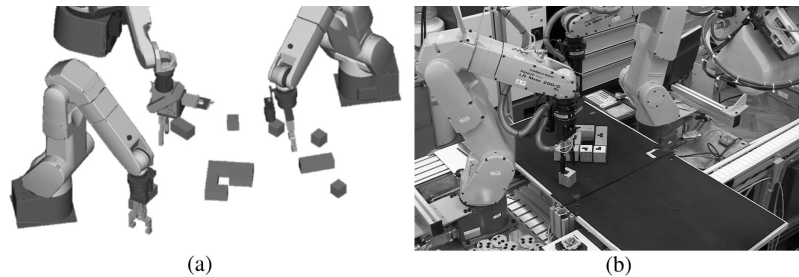


Fig. 5. (a) virtual representation of the multiagent robotics system (b) real agents

Collision detection was an issue that needed to be addressed. Currently there are no algorithms to solve real time agent collision or they exist but with limitations. Collision between two agents with kinematic chains of 3 DOF can be solved in a definite period of time [14]. For the reason of limited computational power and the collision detection not being the centre of this research the function (F) described in chapter 3 was used.

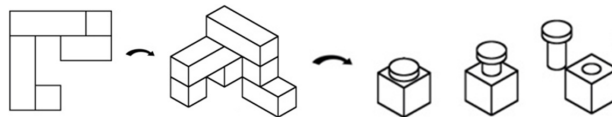


Fig. 6. Planning of 3D structures, following research direction

5 Conclusion and Further Research

The approach presented in this paper ensures higher system robustness regarding decentralized task execution. In future applications agents would decide how to best solve a new problem – which agent has the most adequate tools and how can the rest of them assist etc. Making agents mobile and giving them the ability to exchange tool heads introduces even a greater level of flexibility to the system. This results in more cost optimized solutions. Further generalization will be introduced where agents will be able to autonomously distinguish and solve entirely new problems as illustrated in Fig. 6. First step is implementation of reassembling 3D structures. For that purpose the CB will need to comprise rules regarding “laws of gravity” and etc. Further research will be concentrated on introduction of new objects to the MAS. Taking into consideration the CB (known similar objects) agents will be able to find or construct grasping methods weather they can do it individually or assisted by other agents.

References

1. Akachi, K., Kaneko, K., Kanehira, N., Ota, S., Miyamori, G., Hirata, M., Kajita, S., Kanehiro, F.: Development of humanoid robot hrp-3. In: 5th IEEE/RAS International Conference on Humanoid Robots, pp. 50–55 (2005)
2. Park, I.W., Kim, J.Y., Lee, J., Oh, J.H.: Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot-3: Hubo). In: IEEE/RAS International Conference on Humanoid Robots, pp. 321–326 (2005)
3. Bischoff, R.: From research to products: The development of the KUKA Light-Weight Robot. In: 40th International Symposium on Robotics, Barcelona, Spain (2009)
4. Agur, A.M.R., Lee, M.J.: Grant’s Atlas of Anatomy. Lippincott Williams and Wilkins, Baltimore (1999)
5. Sycara, K., Roth, S., Sadeh, N., Fox, M.: Distributed Constrained Heuristic Search. *IEEE Trans. on System, Man and Cybernetics*, 1446–1461 (1991)
6. Ephrati, E., Rosenschein, J.: Divide and conquer in multiagent planning. In: Proc. of the 12th National Conference on AI, pp. 375–380. AAAI, Seattle (1994)
7. Tang, H.P., Wong, T.N.: Reactive multi-agent system for assembly cell control. *Robotics and Computer-Integrated Manufacturing* 21, 87–98 (2005)
8. Seilonen, I., Pirttioja, T., Koskinen, K.: Extending process automation systems with multi-agent techniques. *Engineering Applications of Artificial Intelligence* 22 (2009)
9. Schumacher, M.: *Objective Coordination in Multi-Agent System Engineering*. Springer, New York (2001)
10. Sandholm, T.: An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. In: Proc. of the 11th Conference on AI, pp. 256–262 (1993)
11. Shoham, Y., Leyton-Brown, K.: *Multiagent Systems: algorithmic, game-theoretic and logical foundations*. Cambridge Uni. Press, New York (2009)
12. Hsieh, F.-S.: Analysis of contract net in multi-agent sys. *Automatica* 42, 733–740 (2006)
13. Stipancic, T., Jerbic, B.: Self-adaptive Vision System. In: Camarinha-Matos, L.M., Pereira, P., Ribeiro, L. (eds.) DoCEIS 2010. IFIP Advances in Information and Communication Technology, vol. 314, pp. 195–202. Springer, Heidelberg (2010)
14. Curkovic, P., Jerbic, B.: Dual-Arm Robot Motion Planning Based on Cooperative Coevolution. In: Camarinha-Matos, L.M., Pereira, P., Ribeiro, L. (eds.) DoCEIS 2010. IFIP Advances in Information and Communication Technology, vol. 314, pp. 169–178. Springer, Heidelberg (2010)